LNCIS

Lecture Notes in Control and Information Sciences

358

Rolf Findeisen
Frank Allgöwer
Lorenz T. Biegler (Eds.)

Assessment and Future Directions of Nonlinear Model Predictive Control

Springer

Lecture Notes in Control and Information Sciences 358

Rolf Findeisen, Frank Allgöwer,
Lorenz T. Biegler (Eds.)

Assessment and Future Directions of Nonlinear Model Predictive Control

Springer

Editors

Dr.-Ing. Rolf Findeisen

Institute for Systems Theory and Automatic
Control
University of Stuttgart
Pfaffenwaldring 9
70550 Stuttgart
Germany
E-mail: findeise@ist.uni-stuttgart.de

Prof. Dr. Lorenz T. Biegler

Chemical Engineering Department
Carnegie Mellon University
Pittsburgh, PA 15213
USA
E-mail: lb01@andrew.cmu.edu

Prof. Dr. Frank Allgöwer

Institute for Systems Theory and Automatic
Control
University of Stuttgart
Pfaffenwaldring 9
70550 Stuttgart
Germany
E-mail: allgower@ist.uni-stuttgart.de

Library of Congress Control Number: 2007927163

ISSN print edition: 0170-8643
ISSN electronic edition: 1610-7411
ISBN-10 3-540-72698-5 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-72698-2 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

Typesetting: by the authors and SPS using a Springer LATEX macro package

Printed on acid-free paper SPIN: 12067183 89/SPS 5 4 3 2 1 0

Preface

The past three decades have seen rapid development in the area of model predictive control with respect to both theoretical and application aspects. Over these 30 years, model predictive control for linear systems has been widely applied, especially in the area of process control. However, today's applications often require driving the process over a wide region and close to the boundaries of operability, while satisfying constraints and achieving near-optimal performance. Consequently, the application of linear control methods does not always lead to satisfactory performance, and here nonlinear methods must be employed. This is one of the reasons why *nonlinear model predictive control* (NMPC) has enjoyed significant attention over the past years, with a number of recent advances on both the theoretical and application frontier. Additionally, the widespread availability and steadily increasing power of today's computers, as well as the development of specially tailored numerical solution methods for NMPC, bring the practical applicability of NMPC within reach even for very fast systems. This has led to a series of new, exciting developments, along with new challenges in the area of NMPC.

In order to summarize these recent developments, and to consider these new challenges, we organized an international workshop entitled "Assesment and Future Directions of Nonlinear Model Predictive Control" (NMPC05), which was held at the Waldhotel Zollernblick, in Freudenstadt-Lauterbad, Germany on August 26-30, 2005. The objective of this workshop was to bring together a diverse group of internationally recognized researchers and industrial practitioners in the area of NMPC, in order to critically assess and discuss the current status, future directions and open questions of NMPC. The number of participants was intentionally kept small in order to promote discussions and the fruitful exchange of ideas. In the spirit of the very successful predecessor workshop held in 1998 in Ascona, Switzerland, all the keynotes, as well as the main talks were given by invited speakers. There were also a limited number of contributed oral and poster presentations. Overall the workshop turned out to be very stimulating and allowed close interactions and discussions among the participants.

This volume contains a selection of papers from this workshop that summarize the key results and challenges of NMPC. We hope that it provides a useful reference, as well as inspiration for future research in this area.

We would like to thank all of the authors for their participation and their interesting contributions to the workshop. Likewise, we are grateful to all of the reviewers involved in the pre- and post-reviews of the contributions. They provided invaluable comments, which ensured the high quality of this book volume. Moreover, the workshop itself, as well as the production of this volume, would not have been possible without the financial support of the Network of Competence: Pro3-Process Technology. We would also like to thank all members of the Institute for Systems Theory and Automatic Control for their help in organizing and running the workshop. Finally, we are especially thankful to Dr. Thomas Ditzinger of the Springer Verlag for his support of this volume.

Stuttgart, Pittsburgh
December 2006

Rolf Findeisen
Frank Allgöwer
Lorenz T. Biegler

Contents

Foundations and History of NMPC

Theoretical Aspects of NMPC

Numerical Aspects of NMPC

Robustness, Robust Design, and Uncertainty

State Estimation and Output Feedback

Industrial Perspective on NMPC

NMPC and Process Control

NMPC for Fast Systems

Novel Applications of NMPC

Distributed NMPC, Obstacle Avoidance, and Path Planning

Nonlinear Model Predictive Control: An Introductory Review

Eduardo F. Camacho and Carlos Bordons

Dept. de Ingenieria de Sistemas y Automatica. University of Seville, Spain
{eduardo, bordons}@esi.us.es

1 Linear and Nonlinear Model Predictive Control

Model Predictive Control (MPC) originated in the late seventies and has developed considerably since then. The term Model Predictive Control does not designate a specific control strategy but rather an ample range of control methods which make explicit use of a model of the process to obtain the control signal by minimizing an objective function. The ideas, appearing in greater or lesser degree in the predictive control family, are basically the explicit use of a model to predict the process output at future time instants (horizon), the calculation of a control sequence minimizing an objective function and the use of a receding strategy, so that at each instant the horizon is displaced towards the future, which involves the application of the first control signal of the sequence calculated at each step.

The success of MPC is due to the fact that it is perhaps the most general way of posing the control problem in the time domain. The use a finite-horizon strategy allows the explicit handling of process and operational constraints by the MPC.

From the end of the 1970s various articles appeared showing an incipient interest in MPC in industry, principally the Richalet *et al.* publications [39][40] presenting Model Predictive Heuristic Control (MPHC) (later known as Model Algorithmic Control (MAC)) and those of Cutler and Ramakter [10] with Dynamic Matrix Control (DMC). A dynamic process model is explicitly used in both algorithms (impulse response in the first and step response in the second) to predict the effect of the future control actions at the output; these are determined by minimizing the predicted error subject to operational restrictions. The optimization is repeated at each sampling period with up-to-date information about the process. These formulations were heuristic and algorithmic and took advantage of the increasing potential of digital computers at the time.

MPC quickly became popular, particularly in chemical process industries, due to the simplicity of the algorithm and to the use of the impulse or step response model which, although possessing many more parameters than the formulations

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 1–16, 2007.
springerlink.com

in the state space or input-output domain, is usually preferred as being more intuitive and requiring less *a priori* information for its identification.

Another line of work arose independently around adaptive control ideas, developing strategies essentially for monovariable processes formulated with input-output models. Some examples of these strategies are Extended Prediction Self Adaptive Control (EPSAC) by De Keyser and Van Cuawenberghe [20] or Generalized Predictive Control (GPC) developed by Clarke *et al.* in 1987 [9].

MPC is considered to be a mature technique for linear and rather slow systems like the ones usually encountered in the process industry. More complex systems, such as nonlinear, hybrid, or very fast processes, were considered beyond the realm of MPC. During the last few years some impressive results have been produced in these fields. Applications of MPC to nonlinear and to hybrid processes have also appeared in the literature. The majority of applications (see surveys by Qin and Badgwell [35] [36]) are in the area of refining, one of the original application fields of MPC, where it has a solid background. An important number of applications can be found in petrochemicals and chemicals. Although MPC technology has not yet penetrated deeply into areas where process nonlinearities are strong and frequent changes in operation conditions occur, the number of nonlinear MPC applications is clearly increasing.

In general, industrial processes are nonlinear, but many MPC applications are based on the use of linear models. There are two main reasons for this: on one hand, the identification of a linear model based on process data is relatively easy and, on the other hand, linear models provide good results when the plant is operating in the neighbourhood of the operating point. Besides, the use of a linear model together with a quadratic objective function gives rise to a convex problem whose solution is well studied with many commercial products available. Notice that MPC of a linear plant with linear constraints gives rise to a nonlinear controller, and that this combination of linear dynamics and linear constraints has influenced on the commercial success of MPC. However, the term *Nonlinear* MPC is used for Predictive Controllers that make use of a nonlinear dynamic model (and therefore nonlinear constraints) and gives rise to the extra complexity.

In many situations the operation of the process requires frequent changes from one operation point to another and, therefore, a nonlinear model must be employed. The use of Nonlinear Model Predictive Control (NMPC) is justified in those areas where process nonlinearities are strong and market demands require frequent changes in operation regimes. Although the number of applications of NMPC is still limited (see [3], [36]), its potential is really great and MPC using nonlinear models is likely to become more common as users demand higher performance and new software tools make nonlinear models more readily available.

From a theoretical point of view using a nonlinear model changes the control problem from a convex QP to a non-convex Non-Linear Program (NLP), the solution of which is much more difficult. There is no guarantee, for example, that the global optimum can be found.

2 Nonlinear Models

Nonlinear dynamics are present in nearly all engineering applications. Developing adequate nonlinear models may be very difficult and there is no model form that is clearly suitable to represent general nonlinear processes. Nonlinear models are difficult to construct, either from input/output data correlation or by the use of first principles from well-known mass and energy conservation laws.

The three main types of models that are used in this area are: empirical, fundamental (that come directly from balance equations, usually called first principle models), and grey box (developed by combining the empirical and fundamental approaches, exploiting the advantages of each type of model). Empirical models are discussed below.

A fundamental difficulty associated with the empirical modelling approach is the selection of a suitable model form. The available nonlinear models used for NMPC are described below, divided into two main classes: input-output and state-space.

2.1 Input-Output Models

The nonlinear discrete-time models used for control can be viewed as mappings between those variables that are available for predicting system behaviour up to the current time and those to be predicted at or after that instant. This kind of model can be represented as a nonlinear autoregressive moving average model with exogenous input (NARMAX), which, for single-input single-output processes, is given by the general equation

$$y(t) = \Phi[y(t-1), \ldots, y(t-n_y), u(t-1), \ldots, u(t-n_u), e(t), \ldots, e(t-n_e+1)] \quad (1)$$

where Φ is a nonlinear mapping, y is the output, u is the input, and e is the noise input. The suitability of this model depends on the choice of the function Φ and the order parameters. Notice that this equation covers a wide range of descriptions, depending mainly on function Φ. Volterra and related models, local model networks and neural networks are detailed in this subsection.

Volterra Models

If only Finite Impulse Response (FIR) models are considered and Φ is restricted to analytic functions, it follows that this function exhibits a Taylor series expansion which defines the class of discrete-time Volterra models. Although Volterra models have their limitations, they represent a simple logical extension of the convolution models that have been so successful in linear MPC. These models are generically well-behaved and their structure can be exploited in the design of the controller. In this particular and useful case, and when the infinite terms are truncated to finite values, the process model is given by

$$y(t) = y_0 + \sum_{i=0}^{N} h_1(i)u(k-i) + \sum_{i=0}^{M}\sum_{j=0}^{M} h_2(i,j)u(t-i)u(t-j) \quad (2)$$

which corresponds to the widely used linear convolution model with the nonlinearity appearing as an extra term, that is, the nonlinearity is additive.

Two special subclasses of the basic model are employed which reduce the complexity of the basic Volterra approach and have a reduced number of parameters. These are the Hammerstein and Wiener models. Hammerstein models belong to the family of block-oriented nonlinear models, built from the combination of linear dynamic models and static nonlinearities. They consist of a single static nonlinearity $g(.)$ connected in cascade to a single linear dynamic model defined by a transfer function $H(z^{-1})$. Because of this, Hammerstein models can be considered *diagonal* Volterra models, since the off-diagonal coefficients are all zero. Notice that this means that the behaviour that can be represented by this type of model is restricted. The Wiener model can be considered as the dual of the Hammerstein model, since it is composed of the same components connected in reverse order. The input sequence is first transformed by the linear part $H(z^{-1})$ to obtain $\Psi(t)$, which is transformed by the static nonlinearity $g(.)$ to get the overall model output. The properties of Volterra and related models are extensively discussed in [11].

Closely related to Volterra models are bilinear models. The main difference between this kind of model and the Volterra approach is that crossed products between inputs and outputs appear in the model. Bilinear models have been successfully used to model and control heat exchangers, distillation columns, chemical reactors, waste treatment plants, and pH neutralisation reactors [16]. It has been demonstrated that this type of model can be represented by a Volterra series [23].

Local Model Networks

Another way of using input-output models to represent nonlinear behaviour is to use a local model network representation. The idea is to use a set of local models to accommodate local operating regimes [17], [44]. A global plant representation is formed using multiple models over the whole operating space of the nonlinear process. The plant model used for control provides an explicit, transparent plant representation which can be considered an advantage over *black-box* approaches such as neural networks (that will be described below).

The basics of this operating regime approach are to decompose the space into zones where linear models are adequate approximations to the dynamical behaviour within that regime, with a trade-off between the number of regimes and the complexity of the local model. The output of each submodel is passed through a local processing function that generates a window of validity of that particular submodel. The complete model output is then given by

$$y(t+1) = F(\Psi(t), \Phi(t)) = \sum_{i=1}^{M} f_i(\Psi(t))\rho_i(\Phi(t))$$

where the M local models $f_i(\Psi(t))$ are linear ARX functions of the measurement vector Ψ (inputs and outputs) and are multiplied by basis functions $\rho_i(\Phi(t))$ of the current operating vector. These basis functions are chosen to give a value close to 1 in regimes where f_i is a good approximation to the unknown F and a value close to 0 in other cases.

Notice that this technique allows the use of a linear predictive controller, avoiding the problems associated to computation time and optimality of the nonlinear solution. These strategies have been successfully tested on a pH neutralizsation plant (see [45] for details).

Neural Networks

The nonlinear dynamics of the process can also be captured by an artificial Neural Network (NN). Neural networks are attractive tools to construct the model of nonlinear processes since they have an inherent ability to approximate any nonlinear function to an arbitrary degree of accuracy [14]. This, together with the availability of training techniques, has made them very successful in many predictive control applications and commercial products. More details of the use of NN for control can be found in [32].

Neural Networks are usually combined with linear models in practical applications, since they are not able to extrapolate beyond the range of their training data set. Based on a model confidence index, the NN is gradually turned off when its prediction looks unreliable, the predictions relying on the linear part.

2.2 State Space Models

The linear state space model can naturally be extended to include nonlinear dynamics. The following state space model can be used to describe a nonlinear plant

$$x(t+1) = f(x(t), u(t)) \qquad y(t) = g(x(t)) \tag{3}$$

where $x(t)$ is the state vector and f and g are generic nonlinear functions. Notice that the same equation can be used for monovariable and multivariable processes. Notice also that this model can easily be derived from the differential equations that describe the model (if they are known) by converting them into a set of first-order equations. Model attainment in this case is straightforward but the procedure is very difficult to obtain from experimental data when no differential equations are available.

This kind of model is the most widely extended for nonlinear plants since it has given rise to a lot of theoretical results: the majority of results about stability and robustness have been developed inside this framework. It is also used in commercial tools such as NOVA NLC or nonlinear PFC.

An important class of nonlinear state-space models are Piece Wise Affine (PWA) models, which can be used to approximate smooth nonlinear processes and are defined by:

$$x_{k+1} = A^i x_k + B^i u_k + f_i \text{ for: } x_k \in \mathcal{X}_i$$

where $\{\mathcal{X}_i\}_{i=1}^{s}$ is a polyhedral partition of the state space.

2.3 State Estimation

When a state space model is being used, the system state is necessary for prediction and therefore has to be known. In some cases it is accessible through measurements, but in general this is not the case and a state observer must be implicitly or explicitly included in the control loop. The choice of an appropriate observer may have influence on the closed-loop performance and stability.

The most extended approach for output-feedback NMPC is based on the *Certainty Equivalence Principle.* The estimate state $\hat{x}$ is computed via a state observer and used in the model predictive controller. Even assuming that the observer error is exponentially stable, often only local stability of the closed-loop is achieved [24], i.e. the observer error must be small to guarantee stability of the closed-loop and in general nothing can be said about the necessary degree of smallness. This is a consequence of the fact that no general valid separation principle for nonlinear systems exists. Nevertheless this approach is applied successfully in many applications. A straightforward extension of the optimal linear filter (Kalman filter) is the Extended Kalman filter EKF. The basic idea of EKF is to perform linearization at each time step to approximate the nonlinear system as a time-varying system affine in the variables to be estimated, and to apply the linear filtering theory to it [26]. Although its theoretical properties remain largely unproven the EKF is popular in industry and usually performs well. Neither the Kalman Filter nor the extended Kalman Filter rely on on-line optimization, and neither handle constraints.

There exists a dual of the NMPC approach for control for the state estimation problem. It is formulated as an on-line optimization similar to NMPC and is named moving horizon estimation (MHE), see for example [1],[29],[41],[47]. It is dual in the sense that a moving window of old measurement data is used to obtain an optimization based estimate of the system state. Moving horizon estimation was first presented for unconstrained linear systems by Kwon et al. [22]. The first use of MHE for nonlinear systems was published by Jang *et al.* [15]. In their work, however, the model does not account for disturbances or constraints. The stability of constrained linear MHE was developed by Muske and Rawlings [30] and Rao *et al.* [37]. The groundwork for constrained nonlinear MHE was developed by Rao et al. [38]. Some applications in the chemical industry have been reported [42].

3 Solution of the NMPC Problem

In spite of the difficulties associated with nonlinear modelling, the choice of appropriate model is not the only important issue. Using a nonlinear model changes the control problem from a convex quadratic program to a nonconvex nonlinear problem, which is much more difficult to solve and provides no guarantee that the global optimum can be found. Since in real-time control the optimum has to be obtained in a prescribed interval, the time needed to find the optimum (or an acceptable approximation) is an important issue.

The problem to be solved at every sampling time is the computation of the control sequence $\mathbf{u}$ that takes the process to the desired regime. This desired operating point ($\mathbf{y}_s$, $\mathbf{x}_s$, $\mathbf{u}_s$) may be determined by a steady-state optimisation which is usually based on economic objectives. The cost function to be minimised may take the general form

$$J = \sum_{j=1}^{N} \|\mathbf{y}(t+j) - \mathbf{y}_s\|_{\mathbf{R}} + \sum_{j=1}^{M-1} \| \triangle \mathbf{u}(t+j)\|_{\mathbf{P}} + \sum_{j=1}^{M-1} \|\mathbf{u}(t+j) - \mathbf{u}_s\|_{\mathbf{Q}} \quad (4)$$

where $\mathbf{P}$, $\mathbf{Q}$, and $\mathbf{R}$ are weighting matrices (notice that 1-norm or $\infty-$ norm could also be used). The minimisation is subject to model constraint (model equation) and to the rest of the inequality constraints that can be considered on inputs and outputs.

The solution of this problem requires the consideration (and at least a partial solution) of a nonconvex, nonlinear problem (NLP) which gives rise to a lot of computational difficulties related to the expense and reliability of solving the NLP online. The problem is often solved using Sequential Quadratic Programming (SQP) techniques. These are extensions of Newton-type methods for converging to the solution of the Karush-Kuhn-Tucker (KKT) conditions of the optimisation problem. The method must guarantee fast convergence and must be able to deal with ill conditioning and extreme nonlinearities.

Many problems may appear when applying the method, such as the availability of the second derivatives or the feasibility of the intermediate solution. This last condition is very important in real-time optimisation since, if time is insufficient, the last iteration $\mathbf{u}_k$, which satisfies the local linear approximation to the constraints, is sent to the plant, although it may violate the original constraints. Several variations of the original method exist that try to overcome the main problems. Convergence properties and variants of the method that enhance efficiency are thoroughly discussed by Biegler [5].

It should be noticed that an iterative algorithm consisting of solving a QP problem (which is itself iterative) is used at each sampling instant. Therefore the computational cost is very high, and this justifies the development of special formulations to solve particular problems or approximate solutions in spite of losing optimality.

4 Techniques for Nonlinear Predictive Control

As has been shown in the previous section, the exact solution of the optimisation problem at every sampling instant is a difficult task. Therefore, a set of efficient formulations that try to avoid the problems associated to nonconvex optimisation has appeared in recent years. They are briefly depicted here.

Suboptimal NMPC: This approach avoids the need to find the minimum of a nonconvex cost function by considering the satisfaction of constraints to be the primary objective. If an optimisation strategy that delivers feasible solutions at every sub-iteration (inside a sampling period) is used and a decrease in the cost function is achieved, optimisation can be stopped when the time is over and stability can still be guaranteed. It can be demonstrated that it is sufficient to achieve a continuous decrease in the cost function to guarantee stability.

The main technique that uses this concept was proposed by Scokaert *et al.* [43], and consists of a dual-mode strategy which steers the state towards a terminal set Ω and, once the state has entered the set, a local controller drives the state to the origin. Now, the first controller does not try to minimise the cost function J, but to find a predicted control trajectory which gives a sufficient reduction of the cost.

Simultaneous Approach: In this approach [12] (also known as *multiple shooting* [6]) the system dynamics at the sampling points enter as nonlinear constraints to the optimization problems, i.e. at every sampling point the following equality constraint must be satisfied:

$$\bar{s}_{i+1} = \bar{x}(t_{i+1}, \bar{s}_i, \bar{u}_i)$$

Here $\bar{s_i}$ is introduced as additional degree in the optimization problem and describes the *initial* condition for the sampling interval i. This constraint requires, once the optimization has converged, that the state trajectory pieces fit together. Thus additionally to the input vector $[\bar{u_1}, \ldots, \bar{u_N}]$ also the vector of the $\bar{s_i}$ appears as optimization variables. For both approaches the resulting optimization problem is often solved using SQP techniques. This approach has different advantages and disadvantages. For example the introduction of the *initial* states $\bar{s_i}$ as optimization variables does lead to a special banded-sparse structure of the underlying QP problem. This structure can be taken into account to lead to a fast solution strategy. A drawback of the simultaneous approach is, that only at the end of the iteration a valid state trajectory for the system is available. Thus if the optimization cannot be finished on time, nothing can be said about the feasibility of the trajectory at all.

Use of Short Horizons: It is clear that short horizons are desirable from a computational point of view, since the number of decision variables of the optimisation problem is reduced. However, long horizons are required to achieve the desired closed-loop performance and stability (as will be shown in the next section). Some approaches have been proposed that try to overcome this problem.

In [46] an algorithm which combines the best features of exact optimisation and a low computational demand is presented. The key idea is to calculate exactly the first control move which is actually implemented, and to approximate the rest of the control sequence which is not implemented. Therefore the number of decision variables is one, regardless of the control horizon. The idea is that if there is not enough time to calculate the complete control sequence, then compute only the first one and approximate the rest as well as possible.

An algorithm that uses only a single degree of freedom is proposed in [21] for nonlinear, control affine plants. A univariate online optimisation is derived by interpolating between a control law which is optimal in the absence of constraints (although it may violate constraints and may not be stabilising) and a sub-optimal control law with a large associated stabilisable set. The interpolation law inherits the desirable optimality and feasibility from these control laws. The stabilising control law uses an optimisation based on only a single degree of freedom and can be performed by including a suitable penalty in the cost or an artificial convergence constraint.

Decomposition of the Control Sequence: One of the key ideas in linear MPC is the use of free and forced response concepts. Although this is no longer valid for nonlinear processes, since the superposition principle does not hold in this case, variation of the idea can be used to obtain implementable formulations of NMPC.

In [7], the prediction of process output is made by adding the *free* response obtained from a nonlinear model of the plant and the *forced* response obtained from an incremental linear model of the plant. The predictions obtained this way are only an approximation because the superposition principle, which permits the mentioned division in *free* and *forced* responses, only applies to linear systems. However, the approximation obtained in this way is shown to be better than those obtained using a linearised process model to compute both responses.

A way to overcome this problem has been suggested in [19] for EPSAC. The key idea is that the manipulated variable sequence can be considered to be the addition of a base control sequence plus a sequence of increments of the manipulated variables. The process output j step ahead prediction is computed as the sum of the response of the process $(y_b(t+j))$ due to the base input sequence plus the response of the process $(y_i(t+j))$ due to the future control increments on the base input sequence. As a nonlinear model is used to compute $y_b(t+j)$ while $y_i(t+j))$ is computed from a linear model of the plant, the cost function is quadratic in the decision variables and it can be solved by a QP algorithm as in linear MPC. The superposition principle does not hold for nonlinear processes and the process output generated this way and the process output generated by the nonlinear controller will only coincide in the case when the sequence of future control moves is zero. If this is not the case, the base is made equal to the last base control sequence plus the optimal control increments found by the QP algorithm. The procedure is repeated until the sequence of future controls is driven close enough to zero.

Feedback Linearisation: In some cases, the nonlinear model can be transformed into a linear model by appropriate transformations. Consider, for example, the process described by the following state space model:

$$x(t+1) = f(x(t), u(t)) \qquad y(t) = g(x(t))$$

The method consists of finding state and input transformation functions $z(t) = h(x(t))$ and $u(t) = p(x(t), v(t))$ such that:

$$z(t+1) = Az(t) + Bv(t) \qquad y(t) = Cz(t)$$

The method has two important drawbacks: the first one is that the transformation functions $z(t) = h(x(t))$ and $u(t) = p(x(t), v(t))$ can be obtained for few cases and the second is that constraints, which are usually linear, are transformed into a nonlinear set of constraints.

MPC Based on Volterra Models: In some cases, the NLP shows a special structure that can be exploited to achieve an online feasible solution to the general optimisation problem. If the process is described by a Volterra model, efficient solutions can be found, especially for second-order models. A control strategy can be devised that can solve the nonlinear problem by iteration of the linear solution, based on the particular structure of Volterra models. This iterative procedure proposed by Doyle *et al.* [11] gives rise to an analytical solution in the unconstrained case or a QP solution if constraints exist and allows an easy solution to the nonlinear problem. If a second-order model is used, the prediction can be written as an extension of the linear process $\mathbf{y} = \mathbf{G}\mathbf{u} + \mathbf{f} + \mathbf{c}(\mathbf{u})$, where $\mathbf{f}$ includes the terms that depends on past and known values and the new term $\mathbf{c}$ takes into account new terms that depend on crossed products between past and future control actions. The prediction depends on the unknowns ($\mathbf{u}$) both in a linear form ($\mathbf{G}\ \mathbf{u}$) and a quadratic form ($\mathbf{c}(\mathbf{u})$) and cannot be solved analytically as in the linear unconstrained case. However, the iterative procedure proposed in [11] starts with an initial value of $\mathbf{c}$ and solves the problem. The new solution is used to recalculate $\mathbf{c}$ and the problem is solved again until the iterated solution is close enough to the previous one. In the constrained case, $\mathbf{u}$ is computed solving a QP. Due to the feasibility of its being implemented in real time, this method has been successfully applied to real plants, such as polymerisation processes [25] or biochemical reactors [11].

In the simplified case that the process can be modelled by a Hammerstein model, the problem can be easily transformed into a linear one by inverting the nonlinear static part, $g(.)$. The same idea can be applied to Wiener models, where the static nonlinearity goes after the linear dynamics. In [33] a pH neutralization process is controlled in this way.

Neural Networks: Artificial Neural Networks, apart from providing a modelling tool that enables accurate nonlinear model attainment from input-output data, can also be used for control. Since NNs are universal approximators, they can *learn* the behaviour of a nonlinear controller and calculate the control signal online with few calculations, since the time-consuming part of the NN (training) is done beforehand. This has been applied to several processes in the process industry [2], [4] and to systems with short sampling intervals (in the range of milliseconds) such as internal combustion engines [31]. An application of an NN controller to a mobile robot is detailed at the end of the paper.

Piecewise Affine Systems: In case the process cand be described by a PWA state-space model the NMPC becomes a Mixed Integer Quadratic Problem (MIQP), which can be solved as a series of QP problems. There are several algorithms to do that and one interesting approach is the one proposed in [34], that belongs to the class of Branch and Bound (B & B) methods. The procedure uses the concepts of reachable set combined to the specific B & B methods, in order to reduce the number of Quadratic Problems needed to be solved by the optimization algorithm.

5 Stability and Nonlinear Model Predictive Control

The efficient solution of the optimal control problem is important for any application of NMPC to real processes, but stability of the closed loop is also of crucial importance. Even in the case that the optimization algorithm finds a solution, this fact does not guarantee closed-loop stability (even with perfect model match). The use of terminal penalties and/or constraints, Lyapunov functions or invariant sets has given rise to a wide family of techniques that guarantee the stability of the controlled system. This problem has been tackled from different points of view, and several contributions have appeared in recent years, always analyzing the regulator problem (drive the state to zero) in a state space framework. The main proposals are the following:

- infinite horizon. This solution was proposed by Keerthi and Gilbert [18] and consists of increasing the control and prediction horizons to infinity, $P, M \to \infty$. In this case, the objective function can be considered a Lyapunov function, providing nominal stability. This is an important concept, but it cannot be directly implemented since an infinite set of decision variables should be computed at each sampling time.
- terminal constraint. The same authors proposed another solution considering a finite horizon and ensuring stability by adding a state terminal constraint of the form $x(k+P) = x_s$. With this constraint, the state is zero at the end of the finite horizon and therefore the control action is also zero; consequently (if there are no disturbances) the system stays at the origin. Notice that this adds extra computational cost and gives rise to a restrictive operating region, which makes it very difficult to implement in practice.
- dual control. This last difficulty made Michalska and Mayne [28] look for a less restrictive constraint. The idea was to define a region around the final state inside which the system could be driven to the final state by means of a linear state feedback controller. Now the constraint is:

  $$x(t+P) \in \Omega$$

  The nonlinear MPC algorithm is used outside the region in such a way that the prediction horizon is considered as a decision variable and is decreased at each sampling time. Once the state enters Ω, the controller switches to a previously computed linear strategy.
- quasi-infinite horizon. Chen and Allgöwer [8] extended this concept, using the idea of terminal region and stabilizing control, but only for the computation of the terminal cost. The control action is determined by solving a finite horizon problem without switching to the linear controller even inside the terminal region. The method adds the term $\|x(t+T_p)\|_P^2$ to the cost function. This term is an upper bound of the cost needed to drive the nonlinear system to

the origin starting from a state in the terminal region and therefore this finite horizon cost function approximates the infinite-horizon one.

These formulations and others with guaranteed stability were summarized in the survey paper by Mayne *et al.* [27]. In this reference, the authors present general sufficient conditions to design a stabilizing constrained MPC and demonstrate that all the aforementioned formulations are particular cases of them.

The key ingredients of the stabilizing MPC are a terminal set and a terminal cost. The terminal state denotes the state of the system predicted at the end of the prediction horizon. This terminal state is forced to reach a terminal set that contains the steady state. This state has an associated cost denoted as terminal cost, which is added to the cost function.

It is assumed that the system is locally stabilizable by a control law $u = h(x)$. This control law must satisfy the following conditions:

- There is a region Ω such that for all $x(t) \in \Omega$, then $h(x(t)) \in U$ (set of admissible control actions) and the state of the closed loop system at the next sample time $x(t+1) \in \Omega$.
- For all $x(t) \in \Omega$, there exists a Lyapunov function $V(x)$ such that

$$V(x(t)) - V(x(t+1)) \geq x(t)^T Rx(t) + h(x(t))^T Sh(x(t))$$

If these conditions are verified, then considering Ω as terminal set and $V(x)$ as terminal cost, the MPC controller (with equal values of prediction and control horizons) asymptotically stabilizes all initial states which are feasible. Therefore, if the initial state is such that the optimization problem has a solution, then the system is steered to the steady state asymptotically and satisfies the constraints along its evolution.

The condition imposed on Ω ensures constraint fulfillment. Effectively, consider that $x(t)$ is a feasible state and $\mathbf{u}^*(t)$ the optimal solution; then a feasible solution can be obtained for $x(t+1)$. This is the composition of the remaining tail of $\mathbf{u}^*(t)$ finished with the control action derived from the local control law $h(x)$. Therefore, since no uncertainty is assumed, $x(t+j|t+1) = x(t+j|t)$ for all $j \geq 1$. Then the predicted evolution satisfies the constraints and $x(t+P|t+1) \in \Omega$, being P the prediction horizon. Thus, applying $h(x(t+P|t+1))$, the system remains in the terminal set Ω. Consequently, if $x(t)$ is feasible, then $x(t+1)$ is feasible too. Since all feasible states are in X, then the system fulfills the constraints.

The second condition ensures that the optimal cost is a Lyapunov function. Hence, it is necessary for the asymptotic convergence of the system to the steady state. Furthermore, the terminal cost is an upper bound of the optimal cost of the terminal state, in a similar way to the quasi-infinite horizon formulation of MPC.

The conditions previously presented are based on a state space representation of the system and full state information available at each sample time. However, most of the time the only available information is the measurement of the system output. In this case the controller can be reformulated using the outputs and under certain observability and controllability conditions [18], closed-loop stability can be proved. However, the most common way of applying MPC in the input-output formulation is by estimating the state by means of an observer. It is well known that even when the state space MPC and the observer are both stable, there is no guarantee that the cascaded closed-loop system is stable. Thus, additional stabilizing conditions must be considered [13].

If stability analysis in NMPC is a complex task, robustness analysis (that is, stability when modelling errors appear) is logically worse. The previously shown stability results are valid only in the case of a perfect model, which is not the case in practice. This can be considered as an open field with only preliminary results. Formulations in the form of a min-max problem or an H_∞-NMPC have been proposed, although the computational requirements are prohibitive.

6 Conclusions

MPC is considered to be a mature technique for linear and rather slow systems like the ones usually encountered in the process industry. More complex systems, such as nonlinear, hybrid, or very fast processes, were considered beyond the realm of MPC. During the last few years some impressive results have been produced in the field. In spite of these results, there are many open problems in practically every aspect of NMPC: modelling, identification, state estimation, stability, robustness and real-time implementation.

Acknowledgements

This work was partially supported by the European Commission in the framework of the Hycon Network of Excellence, contract number FP6-IST-511368.

References

[1] F. Allgöwer, T.A. Badgwell, J.S. Qin, J.B. Rawlings, and S.J. Wright. *Advances in Control (Highlights of ECC 99)*, chapter Nonlinear Predictive Control and Moving Horizon Estimation - An Introductory Overview. Springer, 1999.

[2] M.R. Arahal, M. Berenguel, and E.F. Camacho. Neural Identification Applied to Predictive Control of a Solar Plant. *Control Engineering Practice*, 6:333–344, 1998.

[3] T.A. Badgwell and S.J. Qin. *Nonlinear Predictive Control*, chapter Review of Nonlinear Model Predictive Control Applications. IEE Control Engineering series, 2001.

[4] M. Berenguel, M.R. Arahal, and E.F. Camacho. Modelling Free Response of a Solar Plant for Predictive Control. *Control Engineering Practice*, 6:1257–1266, 1998.

[5] L. T. Biegler. *Nonlinear Model Predictive Control*, chapter Efficient Solution of Dynamic Optimization and NMPC Problems. Birkhäuser, 2000.

[6] H.G. Bock, M.M. Diehl, D.B. Leineweber, and J.P. Schloder. *Nonlinear Model Predictive Control*, chapter a Direct Multiple Shooting Method for Real-time Optimization of Nonlinear DAE Processes. Birkhäuser, 2000.

[7] E.F. Camacho, M. Berenguel, and F.R. Rubio. *Advanced Control of Solar Power Plants*. Springer-Verlag, London, 1997.

[8] H. Chen and F. Allgöwer. A Quasi-infinite Horizon Nonlinear Predictive Control Scheme with Guaranteed Stability. *Automatica*, 34(10):1205–1218, 1998.

[9] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. Generalized Predictive Control. Part I. The Basic Algorithm. *Automatica*, 23(2):137–148, 1987.

[10] C.R. Cutler and B.C. Ramaker. Dynamic Matrix Control- A Computer Control Algorithm. In *Automatic Control Conference, San Francisco*, 1980.

[11] F.J. Doyle, R.K. Pearson, and B.A. Ogunnaike. *Identification and Control Using Volterra Models*. Springer, 2001.

[12] R. Findeisen and F. Allgower. An Introduction to Nonlinear Model Predictive Control. In *21st Benelux Meeting on Systems and Control*, Veldhoven, 2002.

[13] R. Findeisen, L. Imsland, F. Allgöwer, and B.A. Foss. State and Output Nonlinear Model Predictive Control: An Overview. *European Journal of Control*, 9:190–206, 2003.

[14] K. Hornik, M. Stinchombe, and H. White. Multilayer Feedforward Networks are Universal Approximators. *Neural networks*, pages 359–366, 1989.

[15] Shi-Shang Jang, Babu Joseph, and Hiro Mukai. Comparison of two approaches to on-line parameter and state estimation of nonlinear systems. *Industrial and Engineering Chemistry Process Design and Development*, 25:809–814, 1986.

[16] Y. Jin, X. Sun, and C. Fang. Adaptive Control of Bilinear Systems with Bounded Disturbances, and its Application. *Control Engineering Practice*, pages 815–822, 1996.

[17] T.A. Johansen, J.T. Evans, and B.A. Foss. Identification of Nonlinear System Structure and Parameters Using Regime Decomposition. *Automatica*, pages 321–326, 1995.

[18] S.S. Keerthi and E.G. Gilbert. Optimal Infinite-horizon Feedback Laws for a General Class of Constrained Discrete-time Systems: Stability and Moving-horizon Approximations. *J. Optim. Theory Appl.*, 57(2):265–293, 1988.

[19] R.M.C. De Keyser. A Gentle Introduction to Model Based Predictive Control. In *PADI2 International Conference on Control Engineering and Signal Processing, Piura, Peru*, 1998.

[20] R.M.C. De Keyser and A.R. Van Cuawenberghe. Extended Prediction Self-adaptive Control. In *IFAC Symposium on Identification and System Parameter Estimation, York,UK*, pages 1317–1322, 1985.

[21] B. Kouvaritakis, M. Cannon, and J.A. Rossiter. Stability, Feasibility, Optimality and the Number of Degrees of Freedom in Constrained Predictive Control. In *Symposium on Non-linear Predictive Control*. Ascona, Switzerland, 1998.

[22] W. H. Kwon, A. M. Bruckstein, and T. Kailath. Stabilizing state-feedback design via the moving horizon method. *International Journal of Control*, 37(3):631–643, 1983.

[23] W. S. Levine. *The Control Handbook*, chapter Volterra and Fliess Series Expansions for Nonlinear Systems. CRC/IEEE Press, 1996.

[24] L. Magni, G. De Nicolao, and R. Scattolini. Output feedback and tracking of nonlinear systems with model predictive control. *Automatica*, 37:1601–1607, 2001.

[25] B.R. Maner, F. J. Doyle, B.A. Ogunnaike, and R.K. Pearson. Nonlinear Model Predictive Control of a Multivariable Polymerization Reactor unsing Second Order Volterra Models. *Automatica*, 32:1285–1302, 1996.

[26] D.Q. Mayne. Nonlinear model predictive control: An assessment. In *Fifth International Conference on Chemical Process Control - CPC V*, pages 217–231, 1996.

[27] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained Model Predictive Control: Stability and Optimality. *Automatica*, 36:789–814, 2000.

[28] H. Michalska and D.Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. on Automatic Control*, 38(11):1623–1633, 1993.

[29] H. Michalska and D.Q. Mayne. Moving horizon observers and observer-based control. *IEEE Trans. on Automatic Control*, 40(6):995–1006, 1995.

[30] K.R. Muske and J. Rawlings. Model Predictive Control with Linear Models. *AIChE Journal*, 39:262–287, 1993.

[31] G. De Nicolao, L. Magni, and R. Scattolini. *Nonlinear Model Predictive Control*, chapter Nonlinear Receding Horizon Control of Internal Combustion Engines. Birkhäuser, 2000.

[32] M. Norgaard, O. Ravn, N.K. Poulsen, and L.K. Hansen. *Neural Networks for Modelling and Control of Dynamic Systems*. Springer, London, 2000.

[33] S. Norquay, A. Palazoglu, and J.A. Romagnoli. Application of Wiener Model Predictive Control (WMPC) to a pH Neutralization Experiment. *IEEE Transactions on Control Systems Technology*, 7(4):437–445, 1999.

[34] Miguel Peña, E. F. Camacho, S. Piñón, and R. Carelli. Model Predictive Controller for Piecewise Affine System. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Rebublic, 2005.

[35] S.J. Qin and T.A. Badgwell. An Overview of Industrial Model Predictive Control Technology. In Chemical Process Control: Assessment and New Directions for Research. In *AIChE Symposium Series 316, 93. Jeffrey C. Kantor, Carlos E. Garcia and Brice Carnahan Eds. 232-256*, 1997.

[36] S.J. Qin and T.A. Badgwell. An Overview of Nonlinear Model Predictive Control Applications. In *IFAC Workshop on Nonlinear Model Predictive Control. Assessment and Future Directions. Ascona (Switzerland)*, 1998.

[37] Christopher V. Rao, James B. Rawlings, and Jay H. Lee. Constrained linear state estimation. A moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.

[38] Christopher V. Rao, James B. Rawlings, and David Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE Transactions on Automatic Control*, 2000.

[39] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Algorithmic Control of Industrial Processes. In 4^{th} IFAC *Symposium on Identification and System Parameter Estimation. Tbilisi* USSR, 1976.

[40] J. Richalet, A. Rault, J.L. Testud, and J. Papon. Model Predictive Heuristic Control: Application to Industrial Processes. *Automatica*, 14(2):413–428, 1978.

[41] D.G. Robertson and J.H. Lee. A least squares formulation for state estimation. *J. Proc. Contr.*, 5(4):291–299, 1995.

[42] Louis P. Russo and Robert E. Young. Moving horizon state estimation applied to an industrial polymerization process. In *Proceedings of 1999 American Control Conference, San Diego, California*, 1999.

[43] P.O.M. Scokaert, D.Q. Mayne, and J.B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648–654, 1999.

[44] J. Sjoberg, Q. Zhang, L. Ljung, A. Benveniste, B. Deylon, P. Glorennec, H. Hjalmarsson, and A. Juditsky. Nonlinear black-box modeling in system identification: a unified overview. *Automatica*, 31(12):1691–1724, 1995.

[45] S. Townsend and G.W. Irwin. *Nonlinear Predictive Control*, chapter Nonlinear Model Based Predictive Control Using Multiple Local Models. IEE Control Engineering series, 2001.

[46] A. Zheng and W. Zhang. *Nonlinear Predictive Control*, chapter Computationally Efficient Nonlinear Model Predictive Control Algorithm for Control of Constrained Nonlinear Systems. IEE Control Engineering series, 2001.

[47] G. Zimmer. State observation by on-line minimization. *Int. J. Contr.*, 60(4):595–606, 1994.

Hybrid MPC: Open-Minded but Not Easily Swayed

S. Emre Tuna, Ricardo G. Sanfelice, Michael J. Messina, and Andrew R. Teel

Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, USA
{emre,rsanfelice,mmessina,teel}@ece.ucsb.edu

Summary. The robustness of asymptotic stability with respect to measurement noise for discrete-time feedback control systems is discussed. It is observed that, when attempting to achieve obstacle avoidance or regulation to a disconnected set of points for a continuous-time system using sample and hold state feedback, the noise robustness margin necessarily vanishes with the sampling period. With this in mind, we propose two modifications to standard model predictive control (MPC) to enhance robustness to measurement noise. The modifications involve the addition of dynamical states that make large jumps. Thus, they have a hybrid flavor. The proposed algorithms are well suited for the situation where one wants to use a control algorithm that responds quickly to large changes in operating conditions and is not easily confused by moderately large measurement noise and similar disturbances.

1 Introduction

1.1 Objectives

The first objective of this paper is to discuss the robustness of asymptotic stability to measurement noise for discrete-time feedback control systems. We focus on control systems that perform tasks such as obstacle avoidance and regulation to a disconnected set of points. We will compare the robustness induced by pure state feedback algorithms to the robustness induced by *dynamic* state feedback algorithms that have a "hybrid" flavor. Nonlinear model predictive control (MPC), in its standard manifestation, will fall under our purview since 1) it is a method for generating a pure state feedback control (see [14] for an excellent survey), 2) it can be used for obstacle avoidance (see [11, 12, 18]) and regulation to a disconnected set of points (this level of generality is addressed in [9] for example), and 3) dynamic "hybrid" aspects can be incorporated to enhance robustness to measurement noise. The second objective of this paper is to demonstrate such hybrid modifications to MPC. The proposed feedback algorithms are able to respond rapidly to significant changes in operating conditions without getting confused by moderately large measurement noise and related disturbances. The findings in this paper are preliminary: we present two different hybrid modifications of MPC, but we have not investigated sufficiently the differences between these modifications, nor have we characterized their drawbacks.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 17–34, 2007.
springerlink.com

1.2 What Do We Mean by “Hybrid MPC”?

First we discuss the term “hybrid” and consider how it has appeared before in the context of MPC.

“Hybrid” Dynamical Systems

In the context of dynamical systems, “hybrid” usually indicates systems that combine continuous and discrete aspects. Often “continuous” and “discrete” refer to the time domains on which solutions are defined. See, for example, [5, 13, 22]. In this situation, a hybrid dynamical system is one in which solutions are defined on time domains that combine continuous evolution and discrete evolution. (The time domain thus may be a subset of the product of the nonnegative reals and the nonnegative integers. See, for example, [3] and [5, 6, 7, 20]). The state flows continuously, typically via a differential equation, as hybrid time advances continuously; the state jumps, according to a update map or “difference” equation, as the hybrid time advances discretely. Whether flowing or jumping occurs depends on the state. The state may or may not contain logic variables that take values in a discrete set. If such variables exist, they do not change during the continuous evolution. Similarly, state variables that must evolve continuously do not change during jumps. We note here that a continuous-time control system implemented with a sample and hold device is a hybrid dynamical system of this type. Thus, when MPC based on a discrete-time model of a continuous-time process is used to synthesize a state feedback that is implemented with sample and hold, this can be thought of as hybrid control, although perhaps not as “hybrid MPC”.

Other times, “continuous” and “discrete” refer to the domains in which the state components take values, while the time domain is fixed to be discrete. In other words, a hybrid system sometimes means a discrete-time system in which some of the variables can take on any of a continuum of values while other states can take on any values in a discrete set. This appears to be the most common meaning of “hybrid dynamical system” as used in the MPC literature, and we will mention specific work below.

“Hybrid” MPC

We believe that the development of MPC for hybrid systems that involve both flowing and jumping will be a very stimulating area of research. Nevertheless, throughout this paper, we will only consider discrete-time systems (although they can be thought of as coming from sampled continuous-time systems). Thus, our meaning of “hybrid MPC” must be related to the second one given above. The main feature of the MPC that we propose is that it is dynamic, sometimes introducing variables that take discrete values, with the aim of enhancing robustness to measurement noise. We focus on discrete-time control problems that can be solved robustly using pure state feedback but that can be solved more robustly by adding dynamics, perhaps with variables that take on discrete values. The idea of adding dynamics to improve robustness is not new, especially as it pertains to the control of continuous-time systems. See [19], [21], [2], [16]. Our purpose is to emphasize this observation in discrete time and to present general dynamic or "hybrid" algorithms that have potential for wide applicability, are simple conceptually, and that improve robustness to measurement noise.

Regarding results on hybrid MPC that have appeared in the literature previously, it is tempting to try to make distinctions between hybrid MPC for nonhybrid systems and (nonhybrid) MPC for hybrid systems. However, the distinction can be blurred easily by treating logic variables from the controller as part of the state of the plant to be controlled. The class of discrete-time hybrid systems to which MPC is most often applied is the class of so-called piecewise affine (PWA) control systems. The equivalence of this type of hybrid model to several other classes of hybrid models has been established in [10]. Model predictive control for PWA systems has been discussed in [1], where the optimization problems to generate the MPC feedback law are shown to be mixed integer multiparameter programs. In other work, the authors of [16] propose a hybrid MPC strategy for switching between a predetermined robust stabilizing state feedback controller and an MPC controller aimed at performance. An early result in [17] used a "dual-mode" approach that involved switching between MPC and a local controller.

2 Control Systems and Measurement Noise

2.1 Introduction

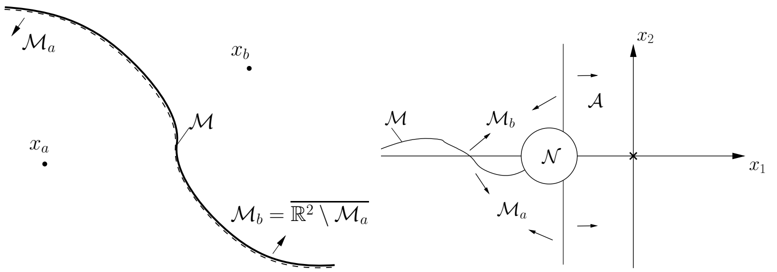
We consider the analysis and design of control algorithms for discrete-time systems of the form

$$x^+ = f(x, u) \; , \tag{1}$$

where $x \in \mathbb{R}^n$ denotes the *state*, x^+ the next value of the state, and $u \in \mathcal{U}$ the *control input*. The function f is assumed to be continuous. At times we will re-write the system (1) as $x^+ = x + \widetilde{f}(x, u)$, where $\widetilde{f}(x, u) := f(x, u) - x$, to emphasize that the discrete-time control system may represent the sampling of a continuous-time control system and that the next state value is not too far from the current state value, i.e., $\widetilde{f}$ is not very large.

In this paper we consider two types of feedback control algorithms: 1) pure state feedback, i.e., $u = \kappa_{\mathrm{PSF}}(x)$, where $\kappa_{\mathrm{PSF}} : \mathbb{R}^n \to \mathcal{U}$ is not necessarily continuous, and 2) dynamic state feedback, i.e., $u = \kappa_{\mathrm{DSF}}(x, \xi)$, $\xi^+ = g_{\mathrm{DSF}}(x, \xi)$, where $\xi \in \mathbb{N}$ and $\kappa_{\mathrm{DSF}} : \mathbb{R}^n \times \mathbb{N} \to \mathcal{U}$ and $g_{\mathrm{DSF}} : \mathbb{R}^n \times \mathbb{N} \to \mathbb{N}$ are not necessarily continuous, where $\mathbb{N} := \{0, 1, \ldots\}$. We are especially interested in the effect of measurement noise. In the case of pure state feedback, this means that $u = \kappa_1(x+e)$, where e represents measurement noise. In the case of dynamic state feedback, this means $u = \kappa_2(x + e, \xi)$, $\xi^+ = g(x + e, \xi)$. We focus on control problems where pure state feedback will have small measurement noise robustness margins, regardless of the control algorithm used. We will show below that problems of this type include controlling continuous-time systems using small sampling periods while attempting to achieve obstacle avoidance and/or regulation to a disconnected set of points.

We frame the discussion around three prototypical control tasks for the continuous-time control system $\dot{x} = v$, where $x \in \mathbb{R}^2$ and $v \in \overline{\mathbb{B}} \subset \mathbb{R}^2$ ($\overline{\mathbb{B}}$ denotes the closed unit ball and $\delta\overline{\mathbb{B}}$ denotes the closed ball of radius δ). The problems are to use sample and hold control with a relatively small sampling period to achieve 1) global regulation to a point, 2) global regulation to a set consisting of two (distinct) points, and 3) global regulation to a target while avoiding an obstacle. In each case, the discrete-time control system is $x^+ = x + u$, where $u \in \delta\overline{\mathbb{B}}$ and $\delta > 0$ represents the sampling period.



(a) Regulation to two distinct points.

(b) Regulation to a target with obstacle avoidance.

Fig. 1. Global regulation to attractors

2.2 Global Regulation to a Point

Suppose we have designed a (family of) continuous feedback(s) $\kappa_\delta : \mathbb{R}^2 \to \delta\overline{\mathbb{B}}$ to achieve stability of and global asymptotic convergence to a point x^* which we take to be the origin without loss of generality. For example, suppose $\delta \in (0, 1]$ and we take

$$\kappa_\delta(x) = \frac{-\delta x}{\max\{1, |x|\}} \ . \tag{2}$$

To analyze the behavior of the system with measurement noise, define, for each $(s, \delta) \in \mathbb{R}_{\geq 0} \times (0, 1]$,

$$\gamma(s, \delta) = \frac{\max\{1, s\} - \delta}{\max\{1, s\}} \ . \tag{3}$$

Note that $\gamma(s, \delta) < 1$ for all $(s, \delta) \in \mathbb{R}_{\geq 0} \times (0, 1]$ and $\gamma(\cdot, \delta)$ is nondecreasing. Then note that

$$|x + \kappa_\delta(x + e)| \leq \frac{|x| \left(\max\{1, |x + e|\} - \delta\right) + \delta|e|}{\max\{1, |x + e|\}} \tag{4}$$

so that when $|e| \leq 0.5|x|$ we have $|x + \kappa_\delta(x + e)| \leq |x|\gamma(1.5|x|, 0.5\delta)$. It follows that the state trajectory will converge to a neighborhood of the origin that is proportional to the worst case size of the measurement noise, regardless of how small δ is. In other words, fast sampling does not make the system more and more sensitive to measurement noise.

2.3 Global Regulation to a Set Consisting of Two Distinct Points

Suppose we have designed a pure state feedback control algorithm $\kappa : \mathbb{R}^2 \to \delta\overline{\mathbb{B}}$ to achieve stability of and global asymptotic convergence to the set $\mathcal{A} := \{x_a, x_b\}$, where $x_a \neq x_b$. Stability implies that if the system starts close to one of the points it will stay close to that point forever. Let $\mathcal{H}_a$, respectively $\mathcal{H}_b$, denote the set of points that produce trajectories converging to x_a, respectively x_b. An illustration is given in Figure 1(a). By uniqueness of solutions, these sets are well defined and disjoint. By global asymptotic stability they cover $\mathbb{R}^2$, and because of the stability property each set is nonempty. We define $\mathcal{H}$ to be the intersection of the closures of $\mathcal{H}_a$ and $\mathcal{H}_b$, i.e., $\mathcal{H} = \overline{\mathcal{H}_a} \cap \overline{\mathcal{H}_b}$. Note that for each point $x \in \mathcal{H}$ there exists a neighborhood of x intersecting both $\mathcal{H}_a$ and $\mathcal{H}_b$. Again, using stability, it follows that $\mathcal{H}$ does not include neighborhoods of $\mathcal{A}$. Now, due to the nature of dynamical systems, the sets $\mathcal{H}_a$ and $\mathcal{H}_b$ are forward invariant. In particular

$$z_a \in \mathcal{H}_a \ , \ z_b \in \mathcal{H}_b \qquad \Longrightarrow \qquad z_a + \kappa(z_a) \in \mathcal{H}_a \ , \ z_b + \kappa(z_b) \in \mathcal{H}_b \ . \tag{5}$$

It can be shown that this fact has the following consequence (in the statement below, $\mathcal{H} + \delta\mathbb{B}$ denotes the set of points having distance less than δ from $\mathcal{H}$):

If $x \in \mathcal{H} + \delta\mathbb{B}$ then there exists e with $|e| < \delta$ such that $x + \kappa(x+e) \in \mathcal{H} + \delta\mathbb{B}$.

In turn it follows that, for each initial condition $x \in \mathcal{H} + \delta\mathbb{B}$ there exists a noise sequence $\mathbf{e} := \{e_k\}_{k=0}^{\infty}$ such that $|e_k| < \delta$ for all k and such that $\phi(k, x, \mathbf{e}) \in \mathcal{H} + \delta\mathbb{B}$ for all k, where $\phi(k, x, \mathbf{e})$ denotes the trajectory starting from x at the kth step under the influence of the measurement noise sequence $\mathbf{e}$. This (small when δ is small) noise sequence does not allow the trajectory to approach the attractor $\mathcal{A}$.

The above statement has the following explanation: Without loss of generality, suppose $x \in \mathcal{H}_a$. Since $x \in \mathcal{H} + \delta\mathbb{B}$, there exists z such that $|x - z| < \delta$ and $z \in \mathcal{H}_b$. In particular, $z + \kappa(z) \in \mathcal{H}_b$. Pick $e = z - x$ and consider $x + \kappa(x+e)$. If $x + \kappa(x+e) \in \mathcal{H}_b$ then $x + \kappa(x+e) \in \mathcal{H} + \delta\mathbb{B}$ since there must be a point on the line connecting x to $x + \kappa(x+e)$ that belongs to $\mathcal{H}$ and since the length of this line is less than δ since $|\kappa(x+e)| < \delta$. If $x + \kappa(x+e) \in \mathcal{H}_a$ then $x + \kappa(x+e) \in \mathcal{H} + \delta\mathbb{B}$ since there must be a point on the line connecting $x + \kappa(x+e)$ and $x + e + \kappa(x+e)$ that belongs to $\mathcal{H}$ and the length of this line must be less than δ since $|e| < \delta$.

To summarize, no matter how we build our pure state feedback algorithm, when δ is small there will be small noise sequences that can keep the system from converging toward the attractor.

2.4 Global Regulation to a Target with Obstacle Avoidance

Suppose we have designed a pure state feedback control algorithm $\kappa : \mathbb{R}^2 \backslash \mathcal{N} \to \delta\overline{\mathbb{B}}$ to achieve stability of and "global" convergence to a point x^* while avoiding an obstacle covering the set $\mathcal{N}$. The situation is depicted in Figure 1(b). For simplicity, we assume that the set $\mathcal{A}$ is made stable and attractive. Basically, this says that if the control finds that the vehicle is nearly past the obstacle it moves in the direction of the target. Let $\mathcal{H}_a$, respectively $\mathcal{H}_b$, denote the set of points that produce trajectories converging to $\mathcal{A}$ by crossing into the set $\mathcal{A}$ above the obstacle, respectively below the obstacle. By uniqueness of solutions, these sets are well defined and disjoint. By "global" asymptotic convergence they cover $\mathbb{R}^2 \backslash (\mathcal{N} \cup \mathcal{A})$, and because of stability and attractivity of $\mathcal{A}$ each set is nonempty. We define $\mathcal{H}$ to be the intersection of the closures of $\mathcal{H}_a$ and $\mathcal{H}_b$. Again using stability of $\mathcal{A}$ it follows that $\mathcal{H}$ does not include neighborhoods of $\mathcal{A}$. Because of this (5) holds, at least when δ is small enough. Using the same reasoning as in the previous subsection, we conclude that measurement noise of size δ can be used to keep the trajectories close to $\mathcal{H}$, which is on the "wrong" side of the obstacle, or else make the vehicle crash into the obstacle.

2.5 A General Principle

We point out here that the ideas put forth above in the discussion about stabilization of an attractor consisting of two distinct points and the discussion about obstacle avoidance generalize. Indeed, let $\mathcal{O} \subset \mathbb{R}^n$ be open and consider the discrete-time system

$$x^+ = x + \widetilde{f}(x) \ . \tag{6}$$

Let $\bar{h} \in \mathbb{N}_{\geq 2}$ and let the sets $\mathcal{H}_i$, for $i \in \{1, \ldots, \bar{h}\}$, satisfy $\bigcup_i \mathcal{H}_i = \mathcal{O}$. Define $\mathcal{H} = \bigcup_{i,j,i \neq j} \overline{\mathcal{H}_i} \cap \overline{\mathcal{H}_j}$.

Lemma 1. *Suppose that for each $z \in \mathcal{H}$ there exist $i, j \in \left\{1, \ldots, \bar{h}\right\}$ with $i \neq j$ and for each $\rho > 0$ there exist points $z_i, z_j \in \{z\} + \rho\mathbb{B}$ so that $z_i + \widetilde{f}(z_i) \in \mathcal{H}_i$ and $z_j + \widetilde{f}(z_j) \in \mathcal{H}_j$. Let $\varepsilon > 0$. If $x \in \mathcal{H} + \varepsilon\mathbb{B}$, $\{x\} + 2\varepsilon\mathbb{B} \subset \mathcal{O}$, and $|\widetilde{f}(x + e)| < \varepsilon$ for all $|e| < \varepsilon$ then there exists e such that $|e| < \varepsilon$ and $x + \widetilde{f}(x + e) \in \mathcal{H} + \varepsilon\mathbb{B}$.*

In turn, we have the following result.

Corollary 1. *Let $\varepsilon > 0$. Let $\mathcal{C} \subset \mathcal{O}$ be such that, for each $\xi \in \mathcal{C}$, $\xi + 2\varepsilon\mathbb{B} \subset \mathcal{O}$ and $|\widetilde{f}(\xi + e)| < \varepsilon$ for all $|e| < \varepsilon$. Then, for each $x_0 \in \mathcal{C} \cap (\mathcal{H} + \varepsilon\mathbb{B})$ there exists a sequence $\{e_k\}$ with $|e_k| < \varepsilon$ such that the sequence generated by $x_{k+1} = x_k + \widetilde{f}(x_k + e_k)$ satisfies $x_k \in \mathcal{H} + \varepsilon\mathbb{B}$ for all k such that $x_i \in \mathcal{C}$ for all $i \in \{0, \ldots, k-1\}$.*

A similar result applies to systems of the form $x^+ = x + \widetilde{f}(x, \kappa(x + e))$ and long as $\widetilde{f}(\cdot, u)$ is locally Lipschitz uniformly over all u in the range of κ. The ideas used to establish a result for such systems parallels the main idea in the proof of [4, Proposition 1.4]. We omit this result because of space limitations.

3 Standard MPC

In this section we review "standard MPC". In standard MPC a pure state feedback function is generated as the mapping from the state x to the solution to an optimization problem, parametrized by x, that uses continuous functions and does not use hard constraints. It has been shown in [15] that standard MPC yields a closed loop with some robustness to measurement noise. (This is in contrast to the situation where the MPC optimization involves hard constraints. Examples have been given in [8] to show that hard constraints can lead to zero robustness margins.) However, as suggested by the discussion in the previous section, the robustness margins may be quite small, especially if the discrete-time plant is coming from a discrete-time model of a continuous-time system using a relatively small sampling period and the control task is obstacle avoidance or regulation to a disconnected set of points.

The control objective is to keep the state in the open state space $\mathcal{X} \subset \mathbb{R}^n$ and stabilize the closed *attractor* $\mathcal{A} \subset \mathbb{X}$. MPC can be used to achieve this objective. The MPC algorithm is described as follows:

We denote an input sequence $\{u_0, u_1, \ldots\}$ by $\mathbf{u}$ where $u_i \in \mathcal{U}$ for all $i \in \mathbb{N}$. Let $\mathbb{E}_{\geq 0}$ denote $[0, \infty]$. Let $\sigma : \mathbb{R}^n \to \mathbb{E}_{\geq 0}$ be a *state measure* with the following properties: (i) $\sigma(x) = 0$ for $x \in \mathcal{A}$, $\sigma(x) \in (0, \infty)$ for $x \in \mathbb{X} \backslash \mathcal{A}$, and $\sigma(x) = \infty$ for $x \in \mathbb{R}^n \setminus \mathbb{X}$, (ii) continuous on $\mathbb{X}$, (iii) $\sigma(x)$ blows up as either x gets unbounded or approaches to the border of $\mathbb{X}$. We let $\ell : \mathbb{R}^n \times \mathcal{U} \to \mathbb{E}_{\geq 0}$ be the *stage cost* satisfying $\ell(x, u) \geq \sigma(x)$ and $g : \mathbb{R}^n \to \mathbb{E}_{\geq 0}$ the *terminal cost* satisfying $g(x) \geq \sigma(x)$. Given a *horizon* $N \in \mathbb{N}$, let us define the *cost function* and the *value function*, respectively, as

$$J_N(x, \mathbf{u}) := \sum_{k=0}^{N-1} \ell(\psi(k, x, \mathbf{u}), u_k) + g(\psi(N, x, \mathbf{u})), \;\; V_N(x) := \inf_{\mathbf{u}} J_N(x, \mathbf{u}) \quad (7)$$

where $\psi(k, x, \mathbf{u})$ is the *solution* to system (1) at time k, starting from the initial condition x, evolved under the influence of the input sequence $\mathbf{u}$. The above optimization is over the set of *admissible* input sequences, i.e. input sequences with each element residing in $\mathcal{U}$. In order to keep the discussion simple, we make the following assumption. (A less restrictive set of assumptions for a more general setting can be found in [9].)

Assumption 3.1 *For all $N \in \mathbb{N}$ and $x \in \mathbb{X}$ a minimizing input sequence $\mathbf{u}$ satisfying $V_N(x) = J_N(x, \mathbf{u})$ exists. V_N is continuous on $\mathbb{X}$ and there exists $L > 0$ such that $V_N(x) \leq L\sigma(x)$ for all $x \in \mathbb{X}$ and $N \in \mathbb{N}$.*

Given a horizon N, for $x \in \mathbb{X}$ we let the *MPC-generated feedback* $\kappa_N(x) := u_0$ where u_0 is the first element of an input sequence satisfying $V_N(x) = J_N(x, \mathbf{u})$. In this setting the following result ensues (see [9] for details).

Theorem 1. *Under Assumption 3.1 there exists $L > 0$ such that, for all horizon N, the value function V_N is continuous and satisfies*

$$\sigma(x) \leq V_N(x) \leq L\sigma(x) \qquad \forall x \in \mathbb{X} \ .$$

Moreover, for each $\rho \in (0, 1)$ there exists $n_\circ \in \mathbb{N}$ such that

$$V_N(f(x, \kappa_N(x))) - V_N(x) \leq -\rho\sigma(x) \qquad \forall x \in \mathbb{X} \ , \ N \geq n_\circ \ .$$

In particular, for N sufficiently large, the set $\mathcal{A}$ is asymptotically stable with basin of attraction $\mathcal{X}$.

4 Modified MPC to Decrease Sensitivity to Measurement Noise

4.1 MPC with Memory

Introduction

When using MPC for stabilization, one simple remedy to the robustness problem discussed in Section 2 seems to be to increase the so called execution horizon. That is, instead of applying the first element of an optimal input sequence and then measuring the state after one step to compute a new optimal input sequence for the new initial condition, one could apply the first $N_{\mathrm{e}} \geq 2$ elements of an optimal input sequence (in an open-loop fashion) before taking a new measurement and optimization. By doing so, if the state is close to where it is most vulnerable to measurement noise, before the next measurement it can be carried sufficiently far away (by choosing a large enough N_{e}) from that location. However, this method may be deleterious for certain applications where the conditions change quickly. This presents a trade-off between wanting to be robust to measurement noise and wanting to react quickly when conditions actually change. A compromise can be attained if one augments the state of the system with a memory variable that keeps record of previous decisions (calculations). With memory, the algorithm can be made to have preference over its previous decisions and the state can still be monitored at each step in order to take action against that preference if necessary or profitable.

Algorithm Description

To be more precise, choose the *buffer gain* $\mu > 1$ and a *memory horizon* $M \in \mathbb{N}$. Define $\Omega := \{\omega_1, \ldots, \omega_M\}$, $\omega_i \in \mathcal{U}$. Given $x \in \mathcal{X}$, let (admissible) input sequences $\mathbf{v} = \{v_0, v_1, \ldots\}$ and $\mathbf{w} = \{w_0, w_1, \ldots\}$ be defined as $\mathbf{v} := \underset{\mathbf{u}}{\operatorname{argmin}}\, J_N(x, \mathbf{u})$ and

$$\mathbf{w} := \underset{\mathbf{u}}{\operatorname{argmin}}\, J_N(x, \mathbf{u}) \quad \text{subject to} \quad u_{i-1} = \omega_i \quad \forall i \in \{1, \ldots, M\}\ .$$

Define

$$W_N(x, \Omega) := \inf_{\mathbf{u}} J_N(x, \mathbf{u}) \quad \text{subject to} \quad u_{i-1} = \omega_i \quad \forall i \in \{1, \ldots, M\}$$

and

$$\bar{\kappa}_N(x, \Omega) := \begin{cases} v_0 & \text{if} \quad W_N(x, \Omega) > \mu V_N(x) \\ w_0 & \text{if} \quad W_N(x, \Omega) \leq \mu V_N(x) \end{cases},$$

$$\pi_N(x, \Omega) := \begin{cases} \{v_1, \ldots, v_M\} & \text{if} \quad W_N(x, \Omega) > \mu V_N(x) \\ \{w_1, \ldots, w_M\} & \text{if} \quad W_N(x, \Omega) \leq \mu V_N(x) \end{cases}$$

Note that when $W_N(x,\, \varOmega) \le \mu V_N(x)$, we have $\bar{\kappa}_N(x,\, \varOmega) = \omega_1$ and $\pi_N(x,\, \varOmega) = \{\omega_2,\, \omega_3,\, \ldots,\, \omega_M,\, w_M\}$. The closed loop generated by this algorithm is

$$x^+ = f(x,\, \bar{\kappa}_N(x,\, \varOmega)) \tag{8}$$

$$\varOmega^+ = \pi_N(x,\, \varOmega)\ . \tag{9}$$

We use $\psi(k,\, x,\, \varOmega,\, \bar{\kappa}_N)$ to denote the solution to (8).

Theorem 2. *Let Assumption 3.1 hold. For each $\rho \in (0,\, 1)$ there exist $n_\circ \in \mathbb{N}$ and positive real numbers K and α such that for all $x \in \mathcal{X}$ and $\varOmega$*

$$W_N(f(x,\, \bar{\kappa}_N(x,\, \varOmega)),\, \pi_N(x,\, \varOmega)) - W_N(x,\, \varOmega) \le -\rho\sigma(x) \tag{10}$$

$$\sigma(\psi(k,\, x,\, \varOmega,\, \bar{\kappa}_N)) \le K\sigma(x)\exp(-\alpha k) \qquad \forall k \in \mathbb{N} \tag{11}$$

for all horizon N and memory horizon M satisfying $N \ge M + n_\circ$.

Robustness with Respect to Measurement Noise

Let us now comment on the possible extra robustness that the MPC with memory algorithm may bring to the stability of a closed loop. Suppose the stability of the closed loop obtained by standard MPC has some robustness with respect to (bounded) measurement noise characterized as (perhaps for x in some compact set)

$$V_N(f(x,\, \kappa_N(x+e))) - V_N(x) \le -\sigma(x)/2 + \alpha_v|e|$$

where N is large enough and $\alpha_v > 0$. Let us choose some $\mu > 1$. Let us be given some $\varOmega = \{\omega_1,\, \ldots,\, \omega_M\}$. Then it is reasonable to expect for M and $N - M$ sufficiently large, at least for systems such as that with a disjoint attractor, that

$$W_N(f(x,\, \omega_1),\, \pi_N(x+e,\, \varOmega)) - W_N(x,\, \varOmega) \le -\sigma(x)/2 + \alpha_w|e|$$

with $\alpha_w > 0$ (much) smaller than α_v as long as $W_N(x,\, \varOmega)$ is not way far off from $V_N(x)$, say $W_N(x,\, \varOmega) \le 2\mu V_N(x)$. Now consider the closed loop (8)-(9) under measurement noise. Suppose $W_N(x+e,\, \varOmega) \le \mu V_N(x+e)$. Then $\bar{\kappa}_N(x+e,\, \varOmega) = \omega_1$. For μ sufficiently large it is safe to assume $W_N(x,\, \varOmega) \le 2\mu V_N(x)$. Therefore we have

$$W_N(f(x,\, \bar{\kappa}_N(x+e,\, \varOmega)),\, \pi_N(x+e,\, \varOmega)) - W_N(x,\, \varOmega) \le -\sigma(x)/2 + \alpha_w|e|\ .$$

Now consider the other case where $W_N(x+e,\, \varOmega) > \mu V_N(x+e)$. Then define $\widetilde{\varOmega} := \{v_0,\, \ldots,\, v_{M-1}\}$ where $\{v_0,\, v_1,\, \ldots\} =: \mathbf{v}$ and $V_N(x+e) = J_N(x+e,\, \mathbf{v})$. Note then that $W_N(x+e,\, \widetilde{\varOmega}) = V_N(x+e)$ and it is safe to assume $W_N(x,\, \widetilde{\varOmega}) \le 2\mu V_N(x)$ as well as $W_N(x,\, \widetilde{\varOmega}) \le W_N(x,\, \varOmega)$ for μ large enough. Note finally that $\bar{\kappa}_N(x+e,\, \varOmega) = \bar{\kappa}_N(x+e,\, \widetilde{\varOmega}) = v_0$ and $\pi_N(x+e,\, \varOmega) = \pi_N(x+e,\, \widetilde{\varOmega})$ in this case. Hence

$$\begin{aligned}
&W_N(f(x,\,\bar{\kappa}_N(x+e,\,\Omega)),\,\pi_N(x+e,\,\Omega)) - W_N(x,\,\Omega)\\
&\qquad = W_N(f(x,\,\bar{\kappa}_N(x+e,\,\widetilde{\Omega})),\,\pi_N(x+e,\,\widetilde{\Omega})) - W_N(x,\,\Omega)\\
&\qquad \le W_N(f(x,\,\bar{\kappa}_N(x+e,\,\widetilde{\Omega})),\,\pi_N(x+e,\,\widetilde{\Omega})) - W_N(x,\,\widetilde{\Omega})\\
&\qquad \le -\sigma(x)/2 + \alpha_w|e|\;.
\end{aligned}$$

The robustness of the closed loop is therefore enhanced.

4.2 MPC with Logic

Algorithm Description

The modification of the algorithm explained in the previous section aims to make the control law more *decisive.* In this section we take a different path that will have a similar effect. We augment the state with a logic (or index) variable q in order for the closed loop to adopt a hysteresis-type behavior. We begin by formally stating the procedure.

For each $q \in \{1,\,2,\,\ldots,\,\bar{q}\} =: \mathcal{Q}$ let $\sigma_q : \mathbb{R}^n \to \mathbb{E}_{\ge 0}$ be a state measure with the following properties: (i) $\sigma_q(x) \in (0,\,\infty)$ for $x \in \mathcal{X}_q \setminus \mathcal{A}$, and $\sigma_q(x) = \infty$ for $x \in \mathbb{R}^n \setminus \mathcal{X}_q$, (ii) is continuous on $\mathcal{X}_q$, (iii) $\sigma_q(x)$ blows up either as x gets unbounded or approaches to the border of $\mathcal{X}_q$, and finally (iv) $\sigma_q(x) \ge \sigma(x)$. We then let $\ell_q : \mathbb{R}^n \times \mathcal{U} \to \mathbb{E}_{\ge 0}$ be our *q-stage cost* satisfying $\ell_q(x,\,u) \ge \sigma_q(x)$ and $g_q : \mathbb{R}^n \to \mathbb{E}_{\ge 0}$ *q-terminal cost* satisfying $g_q(x) \ge \sigma_q(x)$. We let $\bigcup_{q\in\mathcal{Q}} \mathcal{X}_q = \mathcal{X}$. Given a horizon $N \in \mathbb{N}$, we define, respectively, the *q-cost function* and the *q-value function*

$$J_N^q(x,\,\mathbf{u}) := \sum_{k=0}^{N-1} \ell_q(\psi(k,\,x,\,\mathbf{u}),\,u_k) + g_q(\psi(N,\,x,\,\mathbf{u})), \quad V_N^q(x) := \inf_{\mathbf{u}} J_N^q(x,\,\mathbf{u})\;.$$

We make the following assumption on V_N^q which is a slightly modified version of Assumption 3.1.

Assumption 4.1 *For all $N \in \mathbb{N}$, $q \in \mathcal{Q}$, and $x \in \mathcal{X}_q$ a minimizing input sequence $\mathbf{u}$ satisfying $V_N^q(x) = J_N^q(x,\,\mathbf{u})$ exists. V_N^q is continuous on $\mathcal{X}_q$. For each $q \in \mathcal{Q}$ there exist $L_q > 0$ such that $V_N^q(x) \le L_q\sigma_q(x)$ for all $x \in \mathcal{X}_q$ and $N \in \mathbb{N}$. There exists $L > 0$ such that for each $x \in \mathcal{X}$ there exists $q \in \mathcal{Q}$ such that $V_N^q(x) \le L\sigma(x)$ for all $N \in \mathbb{N}$.*

Let $\mu > 1$. Given $x \in \mathbb{X}$, let the input sequence $\mathbf{v}^q := \{v_0^q,\,v_1^q,\,\ldots\}$ be

$$\mathbf{v}^q := \operatorname*{argmin}_{\mathbf{u}} J_N^q(x,\,\mathbf{u})\;.$$

Let $q^* := \operatorname*{argmin}_{\varrho\in\mathcal{Q}} V_N^\varrho(x)$. Then we define

$$\tilde{\kappa}_N(x,\,q) := \begin{cases} v_0^{q^*} & \text{if} \quad V_N^q(x) > \mu V_N^{q^*}(x) \\ v_0^q & \text{if} \quad V_N^q(x) \le \mu V_N^{q^*}(x) \end{cases},\; \theta_N(x,\,q) := \begin{cases} q^* & \text{if} \quad V_N^q(x) > \mu V_N^{q^*}(x) \\ q & \text{if} \quad V_N^q(x) \le \mu V_N^{q^*}(x) \end{cases}$$

Let the closed loop generated by the algorithm be

$$x^+ = f(x,\, \tilde{\kappa}_N(x,\, q)) \tag{12}$$
$$q^+ = \theta_N(x,\, q)\ . \tag{13}$$

We use $\psi(k,\, x,\, q,\, \tilde{\kappa}_N)$ to denote the solution to (12).

Theorem 3. *Let Assumption 4.1 hold. For each $\rho \in (0,\, 1)$ there exist $n_\circ \in \mathbb{N}$ and positive real numbers K and α such that for all $x \in \mathcal{X}$ and $q \in \mathcal{Q}$*

$$V_N^{\theta_N(x,\,q)}(f(x,\, \tilde{\kappa}_N(x,\, q))) - V_N^q(x) \leq -\rho\sigma(x) \tag{14}$$
$$\sigma(\psi(k,\, x,\, q,\, \tilde{\kappa}_N)) \leq K\sigma(x)\exp(-\alpha k) \qquad \forall k \in \mathbb{N} \tag{15}$$

for all horizon $N \geq n_\circ$.

Robustness with Respect to Measurement Noise

We now discuss the robustness of stability of closed loops generated by MPC with logic. By Assumption 4.1, for some large enough fixed horizon N and for all $q \in \mathcal{Q}$ and $x \in \mathbb{X}_q$ it can be shown that $V_N^q(f(x,\, \tilde{\kappa}_N(x, q))) - V_N^q(x) \leq -\sigma_q(x)/2$. For the analysis it makes no difference whether V_N^q is coming from an optimization problem or not. Therefore we might just as well consider the case where we have a number of control Lyapunov functions V^q active on sets $\mathbb{X}_q$ with associated feedbacks κ_q satisfying

$$V^q(f(x,\, \kappa_q(x))) - V^q(x) \leq -\sigma_q(x)/2$$

for each $x \in \mathbb{X}_q$. Suppose each of the closed loops $x^+ = f(x,\, \kappa_q(x))$ has some degree of robustness characterized by (maybe for x in some compact set)

$$V^q(f(x,\, \kappa_q(x+e))) - V^q(x) \leq -\sigma_q(x)/2 + \alpha_q|e|$$

where $\alpha_q > 0$. Now let us compound all these individual systems into a single one by picking $\mu > 1$ and with a switching strategy $q^+ = \theta(x,\, q)$ where θ is defined parallel to θ_N above. In the presence of measurement noise, the closed loop will be

$$x^+ = f(x,\, \kappa_{\theta(x+e,\, q)}(x))$$
$$q^+ = \theta(x+e,\, q)\ .$$

Suppose at some point x we have $\theta(x+e,\, q) = p \neq q$. That means $V^q(x+e) > \mu V^p(x+e)$. When $\mu > 1$ is large enough it is safe to assume, thanks to the continuity of the Lyapunov functions, that $V^q(x) \geq V^p(x)$ since e will be relatively small. Therefore

$$\begin{aligned} V^p(f(x,\, \kappa_p(x+e))) - V^q(x) &\leq V^p(f(x,\, \kappa_p(x+e))) - V^p(x) \\ &\leq -\sigma_p(x)/2 + \alpha_p|e| \\ &\leq -\sigma(x)/2 + \bar{\alpha}|e| \end{aligned}$$

where $\bar{\alpha} := \max_q\{\alpha_q\}$. Therefore if we adopt $V(x,\, q) := V^q(x)$ as the Lyapunov function for our closed loop generated by the logic algorithm we can write

$$V(f(x,\, \kappa_{\theta(x+e,\, q)}(x+e)),\, \theta(x+e,\, q)) - V(x,\, q) \leq -\sigma(x)/2 + \bar{\alpha}|e|$$

for all x and q. Roughly speaking, the strength of robustness of the compound closed loop will be no less than that of the "weakest" individual system, provided that the buffer gain μ is high enough.

Figure 2 depicts the level sets of two Lyapunov functions with minima at two distinct target points. The sets $\{x : V_N^1(x)/V_N^2(x) = \mu\}$ and $\{x : V_N^2(x)/V_N^1(x) = \mu\}$ are indicated by dotted curves. The robustness margin with respect to measurement noise is related to the separation between these curves. A possible closed-loop trajectory in the absence of measurement noise is indicated by the dashed curve. Note that there is more than one switch before the trajectory gets close to one of the two target points.

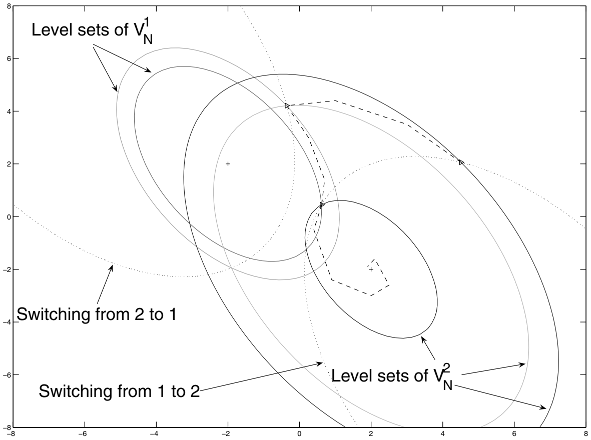


Fig. 2. Level sets of V_N^q for $q \in \{1,\, 2\}$. The dotted curves are the sets $\{x : V_N^1(x)/V_N^2(x) = \mu\}$ and $\{x : V_N^2(x)/V_N^1(x) = \mu\}$. The triangles represent the state at the instants when a switching occurs. The dashed line represents a piece of the solution starting at $x = (4.5,\, 2.1)$, the rightmost triangle.

4.3 Discussion

The two schemes offered have different advantages and disadvantages. Preference would depend on the particular application. However, MPC with memory is easier to employ in the sense that it is a minor modification to the standard algorithm. The difficulty is the determination of the design parameters M and μ; this determination is not obvious. For example, it is not true in general that the larger μ or M are, the more robustness the system has. It may be best to choose them from a range and that range possibly depends on the system and the other MPC related design parameters such as ℓ, g, and N. In the logic case, it is in general not trivial to obtain functions V^q, but it is true that a larger μ will yield more robustness to measurement error, or at least it will not degrade robustness. However, the larger μ, the longer it may take for the closed loop to converge to the desired attractor. Also, a very large μ could make the system incapable of adapting to large changes in conditions.

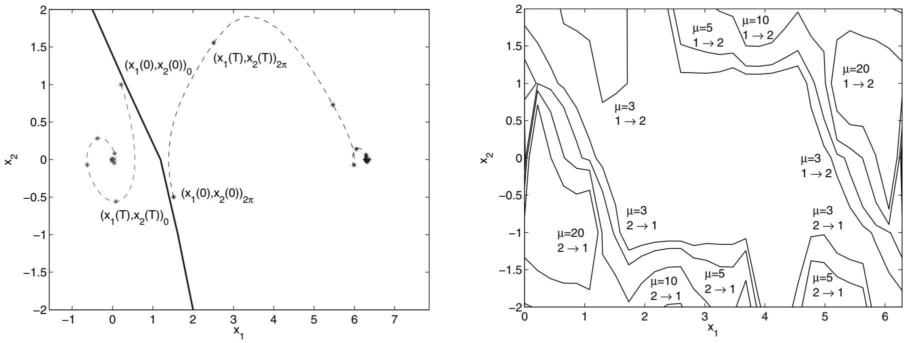
5 Illustrations of Modified MPC Algorithms

5.1 Pendulum Swing Up

Here we consider the problem of swinging up a pendulum and stabilizing its inverted equilibrium. The continuous-time model of the system after an input feedback transformation ($\dot{x} = F(x, v)$ where $x \in \mathbb{R}^2$, $v \in \mathbb{R}$) and normalization is $\dot{x}_1 = x_2; \quad \dot{x}_2 = \sin(x_1) - \cos(x_1)v$, where x_1 is the angle of the pendulum (0 at the upright position) and x_2 is the angular velocity. Following [20, Ex. 8.3], we design three different feedback laws $v_1(\cdot)$, $v_2(\cdot)$, $v_3(\cdot)$ for the system. In [20], each of these control laws are activated in a different prespecified region of the state space to perform the swing-up (the design purpose of $v_1(\cdot)$, $v_2(\cdot)$, $v_3(\cdot)$ is to kick the system from the resting condition, to pump energy into the system, and to stabilize the inverted pendulum to the upright position, respectively). Given a sampling period $T > 0$, for each $u \in \{1, 2, 3\}$ let $x^+ = f(x, u)$ be the discrete-time model of the closed loop $\dot{x} = F(x, v_u(x))$ obtained via integration over an interval of length T, i.e. $f(x, u) = \phi(T)$ where $\phi(\cdot)$ is the solution of $\dot{x} = F(x, v_u(x))$ starting at $\phi(0) = x$. We can now use MPC to decide the swing-up strategy.

We construct the stage cost for standard MPC by adding the kinetic and potential energy of the pendulum. We also include a term in the stage cost that penalizes the control law during the continuous-time horizon to avoid large control efforts. The cost function is periodic in x_1 with period 2π and therefore, there exists a surface on the state space $x_1 - x_2$ where on one side the algorithm tries to reach the upright position rotating the pendulum clockwise and on the other side rotating the pendulum counterclockwise. For one such particular cost, the surface and two different trajectories in opposite directions starting close to the surface are given in Fig. 3(a). As discussed in Section 2, the closed-loop system is vulnerable to small measurement noise in the vicinity of that surface when T is small.

The vulnerability to measurement noise mentioned above can be resolved via the approach discussed in Section 4.2. Despite the fact that $x_1 = 2\pi k$, $k \in \{0, \pm 1, \pm 2, \ldots\}$, correspond to the same physical location, one can construct two stage costs, namely ℓ_q for $q \in \{1, 2\}$, that are not periodic in x_1 such that ℓ_1 vanishes at $x = (0, 0, 0, 0)$ and positive elsewhere and ℓ_2 vanishes at $x = (2\pi, 0, 0, 0)$ and positive elsewhere. By doing so we can attain a robustness margin that does not depend on the size of sampling period T but on μ only, which can be increased to enhance robustness. Fig. 3(b) shows the switching lines for several values of μ for both possible switches ($q = 1 \rightarrow 2$, $q = 2 \rightarrow 1$). For a particular value of μ, the robustness margin is related to the separation of the



(a) Vulnerability of standard MPC to measurement noise: trajectories, denoted by $*$, starting close to the thick line approach different equilibrium points.

(b) MPC with logic: switching lines from $q = 1 \rightarrow 2$ and $q = 2 \rightarrow 1$ for various μ.

Fig. 3. Swing-up with standard MPC and MPC with memory

lines. The margin is independent of the sampling time T as long as NT remains constant, N being the horizon for MPC with logic. The design of an MPC with memory controller and the extension to the case of swinging up the pendulum on a cart follows directly, but due to space limitations we do not include them here.

5.2 Obstacle Avoidance with Constant Horizontal Velocity

Consider a vehicle moving on the plane $x^+ = x + \delta,\ y^+ = y + u\delta$ where $\delta > 0$ and $u \in \{-1, 1\}$ (note that this system can be thought of as sampling the system $\dot{x} = 1,\ \dot{y} = u$). Suppose that the goal for the vehicle is to avoid hitting an obstacle defined by a block of unit height centered about the horizontal axis at $x = 0$ (i.e. the vehicle must leave the region $y \in [-0.5, 0.5]$ before $x = 0$). We design a controller using MPC with logic. Let $q \in \{1, 2\}$, $\ell_1([x, y]^T, u) = \ell_2([x, -y]^T, u) = \exp(y)$, and $g(\cdot) = 0$. Since the costs are invariant on x and symmetric about the x axis, the decision lines defined by μ turn out to be horizontal lines. Let the spacing between these lines be $s(\mu)$. In this case, $s(\mu) = \ln(\mu)$, since $V_N^1([x, y]^T) = \mu V_N^2([x, y]^T)$ when $y = \frac{\ln(\mu)}{2}$ and $V_N^2([x, y]^T) = \mu V_N^1([x, y]^T)$ when $y = -\frac{\ln(\mu)}{2}$ for any N.

Note that when $\mu = 1$ (or $s(\mu) = 0$) MPC with logic is equivalent to the standard MPC algorithm implemented using the stage cost $\ell([x, y]^T, u) = \min\{\ell_1([x, y]^T, u), \ell_2([x, y]^T, u)\}$. As μ is increased, the spacing $s(\mu)$ increases. Table 1 shows the average number of switches and the total number of crashes for 50,000 runs of the system. The initial conditions are set to be $x(0) = -1.5$ and $y(0)$ normally distributed (though kept within $(-1, 1)$) around $y = 0$. The noise is uniformly distributed in $[-0.8, 0.8]$. The key variables of comparison are the spacing of the decision lines $s(\mu)$ and the sampling time δ. With the increased

Table 1. Simulations of system with differing decision line spacing and sampling time for uniformly distributed measurement noise $e \in [-0.8, 0.8]$. Each datum is generated by 50,000 runs starting at $x(0) = -1.5$ and $y(0)$ normally distributed constrained to $(-1, 1)$. "TC" is total number of crashes and "AS" is average number of switches.

δ	0.1		0.06		0.03		0.01		0.006		0.003		0.001	
$s(\mu)$	TC	AS	TC	AS	TC	AS	TC	AS	TC	AS	TC	AS	TC	AS
0.00	5110	2.61	5444	4.36	5791	8.73	5927	26.1	5922	43.8	6107	88.3	6125	263
0.25	3248	1.75	3716	2.88	3862	5.72	4197	17.2	4334	28.9	4375	57.3	4144	169
0.50	1575	1.13	1875	1.76	2253	3.38	2549	10.0	2529	16.7	2625	33.2	2638	101
0.75	428	0.70	609	1.01	863	1.80	1102	5.00	1140	8.26	1155	16.4	1189	48.6
1.00	47	0.46	51	0.57	110	0.86	241	2.05	274	3.21	276	6.14	298	17.7
1.25	1	0.34	2	0.39	1	0.48	6	0.73	9	0.94	8	1.55	17	3.92
1.50	0	0.25	0	0.29	0	0.35	0	0.42	0	0.45	0	0.48	0	0.54

spacing for a given sampling time, there are fewer crashes, as expected, and the trajectories contain fewer switches. The number of switches can be thought of as a measure of the sensitivity to measurement noise. As the sampling time is decreased, the system also becomes more sensitive to measurement noise due to the smaller movements of the system making it difficult to escape the neighborhood of the horizontal axis.

For this system, a crash-free bound on the measurement noise (that solely depends on μ) can be calculated as follows.

Claim. Suppose the MPC with logic controller is implemented with the cost functions ℓ_1, ℓ_2. If the buffer gain $\mu > 1$, the measurement noise is bounded by $\frac{s(\mu)}{2}$, and the horizontal component of the state $x < -\left(\frac{1+s(\mu)}{2}\right)$ then the system will not crash due to measurement noise.

Note that the bound in Claim 5.2 does not depend on the sampling time δ. Hence, the given controller yields a robustness margin independent of δ. For this system, increasing the buffer gain will always increase the robustness margin. However, this may not work on other systems. Increasing the buffer gain too much can cause a system to become obstinate rather than decisive. Choosing the buffer gain then will be very dependent on the task that the system is required to perform. A balance must be made between ignoring (usually small) measurement error and responding to (relatively large) changes in task conditions.

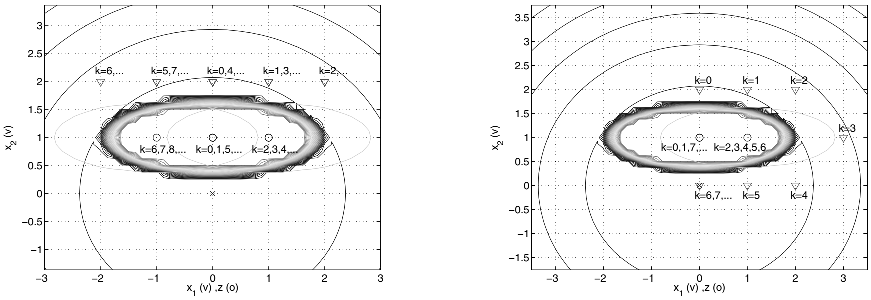
5.3 Avoiding Moving Obstacles

Let the dynamics of a vehicle and obstacle be $x^+ = x + u$ and $z^+ = z + v$, respectively, where $x \in \mathbb{R}^2$, $u \in \{-1, 0, 1\} \times \{-1, 0, 1\}$, and $v \in \{-1, 0, 1\}$. We fix the vertical displacement of the obstacle $h > 0$, and constrain the horizontal displacement to $z \in [-1, 1]$. The goal of the vehicle is to reach some target while avoiding the obstacle whose goal is to reach the vehicle. Both of the agents are considered as single points in $\mathbb{R}^2$ and run MPC to achieve their goals as follows. The stage cost of the vehicle puts a high penalty on the current location of the obstacle and gradually vanishes at the target. The stage cost of the obstacle vanishes whenever $x_1 = z$ and is positive elsewhere.

Applying standard MPC with the vehicle and the obstacle initially aligned vertically at zero horizontal position, the obstacle is able to prevent the vehicle from converging to its target. Suppose the vehicle decides to move in the increasing x_1 direction to avoid the obstacle from the right. The obstacle will follow the vehicle with one step of delay. At some point, it will become necessary for the vehicle to change its course since the optimal path, now that the obstacle has moved, is now to the the left of the obstacle. Hence the vehicle can get stuck possibly as shown in Fig. 4(a).

Using the MPC with memory approach described in Section 4.1, the problem can be resolved. Using the same stage cost, $M = 5$, and $\mu = 1.4$, the vehicle avoids the obstacle. The sequence in memory is effectively used when the obstacle is at $(2, 3)$: the vehicle stays within his initial course of passing the obstacle from the right as shown in Fig. 4(b). Similar results were obtained with MPC with logic using two symmetric respect to x_1 stage cost functions that allow the vehicle avoid the obstacle from the left and from the right, respectively, but the results are omitted because of space limitations.

Note that the moving obstacle can be thought of as noise for the measurement of the vehicle's distance to a static obstacle. Since the displacement of the obstacle has magnitude equal to one, the "measurement noise" for the vehicle is rather large.



(a) Standard MPC: target acquisition failed. Vehicle's vertical position is always equal to two. The thick ellipse corresponds to the bump created by the obstacle in the vehicle's stage cost.

(b) MPC with memory: target acquisition successful. Vehicle goes around the obstacle to the right. The ellipse centered at $(1, 1)$ denotes the bump in the vehicle's stage cost when the obstacle moves.

Fig. 4. Comparison of standard MPC and MPC with memory using identical stage cost and $N = 6$. The vehicle is represented by ∇, the obstacle by $\circ$, and the target by $\times$ (the origin). The discrete time is denoted by k and the level sets of the stage cost for the vehicle are plotted for the initial condition.

References

[1] A. Bemporad, F. Borrelli, and M. Morari. Optimal controllers for hybrid systems: Stability and piecewise linear explicit form. *Proc. IEEE Conf. on Decision and Control*, Vol. 2, pp. 1810–1815, (2000).

[2] F.H. Clarke, S. Yu, S. Ledyaev, L. Rifford, and R.J. Stern. Feedback stabilization and Lyapunov functions. *SIAM Journal on Control and Optimization*, Vol. 39, pp. 25–48, (2000).

[3] P. Collins. A trajectory-space approach to hybrid systems. *Proc. of the International Symposium on Mathematical Theory of Networks and Systems*, (2004).

[4] J-M. Coron and L. Rosier. A relation between continuous time-varying and discontinuous feedback stabilization.*Journal of Mathematical Systems, Estimation, and Control*, Vol. 4, pp. 67–84, (1994).

[5] R. Goebel, J.P. Hespanha, A.R. Teel, C. Cai, and R.G. Sanfelice. Hybrid systems: Generalized solutions and robust stability.*Proc. NOLCOS*, Vol. 1, pp. 1–12, (2004).

[6] R. Goebel and A.R. Teel. Results on solution sets to hybrid systems with applications to stability theory. *Proc. American Control Conf.*, Vol. 1, pp. 557–562, (2005).

[7] R. Goebel and A.R. Teel. Solutions to hybrid inclusions via set and graphical convergence. *Automatica*, Vol. 42, pp. 573–587, (2006).

[8] G. Grimm, M.J. Messina, S.E. Tuna, and A.R. Teel. Examples when nonlinear model predictive control is nonrobust. *Automatica*, Vol. 40, pp. 1729–1738, (2004).

[9] G. Grimm, M.J. Messina, S.E. Tuna, and A.R. Teel. Model predictive control: for want of a local control Lyapunov function, all is not lost. *IEEE Trans. Auto. Control*, Vol. 50, pp. 546–558, (2005).

[10] W.P.M.H Heemels, B. De Schutter, and A. Bemporad. Equivalence of hybrid dynamical models. *Automatica*, Vol. 37, pp.1085–1091, (2001).

[11] Y. Kuwata and J. How. Receding horizon implementation of MILP for vehicle guidance. In *Proc. American Control Conf.*, pp. 2684–2685, (2005).

[12] T. Lapp and L. Singh. Model predictive control based trajectory optimization for nap-of-the-earth (NOE) flight including obstacle avoidance. *Proc. American Control Conf.*, Vol. 1, pp. 891–896, (2004).

[13] J. Lygeros, K.H. Johansson, S.N. Simić, J. Zhang, and S.S. Sastry. Dynamical properties of hybrid automata. *IEEE Trans. Auto. Control*, Vol. 48, pp. 2–17, (2003).

[14] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, Vol. 36, pp. 789–814, (2000).

[15] M.J. Messina, S.E. Tuna, and A.R. Teel. Discrete-time certainty equivalence output feedback: allowing discontinuous control laws including those from model predictive control. *Automatica*, Vol. 41, pp. 617–628, (2005).

[16] P. Mhaskar, N.H. El-Farrah, and P.D. Christofides. Robust hybrid predictive control of nonlinear systems. *Automatica*, Vol. 41, pp. 209–217, (2005).

[17] H. Michalska and D.Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Auto. Control*, Vol. 38, pp. 1623–1633, (1993).

[18] P. Ogren and N.E. Leonard. A convergent dynamic window approach to obstacle avoidance. *IEEE Trans. on Robotics*, Vol. 21, pp. 188–195, (2005).

[19] C. Prieur. Perturbed hybrid systems, applications in control theory. *Nonlinear and Adaptive Control NCN4 2001 (Lecture Notes in Control and Information Sciences Vol. 281)*, pp. 285–294, (2003).

[20] R. G. Sanfelice, R. Goebel, and A.R. Teel. Invariance principles for hybrid systems with connections to detectability and asymptotic stability. To appear in IEEE Trans. Auto. Control, (February 2008).

[21] E.D. Sontag. Clocks and insensitivity to small measurement errors. *Control, Optimisation and Calculus of Variations*, Vol. 4, pp. 537–557, (1999).

[22] A. van der Schaft and H. Schumacher. *An Introduction to Hybrid Dynamical Systems (Lecture Notes in Control and Information Sciences Vol. 251)*, (2000).

Conditions for MPC Based Stabilization of Sampled-Data Nonlinear Systems Via Discrete-Time Approximations

Éva Gyurkovics[1,⋆] and Ahmed M. Elaiw[2]

[1] Budapest University of Technology and Economics, Institute of Mathematics,
Budapest H-1521, Hungary
gye@math.bme.hu

[2] Al-Azhar University (Assiut), Faculty of Science, Department of Mathematics,
Assiut, Egypt
a_m_elaiw@yahoo.com

Summary. This paper is devoted to the stabilization problem of nonlinear continuous-time systems with piecewise constant control functions. The controller is to be computed by the receding horizon control method based on discrete-time approximate models. Multi-rate - multistep control is considered and both measurement and computational delays are allowed. It is shown that the same family of controllers that stabilizes the approximate discrete-time model also practically stabilizes the exact discrete-time model of the plant. The conditions are formulated in terms of the original continuous-time models and the design parameters so that they should be verifiable in advance.

1 Introduction

One of the most popular methods to design stabilizing controllers for nonlinear systems is the receding horizon control, also known as model predictive control. In receding control, a finite horizon optimal control problem is repeatedly solved and the input applied to the system is based on the obtained optimal open-loop control. As a result of substantial efforts of many researchers, several theoretically well-established versions of this method have been proposed in the past one and a half decade both for continuous- and discrete-time models; see e.g. [24], [4], [6], [21] for surveys and the references therein.

In continuous-time setting a great deal of the investigations is devoted to the idealized situation, when the optimization procedure is solved at all time instants, and the initial value of optimal control is applied to the plant (to mention just a few examples, see [23], [13], [14], [1], [3]). This turns out almost always to be an intractable task in practice. A more realistic assumption is that the optimization problem is solved only at disjoint time instants and the resulting optimal control function is implemented in between, which leads to a sampled-data nonlinear model predictive control scheme (see e.g. [2], [18],

⋆ The financial support from the Hungarian National Science Foundation for Scientific Research, grant no. T037491 is gratefully acknowledged.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 35–48, 2007.
springerlink.com

[8]). Being the optimal control in general merely measurable, the troubles of the implementation of such a function are not negligible. The effect of the "sampling and zero-order hold" is considered in [17] assuming the existence of a global control Lyapunov function (CLF), and in [22], where the stabilizing property of a piecewise constant NMPC computed from - and applied to the continuous-time model is investigated without taking into account any approximation in the plant model.

In several technical respects the situation is simpler, if the model of the plant is given in discrete-time. However, such models frequently derived from some continuous-time models as "good" approximations. For this reason, it is important to know conditions which guarantee that the same family of controllers that stabilizes the approximate discrete-time model of the plant also practically stabilizes the exact model of the plant. Sufficient conditions for a controller having these properties are presented in [26] and [25]. As it is emphasized by the title of the latter paper, these results provide a framework for controller design relying on the approximate discrete-time models, but they do not explain how to find controllers that satisfy the given conditions. Within this framework some optimization-based methods are studied in [11]: the design is carried out either via an infinite horizon optimization problem or via an optimization problem over a finite horizon with varying length. To relax the computational burden of these approaches, one can apply a suitable version of the receding horizon control method. Some sets of conditions are formulated and stability results are proved in [15] and [5] for sampled-data receding horizon control method without and with delays based on approximate discrete-time models. In this work we shall investigate the stability property of the equilibrium under a different set of assumptions that are verifyable in advance.

In receding horizon control method, a Bolza-type optimal control problem is solved, in which the design parameters are the horizon length $0 \leq t_1 \leq \infty$, the stage cost l, the terminal cost g (which are usually assumed to be at least non-negative valued) and the terminal constraint set $\mathcal{X}_f$. It is well-known that, if no further requirements for these parameters are stated, then one can show even linear examples, where the resulting closed-loop system is unstable. On the other hand, if that minimal requirement is satisfied that the origin is a locally asymptotically stable equilibrium for the closed-loop system, then one expect to have the largest domain of attraction possible, the least computational efforts possible for finding the controller, and certain robustness, as well. The domain of attraction can probably be increased by increasing the time-horizon, but this involves the increase of the necessary computational efforts, too. Under the terminal constraint one can expect a relatively large domain of attraction with relatively short time horizon. This is the reason, why this constraint is frequently applied in receding horizon. However, if a terminal constraint is included in the optimization problem, then the corresponding value function will not have suitable regularity in general, which ensures an expected robustness. In [9] several examples are presented which show the realization of this phenomenon. Therefore, if stabilization is aimed via an approximate model, the terminal constraint may not be considered explicitly. Several results show that an appropriate choice of the terminal cost g may also enforce stability: in fact, if g is a strict control Lyapunov function within one of its level sets, then the receding horizon controller makes the origin to be asymptotically stable with respect to the closed-loop system with a domain of attraction containing the above mentioned level set of g (the terminal constraint set is implicit e.g. in [14], [18]). This domain of attraction can be enlarged up to an arbitrary compact set, which is asymptotically controllable to the origin, by a suitable - finite - choice of the horizon length. For a substantial class of systems well-established methods exist for the construction of a suitable terminal cost (see e.g. [3], [1], [29]). Sometimes it may be difficult – if not impossible – to derive an appropriate terminal cost. Lately, it has been proven by [19] and [10] that the required stability can be enforced merely by a sufficiently large time horizon, having obvious advantages, but at the cost of a – possibly substantial – enlargement of the computational burden.

Here we consider in details the case when the terminal cost is a control Lyapunov function, and we shall make some remarks on the case of general terminal cost.

2 Stabilization Results with CLF Terminal Cost

2.1 The Models and the Method

Consider the nonlinear control system described by

$$\dot{x}(t) = f\left(x(t), u(t)\right), \qquad x(0) = x_0, \tag{1}$$

where $x(t) \in \mathcal{X} \subset \mathbb{R}^n$, $u(t) \in U \subset \mathbb{R}^m$, $\mathcal{X}$ is the state space, U is the control constraint set, $f\colon \mathbb{R}^n \times U \to \mathbb{R}^n$, with $f(0,0) = 0$, U is closed and $0 \in \mathcal{X}$, $0 \in U$. We shall assume that f is continuous and Lipschitz continuous with respect to x in any compact set. Let $\Gamma \subset \mathcal{X}$ be a given compact set containing the origin and consisting of all initial states to be taken into account.

Consider an auxiliary function $l : \mathbb{R}^n \times U \to \mathbb{R}_+$ with analogous regularity properties as f satisfying the condition $l(0,0) = 0$, and consider the augmented system (1) with

$$\dot{\chi}(t) = l(x(t), u(t)), \quad \chi(0) = 0. \tag{2}$$

For convenience we introduce the notation $\mathcal{Y}_\rho = \mathcal{Y} \cap \mathcal{B}_\rho$, where $\mathcal{B}_\rho$ denotes the ball around the origin with the radius ρ.

The system is to be controlled digitally using piecewise constant control functions $u(t) = u(iT) =: u_i$, if $t \in [iT, (i+1)T)$, $i \in \mathbb{N}$, where $T > 0$ is the control sampling period. We assume that for any $\overline{x} \in \mathcal{X}_{\Delta'}$ and $\overline{u} \in \mathcal{U}_{\Delta''}$, equation (1)–(2) with $u(t) \equiv \overline{u}$, $(t \in [0,T])$ and initial condition $x(0) = \overline{x}$, $\chi(0) = 0)$ has a unique solution on $[0,T]$ denoted by $(\phi^E(., \overline{x}, \overline{u}), \varphi^E(., \overline{x}, \overline{u}))$. Then, the augmented *exact discrete-time model* of the system (1)–(2) can be defined as

$$x_{i+1}^E = F_T^E(x_i^E, u_i), \qquad\qquad x_0^E = x_0, \tag{3}$$

$$\chi_{i+1}^E = \chi_i^E + l_T^E\left(x_i^E, u_i\right), \qquad \chi_0^E = 0, \tag{4}$$

where $F_T^E(x,u) := \phi^E(T; x, u)$, and $l_T^E(x,u) = \varphi^E(T, x, u)$.

We note that, ϕ^E and φ^E are not known in most cases, therefore, the controller design can be carried out by means of an approximate discrete-time model

$$x_{k+1}^A = F_{T,h}^A\left(x_k^A, u_k\right), \qquad x_0^A = x_0, \tag{5}$$

$$\chi_{k+1}^A = \chi_k^A + l_{T,h}^A\left(x_k^A, u_k\right), \qquad \chi_0^A = 0, \tag{6}$$

where $F_{T,h}^A(x,u)$ and $l_{T,h}^A(x,u)$ are typically derived by multiple application of some numerical approximation formula with (possibly variable) step sizes bounded by the parameter h. Given $\mathbf{u} = \{u_0, u_1, \ldots\}$ and initial conditions $x_0^E = x'$ and $x_0^A = x''$, the trajectories of the discrete-time systems (3) and (5)–(6) are denoted, by $\phi_k^E(x', \mathbf{u})$ and $\phi_k^A(x'', \mathbf{u})$, $\varphi_k^A(x'', \mathbf{u})$, respectively.

Concerning the parameters T and h, in principle two cases are possible: $T = h$, and T can be adjusted arbitrarily; $T \neq h, T$ is fixed and h can be chosen arbitrarily small. Having less number of parameters, the first case seems to be simpler, but in practice there exists a lower bound to the smallest achievable T. Since the second case has much more practical relevance, here we shall discuss it in details, and we shall only point out the differences arising in case $T = h$, when appropriate. In what follows, we assume that $T > 0$ is given.

In this paper we address the problem of state feedback stabilization of (3) under the assumption that state measurements can be performed at the time instants jT^m, $j = 0, 1, \ldots$:

$$y_j := x^E(jT^m), \qquad j = 0, 1, \ldots.$$

The result of the measurement y_j becomes available for the computation of the controller at $jT^m + \tau_1$, where $\tau_1 \geq 0$, while the computation requires $\tau_2 \geq 0$ length of time i.e. the (re)computed controller is available at $T_j^* := jT^m + \tau_1 + \tau_2$, $j = 0, 1, \ldots$. We assume that $\tau_1 = \ell_1 T$, $\tau_2 = \ell_2 T$ and $T^m = \ell T$ for some integers $\ell_1 \geq 0$, $\ell_2 \geq 0$ and $\ell \geq \ell_1 + \ell_2 =: \overline{\ell}$.

If $\ell = 1$, $\ell_1 = \ell_2 = 0$, then we can speak about a single rate, one-step receding horizon controller without delays, if $\ell > 1$, then we have multi-rate, multistep controller with or without delays depending on values of ℓ_1 and ℓ_2. Papers [2] and [7] consider the problem of computational delay in connection with the receding horizon control for exact continuous-time models, while [28] develops results analogous to that of [25] for the case of multi-rate sampling with measurement delays.

A "new" controller computed according to the measurement $y_j = x^E(jT^m)$ will only be available from T_j^*, thus in the time interval $[jT^m, T_j^*)$ the "old" controller has to be applied. Since the corresponding exact trajectory is unknown, an approximation ζ_j^A to the exact state $x^E\left(T_j^*\right)$ can only be used, which can be defined as follows. Assume that a control sequence $\left\{u_0\left(\zeta_{j-1}^A\right), \ldots, u_{\ell-1}\left(\zeta_{j-1}^A\right)\right\}$ has been defined for $j \geq 1$. Let $\mathbf{v}^p\left(\zeta_{j-1}^A\right) = \left\{u_{\ell-\overline{\ell}}\left(\zeta_{j-1}^A\right), \ldots, u_{\ell-1}\left(\zeta_{j-1}^A\right)\right\}$ and define ζ_j^A by

$$\zeta_j^A = \mathcal{F}_{\overline{\ell}}^A\left(y_j, \mathbf{v}^p\left(\zeta_{j-1}^A\right)\right), \quad \zeta_0^A = \phi_{\overline{\ell}}^A(x, \mathbf{u}^c), \tag{7}$$

where $\mathcal{F}^A_{\overline{\ell}}\left(y,\{u_0,\ldots,u_{\overline{\ell}-1}\}\right) = F^A_{T,h}\left(\ldots F^A_{T,h}\left(F^A_{T,h}\left(y,u_0\right),u_1\right)\ldots,u_{\overline{\ell}-1}\right)$, and $\mathbf{u}^c$ is some precomputed controller (independent of state measurements). Let $\mathbf{v}^{(j)} = \left\{u^{(j)}_0,\ldots,u^{(j)}_{\ell-1}\right\}$ be computed for ζ^A_j and let the ℓ-step exact discrete-time model be described by

$$\xi^E_{j+1} = \mathcal{F}^E_{\ell}(\xi^E_j,\mathbf{v}^{(j)}), \qquad \xi^E_0 = \phi^E_{\overline{\ell}}(x,\mathbf{u}^c), \tag{8}$$

where $\mathcal{F}^E_{\ell}(\xi^E_j,\mathbf{v}) = \phi^E_{\ell}(\xi^E_j,\mathbf{v})$. In this way the right hand side of (8) depends on $y_j = x^E(jT^m)$ so that (7-8) represents an unconventional feedback system.

Our aim is to define a measurement based algorithm for solving the following problem: for given T, T^m, τ_1 and τ_2 find a control strategy

$$\mathbf{v}_{\ell,h}\colon \widetilde{\Gamma} \to \underbrace{U \times U \times \ldots \times U}_{\ell \text{ times}}$$

$\mathbf{v}_{\ell,h}(x) = \{u_0(x),\ldots,u_{\ell-1}(x)\}$, using the approximate model (5), (7) which stabilizes the origin for the exact system (3) in an appropriate sense, where $\widetilde{\Gamma}$ is a suitable set containing at least Γ.

Remark 1. If $T = T^m$, $\ell = 1$ and $\ell_1 = \ell_2 = 0$, then the single-rate delay-free case is recovered, therefore it is sufficient to discuss the general case in details.

In order to find a suitable controller $\mathbf{v}$, we shall apply a multistep version of the receding horizon method. To do so, we shall consider the following cost function.

Let $0 < N \in \mathbb{N}$ be given. Let (5) be subject to the cost function

$$J_{T,h}(N,x,\mathbf{u}) = \sum_{k=0}^{N-1} l^A_{T,h}(x^A_k,u_k) + g(x^A_N),$$

where $\mathbf{u} = \{u_0,u_1,\ldots,u_{N-1}\}$, $x^A_k = \phi^A_k(x,\mathbf{u})$, $k = 0,1,\ldots,N$ denote the solution of (5), $l^A_{T,h}$ is defined as in (6) and g is a given function.

Consider the optimization problem

$$P^A_{T,h}(N,x)\colon \; \min\left\{J_{T,h}(N,x,\mathbf{u}) : u_k \in U\right\}.$$

If this optimization problem has a solution denoted by $\mathbf{u}^*(x) = \{u^*_0(x),\ldots, u^*_{N-1}(x)\}$, then the first ℓ elements of $\mathbf{u}^*$ are applied at the state x i.e.

$$\mathbf{v}_{\ell,h}(x) = \left\{u^*_0(x),\ldots,u^*_{\ell-1}(x)\right\}.$$

In what follows we shall use the notation $V^A_N(x) = J_{T,h}(N,x,\mathbf{u}^*(x))$.

2.2 Assumptions and Basic Properties

To ensure the existence and the stabilizing property of the proposed controller, several assumptions are needed.

We might formulate this assumptions in part with respect to the approximate discrete-time model as it was done e.g. in [5] and [15]. However, it turns out that in several cases the verification of some conditions is much more tractable for the original model than the approximate one. For this reason, we formulate the assumptions with respect to the exact model (and to the applied numerical approximation method). For the design parameters l and g, we shall make the following assumption.

Assumption 1. (i) $g : \mathbb{R}^n \to \mathbb{R}$ is continuous, positive definite, radially unbounded and Lipschitz continuos in any compact set.

(ii) l is continuous with respect to x and u and Lipschitz continuous with respect to x in any compact set.

(iii) There exist such class-$\mathcal{K}_\infty$ functions $\varphi_1, \overline{\varphi}_1, \varphi_2$ and $\overline{\varphi}_2$ that

$$\varphi_1(\|x\|) + \overline{\varphi}_1(\|u\|) \le l(x,u) \le \varphi_2(\|x\|) + \overline{\varphi}_2(\|u\|), \tag{9}$$

holds for all $x \in \mathcal{X}$ and $u \in U$.

Remark 2. The lower bound in (9) can be substituted by different conditions: e.g. $\overline{\varphi}_1$ may be omitted, if U is compact. If the stage cost for the discrete-time optimization problem is directly given, other conditions ensuring the existence and uniform boundedness of the optimal control sequence can be imposed, as well (see e.g. [10], [15] and [20]). However, having a $\mathcal{K}_\infty$ lower estimation with respect to $\|x\|$ is important in the considerations of the present paper.

The applied numerical approximation scheme has to ensure the closeness of the exact and the approximate models in the following sense.

Assumption 2. For any given $\Delta' > 0$ and $\Delta'' > 0$ there exists a $h_0^* > 0$ such that

(i) $F_{T,h}^A(0,0) = 0$, $l_{T,h}^A(0,0) = 0$, $l_{T,h}^A(x,u) > 0$, $x \neq 0$, $F_{T,h}^A$ and $l_{T,h}^A$ are continuous in both variables uniformly in $h \in (0, h_0^*]$, and they preserve the Lischitz continuity of the exact models, uniformly in h;

(ii) there exists a $\gamma \in \mathcal{K}$ such that

$$\|F_T^E(x,u) - F_{T,h}^A(x,u)\| \le T\gamma(h), \qquad \|l_T^E(x,u) - l_{T,h}^A(x,u)\| \le T\gamma(h),$$

for all $x \in \mathcal{B}_{\Delta'}$, all $u \in U_{\Delta''}$, and $h \in (0, h_0^*]$.

Remark 3. We note that Assumption A2 depends on the numerical approximation method, and it can be proven for reasonable discretization formulas.

Definition 1. *System (3) is asymptotically controllable from a compact set Ω to the origin, if there exist a $\beta(.,.) \in \mathcal{KL}$ and a continuous, positive and nondecreasing function $\sigma(.)$ such that for all $x \in \Omega$ there exists a control sequence $\mathbf{u}(x)$, $u_k(x) \in U$, such that $\|u_k(x)\| \le \sigma(\|x\|)$, and the corresponding solution ϕ^E of (3) satisfies the inequality*

$$\left\|\phi_k^E(x, \mathbf{u}(x))\right\| \le \beta(\|x\|, kT), \qquad k \in \mathbb{N}.$$

The next assumption formulates, roughly speaking, a necessary condition for the existence of a stabilizing feedback.

Assumption 3. (i) The exact discrete-time system (3) is asymptotically controllable from a set Ω containing Γ to the origin.

(ii) There exists a $\Delta_0 > 0$, and a control sequence $\mathbf{u}^c = \left\{ u_0^c, \ldots, u_{\overline{l}-1}^c \right\}$ ($u_i^c \in U$) can be given so that $\Gamma \subset \Omega_{\Delta_0}$, $\phi_k^E(x, \mathbf{u}^c) \in \Omega_{\Delta_0}$, $\phi_k^A(x, \mathbf{u}^c) \in \Omega_{\Delta_0}$, $k = 0, 1, \ldots, \overline{\ell}$ for all $x \in \Gamma$.

In what follows let $\Delta_1 = \beta(\Delta_0, 0)$ and $\Delta_2 = \sigma(\Delta_0)$, where β and σ are given in Definition 1.

Finally, the next assumption implies that the final state penalty has to be a local control Lyapunov function within the sampled data controllers.

Assumption 4. There exist a positive number η and a class-$\mathcal{K}$ function α_g such that for all $x \in \mathcal{G}_\eta = \{x \in \mathcal{X} : g(x) \leq \eta\}$ there is a $\kappa(x) \in U_{\Delta_2}$ such that for $u_0 = \kappa(x)$

$$g\left(F_T^E(x, u_0)\right) - g(x) + l_T^E(x, u_0) \leq -\alpha_g(\|x\|). \tag{10}$$

Remark 4. Sometimes it may be more convenient to verify the analogue of Assumption A4 for the approximate discrete-time system (c.f. [15]). This is the case e.g. if the model has a controllable linearization (c.f. [2], [22]). In other cases, as e.g. in [16], the present form is more advantageous.

Let us consider now the auxiliary problem of the minimization of the cost function

$$J_T^E(N, x, \mathbf{u}) = \sum_{k=0}^{N-1} l_T^E(x_k^E, u_k) + g(x_N^E),$$

subject to the exact system (3), and introduce the notation

$$V_N^E(x) = \inf \left\{ J_T^E(N, x, \mathbf{u}) \colon \mathbf{u} = \{u_0, \ldots, u_{N-1}\},\ u_k \in U \right\}$$

Lemma 1. *If Assumptions A1, A3 and A4 hold true, then there exists a constant $V_{\max}^E$ independent of N, such that $V_N^E(x) \leq V_{\max}^E$, for all $x \in \Omega_{\Delta_0}$ and $N \in \mathbb{N}$.*

Proof. The proof is similar to that of the analogous statement in [15], therefore it is omitted here. ■

Let us introduce the notations $V_{\max}^A = V_{\max}^E + 1$, $\Delta_2^* = \overline{\varphi}_1^{-1}\left(V_{\max}^A / T\right)$, and

$$\Gamma_{\max}(h_0) = \left\{ x \in \mathcal{X} \colon V_N^A(x) \leq V_{\max}^A,\ h \in (0, h_0] \right\},$$

$$M_f(\Delta', \Delta'') = \max_{x \in \mathcal{X}_{\Delta'}} \max_{u \in U_{\Delta''}} \|f(x, u)\|.$$

Theorem 1. *Suppose that Assumptions A1–A4 are valid, and inequality* $s \geq 2TM_f(2s, \Delta_2^*)$ *holds true, if* $s \geq \Delta_0$. *Then there exist constants* N^*, r_0^*, Δ_1^* *and functions* $\sigma_1, \sigma_2 \in \mathcal{K}_\infty$ *so that for any fixed* $N \geq N^*$, $r_0 \in (0, r_0^*]$ *and* $\delta > 0$ *there exists a* $\overline{h} > 0$ *such that for all* $h \in (0, \overline{h}]$

$$\Gamma \subset \Omega_{\Delta_0} \subset \Gamma_{\max}(h) \subset \mathcal{B}_{\Delta_1^*}, \tag{11}$$

$$\sigma_1 \left(\|x\|\right) \leq V_N^A(x) \leq \sigma_2 \left(\|x\|\right), \tag{12}$$

$$\left\|\phi_k^A(x, \mathbf{u}^*(x))\right\| \leq \Delta_1^*, \quad \|u_k^*(x)\| \leq \Delta_2^*, \quad k = 0, 1, \ldots, N-1, \tag{13}$$

if $x \in \Gamma_{\max}(h) \backslash \mathcal{B}_{r_0}$, *and for all* $k = 1, \ldots, \ell$

$$V_N^A \left(\phi_k^A(x, \mathbf{u}^*(x))\right) - V_N^A\left(x\right) \leq -l_{T,h}^A(x, u_0^*(x)) + \delta, \tag{14}$$

where $\mathbf{u}^*(x)$ *denotes the optimal solution of* $P_{T,h}^A(N, x)$. *Moreover,* V_N^A *is locally Lipschitz continuous in* $\Gamma_{\max}(h)$ *uniformly in* $h \in (0, \overline{h}]$.

Proof. The proof is given in the Appendix. ■

Remark 5. If the sampling parameter T and the discretization parameter h coincide and T can be arbitrary adjusted, then – besides some technical problems that can easily be handled – the main difficulty originates from the fact that the lower bound of $l_{T,h}^A$ is no longer independent of the adjustable parameter. Nevertheless, a uniform lower bound for V_N^A can be given for this case, as well (see [15]).

Remark 6. If $\mathcal{X}$ is bounded, then the condition $s \geq 2T\ M_f(2s, \Delta_2^*)$, if $s \geq \Delta_0$ in Theorem 1 is not needed, otherwise the set of possible initial states, the choice of T and the growth of f have to be fitted together.

2.3 Multistep Receding Horizon Control

In this section we outline an approach to the problem how the occurring measurement and computational delays can be taken into account in the stabilization of multi-rate sampled-data systems by receding horizon controller.

Suppose that a precomputed control sequence $\mathbf{u}^c$ satisfying Assumption A3 is given. Then the following Algorithm can be proposed.

Algorithm. Let $N \geq N^*$ be given, let $j = 0$, $T_{-1}^* = 0$ and let $\mathbf{u}^{(0)} = \mathbf{u}^{(p,0)} = \mathbf{u}^c = \{u_0^c, \ldots, u_{\ell-1}^c\}$. Measure the initial state $y(0) = x_0$.
Step j.

(i) Apply the controller $\mathbf{u}^{(j)}$ to the exact system over the time interval $[T_{j-1}^*, T_j^*]$.
(ii) Predict the state of the system at time T_j^* from $y(j)$ by the approximation let $\zeta_j^A = \phi_\ell^A(y(j),\ \mathbf{u}^{(p,j)})$.

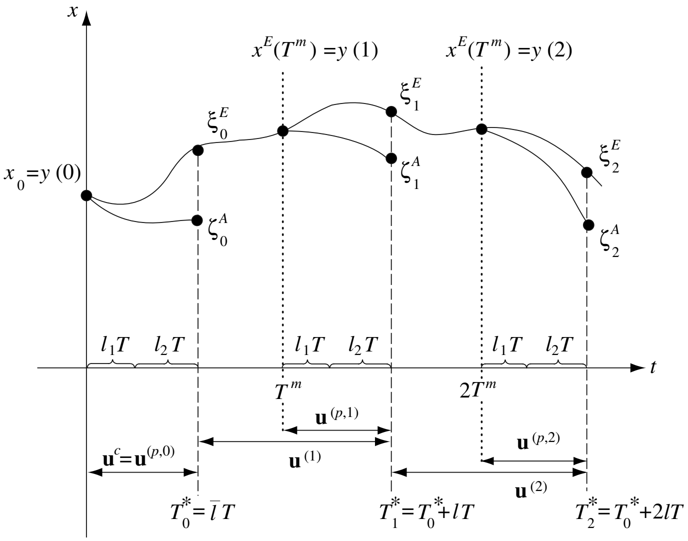


Fig. 1. Sketch to the Algorithm

(iii) Find the solution $\mathbf{u}^* = \{u_0^*, \ldots, u_{N-1}^*\}$ to the problem $P_{T,h}^A(N, \zeta_j^A)$, let $\mathbf{u}^{(j+1)} = \{u_0^*, \ldots, u_{\ell-1}^*\}$ and $\mathbf{u}^{(p,j+1)} = \{u_{\ell-\overline{\ell}}^*, \ldots, u_{\ell-1}^*\}$.

(iv) $j = j + 1$.

A schematic illustration of the Algorithm is sketched in Figure 1.

Theorem 2. *Suppose that the conditions of Theorem 1 hold true. Then there exists a $\beta \in \mathcal{KL}$, and for any $\overline{r} > 0$ there exists a $h^* > 0$ such that for any fixed $N \geq N^*$, $h \in (0, h^*]$ and $x_0 \in \Gamma$, the trajectory of the ℓ-step exact discrete-time system*

$$\xi_{k+1}^E = \mathcal{F}_\ell^E(\xi_k^E, \mathbf{v}_{\ell,h}(\zeta_k^A)), \qquad \xi_0^E = \phi_{\overline{\ell}}^E(x_0, \mathbf{u}^c) \tag{15}$$

with the ℓ-step receding horizon controller $\mathbf{v}_{\ell,h}$ obtained by the prediction

$$\zeta_{k+1}^A = \mathcal{F}_{\overline{\ell}}^A\left(y_{k+1}, \mathbf{v}^p\left(\zeta_k^A\right)\right), \qquad \zeta_0^A = \phi_{\overline{\ell}}^A(x_0, \mathbf{u}^c) \tag{16}$$

satisfies that $\xi_k^E \in \Gamma_{\max}(h)$ and

$$\left\|\xi_k^E\right\| \leq \max\left\{\beta\left(\left\|\xi_0^E\right\|, kT^m\right), \overline{r}\right\}$$

for all $k \geq 0$. Moreover, $\zeta_k^A \in \Gamma_{\max}(h)$, as well, and

$$\left\|\zeta_k^A\right\| \leq \max\left\{\beta\left(\left\|\zeta_0^A\right\|, kT^m\right) + \delta_1, \overline{r}\right\}$$

where δ_1 can be made arbitrarily small by suitable choice of h.

Proof. The proof is given in the Appendix. ■

Remark 7. In the proof of Theorem 2 one also obtains that $\phi_k^E(\xi_{j-1}^E, \mathbf{u}^{(j)})$ converges to the ball $\mathcal{B}_{\overline{r}}$ as $j \to \infty$ for all k. Conclusions about the intersampling behavior can be made on the basis of [27].

Remark 8. We note that the statement of Theorem 2 is similar to the practical asymptotic stability of the closed-loop system (15)–(16) about the origin, but with respect to the initial state ξ_0^E, ζ_0^A. This is not true for the original initial state x_0, because – due to the initial phase – the ball $\mathcal{B}_{\overline{r}}$ is not invariant over the time interval $[0, \overline{\ell}T)$. In absence of measurement and computational delays, the theorem gives the practical asymptotic stability of the closed-loop system (15)–(16) about the origin in the usual sense.

3 Remarks on Other Choices of the Design Parameters

Recently, stability results have been proven for the case, when the terminal cost is not a CLF: see [19] for continuous-time and [10] discrete-time considerations. It is shown in both papers that stability can be achieved under some additional conditions, with arbitrary nonnegative terminal cost, if the time horizon is chosen to be sufficiently long. In respect of the subject of the present work the latter one plays crucial role. In fact, Theorem 1 of [10] provides a Lyapunov function having (almost) the properties which guarantee that the same family of controllers that stabilizes the approximate discrete-time model also practically stabilizes the exact discrete-time model of the plant. To this end, one has to ensure additionally the uniform Lipschitz-continuity of the Lyapunov function, presuming that assumptions of [10] are valid for the approximate discrete-time model. This assumptions can partly be transferred to the original continuous-time data similarly to the way of the previous section.

The main difficulty is connected with assumption SA4 of [10]. This assumption requires the existence of class-$\mathcal{K}_\infty$ upper bound of the value function independent of the horizon length. It is pointed out in [10] that such a bound exists if - roughly speaking - for the approximate discrete-time system the stage cost is exponentially controllable to zero with respect to an appropriate positive definite function. An appropriate choice of such a function is given in [10], if the discrete-time system is homogeneous. However, the derivation of the corresponding conditions for the original data of general systems requires further considerations. (We note that a certain version of the MPC approach for the sampled-data implementation of continuous-time stabilizing feedback laws is investigated in [12] under an assumption analogous to SA4 of [10].)

4 Conclusion

The stabilization problem of nonlinear continuous-time systems with piecewise constant control functions was investigated. The controller was computed by the receding horizon control method based on discrete-time approximate models.

Multi-rate - multistep control was considered and both measurement and computational delays were allowed. It was shown that the same family of controllers that stabilizes the approximate discrete-time model also practically stabilizes the exact discrete-time model of the plant. The conditions were formulated in terms of the original continuous-time models and the design parameters so that they could be verifiable in advance.

References

[1] Chen H, Allgöwer F (1998) A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica 34:1205–1217

[2] Chen W H, Ballance D J, O'Reilly J (2000) Model predictive control of nonlinear systems: Computational burden and stability. IEE Proc Control Theory Appl 147:387–394

[3] De Nicolao G, Magni L, Scattolini R (1998) Stabilizing receding horizon control of nonlinear time-varying system. IEEE Trans. Automat. Control 43:1030–1036

[4] De Nicolao G, Magni L, Scattolini R (2000) Stability and robustness of nonlinear receding horizon control. In: Allgöwer F, Zheng A (eds) Nonlinear Model Predictive Control. Progress in Systems and Control Theory. Birkhäuser Verlag, 3–22

[5] Elaiw A M, Gyurkovics É (2005) Multirate sampling and delays in receding horizon stabilization of nonlinear systems. In: Proc. 16th IFAC World Congress, Prague

[6] Findeisen R, Imsland L, Allgöwer F, Foss B A (2003) State and output feedback nonlinear model predictive control: An overview. European J Control 9:190–206

[7] Findeisen R, Allgöwer F (2004) Computational delay in nonlinear model predictive control. In: Proceedings of Int Symp Adv Control of Chemical Processes

[8] Fontes F A C C (2001) A general framework to design stabilizing nonlinear model predictive controllers. Systems Control Lett. 42:127–143

[9] Grimm G, Messina M J, Teel A R, Tuna S (2004) Examples when nonlinear model predictive control is nonrobust. Automatica 40:1729–1738

[10] Grimm G, Messina M J, Teel A R, Tuna S (2005) Model predictive control: for want of a local control Lyapunov function, all is not lost. IEEE Trans. Automat. Control 50:546–558

[11] Grüne L, Nešić D (2003) Optimization based stabilization of sampled-data nonlinear systems via their approximate discrete-time models. SIAM J. Control Optim 42:98–122

[12] Grüne L, Nešić D, Pannek J (2005) Model predictive control for nonlinear sampled-data systems. In: Proc of Int Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control, Freudenstadt-Lauterbad

[13] Gyurkovics É (1996) Receding horizon control for the stabilization of nonlinear uncertain systems described by differential inclusions. J. Math. Systems Estim. Control 6:1–16

[14] Gyurkovics É (1998) Receding horizon control via Bolza-type optimization. Systems Control Lett. 35:195–200

[15] Gyurkovics É, Elaiw A M (2004) Stabilization of sampled-data nonlinear systems by receding horizon control via discrete-time approximations. Automatica 40:2017–2028

[16] Gyurkovics É, Elaiw A M (2006) A stabilizing sampled-data ℓ-step receding horizon control with application to a HIV/AIDS model. Differential Equations Dynam. Systems 14:323–352

[17] Ito K, Kunisch K (2002) Asymptotic properties of receding horizon optimal control problems. SIAM J. Control Optim. 40:1585–1610
[18] Jadbabaie A, Hauser J (2001) Unconstrained receding horizon control of nonlinear systems. IEEE Trans. Automat. Control 46:776–783
[19] Jadbabaie A, Hauser J (2005) On the stability of receding horizon control with a general terminal cost. IEEE Trans. Automat. Control 50:674–678
[20] Keerthi S S, Gilbert E G (1985) An existence theorem for discrete-time infinite-horizon optimal control problems. IEEE Trans. Automat. Control 30:907–909
[21] Kwon W H, Han S H, Ahn Ch K (2004) Advances in nonlinear predictive control: A survey on stability and optimality. Int J Control Automation Systems 2:15–22
[22] Magni L, Scattolini R (2004) Model predictive control of continuous-time nonlinear systems with piecewise constant control. IEEE Trans. Automat. Control 49:900–906
[23] Mayne D Q, Michalska H (1990) Receding horizon control of nonlinear systems. IEEE Trans. Automat. Control 35:814–824
[24] Mayne D Q, Rawlings J B, Rao C V, Scokaert P O M (2000) Constrained model predictive control: Stability and optimality. Automatica 36:789–814
[25] Nešić D, Teel A R (2004) A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. IEEE Trans. Automat. Control 49:1103–1122
[26] Nešić D, Teel A R, Kokotović P V (1999) Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximation. Systems Control Lett. 38:259–270
[27] Nešić D, Teel A R, Sontag E D (1999) Formulas relating $\mathcal{KL}$ stability estimates of discrete-time and sampled-data nonlinear systems. Systems Control Lett. 38:49–60
[28] Polushin I G, Marquez H J (2004) Multirate versions of sampled-data stabilization of nonlinear systems. Automatica 40:1035–1041
[29] Parisini T, Sanguineti M, Zoppoli R (1998) Nonlinear stabilization by receding-horizon neural regulators. Int. J. Control 70:341–362

Appendix

Proof. (Proof of Theorem 1) To obtain the properties of function V_N^A we shall subsequently introduce several notations. Let $\rho_1 > 0$ be such that $\mathcal{B}_{\rho_1} \subset \mathcal{G}_\eta$,

$$\tau(s) = \begin{cases} \min\left\{T,\ s/(2M_f(2\varDelta_0, \varDelta_2^*))\right\}, & \text{if } 0 \le s \le \varDelta_0, \\ T, & \text{if } \varDelta_0 < s, \end{cases} \tag{17}$$

$$\sigma_1(s) = \varphi_1\left(s/2\right)\tau\left(s\right)/2\ , \qquad \varDelta_1^* = \max\{\sigma_1^{-1}(V_{\max}^A),\ \varDelta_0\}, \tag{18}$$

$$\nu(s) = \max_{\|x\| \le s} g(x) + \alpha_g(s)\ , \qquad r_0^* = \min\{\nu^{-1}(\eta),\ 2TM_f(2\varDelta_0, \varDelta_2^*)\}, \tag{19}$$

$$\sigma_2(s) = \max\{\nu(s)\ ,\ \nu(\rho_1/2) + 2/\rho_1 V_{\max}^A\left(s - \rho_1/2\right)\}, \tag{20}$$

$$N^* = \left[(V_{\max}^A - \eta)\ /\ \varphi_1(\rho_1)\right] + 1. \tag{21}$$

First we observe that, under the conditions of the theorem, functions σ_1 and σ_2 defined by (17)–(18) and (19)–(20), respectively, as well as function $\overline{\varphi}_1$ belong to class-$\mathcal{K}_\infty$, therefore $\varDelta_1^*$ and $\varDelta_2^*$ are well-defined. Let $h_0^* > 0$ be given by

Assumption A2 with $\Delta' = \Delta_1^*$ and $\Delta'' = \Delta_2^*$. A straightforward computation shows that for any $x \in \mathcal{X}_{\Delta_1^*}$, $u \in U_{\Delta_2^*}$ we have $\left\|\phi^E(t,x,u)\right\| \leq 2\Delta_1^*$, if $t \in [0,T]$, and $l_T^E(x,u) \geq \tau(x)\varphi_1(\|x\|/2) + T\overline{\varphi}_1(\|u\|)$. Let $0 < h_1^* \leq h_0^*$ be such that $\tau(r_0)\varphi_1(r_0/2)/2 \geq T\gamma(h_1^*)$, where γ is defined by Assumption A2 (ii). Then for all $h \in (0, h_1^*]$

$$l_{T,h}^A(x,u) \geq \sigma_1(\|x\|) + T\overline{\varphi}_1(\|u\|),$$

if $x \in \mathcal{X}_{\Delta_1^*} \setminus B_{r_0}$ and $u \in U_{\Delta_2^*}$. Therefore for any $x \in \Omega_{\Delta_0}$ problem $P_{T,h}^A(N,x)$ has an optimal solution $\mathbf{u}^*(x)$, function V_N^A is continuous in its domain, $V_N^A(0) = 0$ and $V_N^A(x) > 0$ if $x \neq 0$. Being N fixed, from Assumption A2 and Lemma 1 it follows that there exists a $0 < h_2^* \leq h_0^*$ such that for all $h \in (0, h_2^*]$ estimation $V_N^A(x) \leq V_{\max}^A$ holds, which implies that $\Omega_{\Delta_0} \subset \Gamma_{\max}(h)$. Let $0 < h_3^* \leq h_0^*$ be so small that $(L_g + 1)NT\gamma(h) \leq \alpha_g(r_0)$, if $h \in (0, h_3^*]$. Making use of Assumptions A2 and A4, one can show in a standard way that for any $x \in \mathcal{G}_\eta$

$$V_N^A(x) \leq g(x), \quad \text{if} \quad \|x\| \geq r_0, \quad \text{and} \quad V_N^A(x) \leq \nu(r_0) < \eta, \quad \text{if} \quad \|x\| < r_0. \tag{22}$$

Moreover, if $x \in \Omega_{\Delta_0}$ and for some $0 \leq j < N$, $\phi_j^A(x, \mathbf{u}^*(x)) \in \mathcal{G}_\eta$, then $\phi_N^A(x, \mathbf{u}^*(x)) \in \mathcal{G}_\eta$. As a consequence, we obtain that $\phi_N^A(x, \mathbf{u}^*(x)) \in \mathcal{G}_\eta$ for any $x \in \Omega_{\Delta_0}$, if $N \geq N^*$ and $h \in (0, h']$, where h' is chosen as $h' = \min\{h_1^*,\ h_2^*,\ h_3^*\}$. Consider a $h \in (0, h']$. Being $l_{T,h}^A(\phi_k^A(x, \mathbf{u}^*(x)), u_k^*) \leq V_N^A(x)$, if $k = 0, 1, \ldots, N-1$, the lower estimation in (12), the inclusions in (11) and inequalities (13) follow immediately. Observing that $\sigma_2(\|x\|) \geq V_{\max}^A$, if $\|x\| \geq \rho_1$, and $\sigma_2(\|x\|) \geq g(x)$, the upper estimation in (12) is a consequence of (22). Let $k \in \{1, \ldots, \ell\}$. By repeated use of Assumption A4 together with A2 one can show that for any $h \in (0, h']$ and $x \in \Gamma_{\max}(h)$

$$V_N^A\left(\phi_k^A(x, \mathbf{u}^*(x))\right) - V_N^A(x) \leq -l_{T,h}^A(x, u_0^*(x)) + (L_g + 1)kT\gamma(h).$$

Let $0 < h''$ be so small that $(L_g + 1)\ell T\gamma(h'') \leq \delta$, then (14) holds true, if $h \in (0, \min\{h', h''\}]$. Finally, using Assumption A2 it can be shown by standard arguments that there exist an $h''' > 0$, $L_V > 0$, $\delta_V > 0$ such that for any $h \in (0, h''']$, $\left|V_N^A(x) - V_N^A(y)\right| \leq L_V \|x - y\|$ holds true for all $x, y \in \Gamma_{\max}(h)$ with $\|x - y\| \leq \delta_V$ (see the proof of Lemma 7 of [15]). Choosing $\overline{h} = \min\{h', h'', h'''\}$, all statements of the theorem are true. □

Proof. (Proof of Theorem 2)

Let $\overline{r} > 0$ be arbitrary, let $d = \sigma_1(\sigma_2^{-1}(\sigma_1(\overline{r}))/2)$, let $r_0 = \sigma_2^{-1}(d)/2$ and let $\delta = \sigma_1(r_0)/2$. Let $\overline{h}$, δ_V, L_V be defined by Theorem 1 according to this r_0 and δ, and let $h' = \overline{h}$. The proof is based on the following claim:

Claim A. Let $k \in \{1, 2, \ldots, \ell\}$ be arbitrary and let d be defined above. If for $j \geq 1$ $\xi_{j-1}^E \in \Gamma_{\max}(h')$, $\zeta_{j-1}^A \in \Gamma_{\max}(h')$, and there exists a $\varepsilon_1 \in \mathcal{K}$ such that $\left\|\xi_{j-1}^E - \zeta_{j-1}^A\right\| \leq \varepsilon_1(h)$, if $0 < h \leq h'$, then there exist a $0 < h'' \leq h'$ such that for any $h \in (0, h'']$ inequality

$$\max\left\{V_N^A\left(\phi_k^E(\xi_{j-1}^E, \mathbf{u}^{(j)})\right), V_N^A(\xi_{j-1}^E)\right\} \geq d \tag{23}$$

implies that

$$V_N^A\left(\phi_k^E(\xi_{j-1}^E,\mathbf{u}^{(j)})\right)-V_N^A(\xi_{j-1}^E)\le-\sigma_1(\left\|\xi_{j-1}^E\right\|/2)/2,$$

where $\mathbf{u}^{(j)}$ is the optimal solution of problem $P_{T,h}^A(N,\zeta_{j-1}^A)$.

The proof of this claim can follow the same line as that of Theorem 2 in [26] by taking into account that with the given values of d and r_0, it makes no trouble that the estimations (12) are only valid outside of the ball $\mathcal{B}_{r_0}$. Thus we may return to the proof of the theorem. We observe first that the conditions of the claim for ξ_{j-1}^E and ζ_{j-1}^A, are valid if $j=1$ and ε_1 is chosen as $\varepsilon_1(h)=T\gamma(h)(e^{L_f\overline{\ell}T}-1)/(e^{L_fT}-1)$, where L_f is the Lipschitz constant of f. Assume that Claim A holds true for some $j\ge1$. Let h'' be defined by this claim and consider a $h\in(0,h'']$. Suppose that $V_N^A(\xi_{j-1}^E)\ge d$. Then $\left\|\xi_{j-1}^E\right\|\ge\sigma_2^{-1}(d)=2r_0$, and

$$V_N^A\left(\phi_k^E(\xi_{j-1}^E,\mathbf{u}^{(j)})\right)-V_N^A(\xi_{j-1}^E)\le-\sigma_1(\left\|\xi_{j-1}^E\right\|/2)/2\le-\sigma_1(r_0)/2,$$

hold true. Thus $\phi_k^E(\xi_{j-1}^E,\mathbf{u}^{(j)})\in\Gamma_{\max}(h)$, so that $\xi_j^E,y_j\in\Gamma_{\max}(h)$, as well, and

$$V_N^A\left(\xi_j^E\right)-V_N^A(\xi_{j-1}^E)\le-\sigma_1(r_0)/2. \tag{24}$$

Now we show that $\zeta_j^A\in\Gamma_{\max}(h)$. Let $0<h'''\le\min\{h',h''\}$ be so small that for any $h\in(0,h''']$ inequality $\varepsilon_1(h)\le\min\left\{\sigma_1(r_0)/L_V,\ 2\delta_V\right\}/2$ is satisfied. Then it can be shown that

$$\left\|\phi_k^E(y_j,\mathbf{u}^{(p,j)})-\phi_k^A(y_j,\mathbf{u}^{(p,j)})\right\|\le\varepsilon_1(h).$$

and

$$V_N^A(\zeta_j^A)=V_N^A(\zeta_j^A)-V_N^A(\xi_j^E)+V_N^A(\xi_j^E)\le V_N^A(\xi_{j-1}^E)+L_V\varepsilon_1(h)-\sigma_1(r_0)/2\le V_{\max}^A,$$

if $h\in(0,h''']$. Thus $\zeta_j^A\in\Gamma_{\max}(h)$, and the conditions of the claim hold also for $j+1$ as long as $V_N^A(\xi_{j-1}^E)\ge d$ holds. Therefore (24) implies that after finitely many steps $V_N^A(\xi_{j-1}^E)<d$ will occur. From the claim we get that $V_N^A\left(\phi_k^E(\xi_{j-1}^E,\mathbf{u}^{(j)})\right)<d$ must also be valid for $k=1,\ldots,\overline{\ell}$, thus for ξ_j^E, as well. Choosing $h^*=h'''$, one can show that the ball $\mathcal{B}_{\overline{r}}$ is positively invariant with respect to the exact and the approximate trajectories obtained during the application of the proposed Algorithm. The existence of a suitable function $\beta\in\mathcal{KL}$ can be constructed in the standard way. ■

A Computationally Efficient Scheduled Model Predictive Control Algorithm for Control of a Class of Constrained Nonlinear Systems

Mayuresh V. Kothare[1] and Zhaoyang Wan[2]

[1] Deparment of Chemical Engineering, Lehigh University, 111 Research Drive,
Bethlehem, PA 18017, U.S.A
mayuresh.kothare@lehigh.edu

[2] GE Water and Process Technology, 4636 Somerton Road, Trevose, PA 19053, U.S.A
Zhaoyang.Wan@ge.com

Summary. We present an overview of our results on stabilizing scheduled output feedback Model Predictive Control (MPC) algorithm for constrained nonlinear systems based on our previous publications [19, 20]. Scheduled MPC provides an important alternative to conventional nonlinear MPC formulations and this paper addresses the issues involved in its implementation and analysis, within the context of the NMPC05 workshop. The basic formulation involves the design of a set of local output feedback predictive controllers with their estimated regions of stability covering the desired operating region, and implement them as a single scheduled output feedback MPC which on-line switches between the set of local controllers and achieves nonlinear transitions with guaranteed stability. This algorithm provides a general framework for scheduled output feedback MPC design.

1 Introduction

Most practical control systems with large operating regions must deal with nonlinearity and constraints under output feedback control. Nonlinear Model Predictive Control (NMPC) is a powerful design technique that can stabilize processes in the presence of nonlinearities and constraints. Comprehensive reviews of state feedback NMPC algorithms can be found in [15]. In state feedback NMPC, full state information can be measured and is available as initial condition for predicting the future system behavior. In many applications, however, the system state can not be fully measured, and only output information is directly available for feedback. An output feedback NMPC algorithm can be formulated by combining the state feedback NMPC with a suitable state observer, e.g., moving horizon observer (MHE)[6] [14], extended Kalman filter [16], etc. A good overview of the observer based output feedback NMPC algorithms is provided in [4]

Besides developing efficient techniques such as multiple shooting for solving NLP [6] and parallel programming for control of nonlinear PDE systems [10], researchers have proposed various methods to simplify NMPC on-line computation. In [18], it was proposed that instead of the global optimal solution, an

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 49–62, 2007.
springerlink.com

improved feasible solution obtained at each sampling time is enough to ensure stability. In [11], a stabilizing NMPC algorithm was developed with a few control moves and an auxiliary controller implemented over the finite control horizon. In [7], stability is guaranteed through the use of an a priori control Lyapunov function (CLF) as a terminal cost without imposing terminal state constraints. In [1], nonlinear systems were approximated by linear time varying (LTV) models, and the optimal control problem was formulated as a min-max convex optimization. In [9], nonlinear systems were approximated as linear parameter varying (LPV) models, and a scheduling quasi-min-max MPC was developed with the current linear model known exactly and updated at each sampling time. A hybrid control scheme was proposed in [3] for nonlinear systems under state feedback. This control scheme embeds the implementation of MPC within the stability regions of the bounded controllers and employs these controllers as fall-back in the event that MPC is unable to achieve closed-loop stability [2, 3, 13].

For a control system with a large operating region, it is desirable for the controller to achieve satisfactory performance of the closed-loop system around all setpoints while allowing smooth transfer between them. Pseudolinearization was used in the quasi-infinite horizon NMPC formulation to obtain a closed form expression for the controller parameters as a function of the setpoint [5]. A novel gain scheduling approach was introduced in [12], in which a set of off-line local controllers are designed with their regions of stability overlapping each other, and supervisory scheduling of the local controllers can move the state through the intersections of the regions of stability of different controllers to the desired operating point with guaranteed stability.

In [20], we developed a scheduled output feedback MPC for nonlinear constrained systems, based on the scheduling ideas of [19] and [12]. The basic ideas are (1) locally represent the nonlinear system around an equilibrium point as a linear time varying (LTV) model and develop a local predictive controller with an estimate of its region of stability; (2) expand the region of stability about the desired operating point by piecing together the estimated regions of stability of a set of local predictive controllers; (3) schedule the local predictive controllers based on the local region of stability that contains the system state. The key to estalishing stability of local predictive controllers and stability of scheduling of local predictive controllers is to design an exponentially stable state feedback controller and require the state observer to deliver bounded observer error to ensure asymptotic stability of the output feedback controller. In order to faciliate a finite dimensional formulation for enforcement of exponential stability, we can either represent the local nonlinearity as a LTV model and parameterize the infinite control horizon in terms of a linear feedback law, or we can use the nonlinear model and only enforce the constraint over a finite control horizon with a terminal constraint and a terminal cost. While [20] only used the former formulation, the current paper generalizes a framework for design of scheduled output feedback MPC, which covers both finite dimensional formulations.

2 Local Output Feedback MPC for Constrained Nonlinear Systems

2.1 State Feedback and No Disturbances

Consider a discrete-time nonlinear dynamical system described by

$$x(k+1) = f(x(k), u(k)) \tag{1}$$

where $x\,(k) \in X \subseteq \mathbb{R}^n,\ u\,(k) \in U \subseteq \mathbb{R}^m$ are the system state and control input, respectively, X and U are compact sets. Assume $f(x,u) = \left[\, f_1(x,u) \cdots f_n(x,u) \,\right]^T$ are continuous differentiable in x and u.

Definition 1. *Given a set U, a point $x_0 \in X$ is an equilibrium point of the system (1) if a control $u_0 \in int(U)$ exists such that $x_0 = f(x_0, u_0)$. We call a connected set of equilibrium points an equilibrium surface.*

Suppose (x^{eq}, u^{eq}) is a point on the equilibrium surface. Within a neighborhood around $(x^{\mathrm{eq}}, u^{\mathrm{eq}})$, i.e., $\Pi_x = \{x \in \mathbb{R}^n |\ |x_r - x_r^{\mathrm{eq}}| \le \delta x_r,\ r = 1, ..., n\} \subseteq X$, and $\Pi_u = \{u \in \mathbb{R}^m |\ |u_r - u_r^{\mathrm{eq}}| \le \delta u_r,\ r = 1, ..., m\} \subseteq U$, let $\bar{x} = x - x^{\mathrm{eq}}$ and $\bar{u} = u - u^{\mathrm{eq}}$. The objective is to minimize the infinite horizon quadratic objective function

$$\min_{\bar{u}(k+i|k)} \ J_\infty(k)$$

subject to

$$|\bar{u}_r(k+i|k)| \le \delta u_{r,\max},\ \ i \ge 0,\ r = 1, 2, ..., m \tag{2}$$

$$|\bar{x}_r(k+i|k)| \le \delta x_{r,\max},\ \ i \ge 0,\ r = 1, 2, ..., n \tag{3}$$

where $J_\infty(k) = \sum_{i=0}^{\infty} [\bar{x}(k+i|k)^T Q \bar{x}(k+i|k) + \bar{u}(k+i|k)^T R\, \bar{u}(k+i|k)]$ with $Q > 0,\ \ R > 0$. To derive an upper bound on $J_\infty(k)$, define a quadratic function $V(\bar{x}) = \bar{x}^T Q(k)^{-1} \bar{x},\ Q(k) > 0$. Suppose $V(x)$ satisfies the following exponential stability constraint

$$V(\bar{x}(k+i+1|k) \le \alpha^2 V(\bar{x}(k+i|k)),\ V(\bar{x}(k|k)) \le 1,\ \alpha < \mathbf{1} \tag{4}$$

There exists a $\gamma(k) > 0$ such that

$$\begin{aligned} V(\bar{x}(k+i+1|k)) - V(\bar{x}(k+i|k)) \le -\tfrac{1}{\gamma(k)} \big[& \bar{x}(k+i|k)^T \mathcal{Q} \bar{x}(k+i|k) \\ & + \bar{u}(k+i|k)^T \mathcal{R}\, \bar{u}(k+i|k) \big] \end{aligned} \tag{5}$$

Summing (5) from $i = 0$ to $i = \infty$ and requiring $\bar{x}(\infty|k) = 0$ or $V(\bar{x}(\infty|k)) = 0$, it follows that $J_\infty(k) \le \gamma(k) V(\bar{x}(k|k)) \le \gamma(k)$. Therefore, the optimization is formulated as

$$\underset{\gamma(k), Q(k), \bar{u}(k+i|k), i \ge 0}{min} \gamma(k) \tag{6}$$

subject to (2)-(5).

Algorithm 1 (Exponentially stable MPC). Given the controller design parameter $0 < \alpha < 1$. At each sampling time k, apply $u(k) = \bar{u}(k) + u^{\mathrm{eq}}$ where $\bar{u}(k)$ is obtained from $\underset{\gamma(k),Q(k),\bar{u}(k+i|k),i\geq 0}{min} \gamma(k)$ subject to (2), (4), (5) and (8), where R is obtained offline from the maximization (7) subject to (2)-(5).

Assume that at each sampling time k, a state feedback law $\bar{u}(k) = F(\bar{x}(k))$ is used. Then an ellipsoidal feasible region of the optimization (6) can be defined as $\mathcal{S} = \left\{\bar{x} \in \mathbb{R}^n \;\middle|\; \bar{x}R^{-1}\bar{x} \leq 1\right\}$, where R is the optimal solution Q of the following maximization

$$\max_{\gamma,Q,F(\bullet)} \log\det Q \tag{7}$$

subject to (2)-(5). Then $J_\infty(k)$ is bounded by $\gamma_R\bar{x}(k)R^{-1}\bar{x}(k)$, where γ_R is the solution of γ in (7).

Replacing the state constraint (3) by $\bar{x}(k+i|k) \in \mathcal{S},\ i \geq 0$, or, equivalently

$$R - Q > 0 \tag{8}$$

which confines the current state and all future predicted states inside $\mathcal{S}$, we develop an exponentially stable MPC algorithm with an estimated region of stability.

Remark 1. Enforcement of the exponential stability constraint (4) involves an infinite control horizon. In order to faciliate a finite dimensional formulation, we can either represent the local nonlinearity as a LTV model and parameterize the infinite control horizon in terms of a linear feedback law (see [20]), or we can use the nonlinear model and only enforce the constraint over a finite control horizon with a terminal constraint and a terminal cost. In fact, the estimated region of stability $\mathcal{S} = \left\{\bar{x} \in \mathbb{R}^n \;\middle|\; \bar{x}R^{-1}\bar{x} \leq 1\right\}$ and the cost upper bound $\gamma_R\bar{x}^TR^{-1}\bar{x}$ can serve as the terminal constraint and the terminal cost, respectively. A significant difference between this paper and [20] is that this paper provides a generalized framework, which covers both of the above two finite dimensional formulations.

Theorem 1. *Consider the nonlinear system (1). Suppose* $(x^{\ eq}, u^{eq})$ *is locally stabilizable, then there exist a neighborhood* (Π_x, Π_u) *around* (x^{eq}, u^{eq}) *and a controller design parameter* $0 < \alpha < 1$ *such that Algorithm 1 exponentially stabilizes the closed-loop system with an estimated region of stability* $\mathcal{S} = \left\{\bar{x} \in \mathbb{R}^n \;\middle|\; \bar{x}^TR^{-1}\bar{x} \leq 1\right\}$.

Proof. The proof can be found in the Appendix. ■

2.2 State Feedback and Asymptotically Decaying Disturbances

Consider the nonlinear system (1) subject to the unknown additive asymptotically decaying disturbance $d(k)$, $x^p(k+1) = f(x^p(k), u(k)) + d(k)$, where we have made a distinction between the state of the perturbed system, $x^p(k)$, and the state of the unperturbed system, $x(k)$. In order for $x^p(k+1)$ to remain in the region of stability S , we develop a sufficient condition between the norm bound of $d(k)$ and the controller design parameter α. Let $\bar{x}^p(k) = x^p(k) - x^{\mathrm{eq}}$, $\bar{x}(k+1) = f(x^p(k), u(k)) - x^{\mathrm{eq}}$. Suppose $\bar{x}^p(k) \in \mathcal{S}$, (i.e., $\|\bar{x}^p(k)\|^2_{R^{-1}} \leq 1$), $\|\bar{x}^p(k+1)\|^2_{R^{-1}} = \|\bar{x}(k+1) + d(k)\|^2_{R^{-1}} = \|\bar{x}(k+1)\|^2_{R^{-1}} + 2\bar{x}(k+1)^T R^{-1} d(k) + \|d(k)\|^2_{R^{-1}}$, where $u(k)$ is computed by Algorithm 1. From (8) and (4), we know that $\|\bar{x}(k+1)\|^2_{R^{-1}} \leq \|\bar{x}(k+1)\|^2_{Q(k)^{-1}} \leq \alpha^2 \|\bar{x}^p(k)\|^2_{Q(k)^{-1}} \leq \alpha^2$. Therefore, invariance is guaranteed if $\|\bar{x}^p(k+1)\|^2_{R^{-1}} \leq \alpha^2 + 2\alpha \|d(k)\|_{R^{-1}} + \|d(k)\|^2_{R^{-1}} = (\alpha + \|d(k)\|_{R^{-1}})^2 \leq 1$. A sufficient condition for $x^p(k+1)$ to remain in the region of stability $\mathcal{S}$ is $\|d(k)\|_{R^{-1}} \leq 1 - \alpha$, which means that the disturbance should be bounded in a region $\mathcal{S}^d \triangleq \{d \in \mathbb{R}^n \mid d^T R^{-1} d \leq (1-\alpha)^2\}$. As $d(k)$ is asymptotically decaying, the closed-loop trajectory asymptotically converges to the equilibrium $(x^{\mathrm{eq}}, u^{\mathrm{eq}})$.

2.3 Output Feedback

Consider the nonlinear system (1) with a nonlinear output map

$$y(k) = h(x(k)) \in \mathbb{R}^q \tag{9}$$

where $h(x) = \begin{bmatrix} h_1(x) & \cdots & h_q(x) \end{bmatrix}^T$ are continuous differentiable. For all $x, \hat{x} \in \Pi_x$ and $u \in \Pi_u$, consider a full order nonlinear observer with a constant observer gain L_p,

$$\hat{x}(k+1) = f(\hat{x}(k), u(k)) + L_p(h(x(k)) - h(\hat{x}(k))) \tag{10}$$

The error dynamic system is $e(k+1) = f(x(k), u(k)) - f(\hat{x}(k), u(k)) - L_p(h(x(k)) - h(\hat{x}(k)))$. Define a quadratic function $V_e(x) = e^T P e,\ P > 0$. Suppose for all time $k \geq 0$, $x(k), \hat{x}(k) \in \Pi_x$ and $u(k) \in \Pi_u$, and $V_e(e)$ satisfies the following exponential convergent constraint

$$V_e(e(k+i+1|k)) \leq \rho^2 V_e(e(k+i|k)) \tag{11}$$

In order to facilitate the establishment of the relation between $\|d\|_{R^{-1}}$ and $\|e\|_P$ in §2.4, we want to find a P as close to R^{-1} as possible. Therefore, we minimize γ such that

$$\gamma R^{-1} \geq P \geq R^{-1} \tag{12}$$

Algorithm 2. Consider the nonlinear system (1) and (9) within (Π_x, Π_u) around $(x^{\mathrm{eq}}, u^{\mathrm{eq}})$. Given the observer design parameter $0 < \rho < 1$, the constant observer gain L_p of the full order observer (10) is obtained from $\min\limits_{\gamma, P, L_p} \gamma$ subject to (11) and (12).

Theorem 2. *Consider the nonlinear system (1) and (9). Suppose (x^{eq}, u^{eq}) is locally observable, then there exist a neighborhood (Π_x, Π_u) around $(x^{\ eq}, u^{\ eq})$ and an observer design parameter $0 < \rho < 1$ such that the minimization in Algorithm 2 is feasible. Furthermore, if for all time $k \geq 0$, $x(k), \hat{x}(k) \in \Pi_x$ and $u(k) \in \Pi_u$, then the observer in Algorithm 2 is exponentially convergent.*

Algorithm 3 (Local output feedback MPC for constrained nonlinear systems). Consider the nonlinear system (1) and the output map (9) within the neighborhood (Π_x, Π_u) around $(x^{\text{eq}}, u^{\text{eq}})$. Given the controller and observer design parameters $0 < \alpha < 1$ and $0 < \rho < 1$. At sampling time $k > 0$, apply $u(k) = F(k; (\hat{x}(k) - x^{\text{eq}})) + u^{\text{eq}}$, where $\hat{x}(k)$ is solved by the observer in Algorithm 2 with the output measurement $y(k-1)$ and $F(k; \bullet)$ is solved by the state feedback MPC in Algorithm 1 based on $\bar{x}(k) = \hat{x}(k) - x^{\text{eq}}$.

Proof. The proof can be found in the Appendix. ■

Now we combine the state feedback MPC in Algorithm 1 with the observer in Algorithm 2 to form a local output feedback MPC for the constrained nonlinear system.

2.4 Stability Analysis of Output Feedback MPC

For the output feedback MPC in Algorithm 3 to be feasible and asymptotically stable, it is required that for all time $k \geq 0$, $x(k), \hat{x}(k) \in \Pi_x$. In this subsection, we study conditions on $x(0)$ and $\hat{x}(0)$ such that $x(k), \hat{x}(k) \in \mathcal{S}$ is satisfied for all times $k \geq 0$. Consider the closed-loop system with the output feedback MPC in Algorithm 3,

$$\begin{aligned} x(k+1) &= f(\hat{x}(k), u(k)) + d_1(k) \\ \hat{x}(k+1) &= f(\hat{x}(k), u(k)) + d_2(k) \end{aligned}$$

with $d_1(k) = f(x(k), u(k)) - f(\hat{x}(k), u(k))$ and $d_2(k) = L_p(h(x(k)) - h(\hat{x}(k)))$. At time k, $u(k)$ is obtained by using the state feedback MPC in Algorithm 1 based on $\bar{x}(k|k) = \hat{x}(k) - x^{\text{eq}}$.

Since f is continuous differentiable, within (Π_x, Π_u) there exist $\beta_1, \beta_2 > 0$ such that $\|d_1(k)\|_{R^{-1}} \leq \beta_1 \|e(k)\|_P$ and $\|d_2(k)\|_{R^{-1}} \leq \beta_2 \|e(k)\|_P$. Suppose initially $x(0), \hat{x}(0) \in \mathcal{S}$ and $\|e(0)\|_P \leq \eta := \frac{1-\alpha}{\max\{\beta_1, \beta_2\}}$, then $\|d_1(0)\|_{R^{-1}} \leq 1 - \alpha$ and $\|d_2(0)\|_{R^{-1}} \leq 1 - \alpha$, which in turn lead to $x(1), \hat{x}(1) \in \mathcal{S}$ (see §2.2)and $\|e(1)\|_P \leq \eta$ (see §2.3). And so on. Since for all time $k \geq 0$, $x(k), \hat{x}(k) \in \mathcal{S}$, the state feedback MPC in Algorithm 1 is exponentially stable, the observer in Algorithm 2 is exponentially convergent, and the combination of both asymptotically stabilizes the closed-loop system.

Theorem 3. *Consider the nonlinear system (1) and (9). Suppose* (x^{eq}, u^{eq}) *is locally stabilizable and observable, then there exist a neighborhood* (Π_x, Π_u) *around* (x^{eq}, u^{eq}) *and controller and observer design parameters* $0 < \alpha < 1$ *and* $0 < \rho < 1$ *such that the output feedback MPC in Algorithm 3 asymptotically stabilizes the closed-loop system for any* $x(0), \hat{x}(0) \in \mathcal{S} =$
$\left\{x \in \mathbb{R}^n \,\middle|\, (x - x^{eq})^T R^{-1} (x - x^{eq}) \leq 1\right\}$ *satisfying* $\|x(0) - \hat{x}(0)\|_P \leq \eta$.

Remark 2. In fact, $\|d_1(k)\|_{R^{-1}} \leq \beta_1 \|e(k)\|_P \leq 1 - \alpha$ defines two ellipsoidal regions, i.e., $\mathcal{S}^d \triangleq \{d_1 \in \mathbb{R}^n \,|\, d_1^T R^{-1} d_1 \leq (1-\alpha)^2\}$ and $\mathcal{S}^e \triangleq \{e \in \mathbb{R}^n \,|\, \beta_1^2 e^T P e \leq (1-\alpha)^2\} \subset \mathcal{S}^d$. The effect of the optimization in (12) on the observer performance is to find the maximum $\mathcal{S}^e$ within $\mathcal{S}^d$.

2.5 Observability Analysis of Output Feedback MPC

For the output feedback MPC in Algorithm 3, the state is not measured, but from the output of the system and the estimated state, we can observe the exponential decay of the norm bound of the state estimation error, and thus observe the real state incrementally.

Consider the output feedback MPC in Algorithm 3 which can stabilize any $x(0), \hat{x}(0) \in \mathcal{S}$ satisfying $\|x(0) - \hat{x}(0)\|_P \leq \eta$. Let $T > 0$. At time $k - T \geq 0$, let $x(k-T), \hat{x}(k-T) \in \mathcal{S}$ satisfying $\|x(k-\mathcal{T}) - \hat{x}(k-\mathcal{T})\|_P \leq \eta$. During T steps, an input sequence $\{u(k-T), ..., u(k-1)\} \subset \Pi_u$ is obtained by the controller based on $\{\hat{x}(k-T), ..., \hat{x}(k-1)\} \subset \mathcal{S}$. We know that the state evolution starting from $x(k-T)$ driven by $\{u(k-T), ..., u(k-1)\}$ is inside $\mathcal{S} \subset \Pi_x$. Suppose that the state evolution $\tilde{x}(k+1) = f(\tilde{x}(k), u(k))$ starting from $\tilde{x}(k-T) = \hat{x}(k-T)$ driven by $\{u(k-T), ..., u(k-1)\}$ is also inside Π_x, then we can get

$$\begin{aligned} x(k+1) - \tilde{x}(k+1) &= f(x(k), u(k)) - f(\tilde{x}(k), u(k)) \\ y(k) - \tilde{y}(k) &= h(x(k)) - h(\tilde{x}(k)) \end{aligned}$$

Let $V_{\mathcal{T}} := \sum_{j=k-\mathcal{T}}^{k-1} \|y(j) - \tilde{y}(j)\|^2$. Suppose $(x^{\text{eq}}, u^{\text{eq}})$ is locally observable, then there exist a neighborhood (Π_x, Π_u) around $(x^{\text{eq}}, u^{\text{eq}})$ and $T, \mu > 0$ such that $V_{\mathcal{T}} \geq \mu \|x(k-\mathcal{T}) - \hat{x}(k-\mathcal{T})\|_P^2$, or equivalently, $\|x(k) - \hat{x}(k)\|_P^2 \leq \frac{\rho^{\mathcal{T}} V_{\mathcal{T}}}{\mu}$.

Theorem 4. *Consider the nonlinear system (1) and (9). Suppose* (x^{eq}, u^{eq}) *is locally stabilizable and observable, then there exist a neighborhood* (Π_x, Π_u) *around* (x^{eq}, u^{eq}) *and controller and observer design parameters* $0 < \alpha < 1$ *and* $0 < \rho < 1$, *and* $T, \mu > 0$ *such that the output feedback MPC in Algorithm 3 is asymptotically stable for any* $x(0), \hat{x}(0) \in \mathcal{S} = \left\{x \in \mathbb{R}^n \,\middle|\, (x - x^{eq})^T R^{-1} (x - x^{eq}) \leq 1\right\}$ *satisfying* $\|x(0) - \hat{x}(0)\|_P \leq \eta$. *On-line, let* $x(0), \hat{x}(0) \in \mathcal{S}$ *and* $\|x(0) - \hat{x}(0)\|_P \leq \eta$. *Apply the output feedback controller. At time* $k \geq T$, *if the state evolution starting from* $\hat{x}(k-T)$ *driven by the input sequence* $\{u(k-\mathcal{T}), ..., u(k-1)\}$ *from the controller is inside* Π_x, *then* $\|x(k) - \hat{x}(k)\|_P^2 \leq \frac{\rho^{\mathcal{T}} V_{\mathcal{T}}}{\mu}$.

Proof. The proof can be found in the Appendix. ■

3 Scheduled Output Feedback MPC for Constrained Nonlinear Systems

Algorithm 4 (Design of scheduled output feedback MPC). For the nonlinear system (1) and the output map (9), given an equilibrium surface and a desired equilibrium point $(x^{(0)}, u^{(0)})$. Let $i := 0$.

1. Specify a neighborhood $\left(\Pi_x^{(i)},\ \Pi_u^{(i)}\right)$ around $(x^{(i)}, u^{(i)})$ satisfying $\Pi_x^{(i)} \subseteq X$ and $\Pi_u^{(i)} \subseteq U$.

2. Given $0 < \alpha^{(i)} < 1$ and $0 < \rho^{(i)} < 1$, design Controller $\#i$ (Algorithm 3) with its explicit region of stability

$$\mathcal{S}^{(i)} = \left\{ x \in \mathbb{R}^n \,\middle|\, \left(x - x^{(i)}\right)^T \left(R^{(i)}\right)^{-1} \left(x - x^{(i)}\right) \leq 1 \right\}$$

   Store $x^{(i)}$, $u^{(i)}$, $\left(R^{(i)}\right)^{-1}$, $P^{(i)}$, $\eta^{(i)}$, $T^{(i)}$ and $\mu^{(i)}$ in a lookup table;
3. Select $(x^{(i+1)}, u^{(i+1)})$ satisfying $x^{(i+1)} \in int\left(\mathcal{S}_\theta^{(i)}\right)$ with

$$\mathcal{S}_\theta^{(i)} = \left\{ x \in \mathbb{R}^n \,\middle|\, \left(x - x^{(i)}\right)^T \left(R^{(i)}\right)^{-1} \left(x - x^{(i)}\right) \leq (\theta^{(i)})^2 < 1 \right\}$$

   Let $i := i + 1$ and go to step 1, until the region $\cup_{i=0}^{M} \mathcal{S}^{(i)}$ with $M = \max i$ covers a desired portion of the equilibrium surface.

Remark 3. In general, the scheduled MPC in Algorithm 4 requires a specified path on the equilibrium surface so as to extend the region of stability. An optimal path can be defined by the steady state optimization, which provides a set of operating conditions with optimal economic costs. For the same path on the equilibrium surface, the larger the number of controllers designed, the more overlap between the estimated regions of stability of two adjacent controllers, and the better the transition performance, because control switches can happen without moving the state trajectory close to the intermediate equilibrium points. Yet a larger number of controllers leads to a larger storage space for the lookup table and a longer time to do the search. So there is a trade-off between achievement of good transition performance and computational efficiency.

On-line, we implement the resulting family of local output feedback predictive controllers as a single controller whose parameters are changed if certain switching criteria are satisfied. We call such a controller scheme a scheduled output feedback MPC. For the case that $x(0), \hat{x}(0) \in \mathcal{S}^{(0)}$ satisfying $\|x(0) - \hat{x}(0)\|_{P^{(0)}} \leq \eta^{(0)}$, according to Theorem 3, Controller #0 asymptotically converges the closed-loop system to the desired equilibrium $\left(x^{(0)}, u^{(0)}\right)$. Similarly, for the case that $x(0), \hat{x}(0) \in \mathcal{S}^{(i)}$, $i \neq 0$ satisfying $\|x(0) - \hat{x}(0)\|_{P^{(i)}} \leq \eta^{(i)}$, Controller $\#i$ asymptotically converges the closed-loop system to the equilibrium $\left(x^{(i)}, u^{(i)}\right)$. Because $x^{(i)} \in int\left(\mathcal{S}_\theta^{(i-1)}\right)$, both $x(k)$ and $\hat{x}(k)$ will enter $\mathcal{S}_\theta^{(i-1)}$ in finite time. At time k, in order to switch from Controller $\#i$ to $\#(i-1)$, we need to make sure that the initial conditions for stability of Controller $\#(i-1)$ are satisfied, i.e., $x(k), \hat{x}(k) \in \mathcal{S}^{(i-1)}$ and $\|x(k) - \hat{x}(k)\|_{P^{(i-1)}} \leq \eta^{(i-1)}$.

Suppose $\hat{x}(k) \in \mathcal{S}_\theta^{(i-1)}$. We know that

$$\begin{aligned}
&\left\|x(k) - x^{(i-1)}\right\|_{\left(R^{(i-1)}\right)^{-1}} \\
&\leq \|x(k) - \hat{x}(k)\|_{\left(R^{(i-1)}\right)^{-1}} + \left\|\hat{x}(k) - x^{(i-1)}\right\|_{\left(R^{(i-1)}\right)^{-1}} \\
&\leq \|x(k) - \hat{x}(k)\|_{P^{(i-1)}} + \left\|\hat{x}(k) - x^{(i-1)}\right\|_{\left(R^{(i-1)}\right)^{-1}}
\end{aligned}$$

and $\|x(k) - \hat{x}(k)\|^2_{P^{(i-1)}} \leq \zeta^{i \to (i-1)} \|x(k) - \hat{x}(k)\|^2_{P^{(i)}}$ with $\zeta^{i \to (i-1)}$ solved by the following minimization

$$\min_{\zeta^{i \to (i-1)}} \zeta^{i \to (i-1)} \tag{13}$$

subject to $\zeta^{i \to (i-1)} > 0$ and $\zeta^{i \to (i-1)} P^{(i)} - P^{(i-1)} \geq 0$. Hence $x(k) \in \mathcal{S}^{(i-1)}$ and $\|x(k) - \hat{x}(k)\|_{P^{(i-1)}} \leq \eta^{(i-1)}$ are satisfied, if

$$\|x(k) - \hat{x}(k)\|^2_{P^{(i)}} \leq \frac{1}{\zeta^{i \to (i-1)}} \min \left(\left(1 - \theta^{(i-1)}\right)^2, \left(\eta^{(i-1)}\right)^2 \right) \tag{14}$$

From Theorem 4, we know that if Controller # i has been implemented for at least $T^{(i)}$ time steps, and if the state evolution starting from $\hat{x}(k - T^{(i)})$ driven by the input from the Controller $\#i$ is inside $\Pi_x^{(i)}$, then $\|x(k) - \hat{x}(k)\|^2_{P^{(i)}} \leq \frac{\left(\rho^{(i)}\right)^{\mathcal{T}^{(i)}} V_{\mathcal{T}}^{(i)}}{\mu^{(i)}}$. By imposing an upper bound $\delta^{i \to (i-1)}$ on $V_{\mathcal{T}}^{(i)}$, we can upper bound the state estimation error at current time k. Let

$$\delta^{i \to (i-1)} = \frac{\mu^{(i)}}{\zeta^{i \to (i-1)} \left(\rho^{(i)}\right)^{\mathcal{T}^{(i)}}} \min \left(\left(1 - \theta^{(i-1)}\right)^2, \left(\eta^{(i-1)}\right)^2 \right) \tag{15}$$

the satisfaction of (14) is guaranteed. Furthermore, because the observer is exponentially converging, for any finite $\delta^{i \to (i-1)}$, there exists a finite time such that $V_{\mathcal{T}}^{(i)} \leq \delta^{i \to (i-1)}$ is satisfied.

Algorithm 5. Off-line, construct $M+1$ local predictive controllers by Algorithm 4. On-line, given $x(0), \hat{x}(0) \in \mathcal{S}^{(i)}$ satisfying $\|x(0) - \hat{x}(0)\|_{P^{(i)}} \leq \eta^{(i)}$ for some i. Apply Controller $\#i$. Let $T^{(i)}$ be the time period during which Controller $\#i$ is implemented. If for Controller $\#i > 0$, (1) $T^{(i)} \geq T^{(i)}$, (2) $\hat{x}(k) \in \mathcal{S}_\theta^{(i-1)}$, and (3) the state evolution starting from $\hat{x}(k - T^{(i)})$ driven by the input from Controller $\#i$ is inside $\Pi_x^{(i)}$, and $V_{\mathcal{T}}^{(i)} \leq \delta^{i \to (i-1)}$, then, at the next sampling time, switch from Controller $\#i$ to Controller $\#(i-1)$; Otherwise, continue to apply Controller $\#i$.

Theorem 5. *Consider the nonlinear system (1) and the output map (9). Suppose $x(0), \hat{x}(0) \in \mathcal{S}^{(i)}$ satisfying $\|x(0) - \hat{x}(0)\|_{P^{(i)}} \leq \eta^{(i)}$ for some i, the scheduled output feedback MPC in Algorithm 5 asymptotically stabilizes the closed-loop system to the desired equilibrium $(x^{(0)}, u^{(0)})$.*

4 Example

Consider a two-tank system

$$\begin{aligned} \rho S_1 \dot{h}_1 &= -\rho A_1 \sqrt{2 g h_1} + u \\ \rho S_2 \dot{h}_2 &= \rho A_1 \sqrt{2 g h_1} - \rho A_2 \sqrt{2 g h_2} \\ y &= h_2 \end{aligned} \tag{16}$$

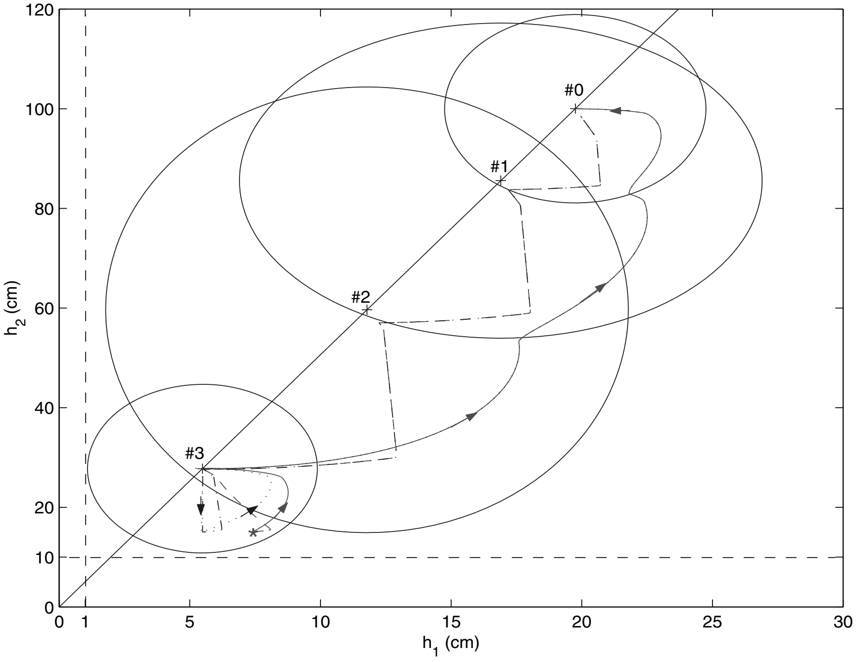


Fig. 1. Phase plots of the regulations from $x(0) = (7.5, 15)^T$ to the equilibrium $\left((19.753, 100)^T, 1.7710\right)$. First formulation: solid line - state; dotted line - estimated state. Second formulation: dashed line - state; dashed dotted line - estimated state.

where $\rho = 0.001\text{kg/cm}^3$, $g = 980\text{cm/s}^2$, $S_1 = 2500\text{cm}^2$, $A_1 = 9\text{cm}^2$, $S_2 = 1600\text{cm}^2$, $A_2 = 4\text{cm}^2$, $1\text{cm} \le h_1 \le 50\text{cm}$, $10 \text{ cm} \le h_2 \le 120\text{cm}$, and $0 \le u \le 2.5\text{kg/s}$. The sampling time is 0.5 sec. Let $Q = diag(0, 1)$, $R = 0.01$ and $\alpha = 0.998$ for all the controller designs, and $\rho = 0.99$ for all the observer designs. Let $\theta = 0.9$, $T = 10$ and $\delta = 10^{-5}$ for all switches.

Consider the regulation from an initial state $h(0) = \begin{bmatrix} 7.5 \\ 15 \end{bmatrix}$ to the equilibrium $(h^{(0)}, u^{(0)}) = \left(\begin{bmatrix} 19.753 \\ 100 \end{bmatrix}, 1.7710\right)$. Initial estimated state is $\hat{h}(0) = h^{(3)}$. Figure 1 shows four regions of stability defined by the optimization (7) for four equilibrium points. There are two controller formulations implemented within each stability region. The first formulation is to represent the local nonlinearity as a LTV model and parameterize the infinite control horizon in terms of a linear feedback law. Consider an equilibrium point $\left(\begin{bmatrix} h_1^{\text{eq}} \\ h_2^{\text{eq}} \end{bmatrix}, u^{\text{eq}}\right)$. The local nonlinearity within a neighborhood (Π_h, Π_u) is expressed as a polytopic

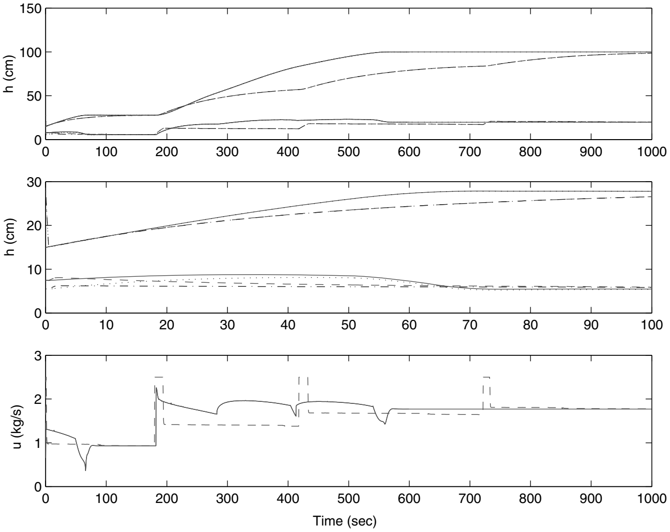


Fig. 2. Time responses of the regulations from $x(0) = (7.5, 15)^T$ to the equilibrium $\left((19.753, 100)^T, 1.7710\right)$. First formulation: solid line - state and input; dotted line - estimated state. Second formulation: dashed line - state and input; dashed dotted line - estimated state.

uncertainty Ω with four vertices $\{J(h_1^{\mathrm{eq}}+\delta h_1, h_2^{\mathrm{eq}}+\delta h_2), J(h_1^{\mathrm{eq}}+\delta h_1, h_2^{\mathrm{eq}}-\delta h_2), J(h_1^{\mathrm{eq}}-\delta h_1, h_2^{\mathrm{eq}}+\delta h_2), J(h_1^{\mathrm{eq}}-\delta h_1, h_2^{\mathrm{eq}}-\delta h_2)\}$, where $J(h_1, h_2)$ is the Jacobian matrix at $(h_1^{\mathrm{eq}}, h_2^{\mathrm{eq}})^T$. The second formulation is to use the nonlinear model within a neighborhood (Π_h, Π_u) and only enforce the constraint over a finite control horizon of $N = 3$ with a terminal constraint and a terminal cost as specified in Remark 1. Figure 1 shows the transitions by using the two controller formulations. Figure 2 shows the time responses. Close-up views of the responses of the state and the estimated state are provided to show convergence of the observers.

The first formulation performs better than the second one, because control switches of the second formulation happen close to the intermediate equilibrium points. (For performance improvement see Remark 3). On a Gateway PC with Pentium III processor (1000MHz, Cache RAM 256KB and total memory 256MB) and using Matlab LMI toolbox for the first formulation and optimization toolbox for the second approach, the numerical complexity of the two controller formulations are 0.5 second and 0.15 second per step, respectively.

5 Conclusions

In this paper, we have proposed a stabilizing scheduled output feedback MPC formulation for constrained nonlinear systems with large operating regions. Since we were able to characterize explicitly an estimated region of stability of the designed local output feedback predictive controller, we could expand it by designing multiple predictive controllers, and on-line switch between the local controllers and achieve nonlinear transitions with guaranteed stability. This algorithm provides a general framework for the scheduled output feedback MPC design. Furthermore, we have shown that this scheduled MPC is easily implementable by applying it to a two tank process.

Acknowledgments

Partial financial support for this research from the American Chemical Society's Petroleum Research Fund (ACS-PRF) and from the P. C. Rossin Assistant Professorship at Lehigh University is gratefully acknowledged.

References

[1] D. Angeli, A. Casavola, and E.Mosca. Constrained predictive control of nonlinear plant via polytopic linear system embedding. *International Journal of Robust & Nonlinear Control*, 10(13):1091–1103, Nov 2000.

[2] N.H. El-Farra, P. Mhaskar, and P.D. Christofides. Uniting bounded control and MPC for stabilization of constraed linear systems. *Automatica*, 40:101–110, 2004.

[3] N.H. El-Farra, P. Mhaskar, and P.D. Christofides. Hybrid predictive control of nonlinear systems: method and applications to chemical processes. *International Journal of Robust and Nonlinear Control*, 14:199–225, 2004.

[4] R. Findeisen, L. Imsland, F. Allgower, and B.A. Foss. State and output feedback nonlinear model predictive control: An overview. *European Journal of Control*, 9 (2-3), pages 190-206, 2003.

[5] R. Findeisen, H. Chen, and F. Allgower. Nonlinear predictive control for setpoint families. Proceedings of the 2000 American Control Conf., volume 1, pages 260-264, Chicago, 2000.

[6] R. Findeisen, M. Diehl, T. Burner, F. Allgower, H.G. Bock, and J.P. Schloder. Efficient output feedback nonlinear model predictive control. Proceedings of the 2002 American Control Conference, pages 4752–4757, Anchorage, 2002.

[7] A. Jadbabaie, J. Yu, and J. Hauser. Unconstrained receding-horizon control of nonlinear systems. *IEEE Transactions on Automatic Control*, 46(5):776–783, May 2001.

[8] H. K. Khalil. *Nonlinear systems.* Macmillan Publishing Company, 1992.

[9] Y. Lu and Y. Arkun. A scheduling quasi-min-max Model Predictive Control algorithm for nonlinear systems. *Journal of Process Control*, 12:589–604, August 2002.

[10] David L. Ma, Danesh K. Tafti, and Richard D. Braatz. Optimal control and simulation of multidimensional crystallization processes. *Computers & Chemical Engineering*, 26(7-8):1103–1116, 2002.

[11] L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37:1351–1362, 2001.

[12] M. W. McConley, B. D. Appleby, M. A. Dahleh, and E. Feron. A computationally efficient Lyapunov-based scheduling procedure for control of nonlinear systems with stability guarantees. *IEEE Transactions on Automatic Control*, 45(1):33–49, January 2000.

[13] P. Mhaskar, N.H. El-Farra, and P.D. Christofides. Hybrid predictive control of process systems. *AIChE Journal*, 50:1242–1259, 2004.

[14] H. Michalska and D. Q. Mayne. Moving horizon observers and observer-based control. *IEEE Transactions on Automatic Control*, 40(6):995–1006, June 1995.

[15] L. Magni, G. De Nicolao and R. Scattolini. Stability and robustness of nonlinear receding horizon control. In F. Allgöwer and A. Zheng, editors, Nonlinear Model Predictive Control, pages 23–44, 2000.

[16] S. L. De Oliveira Kothare and M. Morari. Contractive Model Predictive Control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 45(6):1053–1071, June 2000.

[17] D. E. Seborg, T. F. Edgar, and D. A. Mellichamp. *Process Dynamics and Control.* John Wiley & Sons, New York, 1999.

[18] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Transactions on Automatic Control*, 44(3):648 –654, 1999.

[19] Z. Wan and M. V. Kothare. Efficient scheduled stabilizing model predictive control for constrained nonlinear systems. *International Journal of Robust and Nonlinear Control*, Volume 13, Issue 3-4, Pages 331-346, Mar-Apr 2003.

[20] Z. Wan and M. V. Kothare. Efficient scheduled stabilizing output feedback model predictive control for constrained nonlinear systems. *IEEE Trans. Aut. Control*, 49(7): 1172-1177, July 2004.

Appendix

Proof. Proof of Theorem 1: Within a neighborhood (Π_x, Π_u) around $(x^{\mathrm{eq}}, u^{\mathrm{eq}})$, we locally represent the nonlinear system (1) by a LTV model $\bar{x}(k+1) = A(k)\bar{x}(k) + B(k)\,\bar{u}(k)$ with $\begin{bmatrix} A(k) \; B(k) \end{bmatrix} \in \Omega$. For all $x \in \Pi_x$ and $u \in \Pi_u$, the Jacobian matrix $\left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial u}\right] \in \Omega$ with $\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$ and $\frac{\partial f}{\partial u} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}$. It is straight forward to establish the closed-loop exponential stability within S based on the LTV model. Since the LTV model is a representation of a class of nonlinear systems including the given nonlinear system (1) within the neighborhood (Π_x, Π_u), the closed-loop nonlinear system is exponentially stable within S. ■

Proof. Proof of Theorem 2 and 4: Following the same procedure as in the proof for Theorem 1, we locally represent the nonlinear error dynamics as a LTV model $e(k+1) = (A(k) - L_p C(k))\, e(k)$ with $\left[A(k)^T \; C(k)^T \right]^T \in \Psi$. For all $x, \hat{x} \in \Pi_x$ and $u \in \Pi_u$, the Jacobian matrix $\left[\left(\frac{\partial f}{\partial x} \right)^T \; \left(\frac{\partial h}{\partial x} \right)^T \right]^T \in \Psi$ with $\frac{\partial f}{\partial x} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$ and $\frac{\partial h}{\partial x} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial h_q}{\partial x_1} & \cdots & \frac{\partial h_q}{\partial x_n} \end{bmatrix}$. It is straight forward to establish the exponential convergence of the observer and the norm bound of the state estimation error within S based on the LTV model and the nonlinear model. ■

The Potential of Interpolation for Simplifying Predictive Control and Application to LPV Systems

John Anthony Rossiter[1], Bert Pluymers[2], and Bart De Moor[2]

[1] Department Automatic Control and Systems Engineering, Mappin Street,
University of Sheffield, S1 3JD, UK
j.a.rossiter@sheffield.ac.uk

[2] Department of Electrical Engineering, ESAT-SCD-SISTA, Kasteelpark Arenberg
10, Katholieke Universiteit Leuven, B-3001 Heverlee (Leuven), Belgium
{bert.pluymers,bart.demoor}@esat.kuleuven.be

Summary. This paper first introduces several interpolation schemes, which have been derived for the linear time invariant case, but with an underlying objective of trading off performance for online computational simplicity. It is then shown how these can be extended to linear parameter varying systems, with a relatively small increase in the online computational requirements. Some illustrations are followed with a brief discussion on areas of potential development.

1 Introduction

One of the key challenges in predictive control is formulating an optimisation which can be solved fast enough while giving properties such as guaranteed closed-loop stability and recursive feasibility. Furthermore one would really like good expectations on performance. A typical compromise is between algorithm or computational complexity and performance/feasibility. This paper looks at how reparameterising the input sequence using interpolation gives one possible balance, that is, it focuses on maximising feasible regions for a given algorithm/computational complexity without sacrificing asymptotic performance. The paper also considers some of the barriers to progress and hence suggests possible avenues for further research and in particular the potential for application to nonlinear systems. Several types of interpolation will be discussed, including interpolation between control laws [17, 1], where complexity is linked to the state dimension and interpolations based on parametric programming solutions [4].

Section 2 gives background information and Section 3 introduces the conceptual thinking in how interpolation techniques can widen feasibility while restricting complexity; to aid clarity, this is introduced using linear time invariant (LTI) models. Section 4 then extends these concepts to allow application to LPV and some classes of nonlinear systems. Section 5 gives numerical illustrations and the paper finishes with a discussion.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 63–76, 2007.
springerlink.com

2 Background

This section introduces notation, the LPV model used in this paper, basic concepts of invariance, feasibility and performance, and some prediction equations.

2.1 Model and Objective

Define the LPV model (*uncertain* or *nonlinear case*) to take the form:

$$x(k+1) = A(k)x(k) + B(k)u(k), \qquad k = 0, \ldots, \infty, \tag{1a}$$

$$[A(k)\ B(k)] \in \Omega \triangleq \mathrm{Co}\{[A_1\ B_1], \ldots, [A_m\ B_m]\}, \tag{1b}$$

The specific values of $[A(k)\ B(k)]$ are assumed to be unknown at time k. Other methods [5, 6] can take knowledge of the current values of the system matrices or bounded rates of change of these matrices into account but these cases are not considered in this paper. However, it is conceivable to extend the algorithms presented in this paper to these settings as well.

When dealing with LTI models ($m = 1$), we will talk about the *nominal case.* The following feedback law is implicitly assumed :

$$u(k) = -Kx(k); \quad \forall k. \tag{2}$$

For a given feedback, the constraints at each sample are summarised as:

$$\begin{array}{l} x(k) \in \mathcal{X} = \{x : A_x x \leq \mathbf{1}\},\ \forall k \\ u(k) \in \mathcal{U} = \{u : A_u u \leq \mathbf{1}\},\ \forall k \end{array} \quad \Rightarrow \quad x(k) \in \mathcal{S}_0 = \{x : A_y x \leq \mathbf{1}\}, \forall k. \tag{3}$$

where $\mathbf{1}$ is a column vector of appropriate dimensions containing only 1's and $A_y = [A_x; -A_u K]$. We note that the results of this paper have been proven only for feedback gains giving quadratic stabilisability, that is, for feedback K, there must exist a matrix $P = P^{\mathrm{T}} > 0 \in \mathbb{R}^{n_x \times n_x}$ such that

$$\Phi_j^{\mathrm{T}} P \Phi_j \leq P, \quad \forall j, \quad \Phi_j = A_j - B_j K. \tag{4}$$

Problem 1 (Cost Objective). *For each of the algorithms discussed, the underlying aims are: to achieve* robust *stability, to optimise performance and to guarantee* robust *satisfaction of constraints. This paper uses a single objective throughout. Hence the algorithms will seek to minimise, subject to robust satisfaction of (3), an upper bound on:*

$$J = \sum_{k=0}^{\infty} (x(k)^{\mathrm{T}} Q x(k) + u(k)^{\mathrm{T}} R u(k)). \tag{5}$$

2.2 Invariant Sets

Invariant sets [2] are key to this paper and hence are introduced next.

Definition 1 (Feasibility and robust positive invariance). *Given a system, stabilizing feedback and constraints (1,2,3), a set $\mathcal{S} \subset \mathbb{R}^{n_x}$ is feasible iff $\mathcal{S} \subseteq \mathcal{S}_0$. Moreover, the set is robust positive invariant iff*

$$x \in \mathcal{S} \quad \Rightarrow \quad (A - BK)x \in \mathcal{S}, \quad \forall [A\ B] \in \Omega. \tag{6}$$

Definition 2 (MAS). *The largest feasible invariant set (no other feasible invariant set can contain states outside this set) is uniquely defined and is called the Maximal Admissible Set (MAS, [7]).*

Define the closed-loop predictions for a given feedback K as $x(k) = \Phi^k x(0)$; $u(k) = -K\Phi^{k-1}x(0)$; $\Phi = A - BK$, then, under mild conditions [7] the MAS for a controlled LTI system is given by

$$\mathcal{S} = \bigcap_{k=0}^{n} \{x : \Phi^k x \in \mathcal{S}_0\} = \{x : Mx \leq \mathbf{1}\}, \tag{7}$$

with n a finite number. In future sections, we will for the sake of brevity use the shorthand notation $\lambda S \equiv \{x : Mx \leq \lambda \mathbf{1}\}$. The MCAS (maximum control admissible set) is defined as the set of states stabilisable with robust constraint satisfaction by the specific control sequence:

$$\begin{aligned} u_i &= -Kx_i + c_i, \qquad & i &= 0, ..., n_c - 1, \\ u_i &= -Kx_i, & i &\geq n_c. \end{aligned} \tag{8}$$

By computing the predictions given a model/constraints (1,3) and control law (8), it is easy to show that, for suitable M, N, the MCAS is given as ([18, 19]):

$$\mathcal{S}_{\mathrm{MCAS}} = \{x : \exists C \ \text{ s.t. } \ Mx + NC \leq \mathbf{1}\}; \quad C = [c_0^{\mathrm{T}}\ ...\ c_{n_c-1}^{\mathrm{T}}]^{\mathrm{T}}. \tag{9}$$

In general the MAS/MCAS are polyhedral and hence ellipsoidal invariant sets [9], $\mathcal{S}_E = \{x | x^{\mathrm{T}} P x \leq 1\}$, are suboptimal in volume [12]. Nevertheless, unlike the polyhedral case, a maximum volume $\mathcal{S}_E$ is relatively straightforward to compute for the LPV case. However, recent work [11, 3] has demonstrated the tractability of algorithms to compute MAS for LPV systems. This algorithm requires an outer estimate, e.g. $\mathcal{S}_0$, constraints at each sample (also $\mathcal{S}_0$) and the model Φ.

2.3 Background for Interpolation

Define several stabilizing feedbacks $K_i, i = 1, \ldots, n$, with K_1 the preferred choice.

Definition 3 (Invariant sets). *For each K_i, define closed-loop transfer matrices Φ_{ij} and corresponding robust invariant sets $\mathcal{S}_i$ and also define the convex hull $\overline{\mathcal{S}}$:*

$$\Phi_{ij} = A_j - B_j K_i, \quad j = 1, ..., m; \quad \mathcal{S}_i = \{x : x \in \mathcal{S}_i \Rightarrow \Phi_{ij} x \in \mathcal{S}_i, \forall j\}, \tag{10}$$

$$\overline{\mathcal{S}} \triangleq \mathrm{Co}\{\mathcal{S}_1, \ldots, \mathcal{S}_n\}. \tag{11}$$

Definition 4 (Feasibility). *Let $\Phi_i(k) = A(k) - B(k)K_i$, then [1] the following input sequence and the corresponding state predictions are recursively feasible within $\overline{\mathcal{S}}$:*

$$\begin{array}{l} u(k) = -\sum_{i=1}^{n} K_i \prod_{j=0}^{k-1} \Phi_i(k-1-j)\hat{x}_i, \\ x(k) = \sum_{i=1}^{n} \prod_{j=0}^{k-1} \Phi_i(k-1-j)\hat{x}_i, \end{array} \tag{12}$$

if one ensures that

$$x(0) = \sum_{i=1}^{n} \hat{x}_i, \quad \text{with} \quad \begin{cases} \hat{x}_i = \lambda_i x_i, \\ \sum_{i=1}^{n} \lambda_i = 1, \ \lambda_i \geq 0, \\ x_i \in \mathcal{S}_i. \end{cases} \tag{13}$$

Definition 5 (Cost). *With $\tilde{x} = [\hat{x}_1^{\mathrm{T}} \ \ldots \ \hat{x}_n^{\mathrm{T}}]^{\mathrm{T}}$, Lyapunov theory gives an upper bound $\tilde{x}^{\mathrm{T}} P \tilde{x}$ on the infinite-horizon cost J for predictions (12) using:*

$$P \geq \Gamma_u^{\mathrm{T}} R \Gamma_u + \Psi_i^{\mathrm{T}} \Gamma_x^{\mathrm{T}} Q \Gamma_x \Psi_i + \Psi_i^{\mathrm{T}} P \Psi_i, \quad i = 1, \ldots, m, \tag{14}$$

with $\Psi_i = \mathrm{diag}(A_i - B_i K_1, \ldots, A_i - B_i K_n)$, $\Gamma_x = [I, \ldots, I]$, $\Gamma_u = [K_1, \ldots, K_n]$.

These considerations show that by on-line optimizing over $\tilde{x}$, one implicitly optimizes over a class of input and state sequences given by (12). Due to recursive feasibility of these input sequences, this can be implemented in a receding horizon fashion.

3 Interpolation Schemes for LTI Systems

Interpolation is a different form of methodology to the more usual MPC paradigms in that one assumes knowledge of different feedback strategies with significantly different properties. For instance one may be tuned for optimal performance and another to maximise feasibility. One then interpolates between the predictions (12) associated with these strategies to get the best performance subject to feasibility. The underlying aim is to achieve large feasible regions with fewer optimisation variables, at some small loss to performance, and hence facilitate fast sampling. This section gives a brief overview and critique of some LTI interpolation schemes; the next section considers possible extensions to the LPV case.

3.1 One Degree of Freedom Interpolations [17]

ONEDOF uses trivial colinear interpolation, hence in (12) use:

$$x = \hat{x}_1 + \hat{x}_2; \quad \hat{x}_1 = (1-\alpha)x; \quad \hat{x}_2 = \alpha x; \quad 0 \leq \alpha \leq 1. \tag{15}$$

Such a restriction implies that α is the only d.o.f., hence optimisation is trivial. Moreover, if K_1 is the optimal feedback, minimising J of (5) over predictions (15,12) is equivalent to minimising α, $\alpha \geq 0$. Feasibility is guaranteed only in $\bigcup_i \mathcal{S}_i$.

Algorithm 1. *[ONEDOFa] The first move is* $u = -[(1-\alpha)K_1 + \alpha K_2]x$ *where:*

$$\alpha = \min_{\alpha} \quad \alpha \quad \text{s.t.} \quad [M_1(1-\alpha) + M_2\alpha]x \leq \mathbf{1}; \quad 0 \leq \alpha \leq 1. \tag{16}$$

M_1 and M_2 define mutually consistent [17] invariant sets corresponding to K_1 and K_2 respectively as $\mathcal{S}_i = \{x | M_i x \leq \mathbf{1}\}$.

Algorithm 2. *[ONEDOFb] The first move is* $u = -[(1-\alpha)K_1 + \alpha K_2]x$ *where:*

$$\alpha = \min_{\alpha,\beta} \quad \alpha \quad \text{s.t.} \quad \begin{cases} M_1(1-\alpha)x \leq (1-\beta)\mathbf{1}, \\ M_2\alpha x \leq \beta\mathbf{1}, \\ 0 \leq \beta \leq 1; \quad 0 \leq \alpha \leq 1. \end{cases} \tag{17}$$

This is solved by $\alpha = (\mu - 1)/(\mu - \lambda)$ *where* $\mu = \max(M_1 x), \quad \lambda = \max(M_2 x)$*.*

Summary: It can be shown that ONEDOFa will, in general, outperform ONEDOFb and have a larger feasible region. However, a proof of recursive feasibility has not been found for ONEDOFa whereas it has for ONEDOFb. Convergence proofs only exist for some cases [17], although minor modifications to ensure this are easy to include, e.g. [16]. However, the efficacy of the method relies on the existence of a known controller K_2 with a sufficiently large feasible region.

3.2 GIMPC: MPC Using General Interpolation

GIMPC [1] improves on ONEDOF by allowing full flexibility in the decomposition (12) of x and hence ensures (a priori): (i) a guarantee of both recursive feasibility and convergence is straightforward and (ii) the feasible region is enlarged to $\overline{\mathcal{S}}$. But the number of optimisation variables increases to $n_x + 1$.

Algorithm 3 (GIMPC). *Take a system* (1)*, constraints* (3)*, cost weighting matrices* Q, R*, controllers* K_i *and invariant sets* $\mathcal{S}_i$ *and compute a suitable* P *from (14). Then, at each time instant, solve the following optimization:*

$$\min_{\hat{x}_i, \lambda_i} \quad \tilde{x}^{\mathrm{T}} P \tilde{x}, \quad \textit{subject to } (13), \tag{18}$$

and implement the input $u = -\sum_{i=1}^{n} K_i \hat{x}_i$*.*

Summary: The increased flexibility in the decomposition of x gives two benefits: (i) a guarantee of both recursive feasibility and convergence is straightforward and (ii) the feasible region is enlarged to $\overline{\mathcal{S}}$. The downside is an increase in the number of optimisation variables.

3.3 GIMPC2 Interpolations

GIMPC includes the restriction (13) that $\sum_{i=1}^{n} \lambda_i = 1, \ \lambda_i \geq 0$. However, [15] showed that such a restriction is unnecessary when the sets S_i are polyhedral. Removing the constraints on λ_i: (i) the feasible region may become substantially larger than $\overline{\mathcal{S}}$; (ii) reduces the number of optimisation variables (computation) and (iii) facilitates better performance.

Algorithm 4 (GIMPC2). *Using the same notation as algorithm 3, at each time instant, given the current state x, solve the following optimization problem on-line*

$$\min_{\hat{x}_i} \ \tilde{x}^{\mathrm{T}} P \tilde{x}, \quad \textit{subject to} \quad \begin{cases} \sum_{i=1}^{n} M_i \hat{x}_i \leq \mathbf{1}, \\ x = \sum_{i=1}^{n} \hat{x}_i, \end{cases} \tag{19}$$

and implement the input $u = -\sum_{i=1}^{n} K_i \hat{x}_i$, where the M_i defines a generalized MAS $\mathcal{S}_i'$ with mutually consistent constraints. See Algorithm 6 for details.

Summary: If the constraints on λ_i implicit in algorithm 3 (or eqn.(13)) are removed one gets two benefits: (i) the feasible region may become substantially larger (illustrated later) than $\overline{\mathcal{S}}$ and moreover (ii) the number of optimisation variables reduces. One still has guarantees of recursive feasibility and convergence. So GIMPC2 outperforms GIMPC on feasibility, performance and computational load. The main downside is that the associated set descriptions $\mathcal{S}_i'$ maybe more complex. This is discussed later, for instance in Algorithm 6.

3.4 Interpolations to Simplify Parametric Programming (IMPQP)

One area of research within parametric programming [4] solutions to MPC is how to reduce the number of regions. Interpolation is an under explored and simple avenue. Interpolation MPQP (IMPQP) [16] takes only the outer boundary of the MCAS. In any given region, the associated optimal $\mathbf{C}$ (9) can be summarised as: $x \in \mathcal{R}_i \quad \Rightarrow \quad \mathbf{C} = -K_i x + p_i$. For other x, for which a scaled version (by $1/\rho$) would lie in $\mathcal{R}_i$ on the boundary, then the following control law can be shown to give recursive feasibility and convergence:

$$\frac{x}{\rho} \in \mathcal{R}_i \quad \Rightarrow \quad \mathbf{C} = \rho(-K_i x + p_i). \tag{20}$$

Algorithm 5 (IMPQP). Offline: *Compute the MPQP solution and find the regions contributing to the boundary. Summarise the boundary of the MCAS in the form $M_b x \leq \mathbf{1}$ and store the associated regions/laws.*

Online: *Identify the active facet from $\rho = \max_j M_b(j,:)x$. With this ρ, find a feasible and convergent $\mathbf{C}$ from (20) and then perform the ONEDOFa interpolation*

$$\min_{\alpha} \alpha \quad \text{s.t.} \quad Mx + N\alpha C \leq \mathbf{1}, \tag{21}$$

and implement $u = -Kx + \alpha e_1^T C$.

Summary: For many MPQP solutions, the IMPQP algorithm [16] can be used to reduce complexity by requiring storage only of boundary regions and their associated control laws. Monte-Carlo studies demonstrated that, despite a huge reduction in set storage requirements, the closed-loop behaviour was nevertheless often close to optimal.

3.5 Other Algorithms

For reasons of space we give only a brief statement here. Other avenues currently being explored include so called Triple mode strategies [8], where the prediction structure has an extra non-linear mode to enlarge the terminal region. The design of this extra mode must take account of the LPV case. Another possibility, easily extended to the LPV case, is based on interpolation between the laws associated to the vertices of some invariant set. This technique, as with parametric methods, may suffer from issues of complexity.

4 Extensions to the LPV Case

The previous section dealt with the nominal case. This section shows how the interpolation methods can be extended to nonlinear systems which can be represented by an LPV model. In particular, it is noted that recursive feasibility was established via feasible invariant sets (MAS or MCAS). Hence, the main conjecture is that all of the interpolation algorithms carry across to the LPV case, with only small changes, as long as one can compute the corresponding invariant sets.

4.1 Invariant Sets and Interpolation for GIMPC and ONEDOFb

The GIMPC and ONEDOFb algorithms work on terms of the form $\max_j M(j,:)x_i$. For any given MAS, this value is unique and hence one can use, the set descriptions $\mathcal{S}_i$ of minimal complexity. Thus extension to the LPV case is straightforward, as long as polyhedral sets $\mathcal{S}_i$ exist and one replaces J with a suitable upper bound [1]. The implied online computational load increases marginally because the sets $\mathcal{S}_i$ for the LPV case are likely to be more complex.

An alternative method to perform interpolation in the robust setting is given in [20]. This method requires the use of nested ellipsoidal invariant sets, which can significantly restrict the size of the feasible region, but which allow interpolation to be performed without constructing a state decomposition as in (13).

4.2 Invariant Sets and Interpolation for GIMPC2 and ONEDOFa

The algorithm of [11] was defined to find the minimum complexity MAS of an LPV system for a single control law. Thus redundant constraints are removed at each iterate. However, for the GIMPC2 algorithm, constraints may need to be retained [15] even where they are redundant in the individual $\mathcal{S}_i$, because the implied constraints may not be redundant in the combined form of (16,19). Thus, the MAS must be constructed in parallel to identify and remove redundant constraints efficiently. One possibility, forming an augmented system, is introduced next. (There are alternative ways of forming an augmented system/states [15]; investigations into preferred choices are ongoing.)

Algorithm 6 (Method to find mutually consistent MAS for the LPV case).

1. *Define an augmented system*

$$X(k+1) = \Psi(k)X(k); \tag{22}$$

$$\Psi(k) = \begin{bmatrix} A(k) - B(k)K_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & A(k) - B(k)K_n \end{bmatrix}; \quad X = \begin{bmatrix} \hat{x}_1 \\ \vdots \\ \hat{x}_n \end{bmatrix}.$$

   Define a set $\hat{\Omega}$ with $\Psi \in \hat{\Omega}$, describing the allowable variation in Ψ due to the variations implied by $[A(k)\ B(k)] \in \Omega$.

2. *Constraints (3) need to be written in terms of augmented state X as follows:*

$$A_u \underbrace{[-K_1, -K_2, \cdots]}_{\hat{K}} X(k) \leq \mathbf{1}, \qquad k = 0, \dots, \infty, \tag{23a}$$

$$A_x[I, I, \cdots]X(k) \leq \mathbf{1}, \qquad k = 0, \dots, \infty. \tag{23b}$$

3. *Assume that an outer approximation to the MAS is given by (23). Then letting $u = -\hat{K}X$, this reduces to $\mathcal{S}_o = \{X : M_oX \leq \mathbf{1}\}$ where the definition of M_o is obvious.*
4. *Follow steps 2-5 of Algorithm in [11] to find the robust MAS as $\mathcal{S}_a = \{X : M_aX \leq \mathbf{1}\}$.*

Remark 1 (Feasible region for robust GIMPC2). *Given the constraint $x = \sum_{i=1}^{n} x_i$, then one can find a projection of $\mathcal{S}_a$ to x-space from X-space as follows:*

$$\mathcal{S}_{G2} = \{x : \ \exists X \text{ s.t. } M_aX \leq \mathbf{1},\ x = [I, I, \dots, I]X\}. \tag{24}$$

Algorithm 7 (GIMPC2 for the LPV case). *Given a system* (1)*, constraints* (3)*, cost weighting matrices $Q = Q^{\mathrm{T}} > 0, R = R^{\mathrm{T}} > 0$, asymptotically stabilizing controllers K_i, corresponding polyhedral robust invariant sets $\mathcal{S}_a = \{X : M_aX \leq \mathbf{1}\}$ and P satisfying* (14)*, solve on-line at each time instant, the following problem:*

$$\min_{\hat{x}_i} \tilde{x}^{\mathrm{T}} P \tilde{x}, \textit{subject to} \quad \begin{cases} x = [I, I, \dots, I]X, \\ M_aX \leq \mathbf{1}, \end{cases} \tag{25}$$

and implement input $u = -[K_1,\ K_2,\ \dots\ ,\ K_n]X$.

Theorem 1. *Algorithm 7 guarantees robust satisfaction of* (3) *and is recursively feasible and asymptotically stable for all initial states $x(0) \in \mathcal{S}_{G2}$.*

Proof: from the invariance and feasibility of $\mathcal{S}_a$, irrespective of the values $A(k), B(k)$ (or $\Psi(k)$):

$$x(k) \in \mathcal{S}_{G2} \quad \Rightarrow \quad x(k+1) \in \mathcal{S}_{G2}. \tag{26}$$

As one can always choose new state components to match the previous predictions (one step ahead), repeated choice of the same decomposition gives convergence from quadratic stability (4) associated to each K_i, and hence system Ψ. Deviation away from this will only occur where the cost $J = \tilde{x}^{\mathrm{T}} P \tilde{x}$ can be made smaller still, so the cost function (25) acts as a Lyapunov function. □

Summary: Extension to the LPV case is not straightforward for GIMPC2 and ONEDOFa because the form of constraint inequalities implicit in the algorithms is $M_1 x_1 + M_2 x_2 + ... \leq \mathbf{1}$ and this implies a fixed and mutual consistent structure in M_i; they can no longer be computed independently! This requirement can make the matrices M_i far larger than would be required by say GIMPC. Once consistent sets $\mathcal{S}_i$ have been defined, the interpolation algorithms GIMPC2 and ONEDOFa are identical to the LTI case, so long as the cost J is replaced by a suitable upper bound.

4.3 Extension of IMPQP to the LPV Case

Extension of IMPQP to the LPV case is immediate given the robust MCAS (RMCAS) with the addition of a few technical details such as the use of an upper bound on the cost-to-go. A neat algorithm to find the RMCAS makes use of an autonomous model [10] (that is model (1) in combination with control law (8)) to represent d.o.f. during transients, for instance:

$$z_{k+1} = \Psi z_k; \quad z = \begin{bmatrix} x \\ C \end{bmatrix}; \quad \Psi = \left[\begin{array}{c|c} \Phi & B \; 0 \\ \hline 0 & U \end{array}\right]; \quad U = \begin{bmatrix} 0 & I_{(n_c-1)n_u \times (n_c-1)n_u} \\ 0 & 0 \end{bmatrix}. \tag{27}$$

Given (1), Ψ has an LPV representation. Define the equivalent constraint set as $\mathcal{S}_0 = \{x : \tilde{A}_y z \leq \mathbf{1}\}$. One can now form the MAS for system (27) with these constraints using the conventional algorithm. This set, being linear in both x and C, will clearly take the form of (9) and therefore can be deployed in an MPQP algorithm. One can either form a tight upper bound on the cost [1] or a simpler, but suboptimal choice, would be $J = C^T C$. Guaranteed convergence and recursive feasibility is easy to establish and the main downside is the increase in the complexity of the RMCAS compared to the MCAS.

Summary: Application of IMPQP to the LPV case can be done through the use of an autonomous model to determine the RMCAS. Apart from the increase in offline complexity and obvious changes to the shape of the parametric solution, there is little conceptual difference between the LTI and LPV solutions.

4.4 Summary

We summarize the changes required to extend nominal interpolation algorithms to the LPV case.

1. The simplest ONEDOF interpolations can make use of a robust MAS, in minimal form, and apart from this no changes from the nominal algorithm are needed. The simplest GIMPC algorithm is similar except that the cost needs to be represented as a minimum upper bound.
2. More involved ONEDOF interpolations require non-minimal representations of the robust MAS to ensure consistency between respective $\mathcal{S}_i$, and hence require many more inequalities. The need to compute these simultaneously also adds significantly to the offline computational load.
3. The GIMPC2 algorithm requires both mutual consistency of the MAS and the cost to be replaced by a minimum upper bound.
4. Interpolation MPQP requires the robust MCAS which can be determined using an autonomous model representation, although this gives a large increase in the dimension of the invariant set algorithm. It also needs an upper bound on the predicted cost.

It should be noted that recent results [14] indicate that in the LPV case the number of additional constraints can often be reduced significantly with a modest decrease in feasibility.

5 Numerical Example

This section uses a double integrator example with non-linear dynamics, to demonstrate the various interpolation algorithms, for the LPV case only. The algorithm of [19] (denoted OMPC) but modified to make use of robust MCAS [13] is used as a benchmark.

5.1 Model and Constraints

We consider the nonlinear model and constraints:

$$\begin{aligned} x_{1,k+1} &= x_{1,k} + 0.1(1+(0.1x_{2,k})^2)x_{2,k}, \\ x_{2,k+1} &= x_{2,k} + (1+0.005x_{2,k}^2)u_k, \end{aligned} \tag{28a}$$

$$-0.5 \leq u_k \leq 1, \qquad [-10\ -10]^{\mathrm{T}} \leq x_k \leq [8\ 8]^{\mathrm{T}}, \qquad \forall k. \tag{28b}$$

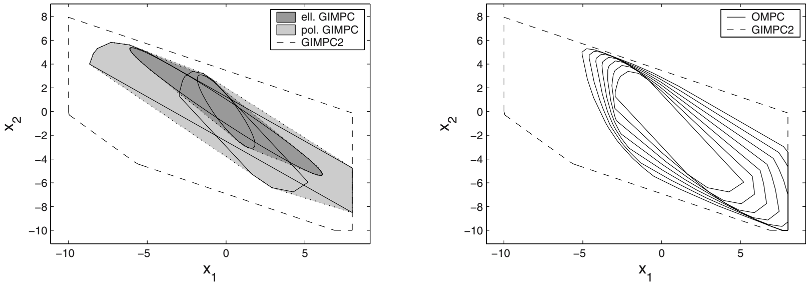
An LPV system bounding the non-linear behaviour is given as:

$$A_1 = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, B_1 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \qquad A_2 = \begin{bmatrix} 1 & 0.2 \\ 0 & 1 \end{bmatrix}, B_2 = \begin{bmatrix} 0 \\ 1.5 \end{bmatrix}. \tag{29}$$

The nominal model ($[A_1 \ B_1]$) is used to design two robustly asymptotically stabilizing feedback controllers: the first is the LQR-optimal controller $K_1 = [0.4858 \ 0.3407]^{\mathrm{T}}$ for $Q = \mathrm{diag}(1, 0.01), R = 3$ and the second $K_2 = [0.3 \ 0.4]^{\mathrm{T}}$ has a large feasible region. Both controllers are robustly asymptotically stabilizing for system (29) and are hence also stabilizing for system (28).

5.2 Feasible Regions and Computational Load

Figure 1(a) presents the feasible regions for the various interpolations and for completeness also demonstrates the improvement compared to using the largest volume invariant ellipsoids. It is clear that GIMPC2 gives substantial feasibility increases compared to GIMPC/ONEDOF and indeed also compared to IMPQP (Figure 1(b)) for $n_c = 6$. The only increase in online computation arising due to the move from LTI to LPV systems is from the number of inequalities describing the invariant sets (work in progress may reduce this significantly). For completeness table 1 shows the numbers of d.o.f. and the numbers of inequalities for each algorithm. IMPQP is excluded from this table as the online computation is linked to the number of regions and hence is fundamentally different.



(a) Feasible regions of GIMPC using ellipsoidal and polyhedral invariant sets and GIMPC2.

(b) Feasible regions of IMPQP for $n_c = 0, \ldots, 6$ and GIMPC2.

Fig. 1. Feasible regions for different algorithms for model (29) using feedback laws K_1 and K_2

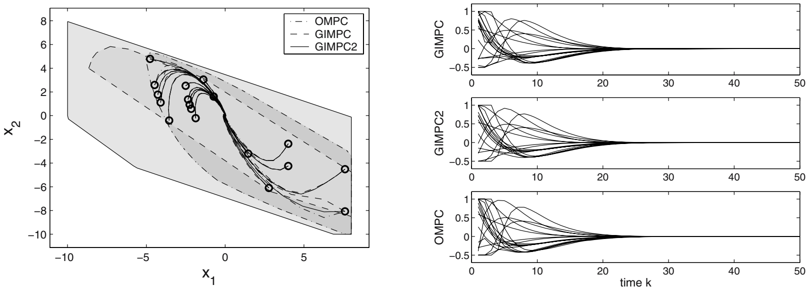
Table 1. Numbers of inequalities and d.o.f. required by GIMPC, GIMPC2 and OMPC for model (29)

	GIMPC	GIMPC2	OMPC
No. inequalities	22	63	506
No. d.o.f.	$n_x + 1 = 3$	$n_x = 2$	$n_c = 6$

5.3 Control Performance and Robust Closed-Loop Behaviour

It is useful to consider how the closed-loop performance, within the respective feasible regions, compares to 'optimal' (here taken as OMPC). Figure 2 depicts simulation results for GIMPC, GIMPC2 and OMPC, starting from initial states on the boundary of the intersection of the respective feasible regions. All three algorithms are stabilizing and result in nearly identical trajectories. The average control cost (according to (5)) of algorithms GIMPC and GIMPC2 is respectively 1.7% and 0.3% higher than OMPC with $n_c = 6$.

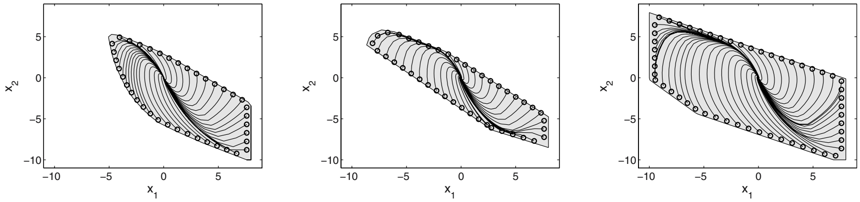
Evidence is also provided by way of closed-loop state trajectories in figure 3 that each of these algorithms is robustly feasible and convergent for the entire feasible region.



(a) State trajectories for the 3 different algorithms.

(b) Input sequences for the 3 different algorithms.

Fig. 2. Trajectories for GIMPC, GIMPC2 and OMPC for plant model (28) using feedback laws K_1 and K_2 and design model (29), starting from initial states at the boundary and the inside of the intersection of the feasible regions



(a) State trajectories for OMPC.

(b) State trajectories for GIMPC.

(c) State trajectories for GIMPC2.

Fig. 3. Trajectories for OMPC, GIMPC and GIMPC2 for plant model (28) using feedback laws K_1 and K_2 and design model (29), starting from initial states at the boundaries of the respective feasible regions

6 Conclusions and Future Directions

This paper has applied interpolation techniques to nonlinear systems which can be represented, locally, by an LPV model. The interpolation algorithms allow a degree of performance optimisation, have guarantees of recursive feasibility and convergence, while only requiring relatively trivial online computation. In fact the main requirement is the offline computation of the MAS or MCAS, with some structural restrictions. Notably, interpolations such as GIMPC2 may give far larger feasible regions than might be intuitively expected.

Nevertheless some questions are outstanding: (i) There is interest in whether interpolation concepts can be used effectively for more complicated non-linearities. (ii) This paper tackles only parameter uncertainty whereas disturbance rejection/noise should also be incorporated - some current submissions tackle that issue. (iii) It is still unclear what may be a good mechanism for identifying the underlying feedbacks K_i or strategies which give large feasible regions although Triple mode ideas [8] seem potentially fruitful. (iv) Interpolation has yet to be tested extensively on high order processes. (v) Finally, there is a need to devise efficient algorithms for computing low complexity, but large, invariant sets for high order systems.

Acknowledgments. To the Royal Society and the Royal Academy of Engineering of the United Kingdom. Research partially supported by research council KUL: GOA AMBioRICS, CoE EF/05/006 Optimization in Engineering, Flemish Government:POD Science: IUAP P5/22.

References

[1] M. Bacic, M. Cannon, Y. I. Lee, and B. Kouvaritakis. General interpolation in MPC and its advantages. *IEEE Transactions on Automatic Control*, 48(6):1092–1096, 2003.

[2] F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 1999.

[3] F. Blanchini, S. Miani, and C. Savorgnan. Polyhedral lyapunov functions computation for robust and gain scheduled design. In *Proceedings of the Symposium on nonlinear Control Systems (NOLCOS), Stuttgart, Germany*, 2004.

[4] F. Borrelli. *Constrained Optimal Control for Linear and Hybrid Systems.* Springer-Verlag, Berlin, 2003.

[5] A. Casavola, F. Domenico, and F. Giuseppe. Predictive control of constrained nonlinear systems via lpv linear embeddings. *International Journal of Robust and Nonlinear Control*, 13:281–294, 2003.

[6] L. Chisci, P. Falugi, and G. Zappa. Gain-scheduling MPC of nonlinear systems. *International Journal of Robust and Nonlinear Control*, 13:295–308, 2003.

[7] E.G. Gilbert and K. T. Tan. Linear systems with state and control constraints : The theory and application of maximal output admissible sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

[8] L. Imsland and J.A. Rossiter. Time varying terminal control. In *Proceedings of the IFAC World Congress 2005, Prague, Czech Republic*, 2005.

[9] M. V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32:1361–1379, 1996.

[10] B. Kouvaritakis, J.A. Rossiter, and J. Schuurmans. Efficient robust predictive control. *IEEE Transactions on Automatic Control*, 45(8):1545–1549, 2000.

[11] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. De Moor. The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty description. In *Proceedings of the American Control Conference (ACC), Portland, USA*, pages 804–809, 2005.

[12] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. De Moor. Interpolation based MPC for LPV systems using polyhedral invariant sets. In *Proceedings of the American Control Conference (ACC), Portland, USA*, pages 810–815, 2005.

[13] B. Pluymers, J. A. Rossiter, J. A. K. Suykens, and B. De Moor. A simple algorithm for robust MPC. In *Proceedings of the IFAC World Congress 2005, Prague, Czech Republic*, 2005.

[14] B. Pluymers, J. A. K. Suykens, and B. De Moor. Construction of reduced complexity polyhedral invariant sets for LPV systems using linear programming. *Submitted for publication*, 2005, (http://www.esat.kuleuven.be/˜sistawww/cgi-bin/pub.pl).

[15] J. A. Rossiter, Y. Ding, B. Pluymers, J. A. K. Suykens, and B. De Moor. Interpolation based MPC with exact constraint handling : the uncertain case. In *Proceedings of the joint European Control Conference & IEEE Conference on Decision and Control, Seville, Spain*, 2005.

[16] J. A. Rossiter and P. Grieder. Using interpolation to improve efficiency of multiparametric predictive control. *Automatica*, 41(4), 2005.

[17] J. A. Rossiter, B. Kouvaritakis, and M. Bacic. Interpolation based computationally efficient predictive control. *International Journal of Control*, 77(3):290–301, 2004.

[18] J. A. Rossiter, M. J. Rice, and B. Kouvaritakis. A numerically robust state-space approach to stable predictive control strategies. *Automatica*, 34:65–73, 1998.

[19] P. O. M. Scokaert and J. B. Rawlings. Constrained linear quadratic regulation. *IEEE Transactions on Automatic Control*, 43(8):1163–1168, 1998.

[20] Z. Wan and M. V. Kothare. An efficient off-line formulation of robust model predictive control using linear matrix inequalities. *Automatica*, 39(5):837–846, 2003.

Techniques for Uniting Lyapunov-Based and Model Predictive Control*

Prashant Mhaskar[1], Nael H. El-Farra[2], and Panagiotis D. Christofides[3]

[1] Department of Chemical Engineering, McMaster University, Hamilton, ON, L8P 4L7, Canada
mhaskar@mcmaster.ca

[2] Department of Chemical Engineering & Materials Science, University of California, Davis, CA 95616
nhelfarra@ucdavis.edu

[3] Department of Chemical & Biomolecular Engineering, University of California, Los Angeles, CA 90095
pdc@seas.ucla.edu

Summary. This paper presents a review of recent contributions that unite predictive control approaches with Lyapunov-based control approaches at the implementation level (Hybrid predictive control) and at the design level (Lyapunov-based predictive control) in a way that allows for an explicit characterization of the set of initial conditions starting from where closed–loop stability is guaranteed in the presence of constraints.

1 Introduction

Virtually all operation of chemical processes is subject to constraints on their manipulated inputs and state variables. Input constraints arise as a manifestation of the physical limitations inherent in the capacity of control actuators (e.g., bounds on the magnitude of valve opening), and are enforced at all times (hard constraints). State constraints, on the other hand, arise either due to the necessity to keep the state variables within acceptable ranges to avoid, for example, runaway reactions (in which case they need to be enforced at all times, and treated as hard constraints) or due to the desire to maintain them within desirable bounds dictated by performance considerations (in which case they may be relaxed, and treated as soft constraints). Constraints automatically impose limitations on our ability to steer the dynamics of the closed-loop system at will, and can cause severe deterioration in the nominal closed-loop performance and may even lead to closed-loop instability if not explicitly taken into account at the stage of controller design.

Currently, model predictive control (MPC), also known as receding horizon control (RHC), is one of the few control methods for handling state and input

* Financial support from the National Science Foundation, CTS–0129571 and CTS–0529295, is gratefully acknowledged.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 77–91, 2007.
springerlink.com

constraints within an optimal control setting and has been the subject of numerous research studies that have investigated the stability properties of MPC. In the literature, several nonlinear model predictive control (NMPC) schemes have been developed (e.g., see [2, 4, 17, 19, 20, 27, 30]) that focus on the issues of stability, constraint satisfaction, uncertainty and performance optimization for nonlinear systems. One of the key challenges that impact on the practical implementation of NMPC is the inherent difficulty of characterizing, *a priori* (i.e., before controller implementation or testing for feasibility), the set of initial conditions starting from where a given NMPC controller is guaranteed to stabilize the closed–loop system. Specifically, the stability guarantee in various MPC formulations (with or without stability conditions, and with or without robustness considerations) is contingent upon the assumption of initial feasibility, and the set of initial conditions starting from where feasibility and stability is guaranteed is not explicitly characterized. For finite–horizon MPC, an adequate characterization of the stability region requires an explicit characterization of the complex interplay between several factors, such as the initial condition, the size of the constraints and uncertainty, the horizon length, the penalty weights, etc. Use of conservatively large horizon lengths to address stability only increases the size and complexity of the optimization problem and could make it intractable.

The desire to implement control approaches that allow for an explicit characterization of their stability properties has motivated significant work on the design of stabilizing control laws using Lyapunov techniques that provide explicitly–defined regions of attraction for the closed–loop system; the reader may refer to [15] for a survey of results in this area, for a more recent review, see [5]. In [6, 7, 8], a class of Lyapunov–based bounded robust nonlinear controllers, inspired by the results on bounded control originally presented in [18], was developed. While these Lyapunov–based controllers have well–characterized stability and constraint–handling properties, they cannot, in general, be designed to be optimal with respect to a pre–specified, arbitrary cost function.

From the above discussion, it is clear that both MPC and Lyapunov–based analytic control approaches possess, by design, their own, distinct stability and optimality properties. Motivated by these considerations, this paper presents a review of recent contributions [9, 10, 22, 23, 24, 25] that unite predictive control approaches with Lyapunov-based control approaches at the implementation level (Hybrid predictive control) and at the design level (Lyapunov-based predictive control) in a way that allows for an explicit characterization of the set of initial conditions starting from where closed–loop stability is guaranteed in the presence of constraints.

2 Preliminaries

We focus on the problem of nonlinear systems with input constraints of the form:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) \tag{1}$$

$$\|u\| \leq u_{max} \tag{2}$$

where $x = [x_1 \cdots x_n]' \in \mathbb{R}^n$ denotes the vector of state variables, $u = [u_1 \cdots u_m]'$ is the vector of manipulated inputs, $u_{max} \geq 0$ denotes the bound on the manipulated inputs, $f(\cdot)$ is a sufficiently smooth $n \times 1$ nonlinear vector function, and $g(\cdot)$ is a sufficiently smooth $n \times m$ nonlinear matrix functions. Without loss of generality, it is assumed that the origin is the equilibrium point of the unforced system (i.e. $f(0) = 0$). Throughout the paper, the notation $\| \cdot \|$ will be used to denote the standard Euclidean norm of a vector, while the notation $\| \cdot \|_Q$ refers to the weighted norm, defined by $\|x\|_Q^2 = x'Qx$ for all $x \in \mathbb{R}^n$, where Q is a positive–definite symmetric matrix and x' denotes the transpose of x. In order to provide the necessary background for our results in sections 3 and 4, we will briefly review in the remainder of this section the design procedure for, and the stability properties of, both the bounded and model predictive controllers, which constitute the basic components of our controllers. We focus on the state feedback control problem where measurements of $x(t)$ are assumed to be available for all t.

2.1 Model Predictive Control

We describe here a symbolic MPC formulation that incorporates most existing MPC formulations as special cases. This is not a new formulation of MPC; the general description is only intended for the purpose of highlighting the fact that the hybrid predictive control structure can incorporate any available MPC formulation. In MPC, the control action at time t is conventionally obtained by solving, on–line, a finite horizon optimal control problem. The generic form of the optimization problem can be described as:

$$\begin{aligned} & u(\cdot) = argmin\{J_s(x, t, u(\cdot))| |u(\cdot) \in S\} \\ & s.t. \quad \dot{x}(t) = f(x(t)) + g(x)u \\ & x(0) = x_0, \ \ x(t+T) \in \Omega_{MPC}(x, t) \end{aligned} \tag{3}$$

$$J_s(x, t, u(\cdot)) = \int_t^{t+T} (x'(s)Qx(s) + u'(s)Ru(s))ds + F(x(t+T)) \tag{4}$$

and $S = S(t, T)$ is the family of piecewise continuous functions, with period Δ, mapping $[t, t+T]$ into the set of admissible controls and T is the horizon length. A control $u(\cdot)$ in S is characterized by the sequence $\{u[k]\}$ where $u[k] := u(k\Delta)$ with $u(t) = u[k]$ for all $t \in [k\Delta, (k+1)\Delta)$. J_s is the performance index, R and Q are strictly positive definite, symmetric matrices and the function $F(x(t+T))$ represents a penalty on the states at the end of the horizon. The set $\Omega_{MPC}(x, t)$ could be a fixed set, or may represent inequality constraints (as in the case of MPC formulations that require some norm of the state, or a Lyapunov function value, to decrease at the end of the horizon). The stability guarantees in MPC formulations depend on the assumption of initial feasibility and obtaining an explicit characterization of the closed–loop stability region of the predictive controller remains a difficult task.

2.2 Bounded Lyapunov-Based Control

Consider the system of Eqs.1-2, for which a family of control Lyapunov functions (CLFs), $V_k(x)$, $k \in \mathcal{K} \equiv \{1, \cdots, p\}$ has been found. Using each control Lyapunov function, we construct, using the results in [18] (see also [6, 7]), the following continuous bounded control law

$$u_k(x) = -k_k(x)(L_gV_k)'(x) \equiv b_k(x) \tag{5}$$

$$k_k(x) = \frac{L_fV_k(x) + \sqrt{(L_fV_k(x))^2 + (u_{max}\|(L_gV_k)'(x)\|)^4}}{\|(L_gV_k)'(x)\|^2\left[1 + \sqrt{1 + (u_{max}\|(L_gV_k)'(x)\|)^2}\right]} \tag{6}$$

$L_fV_k(x) = \dfrac{\partial V_k(x)}{\partial x} f(x)$, $L_gV_k(x) = [L_{g_1}V_k(x) \cdots L_{g_m}V_k(x)]'$ and $g_i(x)$ is the i-th column of the matrix $g(x)$. For the above controller, it can be shown, using standard Lyapunov arguments, that for all initial conditions within the state–space region described by the set

$$\Omega_k(u_{max}) = \{x \in \mathbb{R}^n : V_k(x) \leq c_k^{max}\} \tag{7}$$

where $c_k^{max} > 0$ is the largest number for which $\Phi_k(u_{max}) \supset \Omega_k(u_{max})\backslash\{0\}$ where

$$\Phi_k(u_{max}) = \{x \in \mathbb{R}^n : L_fV_k(x) < u_{max}\|(L_gV_k)'(x)\|\} \tag{8}$$

then the controller continues to satisfy the constraints, and the time-derivative of the Lyapunov function is negative–definite for all times. The union of the invariant regions described by the set

$$\Omega(u_{max}) = \bigcup_{k=1}^{p} \Omega_k(u_{max}) \tag{9}$$

then provides an estimate of the stability region, starting from where the origin of the constrained closed–loop system, under the appropriate control law from the family of Eqs.5-6, is guaranteed to be asymptotically stable. Note that CLF-based stabilization of nonlinear systems has been studied extensively in the nonlinear control literature (e.g., see [1, 11, 18, 29]). The construction of constrained CLFs (i.e. CLFs that take the constraints into account) remains a difficult problem (especially for nonlinear systems) that is the subject of ongoing research. For several classes of nonlinear systems that arise commonly in the modeling of practical systems, systematic and computationally feasible methods are available for constructing unconstrained CLFs (CLFs for the unconstrained system) by exploiting the system structure. Examples include the use of quadratic functions for feedback linearizable systems and the use of back-stepping techniques to construct CLFs for systems in strict feedback form. Furthermore, we note here that the bounded control law of Eqs.5-6 will be used in the remainder of the paper only to illustrate the basic idea of the proposed techniques for uniting

Lyapunov-based and predictive controllers. Our choice of using this particular design is motivated by its explicit structure and well-defined region of stability. However, our results are not restricted to this particular design and any other analytical bounded control law, with an explicit structure and well-defined region of stability, can be used.

3 Hybrid Predictive Control

By comparing the bounded controller and MPC designs presented in the previous section, some tradeoffs with respect to their stability and optimality properties are evident. The bounded controller, for example, possesses a well-defined region of admissible initial conditions that guarantee constrained closed–loop stability. However, its performance may not be optimal with respect to an arbitrary performance criterion. MPC, on the other hand, provides the desired optimality requirement, but poses implementation difficulties and lacks an explicit characterization of the stability region. In this section, we reconcile the two approaches by means of a switching scheme that provides a safety net for the implementation of MPC to nonlinear systems.

3.1 Formulation of the Switching Problem

Consider the constrained nonlinear system of Eqs.1-2, for which the bounded controllers of Eqs.5-6 and predictive controller of Eqs.3-4 have been designed. The control problem is formulated as the one of designing a set of switching laws that orchestrate the transition between MPC and the bounded controllers in a way that guarantees asymptotic stability of the origin of the closed–loop system starting from any initial condition in the set $\Omega(u_{max})$ defined in Eq.9, respects input constraints, and accommodates the optimality requirements whenever possible. For a precise statement of the problem, the system of Eq.1 is first cast as a switched system of the form

$$\dot{x} = f(x) + g(x)u_{i(t)};\ \|u_i\| \leq u_{max};\ i(t) \in \{1, 2\} \tag{10}$$

where $i : [0, \infty) \to \{1, 2\}$ is the switching signal, which is assumed to be a piecewise continuous (from the right) function of time, implying that only a finite number of switches, between the predictive and bounded controllers, is allowed on any finite interval of time. The index, $i(t)$, which takes values in the set $\{1, 2\}$, represents a discrete state that indexes the control input $u(\cdot)$, with the understanding that $i(t) = 1$ if and only if $u_i(x(t)) = M(x(t))$ and $i(t) = 2$ if and only if $u_i(x(t)) = b_k(x(t))$ for some $k \in \mathcal{K}$. Our goal is to construct a switching law $i(t) = \psi(x(t), t)$ that provides the set of switching times that ensure stabilizing transitions between the predictive and bounded controllers, in the event that the predictive controller is unable to enforce closed–loop stability. This in turn determines the time-course of the discrete state $i(t)$. While various switching schemes that focus on closed–loop stability and performance considerations to various degrees are possible [9, 10, 22, 24], we next present one example of a switching scheme (formalized in Theorem 1 below; for the proof, see [9]) that addresses the above problem while focusing on achieving closed–loop stability.

3.2 Controller Switching Logic

Theorem 1. *Consider the constrained nonlinear system of Eq.10, with any initial condition* $x(0) \equiv x_0 \in \Omega_k(u_{max})$*, for some* $k \in \mathcal{K} \equiv \{1, \cdots, p\}$*, where* Ω_k *was defined in Eq.7, under the model predictive controller of Eqs.3-4. Also let* $\bar{T} \geq 0$ *be the earliest time for which either the closed–loop state, under MPC, satisfies*

$$L_f V_k(x(\bar{T})) + L_g V_k(x(\bar{T})) M(x(\bar{T})) \geq 0 \tag{11}$$

or the MPC algorithm fails to prescribe any control move. Then, the switching rule given by

$$i(t) = \begin{Bmatrix} 1, & 0 \leq t < \bar{T} \\ 2, & t \geq \bar{T} \end{Bmatrix} \tag{12}$$

where $i(t) = 1 \Leftrightarrow u_i(x(t)) = M(x(t))$ *and* $i(t) = 2 \Leftrightarrow u_i(x(t)) = b_k(x(t))$*, guarantees that the origin of the switched closed–loop system is asymptotically stable.*

Remark 1. Theorem 1 describes a stability-based switching strategy for control of nonlinear systems with input constraints. The main components of this strategy include the predictive controller, a family of bounded nonlinear controllers, with their estimated regions of constrained stability, and a high–level supervisor that orchestrates the switching between the controllers. A schematic representation of the hybrid control structure is shown in Figure 1. The implementation procedure of this hybrid control strategy is outlined below:

- Given the system model of Eq.1, the constraints on the input and the family of CLFs, design the bounded controllers using Eqs.5-6. Given the performance objective, set up the MPC optimization problem.
- Compute the stability region estimate for each of the bounded controllers, $\Omega_k(u_{max})$, using Eqs.7-8, for $k = 1, \ldots, p$, and $\Omega(u_{max}) = \bigcup_{k=1}^{p} \Omega_k(u_{max})$.
- Initialize the closed–loop system under MPC, at any initial condition, x_0 within Ω, and identify a CLF, $V_k(x)$, for which the initial condition is within the corresponding stability region estimate, Ω_k.
- Monitor the temporal evolution of the closed–loop trajectory (by checking Eq.11 at each time) until the earliest time that either Eq.11 holds or the MPC algorithm prescribes no solution, $\bar{T}$.
- If such a $\bar{T}$ exists, discontinue MPC implementation, switch to the k-th bounded controller (whose stability region contains x_0) and implement it for all future times.

Remark 2. The main idea behind Theorem 1, and behind the hybrid predictive controller (including the designs that address issues of unavailability of

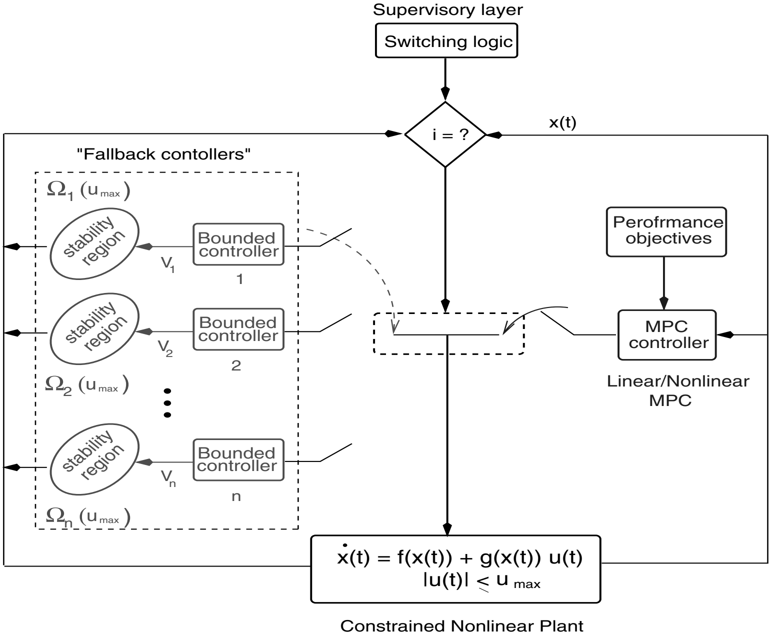


Fig. 1. Schematic representation of the hybrid control structure merging MPC and a family of fall–back bounded controllers with their stability regions

measurements and uncertainty) is as follows: first design a Lyapunov-based controller that allows for an explicit characterization of the set of initial conditions starting from where closed–loop stability is guaranteed in the presence of constraints. For an initial condition within the stability region of the Lyapunov-based controller, the predictive controller is implemented in the closed–loop system, while the supervisor monitors the evolution of the states of the closed–loop system. The supervisor checks switching rules designed to detect instability like behavior under the predictive controller and to guard against the possibility that the closed–loop trajectory under the predictive controller escapes out of the region where the Lyapunov-based controller provides the stability guarantees. In the theorem above, increase in the value of the Lyapunov-function (that is used in the design of the Lyapunov-based controller and in characterizing the stability region) is considered both as instability like behavior and to safeguard against the closed–loop state trajectory escaping the stability region (since the stability region is defined by a level set of the Lyapunov-function). The switching rule therefore dictates switching to the fall-back Lyapunov-based controller in the event of an increase in the value of the Lyapunov function.

Remark 3. The presence of constraints limits the set of initial conditions starting from where closed–loop stability can be achieved (the so called null-controllable region, or X_{max}). While a given controller design typically provides stability from subsets of the null-controllable region, it is important to be able to estimate the set of initial conditions starting from where the controller can guarantee closed–loop stability. The difficulty in characterizing the set of initial conditions starting from where a given predictive controller is guaranteed to be stabilizing motivates the use of backup controllers within the hybrid predictive control structure that provide sufficiently non-conservative estimates of their stability region. The Lyapunov-based controller of Eqs.5-7 provides such an estimate of its stability region that compares well with the null controllable region (how well, is something that can only be determined on a case by case basis; see [21] for a comparison in the case of a linear system with constraints). Note also that estimating the stability region under the controller of Eqs.5-7 requires only algebraic computations and scales well with an increase in number of system states; see [12] for applications to a polyethylene reactor and [26] for an application in the context of fault-tolerant control.

Remark 4. In addition to constraints, other important factors that influence the stabilization problem are the lack of complete measurements of the process state variables and the presence of uncertainty. The problem of lack of availability of measurements is considered in [22] where the hybrid predictive output feedback controller design comprises of the state estimator, the Lyapunov-based and predictive controllers, together with the supervisor. In addition to the set of switching rules being different from the one under state feedback, an important characteristic of the hybrid predictive control strategy under output feedback is the inherent coupling, brought about by the lack of full state measurements, between the tasks of controller design, characterization of the stability region and supervisory switching logic design, on one hand, and the task of observer design, on the other. In [24] we consider the presence of uncertainty in the design of the individual controllers as well as the switching logic in a way that enhances the chances of the use of the predictive control algorithms while not sacrificing guaranteed closed–loop stability.

4 Lyapunov-Based Predictive Control

In this section, we review our recent results on the design of a Lyapunov-based predictive controller, where the design of the (Lyapunov-based) predictive controller uses a bounded controller, with its associated region of stability, only as an auxiliary controller. The Lyapunov-based MPC is shown to possess an explicitly characterized set of initial conditions, starting from where it is guaranteed to be feasible, and hence stabilizing, while enforcing both state and input constraints at all times.

4.1 System Description

Consider the problem of stabilization of continuous-time nonlinear systems with state and input constraints, with the following state-space description:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t);\ u \in U; \quad x \in X \tag{13}$$

where $x = [x_1 \cdots x_n]' \in \mathbb{R}^n$ denotes the vector of state variables, $u = [u^1 \cdots u^m]' \in \mathbb{R}^m$ denotes the vector of manipulated inputs, $U \subseteq \mathbb{R}^m, X \subseteq \mathbb{R}^n$ denote the constraints on the manipulated inputs and the state variables, respectively, $f(\cdot)$ is a sufficiently smooth $n \times 1$ nonlinear vector function, and $g(\cdot)$ is a sufficiently smooth $n \times m$ nonlinear matrix function. Without loss of generality, it is assumed that the origin is the equilibrium point of the unforced system (i.e., $f(0) = 0$).

4.2 Lyapunov-Based Predictive Control Design

Preparatory to the characterization of the stability properties of the Lyapunov-based predictive controller, we first state the stability properties of the bounded controller of Eqs.5–6 in the presence of both state and input constraints. For the controller of Eqs.5–6, one can show, using a standard Lyapunov argument, that whenever the closed–loop state, x, evolves within the region described by the set:

$$\Phi_{x,u} = \{x \in X : L_f^* V(x) \leq u^{max} \|(L_g V)'(x)\|\} \tag{14}$$

then the controller satisfies both the state and input constraints, and the time-derivative of the Lyapunov function is negative-definite. To compute an estimate of the stability region we construct a subset of $\Phi_{x,u}$ using a level set of V, i.e.,

$$\Omega_{x,u} = \{x \in \mathbb{R}^n : V(x) \leq c_{x,u}^{max}\} \tag{15}$$

where $c_{x,u}^{max} > 0$ is the largest number for which $\Omega_{x,u} \subseteq \Phi_{x,u}$. Furthermore, the bounded controller of Eqs.5-6 possesses a robustness property (with respect to measurement errors) that preserves closed–loop stability when the control action is implemented in a discrete (sample and hold) fashion with a sufficiently small hold time (Δ). Specifically, the control law ensures that, for all initial conditions in $\Omega_{x,u}$, the closed–loop state remains in $\Omega_{x,u}$ and eventually converges to some neighborhood of the origin (we will refer to this neighborhood as Ω^b) whose size depends on Δ. This property is exploited in the Lyapunov-based predictive controller design of Section 4.2 and is stated in Proposition 1 below (the proof can be found in [25]). For further results on the analysis and control of sampled-data nonlinear systems, the reader may refer to [13, 14, 28, 31].

Proposition 1. *Consider the constrained system of Eq.1, under the bounded control law of Eqs.5–6 with* $\rho > 0$ *and let* $\Omega_{x,u}$ *be the stability region estimate under continuous implementation of the bounded controller. Let* $u(t) = u(j\Delta)$ *for all* $j\Delta \leq t < (j+1)\Delta$ *and* $u(j\Delta) = b(x(j\Delta))$, $j = 0, \cdots, \infty$. *Then, given any positive real number* d, *there exist positive real numbers* Δ^*, δ' *and* ϵ^* *such that if* $\Delta \in (0, \Delta^*]$ *and* $x(0) := x_0 \in \Omega_{x,u}$, *then* $x(t) \in \Omega_{x,u} \subseteq X$ *and* $\limsup_{t\to\infty} \|x(t)\| \leq d$. *Also, if* $V(x_0) \leq \delta'$ *then* $V(x(\tau)) \leq \delta'$ $\forall\, \tau \in [0, \Delta)$ *and if* $\delta' < V(x_0) \leq c_{x,u}^{max}$, *then* $\dot{V}(x(\tau)) \leq -\epsilon^*$ $\forall\, \tau \in [0, \Delta)$.

We present now a Lyapunov–based MPC formulation that guarantees feasibility of the optimization problem subject to hard constraints on the state and input, and hence constrained stabilization of the closed–loop system from an explicitly characterized set of initial conditions. For this MPC design, the control action at state x and time t is obtained by solving, on-line, a finite horizon optimal control problem of the form:

$$P(x,t) : \min\{J(x,t,u(\cdot))|u(\cdot) \in S,\ x \in X\} \tag{16}$$

$$s.t.\ \ \dot{x} = f(x) + g(x)u \tag{17}$$

$$\dot{V}(x(\tau)) \leq -\epsilon^* \ \ \forall\, \tau \in [t, t+\Delta)\ if\ \ V(x(t)) > \delta^{'} \tag{18}$$

$$V(x(\tau)) \leq \delta^{'}\ \forall\, \tau \in [t, t+\Delta)\ \ if\ \ V(x(t)) \leq \delta^{'} \tag{19}$$

where $S = S(t,T)$ is the family of piecewise continuous functions (functions continuous from the right), with period Δ, mapping $[t, t+T]$ into U and T is the horizon. Eq.17 is the nonlinear model describing the time evolution of the state x, V is the Lyapunov function used in the bounded controller design and $\delta^{'}$, ϵ^* are defined in Proposition 1. A control $u(\cdot)$ in S is characterized by the sequence $\{u[j]\}$ where $u[j] := u(j\Delta)$ and satisfies $u(t) = u[j]$ for all $t \in [j\Delta, (j+1)\Delta)$. The performance index is given by

$$J(x,t,u(\cdot)) = \int\limits_{t}^{t+T} \left[\|x^u(s;x,t)\|_Q^2 + \|u(s)\|_R^2\right] ds \tag{20}$$

where Q is a positive semi-definite symmetric matrix and R is a strictly positive definite symmetric matrix. $x^u(s;x,t)$ denotes the solution of Eq.1, due to control u, with initial state x at time t. The minimizing control $u^0(\cdot) \in S$ is then applied to the plant over the interval $[j\Delta, (j+1)\Delta)$ and the procedure is repeated indefinitely. Closed–loop stability and state and input constraint feasibility properties of the closed–loop system under the Lyapunov–based predictive controller are inherited from the bounded controller under discrete implementation and are formalized in Proposition 2 below (for a proof, please see [25]).

Proposition 2. *Consider the constrained system of Eq.1 under the MPC law of Eqs.16–20 with $\Delta \leq \Delta^*$ where Δ^* was defined in Proposition 1. Then, given any $x_0 \in \Omega_{x,u}$, where $\Omega_{x,u}$ was defined in Eq.15, the optimization problem of Eq.16-20 is feasible for all times, $x(t) \in \Omega_{x,u} \subseteq X$ for all $t \geq 0$ and $\limsup\limits_{t\to\infty} \|x(t)\| \leq d$.*

Remark 5. Note that the predictive controller formulation of Eqs.16–20 requires that the value of the Lyapunov function decrease during the first step only. Practical stability of the closed–loop system is achieved since, due to the receding nature of controller implementation, only the first move of the set of calculated moves is implemented and the problem is re-solved at the next time step. If the optimization problem is initially feasible and continues to be feasible, then every control move that is implemented enforces a decay in the value of the Lyapunov function, leading to stability. Lyapunov-based predictive control approaches (see, for example, [16]) typically incorporate a similar Lyapunov function decay constraint, albeit requiring the constraint of Eq.18 to hold at the *end* of the prediction horizon as opposed to only the first time step. An input trajectory that only requires the value of the Lyapunov function value to decrease at the end of the horizon may involve the state trajectory leaving the level set (and, therefore, possibly out of the state constraint satisfaction region, violating the state constraints), and motivates using a constraint that requires the Lyapunov function to decrease during the first time step (this also facilitates the explicit characterization of the feasibility region).

Remark 6. For $0 < \Delta \leq \Delta^*$, the constraint of Eq.18, is guaranteed to be satisfied (the control action computed by the bounded controller design provides a feasible initial guess to the optimization problem). Note that the constraint requires the Lyapunov function value to decay, not at the *end* of the prediction horizon (as is customarily done in Lyapunov–based MPC approaches), but only during the first time step. Furthermore, since the state is initialized in $\Omega_{x,u}$, which is a level set of V, the closed–loop system evolves so as to stay within $\Omega_{x,u}$, thereby guaranteeing feasibility at future times. Since the level set $\Omega_{x,u}$ is completely contained in the set defining the state constraints, and the state trajectory under the predictive controller continues to evolve within this set, the state constraints are satisfied at all times.

Remark 7. In the event that measurements are not continuously available, but are available only at sampling times $\Delta_s > \Delta^*$, i.e., greater than what a given bounded control design can tolerate (and, therefore, greater than the maximum allowable discretization for the Lyapunov-based predictive controller), it is necessary to redesign the bounded controller to increase the robustness margin, and generate a revised estimate of the feasibility (and stability) region under the predictive controller. A larger value of Δ^* may be achieved by increasing the value of the parameter ρ in the design of the bounded controller. If the value of the sampling time is reasonable, an increase in the value of the parameter ρ, while leading to a shrinkage in the stability region estimate, can increase Δ^* to a value greater than Δ_s and preserve the desired feasibility and stability guarantees of the Lyapunov-based predictive controller.

4.3 Switched Systems with Scheduled Mode Transitions

In many chemical processes, the system is required to follow a prescribed switching schedule, where the switching times are prescribed via an operating schedule. This practical problem motivated the development of a predictive control framework for the constrained stabilization of switched nonlinear processes that transit between their modes of operation at prescribed switching times [23]. We consider the class of switched nonlinear systems represented by the following state-space description

$$\begin{aligned} \dot{x}(t) &= f_{\sigma(t)}(x(t)) + g_{\sigma(t)}(x(t))u_{\sigma(t)}(t) \\ u_{\sigma(t)} &\in \mathcal{U}_{\sigma};\ \sigma(t) \ \in \mathcal{K} \ := \{1, \cdots, p\} \end{aligned} \tag{21}$$

where $x(t) \in \mathbb{R}^n$ denotes the vector of continuous-time state variables, $u_\sigma(t) = [u_\sigma^1(t) \cdots u_\sigma^m(t)]^T \in \mathcal{U}_\sigma \subset \mathbb{R}^m$ denotes the vector of constrained manipulated inputs taking values in a nonempty compact convex set $\mathcal{U}_\sigma := \{u_\sigma \in \mathbb{R}^m : \|u_\sigma\| \leq u_\sigma^{max}\}$, where $\|\cdot\|$ is the Euclidian norm, $u_\sigma^{max} > 0$ is the magnitude of the constraints, $\sigma : [0, \infty) \to \mathcal{K}$ is the switching signal which is assumed to be a piecewise continuous (from the right) function of time, i.e., $\sigma(t_k) = \lim_{t \to t_k^+} \sigma(t)$ for all k, implying that only a finite number of switches is allowed on any finite interval of time. p is the number of modes of the switched system, $\sigma(t)$, which takes different values in the finite index set $\mathcal{K}$, represents a discrete state that indexes the vector field $f(\cdot)$, the matrix $g(\cdot)$, and the control input $u(\cdot)$, which altogether determine $\dot{x}$.

Consider the nonlinear switched system of Eq.21, with a prescribed switching sequence (including the switching times) defined by $\mathcal{T}_{k,in}= \{t_{k_1^{in}}, t_{k_2^{in}}, \ldots\}$ and $\mathcal{T}_{k,out} = \{t_{k_1^{out}}, t_{k_2^{out}}, \ldots\}$. Also, assume that for each mode of the switched system, a Lyapunov–based predictive controller of the form of Eqs.16-20 has been designed and an estimate of the stability region generated. The control problem is formulated as the one of designing a Lyapunov-based predictive controller that guides the closed–loop system trajectory in a way that the schedule described by the switching times is followed and stability of the closed–loop system is achieved. The main idea (formalized in Theorem 2 below) is to design a Lyapunov–based predictive controller for each constituent mode in which the switched system operates, and incorporate constraints in the predictive controller design which upon satisfaction ensure that the prescribed transitions between the modes occur in a way that guarantees stability of the switched closed–loop system.

Theorem 2. *Consider the constrained nonlinear system of Eq.10, the control Lyapunov functions V_k, $k = 1, \cdots, p$, and the stability region estimates Ω_k, $k = 1, \cdots, p$ under continuous implementation of the bounded controller of Eqs.5-6 with fixed $\rho_k > 0$, $k = 1, \cdots, p$. Let $0 < T_{design} < \infty$ be a design parameter. Let t be such that $t_{k_r^{in}} \leq t < t_{k_r^{out}}$ and $t_{m_j^{in}} = t_{k_r^{out}}$ for some m, k. Consider the following optimization problem*

$$P(x,t) : \min\{J(x,t,u_k(\cdot)) | u_k(\cdot) \in S_k\} \tag{22}$$

$$J(x,t,u_k(\cdot)) = \int_t^{t+T} \left[\|x^u(s;x,t)\|_Q^2 + \|u_k(s)\|_R^2\right] ds \tag{23}$$

where T is the prediction horizon given by $T = t_{k_r^{out}} - t$, if $t_{k_r^{out}} < \infty$ and $T = T_{design}$ if $t_{k_r^{out}} = \infty$, subject to the following constraints

$$\dot{x} = f_k(x) + g_k(x)u_k \tag{24}$$

$$\dot{V}_k(x(\tau)) \leq -\epsilon_k \;\; if \;\; V_k(x(t)) > \delta_k^{'}, \; \tau \in [t, t + \Delta_{k_r}) \tag{25}$$

$$V_k(x(\tau)) \leq \delta_k^{'} \;\; if \;\; V_k(x(t)) \leq \delta_k^{'},\; \tau \in [t, t+\Delta_{k_r}) \tag{26}$$

and if $t_{k_r^{out}} = t_{m_j^{in}} < \infty$

$$V_m(x(t_{m_j^{in}})) \leq \left\{ \begin{array}{cr} V_m(x(t_{m_{j-1}^{in}})) - \epsilon^* & , j > 1, V_m(x(t_{m_{j-1}^{in}})) > \delta_m^{'} \\ \delta_m^{'} & , j > 1, V_m(x(t_{m_{j-1}^{in}})) \leq \delta_m^{'} \\ c_m^{max} & , \qquad\qquad j = 1 \end{array} \right\} \tag{27}$$

where ϵ^* *is a positive real number. Then, given a positive real number* d^{max}*, there exist positive real numbers* Δ^* *and* $\delta_k^{'}$, $k = 1, \cdots, m$ *such that if the optimization problem of Eqs.22–27 is feasible at all times, the minimizing control is applied to the system over the interval* $[t, t+\Delta_{k_r}]$*, where* $\Delta_{k_r} \in (0, \Delta^*]$ *and* $t_{k_r^{out}} - t_{k_r^{in}} = l_{k_r}\Delta_{k_r}$ *for some integer* $l_{k_r} > 0$ *and the procedure is repeated, then,* $\limsup\limits_{t\to\infty} \|x(t)\| \leq d^{max}$.

Remark 8. Note that the constraint of Eq.25 is guaranteed to be feasible between mode transitions, provided that the system is initialized within the stability region, and does not require the assumption of feasibility. This stability constraint ensures that the value of the Lyapunov function of the currently active mode keeps decreasing (recall that one of the criteria in the multiple Lyapunov-function stability analysis is that the individual modes of the switched system be stable). The constraint of Eq.25 expresses two transition requirements simultaneously: (1) the MLF constraints that requires that the value of the Lyapunov function be less than what it was the last time the system switched into that mode (required when the switching sequence is infinite, see [3] for details), and (2) the stability region constraint that requires that the state of the process reside within the stability region of the target mode at the time of the switch; since the stability regions of the modes are expressed as level sets of the Lyapunov functions, the MLF-constraint also expresses the stability region constraint. The understanding that it is a reasonably chosen switching schedule (that is, one that does not result in closed–loop instability), motivates assuming the feasibility of the transition constraints for all times. Note that the feasibility of the transition constraints can also be used to validate the switching schedule, and can be used to abort the switching schedule (i.e., to decide that the remaining switches should not be carried out) in the interest of preserving closed–loop stability.

References

[1] Z. Artstein. Stabilization with relaxed control. *Nonlinear Analysis*, 7:1163–1173, 1983.

[2] A. Bemporad and M. Morari. Robust model predictive control: A survey. In : A. Garulli, A. Tesi and A. Vicino (Eds.)*, Robustness in Identification and Control, Lecture Notes in Control and Information Sciences* Vol. 245, pages 207–266, Berlin: Springer, 1999.

[3] M. S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Trans. Automat. Contr.*, 43:475–482, 1998.

[4] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34:1205–1217, 1998.

[5] P. D. Christofides and N. H. El-Farra. *Control of Nonlinear and Hybrid Process Systems: Designs for Uncertainty, Constraints and Time-Delays.* Springer-Verlag, Berlin, Germany, 2005.

[6] N. H. El-Farra and P. D. Christofides. Integrating robustness, optimality and constraints in control of nonlinear processes. *Chem. Eng. Sci.*, 56:1841–1868, 2001.

[7] N. H. El-Farra and P. D. Christofides. Bounded robust control of constrained multivariable nonlinear processes. *Chem. Eng. Sci.*, 58:3025–3047, 2003.

[8] N. H. El-Farra and P. D. Christofides. Coordinating feedback and switching for control of hybrid nonlinear processes. *AIChE J.*, 49:2079–2098, 2003.

[9] N. H. El-Farra, P. Mhaskar, and P. D. Christofides. Hybrid predictive control of nonlinear systems: Method and applications to chemical processes. *Int. J. Rob. & Nonlin. Contr.*, 4:199–225, 2004.

[10] N. H. El-Farra, P. Mhaskar, and P. D. Christofides. Uniting bounded control and MPC for stabilization of constrained linear systems. *Automatica*, 40:101–110, 2004.

[11] R. A. Freeman and P. V. Kokotovic. *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques.* Birkhauser, Boston, 1996.

[12] A. Gani, P. Mhaskar and P. D. Christofides. Fault-Tolerant Control of a Polyethylene Reactor, *J. Proc. Contr.*, 17: 439–451, 2007.

[13] J. W. Grizzle and P. V. Kokotovic. Feedback linearization of sampled-data systems. *IEEE. Trans. Automat. Contr.*, 33:857–859, 1988.

[14] N. Kazantzis. A functional equations approach to nonlinear discrete-time feedback stabilization through pole-placement. *Syst. & Contr. Lett.*, 43:361–369, 2001.

[15] P. Kokotovic and M. Arcak. Constructive nonlinear control: a historical perspective. *Automatica*, 37:637–662, 2001.

[16] S. L. D. Kothare and M. Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Trans. Automat. Contr.*, 45:1053–1071, 2000.

[17] D. Limon, J. M. Bravo, T. Alamo, and E. F. Camacho. Robust MPC of constrained nonlinear systems based on interval arithmetic. *IEE Proc.-Contr. Th.* & *App.*, 152:325–332, 2005.

[18] Y. Lin and E. D. Sontag. A universal formula for stabilization with bounded controls. *Syst. & Contr. Lett.*, 16:393–397, 1991.

[19] L. Magni, G. Nicolao, R. Scattolini, and F. Allgower. Robust model predictive control for nonlinear discrete-time systems. *Int. J. Rob. & Non. Contr.*, 13:229–246, 2003.

[20] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[21] P. Mhaskar. *Hybrid Predictive and Fault-Tolerant Control of Chemical Processes.* Ph.D. thesis, University of California, Los Angeles, 2005.

[22] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Hybrid predictive control of process systems. *AIChE J.*, 50:1242–1259, 2004.

[23] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Predictive control of switched nonlinear systems with scheduled mode transitions. *IEEE Trans. Automat. Contr.*, 50:1670–1680, 2005.

[24] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Robust hybrid predictive control of nonlinear systems. *Automatica*, 41:209–217, 2005.

[25] P. Mhaskar, N. H. El-Farra, and P. D. Christofides. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Syst. & Contr. Lett.,* 55: 650–659, 2006.

[26] P. Mhaskar, A. Gani, and P. D. Christofides. Fault-tolerant control of nonlinear processes: Performance-based reconfiguration and robustness. *Int. J. Rob. & Nonlin. Contr.*, 16:91–111, 2006.

[27] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. Automat. Contr.*, 38:1623–1633, 1993.

[28] D. Nesic, A. R. Teel, and P. V. Kokotovic. Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Syst. & Contr. Lett.*, 38:259–270, 1999.

[29] R. Sepulchre, M. Jankovic, and P. Kokotovic. *Constructive Nonlinear Control.* Springer-Verlag, Berlin-Heidelberg, 1997.

[30] M. Sznaier, R. Suarez, and J. Cloutier. Suboptimal control of constrained nonlinear systems via receding horizon constrained control Lyapunov functions. *Int. J. Rob. & Nonlin. Contr.*, 13:247–259, 2003.

[31] L. Zaccarian, A. R. Teel, and D. Nesic. On finite gain L_P stability of nonlinear sampled-data systems. *Syst. & Contr. Lett.*, 49:201–212, 2003.

Discrete-Time Non-smooth Nonlinear MPC: Stability and Robustness

M. Lazar[1], W.P.M.H. Heemels[1], A. Bemporad[2], and S. Weiland[1]

[1] Eindhoven University of Technology, The Netherlands
m.lazar@tue.nl, m.heemels@tue.nl, s.weiland@tue.nl
[2] Universita di Siena, Italy
bemporad@dii.unisi.it

Summary. This paper considers discrete-time nonlinear, possibly discontinuous, systems in closed-loop with model predictive controllers (MPC). The aim of the paper is to provide a priori sufficient conditions for asymptotic stability in the Lyapunov sense and input-to-state stability (ISS), while allowing for both the system dynamics and the value function of the MPC cost to be discontinuous functions of the state. The motivation for this work lies in the recent development of MPC for hybrid systems, which are inherently discontinuous and nonlinear. For a particular class of discontinuous piecewise affine systems, a new MPC set-up based on infinity norms is proposed, which is proven to be ISS to bounded additive disturbances. This ISS result does not require continuity of the system dynamics nor of the MPC value function.

1 An Introductory Survey

One of the problems in model predictive control (MPC) that has received an increased attention over the years consists in guaranteeing closed-loop stability for the controlled system. The usual approach to ensure stability in MPC is to consider the value function of the MPC cost as a candidate Lyapunov function. Then, if the system dynamics is continuous, the classical Lyapunov stability theory [1] can be used to prove that the MPC control law is stabilizing [2]. The requirement that the system dynamics must be continuous is (partially) removed in [3, 4], where terminal equality constraint MPC is considered. In [3], continuity of the system dynamics on a neighborhood of the origin is still used to prove Lyapunov stability, but not for proving attractivity. Although continuity of the system is still assumed in [4], the Lyapunov stability proof (Theorem 2 in [4]) does not use the continuity property. Later on, an exponential stability result is given in [5] and an asymptotic stability theorem is presented in [6], where sub-optimal MPC is considered. The theorems of [5, 6] explicitly point out that both the system dynamics and the candidate Lyapunov function only need to be continuous at the equilibrium.

Next to closed-loop stability, one of the most studied properties of MPC controllers is robustness. Previous results developed for *smooth* nonlinear MPC, such as the ones in [5, 7], prove that robust asymptotic stability is achieved, if the system dynamics, the MPC value function and the MPC control law are

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 93–103, 2007.
springerlink.com

Lipschitz continuous. Sufficient conditions for input-to-state stability (ISS) [8] of smooth nonlinear MPC were presented in [9, 10] based on Lipschitz continuity of the system dynamics. A similar result was obtained in [11], where the Lipschitz continuity assumption was relaxed to basic continuity. An important warning regarding robustness of smooth nonlinear MPC was issued in [12], where it is pointed out that the absence of a *continuous Lyapunov function* may result in a closed-loop system that has no robustness.

This paper is motivated by the recent development of MPC for hybrid systems, which are inherently discontinuous and nonlinear systems. Attractivity was proven for the equilibrium of the closed-loop system in [13, 14]. However, proofs of Lyapunov stability only appeared in the hybrid MPC literature recently, e.g. [15, 16, 17, 18]. In [17], the authors provide *a priori sufficient conditions* for asymptotic stability in the Lyapunov sense for *discontinuous* piecewise affine (PWA) systems in closed-loop with MPC controllers based on ∞-norm cost functions. Results on robust hybrid MPC were presented in [15] and [19], where dynamic programming and tube based approaches were considered for solving feedback *min-max* MPC optimization problems for *continuous* PWA systems.

In this paper we consider discrete-time nonlinear, *possibly discontinuous*, systems in closed-loop with MPC controllers and we aim at providing a general theorem on asymptotic stability in the Lyapunov sense that unifies most of the previously-mentioned results. Besides closed-loop stability, the issue of *robustness* is particularly relevant for hybrid systems and MPC because, in this case, the system dynamics, the MPC value function and the MPC control law are typically discontinuous. We present an input-to-state stability theorem that can be applied to discrete-time non-smooth nonlinear MPC. For a class of *discontinuous* PWA systems, a new MPC set-up based on ∞-norm cost functions is proposed, which is proven to be ISS with respect to bounded additive disturbances.

2 Preliminaries

Let $\mathbb{R}$, $\mathbb{R}_+$, $\mathbb{Z}$ and $\mathbb{Z}_+$ denote the field of real numbers, the set of non-negative reals, the set of integers and the set of non-negative integers, respectively. We use the notation $\mathbb{Z}_{\geq c_1}$ and $\mathbb{Z}_{(c_1,c_2]}$ to denote the sets $\{k \in \mathbb{Z}_+ \mid k \geq c_1\}$ and $\{k \in \mathbb{Z}_+ \mid c_1 < k \leq c_2\}$, respectively, for some $c_1, c_2 \in \mathbb{Z}_+$. We define with $\mathbb{Z}^N$ the N-dimensional Cartesian product $\mathbb{Z} \times \ldots \times \mathbb{Z}$, for some $N \in \mathbb{Z}_{\geq 1}$. For a sequence $\{z_j\}_{j\in\mathbb{Z}_+}$ with $z_j \in \mathbb{R}^l$ let $\|\{z_j\}_{j\in\mathbb{Z}_+}\| := \sup\{\|z_j\| \mid j \in \mathbb{Z}_+\}$. For a sequence $\{z_j\}_{j\in\mathbb{Z}_+}$ with $z_j \in \mathbb{R}^l$, $z_{[k]}$ denotes the truncation of $\{z_j\}_{j\in\mathbb{Z}_+}$ at time $k \in \mathbb{Z}_+$, i.e. $z_{[k]} = \{z_j\}_{j\in\mathbb{Z}_{[0,k]}}$. For a set $\mathcal{P} \subseteq \mathbb{R}^n$, we denote by $\partial\mathcal{P}$ the boundary of $\mathcal{P}$, by $\mathrm{int}(\mathcal{P})$ its interior and by $\mathrm{cl}(\mathcal{P})$ its closure. Let $\mathcal{P}_1 \sim \mathcal{P}_2 \triangleq \{x \in \mathbb{R}^n \mid x + \mathcal{P}_2 \subseteq \mathcal{P}_1\}$ denote the Pontryagin difference of two arbitrary sets $\mathcal{P}_1$ and $\mathcal{P}_2$. A polyhedron is a convex set obtained as the intersection of a finite number of open and/or closed half-spaces.

Consider now the following discrete-time autonomous nonlinear systems:

$$x_{k+1} = G(x_k), \quad k \in \mathbb{Z}_+, \tag{1a}$$

$$\tilde{x}_{k+1} = \widetilde{G}(\tilde{x}_k, w_k), \quad k \in \mathbb{Z}_+, \tag{1b}$$

where $x_k, \tilde{x}_k \in \mathbb{R}^n$ are the state, $w_k \in \mathbb{R}^l$ is an unknown disturbance input and, $G : \mathbb{R}^n \to \mathbb{R}^n$, $\widetilde{G} : \mathbb{R}^n \times \mathbb{R}^l \to \mathbb{R}^n$ are nonlinear, possibly discontinuous, functions. For simplicity of notation, we assume that the origin is an equilibrium in (1), meaning that $G(0) = 0$ and $\widetilde{G}(0,0) = 0$. Due to space limitations, we refer to [20] for definitions regarding Lyapunov stability, attractivity, asymptotic stability in the Lyapunov sense and exponential stability of the origin for the nominal system (1a).

Definition 1. *A real-valued scalar function* $\varphi : \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathcal{K}$ *if it is continuous, strictly increasing and* $\varphi(0) = 0$*. A function* $\beta : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ belongs to class $\mathcal{KL}$ *if for each fixed* $k \in \mathbb{R}_+$*,* $\beta(\cdot, k) \in \mathcal{K}$ *and for each fixed* $s \in \mathbb{R}_+$*,* $\beta(s, \cdot)$ *is non-increasing and* $\lim_{k\to\infty} \beta(s,k) = 0$*.*

Definition 2. (ISS) *Let* $\mathbb{X}$ *with* $0 \in \text{int}(\mathbb{X})$ *and* $\mathbb{W}$ *be subsets of* $\mathbb{R}^n$ *and* $\mathbb{R}^l$*, respectively. The perturbed system* (1b) *is called* ISS for initial conditions in $\mathbb{X}$ and disturbance inputs in $\mathbb{W}$ *if there exist a* $\mathcal{KL}$*-function* β *and a* $\mathcal{K}$*-function* γ *such that, for each* $x_0 \in \mathbb{X}$ *and all* $\{w_p\}_{p\in\mathbb{Z}_+}$ *with* $w_p \in \mathbb{W}$ *for all* $p \in \mathbb{Z}_+$*, it holds that the state trajectory satisfies* $\|x_k\| \leq \beta(\|x_0\|, k) + \gamma(\|w_{[k-1]}\|)$ *for all* $k \in \mathbb{Z}_{\geq 1}$*.*

Note that the regional ISS property introduced in Definition 2 can be regarded as a local version of the global ISS property defined in [8] and it is similar to the robust asymptotic stability property employed in [11].

3 The MPC Optimization Problem

Consider the following nominal and perturbed discrete-time nonlinear systems:

$$x_{k+1} = g(x_k, u_k), \quad k \in \mathbb{Z}_+, \tag{2a}$$

$$\tilde{x}_{k+1} = \tilde{g}(\tilde{x}_k, u_k, w_k), \quad k \in \mathbb{Z}_+, \tag{2b}$$

where $x_k, \tilde{x}_k \in \mathbb{R}^n$ and $u_k \in \mathbb{R}^m$ are the state and the control input, respectively, and $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, $\tilde{g} : \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^n$ are nonlinear, possibly discontinuous, functions with $g(0,0) = 0$ and $\tilde{g}(0,0,0) = 0$. In the sequel we will consider the case when MPC is used to generate the control input in (2). We assume that the state and the input vectors are constrained for both systems (2a) and (2b), in a compact subset $\mathbb{X}$ of $\mathbb{R}^n$ and a compact subset $\mathbb{U}$ of $\mathbb{R}^m$, respectively, which contain the origin in their interior. For a fixed $N \in \mathbb{Z}_{\geq 1}$, let $\mathbf{x}_k(x_k, \mathbf{u}_k) \triangleq (x_{1|k}, \ldots, x_{N|k})$ denote the state sequence generated by the nominal system (2a) from initial state $x_{0|k} \triangleq x_k$ and by applying the input sequence $\mathbf{u}_k \triangleq (u_{0|k}, \ldots, u_{N-1|k}) \in \mathbb{U}^N$, where $\mathbb{U}^N \triangleq \mathbb{U} \times \ldots \times \mathbb{U}$. Furthermore, let $\mathbb{X}_T \subseteq \mathbb{X}$ denote a desired target set that contains the origin. The class of *admissible input sequences* defined with respect to $\mathbb{X}_T$ and state $x_k \in \mathbb{X}$ is $\mathcal{U}_N(x_k) \triangleq \{\mathbf{u}_k \in \mathbb{U}^N \mid \mathbf{x}_k(x_k, \mathbf{u}_k) \in \mathbb{X}^N,\ x_{N|k} \in \mathbb{X}_T\}$.

Problem 1. Let the target set $\mathbb{X}_T \subseteq \mathbb{X}$ and $N \geq 1$ be given and let $F : \mathbb{R}^n \to \mathbb{R}_+$ with $F(0) = 0$ and $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}_+$ with $L(0,0) = 0$ be mappings, possibly discontinuous. At time $k \in \mathbb{Z}_+$ let $x_k \in \mathbb{X}$ be given and minimize the cost function $J(x_k, \mathbf{u}_k) \triangleq F(x_{N|k}) + \sum_{i=0}^{N-1} L(x_{i|k}, u_{i|k})$, with prediction model (2a), over all input sequences $\mathbf{u}_k \in \mathcal{U}_N(x_k)$.

In the MPC literature, $F(\cdot)$, $L(\cdot,\cdot)$ and N are called the terminal cost, the stage cost and the prediction horizon, respectively. We call an initial state $x \in \mathbb{X}$ *feasible* if $\mathcal{U}_N(x) \neq \emptyset$. Similarly, Problem 1 is said to be *feasible* for $x \in \mathbb{X}$ if $\mathcal{U}_N(x) \neq \emptyset$. Let $\mathbb{X}_f(N) \subseteq \mathbb{X}$ denote the set of *feasible initial states* with respect to Problem 1 and let

$$V_{\mathrm{MPC}} : \mathbb{X}_f(N) \to \mathbb{R}_+, \quad V_{\mathrm{MPC}}(x_k) \triangleq \inf_{\mathbf{u}_k \in \mathcal{U}_N(x_k)} J(x_k, \mathbf{u}_k) \tag{3}$$

denote the MPC value function corresponding to Problem 1. We assume that there exists an optimal sequence of controls $\mathbf{u}_k^* \triangleq (u_{0|k}^*, u_{1|k}^*, \ldots, u_{N-1|k}^*)$ for Problem 1 and any state $x_k \in \mathbb{X}_f(N)$. Hence, the infimum in (3) is a minimum and $V_{\mathrm{MPC}}(x_k) = J(x_k, \mathbf{u}_k^*)$. Then, *the MPC control law* is defined as

$$u^{\mathrm{MPC}}(x_k) \triangleq u_{0|k}^*; \quad k \in \mathbb{Z}_+. \tag{4}$$

The following stability analysis also holds when the optimum is not unique in Problem 1, i.e. all results apply irrespective of which optimal sequence is selected.

4 General Results on Stability and ISS

Let $h : \mathbb{R}^n \to \mathbb{R}^m$ denote an arbitrary, possibly discontinuous, nonlinear function with $h(0) = 0$ and let $\mathbb{X}_{\mathbb{U}} \triangleq \{x \in \mathbb{X} \mid h(x) \in \mathbb{U}\}$.

The following theorem was obtained as a kind of general and unifying result by putting together the previous results on stability of discrete-time nonlinear MPC that were mentioned in the introductory survey.

Assumption 1. *Terminal cost and constraint set*: There exist $\alpha_1, \alpha_2 \in \mathcal{K}$, a neighborhood of the origin $\mathcal{N} \subseteq \mathbb{X}_f(N)$ and a feedback control law $h(\cdot)$ such that $\mathbb{X}_T \subseteq \mathbb{X}_{\mathbb{U}}$, with $0 \in \mathrm{int}(\mathbb{X}_T)$, is a positively invariant set [20] for system (2a) in closed-loop with $u = h(x)$, $L(x,u) \geq \alpha_1(\|x\|)$ for all $x \in \mathbb{X}_f(N)$ and all $u \in \mathbb{U}$, $F(x) \leq \alpha_2(\|x\|)$ for all $x \in \mathcal{N}$ and

$$F(g(x, h(x))) - F(x) + L(x, h(x)) \leq 0 \quad \text{for all} \quad x \in \mathbb{X}_T. \tag{5}$$

Assumption 2. *Terminal equality constraint*: $\mathbb{X}_T = \{0\}$, $F(x) = 0$ for all $x \in \mathbb{X}$ and there exist $\alpha_1, \alpha_2 \in \mathcal{K}$ and a neighborhood of the origin $\mathcal{N} \subseteq \mathbb{X}_f(N)$ such that $L(x,u) \geq \alpha_1(\|x\|)$ for all $x \in \mathbb{X}_f(N)$ and all $u \in \mathbb{U}$ and $L(x_{i|k}^*, u_{i|k}^*) \leq \alpha_2(\|x_k\|)$, for any optimal $\mathbf{u}_k^* \in \mathcal{U}_N(x_k)$, initial state $x_k =: x_{0|k}^* \in \mathcal{N}$ and $i = 0, \ldots, N-1$, where $(x_{1|k}^*, \ldots, x_{N|k}^*) =: \mathbf{x}_k(x_k, \mathbf{u}_k^*)$.

Theorem 1. (Stability of Non-smooth Nonlinear MPC) *Fix $N \geq 1$ and suppose that either Assumption 1 holds or Assumption 2 holds. Then:*

(i) *If Problem 1 is feasible at time $k \in \mathbb{Z}_+$ for state $x_k \in \mathbb{X}$, Problem 1 is feasible at time $k+1$ for state $x_{k+1} = g(x_k, u^{MPC}(x_k))$. Moreover, $\mathbb{X}_T \subseteq \mathbb{X}_f(N)$;*

(ii) *The origin of the MPC closed-loop system* (2a)-(4) *is asymptotically stable in the Lyapunov sense for initial conditions in $\mathbb{X}_f(N)$;*

(iii) *If Assumption 1 or Assumption 2 holds with $\alpha_1(s) \triangleq as^\lambda$, $\alpha_2(s) \triangleq bs^\lambda$ for some constants $a, b, \lambda > 0$, the origin of the MPC closed-loop system* (2a)-(4) *is exponentially stable in $\mathbb{X}_f(N)$.*

The interested reader can find the proof of Theorem 1 in [20]. Next, we state sufficient conditions for ISS (in the sense of Definition 2) of discrete-time non-smooth nonlinear MPC.

Theorem 2. (ISS of Non-smooth Nonlinear MPC) *Let $\mathbb{W}$ be a compact subset of $\mathrm{I\!R}^l$ that contains the origin and let $\mathbb{X}$ be a robustly positively invariant (RPI) set* [20] *for the MPC closed-loop system* (2b)-(4) *and disturbances in $\mathbb{W}$, with $0 \in \mathrm{int}(\mathbb{X})$. Let $\alpha_1(s) \triangleq as^\lambda$, $\alpha_2(s) \triangleq bs^\lambda$, $\alpha_3(s) \triangleq cs^\lambda$ for some positive constants a, b, c, λ and let $\sigma \in \mathcal{K}$. Suppose $L(x, u) \geq \alpha_1(\|x\|)$ for all $x \in \mathbb{X}$ and all $u \in \mathbb{U}$, $V_{MPC}(x) \leq \alpha_2(\|x\|)$ for all $x \in \mathbb{X}$ and that:*

$$V_{MPC}(\tilde{g}(x, u^{MPC}(x), w)) - V_{MPC}(x) \leq -\alpha_3(\|x\|) + \sigma(\|w\|), \quad \forall x \in \mathbb{X},\ \forall w \in \mathbb{W}. \tag{6}$$

Then, the perturbed system (2b) *in closed-loop with the MPC control* (4) *obtained by solving Problem 1 at each sampling-instant is ISS for initial conditions in $\mathbb{X}$ and disturbance inputs in $\mathbb{W}$. Moreover, the ISS property of Definition 2 holds for $\beta(s, k) \triangleq \alpha_1^{-1}(2\rho^k \alpha_2(s))$ and $\gamma(s) \triangleq \alpha_1^{-1}\left(\frac{2\sigma(s)}{1-\rho}\right)$, where $\rho \triangleq 1 - \frac{c}{b} \in [0, 1)$.*

For a proof of Theorem 2 we refer the reader to [20]. Note that the hypotheses of Theorem 1 and Theorem 2 allow $g(\cdot, \cdot)$, $\tilde{g}(\cdot, \cdot, \cdot)$ and $V_{\mathrm{MPC}}(\cdot)$ to be discontinuous when $x \neq 0$. They *only* imply continuity at the point $x = 0$, and *not* necessarily on a neighborhood of $x = 0$.

5 A Robust MPC Scheme for Discontinuous PWA Systems

In this section we consider the class of discrete-time piecewise affine systems, i.e.

$$x_{k+1} = g(x_k, u_k) \triangleq A_j x_k + B_j u_k + f_j \qquad \text{if} \quad x_k \in \Omega_j, \tag{7a}$$

$$\tilde{x}_{k+1} = \tilde{g}(\tilde{x}_k, u_k, w_k) \triangleq A_j \tilde{x}_k + B_j u_k + f_j + w_k \quad \text{if} \quad \tilde{x}_k \in \Omega_j, \tag{7b}$$

where $w_k \in \mathbb{W} \subset \mathrm{I\!R}^n$, $k \in \mathbb{Z}_+$, $A_j \in \mathrm{I\!R}^{n \times n}$, $B_j \in \mathrm{I\!R}^{n \times m}$, $f_j \in \mathrm{I\!R}^n$, $j \in \mathcal{S}$ with $\mathcal{S} \triangleq \{1, 2, \ldots, s\}$ a *finite set* of indices. The collection $\{\Omega_j \mid j \in \mathcal{S}\}$ defines a partition of $\mathbb{X}$, meaning that $\cup_{j \in \mathcal{S}} \Omega_j = \mathbb{X}$ and $\mathrm{int}(\Omega_i) \cap \mathrm{int}(\Omega_j) = \emptyset$ for $i \neq j$. Each Ω_j is assumed to be a polyhedron (not necessarily closed). Let

$\mathcal{S}_0 \triangleq \{j \in \mathcal{S} \mid 0 \in \mathrm{cl}(\Omega_j)\}$ and let $\mathcal{S}_1 \triangleq \{j \in \mathcal{S} \mid 0 \notin \mathrm{cl}(\Omega_j)\}$, so that $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1$. We assume that the origin is an equilibrium state for (7a) with $u = 0$. Therefore, we require that $f_j = 0$ for all $j \in \mathcal{S}_0$. Note that this does not exclude PWA systems which are *discontinuous over the boundaries*. Next, let $\|\cdot\|$ denote the ∞-norm and consider the case when the ∞-norm is used to define the MPC cost function, i.e. $F(x) \triangleq \|Px\|$ and $L(x, u) \triangleq \|Qx\| + \|Ru\|$. Here $P \in \mathbb{R}^{p\times n}$, $Q \in \mathbb{R}^{q\times n}$ and $R \in \mathbb{R}^{r\times m}$ are assumed to be known matrices that have full-column rank. In the PWA setting we take the auxiliary controller $h(x) \triangleq K_j x$ when $x \in \Omega_j$, where $K_j \in \mathbb{R}^{m\times n}$, $j \in \mathcal{S}$.

In [17] the authors developed ways to compute (off-line) the terminal weight P and the feedbacks $\{K_j \mid j \in \mathcal{S}\}$ such that inequality (5) holds and $\mathbb{X}_T$ is a positively invariant set for the PWA system (7a) in closed-loop with the piecewise linear (PWL) state-feedback $h(\cdot)$. Then, it can be shown that PWA systems in closed-loop with MPC controllers calculated as in (4) and using an ∞-norm based cost in Problem 1 satisfy the hypothesis of Theorem 1, thereby establishing Lyapunov stability for the origin of the closed-loop system. A similar result for quadratic cost based MPC and PWA prediction models can be found in [20]. However, since both the system (7) and the hybrid MPC value function will be discontinuous in general, it follows, as pointed out in [12], that the closed-loop system may not be robust (ISS) to *arbitrarily small* disturbances, despite the fact that nominal asymptotic stability is guaranteed.

In this section we present a new design method based on tightened constraints for setting up ISS MPC schemes for a class of *discontinuous* PWA systems. One of the advantages of the proposed approach is that the resulting MPC optimization problem can still be formulated as a *mixed integer linear programming* (MILP) problem, which is a standard problem in hybrid MPC. Note that in this case the assumption of Section 3 on the existence of an optimal sequence of controls is satisfied, see, for example, [14, 20].

Let $\eta \triangleq \max_{j\in\mathcal{S}} \|A_j\|$, $\xi \triangleq \|P\|$ and define, for any $\mu > 0$ and $i \in \mathbb{Z}_{\geq 1}$,

$$\mathcal{L}_\mu^i \triangleq \left\{ x \in \mathbb{R}^n \mid \|x\| \leq \mu \sum_{p=0}^{i-1} \eta^p \right\}.$$

Consider now the following (tightened) set of admissible input sequences:

$$\widetilde{\mathcal{U}}_N(x_k) \triangleq \left\{ \mathbf{u}_k \in \mathbb{U}^N \mid x_{i|k} \in \mathbb{X}_i,\ i = 1, \ldots, N-1,\ x_{N|k} \in \mathbb{X}_T \right\},\ k \in \mathbb{Z}_+, \quad (8)$$

where $\mathbb{X}_i \triangleq \cup_{j\in\mathcal{S}}\{\Omega_j \sim \mathcal{L}_\mu^i\} \subseteq \mathbb{X}$ for all $i = 1, \ldots, N-1$ and $(x_{1|k}, \ldots, x_{N|k})$ is the state sequence generated from initial state $x_{0|k} \triangleq x_k$ and by applying the input sequence $\mathbf{u}_k$ to the PWA model (7a). Let $\widetilde{\mathbb{X}}_f(N)$ denote the set of feasible states for Problem 1 with $\widetilde{\mathcal{U}}_N(x_k)$ instead of $\mathcal{U}_N(x_k)$, and let $\widetilde{V}_{\mathrm{MPC}}(\cdot)$ denote the corresponding MPC value function. For any $\mu > 0$, define $\mathcal{B}_\mu \triangleq \{w \in \mathbb{R}^n \mid \|w\| \leq \mu\}$ and recall that $\mathbb{X}_{\mathbb{U}} = \{x \in \mathbb{X} \mid h(x) \in \mathbb{U}\}$.

Theorem 3. *Assume that* $0 \in \mathrm{int}(\Omega_{j^*})$ *for some* $j^* \in \mathcal{S}$. *Take* $N \in \mathbb{Z}_{\geq 1}$, $\theta > \theta_1 > 0$ *and* $\mu > 0$ *such that* $\mu \leq \frac{\theta - \theta_1}{\xi \eta^{N-1}}$,

$$\mathbb{F}_\theta \triangleq \{x \in \mathbb{R}^n \mid F(x) \leq \theta\} \subseteq (\Omega_{j^*} \sim \mathcal{L}_\mu^{N-1}) \cap \mathbb{X}_{\mathbb{U}}$$

and $g(x, h(x)) \in \mathbb{F}_{\theta_1}$ *for all* $x \in \mathbb{F}_\theta$*. Set* $\mathbb{X}_T = \mathbb{F}_{\theta_1}$*. Furthermore, suppose that Assumption 1 holds and inequality* (5) *is satisfied for all* $x \in \mathbb{F}_\theta$*. Then:*

(i) *If* $\tilde{x}_k \in \widetilde{\mathbb{X}}_f(N)$*, then* $\tilde{x}_{k+1} \in \widetilde{\mathbb{X}}_f(N)$ *for all* $w_k \in \mathcal{B}_\mu$*, where* $\tilde{x}_{k+1} = A_j\tilde{x}_k + B_j u^{MPC}(\tilde{x}_k) + f_j + w_k$*. Moreover,* $\mathbb{X}_T \subseteq \widetilde{\mathbb{X}}_f(N)$*.*

(ii) *The perturbed PWA system* (7b) *in closed-loop with the MPC control* (4) *obtained by solving Problem 1 (with* $\widetilde{\mathcal{U}}_N(x_k)$ *instead of* $\mathcal{U}_N(x_k)$ *and* (7a) *as prediction model) at each sampling instant is ISS for initial conditions in* $\widetilde{\mathbb{X}}_f(N)$ *and disturbance inputs in* $\mathcal{B}_\mu$*.*

The proof of Theorem 3 is given in the appendix. The tightened set of admissible input sequences (8) may become very conservative as the prediction horizon increases, since it requires that the state trajectory must be kept farther and farther away from the boundaries. The conservativeness can be reduced by introducing a pre-compensating state-feedback, which is a common solution in robust MPC.

6 Illustrative Example

To illustrate the application of Theorem 3 and how to construct the parameters θ, θ_1 and μ for a given $N \in \mathbb{Z}_{\geq 1}$, we present an example. Consider the following *discontinuous* PWA system:

$$x_{k+1} = \tilde{g}(x_k, u_k, w_k) \triangleq g(x_k, u_k) + w_k \triangleq A_j x_k + B_j u_k + w_k \text{ if } x_k \in \Omega_j,\ j \in \mathcal{S}, \tag{9}$$

where $\mathcal{S} = \{1, \ldots, 5\}$, $A_1 = \left[\begin{smallmatrix} -0.0400 & -0.4610 \\ -0.1390 & 0.3410 \end{smallmatrix}\right]$, $A_2 = \left[\begin{smallmatrix} 0.6552 & 0.2261 \\ 0.5516 & -0.0343 \end{smallmatrix}\right]$, $A_3 = \left[\begin{smallmatrix} -0.7713 & 0.7335 \\ 0.4419 & 0.5580 \end{smallmatrix}\right]$, $A_4 = \left[\begin{smallmatrix} -0.0176 & 0.5152 \\ 0.6064 & 0.2168 \end{smallmatrix}\right]$, $A_5 = \left[\begin{smallmatrix} -0.0400 & -0.4610 \\ -0.0990 & 0.6910 \end{smallmatrix}\right]$, $B_1 = B_2 = B_3 = B_4 = \left[\begin{smallmatrix} 1 & 0 \end{smallmatrix}\right]^\top$ and $B_5 = \left[\begin{smallmatrix} 0 & 1 \end{smallmatrix}\right]^\top$. The state and the input of system (9) are constrained at all times in the sets $\mathbb{X} = [-3, 3] \times [-3, 3]$ and $\mathbb{U} = [-0.2, 0.2]$, respectively. The state-space partition is plotted in Figure 1. The method presented in [17] was employed to compute the terminal weight matrix $P = \left[\begin{smallmatrix} 2.3200 & 0.3500 \\ -0.2100 & 2.4400 \end{smallmatrix}\right]$ and the feedback $K = \left[\begin{smallmatrix} -0.04 & -0.35 \end{smallmatrix}\right]$ such that inequality (5) of Assumption 1 holds for all $x \in \mathbb{R}^2$, the ∞-norm MPC cost with $Q = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$, $R = 0.01$ and $h(x) = Kx$. Based on inequality (5), it can be shown that the sublevel sets of the terminal cost $F(\cdot)$, i.e. also $\mathbb{F}_\theta$, are λ-contractive sets [20] for the dynamics $g(x, h(x))$, with $\lambda = 0.6292$. Then, for any θ_1 with $\theta > \theta_1 \geq \lambda\theta$ it holds that $g(x, h(x)) \in \mathbb{F}_{\theta_1}$ for all $x \in \mathbb{F}_\theta$. This yields $\mu \leq \frac{(1-\lambda)\theta}{\xi\eta^{N-1}}$. However, μ and θ must also be such that $\mathbb{F}_\theta \subseteq (\Omega_5 \sim \mathcal{L}_\mu^{N-1}) \cap \mathbb{X}_{\mathbb{U}}$. Hence, a trade-off must be made in choosing θ and μ. A large θ implies a large μ, which is desirable since μ is an upper bound on $\|w\|$, but θ must also be small enough to ensure the above inclusion. We chose $\theta = 0.96$ and $\theta_1 = \lambda\theta = 0.6040$. Then, with $\eta = 1.5048$, $\xi = 2.67$ and a prediction horizon $N = 2$ one obtains that any μ with $0 \leq \mu \leq 0.0886$ is an admissible upper bound on $\|w\|$. For $\mu = 0.0886$ it holds that $\mathbb{F}_\theta \subseteq (\Omega_5 \sim \mathcal{L}_\mu^1) \cap \mathbb{X}_{\mathbb{U}}$ (see Figure 2 for an illustrative plot). Hence, the hypothesis of Theorem 3 is satisfied for any $w \in \mathcal{B}_\mu = \{w \in \mathbb{R}^2 \mid \|w\| \leq 0.0886\}$.

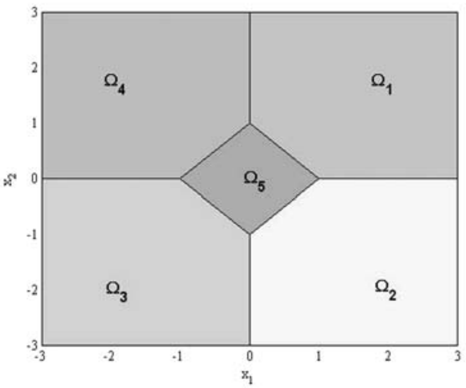


Fig. 1. State-space partition for system (9)

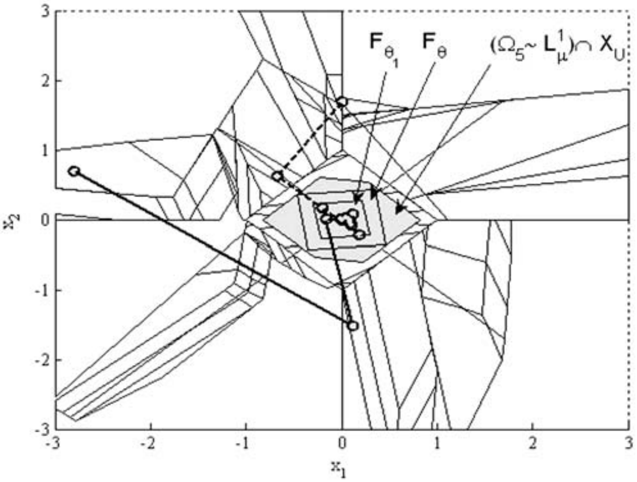


Fig. 2. State trajectories for the MPC closed-loop system (9)-(4) with $x_0 = [0.003\ 1.7]^\top$ - dashed line and $x_0 = [-2.8\ 0.7]^\top$ - solid line

Then, we used the multi parametric toolbox (MPT) [21] to calculate the MPC control law (4) as an explicit PWA state-feedback, and to simulate the resulting MPC closed-loop system (9)-(4) for randomly generated disturbances in $\mathcal{B}_\mu$. The explicit MPC controller is defined over 132 state-space regions. The set of feasible states $\widetilde{\mathbb{X}}_f(2)$ is plotted in Figure 2 together with the partition corresponding to the explicit MPC control law.

Note that, by Theorem 3, ISS is ensured for the closed-loop system for initial conditions in $\widetilde{\mathbb{X}}_f(2)$ and disturbances in $\mathcal{B}_\mu$, without employing a *continuous MPC value function*. Indeed, for example, $\widetilde{V}_{\text{MPC}}(\cdot)$ and the closed-loop PWA dynamics (9)-(4) are discontinuous at $x = [0\ \ 1]^\top \in \text{int}(\widetilde{\mathbb{X}}_f(2))$.

7 Conclusion

In this paper we have presented an overview of stability and robustness theory for discrete-time nonlinear MPC while focusing on the application and the extension of the classical results to *discontinuous* nonlinear systems. A stability theorem has been developed, which unifies many of the previous results. An ISS result for discrete-time discontinuous nonlinear MPC has also been presented. A new MPC scheme with an ISS guarantee has been developed for a particular class of discontinuous PWA systems.

Acknowledgements

The authors are grateful to the reviewers for their helpful comments. This research was supported by the Dutch Science Foundation (STW), Grant "Model Predictive Control for Hybrid Systems" (DMR. 5675) and the European Community through the Network of Excellence HYCON (contract number FP6-IST-511368).

References

[1] Kalman, R.E., Bertram, J.E.: Control system analysis and design via the second method of Lyapunov, II: Discrete-time systems. Transactions of the ASME, Journal of Basic Engineering **82** (1960) 394–400

[2] Keerthi, S.S., Gilbert, E.G.: Optimal, infinite horizon feedback laws for a general class of constrained discrete time systems: Stability and moving-horizon approximations. Journal of Optimization Theory and Applications **57** (1988) 265–293

[3] Alamir, M., Bornard, G.: On the stability of receding horizon control of nonlinear discrete-time systems. Systems and Control Letters **23** (1994) 291–296

[4] Meadows, E.S., Henson, M.A., Eaton, J.W., Rawlings, J.B.: Receding horizon control and discontinuous state feedback stabilization. International Journal of Control **62** (1995) 1217–1229

[5] Scokaert, P.O.M., Rawlings, J.B., Meadows, E.B.: Discrete-time stability with perturbations: Application to model predictive control. Automatica **33** (1997) 463–470

[6] Scokaert, P.O.M., Mayne, D.Q., Rawlings, J.B.: Suboptimal model predictive control (feasibility implies stability). IEEE Transactions on Automatic Control **44** (1999) 648–654

[7] Magni, L., De Nicolao, G., Scattolini, R.: Output feedback receding-horizon control of discrete-time nonlinear systems. In: 4th IFAC NOLCOS. Volume 2., Oxford, UK (1998) 422–427

[8] Jiang, Z.P., Wang, Y.: Input-to-state stability for discrete-time nonlinear systems. Automatica **37** (2001) 857–869

[9] Limon, D., Alamo, T., Camacho, E.F.: Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In: 41st IEEE Conference on Decision and Control, Las Vegas, Nevada (2002) 4619–4624

[10] Magni, L., Raimondo, D.M., Scattolini, R.: Regional input-to-state stability for nonlinear model predictive control. IEEE Transactions on Automatic Control **51** (2006) 1548–1553

[11] Grimm, G., Messina, M.J., Tuna, S.E., Teel, A.R.: Nominally robust model predictive control with state contraints. In: 42nd IEEE Conference on Decision and Control, Maui, Hawaii (2003) 1413–1418

[12] Grimm, G., Messina, M.J., Tuna, S.E., Teel, A.R.: Examples when nonlinear model predictive control is nonrobust. Automatica **40** (2004) 1729–1738

[13] Bemporad, A., Morari, M.: Control of systems integrating logic, dynamics, and constraints. Automatica **35** (1999) 407–427

[14] Borrelli, F.: Constrained optimal control of linear and hybrid systems. Volume 290 of Lecture Notes in Control and Information Sciences. Springer (2003)

[15] Kerrigan, E.C., Mayne, D.Q.: Optimal control of constrained, piecewise affine systems with bounded disturbances. In: 41st IEEE Conference on Decision and Control, Las Vegas, Nevada (2002) 1552–1557

[16] Mayne, D.Q., Rakovic, S.V.: Model predictive control of constrained piecewise affine discrete-time systems. International Journal of Robust and Nonlinear Control **13** (2003) 261–279

[17] Lazar, M., Heemels, W.P.M.H., Weiland, S., Bemporad, A., Pastravanu, O.: Infinity norms as Lyapunov functions for model predictive control of constrained PWA systems. In: Hybrid Systems: Computation and Control. Volume 3414 of Lecture Notes in Computer Science., Zürich, Switzerland, Springer Verlag (2005) 417–432

[18] Grieder, P., Kvasnica, M., Baotic, M., Morari, M.: Stabilizing low complexity feedback control of constrained piecewise affine systems. Automatica **41** (2005) 1683–1694

[19] Rakovic, S.V., Mayne, D.Q.: Robust model predictive control of contrained piecewise affine discrete time systems. In: 6th IFAC NOLCOS, Stuttgart, Germany (2004)

[20] Lazar, M.: Model predictive control of hybrid systems: Stability and robustness. PhD thesis, Eindhoven University of Technology, The Netherlands (2006)

[21] Kvasnica, M., Grieder, P., Baotic, M., Morari, M.: Multi Parametric Toolbox (MPT). In: Hybrid Systems: Computation and Control. Lecture Notes in Computer Science, Volume 2993, Pennsylvania, Philadelphia, USA, Springer Verlag (2004) 448–462 Toolbox available for download at http://control.ee.ethz.ch/∼mpt.

A Proof of Theorem 3

Let $(x^*_{1|k}, \ldots, x^*_{N|k})$ denote the state sequence obtained from initial state $x_{0|k} \triangleq \tilde{x}_k$ and by applying the input sequence $\mathbf{u}^*_k$ to (7a). Let $(x_{1|k+1}, \ldots, x_{N|k+1})$ denote the state sequence obtained from the initial state $x_{0|k+1} \triangleq \tilde{x}_{k+1} = x_{k+1} + w_k = x^*_{1|k} + w_k$ and by applying the input sequence $\mathbf{u}_{k+1} \triangleq (u^*_{1|k}, \ldots, u^*_{N-1|k}, h(x_{N-1|k+1}))$ to (7a).

(i) The constraints in (8) are such that: (P1) $(x_{i|k+1}, x^*_{i+1|k}) \in \Omega_{j_{i+1}} \times \Omega_{j_{i+1}}$, $j_{i+1} \in \mathcal{S}$, for all $i = 0, \ldots, N-2$ and, $\|x_{i|k+1} - x^*_{i+1|k}\| \leq \eta^i \mu$ for $i = 0, \ldots, N-1$. This is due to the fact that $x_{0|k+1} = x^*_{1|k} + w_k$, $x_{i|k+1} = x^*_{i+1|k} + \prod_{p=1}^{i} A_{j_p} w_k$ for $i = 1, \ldots, N-1$ and $\|\prod_{p=1}^{i} A_{j_p} w_k\| \leq \eta^i \mu$, which yields $\prod_{p=1}^{i} A_{j_p} w_k \in \mathcal{L}^{i+1}_{\mu}$. Pick the indices $j_{i+1} \in \mathcal{S}$ such that $x^*_{i+1|k} \in \Omega_{j_{i+1}}$ for all $i = 1, \ldots, N-2$. Then, due to $x^*_{i+1|k} \in \Omega_{j_{i+1}} \sim \mathcal{L}^{i+1}_\mu$, it follows by Lemma 2 of [9] that $x_{i|k+1} \in \Omega_{j_{i+1}} \sim \mathcal{L}^i_\mu \subset \mathbb{X}_i$ for $i = 1, \ldots, N-2$. From $x_{N-1|k+1} = x^*_{N|k} + \prod_{p=1}^{N-1} A_{j_p} w_k$ it follows that $F(x_{N-1|k+1}) - F(x^*_{N|k}) \leq \xi\eta^{N-1}\mu$, which implies that $F(x_{N-1|k+1}) \leq \theta_1 + \xi\eta^{N-1}\mu \leq \theta$ due to $x^*_{N|k} \in \mathbb{X}_T = \mathbb{F}_{\theta_1}$ and $\mu \leq \frac{\theta-\theta_1}{\xi\eta^{N-1}}$. Hence, $x_{N-1|k+1} \in \mathbb{F}_\theta \subset \mathbb{X}_\mathbb{U} \cap (\Omega_{j^*} \sim \mathcal{L}^{N-1}_\mu) \subset \mathbb{X}_\mathbb{U} \cap \mathbb{X}_{N-1}$ so that $h(x_{N-1|k+1}) \in \mathbb{U}$ and $x_{N|k+1} \in \mathbb{F}_{\theta_1} = \mathbb{X}_T$. Thus, the sequence of inputs $\mathbf{u}_{k+1}$ is feasible at time $k+1$ and Problem 1 with $\widetilde{\mathcal{U}}_N(x_k)$ instead of $\mathcal{U}_N(x_k)$ remains feasible. Moreover, from $g(x, h(x)) \in \mathbb{F}_{\theta_1}$ for all $x \in \mathbb{F}_\theta$ and $\mathbb{F}_{\theta_1} \subset \mathbb{F}_\theta$ it follows that $\mathbb{F}_{\theta_1}$ is a positively invariant set for system (7a) in closed-loop with $u_k = h(x_k)$, $k \in \mathbb{Z}_+$. Then, since

$$\mathbb{F}_{\theta_1} \subset \mathbb{F}_\theta \subseteq (\Omega_{j^*} \sim \mathcal{L}^{N-1}_\mu) \cap \mathbb{X}_\mathbb{U} \subset \mathbb{X}_i \cap \mathbb{X}_\mathbb{U} \quad \text{for all} \quad i = 1, \ldots, N-1$$

and $\mathbb{X}_T = \mathbb{F}_{\theta_1}$, the sequence of control inputs $(h(x_{0|k}), \ldots, h(x_{N-1|k}))$ is feasible with respect to Problem 1 (with $\widetilde{\mathcal{U}}_N(x_k)$ instead of $\mathcal{U}_N(x_k)$) for all $x_{0|k} \triangleq \tilde{x}_k \in \mathbb{F}_{\theta_1}$. Therefore, $\mathbb{X}_T = \mathbb{F}_{\theta_1} \subseteq \widetilde{\mathbb{X}}_f(N)$.

(ii) The result of part (i) implies that $\widetilde{\mathbb{X}}_f(N)$ is a RPI set for system (7b) in closed-loop with the MPC control (4) and disturbances in $\mathcal{B}_\mu$. Moreover, since $0 \in \text{int}(\mathbb{X}_T)$, we have that $0 \in \text{int}(\widetilde{\mathbb{X}}_f(N))$. The choice of the terminal cost and of the stage cost ensures that there exist $a, b > 0$, $\alpha_1(s) \triangleq as$ and $\alpha_2(s) \triangleq bs$ such that $\alpha_1(\|x\|) \leq \widetilde{V}_{\text{MPC}}(x) \leq \alpha_2(\|x\|)$ for all $x \in \widetilde{\mathbb{X}}_f(N)$. Let $\tilde{x}_{k+1}$ denote the solution of (7b) in closed-loop with $u^{\text{MPC}}(\cdot)$ obtained as indicated in part (i) of the proof and let $x^*_{0|k} \triangleq \tilde{x}_k$. Due to full-column rank of Q there exists $\gamma > 0$ such that $\|Qx\| \geq \gamma\|x\|$ for all x. Then, by optimality, property (P1), $x_{N-1|k+1} \in \mathbb{F}_\theta$ and from inequality (5) it follows that:

$$\begin{aligned}
&\widetilde{V}(\tilde{x}_{k+1}) - \widetilde{V}(\tilde{x}_k) \leq J(\tilde{x}_{k+1}, \mathbf{u}_{k+1}) - J(\tilde{x}_k, \mathbf{u}^*_k) = -L(x^*_{0|k}, u^*_{0|k}) + F(x_{N|k+1}) \\
&\quad + [-F(x_{N-1|k+1}) + F(x_{N-1|k+1})] - F(x^*_{N|k}) + L(x_{N-1|k+1}, h(x_{N-1|k+1})) \\
&\quad + \sum_{i=0}^{N-2} \Big[L(x_{i|k+1}, \mathbf{u}_{k+1}(i+1)) - L(x^*_{i+1|k}, u^*_{i+1|k}) \Big] \\
&\quad \leq -L(x^*_{0|k}, u^*_{0|k}) + F(x_{N|k+1}) - F(x_{N-1|k+1}) + L(x_{N-1|k+1}, h(x_{N-1|k+1})) \\
&\quad + \left(\xi\eta^{N-1} + \|Q\| \sum_{p=0}^{N-2} \eta^p \right) \|w_k\| \\
&\quad \overset{(5)}{\leq} -\|Qx^*_{0|k}\| + \sigma(\|w_k\|) \leq -\alpha_3(\|\tilde{x}_k\|) + \sigma(\|w_k\|),
\end{aligned}$$

with $\sigma(s) \triangleq (\xi\eta^{N-1} + \|Q\| \sum_{p=0}^{N-2} \eta^p)s$ and $\alpha_3(s) \triangleq \gamma s$. Thus, it follows that $\widetilde{V}_{\text{MPC}}(\cdot)$ satisfies the hypothesis of Theorem 3. Hence, the closed-loop system (7b)-(4) is ISS for initial conditions in $\widetilde{\mathbb{X}}_f(N)$ and disturbance inputs in $\mathcal{B}_\mu$. □

Model Predictive Control for Nonlinear Sampled-Data Systems

L. Grüne[1], D. Nešić[2], and J. Pannek[3]

[1] Mathematical Institute, University of Bayreuth
lars.gruene@uni-bayreuth.de
[2] EEE Department, University of Melbourne, Australia
d.nesic@ee.mu.oz.au
[3] Mathematical Institute, University of Bayreuth
juergen.pannek@uni-bayreuth.de

Summary. The topic of this paper is a new model predictive control (MPC) approach for the sampled–data implementation of continuous–time stabilizing feedback laws. The given continuous–time feedback controller is used to generate a reference trajectory which we track numerically using a sampled-data controller via an MPC strategy. Here our goal is to minimize the mismatch between the reference solution and the trajectory under control. We summarize the necessary theoretical results, discuss several aspects of the numerical implemenation and illustrate the algorithm by an example.

1 Introduction

Instead of designing a static state feedback with sampling and zero order hold by designing a continuous–time controller which is stabilizing an equilibrium and discretizing this controller ignoring sampling errors which leads to drawbacks in stability, see [5, 8], our approach is to use a continuous–time feedback and to anticipate and minimize the sampling errors by model predictive control (MPC) with the goal of allowing for large sampling periods without loosing performance and stability of the sampled–data closed loop. Therefore we consider two systems, the first to be controlled by the given continuous–time feedback which will give us a reference trajectory, and a second one which we are going to control using piecewise constant functions to construct an optimal control problem by introducing a cost functional to measure and minimize the mismatch between both solutions within a time interval.

In order to calculate a feedback instead of a time dependent control function and to avoid the difficulties of solving a Hamilton-Jacobi-Bellman equation for an infinite horizon problem we reduce the infinite time interval to a finite one by introducing a positive semidefinite function as cost–to–go. To re–gain the infinite control sequence we make use of a receding horizon technique. For this approach we will show stability and (sub-)optimality of the solution under certain standard assumptions.

We will also show how to implement an algorithm to solve this process of iteratively generating and solving optimal control problems. The latter one is done

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 105–113, 2007.
springerlink.com

using a direct approach and full discretization that will give us one optimization problem per optimal control problem which can be solved using an SQP method.

Therefore in Section 2 the problem, the necessary assumptions and our control scheme will be presented. In Section 3 we review the theoretical background results about stability and inverse optimality from [14]. Having done this the numerical implementation will be presented and discussed in Section 4 and its performance will be demonstrated by solving an example in Section 5. Finally conclusions will be given in Section 6.

2 Problem Formulation

The set of real numbers is denoted as $\mathbb{R}$. A function $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is called class $\mathcal{G}$ if it is continuous, zero at zero and non-decreasing. It is of class $\mathcal{K}$ if it is continuous, zero at zero and strictly increasing. It is of class $\mathcal{K}_\infty$ if it is also unbounded. It is of class $\mathcal{L}$ if it is strictly positive and it is decreasing to zero as its argument tends to infinity. A function $\beta : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is of class $\mathcal{KL}$ if for every fixed $t \geq 0$ the function $\beta(\cdot, t)$ is of class $\mathcal{K}$ and for each fixed $s > 0$ the function $\beta(s, \cdot)$ is of class $\mathcal{L}$. Given vectors $\xi, x \in \mathbb{R}^n$ we often use the notation $(\xi, x) := (\xi^T, \ x^T)^T$ and denote the norm by $| \cdot |$.

We consider a nonlinear feedback controlled plant model

$$\dot{x}(t) = f(x(t), u(x(t))) \tag{1}$$

with vector field $f : \mathbb{R}^n \times \mathbb{U} \to \mathbb{R}^n$ and state $x(t) \in \mathbb{R}^n$, where $u : \mathbb{R}^n \to \mathbb{U} \subset \mathbb{R}^m$ denotes a known continuous–time static state feedback which (globally) asymptotically stabilizes the system. We want to implement the closed loop system using a digital computer with sampling and zero order hold at the sampling time instants $t_k = k \cdot T$, $k \in \mathbb{N}$, $T \in \mathbb{R}_{>0}$. Then for a feedback law $u_T(x)$ the sampled-data closed loop system becomes

$$\dot{x}(t) = f(x(t), u_T(x(t_k))), \quad t \in [t_k, t_{k+1}). \tag{2}$$

Our goal is now to design $u_T(x)$ such that the corresponding sampled–data solution of (2) reproduces the continuous–time solution $x(t)$ of (1) as close as possible. The solution of the system (1) at time t emanating from the initial state $x(0) = x_0$ will be denoted by $x(t, x_0)$. Also we will assume $f(x, u(x))$ to be locally Lipschitz in x, hence a unique solution of the continuous–time closed loop system to exist for any $x(0) = x_0$ in a given compact set $\mathit{\Gamma} \subset \mathbb{R}^n$ containing the origin.

Remark 1. *The simplest approach to this problem is the emulation design in which one simply sets $u_T(x) := u(x)$. This method can be used for this purpose but one can only prove practical stability of the sampled–data closed loop system if the sampling time T is sufficiently small, see [8].*

In order to determine the desired sampled–data feedback u_T we first search for a piecewise constant control function v whose corresponding solution approximates the solution of the continuous–time closed loop system. Therefore the mismatch between the solutions of

$$\dot{x}(t) = f(x(t), u(x(t))),\ x(t_0) = x_0 \tag{3}$$

$$\dot{\xi}(t) = f(\xi(t), v_{[0,\infty]}),\ \xi(t_0) = \xi_0 \tag{4}$$

can be measured. Here $\xi(t, \xi_0)$ denotes the solution of the system under control and $v_{[0,\infty]}$ is a piecewise constant function with discontinuities only at the sampling instants $t_k := k \cdot T$, $k \in \mathbb{N}$. In order to measure and minimize the difference between both trajectories a cost functional of the form

$$J(\xi(t), x(t), v_{[0,\infty)}) := \sum_{j=0}^{\infty} \int_0^T l(\xi(t) - x(t), v_j)dt \tag{5}$$

is needed where $l : \mathbb{R}^n \times \mathbb{U} \to \mathbb{R}_{\geq 0}$. This results in an optimal control problem with infinite horizon which involves solving a Hamilton-Jacobi-Bellman type equation. In the linear case solutions to different H_2 and H_∞ control designs are known but the nonlinear case is typically too hard to be solved.

In order to avoid this computational burden we consider a reduced problem in a first step by limiting the horizon to a finite length. This will give us a suboptimal MPC controller whose numerical computation is manageable. Since T is fixed due to the problem formulation the length of the horizon H can be given by $M \in \mathbb{N}$ via $H = M \cdot T$. Hence the cost functional can be written as

$$J_M(\xi(t), x(t), v_{[0,M-1]}) := \sum_{j=0}^{M-1} \int_0^T l(\xi(t) - x(t), v_j)dt \, + \, F(\xi(t_M), x(t_M)) \tag{6}$$

using the function F to measure the cost-to-go $\sum\limits_{j=M}^{\infty} \int\limits_0^T l(\xi(t) - x(t), v_j)dt$.

Remark 2. *It is not necessary for F to be a control-Lyapunov-function of* (3), (4) *to prove semiglobal practical stability of the closed loop system. Moreover terminal costs of the form $F(\xi(t_M), x(t_M))$ instead of $F(\xi(t_M) - x(t_M))$ are considered since the infinite horizon value function $V_\infty(\xi, x) := \inf\limits_{v_{[0,\infty)}} J(\xi, x, v_{[0,\infty)})$ does not have in general the form $V_\infty(\xi - x)$.*

Using this approach an optimal control problem with finite horizon has to be solved which will return a finite control sequence $\hat{u}_{[0,M-1]}$. In order to determine the sampled–data feedback law u_T an infinite sequence of optimal control problems can be generated and solved using a receding horizon approach. To this end in a second step only the first control

$$u = u_M(\xi, x) := \hat{u}_0(\xi, x) \tag{7}$$

is implemented and the horizon is shifted forward in time by T. Hence a new optimal control problem is given and the process can be iterated. According to this procedure the receding horizon control law $u_T = u_M$ is a static state feedback for the coupled system that is implemented in a sampled-data fashion. Then the overall closed loop system is given by

$$\dot{\xi}(t) = f(\xi(t), u_M(\xi(t_k), x(t_k))), \quad \xi(0) = \xi_0, \ \ t \in [t_k, t_{k+1}), \tag{8}$$
$$\dot{x}(t) = f(x(t), u(x(t))), \quad x(0) = x_0. \tag{9}$$

Remark 3. *We like to emphasize that it is not only our goal to obtain asymptotical stability of* (8)*,* (9) *which implies tracking since we have that*

$$|(\xi(t), x(t))| \leq \beta(|(\xi_0, x_0)|, t) \quad \forall t \geq 0, \tag{10}$$

but also that we achieve this in an appropriate sub–optimal manner.

3 Stability and Inverse Optimality

Since most of the time one can only work with approximated discrete-time models consistency with the exact discrete-time model as described in [11, 12] is needed. Under the consistency condition given by Definition 1 in [10] and suitable mild additional assumptions one can conclude that asymptotic stability of the approximate model carries over to semiglobal practical asymptotic stability for the exact model, see [11, 12] for a general framework and [3] for corresponding results for MPC algorithms. This justifies the use of numerical approximations, cf. also Remark 6, below. To conclude semiglobal asymptotical stability of the closed loop system using the proposed MPC controller we present the following theorem, which relies on Theorem 1 in [1].

Theorem 1 (Stability)
Suppose the following conditions hold:

1. l and F are continuous;
2. $\mathbb{U}$ is bounded;
3a. The continuous–time system (1) *is globally asymptotically stable;*
3b. There exists a constant $r_0 > 0$ and a function $\gamma \in \mathcal{K}_\infty$ with

$$l(y, u) \geq \max\left\{ \max_{|x| \leq 2|y|} |f(x, u)|, \gamma(|y|) \right\}, \quad \forall |y| \geq r_0;$$

3c. $f(\cdot, \cdot)$ and $u(\cdot)$ are locally Lipschitz in their arguments;
4. The value function is such that for some $\bar{\alpha} \in \mathcal{K}_\infty$ we have that $V_i(\xi, x) \leq \bar{\alpha}(|(\xi, x)|)$ for all $i \geq 0$ and all $(\xi, x) \in \mathbb{R}^{2n}$.

Then there exists a function $\beta \in \mathcal{KL}$ such that for each pair of strictly positive real numbers (Δ, δ) there exists a constant $M_1^* \in \mathbb{Z}_{\geq 1}$ such that for all $(\xi, x) \in B_\Delta$ and $M \geq M_1^*$ the solutions of the continuous–time system (8)*,* (9) *satisfy*

$$|(\xi(t), x(t))| \leq \max\{\beta(|(\xi_0, x_0)|, t), \delta\} \quad \forall t \geq 0. \tag{11}$$

Proof. Make use of the underlying discrete-time system via Theorem 1 in [1] and Theorem 1 in [10], see [14] for details. ■

Therefore one can apply the calculated MPC control coming out of an approximated model in reality without loss of stability.

Remark 4. *If F is a control Lyapunov function for the exact discrete-time model of the uncontrolled sampled-data system*

$$\xi^+ = G(\xi, u) := \xi(T, \xi, u), \quad \xi(0) = \xi_0, \tag{12}$$

$$x^+ = H(x) := x(T, x), \quad x(0) = x_0 \tag{13}$$

then it follows from [1] that the theoretical bound M_1^* for the necessary prediction horizon decreases which was confirmed in our numerical simulations.

Remark 5. *Explicit bounds to guarantee the stability properties of the underlying discrete-time system can be found in [1].*

In order to show inverse optimality of our approach we suppose that F is such that there exists a closed set $X_f \subset \mathbb{R}^{2n}$ and a control law $u = u_f(\xi, x)$ with

1. $u_f(\xi, x) \in \mathbb{U} \;\; \forall(\xi, x) \in X_f$
2. If (ξ, x) is $\in X_f$ then also $(G(\xi, u_f(\xi, x)), H(X))$ is $\in X_f$.
3. For all $(\xi, x) \in X_f$ we have that

$$F(G(\xi, u_f(\xi, x)), H(x)) - F(\xi, x) \le -\int_0^T l(\xi(s, \xi, u_f) - x(s, x), u_f)ds.$$

Theorem 2 (Inverse (Sub-)Optimality)
Consider the discrete-time model (12), (13) *and suppose that the previous assumptions are valid. Then there exists a set $X_M \subset \mathbb{R}^{2n}$ and a function $\mathcal{Q} : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{U} \to \mathbb{R}$ with*

$$\mathcal{Q}(\xi, x, u_M) \ge Q(\xi, x, u_M) := \int_0^T l(\xi(s, \xi, u_f) - x(s, x), u_f)ds \tag{14}$$

$\forall(\xi, x) \in X_M$, $u \in \mathbb{U}$ *such that for all* $(\xi, x) \in X_M$ *we have that the controller* (7) *minimizes the cost functional*

$$\mathcal{J}(\xi, x, u_{[0,\infty)}) := \sum_{i=0}^{\infty} \mathcal{Q}(\xi_i, x_i, u_i). \tag{15}$$

Proof. The principle of optimality and the stated assumptions are utilised to show $\mathcal{Q}(\xi, x, u_M) \ge Q(\xi, x, u_M)$, see [14] for details. ■

4 Numerical Solution

For the solution of the optimal control problem we use a direct approach and therefore replace the problem to minimize (6) with dynamics (3), (4) by numerical approximations $\tilde{\xi}(t,\xi_0,u)$ of $\xi(t,\xi_0,u)$ and $\tilde{x}(t,x_0,u)$ of $x(t,x_0)$, respectively. For this approach convergence has been investigated in [7] and under suitable conditions one can guarantee that the order of convergence is $O(T)$.

From this formulation one obtains an optimization problem by introducing the variable $z = (\xi^0,\ldots,\xi^M,\, x^0,\ldots,x^M,\, u^0,\ldots,u^M)$ and rewriting the approximated optimal control problem as

$$\text{Minimize } F(z) := \sum_{j=0}^{M-1} \int_0^T l(\tilde{\xi}(s,\xi^j,v_j) - \tilde{x}(s,x^j),v_j)ds + F(\xi^M,x^M)$$

$$\text{s.t. } G(z) := \begin{pmatrix} [-\xi^{j+1} + \tilde{\xi}(h,\xi^j,v_j)]_{j=0,\ldots,M-1} \\ [-x^{j+1} + \tilde{x}(h,x^j)]_{j=0,\ldots,M-1} \\ (\xi^0,x^0) - (\xi_0,x_0) \end{pmatrix} = 0$$

with the constraints coming along with the approximation. This is a well known problem that can be solved using the KKT conditions by SQP methods if the cost functional and the constraints are sufficiently often differentiable in a sufficiently large neighborhood $N(z^*)$ of the local minima z^*. These methods are known to be stable and efficient even for large scale systems.

The used algorithm computes a sequence $(z^{[k]})$ via $z^{[k+1]} = z^{[k]} + \alpha^{[k]}p^{[k]}$. Within this iteration the search direction $p^{[k]}$ is calculated by generating and solving quadratic subproblems of the form

$$\begin{aligned} \min_{p\in\mathbb{R}^{N_z}} \quad & \nabla_z F(z^{[k]})p + \frac{1}{2}p^T B^{[k]} p \\ \text{s.t.} \quad & G(z^{[k]}) + \nabla_z G(z^{[k]})p = 0. \end{aligned}$$

The algorithm computes the derivatives by forward difference schemes if they are not given by the user and the matrix $B^{[k]}$ is an approximation of the Hesse matrix where a BFGS-Rank 2 update is implemented so that the Hesse matrix has to be calculated only once. Therefore the usual quadratic order of convergence of the Newton method is reduced but superlinear convergence can still be shown. The step size $\alpha^{[k]}$ is obtained by minimizing a merit function $\tilde{L}(z,\eta,\rho) = L(z,\eta) + \frac{1}{2}\sum_{j=1}^{N_z} \rho_j G_j^2(z)$ such that the natural step size $\alpha^{[k]} = 1$ of the Newton method is reduced but one can expect it to be close to 1 in a small neighborhood of z^*.

Remark 6. *Since our aim is to allow for large sampling periods T an adaptive step size control algorithm such as DoPri5, see [4], is necessary within each interval $[kT,(k+1)T)$, $k \in \boldsymbol{N}$, in order to avoid errors in the state trajectories and the cost functional which therefore has to be transformed. Note that the local accuracy guaranteed by the step size control here plays the role of the accuracy parameter δ in the consistency Definition 1 in [10].*

Remark 7. *The case of an integration step size h that is different from the sampling period T has been analysed theoretically for MPC schemes in [3]. An important aspect of this analysis is that h, or — more generally — the numerical accuracy parameter, can be assigned arbitrarily and independently of T (where of course one has to ensure that the sampling instants are included in the set of gridpoints used for integration in order to match the discontinuities of the control function). It should be noted that our algorithm fulfils this requirement. In fact, when we remove the x–subsystem (3) from our scheme and use a local Lyapunov function as a terminal cost we obtain exactly the direct MPC algorithm discussed theoretically in [3].*

Compared to this standard MPC approach the main difference of our scheme lies in the fact that we can directly enforce a desired transient behavior induced by the continuous–time feedback, while in standard MPC schemes the transient behavior can only be influenced indirectly through the choice of the cost functional. Clearly, the design of a suitable continuous–time controller requires a considerable amount of a priori work, but this may be rewarded by a better performance of the resulting sampled–data closed loop.

Remark 8. *An important problem is the choice of a good initial guess $v_{[0,M-1]}$ for the optimization, keeping in mind that we deal with a nonlinear optimization problem. Even though suboptimal solutions to this problem may be sufficient to ensure stability, see [6], here we also aim at good performance. Convergence to the global optimum, however, can only be expected when the initial solution is already close to it. When passing from t_k to t_{k+1} the shifted optimal control sequence from the previous step typically yields such a good initial guess, which is confirmed by our numerical experience that the computational costs for the optimization are decreasing monotonically during the iteration process.*

A more severe problem is the choice of the initial guess at t_0 when no previous optimal control is known. In this case, in our approach the known continuous–time feedback can be exploited for this purpose when the emulated feedback from Remark 1 yields solutions which do not deviate too far from the continuous–time reference. However, this method fails when the emulated feedback leads to unstable solutions and the time horizon $H = M{\cdot}T$ is rather large. Such situations can sometimes be handled by reducing the length of the horizon $H = M \cdot T$ but proceeding this way one has to keep in mind that there exists a lower bound for H from the stability proof. Also, simulations have shown that while on one hand computational costs grow with the length of the horizon, on the other hand better performance can be achieved using longer horizons. Therefore, at the moment it is up to simulations to find a good set of parameters and a good initial guess of $v_{[0,M-1]}$.

A promising alternative approach and topic of future research is whether some of the methods developed in [9, 13] can be used in order to construct the initial guess, an approach that would lead to a predictor–corrector type algorithm in which the MPC strategy plays the role of the corrector.

5 Example

Here we present a model of a synchronous generator taken from [2]

$$\dot{x}_1 = x_2, \quad \dot{x}_2 = -b_1 x_3 \sin x_1 - b_2 x_2 + P, \quad \dot{x}_3 = b_3 \cos x_1 - b_4 x_3 + E + u. \tag{16}$$

We use the parameter $b_1 = 34.29$, $b_2 = 0.0$, $b_3 = 0.149$, $b_4 = 0.3341$, $P = 28.22$ and $E = 0.2405$, as well as the continuous–time feedback law $u(x) = a_1((x_1 - x_1^*)b_4 + x_2)$ with feedback gain $a_1 > 0$, whose purpose is to enlarge the domain of attraction of the locally stable equilibrium $x^* \approx (1.12, 0.0, 0.914)$ (note that this equilibrium is locally asymptotically stable also for $u \equiv 0$). As initial value we used the vector $x_0 = (0.5, 0.0, 2.0)$ and generated results for $T = 0.1$, $T = 0.5$ and $a_1 = 0.45$.

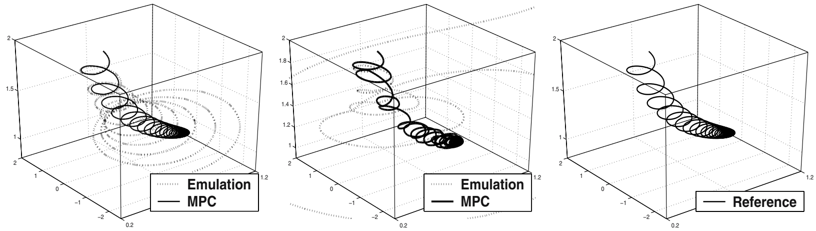


Fig. 1. Improvement by MPC control over emulation for $a_1 = 0.45$, $T = 0.1$ (left), $T = 0.5$ (middle) and reference solution using continuous–time feedback (right)

One can see that the fast dynamics of the problem require small sampling periods to maintain stability using the emulated feedback law. The MPC control on the other hand not only stabilizes the equilibrium even for rather large T but also keeps the sampled–data solution close to the reference.

6 Conclusion

We proposed an unconstrained model predictive algorithm for the sampled–data implementation of continuous–time stabilizing feedback laws. Stability and inverse optimality results were briefly revisited and numerical issues were discussed. Compared to direct MPC approaches without using continuous–time feedbacks, advantages of our method are that the sampled–data solutions inherit the performance properties of the continuous–time controller and that the knowledge of the continuous–time controller helps to reduce the computational cost of the numerical optimization. Future research will include a systematic study about how this knowledge can be used in a numerically efficient way and an extension of our approach to dynamic continuous–time controllers.

References

[1] Grimm G, Messina M J, Teel A R, Tuna S (2005) Model predictive control: for want of a local control Lyapunov function, all is not lost. In: IEEE Trans. Automat. Contr. 50: 546–558

[2] Grüne L (2001) Subdivision Techniques for the Computation of Domains of Attractions and Reachable Sets. In: Proceedings of NOLCOS 2001: 762–767

[3] Gyurkovics E, Elaiw A M (2004) Stabilization of sampled–data nonlinear systems by receding horizon control via discrete–time approximations. In: Automatica 40: 2017–2028

[4] Hairer E, Nørsett S P, Wanner G (1993) Solving Ordinary Differential Equations I. Nonstiff Problems. 2nd Ed., Springer–Verlag

[5] Jadbabaie A, Hauser J, Yu J (2001) Unconstrained receding horizon control of nonlinear systems. In: IEEE Trans. Automat. Contr. 46: 776–783

[6] Jadbabaie A, Hauser J (2005) On the stability of receding horizon control with a general terminal cost. In: IEEE Trans. Automat. Contr. 50: 674–678

[7] Malanowski K, Büskens Ch, Maurer H (1997) Convergence of approximations to nonlinear optimal control problems. In Fiacco, A. (ed.) Mathematical Programming with Data Perturbations. Marcel Dekker

[8] Laila D S, Nešić D, Teel A R (2002) Open and closed loop dissipation inequalities under sampling and controller emulation. In: Europ. J. Control 18: 109–125

[9] Monaco S, Normand–Cyrot D (2001) Issues on nonlinear digital control. In: Europ. J. Control 7: 160–178

[10] Nešić D, Teel A R, Sontag E D (1999) Formulas relating KL stability estimates of discrete–time and sampled–data nonlinear systems. In: Syst. Contr. Lett. 38: 49–60

[11] Nešić D, Teel A R (2004) A framework for stabilization of nonlinear sampled–data systems based on their approximate discrete–time models. In: IEEE Trans. Automat. Contr. 49: 1103-1122

[12] Nešić D, Teel A R, Kokotovic P V (1999) Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. In: Sys. Contr. Lett. 38: 259–270

[13] Nešić D, Grüne L (2005) Lyapunov based continuous–time nonlinear controller redesign for sampled–data implementation. In: Automatica 41: 1143–1156

[14] Nešić D, Grüne L (2006) A receding horizon control approach to sampled–data implementation of continuous–time controllers. In: Syst. Contr. Lett. 55: 546–558

Sampled-Data Model Predictive Control for Nonlinear Time-Varying Systems: Stability and Robustness*

Fernando A.C.C. Fontes[1], Lalo Magni[2], and Éva Gyurkovics[3]

[1] Officina Mathematica, Departamento de Matemática para a Ciência e Tecnologia, Universidade do Minho, 4800-058 Guimarães, Portugal
ffontes@mct.uminho.pt

[2] Dipartimento di Informatica e Sistimistica, Università degli Studi di Pavia, via Ferrata 1, 27100 Pavia, Italy
lalo.magni@unipv.it

[3] Budapest University of Technology and Economics, Institute of Mathematics, Budapest H-1521, Hungary
gye@math.bme.hu

Summary. We describe here a sampled-data Model Predictive Control framework that uses continuous-time models but the sampling of the actual state of the plant as well as the computation of the control laws, are carried out at discrete instants of time. This framework can address a very large class of systems, nonlinear, time-varying, and nonholonomic.

As in many others sampled-data Model Predictive Control schemes, Barbalat's lemma has an important role in the proof of nominal stability results. It is argued that the generalization of Barbalat's lemma, described here, can have also a similar role in the proof of robust stability results, allowing also to address a very general class of nonlinear, time-varying, nonholonomic systems, subject to disturbances. The possibility of the framework to accommodate discontinuous feedbacks is essential to achieve both nominal stability and robust stability for such general classes of systems.

1 Introduction

Many Model Predictive Control (MPC) schemes described in the literature use continuous-time models and sample the state of the plant at discrete instants of time. See e.g. [3, 7, 9, 13] and also [6]. There are many advantages in considering a continuous-time model for the plant. Nevertheless, any implementable MPC scheme can only measure the state and solve an optimization problem at discrete instants of time.

In all the references cited above, Barbalat's lemma, or a modification of it, is used as an important step to prove stability of the MPC schemes. (Barbalat's

* The financial support from MURST Project "New techniques for the identification and adaptive control of industrial systems", from FCT Project POCTI/MAT/61842/2004, and from the Hungarian National Science Foundation for Scientific Research grant no. T037491 is gratefully acknowledged.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 115–129, 2007.
springerlink.com

lemma is a well-known and powerful tool to deduce asymptotic stability of nonlinear systems, especially time-varying systems, using Lyapunov-like approaches; see e.g. [17] for a discussion and applications). To show that an MPC strategy is stabilizing (in the nominal case), it is shown that if certain design parameters (objective function, terminal set, etc.) are conveniently selected, then the value function is monotone decreasing. Then, applying Barbalat's lemma, attractiveness of the trajectory of the nominal model can be established (i.e. $x(t) \to 0$ as $t \to \infty$). This stability property can be deduced for a very general class of nonlinear systems: including time-varying systems, nonholonomic systems, systems allowing discontinuous feedbacks, etc. If, in addition, the value function possesses some continuity properties, then Lyapunov stability (i.e. the trajectory stays arbitrarily close to the origin provided it starts close enough to the origin) can also be guaranteed (see e.g. [11]). However, this last property might not be possible to achieve for certain classes of systems, for example a car-like vehicle (see [8] for a discussion of this problem and this example).

A similar approach can be used to deduce robust stability of MPC for systems allowing uncertainty. After establishing monotone decrease of the value function, we would want to guarantee that the state trajectory asymptotically approaches some set containing the origin. But, a difficulty encountered is that the predicted trajectory only coincides with the resulting trajectory at specific sampling instants. The robust stability properties can be obtained, as we show, using a generalized version of Barbalat's lemma. These robust stability results are also valid for a very general class of nonlinear time-varying systems allowing discontinuous feedbacks.

The optimal control problems to be solved within the MPC strategy are here formulated with very general admissible sets of controls (say, measurable control functions) making it easier to guarantee, in theoretical terms, the existence of solution. However, some form of finite parameterization of the control functions is required/desirable to solve on-line the optimization problems. It can be shown that the stability or robustness results here described remain valid when the optimization is carried out over a finite parameterization of the controls, such as piecewise constant controls (as in [13]) or as bang-bang discontinuous feedbacks (as in [9]).

2 A Sampled-Data MPC Framework

We shall consider a nonlinear plant with input and state constraints, where the evolution of the state after time t_0 is predicted by the following model.

$$\dot{x}(s) = f(s, x(s), u(s)) \qquad \text{a.e. } s \geq t_0, \tag{1a}$$

$$x(t_0) = x_{t_0} \in X_0, \tag{1b}$$

$$x(s) \in X \subset \mathbb{R}^n \qquad \text{for all } s \geq t_0, \tag{1c}$$

$$u(s) \in U \qquad \text{a.e. } s \geq t_0. \tag{1d}$$

The data of this model comprise a set $X_0 \subset \mathbb{R}^n$ containing all possible initial states at the initial time t_0, a vector x_{t_0} that is the state of the plant measured at time t_0, a given function $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$, and a set $U \subset \mathbb{R}^m$ of possible control values.

We assume this system to be asymptotically controllable on X_0 and that for all $t \geq 0$ $f(t, 0, 0) = 0$. We further assume that the function f is continuous and locally Lipschitz with respect to the second argument.

The construction of the feedback law is accomplished by using a sampled-data MPC strategy. Consider a sequence of sampling instants $\pi := \{t_i\}_{i \geq 0}$ with a constant inter-sampling time $\delta > 0$ such that $t_{i+1} = t_i + \delta$ for all $i \geq 0$. Consider also the control horizon and predictive horizon, T_c and T_p, with $T_p \geq T_c > \delta$, and an auxiliary control law $k^{aux} : \mathbb{R} \times \mathbb{R}^n \to \mathbb{R}^m$. The feedback control is obtained by repeatedly solving online open-loop optimal control problems $\mathcal{P}(t_i, x_{t_i}, T_c, T_p)$ at each sampling instant $t_i \in \pi$, every time using the current measure of the state of the plant x_{t_i}.

$\mathcal{P}(t, x_t, T_c, T_p)$: Minimize

$$\int_t^{t+T_p} L(s, x(s), u(s))ds + W(t + T_p, x(t + T_p)), \tag{2}$$

subject to:

$$\begin{aligned}
&\dot{x}(s) = f(s, x(s), u(s)) && \text{a.e. } s \in [t, t + T_p], && (3)\\
&x(t) = x_t, && && \\
&x(s) \in X && \text{for all } s \in [t, t + T_p], && \\
&u(s) \in U && \text{a.e. } s \in [t, t + T_c], && \\
&u(s) = k^{aux}(s, x(s)) && \text{a.e. } s \in [t + T_c, t + T_p], && \\
&x(t + T_p) \in S. && && (4)
\end{aligned}$$

Note that in the interval $[t + T_c, t + T_p]$ the control value is selected from a singleton and therefore the optimization decisions are all carried out in the interval $[t, t + T_c]$ with the expected benefits in the computational time.

The notation adopted here is as follows. The variable t represents real time while we reserve s to denote the time variable used in the prediction model. The vector x_t denotes the actual state of the plant measured at time t. The process (x, u) is a pair trajectory/control obtained from the model of the system. The trajectory is sometimes denoted as $s \mapsto x(s; t, x_t, u)$ when we want to make explicit the dependence on the initial time, initial state, and control function. The pair $(\bar{x}, \bar{u})$ denotes our optimal solution to an open-loop optimal control problem. The process (x^*, u^*) is the closed-loop trajectory and control resulting from the MPC strategy. We call *design parameters* the variables present in the open-loop optimal control problem that are not from the system model (i.e. variables we are able to choose); these comprise the control horizon T_c, the prediction horizon T_p, the running cost and terminal costs functions L and W, the auxiliary control law k^{aux}, and the terminal constraint set $S \subset \mathbb{R}^n$.

The MPC algorithm performs according to a receding horizon strategy, as follows.

1. Measure the current state of the plant $x^*(t_i)$.
2. Compute the open-loop optimal control $\bar{u} : [t_i, t_i + T_c] \to \mathrm{I\!R}^n$ solution to problem $\mathcal{P}(t_i, x^*(t_i), T_c, T_p)$.
3. Apply to the plant the control $u^*(t) := \bar{u}(t; t_i, x^*(t_i))$ in the interval $[t_i, t_i + \delta)$ (the remaining control $\bar{u}(t), t \geq t_i + \delta$ is discarded).
4. Repeat the procedure from (1.) for the next sampling instant t_{i+1} (the index i is incremented by one unit).

The resultant control law u^* is a "sampling-feedback" control since during each sampling interval, the control u^* is dependent on the state $x^*(t_i)$. More precisely the resulting trajectory is given by

$$x^*(t_0) = x_{t_0}, \quad \dot{x}^*(t) = f(t, x^*(t), u^*(t)) \quad t \geq t_0,$$

where

$$u^*(t) = k(t, x^*(\lfloor t \rfloor_\pi)) := \bar{u}(t; \lfloor t \rfloor_\pi, x^*(\lfloor t \rfloor_\pi)) \quad t \geq t_0.$$

and the function $t \mapsto \lfloor t \rfloor_\pi$ gives the last sampling instant before t, that is

$$\lfloor t \rfloor_\pi := \max_i \{t_i \in \pi : t_i \leq t\}.$$

Similar sampled-data frameworks using continuous-time models and sampling the state of the plant at discrete instants of time were adopted in [2, 6, 7, 8, 13] and are becoming the accepted framework for continuous-time MPC. It can be shown that with this framework it is possible to address —and guarantee stability, and robustness, of the resultant closed-loop system — for a very large class of systems, possibly nonlinear, time-varying and nonholonomic.

3 Nonholonomic Systems and Discontinuous Feedback

There are many physical systems with interest in practice which can only be modelled appropriately as nonholonomic systems. Some examples are the wheeled vehicles, robot manipulators, and many other mechanical systems.

A difficulty encountered in controlling this kind of systems is that any linearization around the origin is uncontrollable and therefore any linear control methods are useless to tackle them. But, perhaps the main challenging characteristic of the nonholonomic systems is that it is not possible to stabilize it if just time-invariant continuous feedbacks are allowed [1]. However, if we allow discontinuous feedbacks, it might not be clear what is the solution of the dynamic differential equation. (See [4, 8] for a further discussion of this issue).

A solution concept that has been proved successful in dealing with stabilization by discontinuous feedbacks for a general class of controllable systems is the concept of "sampling-feedback" solution proposed in [5]. It can be seen that sampled-data MPC framework described can be combined naturally with a “sampling-feedback” law and thus define a trajectory in a way which is very similar to the concept introduced in [5]. Those trajectories are, under mild conditions, well-defined even when the feedback law is discontinuous.

There are in the literature a few works allowing discontinuous feedback laws in the context of MPC. (See [8] for a survey of such works.) The essential feature of those frameworks to allow discontinuities is simply the sampled-data feature — appropriate use of a positive inter-sampling time, combined with an appropriate interpretation of a solution to a discontinuous differential equation.

4 Barbalat’s Lemma and Variants

Barbalat’s lemma is a well-known and powerful tool to deduce asymptotic stability of nonlinear systems, especially time-varying systems, using Lyapunov-like approaches (see e.g. [17] for a discussion and applications).

Simple variants of this lemma have been used successfully to prove stability results for Model Predictive Control (MPC) of nonlinear and time-varying systems [7, 15]. In fact, in all the sampled-data MPC frameworks cited above, Barbalat’s lemma, or a modification of it, is used as an important step to prove stability of the MPC schemes. It is shown that if certain design parameters (objective function, terminal set, etc.) are conveniently selected, then the value function is monotone decreasing. Then, applying Barbalat’s lemma, attractiveness of the trajectory of the nominal model can be established (i.e. $x(t) \to 0$ as $t \to \infty$). This stability property can be deduced for a very general class of nonlinear systems: including time-varying systems, nonholonomic systems, systems allowing discontinuous feedbacks, etc.

A recent work on robust MPC of nonlinear systems [9] used a generalization of Barbalat’s lemma as an important step to prove stability of the algorithm. However, it is our believe that such generalization of the lemma might provide a useful tool to analyse stability in other robust continuous-time MPC approaches, such as the one described here for time-varying systems.

A standard result in Calculus states that if a function is lower bounded and decreasing, then it converges to a limit. However, we cannot conclude whether its derivative will decrease or not unless we impose some smoothness property on $\dot{f}(t)$. We have in this way a well-known form of the Barbalat’s lemma (see e.g. [17]).

Lemma 1 (Barbalat’s lemma 1). *Let $t \mapsto F(t)$ be a differentiable function with a finite limit as $t \to \infty$. If $\dot{F}$ is uniformly continuous, then $\dot{F}(t) \to 0$ as $t \to \infty$.*

A simple modification that has been useful in some MPC (nominal) stability results [7, 15] is the following.

Lemma 2 (Barbalat’s lemma 2). *Let M be a continuous, positive definite function and x be an absolutely continuous function on $\mathbb{R}$. If $\|x(\cdot)\|_{L^\infty} < \infty$, $\|\dot{x}(\cdot)\|_{L^\infty} < \infty$, and $\lim_{T\to\infty} \int_0^T M(x(t))\,dt < \infty$, then $x(t) \to 0 \quad$ as $\quad t \to \infty$.*

Now, suppose that due to disturbances we have no means of guaranteeing that all the hypothesis of the lemma are satisfied for the trajectory x^* we want to analyse. Instead some hypothesis are satisfied on a neighbouring trajectory $\hat{x}$ that coincides with the former at a sequence of instants of time. [1] Furthermore, suppose that instead of approaching the origin we would like to approach some set containing the origin. These are the conditions of the following lemma.

Definition 1. *Let A be a nonempty, closed subset of $\mathbb{R}^n$. The function $x \mapsto d_A(x)$, from $\mathbb{R}^n$ to $\mathbb{R}$, denotes the distance from a point x to the set A (i.e., $d_A(x) := \min_{y \in A} \|x - y\|$).*

We say that a function M is positive definite with respect to the set A *if $M(x) > 0$ for all $x \notin A$ and $M(x) = 0$ for some $x \in A$.*

Lemma 3 (A generalization of Barbalat's lemma). *Let A be subset of $\mathbb{R}^n$ containing the origin, and $M : \mathbb{R}^n \to \mathbb{R}$ be a continuous function which is positive definite with respect to A.*

Let $\Delta > 0$ be given and for any $\delta \in (0, \Delta)$ consider the functions x^*_δ and $\hat{x}_\delta$ from $\mathbb{R}^+$ to $\mathbb{R}^n$ satisfying the following properties:

- *The function x^*_δ is absolutely continuous, the function $\hat{x}_\delta$ is absolutely continuous on each interval $[i\delta, (i+1)\delta)$, for all $i \in \mathbb{N}_0$, and $\hat{x}_\delta(i\delta) = x^*_\delta(i\delta)$ for all $i \in \mathbb{N}_0$.*
- *There exist positive constants K_1, K_2 and K_3 such that for all $\delta \in (0, \Delta)$*

$$\|\dot{x}^*_\delta(\cdot)\|_{L^\infty(0,\infty)} < K_1, \quad \|\hat{x}_\delta(\cdot)\|_{L^\infty(0,\infty)} < K_2, \quad \|\dot{\hat{x}}_\delta(\cdot)\|_{L^\infty(0,\infty))} < K_3.$$

Moreover,

$$\lim_{\tau \to \infty} \int_0^\tau M(\hat{x}_\delta(t))\, dt < \infty, \tag{5}$$

Then for any $\epsilon > 0$ there is a $\delta(\epsilon)$ and for any $\delta \in (0, \delta(\epsilon))$ there is a $T = T(\epsilon, \delta) > 0$ such that

$$d_A(\hat{x}_\delta(t)) \leq \epsilon, \quad d_A(x^*_\delta(t)) \leq \epsilon \quad \textit{for all } t \geq T. \tag{6}$$

Proof of Lemma 3

First we shall show that the statement is true for the parameterized family $\widehat{x}_\delta(.)$. Suppose the contrary: there exists an ε_0 such that for any $\overline{\delta} \in (0, \Delta]$ – thus for $\overline{\delta} = \min\{\Delta, \varepsilon_0/(2K_3)\}$, as well – one can show a $\widetilde{\delta} \in (0, \overline{\delta}]$ and a sequence $\{t_k\}_{k=0}^\infty$ with $t_k \to \infty$ as $k \to \infty$ so that $d_A(\widehat{x}_\delta(t_k)) > \varepsilon_0$ for all $k \in \mathbb{N}$.

Without loss of generality, we may assume that $t_{k+1} - t_k \geq \widetilde{\delta}$, $k \in \mathbb{N}$. Let $R \geq K_2$ be such that the set $B = \{x \in \mathbb{R}^n : d_A(x)) \geq \varepsilon_0/2 \text{ and } \|x\| \leq R\}$ is nonempty. Since B is compact, M is continuous and $M(x) > 0$, if $x \in B$, there

[1] In an NMPC context, $\hat{x}$ would represent the concatenation of predicted trajectories; see equation (12).

exists an $m > 0$ such that $m \leq M(x)$ for all $x \in B$. Since $t_k \to \infty$ as $k \to \infty$, for any $k \in \mathbb{N}$ one can show a j_k so that $t_k \in [j_k\widetilde{\delta},\, (j_k+1)\widetilde{\delta})$. Note that for any $t \in [j_k\widetilde{\delta},\, (j_k+1)\widetilde{\delta})$ we have

$$\left\|\widehat{x}_{\widetilde{\delta}}(t) - \widehat{x}_{\widetilde{\delta}}(t_k)\right\| \leq \int_{t_k}^{t} \left\|\dot{\widehat{x}}_{\widetilde{\delta}}(s)\right\| ds \leq K_3\widetilde{\delta} \leq \varepsilon_0/2.$$

Then, by the triangle inequality

$$d_A(\widehat{x}_{\widetilde{\delta}}(t)) \geq d_A(\widehat{x}_{\widetilde{\delta}}(t_k)) - \left\|\widehat{x}_{\widetilde{\delta}}(t) - \widehat{x}_{\widetilde{\delta}}(t_k)\right\| \geq \varepsilon_0/2.$$

Therefore $\widehat{x}_{\widetilde{\delta}}(t) \in B$, if $t \in [j_k\widetilde{\delta},\, (j_k+1)\widetilde{\delta})$, thus

$$\int_{j_k\widetilde{\delta}}^{(j_k+1)\widetilde{\delta}} M(\widehat{x}_{\widetilde{\delta}}(s))ds \geq m\widetilde{\delta}.$$

This would imply that $\lim_{\tau\to\infty} \int_0^\tau M(\widehat{x}_\delta\,(s))ds \to \infty$ contradicting (5).

Now let $\varepsilon > 0$ be arbitrarily given, let $\delta_1 := \varepsilon/(2K_1)$, and let $\varepsilon_1 := \varepsilon/2$. From the preceding part of the proof we already know that there is a $\delta_2 = \widehat{\delta}(\varepsilon_1)$, and for any $0 < \delta < \min\{\delta_1, \delta_2\}$ there is a $\widehat{T}(\varepsilon_1, \delta)$ such that

$$d_A(\widehat{x}_\delta(t)) \leq \varepsilon_1 \qquad \text{for all } t \geq \widehat{T}(\varepsilon_1, \delta).$$

On the other hand, if $t \geq \widehat{T}(\varepsilon_1, \delta)$ is arbitrary but fixed, then $t \in [i\delta,\, (i+1)\delta)$ for some i, thus by the triangle inequality and by the assumptions of the lemma we have

$$d_A(x^*_\delta(t)) \leq d_A(x^*_\delta(i\delta)) + \|x^*_\delta(t) - x^*_\delta(i\delta)\| \leq \varepsilon_1 + K_1\delta \leq \varepsilon.$$

Therefore $\delta(\varepsilon) = \min\{\delta_1, \delta_2\}$ and $T(\varepsilon, \delta) = \widehat{T}(\varepsilon_1, \delta)$ are suitable. □

5 Nominal Stability

A stability analysis can be carried out to show that if the design parameters are conveniently selected (i.e. selected to satisfy a certain sufficient stability condition, see e.g. [7]), then a certain MPC value function V is shown to be monotone decreasing. More precisely, for some $\delta > 0$ small enough and for any $t'' > t' > 0$

$$V(t'', x^*(t'')) - V(t', x^*(t')) \leq -\int_{t'}^{t''} M(\hat{x}(s))ds. \tag{7}$$

where M is a continuous, radially unbounded, positive definite function. The MPC value function V is defined as

$$V(t,x) := V_{\lfloor t \rfloor_\pi}(t,x)$$

where $V_{t_i}(t,x_t)$ is the value function for the optimal control problem $\mathcal{P}(t,x_t,T_c-(t-t_i),T_c-(t-t_i))$ (the optimal control problem defined where the horizon is shrank in its initial part by $t-t_i$).

From (7) we can then write that for any $t \geq t_0$

$$0 \leq V(t,x^*(t)) \leq V(t_0,x^*(t_0)) - \int_{t_0}^{t} M(x^*(s))ds.$$

Since $V(t_0,x^*(t_0))$ is finite, we conclude that the function $t \mapsto V(t,x^*(t))$ is bounded and then that $t \mapsto \int_{t_0}^{t} M(x^*(s))ds$ is also bounded. Therefore $t \mapsto x^*(t)$ is bounded and, since f is continuous and takes values on bounded sets of (x,u), $t \mapsto \dot{x}^*(t)$ is also bounded. All the conditions to apply Barbalat's lemma 2 are met, yielding that the trajectory asymptotically converges to the origin. Note that this notion of stability does not necessarily include the Lyapunov stability property as is usual in other notions of stability; see [8] for a discussion.

6 Robust Stability

In the last years the synthesis of robust MPC laws is considered in different works [14].

The framework described below is based on the one in [9], extended to time-varying systems.

Our objective is to drive to a given target set Θ ($\subset \mathbb{R}^n$) the state of the nonlinear system subject to bounded disturbances

$$\dot{x}(t) = f(t,x(t),u(t),d(t)) \qquad \text{a.e. } t \geq t_0, \tag{8a}$$

$$x(t_0) = x_0 \in X_0, \tag{8b}$$

$$x(t) \in X \qquad \text{for all } t \geq t_0, \tag{8c}$$

$$u(t) \in U \qquad \text{a.e. } t \geq t_0, \tag{8d}$$

$$d(t) \in D \qquad \text{a.e. } t \geq t_0, \tag{8e}$$

where $X_0 \subset \mathbb{R}^n$ is the set of possible initial states, $X \subset \mathbb{R}^n$ is the set of possible states of the trajectory, $U \subset \mathbb{R}^m$ is a bounded set of possible control values, $D \subset \mathbb{R}^p$ is a bounded set of possible disturbance values, and $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^p \to \mathbb{R}^n$ is a given function. The state at time t from the trajectory x, starting from x_0 at t_0, and solving (8a) is denoted $x(t;t_0,x_0,u,d)$ when we want to make explicit the dependence on the initial state, control and disturbance. It is also convenient to define, for $t_1,\ t_2 \geq t_0$, the function spaces

$$\mathcal{U}([t_1,t_2]) := \{u : [t_1,t_2] \to \mathbb{R}^m : u(s) \in U,\ s \in [t_1,t_2]\},$$

$$\mathcal{D}([t_1,t_2]) := \{d : [t_1,t_2] \to \mathbb{R}^p : d(s) \in D,\ s \in [t_1,t_2]\}.$$

The target set Θ is a closed set, contains the origin and is robustly invariant under no control. That is, $x(t;t_0,x_0,u,d) \in \Theta$ for all $t \geq t_0$, all $x_0 \in \Theta$, and all $d \in \mathcal{D}([t_0,t))$ when $u \equiv 0$. We further assume that f is a continuous function and locally Lipschitz continuous with respect to x.

Consider a sequence of sampling instants $\pi := \{t_i\}_{i\geq 0}$ with constant inter-sampling times $\delta > 0$ such that $t_{i+1} = t_i + \delta$ for all $i \geq 0$. Let the control horizon T_c and prediction horizon T_p, with $T_c \leq T_p$, be multiples of δ ($T_c = N_c\delta$ and $T_p = N_p\delta$ with $N_c, N_p \in \mathbb{N}$). Consider also a terminal set S ($\subset \mathbb{R}^n$), a terminal cost function $W : \mathbb{R}^n \to \mathbb{R}$ and a running cost function $L : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$. The optimization problem is a finite horizon differential game where the disturbance d acts as the maximizing player and the control u acts as the minimizing player. We shall assume that the minimizing player uses a sampled-data information structure. The space of the corresponding strategies over $[t_1,t_2]$ we denote by $\mathcal{K}([t_1,t_2])$. For any $t \in \pi$, let $k_t^{aux} \in \mathcal{K}([t+T_c, t+T_p])$ be an a priori given auxiliary sampled-data strategy. The quantities time horizons T_c and T_p, objective functions L and W, terminal constraint set S, the inter-sampling time δ, and auxiliary strategy k_t^{aux} are the quantities we are able to tune — the so-called *design parameters* — and should be chosen to satisfy the robust stability condition described below.

At a certain instant $t \in \pi$, we select for the prediction model the control strategy for the intervals $[t, t+T_c)$ and $[t+T_c, t+T_p)$ in the following way. In the interval $[t, t+T_c)$, we should select, by solving an optimization problem, the strategy k_t in the interval $[t, t+T_c]$. The strategy k_t^{aux}, known a priori, is used in the interval $[t+T_c, t+T_p]$.

The robust feedback MPC strategy is obtained by repeatedly solving on-line, at each sampling instant t_i, a min-max optimization problem $\mathcal{P}$, to select the feedback k_{t_i}, every time using the current measure of the state of the plant x_{t_i}.

$\mathcal{P}(t, x_t, T_c, T_p)$: $\text{Min}_{k\in\mathcal{K}([t,t+T_c])}\text{Max}_{d\in\mathcal{D}([t,T_p])}$

$$\int_t^{t+T_p} L(x(s),u(s))ds + W(x(t+T_p)) \tag{9}$$

subject to:

$$\begin{aligned}
&x(t) = x_t \\
&\dot{x}(s) = f(s, x(s), u(s), d(s)) && \text{a.e. } s \in [t, t+T_p] \qquad (10)\\
&x(s) \in X && \text{for all } s \in [t, t+T_p] \\
&u(s) \in U && \text{a.e. } s \in [t, t+T_p] \\
&x(t+T_p) \in S, && \qquad (11)
\end{aligned}$$

where

$$u(s) = k_t(s, x(\lfloor s \rfloor_\pi)) \quad \text{for } s \in [t, t+T_c)$$
$$u(s) = k_t^{aux}(s, x(\lfloor s \rfloor_\pi)) \quad \text{for} \quad s \in [t+T_c, t+T_p).$$

In this optimization problem we use the convention that if some of the constraint is not satisfied, then the value of the game is $+\infty$. This ensures that when the value of the game is finite, the optimal control strategy guarantees the satisfaction of the constraints for all possible disturbance scenarios.

The MPC algorithm performs according to a Receding Horizon strategy, as follows:

1. Measure the current state of the plant $x^*(t_i)$.
2. Compute the feedback k_{t_i}, solution to problem $\mathcal{P}(t_i, x^*(t_i), T_c, T_p)$.
3. Apply to the plant the control given by the feedback law k_{t_i} in the interval $[t_i, t_i+\delta)$, (discard all the remaining data for $t \geq t_i + \delta$).
4. Repeat the procedure from (1.) for the next sampling instant t_{i+1}.

The main stability result states that if the design parameters are chosen to satisfy the robust stability conditions RSC, then the MPC strategy ensures steering to a certain target set Θ. The following definitions will be used.

Definition 2. *The sampling-feedback k is said to* robustly stabilize *the system to the target set Θ if for any $\epsilon > 0$ there exists a sufficiently small inter-sample time δ such that we can find a scalar $T > 0$ satisfying $d_\Theta(x(t)) \leq \epsilon$ for all $t \geq T$.*

Definition 3. *The* playable set *$\Omega(t, T_c, T_p, S)$ is the set of all initial states x_t for which using the inter-sampling time $\delta \in (0, \Delta]$ and the auxiliary strategy k_t^{aux} there exists some control strategy $k_t \in \mathcal{K}$ for $[t, t+T_p]$ with $k_t(s,.) = k_t^{aux}(s,.)$ for $s \in [t+T_c, t+T_p]$ such that $x(t+T_p; t, x_t, k_t, d) \in S$ for all $d \in \mathcal{D}([t, t+T_p])$.*

Consider the following robust stability condition

RSC: The design parameters: time horizons T_c and T_p, objective functions L and W, terminal constraint set S, inter-sampling time δ, and auxiliary feedback strategy k_t^{aux} satisfy

RSC1 The set S is closed, contains the origin, and is contained in X. Also $k_t^{aux}(s,x) \in U$ for all $x \in X$, $s \in [t+T_c, t+T_p]$ and $t \in \pi$.

RSC2 The function L is continuous, $L(0,0) = 0$, and for all $u \in U$ we have that $L(x,u) \geq M(x)$ for some function $M : \mathbb{R}^n \to \mathbb{R}_+$ which is continuous, radially unbounded and positive definite with respect to the set Θ.

RSC3 The function W is Lipschitz continuous and $W(x) \geq 0$ for all $x \in \mathbb{R}^n \setminus \{0\}$.

RSC4 The set of initial states X_0 is contained in the playable set $\Omega(t_0, T_c, T_p, S)$.

RSC5 For each sampling instant $t \in \pi$ and each $x_t \in S \setminus \Theta$, and for all possible disturbances $d \in \mathcal{D}([t, t+\delta)])$, using the notation $x(s) = x(s; t, x_t, k_t^{aux}, d)$, we have

$$W(x(t+\delta)) - W(x_t) \leq - \int_t^{t+\delta} L(x(s), k_t^{aux}(s, x_t))\ \mathrm{d}s \qquad (RSC5a)$$

$$x(t+\delta) \in S. \qquad (RSC5b)$$

We are in the conditions to state the following stability result.

Theorem 1. *Assume condition RSC is satisfied and that the differential games $\mathcal{P}(t, x_t, T_c, T_p)$ have a value for all $x_t \in X$ and $t \geq t_0$. Then, the robust MPC strategy robustly stabilizes the system to the target set Θ.*

Proof

The proof starts by establishing a monotone decreasing property of the MPC Value function. Then the application of the generalized Barbalat's Lemma yields the robust stability result.

Let $V(t,x)$ be the value function of $\mathcal{P}(t, x, T_c - (t - t_i), T_p - (t - t_i))$ with $t_i = \lfloor t \rfloor_\pi$. Let also $\hat{x}$ be the concatenation of predicted trajectories $\bar{x}$ for each optimization problem. That is for $i \geq 0$

$$\hat{x}(t) = \bar{x}^i(t) \quad \text{for all } t \in [t_i, t_i + \delta) \tag{12}$$

where $\bar{x}^i$ is the trajectory of a solution to problem $\mathcal{P}(t_i, x^*(t_i), T_c, T_p)$. Note that $\hat{x}$ coincides with x^* at all sampling instants $t_i \in \pi$, but they are typically not identical on $[t_i, t_i + \delta)$, since they correspond to different disturbances.

The following lemma establishes a monotone decreasing property of V .

Lemma 4. *([9], Lemma 4.4) There exists an inter-sample time $\delta > 0$ small enough such that for any $t' < t''$, if $x^*(t''), x^*(t') \notin \Theta$ then*

$$V(t'', x^*(t'')) - V(t', x^*(t')) \leq -\int_{t'}^{t''} M(\hat{x}(s))ds.$$

We can then write that for any $t \geq t_0$

$$0 \leq V(t, x^*(t)) \leq V(t_0, x^*(t_0)) - \int_{t_0}^{t} M(\hat{x}(s))ds.$$

Since $V(t_0, x^*(t_0))$ is finite, we conclude that the function $t \mapsto V(t, x^*(t))$ is bounded and then that $t \mapsto \int_{t_0}^{t} M(\hat{x}(s))ds$ is also bounded. Therefore $t \mapsto \hat{x}(t)$ is bounded and, since f is continuous and takes values on bounded sets of (x, u, d), $t \mapsto \dot{\hat{x}}(t)$ is also bounded. Using the fact that x^* is absolutely continuous and coincides with $\hat{x}$ at all sampling instants, we may deduce that $t \mapsto \dot{x}^*(t)$ and $t \mapsto x^*(t)$ are also bounded. We are in the conditions to apply the previously established Generalization of Barbalat's Lemma 3, yielding the assertion of the theorem. □

7 Finite Parameterizations of the Control Functions

The results on stability and robust stability were proved using an optimal control problem where the controls are functions selected from a very general set (the set of measurable functions taking values on a set U, subset of R^m). This is adequate to prove theoretical stability results and it even permits to use the results on existence of a minimizing solution to optimal control problems (e.g. [7, Proposition 2]). However, for implementation, using any optimization algorithm, the control functions need to be described by a finite number of parameters (the so called finite parameterizations of the control functions). The control can be parameterized as piecewise constant controls (e.g. [13]), polynomials or splines described by a finite number of coeficients, bang-bang controls (e.g. [9, 10]), etc. Note that we are not considering discretization of the model or the dynamic equation. The problems of discrete approximations are discussed in detail e.g. in [16] and [12].

But, in the proof of stability, we just have to show at some point that the optimal cost (the value function) is lower than the cost of using another *admissible* control. So, as long as the set of admissible control values U is constant for all time, an easy, but nevertheless important, corollary of the previous stability results follows

> *If we consider the set of admissible control functions (including the auxiliary control law) to be a finitely parameterizable set such that the set of admissible control values is constant for all time, then both the nominal stability and robust stability results here described remain valid.*

An example, is the use of discontinuous feedback control strategies of bang-bang type, which can be described by a small number of parameters and so make the problem computationally tractable. In bang-bang feedback strategies, the controls values of the strategy are only allowed to be at one of the extremes of its range. Many control problems of interest admit a bang-bang stabilizing control. Fontes and Magni [9] describe the application of this parameterization to a unicycle mobile robot subject to bounded disturbances.

References

[1] R. W. Brockett. Asymptotic stability and feedback stabilization. In R. W. Brockett, R. S. Millman, and H. S. Sussmann, editors, *Differential Geometric Control Theory*, pages 181–191. Birkhouser, Boston, 1983.

[2] H. Chen and F. Allgöwer. Nonlinear model predictive control schemes with guaranteed stability. In R. Berber and C. Kravaris, editors, *Nonlinear Model Based Process Control*. Kluwer, 1998.

[3] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217, 1998.

[4] F. H. Clarke. Nonsmooth analysis in control theory: a survey. *European Journal of Control; Special issue: Fundamental Issues in Control*, 7:145–159, 2001.

[5] F. H. Clarke, Y. S. Ledyaev, E. D. Sontag, and A. I. Subbotin. Asymptotic controllability implies feedback stabilization. *IEEE Transactions on Automatic Control*, 42(10):1394–1407, 1997.

[6] R. Findeisen, L. Imsland, F. Allgower, and B. Foss. Towards a sampled-data theory for nonlinear model predictive control. In W. Kang, M. Xiao, and C. Borges, editors, *New Trends in Nonlinear Dynamics and Control, and their applications*, volume 295 of *Lecture Notes in Control and Information Sciences*, pages 295–311. Springer Verlag, Berlin, 2003.

[7] F. A. C. C. Fontes. A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42:127–143, 2001.

[8] F. A. C. C. Fontes. Discontinuous feedbacks, discontinuous optimal controls, and continuous-time model predictive control. *International Journal of Robust and Nonlinear Control*, 13(3–4):191–209, 2003.

[9] F. A. C. C. Fontes and L. Magni. Min-max model predictive control of nonlinear systems using discontinuous feedbacks. *IEEE Transactions on Automatic Control*, 48:1750–1755, 2003.

[10] L. Grüne. Homogeneous state feedback stabilization of homogeneous systems. *SIAM Journal of Control and Optimization*, 98(4):1288–1308, 2000.

[11] E. Gyurkovics. Receding horizon control via Bolza-time optimization. *Systems and Control Letters*, 35:195–200, 1998.

[12] E. Gyurkovics and A. M. Elaiw. Stabilization of smpled-data nonlinear systems by receding horizon control via discrete-time aproximations. *Automatica*, 40:2017–2028, 2004.

[13] L. Magni and R. Scattolini. Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Transactions on Automatic Control*, 49:900–906, 2004.

[14] L. Magni and R. Scattolini. Robustness and robust design of mpc for nonlinear discrete-time systems. In *Preprints of NMPC05 - International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control*, pages 31–46, IST, University of Stuttgart, August 26-30, 2005.

[15] H. Michalska and R. B. Vinter. Nonlinear stabilization using discontinuous moving-horizon control. *IMA Journal of Mathematical Control and Information*, 11:321–340, 1994.

[16] D. Nesić and A. Teel. A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. *IEEE Transactions on Automatic Control*, 49:1103–1034, 2004.

[17] J. E. Slotine and W. Li. *Applied Nonlinear Control.* Prentice Hall, New Jersey, 1991.

Appendix

Proof of Lemma 4

At a certain sampling instant t_i, we measure the current state of the plant x_{t_i} and we solve problem $\mathcal{P}(x_{t_i}, T_c, T_p)$ obtaining as solution the feedback strategy $\bar{k}$ to which corresponds, in the worst disturbance scenario, the trajectory $\bar{x}$ and control $\bar{u}$. The value of differential game $\mathcal{P}(x_{t_i}, T_c, T_p)$ is given by

$$V_{t_i}(t_i, x_{t_i}) = \int_{t_i}^{t_i+T_p} L(\bar{x}(s), \bar{u}(s))ds + W(\bar{x}(t_i + T_p)). \tag{13}$$

Consider now the family of problems $\mathcal{P}(x_t, T_c - (t - t_i), T_p - (t - t_i))$ for $t \in [t_i, t_i + \delta)$. These problems start at different instants t, but all terminate at the same instant $t_i + T_p$. Therefore in the worst disturbance scenario, by Bellman's principle of optimality we have that

$$V_{t_i}(t,\bar{x}(t)) = \int_{t}^{t_i+T_p} L(\bar{x}(s),\bar{u}(s))ds + W(\bar{x}(t_i+T_p)). \tag{14}$$

Suppose that the worst disturbance scenario did not occur and so, at time t, we are at state $x^*(t)$ which is, in general, distinct from $\bar{x}(t)$. Because such scenario is more favorable, and by the assumption on the existence of value to the differential game we have that

$$V_{t_i}(t,x^*(t)) \leq V_{t_i}(t,\bar{x}(t)) \qquad \text{for all } t \in [t_i, t_i+\delta). \tag{15}$$

We may remove the subscript t_i from the value function if we always choose the subscript t_i to be the sampling instant immediately before t, that is (recall that $\lfloor t \rfloor_\pi = \max_i\{t_i \in \pi : t_i \leq t\}$)

$$V(t,x) := V_{\lfloor t \rfloor_\pi}(t,x).$$

For simplicity define the function

$$V^*(t) = V(t,x^*(t)).$$

We show that $t \mapsto V^*(t)$ is decreasing in two situations:

(i) on each interval $[t_i, t_i+\delta)$

$$V^*(t) \leq V^*(t_i) - \int_{t_i}^{t} M(\bar{x}(s))\, ds \quad \text{for all } t \in [t_i, t_i+\delta), \text{ and all } i \geq 0;$$

(ii) from one interval to the other

$$V^*(t_i+\delta^+) \leq V^*(t_i+\delta^-) \quad \text{for all } i \geq 0;$$

therefore yielding the result.

(i) The first assertion is almost immediate from (15), (14) and (13).

$$\begin{aligned} V^*(t) &\leq V_{t_i}(t,\bar{x}(t)) \\ &= V_{t_i}(t_i,\bar{x}(t_i)) - \int_{t_i}^{t} L(\bar{x}(s),\bar{u}(s))\, ds \\ &= V_{t_i}(t_i,x^*(t_i)) - \int_{t_i}^{t} L(\bar{x}(s),\bar{u}(s))\, ds \\ &\leq V^*(t_i) - \int_{t_i}^{t} M(\bar{x}(s))\, ds \end{aligned}$$

for all $t \in [t_i, t_i+\delta)$, and all $i \geq 0$;

(ii) Let the pair $(\bar{x}, \bar{u})|_{[t_i+\delta^-, t_i+T_p]}$ be an optimal solution to $\mathcal{P}(x(t_i + \delta^-), T_c - \delta, T_p - \delta)$. Then

$$\begin{aligned} V^*(t_i + \delta^-) &= V_{t_i}(t_i + \delta^-, \bar{x}(t_i + \delta^-)) \\ &= \int\limits_{t_i+\delta}^{t_i+T_p} L(\bar{x}(s), \bar{u}(s))\, ds + W(\bar{x}(t_i + T_p)). \end{aligned}$$

Now, extend the process $(\bar{x}, \bar{u})$ to the interval $[t_i + \delta^-, t_i + T_p + \delta^-]$ using in the last δ seconds the auxiliary control law k^{aux}. This is an admissible (the set S is invariant under k^{aux} and is contained in X), suboptimal solution to $\mathcal{P}(x(t_i + \delta^+), T_c - \delta, T_p - \delta)$. Therefore

$$\begin{aligned} V^*(t_i + \delta^+) &= V_{t_{i+1}}(t_i + \delta^+, \bar{x}(t_i + \delta^+)) \\ &\le \int\limits_{t_i+\delta}^{t_i+T_p+\delta} L(\bar{x}(s), \bar{u}(s))\, ds + W(\bar{x}(t_i + T_p + \delta)) \\ &= V^*(t_i + \delta^-) + W(\bar{x}(t_i + T_p + \delta)) - W(\bar{x}(t_i + T_p)) \\ &\qquad + \int\limits_{t_i+T_p}^{t_i+T_p+\delta} L(\bar{x}(s), \bar{u}(s))\, ds. \end{aligned}$$

Using RSC5a in the interval $[t_i + T_p, t_i + T_p + \delta]$ we obtain

$$V^*(t_i + \delta^+) \le V^*(t_i + \delta^-) \quad \text{for all } i \ge 0$$

as required. □

On the Computation of Robust Control Invariant Sets for Piecewise Affine Systems

T. Alamo[1], M. Fiacchini[1], A. Cepeda[1], D. Limon[1], J.M. Bravo[2], and E.F. Camacho[1]

[1] Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Sevilla, Spain
{alamo, mirko, cepeda, limon, eduardo} @cartuja.us.es

[2] Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática. Universidad de Huelva, Huelva, Spain
caro@uhu.es

Summary. In this paper, an alternative approach to the computation of control invariant sets for piecewise affine systems is presented. Based on two approximation operators, two algorithms that provide outer and inner approximations of the maximal robust control invariant set are presented. These algorithms can be used to obtain a robust control invariant set for the system. An illustrative example is presented.

1 Introduction

In the context of nonlinear MPC, the stable and admissible closed-loop behavior is typically based on the addition of a terminal constraint and cost [1]. The terminal constraint is chosen to be an admissible robust control invariant set of the system. The size of this control invariant set determines, in many cases, the feasibility region of the nonlinear MPC controller [2]. It is shown in [2] that the domain of attraction of MPC controllers can be enlarged by means of a sequence of controllable (not necessarily invariant) sets.

The stability analysis of piecewise affine systems (PWA systems) plays an important role in the context of hybrid systems control. This is mainly due to the fact that piecewise affine systems can model a broad class of hybrid systems (see [3]). Therefore, it is of paramount relevance in the context of hybrid MPC the computation of controllable sets for this class of nonlinear systems [4].

The estimation of the domain of attraction of piecewise affine systems has been addressed by a number of authors. Quadratic (and piecewise quadratic) Lyapunov functions for hybrid systems have been proposed in [5, 6, 7]. A polyhedral approach is presented in [8]. Piecewise affine Lyapunov functions are considered in [9]. In [10] an algorithm to compute the maximal robust control invariant set for a PWA system is presented. Moreover, sufficient conditions to guarantee that the algorithm is finitely determined are also given.

It is well-known (see, for example, [8]) that the computation of the maximal robust control invariant set for a piecewise affine system requires such a computational burden that it is difficult to obtain it in an exact manner.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 131–139, 2007.
springerlink.com

In this paper we propose an algorithm that circumvents the huge computational complexity associated to the obtainment of the maximal robust control invariant set. Two new algorithms are proposed. The first one provides a convex polyhedral outer bound of the maximal control invariant set for the piecewise affine system. This outer estimation is used, by the second proposed algorithm, to obtain a robust control invariant set for the system (not necessarily the maximal one). The algorithms are based on inner and outer approximations of a given non-convex set.

The paper is organized as follows: Section 2 presents the problem statement. Section 3 presents an algorithm that computes an outer bound of the maximal robust control invariant set of the piecewise affine system. A procedure to obtain a robust control invariant set is proposed in section 4. An illustrative example is given in section 5. The paper draws to a close with a section of conclusions.

2 Problem Statement

Let us suppose that X is a bounded convex polyhedron. Suppose also that the convex polyhedra X_i, $i = 1, \ldots, r$, with disjoint interiors, form a partition of X. That is, $X = \bigcup\limits_{i=1}^{r} X_i$.

We consider the following piecewise affine system:

$$x^+ = f(x,u,w) = A_i x + B_i u + E_i w + q_i \ \text{ if } \ x \in X_i \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ is the state vector; x^+ denotes the successor state; $u \in U = \{ \, u \in \mathbb{R}^{n_u} \ : \ \|u\|_\infty \leq u_{max} \, \}$ is the control input; w denotes a bounded additive uncertainty: $w \in W = \{ \, w \in \mathbb{R}^{n_w} \ : \ \|w\|_\infty \leq \epsilon \, \}$.

In order to present the results of this paper it is important to refer to the notion of the one step set [8].

Definition 1 (one step set). *Given a region Ω, and system (1), the following sets are defined:*

$$Q(\Omega) = \{ \, x \in X \ : \ \textit{there is } \ u \in U \textit{ such that } f(x,u,w) \in \Omega, \forall w \in W \, \}$$

$$Q_i(\Omega) = \{ \, x \in X_i \ : \textit{there is } u \in U \textit{ such that } A_i x + B_i u + E_i w + q_i \in \Omega, \forall w \in W \}$$

The following well-known properties allow us to compute $Q(\Omega)$ for a piecewise affine system [8]:

Property 1. Given a convex polyhedron Ω: $Q(\Omega) = \bigcup\limits_{i=1}^{r} Q_i(\Omega)$, where $Q_i(\Omega)$, $i = 1, \ldots, r$ are polyhedra.

Property 2. If $\Omega = \bigcup\limits_{j=1}^{s} P_j$ and $P_1, P_2, \ldots, P_s$ are convex polyhedra, then $Q(\Omega) = \bigcup\limits_{i=1}^{r} \bigcup\limits_{j=1}^{s} Q_i(P_j)$, where $Q_i(P_j)$, $i = 1, \ldots, r$, $j = 1, \ldots, s$ are convex polyhedra.

Based on these definitions and properties, the maximal robust control invariant set can be obtained by means of the following algorithm [8]:

Algorithm 1

(i) Set the initial region C_0 equal to X.
(ii) $C_{k+1} = Q(C_k)$.
(iii) If $C_{k+1} = C_k$ then $C_k = C_\infty$. Stop. Else, set $k = k + 1$ and return to step (ii).

Note that the evolution of any initial condition belonging to set C_k can be robustly maintained in X at least k sample times. Therefore, $C_\infty = \lim\limits_{k\to\infty} C_k$ constitutes the set of initial condition for which the system is robustly controllable in an admissible way. That is, C_∞ is the maximal robust control invariant set.

Suppose that algorithm 1 converges to C_∞ in k_d steps. Then, applying property 2 in a recursive way it is possible to state that $C_{k_d} = C_\infty$ can be represented by means of the union of r^{k_d} convex polyhedra. This worst-case estimation of the number of convex polyhedra required to represent C_∞ clearly shows that the exact computation of C_∞ for a piecewise-affine system might not be possible in the general case: the number of sets required to represent the maximal robust control invariant set grows in an exponential way with the number of iterations of algorithm 1. Even in the case that algorithm 1 obtains (theoretically) the maximal robust control invariant set in a finite number of steps, the complexity of the representation might make it impossible to run the algorithm beyond a reduced number of steps (normally insufficient to attain the greatest domain of attraction). Therefore, it is compulsory to consider approximated approaches to the computation of C_∞.

In this paper we propose an algorithm (based on convex outer (and inner) approximations of the one step set) that can be used to compute a convex robust control invariant set for the piecewise affine system.

3 Outer Bound of the Maximal Robust Control Invariant Set

One of the objectives of this paper consists in providing a procedure to obtain a convex outer approximation of C_∞ for a piecewise affine system. This outer bound has a number of practical and relevant applications:

(i) It captures the geometry of C_∞ and makes the computation of a robust control invariant set for the system easier (this use is explored in section 4). Moreover, it can be used as the initial set in algorithm 1. If the outer bound is a good approximation, then algorithm 1 might require an (implementable) reduced number of iterations.
(ii) The constraints that define the outer bound can be included as hard constraints in a hybrid MPC scheme. Moreover, the inclusion of the aforementioned constraints can be used to improve the convex relaxations of the nonlinear optimization problems that appear in the context of hybrid MPC.

(iii) The outer bound can be used as a measure of the controllable region of a hybrid system. This can be used in the design of the hybrid system itself.
(iv) The obtained convex region can be also used to induce a control Lyapunov function.

The following algorithm provides a convex polyhedron that serves as an outer bound of the maximal robust control invariant set of a piecewise affine system:

Algorithm 2

(i) $k = 0$, $\hat{C}_0 = X$.
(ii) $\hat{C}_{k+1} = Co\{\bigcup\limits_{i=1}^{r} Q_i(\hat{C}_k)\} \bigcap \hat{C}_k$
(iii) If $\hat{C}_{k+1} = \hat{C}_k$, then Stop. Else, k=k+1, Go to step (ii).

Remark 1. Note that the convex hull operator (Co) required to implement the algorithm can be substituted by any outer approximation of the convex hull. For example, the envelope operator of [11], or the outer approximation provided in [12]. The algorithm can be stopped when there is no significant improvement of the outer bound. That is, when $\hat{C}_k$ is almost identical to $\hat{C}_{k-1}$. For example, the algorithm could be stopped when $(1 - \epsilon_s)\hat{C}_k \subseteq \hat{C}_{k+1}$, where $\epsilon_s > 0$ is a arbitrarily small tuning parameter.

Property 3. Each one of the polyhedrons $\hat{C}_k$ obtained by means of algorithm 2 constitutes an outer bound of the maximal robust control invariant set of the piecewise affine system. That is, $C_\infty \subseteq \hat{C}_k$, for all $k \geq 0$.

Proof
It is clear that $C_\infty \subseteq X = \hat{C}_0$. To prove the property it suffices to show that $C_\infty \subseteq \hat{C}_k$ implies $C_\infty \subseteq \hat{C}_{k+1}$, for every $k \geq 0$. Suppose that $C_\infty \subseteq \hat{C}_k$:

$$C_\infty = C_\infty \bigcap \hat{C}_k = Q(C_\infty) \bigcap \hat{C}_k \subseteq Q(\hat{C}_k) \bigcap \hat{C}_k = \left(\bigcup_{i=1}^{r} Q_i(\hat{C}_k) \right) \bigcap \hat{C}_k \subseteq \hat{C}_{k+1}$$

4 Inner Approximation of the Maximal Robust Control Invariant Set

In this section, an algorithm that computes an inner approximation of C_∞ is presented. Such an algorithm is based on the complementary set of the one step operator and on the notion of inner supporting constraint.

4.1 Complementary Set of $Q(\Omega)$

The inner approximations of C_∞ presented in this paper for $Q(\Omega)$ rely on the computation of the complementary set of $Q(\Omega)$:

Definition 2. *Given set Ω, $Q^c(\Omega)$ denotes the complementary set of $Q(\Omega)$ in X. That is, $Q^c(\Omega) = \{ x \in X \ : \ x \notin Q(\Omega) \}$. Given set Ω, $Q_i^c(\Omega)$ denotes the complementary set of $Q_i(\Omega)$ in X_i. That is, $Q_i^c(\Omega) = \{ x \in X_i \ : \ x \notin Q_i(\Omega) \}$.*

It is inferred from the previous definition that given set Ω: $Q^c(\Omega) = \bigcup\limits_{i=1}^{r} Q_i^c(\Omega)$. Therefore, in order to compute $Q^c(\Omega)$, it suffices to compute $Q_i^c(\Omega)$, $i = 1, \ldots, r$. The following property (see [13] for a proof) shows how to compute $Q_i^c(\Omega)$.

Property 4. Suppose that $Q_i(\Omega) = \{ x \in X_i \ : \ G_i x \leq g_i \}$, where $G_i \in \mathbb{R}^{L_i \times n_x}$ and $g_i \in \mathbb{R}^{L_i}$. Then,

$$Q_i^c(\Omega) = \bigcup_{j=1}^{L_i} S_{i,j}(G_i, g_i)$$

where $S_{i,j}(G_i, g_i) = \{ x \in X_i \ : \ G_i(j)x > g_i(j), G_i(l)x \leq g_i(l) \text{ for } l = 1, \ldots, j-1 \}$.

4.2 Inner Supporting Constraint

The construction of the proposed inner approximation of C_∞ is based on the notion of inner supporting constraint:

Definition 3. *Suppose that S is a compact convex set that does not contain the origin and that R is a bounded set. We say that $\{ x \ : \ c^\top x \leq 1 \}$ is an inner supporting constraint of S over R if c is the solution of the following minimization problem*

$$\begin{aligned} \min_{c,\rho} \ & \rho \\ s.t. \ & c^\top x > 1, \forall x \in S \\ & c^\top x \leq 1, \forall x \in \frac{1}{\rho} R \end{aligned}$$

The following property allows us to compute an inner supporting constraint of a polyhedron S over $Q(\Omega) = \bigcup\limits_{i=1}^{r} T_i$ (where $T_i = Q_i(\Omega)$, $i = 1, \ldots, r$) by means of the solution of a linear optimization problem. The proof of the property is similar to the proof of an analogous property in [12] and it is omitted because of space limitations.

Property 5. Consider polyhedron $S = \{ x \ : \ Fx \leq f \}$ and the polyhedrons $T_l = \{ x \ : \ M_l x \leq m_l \}$, $l = 1, \ldots, r$. Suppose that the scalar ρ and the vectors c, λ, and β_l, $l = 1, \ldots, r$ satisfy the following constraints:

$$\rho > 0 \tag{2}$$

$$\lambda \geq 0 \tag{3}$$

$$\beta_l \geq 0, \ \ l = 1, \ldots, r \tag{4}$$

$$1 + f^\top \lambda < 0 \tag{5}$$
$$-\rho + m_l^\top \beta_l \leq 0, \quad l = 1, \dots, r \tag{6}$$
$$c + F^\top \lambda = 0 \tag{7}$$
$$c - M_l^\top \beta_l = 0, \quad l = 1, \dots, r \tag{8}$$

then

$$c^\top x > 1, \quad \forall x \in S \tag{9}$$
$$c^\top x \leq 1, \quad \forall x \in \bigcup_{l=1}^{r} \frac{1}{\rho} T_l \tag{10}$$

4.3 Robust Control Invariant Set: Proposed Algorithm

The following algorithm serves to compute a robust control invariant set for a piecewise affine system:

Algorithm 3

(i) Set $k = 0$ and choose a contracting factor $\tilde{\lambda} \in (0, 1)$.
(ii) Make $\tilde{C}_0$ equal to the outer approximation of C_∞ obtained by means of algorithm 2.
(iii) Given $\tilde{C}_k = \{ \, x \, : \, Hx \leq h \, \}$, obtain $T_i = Q_i(\tilde{\lambda}\tilde{C}_k)$, $i = 1, \dots, r$.
(iv) Obtain $Q^c(\tilde{\lambda}\tilde{C}_k) = \bigcup\limits_{i=1}^{r} Q_i^c(\tilde{\lambda}\tilde{C}_k) = \bigcup\limits_{j=1}^{n_c} S_j$ by means of property 4.
(v) For every $j = 1, \dots, n_c$ obtain $\{ \, x \, : \, c_j^\top x \leq 1 \, \}$, the inner supporting constraint of S_j over $Q(\tilde{\lambda}\tilde{C}_k) = \bigcup\limits_{i=1}^{r} T_i$. This can be achieved by means of property 5.
(vi) Make $\tilde{C}_{k+1} = \bigcap\limits_{j=1}^{n_c} \{ \, x \, : \, c_j^\top x \leq 1 \, \}$.
(vii) If $\tilde{C}_{k+1} \subseteq Q(\tilde{C}_{k+1})$ then $\tilde{C}_{k+1}$ is a robust control invariant set. Stop. Else, set $k = k + 1$ and go to step (iii).

Bearing in mind the λ contractive procedure of [14], a contracting factor $\tilde{\lambda} \in (0, 1)$ has been included in the algorithm. Note that algorithm 3 finishes only if $\tilde{C}_{k+1} \subseteq Q(\tilde{C}_{k+1})$. In virtue of the geometrical condition of robust invariance [14], it is inferred that $\tilde{C}_k$ is a robust control invariant set. That is, if algorithm 3 finishes then a robust control invariant set is obtained. This set serves as an inner approximation of C_∞. Due to the approximate nature of the algorithm, it is not guaranteed that algorithm 3 converges to a robust control invariant set. Note, however, that it can be shown that each one of the obtained sets $\tilde{C}_k$ constitutes an inner approximation of C_k. The proof of this statement is based on the fact that, by definition of inner supporting constraint, $S_j \bigcap \{ \ x \ : \ c_j^\top x \leq 1 \ \}$ equals the empty set. Thus,

$$
\begin{aligned}
&Q^c(\tilde{\lambda}\tilde{C}_k) \bigcap \tilde{C}_{k+1} = Q^c(\tilde{\lambda}\tilde{C}_k) \bigcap (\bigcap_{m=1}^{n_c} \{ \ x \ : \ c_m^\top x \leq 1 \ \}) \\
&= (\bigcup_{j=1}^{n_c} S_j) \bigcap (\bigcap_{m=1}^{n_c} \{ \ x \ : \ c_m^\top x \leq 1 \ \}) = \bigcup_{j=1}^{n_c} \left(S_j \bigcap (\bigcap_{m=1}^{n_c} \{ \ x \ : \ c_m^\top x \leq 1 \ \}) \right) \\
&\subseteq \bigcup_{j=1}^{n_c} (S_j \bigcap \{ \ x \ : \ c_j^\top x \leq 1 \ \}) = \emptyset.
\end{aligned}
$$

That is, $Q^c(\tilde{\lambda}\tilde{C}_k) \bigcap \tilde{C}_{k+1} = \emptyset$. This implies that $\tilde{C}_{k+1} \subseteq Q(\tilde{\lambda}\tilde{C}_k)$.

5 Numerical Example

In this example, region $X = \{ \ x \ : \ \|x\|_\infty \leq 15 \ \}$ is subdivided into the subregions X_1, X_2 and X_3. These subregions are defined as follows:

$$
\begin{aligned}
X_1 &= \{ \ x \in X \ : \ x_1 - x_2 \leq 0 \ \} \\
X_2 &= \{ \ x \in X \ : \ x_1 - x_2 > 0 \ \text{ and } \ x_1 + x_2 \geq 0 \ \} \\
X_3 &= \{ \ x \in X \ : \ x_1 - x_2 > 0 \ \text{ and } \ x_1 + x_2 < 0 \ \}
\end{aligned}
$$

Consider the following piecewise affine system:

$$
x^+ = \begin{cases}
\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w & \text{if } x \in X_1 \\
\begin{bmatrix} 1 & 1 \\ 0.5 & 1.5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w & \text{if } x \in X_2 \\
\begin{bmatrix} 1 & -0.5 \\ 0 & 1.5 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1.5 \end{bmatrix} u + \begin{bmatrix} 1 \\ 0 \end{bmatrix} w & \text{if } x \in X_3
\end{cases}
$$

In this example it is assumed that $U = \{ \ u \in \mathbb{R} \ : \ \|u\|_\infty \leq 2 \ \}$ and $W = \{ \ w \in \mathbb{R} \ : \ \|w\|_\infty \leq 0.1 \ \}$. The contracting factor for algorithm 3 has been set equal to 0.95. In figure 1 the sequence of outer bounds $\hat{C}_k$ is displayed (the outer approximation of the convex hull operator presented in [12] has been used). The most inner polyhedron is used in algorithm 3 as initial guess to obtain a robust control invariant set. In figure 2 a sequence of sets $\tilde{C}_k$ leading to a robust control invariant set is displayed. The most inner polyhedron is a robust control invariant set for the piecewise affine system.

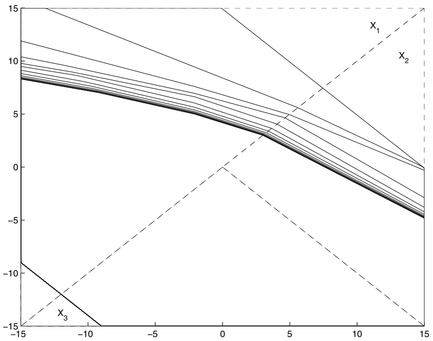


Fig. 1. Sequence of outer bounds $\hat{C}_k$ leading to an outer approximation of C_∞

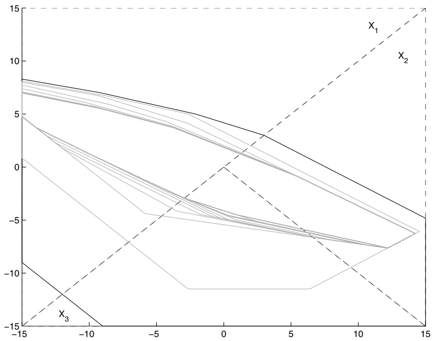


Fig. 2. Sequence of sets $\tilde{C}_k$ leading to a robust control invariant set

6 Conclusions

In this paper, an algorithm that provides an outer approximation of C_∞ is presented. This outer approximation has a number of practical and relevant applications. Based on the outer approximation of C_∞, an algorithm that provides an inner approximation of C_∞ is given. This algorithm can be used to obtain a robust control invariant set for a piecewise affine system. An illustrative example is presented.

References

[1] Mayne, D.Q., Rawlings, J.B., Rao, C.V. and Scokaert, P.O.M. "Constrained model predictive control: Stability and optimality". *Automatica*, **36**:789–814, (2000).

[2] Limon, D., Alamo, T. and Camacho, E.F. "Enlarging the domain of attraction of MPC controllers". *Automatica*, **41**(4):629–635, (2005).

[3] Heemels,W., De Schutter, B. and Bemporad, A., "Equivalence of hybrid dynamical models". *Automatica*, **37**:1085–1091, (2001).

[4] Lazar, M., Heemels, W.P.M.H., Weiland, S. and Bemporad, A. "Stabilization conditions for model predictive control of constrained PWA systems". *Proceedings of the 43rd IEEE Conference on Decision and Control.* 4595-4600, (2004).

[5] Johansson, M. and Rantzer, A., "Computation of piecewise quadratic lyapunov functions for hybrid systems". *IEEE Transactions on Automatic Control*, **34**(4):555–559, (1998).

[6] Mignone, D., Ferrari-Trecate, G., and Morari, M. "Stability and Stabilization of Piecewise Affine and Hybrid Systems: An LMI Approach. *Proceedings of the 39th IEEE Conference on Decision and Control.* 504-509, (2000).

[7] Rodrigues, Luis. "Stability analysis of piecewise-affine systems using controlled invariant sets". *Systems and Control Letters*, **53**(2):157-169, (2004).

[8] Kerrigan, E.C. " Robust Constraint Satisfaction: Invariant Sets and Predictive Control". *PhD thesis, University of Cambridge*, (2000).

[9] Grieder, P., Kvanisca, M., Baotic, M. and Morari, M. ”Stabilizing low complexity feedback control of constrained piecewise affine systems”. *Automatica*, **41**:1683–1694, (2005).

[10] Rakovic, S.V., Grieder, P., Kvasnica, M., Mayne, D.Q. and Morari, M. ”Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances”. *Proceedings of the 43rd IEEE Conference on Decision and Control.* 1418-1423, (2004).

[11] Bemporad, A., Fukuda, K. and Torrisi, F. ”Convexity recognition of the union of polyhedra”. *Computational Geometry*, **18**, 141-154, (2001).

[12] Alamo, T., Cepeda, A., Limon, D., Bravo, J.M., Fiacchini, M. and Camacho, E.F. ”On the computation of robust control invariant sets for piecewise affine systems” *Workshop on Assessment and Future Directions on NMPC. Freudentadt-Lauterbad, Germany, August 26-30*(2005).

[13] Bemporad, A., Morari, M., Dua, V. and Pistikopoulos, E.N. ”The Explicit Linear Quadratic Regulator for Constrained Systems”. *Automatica*, **38**(1), 3-20, (2002).

[14] Blanchini F., ”Set invariance in control”. *Automatica*, **35**:1747–1767, (1999).

Nonlinear Predictive Control of Irregularly Sampled Data Systems Using Identified Observers

Meka Srinivasarao[1], Sachin C. Patwardhan[2], and R.D. Gudi[3]

Department of Chemical Engineering, Indian Institute of Technology, Bombay, Mumbai, India
sachinp@che.iitb.ac.in

1 Introduction

In many practical situations in process industry, the measurements of process quality variables, such as product concentrations, are available at different sampling rates and than other measured variables and also at irregular sampling intervals. Thus, from the process control viewpoint, multi-rate systems in which measurements are available at slow and/or differing rates and in which the manipulations are updated at relatively fast rate are of particular interest.

In recent years, there has been considerable amount of interest in developing control schemes for multi-rate sampled data systems. A key component of these control schemes is a fast rate model, which is seldom available in practice. As a consequence, the research focus has shifted to the development of fast rate models from the multi-rate sampled data in the recent years [1]. However, the research efforts in this area have been limited to the development of linear models and multi-rate nonlinear system identification has received very little attention.

Many key unit operations in chemical plants exhibit strongly nonlinear behavior when operated over a wide range. Bequette et al [2] proposed a multi-rate nonlinear MPC (NMPC) formulation based on multi-rate extended Kalman filter. Their approach, however, uses a nonlinear model derived from first principles. The development of models from first principles can be a difficult and time consuming process particularly in a multi-rate scenario. Moreover, in the mechanistic model based approach, the noise model parameters are typically used as tuning knobs and this can result in sub-optimal state estimation and poor regulatory control. It is to be noted that, in the presence of significant unmeasured disturbances, reasonably accurate stochastic models are necessary for developing an effective inter-sample estimation strategy. Thus, to achieve good servo and regulatory control of nonlinear multi-rate sampled data systems, it is important to identify nonlinear models directly from multi-rate input-output data.

This work aims at the identification of a nonlinear fast rate model and a nonlinear multi-rate time varying state observer from irregularly sampled (multi-rate) data, which is corrupted with unmeasured disturbances and measurement noise. The deterministic and stochastic components of the proposed model have

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 141–149, 2007.
springerlink.com

Weiner structure. The linear dynamic component of these models is parameterized using generalized orthonormal basis filters (GOBF) [3, 4]. We then proceed to show how the identified models can be used for inter-sample inferential estimation of the slowly sampled variable and also for model predictive control of such irregularly sampled, multi-rate systems. The efficacy of the proposed modeling and control scheme is demonstrated by conducting simulation studies on a benchmark CSTR system [5] which exhibits input multiplicity and change in the sign of steady state gain in the operating region.

2 Development of Multi-rate NOE+NARMA Model

In this work, we make following assumptions: (a) Sampling rates for all measurements are integer multiples of some time period called 'shortest time unit' (T).(b) All actuators are to be manipulated at a frequency corresponding to the 'shortest time unit' (T).and (c) The effect of unmeasured disturbances on the outputs can be adequately captured by an additive nonlinear noise model. Thus, the manipulated inputs are changed at $\{t_k = kT : k = 0, 1, 2, ...\}$ while the ith output measurements are assumed to be available only at sampling instants given by the sub-sequence $\{k_{i0}, k_{i1}, k_{i2,}...\}$ such that the difference $k_{il} - k_{il-1} = q_{il}$ (> 1) where q_{il} is an integer.

We propose to use nonlinear output error (NOE) structure to model the deterministic component in the data. This choice is motivated by the fact that the internal model in an NMPC formulation is required to have good prediction ability with respect to the manipulated inputs. It can be argued that the NOE models, which are driven only by the manipulated inputs, have good long range prediction ability. The choice of NOE structure implies that the deterministic component of a $r \times m$ MIMO system can be modeled as r MISO NOE models. The irregularly sampled model residuals generated while identifying the deterministic component are further modeled as r SISO nonlinear stochastic models. Thus, in order to simplify the notation, without loss of generality, we carry out the model development for a MISO system. Therefore in the sequel we drop the subscript 'i' that we used above to indicate variables associated with i'th output . Given input sequence $\{\mathbf{u}(k) : k = 0, 1, 2, ...N\}$ and the corresponding irregularly sampled output data $\{y(k_l) : k_l = k_0, k_1, k_2, ...\}$ collected from a plant where k_l represents sampling instants. We propose a two step approach to the development of deterministic and stochastic models for a general multivariate, nonlinear system with fading memory. In the first step, we develop a MISO fast rate Weiner type NOE model of the form,

$$\mathbf{X}_u(k+1) = \Phi_u \mathbf{X}_u(k) + \Gamma_u \mathbf{u}(k) \tag{1}$$

$$\widehat{y}_u(k) = \Omega_u \left[\mathbf{X}_u(k) \right] \tag{2}$$

$$\text{for } k_l \leq k < k_{l+1} \tag{3}$$

using the fast sampled inputs and the slowly sampled outputs. Here, $\Omega_u(.) : R^{n_u} \to R$ represents a nonlinear state-output map and $\widehat{y}_u(k)$ represents the estimated output at instant k based on input information alone. At the sampling instant (k_l), we have

$$y(k_l) = \widehat{y}_u(k_l) + v(k_l) \tag{4}$$

In the second step, the residual sequence $\{v(k_l)\}$ generated at the slow rate, under the assumption that it is uncorrelated with the input sequence, can be used to develop a SISO disturbance model. Figure (1) shows a schematic representation of the total model. We propose to develop a Nonlinear Auto-Regressive (NAR) model of the form:

$$\mathbf{x}_v(k_{l+1}) = \varPhi_v\,(k_{l+1}, k_l)\ \mathbf{x}_v(k_l) + \varGamma_v\,(k_{l+1}, k_l)\ \ v(k_l) \tag{5}$$

$$v(k_l) = \varOmega_v\left[\mathbf{x}_v(k_l)\right] + e(k_l) \tag{6}$$

where $e(k_l)$ represents a white noise sequence. The NAR model can be rearranged into a NARMA form by simple rearrangement as follows

$$\mathbf{x}_v(k_{l+1}) = \mathcal{F}\left[k_{l+1}, k_l, \mathbf{x}_v(k_l)\right] + \varGamma_v\,(k_{l+1}, k_l)\, e(k_l) \tag{7}$$

$$\mathcal{F}\left[k_{l+1}, k_l, \mathbf{x}_v(k_l)\right] = \left[\varPhi_v\,(k_{l+1}, k_l)\ \mathbf{x}_v(k_l) + \varGamma_v\,(k_{l+1}, k_l)\ \ \varOmega_v\,(\,\mathbf{x}_v(k_l))\right] \tag{8}$$

The combined output of NOE+NARMA model in this case becomes

$$y(k_l) = \varOmega_u\left[\mathbf{X}_u(k_l)\right] + \varOmega_v\left[\mathbf{x}_v(k_l)\right] + e(k_l) \tag{9}$$

The linear dynamic component and the static nonlinear maps in the proposed general form proposed above can be selected in variety of ways . In this work, we choose to parameterize the linear dynamic component (5) using Generalized Orthonormal Basis Filters. While working with multi-rate systems, it is convenient to parameterize GOBF pole vector (ξ) through equivalent continuous time pole locations. In case of irregularly sampled system, a continuous time pole p_j maps to a time varying discrete time pole ξ_j as follows

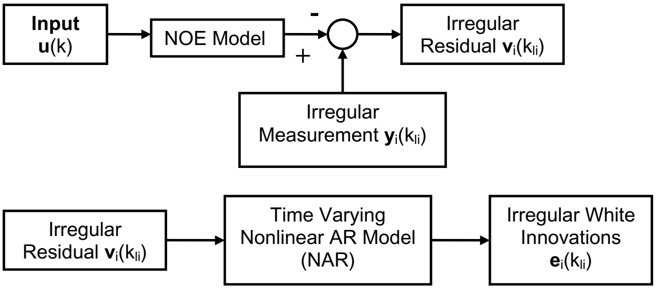


Fig. 1. Multi-rate Model: Schematic Representation

$$\xi_j(k_{l+1}, k_l) = \exp\left[p_j T\,(k_{l+1} - k_l)\right] \tag{10}$$

where $T\left(k_{l+1}-k_l\right)$ represents the irregular sampling interval. A time varying discrete state realization of the form

$$\mathbf{x}(k_{l+1}) = \varPhi\left(\mathbf{p},k_{l+1},k_l\right)\ \mathbf{x}(k_l) + \varGamma\left(\mathbf{p},k_{l+1},k_l\right)\ \upsilon(k_l) \tag{11}$$

can be generated using $\xi_j(k_{l+1},k_l)$ and by following the procedure given by Patwardhan and Shah [3] for state realization. Note that the requirement $|\xi_j| < 1$ translates to $Re\left[p_j\right] < 0$, where $Re\left[.\right]$ represents real part of continuous time pole $p_j.$

The nonlinear state output maps $\varOmega_u[.] : R^{n_u} \to R$ is parameterized as a quadratic polynomial function. Given GOBF pole vector $\mathbf{p}$, the parameter of the state output map can be estimated using linear regression. Thus, the GOBF poles and parameters of state output map are estimated using a two tier nested optimization procedure (see[3, 6] for details). The parameter estimation procedure for NOE component yields the residual sequence $\left\{\widehat{\upsilon}(k_l) : k_l = k_1, k_2, ...k_N\right\}.$ The parameters of the NAR model can be estimated by applying similar nested optimization procedure. The main advantage of using NAR structure is that the state sequence can be generated without requiring knowledge of innovation sequence.

3 Nonlinear Predictive Control Formulation

The model identification exercise described above yields r multi-rate NOE +NARMA type MISO state space observers. These observers can be used for the current state estimation whenever the measurement becomes available for the i'th output as follows

$$\mathbf{X}_u^{(i)}(k_{il+1}) = \mathbf{\Phi}_u^{(i)}\,\mathbf{X}_u^{(i)}(k_{il+1}-1) + \mathbf{\Gamma}_u^{(i)}\mathbf{u}(k_{il+1}-1) \tag{12}$$

$$\mathbf{x}_v(k_{il+1}) = \mathcal{F}^{(i)}\left[k_{il+1}, k_{il}, \mathbf{x}_v^{(i)}(k_{il})\right] + \varGamma_v^{(i)}\left(k_{il+1}, k_{il}\right)\mathbf{e}_i(k_{il}) \tag{13}$$

$$\mathbf{y}_i(k_{il}) = \varOmega_u^{(i)}\left[\mathbf{X}_u^{(i)}(k_{il})\right] + \varOmega_v^{(i)}\left[\mathbf{x}_v^{(i)}(k_{il})\right] + \mathbf{e}_i(k_{il}) \tag{14}$$

where $\mathbf{y} \in R^r$ and $\mathbf{e} \in R^r$ represents measurement and innovation vectors, respectively. The superscript (i) in the above set of equations represents the i'th MISO observer corresponding to the i'th output. An important consideration from control view-point in any multi-rate scenario is the ability of the model to carry out fast rate inter-sample prediction of the irregularly sampled variable. The inter-sample predictions can be generated as follows

$$\widehat{\mathbf{X}}_u^{(i)}(k|k-1) = \varPhi_u\widehat{\mathbf{X}}_u^{(i)}(k-1|k-2) + \varGamma_u\mathbf{u}(k-1) \tag{15}$$

$$\widehat{\mathbf{x}}_v^{(i)}(k|k_{il}) = \mathcal{F}\left[k, k_l, \widehat{\mathbf{x}}_v^{(i)}(k|k_{il})\right] + \varGamma_v\left(k, k_l\right)\mathbf{e}_i(k_l) \tag{16}$$

$$\mathcal{F}\left[k, k_l, \widehat{\mathbf{x}}_v^{(i)}(k|k_{il})\right] = \left[\varPhi_v\left(k, k_l\right)\widehat{\mathbf{x}}_v^{(i)}(k_{il}) + \varGamma_v\left(k, k_l\right)\ \varOmega_v\left(\widehat{\mathbf{x}}_v^{(i)}(k_{il})\right)\right]$$

$$\widehat{\mathbf{y}}_i(k|k_l) = \varOmega_u^{(i)}\left[\widehat{\mathbf{X}}_u^{(i)}(k|k-1)\right] + \varOmega_v^{(i)}\left[\widehat{\mathbf{x}}_v^{(i)}(k|k_{il})\right] \tag{17}$$

for $k_l \leq k < k_{l+1}$. Here matrices $\Phi_v^{(i)}\,(k,k_l)$ and $\Gamma_v^{(i)}\,(k,k_l)$ are recomputed at each minor sampling instant k. In the multi-rate system under consideration, the control action is taken at every minor sampling instant k. At the k'th instant, given a set of future manipulated input sequence $\{\mathbf{u}(k|k), \mathbf{u}(k+1|k), ...\mathbf{u}(k+p-1|k)\}$, future predictions for the i'th output over a prediction horizon of p can be generated as follows

$$\widehat{\mathbf{X}}_u^{(i)}(k+j+1|k) = \Phi_u \widehat{\mathbf{X}}_u^{(i)}(k+j|k) + \Gamma_u \mathbf{u}(k+j|k) \tag{18}$$

$$\begin{aligned}\widehat{\mathbf{x}}_v^{(i)}(k+j|k_{il}) = {} & F^{(i)}\left[k+j, k_{il}, \widehat{\mathbf{x}}_v^{(i)}(k_{il})\right] \\ & + \Gamma_v\,(k+j,k_{il})\,\mathbf{d}_i(k+j|k)\end{aligned} \tag{19}$$

$$\begin{aligned}\widehat{\mathbf{y}}_i(k+j|k) = {} & \Omega_u\left[\widehat{\mathbf{X}}_u^{(i)}(k+j|k)\right] \\ & + \Omega_v\left[\widehat{\mathbf{x}}_v(k+j|k_{il})\right] + \mathbf{d}_i(k+j|k)\end{aligned} \tag{20}$$

$$\mathbf{d}_i(k+j+1|k) = \mathbf{d}_i(k+j|k);\ \mathbf{d}_i(k|k) = \mathbf{e}_i(k_{il});\ j = 1,2,.....p \tag{21}$$

Here, k_{il} represents the last major sampling instant at which the measurement was available for the i'th output. Note that the perdition of deterministic and stochastic components is carried out by different approaches.

Given a future set-point trajectory $\{\mathbf{r}(k+j|k) : j = 1,2,....p\}$, we define the future prediction error vector $\mathbf{e}_f(k+i|k)$ as

$$\mathbf{e}_f(k+i|k) = \mathbf{r}(k+i|k) - \widehat{\mathbf{y}}(k+i|k) \tag{22}$$

The nonlinear model predictive control problem at the sampling instant k is defined as follows

$$\begin{aligned}\min_{\mathbf{u}(k|k),...\mathbf{u}(k+m_{q-1}|k)} & \sum_{i=1}^{p} \mathbf{e}_f(k+i|k)^T \mathbf{W}_E \mathbf{e}_f(k+i|k) \\ + & \sum_{i=1}^{q-1} \left\{\Delta\mathbf{u}(k+i|k)^T\, \mathbf{W}_U\, \Delta\mathbf{u}(k+i|k)\right\}\end{aligned} \tag{23}$$

subject to the following constraints

$$\mathbf{u}^L \leq \mathbf{u}(k+i|k) \leq \mathbf{u}^H \quad for \ \ i = 0,1,..,q-1 \tag{24}$$

$$\Delta\mathbf{u}^L \leq \Delta\mathbf{u}(k+i|k) \leq \Delta\mathbf{u}^H \quad for \ \ i = 0,1,..,q-1 \tag{25}$$

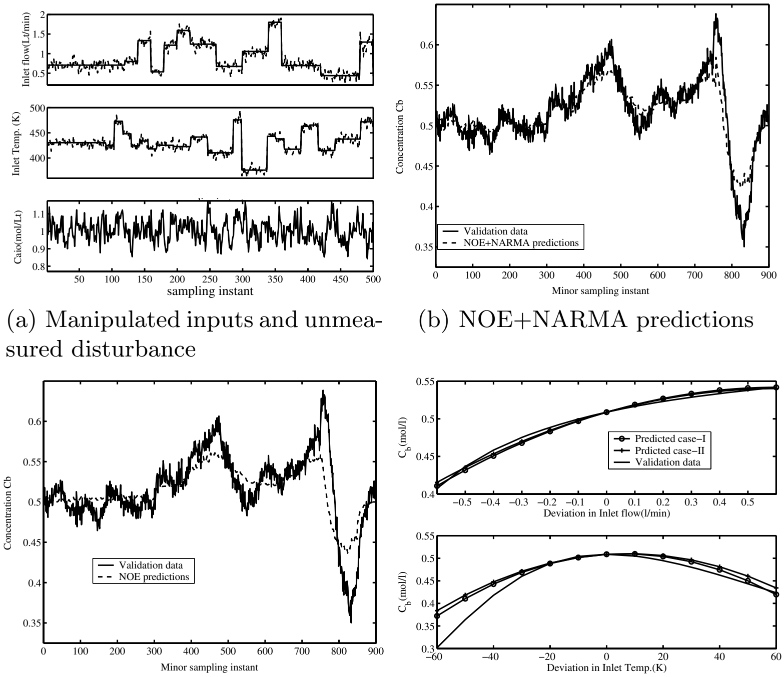
The resulting constrained optimization problem can be solved using any nonlinear programming technique. The controller is implemented in a moving horizon framework.

4 Simulation Studies

In this section, simulation studies are presented to demonstrate the ability of proposed modeling scheme to capture the dynamics of a system exhibiting input multiplicities. We then proceed to demonstrate that the identified models can be used to achieve satisfactory servo and regulatory control at the optimum operating point of the system. The performance of the identified models is evaluated using the following statistical criterion a)Percentage Prediction Error (PPE) and b)Percentage Estimation Error (PEE) (Srinivasarao et al. ([6])). The PPE values are computed with using multi-rate measurements while the PEE values are computed using noise free fast rate outputs of the process (at minor sampling instants) obtained from simulations. The PEE values are indicative of inter-sample prediction ability of the models while the PPE quantify long range prediction capability of the models.

The system under consideration consists of a CSTR in which a reversible exothermic reaction $A \rightleftharpoons B$ is carried out. The model equations and the nominal parameters used in the simulation can be found in Li and Biegler ([5]). This system exhibits input multiplicity and change in the sign of steady state gain in the operating region. The controlled variables are reactor level (L) and reactor concentration (C_b) while inlet flow rate (F) and feed inlet temperature (T_i) are used as manipulated variables. In the present work, reactor level was assumed to be regularly sampled measurement at the base sampling interval ($T = 0.1$ min.) where as the output concentration (C_b) in the CSTR was assumed to be sampled at irregular sampling intervals varying between 5 to 15 times that of base sampling interval. Also, the manipulated input moves were computed at the fast rate after every 0.1 min. interval. The inlet feed stream contains only A and its concentration C_{ai} was assumed to fluctuate according to a stochastic process. The concentration fluctuations were generated by subjecting a discrete unity gain first order filter (with pole at 0.95) to a zero mean white noise sequence of standard deviation 0.02. Note that $C_{ai}(k)$ was assumed to be a piecewise constant function during simulations. In addition, it was assumed that manipulated inputs are subject to unknown fluctuations. These fluctuations in the inlet flow rate and the feed inlet temperature were generated by subjecting discrete unity gain filters (with poles at 0.85) to zero mean white noise sequences with covariance $2.5X10^{-3}$and 10.24, respectively. It was further assumed that concentration and temperature measurements are corrupted with measurement noise, which is a zero mean Gaussian white noise signal with standard deviations equal to 0.007 and 0.4, respectively. Multi-level Pseudo Random Signals (MPRS), with standard deviations of 0.275 m^3/s and 19.766 K and switching times of 0.3 and 0.5 min, were generated by modifying the PRBS generated using "*idinput*" function in system identification tool box of *MATLAB*. These MPRS signals were used to introduce simultaneous perturbations in the inlet flow rate (F_{in}) and inlet temperature (T_i), respectively. The inputs used to generate the validation data set and the unmeasured disturbance introduced in input concentration are given in figure (2(a)).

Model identification was carried out using data generated for 800 minutes. The number of filters n_{ij} is chosen as $n_{ij} = 2$ with respect to all the input arguments to the state dynamics in NOE as well as NAR components of the model. The resulting state vector has 6 states in the combined NOE + NARMA model. The comparison of predictions generated by NOE+NARMA and NOE



(a) Manipulated inputs and unmeasured disturbance

(b) NOE+NARMA predictions

(c) NOE predictions

(d) Steady state plots

Fig. 2. Model Validation Results

Table 1. Model validation statistics

Prediction	PPE	PEE
NOE+NARMA	18.5160	12.0682
NOE	32.7957	28.7206

models with the validation data is presented in Figure(2(b)) and Figure(2(c)), respectively. The results of model validation are also summarized in Table(1). If we compare the PEE values for NOE+NARMA model with that of NOE model and model predictions presented in Figures (2(b)) and (2(c)), it is clear that the NARMA component significantly contributes towards improving the inter-sample predictions. Figure (2(c)) demonstrates that the NOE component of the model has very good long range prediction capability. Figure (2(d)) represents comparison of the steady state behavior of the plant with identified models for reactor concentration with respect to manipulated variables. As can be observed form this figure, the NOE +NAR models capture the steady state behavior of the system over a wide operating range around extremum operating point. In particular, the predictions capture the change in the sign of steady state gain reasonably well.

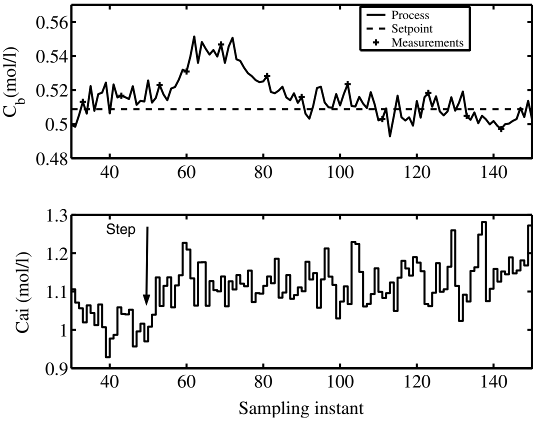


Fig. 3. Regulatory responses of multi-rate NMPC

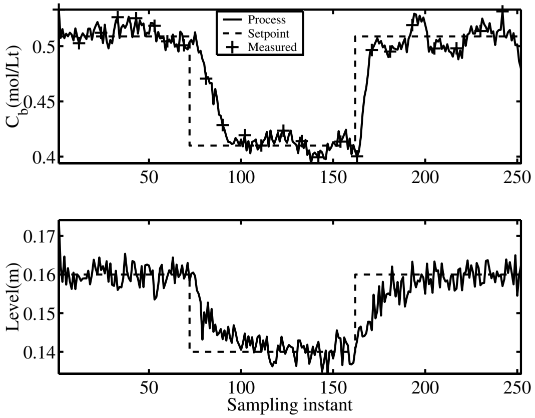


Fig. 4. Servo responses of multi-rate NMPC

The identified NOE+NARMA was further used to formulate an multi-rate NMPC scheme. The NMPC controls reactor level (h) and reactor concentration (C_b) by manipulating inlet flow rate (F_i) and inlet feed temperature (T_i). Our main focus here was to asses the regulatory and servo performances of the proposed NMPC scheme. The operating point was chosen as the peak point, which is a singular point where steady state gain reduces to zero and changes it sign across the peak point. The simulations studies are carried out with prediction horizon of $p = 40$,control horizon of $q = 1$, $W_u = \mathbf{I}$ and $W_e = diag\left[1\ 100\right]$. To assess the regulatory performance, we introduce a step jump in the input concentration at 50^{th} sampling instant of magnitude 0.15. From Figure 3, it can be observed that the controller quickly rejects the step jump in (C_{ai}) and maintain the concentration at the desired setpoint. The servo performance of the NMPC, when simultaneous step changes are introduced in both the set-points, is presented in Figure 4. It can be observed that the controller tracks the set-point smoothly with out any offset and controls the process at the singular operating point even when the concentration measurements are available at irregular intervals.

5 Conclusions

The proposed NOE+NAR type nonlinear observers identified from irregularly sampled output data generate excellent inter sample predictions and can be used for formulating inferential control schemes. These models also generate very good long range predictions and are suitable for developing multi-rate NMPC formulation. The proposed inferential NMPC formulation based on these models generates satisfactory regulatory as well as servo performance in the irregularly sampled multi-rate scenario.

References

[1] Li, D., S.L. Shah and T. Chen . Identification of Fast-Rate Models from Multi-rate data. *Int. J. Control*,**74(7)**, 680-689, (2001) .

[2] Bequette, B.W., "Nonlinear predictive control using multi-rate sampling " *Canadian Journal of Chemical Engineering*, 69,136-143, (1991).

[3] Patwardhan,S.C, and S. L. Shah, From data to diagnosis and control using generalized orthonormal basis filters. Part I: Development of state observers, *Journal of Process Control*, 15, (7), 819-835,(2005)

[4] Ninness, B. M. and F. Gustafsson, A Unifying Construction of Orthonormal Basis for System Identification, *IEEE Transactions on Automatic Control*, 42, (4), 515-521, (1997).

[5] Li, W. C. and L. T. Biegler . Process Control Strategies for Constrained Nonlinear Systems, *Ind. Eng. Chem. Res.*, 27, 142,(1988).

[6] Srinivasaro,M.,R.D.Gudi, Sachin C. Patwardhan,"Identification of fast-rate nonlinear output error models from multi-rate data", *Proc. of 16th IFAC World Congress*, Prague, (2005).

Nonlinear Model Predictive Control: A Passivity-Based Approach

Tobias Raff, Christian Ebenbauer, and Frank Allgöwer

Institute for Systems Theory and Automatic Control, University of Stuttgart,
Pfaffenwaldring 9, 70550 Stuttgart, Germany
{raff,ce,allgower}@ist.uni-stuttgart.de

Summary. This paper presents a novel approach for nonlinear model predictive control based on the concept of passivity. The proposed nonlinear model predictive control scheme is inspired by the relationship between optimal control and passivity as well as by the relationship between optimal control and model predictive control. In particular, a passivity-based state constraint is used to obtain a nonlinear model predictive control scheme with guaranteed closed loop stability. Since passivity and stability are closely related, the proposed approach can be seen as an alternative to control Lyapunov function based approaches. To demonstrate its applicability, the passivity-based nonlinear model predictive control scheme is applied to control a quadruple tank system.

1 Introduction

In this paper, nonlinear model predictive control is used to control unconstrained nonlinear systems. Due to the fact that this control strategy does not naturally guarantee closed loop stability [1], many different approaches have been developed to circumvent this problem [11]. Most of these nonlinear model predictive control schemes achieve closed loop stability by using the concept of Lyapunov stability, e.g., in the model predictive control setup a control Lyapunov function is included as a state constraint [16], or as a terminal cost [6], or as a terminal cost in conjunction with a terminal region [3] to achieve stability.

A nonlinear model predictive control scheme based on the relationships between optimal control, nonlinear model predictive control, and control Lyapunov function was developed in [16]. In this paper, a nonlinear model predictive control scheme is developed based on the concept of passivity in this spirit, i.e., the proposed nonlinear model predictive control scheme is inspired by the relationship between optimal control and passivity [8, 12] as well as by the relationship between optimal control and nonlinear model predictive control [16]. The relationship between optimal control and passivity is that an input affine nonlinear system can be optimal if and only if it satisfies a passivity property with respect to the optimal feedback [8, 12]. Optimal control and nonlinear model predictive control are linked by the fact that model predictive control is a computational tractable approximation of the optimal control problem by repeatedly solving on-line a finite horizon optimal control problem. Based on these relationships it

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 151–162, 2007.
springerlink.com

is shown that passivity-based concepts and nonlinear model predictive control can be merged into a nonlinear model predictive control scheme which maintains the advantages of the individual concepts, i.e., closed loop stability due passivity and good performance due to one-line optimization in nonlinear model predictive control.

The remainder of the paper is organized as follows: In Section 2, the necessary background of passivity is given. Furthermore, the connection between passivity and optimal control is briefly reviewed which is the underlying motivation of the passivity-based nonlinear model predictive control scheme. In Section 3, nonlinear model predictive control is introduced and a short overview of different model predictive control schemes with guaranteed stability is given. In Section 4, the main result is presented and discussed, namely, a passivity-based nonlinear model predictive control scheme. To demonstrate its applicability, the proposed nonlinear model predictive control scheme is applied to control a quadruple tank system in Section 5. Finally, conclusions are drawn in Section 6.

2 Passivity

The concept of passivity is often used in the analysis and synthesis of nonlinear systems [10, 17]. In the following, the necessary background, adjusted for the paper, is given. Consider the nonlinear system given by

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x), \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the state, $u \in \mathbb{R}^m$ is the input and $y \in \mathbb{R}^m$ the output. It is assumed that all data is locally Lipschitz and that $x = 0, u = 0$ is an equilibrium point. The system (1) is said to be passive if there exists a positive semidefinite storage function S such that the inequality

$$S(x(t_1)) - S(x(t_0)) \leq \int_{t_0}^{t_1} u^T(t)y(t)\mathrm{d}t \tag{2}$$

is satisfied for all $t_0 \leq t_1$ when $(u(t), x(t), y(t))$ satisfy the system dynamics (1). The definition of passivity is motivated by the following consideration. In the context of electrical network theory S can be considered as the energy stored in the network, u as the port voltage, and y as the port current. Passivity of such a network means that it cannot supply more energy to its environment than energy was supplied to the network. If S is differentiable as a function of time then inequality (2) can be written as

$$\dot{S}(x(t)) \leq u^T(t)y(t), \tag{3}$$

which is often a more useful notion for analyzing passive systems than inequality (2). A further characterization of passive systems is also possible in terms of relative degree and minimum phase property. In case the system (1) has a well-defined normal form, it must be weakly minimum phase and must have a vector relative degree of one. Of importance in the concept of passivity is also the relationship between passivity and stabilization, which is summarized next.

2.1 Passivity and Stabilization

The relationship between passivity and stabilization can be established by using the storage function S as a control Lyapunov function. However, the concept of passivity requires only a positive semidefinite storage function S [17, 18]. To achieve stability, one has to assume in addition that the system (1) is zero-state detectable [10, 17], i.e., the solution of $\dot{x} = f(x)$ satisfies $\lim_{t\to\infty} x(t) = 0$ for $y(t) = 0$ for $t \geq 0$. Under these assumptions, the system (1) can be stabilized with the simple feedback $u = -y$ [17] due to

$$\dot{S}(x(t)) \leq u^T(t)y(t) \leq -y^T(t)y(t) \leq 0. \tag{4}$$

Furthermore, if S is radially unbounded, and all solutions of (1) are bounded, then the system (1) is globally stabilized by the feedback $u = -y$. At first glance, it seems that the concept of passivity is restrictive for stabilization purposes. However, in case of optimal control, the concept of passivity appears in a natural way.

2.2 Passivity and Optimality

The infinite horizon optimal control problem is defined by

$$\begin{aligned} V^\star(x_0) = \min_{u(\cdot)} \quad & \int_0^\infty \left(q(x(\tau)) + u^T(\tau)u(\tau)\right)\, d\tau \\ s.t. \quad & \dot{x} = f(x) + g(x)u,\ x(0) = x_0, \end{aligned} \tag{5}$$

where q is a positive semidefinite function and $V^\star$ is the value function, i.e., the minimal cost of (5). The optimal feedback $u^\star$ which stabilizes the system (1) and minimizes the performance index (5) is given by

$$u^\star = -k^\star(x) = -\frac{1}{2}g^T(x)\frac{\partial V^\star}{\partial x}^T, \tag{6}$$

under the assumption that $V^\star$ is the positive semidefinite and continuously differentiable solution of the Hamilton-Jacobi-Bellman equation

$$\frac{\partial V^\star}{\partial x}f(x) - \frac{1}{4}\frac{\partial V^\star}{\partial x}g(x)g^T(x)\frac{\partial V^\star}{\partial x}^T + q(x) = 0. \tag{7}$$

Note, that it is in general very difficult to solve the equation (7) and therefore the infinite horizon optimal control problem (5). The relationship between optimal control and passivity can be established [8, 12, 17] by using the value function as a storage function and the optimal feedback as an output of the system (1). Then the feedback (6) stabilizes the system (1) and minimizes the performance index (5) if and only if the system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= k^{\star}(x) \end{aligned} \tag{8}$$

is zero-state detectable and output feedback passive, i.e., $\dot{S}(x(t)) \leq u^T(t)y(t) + \frac{1}{2}y^T(t)y(t)$ with $S = \frac{1}{2}V^{\star}$ [17]. This equivalence becomes clear considering that the system (8) is stabilized by the feedback $u = -y$.

3 Nonlinear Model Predictive Control

Since stabilization of nonlinear systems (1) subject to the infinite horizon optimal problem (5) is difficult due to solving equation (7), computationally tractable approximations of (5) have been developed. One approximation approach is nonlinear model predictive control. The basic idea of nonlinear model predictive control is to determine the control input by repeatedly solving on-line the finite horizon optimal control problem

$$\begin{aligned} \min_{u(\cdot)} \quad & \int_t^{t+T} \left(q(x(\tau)) + u^T(\tau)u(\tau)\right)\, d\tau \\ s.t. \quad & \dot{x} = f(x) + g(x)u. \end{aligned} \tag{9}$$

However, this control strategy does not naturally guarantee closed loop stability [1]. To overcome this problem, several nonlinear model predictive control schemes have been developed which achieve closed loop stability [3, 6, 11, 14, 15, 16]. In the following, some approaches are summarized with respect to their stability conditions. For a detailed and more rigorous treatment of nonlinear model predictive control, see for example [11] and the references quoted therein. In [11] it was shown that most nonlinear model predictive control schemes with guaranteed stability can be summarized in the setup

$$\begin{aligned} \min_{u(\cdot)} \quad & \varphi(x(t+T)) + \int_t^{t+T} \left(q(x(\tau)) + u^T(\tau)u(\tau)\right)\, d\tau \\ s.t. \quad & \dot{x} = f(x) + g(x)u, \\ & x(t+T) \in W, \end{aligned} \tag{10}$$

where φ is a terminal cost and W a terminal region. In [6, 14] it was shown that with the setup $W = \mathbb{R}^n$ and $\varphi(x(t+T)) = \int_{t+T}^{\infty}(q(\overline{x}(\tau)) + k^T(\overline{x}(\tau))k(\overline{x}(\tau)))d\tau$ closed loop stability can be achieved, where $u = -k(x)$ is a locally stabilizing feedback of the system (1) and $\overline{x}$ is the closed loop state trajectory with the feedback $u = -k(x)$. In this approach, φ is the cost of stabilizing the system (1) with the feedback $u = -k(x)$ over the time interval $[t, \infty)$. Another possibility is to choose $W = \mathbb{R}^n$ and $\varphi(x(t+T)) = V(x(t+T))$, where V is a control Lyapunov function of the nonlinear system (1) [6]. In this approach, the cost-to-go is approximated by the control Lyapunov function V. It is further possible to achieve stability by using a terminal region W and no terminal cost, i.e., $\varphi = 0$. Using this idea, it is possible to achieve stability by imposing the terminal state constraint $x(t+T) = 0$, i.e., $W = \{0\}$ [9]. Since this approach is computationally demanding, a relaxed setup was developed in [15]. In [15] it was shown that closed loop stability is achieved by steering the final system state $x(t+T)$ in the terminal region W and by stabilizing the system with a locally stabilizing feedback $u = -k(x)$ inside the terminal region W. Other approaches with guaranteed stability use both a terminal cost φ and a terminal region W [3]. For example, the approach developed in [3] uses W as a level set of the terminal cost $\varphi(x(t+T)) = \frac{1}{2}x^T(t+T)Px(t+T)$, where $\frac{1}{2}x^TPx$ is a local control Lyapunov function of the linearized system of (1). Finally, another approach, using a form different from (10) but which is in the same line as the approach presented in Section 4, is given by

$$\begin{aligned} \min_{u(\cdot)} \quad & \int_t^{t+T} \left(q(x(\tau)) + u^T(\tau)u(\tau) \right) \, d\tau \\ s.t. \quad & \dot{x} = f(x) + g(x)u, \\ & \frac{\partial V}{\partial x}\left[f(x) + g(x)u \right] < 0 \\ & x(t+T) \in W, \end{aligned} \tag{11}$$

where V is a control Lyapunov function for the system (1) [16]. In this approach, closed loop stability is achieved by requiring that the derivate of the control Lyapunov function V is negative along the state trajectory of the closed loop system. Furthermore, the terminal region W is used a performance constraint to recover the optimal controller in case the level curves of the control Lyapunov function V correspond to the level curves of the optimal value function $V^\star$.

In summary, all nonlinear model predictive schemes summarized above share one common property, namely, the stability is achieved by using a control Lyapunov function in their setups. Of course, as shown in [16], there is a strong relationship between optimal control, nonlinear model predictive control, and control Lyapunov functions. However, as reviewed in Section 2, there is also a strong relationship between optimal control and passivity. Based on this second relationship, a nonlinear model predictive control scheme is developed in the next section.

4 Passivity-Based Nonlinear Model Predictive Control

In this section, the passivity-based nonlinear model predictive control scheme is introduced. Inspired by the idea of combining control Lyapunov functions and nonlinear model predictive control based on their relationship to optimal control [16], passivity is merged with nonlinear model predictive control in the same spirit. Suppose that the system (1) is passive with a continuously differentiable storage function S and zero-state detectable. Then the passivity-based nonlinear model predictive control scheme for the system (1) is given by

$$\begin{array}{ll} \min\limits_{u(\cdot)} & \int\limits_{t}^{t+T} \big(q(x(\tau)) + u^T(\tau)u(\tau)\big)\ d\tau \\ s.t. & \dot{x} = f(x) + g(x)u \\ & y = h(x) \\ & u^T(t)y(t) \leq -y^T(t)y(t). \end{array} \tag{12}$$

The passivity-based state constraint in the last line in the setup (12) is motivated by the fact that in case the system (1) is passive and zero-state detectable, it can be stabilized with the feedback $u = -y$. Hence, the passivity-based state constraint is a stability constraint which guarantees closed loop stability. Furthermore, if the storage function S is radially unbounded, and all solutions of the system are bounded, then the closed loop system is globally asymptotically stable. In contrast to many other nonlinear model predictive control schemes [11] which achieve stability by enforcing a decrease of the value function along the solution trajectory, the stability of the proposed nonlinear model predictive control scheme is achieved by using directly a state constraint. Hence, one obtains the following stability theorem of the passivity-based nonlinear model predictive control scheme (12):

Theorem 1. *The passivity-based nonlinear model predictive control scheme (12) locally asymptotically stabilizes the system (1) if it is passive with a continously differentiable storage function S and zero-state detectable.*

Proof. The proof of Theorem 1 is divided into two parts. In the first part it is shown that the nonlinear model predictive control scheme (12) is always feasible. In the second part it is then shown that the scheme (12) asymptotically stabilizes the system (1).

Feasibility: Feasibility is guaranteed due to the known stabilizing feeback $u = -y$. Stability: Let S be the storage function of the passive system (1). With the differentiable storage function S and the state constraint in the model predictive control scheme (12), one obtains

$$\dot{S}(x(t)) \leq u^T(t)y(t) \leq -y^T(t)y(t).$$

Using the fact that the system (1) is zero-state detectable, the same arguments presented in Theorem 2.28 of [17] can be used in order to show asymptotic stability of the origin $x = 0$. Hence, the passivity-based nonlinear model predictive control scheme (12) asymptotically stabilizes system (1) if it is passive and zero-state detectable. ■

At the first glance, it seems that the nonlinear model predictive scheme (12) is very restrictive since it is only applicable for passive systems. However, for stabilization purposes, no real physical output y is needed. Instead it is enough to know one fictitious output $\eta = \sigma(x)$ in order to stabilize the system. Once such a fictitious output η is known, the fictitious passive system

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ \eta &= \sigma(x) \end{aligned} \tag{13}$$

can be stabilized with the passivity-based scheme (12). Note that a fictitious output η always exists, as long as a control Lyapunov function exists. Since then, by definition, $L_gV(x)$ is a fictitious output. Unfortunately, there is no efficient way to construct a passive output. However, it is often possible to find a fictitious passive output since passivity is physically motivated concept. Furthermore, if the linearized system $\dot{x} = Ax + Bu$ of (13) is stabilizable, a passive output for the linearized system is given by $\sigma(x) = B^TPx$, where P is the solution of the Riccati equation $A^TP + PA + Q - PBB^TP = 0$. This may help to find a fictitious passive output for local stabilization purposes. In the following, another property of the nonlinear model predictive control scheme (12) is discussed. Namely, the scheme (12) recovers the passivity-based feeback $u = -y$ as the prediction horizon T goes to zero. This property is summarized in the next theorem.

Theorem 2. *The passivity-based nonlinear model predictive control scheme recovers the passivity-based feedback $u = -y$ for $T \to 0$.*

Proof. To show this property, the same arguments are used as in [16]. By rewriting the the performance index to $\frac{1}{T}\int_t^{t+T}\left(q(x(\tau)) + u^T(\tau)u(\tau)\right)\,d\tau$ the optimization problem is not changed. For $T \to 0$ one obtains $q(x(t)) + u^T(t)u(t)$. Since the system state $x(t)$ at time instant t is fixed, one finally obtains

$$\begin{aligned} \min_{u(t)} \quad & u^T(t)u(t) \\ s.t. \quad & \dot{x} = f(x) + g(x)u \\ & y = h(x) \\ & u^T(t)y(t) \le -y^T(t)y(t). \end{aligned} \tag{14}$$

Based on the concept of the pointwise min-norm controller [5], the state constraint will be always active in order to minimize the control input u and the resulting feedback is therefore the passivity-based feedback $u = -y$. ■

If the value function $V^\star$ of the infinite horizon optimal control problem (5) is known and the output y is set to be equal to $y = k^\star(x)$, then the optimal performance is recovered for $T \to 0$, which is summarized below:

Corollary 1. *The optimal performance of the infinite horizon optimal control problem* (5) *is recovered by the passivity-based model predictive control scheme* (12)*, if $y = \frac{1}{2}g^T(x)\frac{\partial V^\star}{\partial x}^T$ and $T \to 0$, where $V^\star$ is the value function of the infinite horizon optimal control problem* (5)*.*

Proof. Proof follows from Theorem 2. ■

Hence, for $T \to 0$ the passivity-based feedback $u = -y$ is obtained and for any horizon $T > 0$ one can expect a better closed loop performance with the approach (12) than with the passivity-based feedback $u = -y$ due to on-line optimization in nonlinear model predictive control. Summarizing, the passivity-based nonlinear model predictive control combines passivity with model predictive control while maintaining the advantages of the individual concepts, i.e., stability due to passivity and good performance due to one-line optimization in nonlinear model predictive control. Furthermore, the passivity-based state constraint was incorporated in the model predictive control setup (12) in such a way that theoretically interesting properties are obtained, e.g., stability due to passivity and recovery of the passivity-based feedback $u = -y$ for $T \to 0$. Finally, the underlying idea of of the approach (12) are the relationships of passivity and nonlinear model predictive control to optimal control.

4.1 Extension of the Passivity-Based Nonlinear Model Predictive Control Scheme

In Corollary 1 it was shown that the passivity-based nonlinear model predictive control scheme (12) can recover the optimal feedback $u = -k^\star(x)$ for $T \to 0$ in case the optimal feeback is available. However, nothing can be said about the performance for a prediction horizon $T > 0$. The reason for this fact is that the proposed approach (12) does not take into account the cost-to-go. Many nonlinear model predictive control schemes [6, 11, 14, 16] approximate the cost-to-go in order to improve the performance. As seen in Section 3, there exists many approaches which incorporate an approximation of the cost-to-go in the nonlinear model predictive control setup. These approaches can be also used in order to improve the performance of the passivity-based nonlinear model predictive controller for any prediction horizon $T > 0$. One possibility is to introduce a terminal cost φ. By introducing a terminal cost in (12), the passivity-based nonlinear model predictive control scheme becomes

$$\begin{aligned} \min_{u(\cdot)} \quad & \varphi(x(t+T)) + \int_t^{t+T} \big(q(x(\tau)) + u^T(\tau)u(\tau)\big)\, d\tau \\ s.t. \quad & \dot{x} = f(x) + g(x)u \\ & y = h(x) \\ & u^T(t)y(t) \leq -y^T(t)y(t). \end{aligned} \tag{15}$$

One possibility is to choose the terminal cost as the storage function of the system (1), i.e., $\varphi(x(t{+}T)) = S(x(t{+}T))$. In case the optimal value function $V^\star$ is known and used as a storage function of the system (1), the optimal performance can be recovered for any prediction horizon $T > 0$. Note, in case the value function $V^\star$ is not available, stability is achieved due to the passivity-based state constraint, since the storage function of the system (1) is in general not positive definite and can therefore not be used as a control Lyapunov function. Another possibility based on [6, 14] is to choose $\varphi(x(t+T)) = \int_{t+T}^{\infty}(q(\overline{x}(\tau)) + h^T(\overline{x}(\tau))h(\overline{x}(\tau)))d\tau$, where $u = -h(x)$ is the stabilizing feedback of the system (1) and $\overline{x}$ is the corresponding closed loop system trajectory.

5 Example

In this section, the passivity-based nonlinear model predictive scheme is applied to control a quadruple tank system. The dynamical model of the quadruple tank systems [7] is given by

$$\dot{x} = f(x) + g(x)u,$$

with

$$f(x) = \begin{bmatrix} -\frac{a_1}{A_1}\sqrt{2gx_1} + \frac{a_3}{A_1}\sqrt{2gx_3} \\ -\frac{a_2}{A_2}\sqrt{2gx_2} + \frac{a_4}{A_2}\sqrt{2gx_4} \\ -\frac{a_3}{A_3}\sqrt{2gx_3} \\ -\frac{a_4}{A_4}\sqrt{2gx_4} \end{bmatrix},$$

$$g(x) = \begin{bmatrix} \frac{\gamma_1}{A_1} & 0 \\ 0 & \frac{\gamma_2}{A_2} \\ 0 & \frac{(1-\gamma_2)}{A_3} \\ \frac{(1-\gamma_1)}{A_1} & 0 \end{bmatrix},$$

and $x = [x_1, x_2, x_3, x_4]^T$, $u = [u_1, u_2]^T$. The variables x_i, $i = 1, ..., 4$ represent the water levels of the tanks, u_i, $i = 1, 2$ the control inputs, A_i, $i = 1, ..., 4$ the cross-sections of the tanks, a_i, $i = 1, ..., 4$ the cross-sections of the outlet holes, $\gamma_i = 0.4$, $i = 1, 2$ positive constants, and $g = 981\frac{cm}{s^2}$ the gravitational constant. The parameter values of A_i and a_i are taken from the laboratory experiment [4] and are given in Table 1. In order to control the quadruple tank system via the passivity-based nonlinear model predictive control, one has to search for a fictitious passive output. By using the storage function $S(x) = \frac{1}{2}x_3^2 + \frac{1}{2}x_4^2$, one obtains for $x_3, x_4 \geq 0$ [the levels of the tanks cannot be negative].

Table 1. Parameter values of the system

	A_i	a_i
$i = 1$	$50.3\ cm^2$	$0.2\ cm^2$
$i = 2$	$50.3\ cm^2$	$0.2\ cm^2$
$i = 3$	$28.3\ cm^2$	$0.1\ cm^2$
$i = 4$	$28.3\ cm^2$	$0.1\ cm^2$

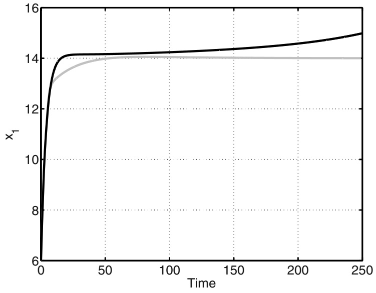


Fig. 1. Closed loop trajectory x_1: Passivity-based MPC (gray) and MPC without guaranteed stability (black)

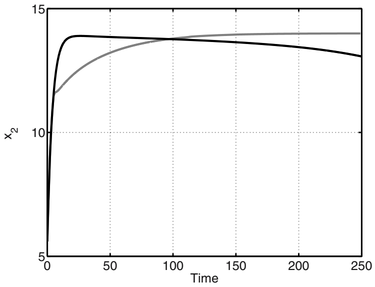


Fig. 2. Closed loop trajectory x_2: Passivity-based MPC (gray) and MPC without guaranteed stability (black)

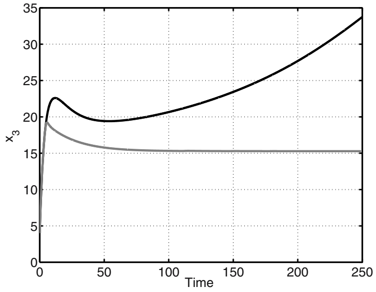


Fig. 3. Closed loop trajectory x_3: Passivity-based MPC (gray) and MPC without guaranteed stability (black)

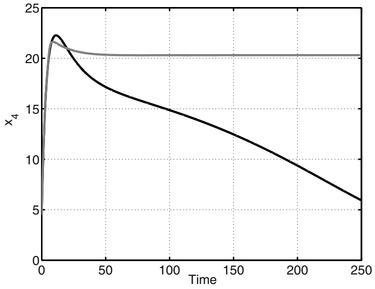


Fig. 4. Closed loop trajectory x_4: Passivity-based MPC (gray) and MPC without guaranteed stability (black)

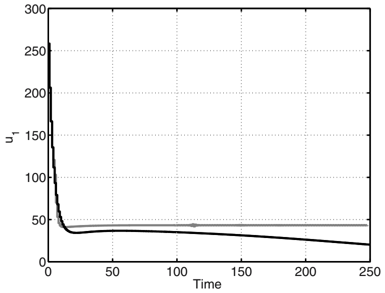


Fig. 5. Control input u_1: Passivity-based MPC (gray) and and MPC without guaranteed stability (black)

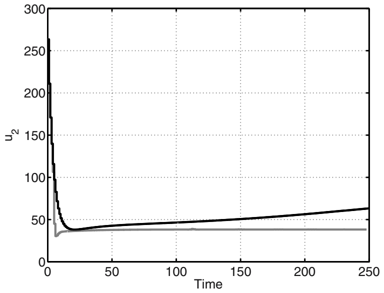


Fig. 6. Control input u_2: Passivity-based MPC (gray) and and MPC without guaranteed stability (black)

$$\begin{aligned}\dot{S}(x) &\leq \left[-\frac{a_3}{A_3}\sqrt{2gx_3} + \frac{(1-\gamma_2)}{A_3}u_2\right]x_3 + \left[-\frac{a_4}{A_4}\sqrt{2gx_4} + \frac{(1-\gamma_1)}{A_1}u_1\right]x_4,\\ &\leq x_3u_2 + x_4u_1.\end{aligned}$$

Consequentely, $y = [x_3, x_4]^T$ is a passive output for the quadruple tank system with the storage function $S(x) = \frac{1}{2}x_3^2 + \frac{1}{2}x_4^2$. Since the quadruple tank system is zero state detectable with respect to the output $y = [x_3, x_4]^T$, the passivity-based nonlinear model predictive control (12) can be used to asymptotically stabilize the quadruple tank system. In the following, the control task is to stabilize the the quadruple tank system at the equilibrium point $x_s = [14\ cm\ \ 14\ cm\ \ 15.3\ cm\ \ 20.3\ cm]^T$. The steady state control input u_s at the equilibrium point x_s is $u_s = [43.2\ \frac{ml}{s}\ \ 38.2\ \frac{ml}{s}]^T$. The performance index (5) was chosen as $1000([x_1 - x_{1s}]^2 + [x_2 - x_{2s}]^2) + (u_1 - u_{1s})^2 + (u_2 - u_{2s})^2$. Furthermore, the nonlinear model predictive control schemes were implemented with a prediction horizon $T = 60\ s$ and a sampling time $\delta = 1\ s$ in the nonlinear model predictive control toolbox [13]. Figure 1 to Figure 6 show the dynamic behavior with the initial condition $x_0 = [5\ cm\ \ 5.6\ cm\ \ 4\ cm\ \ 4.5\ cm]^T$ of the passivity-based nonlinear model predictive controller (12) and the model predictive controller (9), i.e., a model predictive controller without guaranteed closed loop stability. As it can be seen from the figures, the closed loop system is unstable with the nonlinear model predictive controller (9). This instability is not a new fact in model predictive control [1] and has motivated the development of model predictive control schemes with guaranteed closed loop stability. In contrast to the model predictive controller (9), the passivity-based nonlinear model predictive controller (12) asymptotically stabilizes the quadruple system. Hence, this examples illustrates nicely that the proposed approach achieves closed loop stability while improving the performance.

6 Conclusions

In this paper a nonlinear model predictive control scheme based on the concept of passivity was developed. It was shown that by using a specific passivity-based state constraint closed loop stability is guaranteed. The basic idea of the passivity-based nonlinear model predictive control scheme is to unify optimal control, passivity and nonlinear model predictive control based on their relationships. Since passivity and stability are closely related, the proposed approach can be seen as an alternative to the control Lyapunov function based nonlinear model predictive control scheme [16]. Finally, the passivity-based nonlinear model predictive control scheme was applied to control a quadruple tank system in order to demonstrate its applicability.

References

[1] R. R. Bitmead, M. Gevers, I. R. Petersen, and R. J. Kaye, "Monotonicity and Stabilizability Properties of Solutions of the Riccati Difference Equation: Propositions, Lemmas, Theorems, Fallacious Conjectures and Counterexamples", *Systems and Control Letters* , pages 309-315, (1985).

[2] C.I. Byrnes and A. Isidori and J.C. Willems, “Passivity, Feedback Equivalence, and the Global Stabilization of Minimum Phase Nonlinear Systems”, *IEEE Transactions on Automatic Control*, pages 1228-1240, (1991).

[3] C. Chen and F. Allgöwer, “A Quasi-Infinite Horizon Nonlinear Model Predictive Control Scheme with Guaranteed Stability”, *Automatica*, pages 1205-1217, (1998).

[4] C. Ebenbauer, “Control of a Quadruple Tank System”, *Institute for Systems Theory in Engineering Laboratory Lecture Notes*, (2004).

[5] R.A. Freeman P.V. Kokotovic, “Robust Nonlinear Control Design: State-Space and Lyapunov Techniques”, *Birkhäuser*, (1996).

[6] A. Jadbabaie, J. Yu, and J. Hauser, “ Stabilizing receding horizon control of nonlinear systems: a control Lyapunov function approach”, *In Proceedings of the American Control Conference*, pages 1535 - 1539, (1999).

[7] K.H. Johansson , “The Quadruple-Tank Process: A Multivariable Laboratory Process with Adjustable Zero”, *IEEE Transactions on Control Systems Technology*, pages 456-465 (2000).

[8] R.E. Kalman, “When is a Linear Control System Optimal”, *Transactions of the ASME, Journal of Basic Engineering*, pages 1-10, (1964).

[9] S. Keerthi and E. Gilbert, “Optimal Infinite-Horizon Feeback Laws for a General Class of Constrained Discrete-Time Systems: Stability and Moving-Horizon Approximations ”, *Journal of Optimiztaion Theory and Applications*, pages 265-293 (1988).

[10] H.K. Khalil, “Nonlinear Systems”, *Prentice Hall*, (1996).

[11] D.Q. Mayne and J.B. Rawlings and C.V. Rao and P.O.M. Scokaert, “Constrained Model Predictive Control: Stability and Optimality ”, *Automatica*, pages 789-814, (2000).

[12] P. Moylan, “Implications of Passivity for a Class of Nonlinear Systems”, *IEEE Transactions on Automatic Control*, pages 373-381, (1974).

[13] Z.K. Nagy, “Nonlinear model predictive control toolbox”, (2005).

[14] G. De Nicolao, L. Magni, and R. Scattolni, “Stabilizing Receding-Horizon Control of Nonlinear Time-Varying Systems”, *IEEE Transactions on Automatic Control*, pages 1031-1037,(1998).

[15] H. Michalska and D. Q. Mayne, “Robust Receding Horizon Control of Constrained Nonlinear Systems ”, *IEEE Transactions on Automatic Control*, pages 1623 - 1633, (1993).

[16] J. A. Primbs, V. Nevistic, and J.C. Doyle, “A receding horizon generalization of pointwise min-norm controllers”, *IEEE Transactions on Automatic Control*, pages 898 - 909, (2000).

[17] R. Sepulchre M. Jankovic P. Kokotovic, “Constructive Nonlinear Control”, *Springer*, (1997).

[18] J.C. Willems, “Dissipative Dynamical Systems. Part I: General Theory”, *Archive for Rational Mechanics and Analysis*, pages 321-351, (1972).

Numerical Methods for Efficient and Fast Nonlinear Model Predictive Control

Hans Georg Bock, Moritz Diehl, Peter Kühl, Ekaterina Kostina,
Johannes P. Schlöder, and Leonard Wirsching

Interdisciplinary Center for Scientific Computing (IWR), University of Heidelberg,
Im Neuenheimer Feld 368, 69120 Heidelberg, Germany
{bock,m.diehl}@iwr.uni-heidelberg.de

Summary. The paper reports on recent progress in the real-time computation of constrained closed-loop optimal control, in particular the special case of nonlinear model predictive control, of large differential algebraic equations (DAE) systems arising e.g. from a MoL discretization of instationary PDE. Through a combination of a direct multiple shooting approach and an initial value embedding, a so-called "real-time iteration" approach has been developed in the last few years. One of the basic features is that in each iteration of the optimization process, new process data are being used. Through precomputation - as far as possible - of Hessian, gradients and QP factorizations the response time to perturbations of states and system parameters is minimized. We present and discuss new real-time algorithms for fast feasibility and optimality improvement that do not need to evaluate Jacobians online.

1 Introduction

Feedback control based on an online optimization of nonlinear dynamic process models subject to constraints, and its special case, nonlinear model predictive control (NMPC) [1], is an emerging optimal control technique, mainly applied to problems in chemical engineering [17]. Currently, NMPC is also transferred to new fields of application such as automotive engineering, where the principal dynamics are much faster. Among the advantages of NMPC are the capability to directly handle equality and inequality constraints as well as the flexibility provided in formulating the objective function and the process model. Lately, a major aim has become to develop algorithms that are able to treat *large-scale* nonlinear first principle models without further need of re-modeling or model reduction.

In this paper we present and investigate several variants of the "real-time iteration" approach to online computation of constrained optimal feedback control laws. The present realization of this approach is based on the direct multiple shooting method [5] for DAE models [15]. Some of the ideas in this paper, and details on the standard variant of the real-time iteration approach can e.g. be found in [3, 8, 10]. The main contribution of this paper is to bring together three

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 163–179, 2007.
springerlink.com

real-time iteration levels (first introduced in [2]) that lead to cheap feasibility or optimality refinements with very fast response times for feedback. They are based on approximating both the Hessian of the optimization problem as well as the constraint Jacobians. The overall idea behind the presented ideas is to move away from the paradigm of solving the optimal control problem inherent to NMPC to convergence during each sampling period. Instead, different levels of real-time iterations are proposed that always use the most current information from the evolving process and allow to stay close to the – also evolving – optimal NMPC solution. It is important to note that the presented refinement strategies work in the presence of inequality constraints and active set changes.

A dynamic mechanical model is used to demonstrate that NMPC based on the proposed real-time iteration variants enters new time scales in computation time in the range of milliseconds.

Overview

In Section 2 we give some background material on the classical "direct" multiple shooting method for optimization problems subject to instationary differential equations, and outline in Section 3 the major ideas of the standard "real-time iteration" scheme. In Section 4 three real-time iteration variants are presented that largely avoid costly approximations of Jacobians and Hessians as well as decompositions during the runtime of the online algorithm, and therefore are particularly suitable for large scale models:

- In the linearized optimal feedback control variant presented in Section 4.2, only a matrix vector multiplication and a solution of a small scale quadratic program (QP) are necessary online.
- In the "feasibility improving" suboptimal feedback control variant presented in Section 4.3, one additional forward simulation of the nonlinear DAE system is necessary. In the limit, this variant yields feasible, but only approximately optimal controls.
- In the online approach in Section 4.4 also the gradient of the Lagrangian is needed. In the limit this method yields both feasible and optimal controls, at still lower costs than the standard real-time iteration.

In Section 5, the last variant is used to control the motion of a chain of balls connected by springs. This example demonstrates the real-time character of the presented schemes. The paper concludes with a summary and final remarks in Section 6.

2 Direct Multiple Shooting to Solve NMPC Problems

In this section we prepare the ground for the different variants of the *real-time iteration scheme*. First, the model class is presented and the open-loop optimal control problem for NMPC is set up. Then, we briefly review the direct multiple shooting approach which forms the basis of the real-time iteration variants to be discussed.

2.1 Differential Algebraic Equation Systems

Throughout this paper, we consider DAE models of index one in the following form

$$B(x(t),z(t),u(t),p)\,\dot{x}(t) = f(x(t),z(t),u(t),p) \tag{1}$$
$$0 = g(x(t),z(t),u(t),p) \tag{2}$$

Here, x and z denote the differential and the algebraic state vectors, respectively, u is the vector valued control function, whereas p is a vector of system parameters. This equation type covers many problems in practical engineering applications, from systems of ODE to reactive flow problems, e.g. the Navier-Stokes equation with chemical reactions. For the sake of simplicity we restrict ourselves in this paper to DAE of index 1, however, the generalization to higher index problems by reduction to index 1 problems with invariants can be derived following well-known techniques [18]. For notational simplicity, we will omit the parameters p in the following.

2.2 Nonlinear Model Predictive Control

Given a (possibly estimated) system state x_0, a Nonlinear Model Predictive Control (NMPC) scheme obtains a feedback control $\bar{u}(x_0)$ from the solution of an open-loop optimal control problem on a prediction and control horizon $[0, T_p]$ with length T_p:

$$\min_{u(\cdot),x(\cdot),z(\cdot)} \textstyle\int_0^{T_p} L(x(t),z(t),u(t))\;dt + E(x(T_p)) \tag{3a}$$

$$\text{subject to}\quad x(0) = x_0 \tag{3b}$$
$$B(\cdot)\dot{x}(t) = f(x(t),z(t),u(t)), \quad \forall t \in [0,T_p], \tag{3c}$$
$$0 = g(x(t),z(t),u(t)), \quad \forall t \in [0,T_p], \tag{3d}$$
$$0 \le h(x(t),z(t),u(t)), \quad \forall t \in [0,T_p], \tag{3e}$$
$$0 \le r\left(x\left(T_p\right)\right). \tag{3f}$$

Here, (3b) denotes the initial value constraint and (3c,3d) the DAE system. Additional state and control inequality constraints are expressed in (3e), and (3f) are terminal constraints that have to be satisfied.

Solving this problem for a given initial value x_0, we obtain an open-loop optimal control $u^*(t;x_0)$ and corresponding state trajectories $x^*(t;x_0), z^*(t;x_0)$. Based on this solution, a constrained nonlinear feedback control law is given by

$$\bar{u}(x_0) := u^*(0;x_0). \tag{4}$$

Due to its origin from an optimal control formulation, the NMPC feedback law has several appealing properties: among them are the possibility to base the feedback on economic criteria, to make use of important process knowledge in the form of nonlinear first principle models, and to include constraints (3e) in a straightforward way. Given suitable choices of the objective function and the final state constraint (3f), stability of the nominal NMPC dynamics can be proven [6, 7, 16].

The present article is concerned with efficient ways to calculate the feedback control $\bar{u}(x_0)$ or a suitable approximation *in real-time* while the considered process moves on.

2.3 Direct Multiple Shooting for DAE

Our approaches to the online solution of the optimal control problem (3a)–(3f) – the real-time iteration schemes – are based on the direct multiple shooting method [5] for DAE models [15], which is briefly reviewed in this section.

Parameterization of the Infinite Optimization Problem

The parameterization of the infinite optimization problem consists of two steps. For a suitable partition of the time horizon $[0, T_p]$ into N subintervals $[t_i, t_{i+1}]$,, $0 = t_0 < t_1 < \ldots < t_N = T_p$, not necessarily equidistant, we first parameterize the control function u as $u(t) = \phi_i(t, u_i)$ for $t \in [t_i, t_{i+1}]$.

Note that any parameterization ϕ_i with local support can be used without changing the structure of the problem as analyzed in the next sections.

In a second step, the DAE solutions are parameterized by *multiple shooting*. For simplicity of presentation we choose the same grid points here as for the controls. The DAE solution is decoupled on the N intervals $[t_i, t_{i+1}]$ by introducing the initial values s_i^x and s_i^z of differential and algebraic states at times t_i as additional optimization variables.

On each subinterval $[t_i, t_{i+1}]$ independently, the trajectories $x_i(t)$ and $z_i(t)$ can be computed as solutions of an initial value problem:

$$B(\cdot)\dot{x}_i(t) = f(x_i(t), z_i(t), \phi_i(t, u_i)) \tag{5a}$$

$$0 = g(x_i(t), z_i(t), \phi_i(t, u_i)) - \alpha_i(t) g(s_i^x, s_i^z, \phi_i(t_i, u_i)) \tag{5b}$$

$$x_i(t_i) = s_i^x, \ \ z_i(t_i) = s_i^z \tag{5c}$$

Here, the subtrahend in (5b) is deliberately introduced to relax the DAE and allow an efficient solution for initial values and controls s_i^x, s_i^z, u_i that may violate temporarily the consistency conditions (3d). This allows to avoid consistency iterations at the start of the integration. The scalar damping factor $\alpha_i(t)$ is chosen such that $\alpha_i(t_i) = 1$, and $\alpha_i(t) > 0$ is non-increasing on $t \in [t_i, t_{i+1}]$. Consistency of the algebraic states is ensured by adding consistency conditions (7c) for each multiple shooting node at t_i in the overall NLP defined in the next section. For more details on the relaxation of the DAE the reader is referred, e.g. to [4, 15, 18].

Since the trajectories $x_i(t)$ and $z_i(t)$ on the interval $[t_i, t_{i+1}]$ are functions of the initial values $s_i := (s_i^x, s_i^z)$ and control parameters u_i only, they will be referred to as $x_i(t; s_i, u_i)$ and $z_i(t; s_i, u_i)$ in the following. The integral part of the cost function is evaluated on each interval independently:

$$L_i(s_i, u_i) := \int_{t_i}^{t_{i+1}} L(x_i(t), z_i(t), \phi_i(t, u_i))\, dt. \tag{6}$$

Note that up to now the multiple shooting parameterization does not involve any discretization of differential operators f, g, but is exact.

Structured Nonlinear Programming Problem

The parameterization of problem (3a)–(3f) using multiple shooting and a suitable control representation leads to the following structured nonlinear programming (NLP) problem :

$$\min_{u,s} \sum_{i=0}^{N-1} \quad L_i(s_i, u_i) \quad + \quad E(s_N^x) \tag{7a}$$

$$\text{subject to} \quad s_0^x = x_0, \tag{7b}$$

$$0 = g(s_i^x, s_i^z, \phi_i(t_i, u_i)), \qquad i = 0, 1, \ldots N-1, \tag{7c}$$

$$s_{i+1}^x = x_i(t_{i+1}; s_i, u_i), \qquad i = 0, 1, \ldots N-1, \tag{7d}$$

$$r(s_N^x) \geq 0, \tag{7e}$$

with initial condition (7b), consistency conditions (7c), continuity conditions (7d), and terminal constraint (7e). Additional control and path constraints are supposed to be imposed pointwise for a suitable discretization (at n_i points τ_{ij} on each interval, $\tau_{ij} \in [t_i, t_{i+1})$, $j = 0, \ldots, n_i - 1$)

$$h(x_i(\tau_{ij}; s_i, u_i), z_i(\tau_{ij}; s_i, u_i), u_i) \geq 0, \quad j = 0, \ldots, n_i - 1, \quad i = 0, \ldots N-1. \tag{7f}$$

The NLP (7a)–(7e) can be summarized as

$$P(x_0): \qquad \min_{w} \quad a(w) \quad \text{subject to} \quad \begin{cases} b_{x_0}(w) = 0 \\ c(w) \geq 0, \end{cases} \tag{8}$$

where w contains all the multiple shooting state variables and controls:

$$w = (s_0^x, s_0^z, u_0, s_1^x, s_1^z, u_1, \ldots, u_{N-1}, s_N^x) \in \mathbb{R}^{n_w}.$$

The function $a(w)$ is the objective (7a), the vector valued equation $b_{x_0}(w) = 0$ summarizes all equalities from (7b)-(7d), and the vector valued $c(w) \geq 0$ contains the inequality constraints (7f) and (7e).

It is important to note that the initial condition (7b) is a linear constraint among the equality constraints, with the varying parameter x_0 entering linearly only in this constraint, so that

$$b_{x_0}(w) = \begin{bmatrix} s_0^x - x_0 \\ g(s_0^x, s_0^z, \phi_0(t_0, u_0)) \\ s_1^x - x_0(t_1; s_0^x, s_0^z, u_0) \\ \vdots \end{bmatrix} = b_0(w) + L x_0 \quad \text{with} \quad L := \begin{bmatrix} -\mathbb{I}_{n_x} \\ 0 \\ 0 \\ \vdots \end{bmatrix}. \tag{9}$$

Structure of the NLP

Due to the deliberate choice of state and control parameterizations, in [5] it was observed that the NLP problem (8) has a particular structure: its Lagrangian $\mathcal{L}_{x_0}(w, \lambda, \mu) = a(w) - \lambda^T b_{x_0}(w) - \mu^T c(w)$ (with Lagrange multipliers λ and μ) is *partially separable* so that its Hessian $\nabla_w^2 \mathcal{L}(w, \lambda, \mu)$ is *block diagonal*, and obviously independent of x_0 (such that we drop the index x_0 in $\nabla_w^2 \mathcal{L}_{x_0}$). Similarly, the multiple shooting parameterization introduces a characteristic *block sparse structure* of the constraint Jacobian that is also independent of x_0, e.g.[1]

$$\nabla_w b(w)^T = \begin{pmatrix} \mathbb{I} & & & & & & \\ Z_0^x & Z_0^z & Z_0^u & & & & \\ -X_0^x & -X_0^z & -X_0^u & \mathbb{I} & & & \\ & & & \ddots & \ddots & \ddots & \ddots \end{pmatrix}. \tag{10}$$

Furthermore, if the variables w are split into the *state trajectory* $s := (s_0^x,\ s_0^z,\ s_1^x,\ s_1^z,\ \ldots,\ s_N^x)$ and the *control trajectory* $u := (u_0,\ u_1,\ \ldots,\ u_{N-1})$, it is easily seen that $\nabla_s b(w)^T$ is nonsingular, a property to be exploited in the QP linear algebra.

A Newton-Type Method Solution Framework

Throughout the paper, we will work within a *Newton-type method* framework for the solution of the NLP (8). Starting with an initial guess (w_0, λ_0, μ_0), a standard full step iteration for the NLP is

$$w_{k+1} = w_k + \Delta w_k, \tag{11}$$

$$\lambda_{k+1} = \lambda_k^{\mathrm{QP}}, \quad \mu_{k+1} = \mu_k^{\mathrm{QP}}, \tag{12}$$

where $(\Delta w_k, \lambda_k^{\mathrm{QP}}, \mu_k^{\mathrm{QP}})$ is the solution of a quadratic program (QP). Other than in the classical Gauss-Newton or SQP approaches, we will rather use the more convenient form of the QP

$$\begin{array}{ll} \underset{\Delta w \,\in\, \mathbb{R}^{n_w}}{\min} & \frac{1}{2}\ \Delta w^T\, A_k\, \Delta w \quad + \quad a_k^T \Delta w \\ \text{subject to} & \begin{cases} b_{x_0}(w_k) + B_k \Delta w = 0 \\ \quad c(w_k) + C_k \Delta w \geq 0 \end{cases} \end{array} \tag{13}$$

where $A_k \approx \nabla_w^2 \mathcal{L}(w_k, \lambda_k, \mu_k)$ is an approximation of the Hessian of the Lagrangian, $a_k = \nabla_w \mathcal{L}(w_k, \lambda_k, \mu_k) + B_k^T \lambda_k + C_k^T \mu_k$, and B_k and C_k are approximations of the constraint Jacobians. Depending on the errors of these approximations we may expect linear or even super-linear convergence (see [13] for more information on quasi-Newton Jacobian updates, where superlinear convergence is proved under mild assumptions). These errors however do not influence

[1] We use the definition $\{\nabla_x f\}_{ij} := \frac{\partial f_j}{\partial x_i}$ throughout the paper.

the accuracy of the solution of the NLP which only depends on the (discretization) errors made in the evaluation of $\nabla_w \mathcal{L}$, b_{x_0} and c. In this paper we restrict ourselves to the mentioned full step iteration. In the case of NMPC where a sequence of neighboring problems is solved, this turns out to be sufficiently robust and offers the advantage of fast convergence.

3 Initial Value Embedding and Real-Time Iterations

In theoretical approaches towards constrained feedback control, including NMPC, optimal control problems have to be solved online for *varying* initial values x_0. To emphasize the dependence on a varying x_0 we write the problems (7a)–(7e) resp. (8) as $P(x_0)$. An obvious question then is how to determine an initial guess w_0 for the Newton-type iterations in each problem $P(x_0)$.

3.1 Initial Value Embedding

From previous optimization steps a solution $w^*(x_0')$ of a neighboring optimization problem $P(x_0')$ is known, including multipliers $\lambda^*(x_0')$ and $\mu^*(x_0')$. A conventional approach hence would be to use the latest information available, namely to use the old control trajectory, and to compute new state trajectory by integrating the DAE over the whole horizon using the old control trajectory and the new initial state x_0.

Instead, the principle of the initial value embedding suggests *not* to make use of x_0, but to use the solution of the previous problem $P(x_0')$ *without any modification*. This initialization for the current problem $P(x_0)$ results, of course, in a violation of the initial value constraint (7b) in the NLP (7a)–(7e), because $s_0^x = x_0' \neq x_0$. However, the constraint is already perfectly satisfied after the first full step Newton-type iteration, due to its linearity. The formulation of the initial value constraint (7b) in the NLP (7a)–(7e) can be considered a linear embedding of each optimization problem into the manifold of perturbed problems, therefore the name "initial value embedding". It allows for an efficient transition from one optimization problem to the next.

In practical applications one observes that the first iteration already yields an excellent approximation of the solution. Indeed, one can show as an obvious application of the implicit function theorem (assuming the classical regularity properties) in the case considered (exact Jacobian and Hessian, initialization at solution of $P(x_0')$), that the first QP solution delivers a tangential predictor w_1 to the solution $w^*(x_0)$ of $P(x_0)$

$$\|w_1 - w^*(x_0)\| = O\left(\|x_0' - x_0\|^2\right).$$

It is remarkable that this property even holds if the change from x_0' to x_0 requires a change of the active set, which can be proven under mild conditions [8].

Also note that the solution of the first QP not only gives us directional sensitivity feedback in a small neighborhood of x_0' where the active set does not change anymore, but in an even larger neighborhood where the linearization is still valid, see the illustration in Figure 1. In the case that only approximations of Jacobian and Hessian are used within the QP, we still obtain

$$\|w_1 - w^*(x_0)\| \leq \kappa \, \|x_0' - x_0\| ,$$

with κ being small if the quality of the approximations is good.

It is interesting to note that the good prediction properties are independent from the class of optimization problems. Therefore, the scheme can be applied to solve both tracking problems and problems with an economic objective. An example for the latter can be found in [14].

3.2 Standard Real-Time Iteration Scheme

Let us now consider the full real-time scenario, where we want to solve a sequence of optimization problems $P(x(t))$ where $x(t)$ is the system state that changes continuously with time and which is used as initial value x_0 in problem (8).

In the standard real-time iteration scheme [8, 10] we proceed as follows:

Start with an initial guess (w_0, λ_0, μ_0), and perform the following steps for $k = 0, 1, \ldots$:

1. Preparation: Based on the current solution guess (w_k, λ_k, μ_k), compute all functions and their exact Jacobians that are necessary to build the QP (13), and prepare the QP solution as far as possible without knowledge of x_0 (see Section 4.1 for more details). This corresponds to the initialization needed for the initial value embedding stated above.
2. Feedback Response: at time t_k, obtain the initial value $x_0 := x(t_k)$ from the real system state; solve the QP (13) to obtain the step $\Delta w_k = (\Delta s^x_{0\,k}, \Delta s^z_{0\,k}, \Delta u_{0\,k}, \ldots)$, and give the approximation $\tilde{u}(x(t_k)) := u_{0\,k} + \Delta u_{0\,k}$ immediately to the real system.
3. Transition: Set the next solution guess as

$$w_{k+1} := w_k + \Delta w_k, \quad \lambda_{k+1} := \lambda_k^{\mathrm{QP}}, \quad \text{and} \quad \mu_{k+1} := \mu_k^{\mathrm{QP}}.$$

3.3 Nominal Stability of the Real-Time Iteration Scheme

A central question in NMPC is nominal stability of the closed loop. For the real-time iteration scheme, the state vector of the closed loop consists of the real system state $x(t_k)$ and the content (w_k, λ_k, μ_k) of the prediction horizon in the optimizer. Due to the close connection of system and optimizer, stability of the closed loop system can only be addressed by combining concepts from both, NMPC stability theory and convergence analysis of Newton-type optimization methods. For the standard real-time iteration scheme this analysis has been carried out in [11], and for a related scheme with shift in [12]. In these papers, proofs of nominal stability of the closed loop are given under reasonable assumptions. The class of feedback controls for shrinking horizon problems is treated in [9].

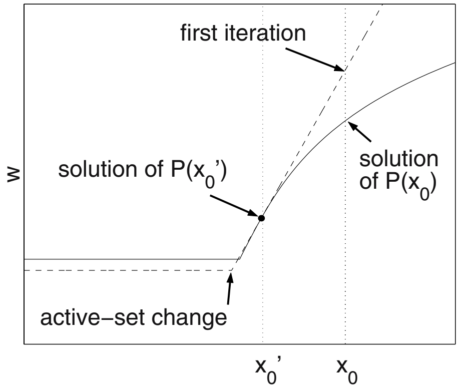


Fig. 1. Solution manifold (solid line) and tangential predictor after initial value embedding (dashed line), when initialized with the solution of $P(x_0')$. The first iteration already delivers a good predictor for the exact solution of $P(x_0)$.

4 Real-Time Iteration Variants

In the standard version of the real-time iteration scheme the time for each cycle corresponds to the time of one SQP iteration. In this article, however, we discuss four different levels of the real-time iteration scheme that differ in their need to evaluate Jacobians online. The basic idea for all these variants is to replace the QP (13) in the standard real-time iteration scheme by a generic approximated QP. This QP leaves the Hessian A as well as the Jacobians B and C constant and contains only parts of new information; a possible choice for A, B, C is $A := \nabla_w^2 \mathcal{L}(\bar{w}, \bar{\lambda}, \bar{\mu})$, $B := \nabla_w b(\bar{w})^T$, and $C := \nabla_w c(\bar{w})^T$ at some reference solution $(\bar{w}, \bar{\lambda}, \bar{\mu})$. In the following, $x_k := x(t_k)$ is the current system state at time t_k:

$$\min_{\Delta w \in \mathbb{R}^{n_w}} \quad \frac{1}{2}\ \Delta w^T\, A\, \Delta w + a_k^T \Delta w \tag{14a}$$

$$\text{subject to} \quad L x_k + b_k + B \Delta w = 0 \tag{14b}$$

$$c_k + C \Delta w \geq 0 \tag{14c}$$

The methods, that differ by the choices of a_k, b_k, c_k, proceed by performing the same three steps as in the standard real-time iteration scheme presented in Section 3.2, with the only difference that now the approximated version (14) is prepared and solved in each iteration instead of a QP (13) with exact Jacobians. As the matrices A, B, C are constant, a large share of the computations for preparation and solution of the QP can for all variants already be performed offline, leading to a considerably shorter preparation phase.

We want to point out that for strongly nonlinear processes it might be necessary to update the matrices A, B, C from time to time to ensure sufficient contractivity of the real-time iterates, but that we leave this rare updating unconsidered here for simplicity of presentation. For mildly nonlinear systems, however, the matrices A, B, C might really be kept constant without any updates, for example evaluated once for all at a reference solution.

In what follows, three different variants of the real-time iteration scheme will be shown in detail, differing in the choice of a_k, b_k, and c_k. While variant A is nothing else than linear MPC, variant B converges to nonlinearly feasible (but suboptimal) MPC solutions. Variant C will even converge to the true nonlinear MPC feedback - without the need to evaluate any derivative matrix online. But before we proceed, a preliminary remark on condensing of a QP is necessary.

4.1 A Prerequisite: Offline Condensing

In all approaches we use fixed approximations of the Jacobians B and C, e.g. by evaluating offline $\nabla_w b(\bar{w})^T$ and $\nabla_w c(\bar{w})^T$ for a reference trajectory $\bar{w}$ that may be an exact or approximate solution of an NLP $P(\bar{x})$ for some state $\bar{x}$. We also use a fixed approximation A of the Hessian, that may be based on the reference solution of $P(\bar{x})$ and computed as $\nabla_w^2 \mathcal{L}(\bar{w}, \bar{\lambda}, \bar{\mu})$, or be chosen otherwise. Online, we use these fixed components A, B, C to formulate a QP of the form (14), where only the vectors a_k, b_k, c_k and the initial value x_k are changing online. It is well known that because $\nabla_s b(\bar{w})$ is invertible, the online QP solution can be prepared by a *condensing* of the QP [5, 8]: We divide Δw into its state and control components Δs and Δu, and resolve the equality constraints (14b) to obtain Δs as a linear function of Δu (and x_k), such that we can substitute

$$\Delta w = m(b_k) + \tilde{L} x_k + M \Delta u. \tag{15}$$

Note that the matrices $\tilde{L}$ and M are independent of a_k, b_k, c_k, x_k and can in all variants be precomputed offline, exploiting the structure of B in Eq. (10). In Eq. (17) below, we show how $m(b_k)$ can be computed efficiently online. We use expression (15) to substitute Δw wherever it appears in the QP, to yield the condensed QP:

$$\begin{array}{ll} \min\limits_{\Delta u} & \frac{1}{2}\; \Delta u^T \, A^{\mathrm{c}} \, \Delta u \;\; + \;\; \Big(a^{\mathrm{c}}(a_k, b_k) + \tilde{A} x_k\Big)^T \, \Delta u \\ \text{subject to} & \Big(c^{\mathrm{c}}(c_k, b_k) + \tilde{C} x_k\Big) + C^{\mathrm{c}} \Delta u \geq 0 \end{array} . \tag{16}$$

All matrices, $A^{\mathrm{c}} := M^T A M, \tilde{A} := M^T A \tilde{L}, \tilde{C} := C \tilde{L}, C^{\mathrm{c}} := C M$ of this condensed QP are precomputed offline. Online, only the vectors $a^{\mathrm{c}}(a_k, b_k) + \tilde{A} x_k$ and $c^{\mathrm{c}}(c_k, b_k) + \tilde{C} x_k$ need to be computed, as shown in the following.

4.2 Variant A: Linear MPC Based on a Reference Trajectory

In the first approach [3, 8], we compute offline the *fixed* vectors $b := b_0(\bar{w}), c := c(\bar{w})$ and $a := \nabla_w a(\bar{w})$, and set $a_k := a$, $b_k := b$, $c_k := c$ in all real-time iterations. We can therefore precompute $m := m(b)$ and also $a^{\mathrm{c}} := a^{\mathrm{c}}(a, b) = M^T(Am + a)$ and $c^{\mathrm{c}} := c^{\mathrm{c}}(c, b) = c + Cm$.

Online, once x_k becomes known, only two sparse matrix-vector products and two vector additions are needed for computation of $a^{\mathrm{c}} + \tilde{A}x_k$ and $c^{\mathrm{c}} + \tilde{C}x_k$, and the condensed QP (16) in variables $\varDelta u \in \mathbb{R}^{n_u \times N}$ must be solved. The solution of a QP of this size is standard in linear MPC applications and can usually be achieved quickly, in particular if an online active set strategy is used. Note that the dimension of the condensed QP (16) does *not* depend on the dimensions n_x and n_z of the state vectors, and that the cost of the matrix-vector products grows linearly with n_x and is independent of n_z.

4.3 Variant B: Online Feasibility Improvement

In the second variant of the real-time iteration scheme (originally proposed in [3]), we extend the online computational burden by one additional evaluation of $b_0(w_k)$ and $c(w_k)$, i.e. we set $b_k := b_0(w_k)$ and $c_k := c(w_k)$ in the QP (14). This allows to yield a feasibility improvement for nonlinear constraints. Offline, in addition to A, B, C we also compute a fixed objective gradient, e.g. $a = \nabla_w a(\bar{w})$, and then simply set $a_k := a + A(w_k - \bar{w})$ in each iteration.

Recalling the precomputed form of the condensed QP (16), only the vectors $a^{\mathrm{c}}(a_k, b_k)$ and $c^{\mathrm{c}}(c_k, b_k)$ have to be computed online, during the preparation phase.

Based on the block sparse structure of B shown in Eq. (10), the vector $m(b_k) = (m_0^x, m_0^z, m_0^u, \ldots, m_N^x)$ in (15) is for given $b_k = (b_0^x, b_0^z, b_1^x, b_1^z, \ldots, b_N^x)$ efficiently computed by a recursion. Starting with $m_0^x := b_0^x$, we compute for $i = 0, \ldots, N-\ 1$[2]:

$$m_i^u := 0, \quad m_i^z := -\left(Z_i^z\right)^{-1}\left(b_i^z + Z_i^x m_i^x\right), \quad m_{i+1}^x := b_{i+1}^x + X_i^x m_i^x + X_i^z m_i^z. \quad (17)$$

Based on $m(b_k)$, we can quickly compute $a^{\mathrm{c}}(a_k, b_k) = M^T(Am(b_k) + a_k)$ and $c^{\mathrm{c}}(a_k, b_k) = c_k + Cm(b_k)$ (with $a_k = a + A(w_k - \bar{w})$). This computation involves only structured matrix vector products. This is the end of step 1, the preparation phase. Once x_k is known, the condensed QP (16) is solved in the feedback step 2, as in level A.

However, we do have to perform a transition step 3, i.e., update $w_{k+1} = w_k + \varDelta w_k$, to meet nonlinear constraints. This means that the matrix vector multiplication $M\varDelta u_k$ and the additions $\varDelta w_k = m(b_k) + Lx_k + M\varDelta u_k$ need to be done online.

This online algorithm does not make use of λ_k and μ_k and therefore does not need to update them during online computations.

4.4 Variant C: Online Optimality Improvement

In the third variant of the real-time iteration scheme, we further extend the online computational burden by one additional evaluation of the gradient of the Lagrangian $\nabla_w \mathcal{L}(w_k, \lambda_k, \mu_k)$. In the online QP (14), we set $a_k := \nabla_w \mathcal{L}(w_k, \lambda_k, \mu_k) + B^T\lambda_k + C^T\mu_k$, as well as $b_k := b_0(w_k)$ and $c_k := c(w_k)$. This approach allows

[2] The matrices Z_i^z can be pre-factored offline.

to yield not only an improvement of feasibility, but also of optimality for the original NLP (8).

The remaining online computations are slightly more expensive than for levels B and C, as we need to recover the multipliers $\lambda_k^{\mathrm{QP}}, \mu_k^{\mathrm{QP}}$ of the uncondensed QP (14) for the transition step 3, as follows:

First, the inequality multipliers μ_k^{QP} are directly obtained as the multipliers μ_k^{cQP} of the condensed QP (16): $\mu_k^{\mathrm{QP}} := \mu_k^{\mathrm{cQP}}$. Second, the equality multipliers λ_k^{QP} can be computed as $\lambda_k^{\mathrm{QP}} := (BS^T)^{-T} S(A\Delta w_k + a_k - C^T \mu_k^{\mathrm{QP}})$ where S is a projection matrix that maps w to its subvector s.

The matrix BS^T contains only those columns of B that correspond to the variables s, cf Eq. (10), and is thus invertible. Abbreviating $a := S(A\Delta w_k + a_k - C^T \mu_k^{\mathrm{QP}})$, $a = (a_0^x, a_0^z, \ldots, a_N^x)$, we can compute $\lambda_k^{\mathrm{QP}} = (\lambda_0^x, \lambda_0^z, \ldots, \lambda_N^x)$ recursively backwards: Starting with $\lambda_N^x := a_N^x$, we compute, for $i = N-1, N-2, \ldots, 0$:

$$\lambda_i^z = (Z_i^z)^{-T} \left(a_i^z + (X_i^z)^T \lambda_{i+1}^x \right), \quad \lambda_i^x = a_i^x + (X_i^x)^T \lambda_{i+1}^x - (Z_i^x)^T \lambda_i^z .$$

where we employ the submatrix notation of Eq. (10) for the matrix B, respectively BS^T. The proof of nominal stability of NMPC based on this variant follows the lines of the proof for the standard scheme mentioned in section 3.3.

5 A Real-Time NMPC Example

To demonstrate the real-time applicability of the NMPC schemes discussed above, a simulation experiment has been set up. In this experiment, a chain of massive balls connected by springs is perturbed at one end, and the control task is to bring the system back to steady state.

An ODE Model for a Chain of Spring Connected Masses

Consider the following nonlinear system of coupled ODEs

$$\ddot{x}_i + \beta\, \dot{x}_i - \frac{1}{m} (F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}) - g = 0, \qquad i = 1, 2, \ldots, N-1 \quad \text{(18a)}$$

$$F_{i+\frac{1}{2}} \triangleq S \left(1 - \frac{L}{\|x_{i+1} - x_i\|} \right) (x_{i+1} - x_i), \qquad i = 0, 1, \ldots, N-1. \quad \text{(18b)}$$

$$x_0(t) \equiv 0, \qquad \dot{x}_N(t) = u(t), \quad \text{(18c)}$$

for the ball positions $x_0(t), \ldots, x_N(t) \in \mathbb{R}^3$ with boundary conditions (18c) and a prescribed control function $u(t) \in \mathbb{R}^3$. Equations (18a)–(18c) describe the motion of a chain that consists of eleven balls (i.e. $N = 10$), numerated from 0 to 10, that are connected by springs. At one end, the first ball is fixed in the origin, the velocity of the other end (the "free" end) is prescribed by the function u. The motion of the chain is affected by both laminar friction and gravity (gravitational acceleration $g = 9.81$ m/s^2) as an external force. The model parameters are: mass $m = 0.03$ kg, spring constant $S = 1$ N/m, rest length of a spring $L = 0.033$ m, and friction coefficient $\beta = 0.1\ \mathrm{s}^{-1}$. The chain movement can be controlled by adjusting the velocity of ball no. 10 at the "free" end (for the sake of simplicity we assume that it is possible to directly adjust this velocity). Figure 2 illustrates the example.

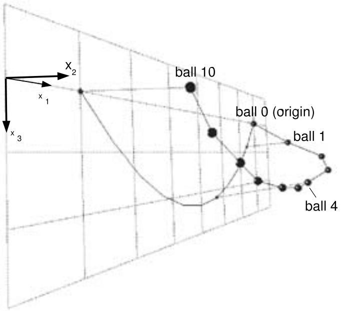


Fig. 2. A chain of 11 balls connected by springs. The first ball (in the back) is fixed, the last one can be moved freely; its Cartesian velocities serve as controls.

Optimal Control Problem Formulation

Aim of the controller is to bring the perturbed chain back to steady state. The open loop control problem is formulated with the following cost function:

$$L(x(t), u(t)) = \gamma \, \|x^N(t) - x^{\mathrm{end}}\|_2^2 + \delta \sum_{j=1}^{N-1} \|\dot{x}^j(t)\|_2^2 + \epsilon \|u(t)\|_2^2, \tag{19}$$

with the weighting factors $\gamma = 25$, $\delta = 1$, and $\epsilon = 0.01$. The chosen cost function (19) implicitly describes the steady state and allows to omit calculating the steady state position for each ball. The optimal control problem is

$$\min_{u(\cdot),x(\cdot)} \int_0^{T_p} L(x(t), u(t)) \; dt \tag{20a}$$

subject to the system equations (18) and the input constraint

$$\|u(t)\|_\infty \leq 1, \quad \forall t \in [0, T_p] \tag{20b}$$

The control horizon T_p is set to 8 s, while the sampling time is only 200 ms. It is clear that the NMPC has to update the controls for every new sampling instant. If the time needed to solve the optimization problem exceeds 200 ms, then this delay will deteriorate the control performance and eventually even miss to bring the chain back to steady state. In the case of delayed control updates, the obtained velocities act as new disturbances on the process rather than as controls. This illustrates why waiting for the exact solution of the optimal control problem is not a good idea for such fast processes. The control problem is solved with the direct multiple shooting method on 40 multiple shooting intervals.

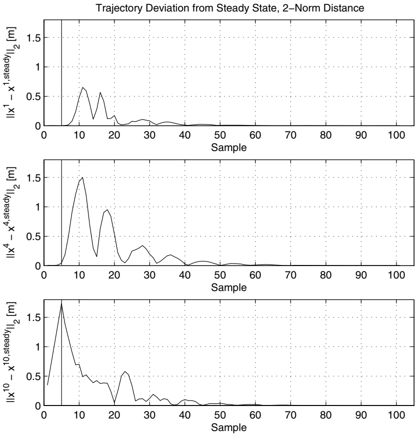


Fig. 3. Deviation of balls no. 1,4, and 10 from steady state in the controlled case. The deviation is expressed in the 2-norm of all Cartesian components.

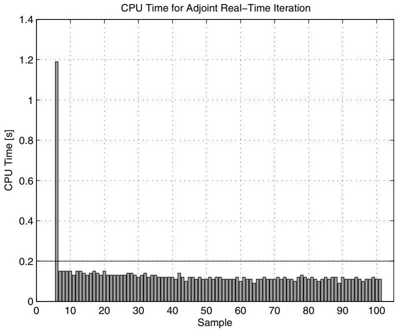


Fig. 4. CPU time for each sampling period. Note that the CPU time consists of both preparation and feedback phase. However, the feedback phase is in the range of 2 ms only! The preparation phase is then carried out while waiting for the new measurement update. The large CPU time peak in the beginning is due to the preparatory calculations that are done off-line.

In the simulation experiment, the chain initially is in steady state. Then, it is perturbed over five sampling periods by a constant velocity vector $u_{\text{pert}} = [-1\ 1\ 1]^T$ at the free end, before the controller becomes active, using the same end for disturbance rejection. The controls are calculated with variant C of the real-time iteration scheme (Section 4.4).

The resulting closed loop response for three different balls and the control moves can be seen in Figures 3 (2-norm deviation from steady state for three different balls) and 5 (absolute deviation from steady-state in y-direction). The corresponding control moves are shown in Figure 6. In all figures, x represents the vector of Cartesian coordinates of a ball. A superscript denotes the number of the ball (also see Figure 2), while a subscript picks one of the three coordinates. The simulation has been run on an Intel Pentium 4 machine with 2.8 GHz, 1024 kB L2 cache, 1 GB main memory, under Linux operating system Suse 9.3.

The computation times are depicted in Figure 4. Here, the entire CPU time of variant C needed to compute a control update has been measured. It is important to note that this comprises both the preparation and the feedback phase. The CPU time for the feedback phase only is in the range of 2 ms, meaning that the feedback delay between the new measurement update and the control update is negligible. The preparation phase is carried out *after* the new controls are given to the process in order to prepare the next optimization step. The large CPU time at the beginning of the control action is due to preparation calculations which are done off-line, before the state disturbance is measured. They correspond to the cost of a standard real-time iteration.

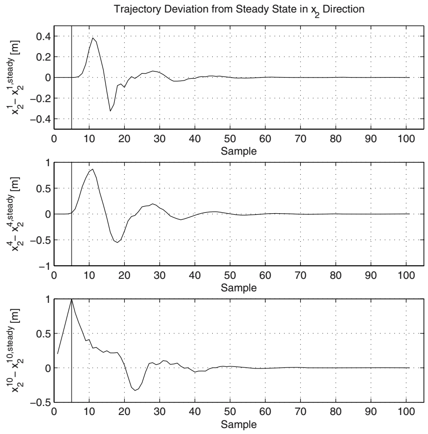


Fig. 5. Absolute deviation from steady state in y-direction of balls no. 1,4, and 10. The horizontal line at sample no. 5 marks the beginning of control action. Before, ball no. 10 was perturbed by a constant velocity.

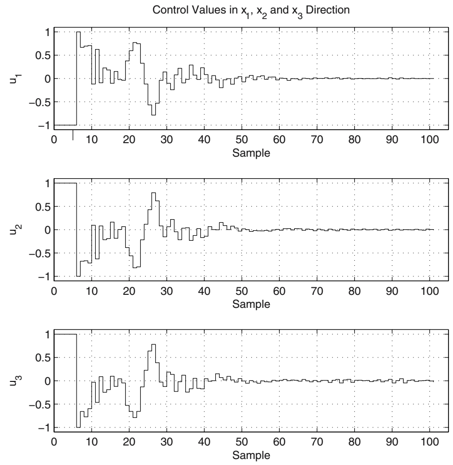


Fig. 6. Control moves calculated by the NMPC to reject the disturbance. The controls are the three Cartesian velocities of ball no. 10. Note that during the first 5 samples, these controls serve as disturbances to the chain.

6 Conclusions

We have discussed a class of methods for online computation of constrained optimal feedback controls in NMPC that are based on the direct multiple shooting method and a "real-time iteration" approach. They use an initial value embedding for efficient initialization of subsequent optimization problems, and treat in each iteration of the optimization process a different optimization problem, always with the most current system state x_k as initial value.

Three real-time iteration variants have been proposed that do not need to evaluate Jacobians during the runtime of the online algorithm and are therefore suitable for large scale DAE models with short timescales. In the presented variants, online computations with different properties are performed:

- Variant A requires only the online solution of a condensed QP. It can be interpreted as a linear MPC based on reference trajectories.
- In variant B, one additional DAE simulation is needed to evaluate the constraint functions. This variant yields a feasible but sub-optimal solution.
- Variant C requires also the gradient of the Lagrangian and is able to achieve feasibility as well as optimality for inequality constrained problems, still without evaluating Jacobians online.

The practical performance of variant C has been demonstrated in a simulation experiment, controlling the nonlinear mechanical system of a chain of balls and springs. A natural idea that arises is to combine the standard real-time iteration scheme and its variants to a multi level real-time iteration which employs the variants A, B, C and the standard real-time iteration scheme in a hierarchical fashion. This algorithm aims at combining good local convergence properties with short sampling times and is subject of future investigation.

Acknowledgments. The authors gratefully acknowledge support by DFG grant BO864/10-1.

References

[1] F. Allgöwer, T.A. Badgwell, J.S. Qin, J.B. Rawlings, and S.J. Wright. Nonlinear predictive control and moving horizon estimation – An introductory overview. In P. M. Frank, editor, *Advances in Control, Highlights of ECC'99*, pages 391–449. Springer, 1999.

[2] H.G. Bock, M. Diehl, E.A. Kostina, and J.P. Schlöder. Constrained optimal feedback control of systems governed by large differential algebraic equations. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, editors, *Real-Time and Online PDE-Constrained Optimization.* SIAM, 2005. (in print).

[3] H.G. Bock, M. Diehl, D.B. Leineweber, and J.P. Schlöder. A direct multiple shooting method for real-time optimization of nonlinear DAE processes. In F. Allgöwer and A. Zheng, editors, *Nonlinear Predictive Control*, volume 26 of *Progress in Systems Theory*, pages 246–267, Basel, 2000. Birkhäuser.

[4] H.G. Bock, E. Eich, and J.P. Schlöder. Numerical solution of constrained least squares boundary value problems in differential-algebraic equations. In K. Strehmel, editor, *Numerical Treatment of Differential Equations.* Teubner, Leipzig, 1988.

[5] H.G. Bock and K.J. Plitt. A multiple shooting algorithm for direct solution of optimal control problems. In *Proc. 9th IFAC World Congress Budapest*, pages 243–247. Pergamon Press, 1984.

[6] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1218, 1998.

[7] G. De Nicolao, L. Magni, and R. Scattolini. Stability and robustness of nonlinear receding horizon control. In F. Allgöwer and A. Zheng, editors, *Nonlinear Predictive Control*, volume 26 of *Progress in Systems Theory*, pages 3–23, Basel, 2000. Birkhäuser.

[8] M. Diehl. *Real-Time Optimization for Large Scale Nonlinear Processes*, volume 920 of *Fortschr.-Ber. VDI Reihe 8, Mess-, Steuerungs- und Regelungstechnik.* VDI Verlag, Düsseldorf, 2002. http://www.ub.uni-heidelberg.de/archiv/1659/.

[9] M. Diehl, H.G. Bock, and J.P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM J. Control Optim.*, 43(5):1714–1736, 2005.

[10] M. Diehl, H.G. Bock, J.P. Schlöder, R. Findeisen, Z. Nagy, and F. Allgöwer. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *J. Proc. Contr.*, 12(4):577–585, 2002.

[11] M. Diehl, R. Findeisen, and F. Allgöwer. A stabilizing real-time implementation of nonlinear model predictive control. In L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, editors, *Real-Time and Online PDE-Constrained Optimization.* SIAM, 2004. (in print).

[12] M. Diehl, R. Findeisen, F. Allgöwer, H.G. Bock, and J.P. Schlöder. Nominal stability of the real-time iteration scheme for nonlinear model predictive control. *IEE Proc.-Control Theory Appl.*, 152(3):296–308, May 2005.

[13] M. Diehl, A. Walther, H.G. Bock, and E. Kostina. An adjoint-based SQP algorithm with quasi-Newton Jacobian updates for inequality constrained optimization. Technical Report MATH-WR 02-2005, Technical University Dresden, Germany, 2005.

[14] P. Kühl, A. Milewska, M. Diehl, E. Molga, and H.G. Bock. NMPC for runaway-safe fed-batch reactors. In *Proc. Int. Workshop on Assessment and Future Directions of NMPC*, pages 467–474, 2005.

[15] D.B. Leineweber. *Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models*, volume 613 of *Fortschr.-Ber. VDI Reihe 3, Verfahrenstechnik*. VDI Verlag, Düsseldorf, 1999.

[16] D.Q. Mayne. Nonlinear model predictive control: Challenges and opportunities. In F. Allgöwer and A. Zheng, editors, *Nonlinear Predictive Control*, volume 26 of *Progress in Systems Theory*, pages 23–44, Basel, 2000. Birkhäuser.

[17] S.J. Qin and T.A. Badgwell. Review of nonlinear model predictive control applications. In B. Kouvaritakis and M. Cannon, editors, *Nonlinear model predictive control: theory and application*, pages 3–32, London, 2001. The Institute of Electrical Engineers.

[18] V.H. Schulz, H.G. Bock, and M.C. Steinbach. Exploiting invariants in the numerical solution of multipoint boundary value problems for DAEs. *SIAM J. Sci. Comp.*, 19:440–467, 1998.

Computational Aspects of Approximate Explicit Nonlinear Model Predictive Control

Alexandra Grancharova[1,2], Tor A. Johansen[1], and Petter Tøndel[1]

[1] Department of Engineering Cybernetics, Norwegian University of Science and Technology, N-7491 Trondheim, Norway
Tor.Arne.Johansen@itk.ntnu.no, Petter.Tondel@itk.ntnu.no

[2] Institute of Control and System Research, Bulgarian Academy of Sciences, P.O.Box 79, Sofia 1113, Bulgaria
alexandra.grancharova@abv.bg

Summary. It has recently been shown that the feedback solution to linear and quadratic constrained Model Predictive Control (MPC) problems has an explicit representation as a piecewise linear (PWL) state feedback. For nonlinear MPC the prospects of explicit solutions are even higher than for linear MPC, since the benefits of computational efficiency and verifiability are even more important. Preliminary studies on approximate explicit PWL solutions of convex nonlinear MPC problems, based on multi-parametric Nonlinear Programming (mp-NLP) ideas show that sub-optimal PWL controllers of practical complexity can indeed be computed off-line. However, for non-convex problems there is a need to investigate practical computational methods that not necessarily lead to guaranteed properties, but when combined with verification and analysis methods will give a practical tool for development and implementation of explicit NMPC. The present paper focuses on the development of such methods. As a case study, the application of the developed approaches to compressor surge control is considered.

1 Introduction

Nonlinear Model Predictive Control (MPC) involves the solution at each sampling instant of a finite horizon optimal control problem subject to nonlinear system dynamics and state and input constraints [1]–[5]. A recent survey of the main on-line optimization strategies of Nonlinear MPC (NMPC) is given in [6].

It has recently been shown that the feedback solution to linear and quadratic constrained MPC problems has an explicit representation as a piecewise linear (PWL) state feedback defined on a polyhedral partition of the state space [7]. The benefits of an explicit solution, in addition to the efficient on-line computations, include also verifiability of the implementation, which is an essential issue in safety-critical applications. For nonlinear MPC the prospects of explicit solutions are even higher than for linear MPC, since the benefits of computational efficiency and verifiability are even more important. In [8], it has been shown that the nonlinear predictive control law for the class of unconstrained input-affine nonlinear systems can be derived in an analytical form that involves the current

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 181–192, 2007.
springerlink.com

states of the system and the stability of the closed-loop system is guaranteed. An approach for NMPC design for constrained input-affine nonlinear systems has been suggested in [9], which deploys state space partitioning and graph theory to retain the on-line computational efficiency. In [10], [11], [12], approaches for off-line computation of explicit sub-optimal PWL predictive controllers for general nonlinear systems with state and input constraints have been developed, based on the multi-parametric Nonlinear Programming (mp-NLP) ideas [13]. It has been shown that for convex mp-NLP problems, it is straightforward to impose tolerances on the level of approximation such that theoretical properties like asymptotic stability of the sub-optimal feedback controller can be ensured [11], [14]. However, for non-convex problem there is a need to investigate practical computational methods that not necessarily lead to guaranteed properties, but when combined with verification and analysis methods will give a practical tool for development and implementation of explicit NMPC.

The present paper focuses on computational and implementation aspects of explicit NMPC for general nonlinear systems with state and input constraints and is structured as follows. In section 2, the formulation of the NMPC problem is given. In section 3, computational methods for approximate explicit NMPC are suggested. The application of the developed approaches to compressor surge control is considered in section 4.

2 Formulation of Nonlinear Model Predictive Control Problem

Consider the discrete-time nonlinear system:

$$x(t+1) = f(x(t), u(t)) \tag{1}$$

$$y(t) = Cx(t) \tag{2}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$, and $y(t) \in \mathbb{R}^p$ are the state, input and output variable. It is also assumed that the function f is sufficiently smooth. It is supposed that a full measurement of the state $x(t)$ is available at the current time t. For the current $x(t)$, MPC solves the following optimization problem:

$$V^*(x(t)) = \min_U J(U, x(t)) \tag{3}$$

subject to $x_{t|t} = x(t)$ and:

$$y_{\min} \le y_{t+k|t} \le y_{\max},\ k = 1, ..., N \tag{4}$$

$$u_{\min} \le u_{t+k} \le u_{\max},\ k = 0, 1, ..., N-1 \tag{5}$$

$$x_{t+N|t}^T x_{t+N|t} \le \delta \tag{6}$$

$$x_{t+k+1|t} = f(x_{t+k|t}, u_{t+k}),\ k \ge 0 \tag{7}$$

$$y_{t+k|t} = Cx_{t+k|t},\ k \ge 0 \tag{8}$$

with $U = \{u_t, u_{t+1}, ..., u_{t+N-1}\}$ and the cost function given by:

$$J(U,x(t)) = \sum_{k=0}^{N-1} \left[x_{t+k|t}^T Q x_{t+k|t} + u_{t+k}^T R u_{t+k} \right] + x_{t+N|t}^T P x_{t+N|t} \tag{9}$$

Here, N is a finite horizon. From a stability point of view it is desirable to choose δ in (6) as small as possible [15]. If the system is asymptotically stable (or pre-stabilized) and N is large, then it is more likely that the choice of a small δ will be possible. The following assumptions are made:

A1. $P, Q, R \succ 0$.
A2. $y_{\min} < 0 < y_{\max}$.
A3. There exists $u_{st} \in \mathbb{R}^m$ satisfying $u_{\min} \leq u_{st} \leq u_{\max}$, and such that $f(0, u_{st}) = 0$.

Assumption A3 means that the point $x = 0$, $u = u_{st}$, is a steady state point for system (1). The optimization problem can be formulated in a compact form as follows:

$$V^*(x(t)) = \min_{U} J(U, x(t)) \tag{10}$$

subject to:

$$G(U, x(t)) \leq 0 \tag{11}$$

This MPC problem defines an mp-NLP, since it is NLP in U parameterized by $x(t)$. An optimal solution to this problem is denoted $U^* = \left\{u_t^*, u_{t+1}^*, ..., u_{t+N-1}^*\right\}$ and the control input is chosen according to the receding horizon policy $u(t) = u_t^*$. Define the set of N-step feasible initial states as follows:

$$X_f = \{x \in \mathbb{R}^n \mid G(U,x) \leq 0 \text{ for some } U \in \mathbb{R}^{Nm}\} \tag{12}$$

If assumption A3 is satisfied and δ in (6) is chosen such that the problem (3)–(9) is feasible, then X_f is a non-empty set. Then, due to assumption A2, the origin is an interior point in X_f. In parametric programming problems one seeks the solution $U^*(x)$ as an explicit function of the parameters x in some set $X \subseteq X_f \subseteq \mathbb{R}^n$ [13]. The explicit solution allows us to replace the computationally expensive real-time optimization with a simple function evaluation. However, for general nonlinear functions J and G an exact explicit solution can not be found. In this paper we suggest practical computational methods for constructing an explicit approximate PWL solution of general non-convex nonlinear MPC problems. They can be considered as a further extension of the method proposed in [11] where the NMPC problem was assumed to be convex.

3 Computational Aspects of Approximate Explicit Nonlinear Model Predictive Control

3.1 Close-to-Global Solution of Mp-NLPs

In general, the cost function J can be non-convex with multiple local minima. Therefore, it would be necessary to apply an efficient initialization of the mp-NLP problem (10)–(11) so to find a close-to-global solution. One possible way to obtain this is to find a close-to-global solution at a point $w_0 \in X$ by comparing the local minima corresponding to several initial guesses and then to use this solution as an initial guess at the neighbouring points $w_i \in X$, $i = 1, 2, ..., l$, i.e. to propagate the solution. This is described in the following procedure:

Procedure 1. (close-to-global solution of mp-NLP)
Consider any hyper-rectangle $X_0 \subseteq X_f$ with vertices $\Theta^0 = \{\theta_1^0, \theta_2^0, ..., \theta_M^0\}$ and center point w_0. Consider also the hyper-rectangles $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$ with vertices respectively $\Theta^j = \left\{\theta_1^j, \theta_2^j, ..., \theta_M^j\right\}$, $j = 1, 2, ..., N_j$. Suppose $X_0^1 \subset X_0^2 \subset ... \subset X_0^{N_j}$. For each of the hyper-rectangles X_0 and $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$ determine a set of points that belongs to its facets and denote this set $\Psi^j = \left\{\psi_1^j, \psi_2^j, ..., \psi_{N_\Psi}^j\right\}$, $j = 0, 1, 2, ..., N_j$. Define the set of all points $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$, where $w_i \in \left\{\bigcup_{j=0}^{N_j} \Theta^j\right\} \cup \left\{\bigcup_{j=0}^{N_j} \Psi^j\right\}$, $i = 1, 2, ..., N_1$. Then:

a). *Determine a close-to-global solution of the NLP (10)–(11) at the center point w_0 through the following minimization:*

$$U^*(w_0) = \arg \min_{U_i^{local} \in \left\{U_1^{local}, U_2^{local}, ..., U_{N_U}^{local}\right\}} J(U_i^{local}, w_0) \tag{13}$$

where U_i^{local}, $i = 1, 2, ..., N_{N_U}$ correspond to local minima of the cost function $J(U, w_0)$ obtained for a number of initial guesses U_i^0, $i = 1, 2, ..., N_{N_U}$.

b). *Determine a close-to-global solution of the NLP (10)–(11) at the points $w_i \in W$, $i = 1, 2, ..., N_1$ in the following way:*

1. *Determine a close-to-global solution of the NLP (10)–(11) at the center point w_0 by solving problem (13). Let $i = 1$.*
2. *Let $W^s = \{w_0, w_1, w_2, ..., w_{N_2}\} \subset W$ be the subset of points at which a feasible solution of the NLP (10)–(11) has been already determined.*
3. *Find the point $\widetilde{w} \in W^s$ that is most close to the point w_i, i.e. $\widetilde{w} = \arg\min_{w \in W^s} \|w - w_i\|$. Let the solution at $\widetilde{w}$ be $U^*(\widetilde{w})$.*
4. *Solve the NLP (10)–(11) at the point w_i with initial guess for the optimization variables set to $U^*(\widetilde{w})$.*
5. *If a solution of the NLP (10)–(11) at the point w_i has been found, mark w_i as feasible and add it to the set W^s. Otherwise, mark w_i as infeasible.*
6. *Let $i = i + 1$. If $i \leq N_1$, go to step 2. Otherwise, terminate.* □

3.2 Computation of Feasible Approximate Solution

Definition 1. *Let $X = \{w_1, w_2, ..., w_L\} \subset \mathbb{R}^n$ be a discrete set. A function $U(x)$ is feasible on X if $G(U(w_i), w_i) \leq 0$, $i \in \{1, 2, ..., L\}$.*

We restrict our attention to a hyper-rectangle $X \subset \mathbb{R}^n$ where we seek to approximate the optimal solution $U^*(x)$ to the mp-NLP (10)–(11). We require that the state space partition is orthogonal and can be represented as a $k-d$ tree [16], [17]. The main idea of the approximate mp-NLP approach is to construct a feasible piecewise linear (PWL) approximation $\widehat{U}(x)$ to $U^*(x)$ on X, where the constituent affine functions are defined on hyper-rectangles covering X. In case of convexity, it suffices to compute the solution of problem (10)–(11) at the 2^n vertices of a considered hyper-rectangle X_0 by solving up to 2^n NLPs. In case of non-convexity, it would not be sufficient to impose the constraints only at the vertices of the hyper-rectangle X_0. One approach to resolve this problem is to include some interior points in addition to the set of vertices of X_0 [11]. These additional points can represent the vertices and the facets centers of one or more hyper-rectangles contained in the interior of X_0. Based on the solutions at all points, a feasible local linear approximation $\widehat{U}_0(x) = K_0x + g_0$ to the optimal solution $U^*(x)$, valid in the whole hyper-rectangle X_0, is determined by applying the following procedure:

Procedure 2. (computation of approximate solution)
Suppose A1–A3 hold, and consider any hyper-rectangle $X_0 \subseteq X_f$ *with vertices* $\Theta^0 = \left\{\theta_1^0, \theta_2^0, ..., \theta_M^0\right\}$ *and center point* w_0*. Consider also the hyper-rectangles* $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$ *with vertices respectively* $\Theta^j = \left\{\theta_1^j, \theta_2^j, ..., \theta_M^j\right\}$, $j = 1, 2, ..., N_j$*. Suppose* $X_0^1 \subset X_0^2 \subset ... \subset X_0^{N_j}$*. For each of the hyper-rectangles* X_0 *and* $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$*, determine a set of points that belongs to its facets and denote this set* $\Psi^j = \left\{\psi_1^j, \psi_2^j, ..., \psi_{N_\Psi}^j\right\}$, $j = 0, 1, 2, ..., N_j$*. Define the set of all points* $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$*, where* $w_i \in \left\{\bigcup_{j=0}^{N_j} \Theta^j\right\} \cup \left\{\bigcup_{j=0}^{N_j} \Psi^j\right\}$, $i = 1, 2, ..., N_1$*. Compute* K_0 *and* g_0 *by solving the following NLP:*

$$\min_{K_0, g_0} \sum_{i=0}^{N_1} (J(K_0w_i + g_0, w_i) - V^*(w_i) + \mu \left\|K_0w_i + g_0 - U^*(w_i)\right\|_2^2) \tag{14}$$

subject to:

$$G(K_0w_i + g_0, w_i) \leq 0, \; i \in \{0, 1, 2, ..., N_1\} \tag{15}$$

where N_1 *is the total number of points.* □

In order to give an appropriate initialization of the NLP problem (14)–(15) for the region X_0, the already computed solutions of this problem in some of the neighbouring regions can be used as initial guesses.

3.3 Estimation of Error Bounds

Suppose that a state feedback $\widehat{U}_0(x)$ that is feasible in X_0 has been determined by applying Procedure 2. Then it follows that the sub-optimal cost $\widehat{V}(x) = J(\widehat{U}_0(x), x)$ is an upper bound on $V^*(x)$ in X_0, such that for all $x \in X_0$ we have:

$$0 \leq \widehat{V}(x) - V^*(x) \leq \varepsilon_0 \tag{16}$$

As already mentioned, the cost function J can be non-convex with multiple local minima. Therefore, in (16) $V^*(x)$ denotes a close-to-global solution. The following procedure can be used to obtain an estimate $\widehat{\varepsilon}_0$ of the maximal approximation error ε_0 in X_0.

Procedure 3. (computation of the error bound)
Consider any hyper-rectangle $X_0 \subseteq X_f$ with vertices $\Theta^0 = \{\theta_1^0, \theta_2^0, ..., \theta_M^0\}$ and center point w_0. Consider also the hyper-rectangles $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$ with vertices respectively $\Theta^j = \left\{\theta_1^j, \theta_2^j, ..., \theta_M^j\right\}$, $j = 1, 2, ..., N_j$. Suppose $X_0^1 \subset X_0^2 \subset ... \subset X_0^{N_j}$. For each of the hyper-rectangles X_0 and $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$, determine a set of points that belongs to its facets and denote this set $\Psi^j = \left\{\psi_1^j, \psi_2^j, ..., \psi_{N_\Psi}^j\right\}$, $j = 0, 1, 2, ..., N_j$. Define the set of all points $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$, where $w_i \in \left\{\bigcup_{j=0}^{N_j} \Theta^j\right\} \cup \left\{\bigcup_{j=0}^{N_j} \Psi^j\right\}$, $i = 1, 2, ..., N_1$. Compute an estimate $\widehat{\varepsilon}_0$ of the error bound ε_0 through the following maximization:

$$\widehat{\varepsilon}_0 = \max_{i \in \{0,1,2,...,N_1\}} (\widehat{V}(w_i) - V^*(w_i)) \tag{17}$$

where N_1 is the total number of points. □

3.4 Procedure and Heuristic Rules for Splitting a Region

The following procedure is applied to determine the best split of a region X_0 for which a feasible local state feedback $\widehat{U}_0(x)$ is found, but the required accuracy is not achieved.

Procedure 4. (determination of the best split of a region)
Consider a hyper-rectangle X_0 and suppose that a feasible local state feedback $\widehat{U}_0(x)$ was found by applying Procedure 2. Suppose also that the required accuracy is not achieved. Then, determine the best split of X_0 in the following way:

1. *Let $j = 1$.*
2. *Split X_0 by a hyperplane through its center and orthogonal to the axis x_j. Denote the new hyper-rectangles with X_1^j and X_2^j.*
3. *Compute feasible local state feedbacks $\widehat{U}_1^j(x)$ and $\widehat{U}_2^j(x)$, valid respectively in X_1^j and X_2^j, by applying Procedure 2.*
4. *Compute estimates $\widehat{\varepsilon}_1^j$ and $\widehat{\varepsilon}_2^j$, respectively of the error bounds ε_1^j in X_1^j and ε_2^j in X_2^j, by applying Procedure 3. Let $\widehat{\varepsilon}^j = \widehat{\varepsilon}_1^j + \widehat{\varepsilon}_2^j$.*
5. *Let $j = j + 1$. If $j \leq n$, go to step 2.*
6. *Split X_0 by a hyperplane through its center and orthogonal to the axis x_j where $\widehat{\varepsilon}^j$ is minimal.* □

The following rule is applied when no feasible solution to the NLP problem (10)–(11) was found at some of the points $w_i \in W$, $w_i \neq w_0$, where the set $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$ is defined in Procedure 1.

Heuristic splitting rule 1. (handling infeasibility)
Consider the following two cases:

1. *The set of the feasible points in X_0 includes the center point w_0 and some of the points $w_i \in W$, $w_i \neq w_0$ (the set $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$ is defined in Procedure 1). Then, split X_0 into two types of hyper-rectangles by hyperplanes containing some of the feasible points $w_i \in W$:*
   - i. *Hyper-rectangles $X_1^f, X_2^f, ..., X_{N_f}^f$ containing only feasible points.*
   - ii. *Hyper-rectangles $X_1^{nf}, X_2^{nf}, ..., X_{N_{nf}}^{nf}$ containing some infeasible points.*

   Denote the number of the new hyper-rectangles $N_s = N_f + N_{nf}$. The optimal choice of dividing hyperplanes is the one which minimizes the number N_s of the new hyper-rectangles.
2. *The center point w_0 of X_0 is the only feasible point. Then, split X_0 on all state space axes by hyperplanes through w_0.* □

The following rule is applied when there is no feasible solution to the NLP problem (10)–(11) at the center point w_0 of the hyper-rectangle X_0.

Heuristic splitting rule 2. (handling infeasibility)
If there is no feasible solution of the NLP (10)–(11) at the center point w_0 of X_0, split the hyper-rectangle X_0 by a hyperplane through w_0 and orthogonal to an arbitrary axis. □

The following rule is used when the NLP problem (14)–(15) in Procedure 2 has no feasible solution.

Heuristic splitting rule 3. (handling infeasibility)
If the NLP problem (14)–(15) in Procedure 2 is infeasible, split the hyper-rectangle X_0 by a hyperplane through its center and orthogonal to an arbitrary axis. □

3.5 Approximate Algorithm for Explicit Solution of Mp-NLPs

Assume the tolerance $\overline{\varepsilon} > 0$ of the cost function approximation error is given. The following algorithm is proposed to design explicit NMPC controller for constrained nonlinear systems:

Algorithm 1. (approximate explicit mp-NLP)

1. Initialize the partition to the whole hyper-rectangle, i.e. $P = \{X\}$. Mark the hyper-rectangle X as unexplored.
2. Select any unexplored hyper-rectangle $X_0 \in P$. If no such hyper-rectangle exists, the algorithm terminates successfully.

3. Compute a solution to the NLP (10)–(11) at the center point w_0 of X_0 by applying Procedure 1. If the NLP has a feasible solution, go to step 4. Otherwise, split the hyper-rectangle X_0 into two hyper-rectangles X_1 and X_2 by applying *the heuristic splitting rule 2*. Mark X_1 and X_2 unexplored, remove X_0 from P, add X_1 and X_2 to P, and go to step 2.
4. Define a set of hyper-rectangles $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$ contained in the interior of X_0. For each of the hyper-rectangles X_0 and $X_0^j \subset X_0$, $j = 1, 2, ..., N_j$, in addition to its vertices, determine a set of points that belongs to its facets. Denote the set of all points (including the center point w_0) with $W = \{w_0, w_1, w_2, ..., w_{N_1}\}$.
5. Compute a solution to the NLP (10)–(11) for x fixed to each of the points w_i, $i = 1, 2, ..., N_1$ of the set W by applying Procedure 1. If all NLPs have a feasible solution, go to step 7. Otherwise, go to step 6.
6. Compute the size of X_0 using some metric. If it is smaller than some given tolerance, mark X_0 infeasible and explored and go to step 2. Otherwise, split the hyper-rectangle X_0 into hyper-rectangles X_1, X_2,..., X_{N_s} by applying *the heuristic splitting rule 1*. Mark X_1, X_2,..., X_{N_s} unexplored, remove X_0 from P, add X_1, X_2,..., X_{N_s} to P, and go to step 2.
7. Compute an affine state feedback $\widehat{U}_0(x)$ using Procedure 2, as an approximation to be used in X_0. If no feasible solution was found, split the hyper-rectangle X_0 into two hyper-rectangles X_1 and X_2 by applying *the heuristic splitting rule 3*. Mark X_1 and X_2 unexplored, remove X_0 from P, add X_1 and X_2 to P, and go to step 2.
8. Compute an estimate $\widehat{\varepsilon}_0$ of the error bound ε_0 in X_0 by applying Procedure 3. If $\widehat{\varepsilon}_0 \leq \overline{\varepsilon}$, mark X_0 as explored and feasible and go to step 2. Otherwise, split the hyper-rectangle X_0 into two hyper-rectangles X_1 and X_2 by applying Procedure 4. Mark X_1 and X_2 unexplored, remove X_0 from P, add X_1 and X_2 to P, and go to step 2. □

In contrast to the conventional MPC based on real-time optimization, the explicit MPC makes the rigorous verification and validation of the controller performance much easier [11]. Hence, problems due to lack of convexity and numerical difficulties can be addressed during the design and implementation.

4 Application of the Approximate Explicit NMPC Approach to Compressor Surge Control

Consider the following 2-nd order compressor model [10], [18] with x_1 being normalized mass flow, x_2 normalized pressure and u normalized mass flow through a close-coupled valve in series with the compressor:

$$\dot{x}_1 = B(\Psi_e(x_1) - x_2 - u) \tag{18}$$

$$\dot{x}_2 = \frac{1}{B}(x_1 - \Phi(x_2)) \tag{19}$$

The following compressor and valve characteristics are used:

$$\Psi_e(x_1) = \psi_{c0} + H \left(1 + 1.5 \left(\frac{x_1}{W} - 1 \right) - 0.5 \left(\frac{x_1}{W} - 1 \right)^3 \right) \tag{20}$$

$$\Phi(x_2) = \gamma sign(x_2) \sqrt{|x_2|} \tag{21}$$

with $\gamma = 0.5$, $B = 1$, $H = 0.18$, $\psi_{c0} = 0.3$ and $W = 0.25$. Like in [10], the control objective is to avoid surge. This is formulated as [10]:

$$\begin{aligned} J(U, x(t)) = & \sum_{k=0}^{N-1} \left[\alpha (x_{t+k|t} - x^*)^T (x_{t+k|t} - x^*) + k u_{t+k}^2 \right] + R v^2 \\ & + \beta (x_{t+N|t} - x^*)^T (x_{t+N|t} - x^*) \end{aligned} \tag{22}$$

with $\alpha, \beta, k, R \geq 0$ and the set-point $x_1^* = 0.4$, $x_2^* = 0.6$ corresponds to an unstable equilibrium point. We have chosen $\alpha = 1$, $\beta = 0$ and $k = 0.08$. The horizon is chosen as $T = 12$, which is split into $N = 15$ equal-sized intervals, leading to a piecewise constant control input parameterization. Valve capacity requires the following constraint to hold:

$$0 \leq u(t) \leq 0.3 \tag{23}$$

The pressure constraint:

$$x_2(t) \geq 0.4 - v \tag{24}$$

avoids operation too far left of the operating point. The variable $v \geq 0$ is a slack variable introduced in order to avoid infeasibility and $R = 8$ is a large weight. Numerical analysis of the cost function shows that it is non-convex [10]. It can be seen that this NMPC problem formulation differs from that in section 2 in the absence of a terminal constraint and in the use of a slack variable.

The NLP (10)–(11) has 16 free variables and 46 constraints, while the NLP (14)–(15) has 46 free variables and 811 constraints. One internal region $X_0^1 \subset X_0$ is used in Procedures 1, 2 and 3. In (14), it is chosen $\mu = 10$ and the control input only at the first sample is considered. The approximation tolerance is chosen to depend on X_0 such that:

$$\overline{\varepsilon}(X_0) = \max(\overline{\varepsilon}_a, \overline{\varepsilon}_r V_{\min}^*) \tag{25}$$

where $\overline{\varepsilon}_a = 0.0001$ and $\overline{\varepsilon}_r = 0.02$ can be interpreted as absolute and relative tolerances, respectively, and $V_{\min}^* = \min\limits_{x \in X_0} V^*(x)$. Here, $V^*(x)$ denotes a close-to-global solution.

The partition of the approximate explicit NMPC controller is shown in Fig.1. It has 595 regions and 12 levels of search. With one scalar comparison required at each level of the $k-d$ tree, 12 arithmetic operations are required in the worst case to determine which region the state belongs to. Totally, 16 arithmetic operations are needed in real-time to compute the control input and 1368 numbers needs to be stored in real-time computer memory. The off-line computation of the

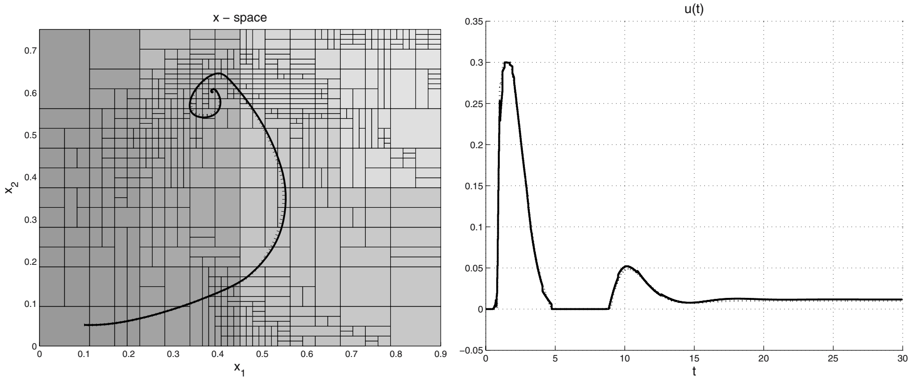


Fig. 1. State space partition of the approximate explicit NMPC (left) and the control input for $x(0) = [0.1\ 0.05]^T$ (right). The solid curves are with the approximate explicit NMPC and the dotted curves are with the exact NMPC.

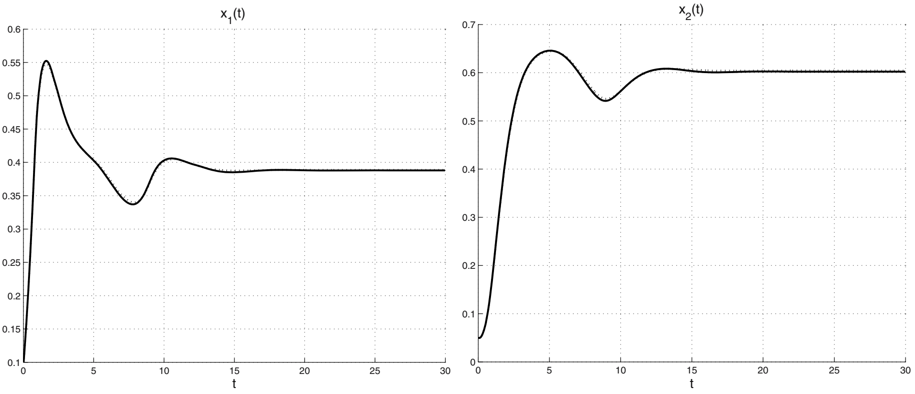


Fig. 2. State variables x_1 and x_2: the solid curve is with the approximate explicit NMPC and the dotted curve is with the exact NMPC

partition is performed on a 1.8 GHz AMD Athlon(tm)XP 2200+, where the CPU time needed for solving the NLP (10)–(11) at a single point in the state space is about 3 sec and the CPU time necessary to solve the NLP (14)–(15) for a single region is about 210 sec.

The performance of the closed-loop system is simulated for initial condition $x(0) = [0.1\ 0.05]^T$ and with sampling time $T_s = 0.02$. Euler integration with step size T_s is applied to solve the ordinary differential equations (18)–(19). The resulting closed-loop response is depicted in the state space (Fig.1 (left)), as well as trajectories in time (Fig.1 (right) and Fig.2). In Fig.1 and Fig.2 the exact NMPC solution is also shown, which at each time step is determined by comparing the local minima of the cost function (22) corresponding to several initial guesses for the optimization variables.

5 Conclusions

In this paper, practical computational methods for constructing approximate explicit PWL solutions of NMPC problems are developed. They represent an extension of the approximate approach in [11] since they provide some additional mechanisms to practically handle also the case of non-convexity of the resulting mp-NLP problem. As a case study, the design of an approximate explicit NMPC for compressor surge control is considered.

References

[1] Keerthi S S, Gilbert E G (1988) Optimal infinite horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving horizon approximations. Journal of Optimization Theory and Applications 57:265–293

[2] Mayne D Q, Michalska H (1990) Receding horizon control of nonlinear systems. IEEE Transactions on Automatic Control 35:814–824

[3] Michalska H, Mayne D Q (1993) Robust receding horizon control of constrained nonlinear systems. IEEE Transactions on Automatic Control 38:1623–1633

[4] Chen H, Allgöwer F (1998) A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica 34:1205–1217

[5] Magni L, De Nicolao G, Magnani L, Scattolini R (2001) A stabilizing model-based predictive control for nonlinear systems. Automatica 37:1351–1362

[6] Magni L, Scattolini R (2004) Stabilizing model predictive control of nonlinear continuous time systems. Annual Reviews in Control 28:1-11

[7] Bemporad A, Morari M, Dua V, Pistikopoulos E N (2002) The explicit linear quadratic regulator for constrained systems. Automatica 38:3-20

[8] Chen W H, Ballance D J, Gawthrop P J (2003) Optimal control of nonlinear systems: a predictive control approach. Automatica 39:633–641

[9] Bacic M, Cannon M, Kouvaritakis B (2003) Constrained NMPC via state-space partitioning for input-affine non-linear systems. In: Proceedings of American Control Conference, Denver, Colorado

[10] Johansen T A (2002) On multi-parametric nonlinear programming and explicit nonlinear model predictive control. In: Proceedings of IEEE Conference on Decision and Control, Las Vegas, NV

[11] Johansen T A (2004) Approximate explicit receding horizon control of constrained nonlinear systems. Automatica 40:293-300

[12] Grancharova A, Johansen T A (2005) Explicit min-max model predictive control of constrained nonlinear systems with model uncertainty. In: Proceedings of 16-th IFAC World Congress, Prague, Czech Republic

[13] Fiacco A V (1983) Introduction to sensitivity and stability analysis in nonlinear programming. Academic Press, Orlando, Florida

[14] Bemporad A, Filippi C (2003) Approximate multiparametric convex programming. In: Proceedings of 42th IEEE Conference on Decision and Control, Maui, Hawaii

[15] Mayne D Q, Rawlings J B, Rao C V, Scokaert P O M (2000) Constrained model predictive control: Stability and optimality. Automatica 36:789-814

[16] Bentley J L (1975) Multidimensional binary search trees used for associative searching. Communications of the ACM 18:509–517

[17] Grancharova A, Johansen T A (2002) Approximate explicit model predictive control incorporating heuristics. In: Proceedings of IEEE International Symposium on Computer Aided Control System Design, Glasgow, Scotland, U.K.

[18] Gravdahl J T, Egeland O (1997) Compressor surge control using a close-coupled valve and backstepping. In: Proceedings of American Control Conference, Albuquerque, NM.

Towards the Design of Parametric Model Predictive Controllers for Non-linear Constrained Systems

V. Sakizlis, K.I. Kouramas, N.P. Faísca, and E.N. Pistikopoulos

Centre for Process System Engineering, Department of Chemical Engineering,
Imperial College London Roderic Hill Building, South Kensington Campus,
London SW7 2AZ, UK
v.sakizlis@imperial.ac.uk, k.kouramas@imperial.ac.uk,
n.faisca@imperial.ac.uk, e.pistikopoulos@imperial.ac.uk

Summary. The benefits of parametric programming for the design of optimal controllers for constrained systems are widely acknowledged, especially for the case of linear systems. In this work we attempt to exploit these benefits and further extend the theoretical contributions to multi-parametric Model Predictive Control (mp-MPC) for non-linear systems with state and input constraints. The aim is to provide an insight and understanding of multi-parametric control and its benefits for non-linear systems and outline key issues for ongoing research work.

1 Introduction

The explicit, multi-parametric MPC (mp-MPC) has been extensively investigated for linear systems. Major results have been presented for the discrete-time case ([2]) and recently for the linear, continuous-time systems case ([20]). The key advantage of mp-MPC is that the on-line optimization, typically involved in MPC, can be performed off-line to produce an explicit mapping of the optimal control actions and the objective function in the space of the current states. The on-line implementation of the controller is then reduced to a simple function evaluation ([17]).

Although mp-MPC has received much attention for the linear systems case, there is relatively little progress for the non-linear systems case. Most of the research on MPC has focused in on-line implicit MPC methods that usually rely, for the case of continuous-time systems, on numerical dynamic optimization techniques ([5, 6, 13]) or for the case of discrete-time systems, on on-line Non-linear Optimization ([15, 18]). A first attempt towards the design of approximate, linear mp-MPC controllers for the non-linear MPC problem, is presented in [11] and [12]. The first work proposes a method for a local mp-QP approximation to the continuous-time, mp-NLP control problem. The second work focuses on the MPC problem for non-linear systems with linear state and input constraints, and a quadratic cost.

This work aims to provide an insight and understanding of multi-parametric control and outline the benefits of non-linear mp-MPC. Two approaches are

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 193–205, 2007.
springerlink.com

presented here. The first approach presents a method for obtaining a Piecewise Affine (PWA) approximation to the solution of the non-linear MPC for discrete-time systems, by exploiting the fact that the non-linear MPC problem for non-linear systems with non-linear state and input constraints, and non-linear objective, is a non-linear optimization problem where the input is the optimisation variable and the initial state is the optimisation parameter. This approach is based on recent developments on parametric programming techniques ([7]).

The second approach deals with the explicit solution of the non-linear MPC, for certain classes of non-linear, continuous-time models for which there exists an analytical solution to the dynamic systems arising from the first order optimality conditions. The optimisation is then solved off-line, based on a recently developed multi-parametric, dynamic optimisation algorithm ([20]) - the control law is then derived as an explicit, non-linear function of the states. In both approaches the implementation of the controller is simply reduced to a sequence of function evaluations, instead of solving the on-line, non-linear optimal control problem, which is usually the typical procedure of non-linear MPC. The two procedures are then applied, in the end of this paper, on the classical, constrained Brachistochrone problem to illustrate the key features of the new developments.

2 Piecewise Affine Approximation to the Discrete - Time Non-linear MPC

The main multi-parametric programming problem that is frequently encountered in various engineering applications, including non-linear MPC, is the following

$$z(\theta) = \min_{x} f(x) \tag{1a}$$

$$\text{s.t. } g(x) \leq b + F\theta \tag{1b}$$

$$x \in \mathcal{X} \tag{1c}$$

$$\theta \in \Theta \tag{1d}$$

where x is a vector of continuous variables, f a scalar, continuously differentiable function of x, g a vector of continuously differentiable functions of x, b a constant vector, F are constant matrices of appropriate dimensions, θ a vector of parameters and $\mathcal{X}$ and Θ are compact subsets of the x and θ-space respectively. A representative example of this problem is the discrete-time constrained linear quadratic regulator problem ([2]), where x is the sequence of control inputs over a finite time horizon, $f(x)$ is a strictly convex quadratic function of x, $g(x)$ is a linear function of x, θ is the initial state and $\mathcal{X}$ and Θ are convex, polyhedral sets. Although, solving (1) has been proved to be a difficult task, an algorithm was presented recently in [7, 8] which can obtain a linear, PWA approximation to $z(\theta)$ with a prescribed accuracy. The value function $z(\theta)$ as well as the optimization variable $x(\theta)$ are linear, PWA function of θ. Given a value of θ then $z(\theta)$ and $x(\theta)$ can be obtained by simple function evaluations.

The mathematical framework for the discrete-time nonlinear MPC can be shortly summarised as the following constrained non-linear programming (NLP) problem ([15, 18])

$$z^o(x_t) = \min_U J(U, x_t) \tag{2a}$$

$$\text{s.t } h(U, x_t) \le b \tag{2b}$$

$$U_L \le U \le U_U \ , \ x_L \le x \le x_U \tag{2c}$$

where x_t is the state at the current time instant, $U = \{u_t, u_{t+1}, \ldots, u_{t+N-1}\}$ is the sequence of control inputs over the prediction horizon N, $J(U, x_t)$ is a scalar objective function, $h(U, x_t)$ is a vector of non-linear functions, U_L, U_U are lower and upper bounds for U and x_L and x_U are lower and upper bounds for x. The functions $J(U, x_t)$ and $h(U, x_t)$ are generally non-linear, although the analysis that follows can be applied for the linear case as well, and may include any terminal cost function and terminal constraints respectively to ensure stability ([15]).

Transforming the NLP (2) to (1) can been done in two steps. First, if $J(U, x_t)$ is only a function of U then simply replace (2a) by simply $J(U)$ and the objective function of (2) is the same with (1). Otherwise, introduce a new scalar $\epsilon \in \mathbb{R}$ and transform (2) into the following NLP

$$\bar{z}(x_t) = \min_U \epsilon \tag{3a}$$

$$\text{s.t. } J(U, x_t) \le \epsilon \ , \ h(U, x_t) \le b \tag{3b}$$

$$U_L \le U \le U_U \ , \ x_L \le x \le x_U \tag{3c}$$

or simply to

$$\bar{z}(x_t) = \min_U \epsilon \tag{4a}$$

$$\text{s.t. } \bar{h}(U, x_t) \le \bar{b} \ , \ U_L \le U \le U_U \ , \ x_L \le x \le x_U \tag{4b}$$

where $\bar{h}(U, x_t) = [J(U, x_t) \ h^T(U, x_t)]^T$ and $\bar{b} = [\epsilon \ b^T]^T$.

A simple but conservative way to solve the above problem is by linearising the inequalities in (4) and solving off-line the linearized problem. More specifically, choose an initial x_t^* and solve (4) to acquire U^*. Then linearize the inequalities in (4) over x_t^*, U^* to obtain the following approximating, mp-LP problem over x_t and U

$$\breve{z}(x_t) = \min_U \epsilon \tag{5}$$

$$\bar{h}(U^*, x_t^*) + \frac{\partial \bar{h}(U^*, x_t^*)}{\partial U}(U - U^*) \le \bar{b} - \frac{\partial \bar{h}(U^*, x_t^*)}{\partial x_t}(x_t - x_t^*) \tag{6}$$

$$U_L \le U \le U_U \ , \ x_L \le x \le x_U \tag{7}$$

which now of form (1), where x is U and θ is x_t. The solution to the mp-LP (5) is a linear, PWA function of x_t, $\breve{z}(x_t)$ ([8]). The control sequence $U(x_t)$ is also a linear PWA function of x_t and hence the first control input $u_t(x_t)$ of the control sequence, is a linear PWA function of x_t. The solutions $\breve{z}(x_t)$ and $u_t(x_t)$ are only valid in a critical region $\mathcal{CR}$ of x_t which is defined as the feasible region of x_t associated with an optimal basis ([7, 8]). In the next step choose x_t outside the region of $\mathcal{CR}$ and repeat the procedure until the space of interest is covered.

A different procedure for transforming the NLP (4) into (1) is obtained if one considers that a nonlinear function $\bar{h}_i(U, x_t)$ consists of the addition, subtraction, multiplication and division of five simple non-linear functions of $U_i, x_{t,i}$ ([7, 9, 16, 21]): a) linear $f_L(U_i, x_{t,i})$, b) bilinear $f_B(U_i, x_{t,i})$, c) fractional $f_F(U_i, x_{t,i})$, d) exponential $f_{exp}(U_i, x_{t,i})$ and e) univariate concave $f_{uc}(U_i, x_{t,i})$ functions of $U_i, x_{t,i}$. If $f_L(U_i, x_{t,i})$, $f_B(U_i, x_{t,i})$, $f_F(U_i, x_{t,i})$, $f_{exp}(U_i, x_{t,i})$ and $f_{uc}(U_i, x_{t,i})$ are simply functions of U_i then they are simply retained without further transforming them. If, however, they are functions of both $U_i, x_{t,i}$ then a new variable is assigned for each of the non-linear functions and a convex approximating function can be obtained which is linear with respect to $x_{t,i}$. For example consider the non-linear inequality

$$\sin(U_i) + \frac{1}{U_i x_{t,j} + 1} \leq 0 \tag{8}$$

The term $\sin U_i$ is preserved without further manipulation as it is a non-linear function of U_i. Set $w = U_i x_{t,j} + 1$. This equality contains a bilinear term of $U_i x_{t,j}$. A convex approximation can then be obtain for this equality by employing the McCormick ([7, 9, 16]) over- and underestimators for bilinear functions

$$-w + x_{t,j}^L U_i \leq U_i^L x_{t,j}^L - U_i^L x_{t,j} \ , \ -w + x_{t,j}^U U_i \leq U_i^U x_{t,j}^U - U_i^U x_{t,j} \tag{9a}$$

$$w - x_{t,j}^U U_i \leq -U_i^L x_{t,j}^U + U_i^L x_{t,j} \ , \ w - x_{t,j}^L U_i \leq -U_i^U x_{t,j}^L + U_i^U x_{t,j} \tag{9b}$$

Moreover, (8) can be re-written as $\sin U_i + 1/w \leq 0$. It can be easily noticed that the above inequality and (9) have the same form with the inequalities in (1). Convex approximations to non-linear functions have been extensively investigated in [7, 9, 16, 21]. Since it is difficult to fully present the theory of convex approximations in this paper due to lack of space, the interested reader can look in the relevant literature and the references within, cited here in [7, 9, 16, 21].

Following the above procedure, one can transform the NLP (4) to the mp-NLP problem (1) as following

$$\hat{z}(x_t) = \min_{U,W} \epsilon \tag{10a}$$

$$\text{s.t. } \hat{h}(U, W) \leq \hat{b} + \hat{F} x_t \tag{10b}$$

where W is the vector of all new variables w which were introduced to replace the non-linear terms $f_L(U_i, x_{t,i})$, $f_B(U_i, x_{t,i})$, $f_F(U_i, x_{t,i})$, $f_{exp}(U_i, x_{t,i})$ and $f_{uc}(U_i, x_{t,i})$. The algorithm in [7, 8] can then be used to solve the above problem and obtain a linear, PWA approximation to the non-linear MPC problem for u_t. The control input u_t as well as the value function $\hat{z}(x_t)$ are both PWA function of x_t hence a feedback control policy is obtained.

The main disadvantage of the above method is that both problems (5) and (10) only provide an approximation for the optimal solution of (2). This could result to violation of the constraints of (2), although the constraints in both (5) and (10) are satisfied, thus resulting into state and input constraints violation for the system. However, as far as the authors are aware of, there is currently no alternative multi-parametric MPC method which can guarantee constraint satisfaction for non-linear, discrete-time systems, since most methods rely on the approximation of the initial non-linear programming problem (2). An alternative method, for obtaining the optimal solution and guarantee constraint satisfaction is to address the problem in continuous-time and not in discrete-time. This will be shown in the next section.

3 Multi-parametric Non-linear Optimal Control Law for Continuous - Time Dynamic Systems

It is a common practise to deal with the problem of non-linear MPC in discrete time by transforming the continuous-time optimal control problem involved into a discrete-time one. The interest of the relevant research has long being focused on solving the discrete-time non-linear MPC problem. However, the continuous-time case remain of great importance since in practise most of the systems of interest are continuous-time. In this section a novel approach is presented that derives off-line the optimal control law in a continuous-time optimal control problem with state and input constraints. More specifically consider the following continuous-time, optimal control problem

$$\hat{\phi} = \min_{x(t),u(t)} \phi(x_{t_f,t_f}) \tag{11a}$$

$$\text{s.t. } \dot{x} = f(x(t), u(t), t) \tag{11b}$$

$$\psi^g(x_{t_f}) \leq 0 \tag{11c}$$

$$g(x(t), u(t)) \leq 0 \tag{11d}$$

$$x(t_0) = x_0 \tag{11e}$$

$$t_0 \leq t \leq t_f \tag{11f}$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ are the systems states, $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ are the control variables, $g : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^q$ are the path constraints and $\psi^g : \mathbb{R}^n \to \mathbb{R}^{Q_g}$ is the terminal constraint. The objective function $\phi : \mathbb{R}^n \times \mathbb{R} \to \mathbb{R}$ is a continuous, differentiable, non-linear function of $x(t_f)$ at the final time t_f.

The objective is to obtain the solution of problem (11) i.e. the optimal value of the performance index $\hat{\phi}$ and the optimal profiles of the control inputs $u(t)$, as explicit function of the initial states x_0. Hence, by treating x_0 as a parameter, the optimal control problem (11) is recast as a multi-parametric Dynamic Optimization (mp-DO) problem where $\hat{\phi}$ is the value function, $u(t)$ the optimal control profiles and x_0 is the parameter of the problem. Problem (11) has been thoroughly studied for the case of the continuous-time, linear quadratic optimal control problem ([20]), however this is the first time this problem is treated for non-linear systems. Our purpose here is to extend the results of [20] for the continuous-time, non-linear optimal control problem described in (11).

Let define the order of a path constraint before we proceed

Definition 1. *The constraint $g_i(x,u)$ is said to be of order $\hat{l} \geq 1$ with respect to the dynamics, if*

$$\frac{\partial g_i(x,u)^j}{\partial u_k} = 0,\ j = 1,2,\ldots,\hat{l}-1,\ k = 1,\ldots,m$$

$$\frac{\partial g_i(x,u)^{\hat{l}}}{\partial u_k} \neq 0,\ \textit{for at least one } k,\ k = 1,\ldots,m$$

where the index j denotes time derivatives. The constraint $g_i(x,u)$ is said to be of zero-th order if

$$\frac{\partial g_i(x,u)}{\partial u_k} \neq 0,\ \textit{for at least one } k,\ k = 1,\ldots,m$$

The Karush-Kuhn-Tucker conditions for the optimal control problem (11) derived from the Euler-Lagrange equations and for $\hat{l} \geq 1$ are given as ([1, 4, 14])

ORDINARY DIFFERENTIAL EQUATION (ODE)

$$\dot{x} = f(x(t),u(t),t),\ t_0 \leq t \leq t_f \tag{12}$$

BOUNDARY CONDITIONS FOR THE ADJOINTS

$$x(t_0) = x_0 \tag{13}$$

$$\lambda(t_f) = \left(\frac{\partial \phi(x_{t_f},t_f)}{\partial x(t_f)}\right)^T + \left(\frac{\partial \psi^g(x(t_f))}{\partial x(t_f)}\right)^T \cdot \nu \tag{14}$$

COMPLEMENTARITY CONDITIONS

$$0 = \nu_j \cdot \psi_j^g(x(t_f)) \tag{15}$$

$$\nu_j \geq 0,\ j = 1,\ldots,Q_g \tag{16}$$

ADJOINT DIFFERENTIAL SYSTEM

$$\mu_i(t) \geq 0,\ g_i(x(t),u(t)) \cdot \mu_i(t) = 0,\ i = 1,\ldots,q \tag{17}$$

$$\dot{\lambda}(t) = -\left(\frac{\partial f(x(t),u(t),t)}{\partial x(t)}\right)^T \cdot \lambda(t) - \sum_{i=1}^{q} \left(\frac{\partial g_i^{\hat{l}_i}(x(t),u(t))}{\partial x(t)}\right)^T \cdot \mu_i(t) \tag{18}$$

$$0 = \left(\frac{\partial f(x(t), u(t), t)}{\partial u(t)} \right)^T \cdot \lambda(t) + \sum_{i=1}^{q} \left(\frac{\partial g_i^{\hat{l}_i}(x(t), u(t))}{\partial u(t)} \right)^T \cdot \mu_i(t) \tag{19}$$

$$t_0 \leq t \leq t_f \tag{20}$$

$$\text{Assume: } t_{n_{kt}+n_{kx}+1} = t_f, \text{ and Define:} \tag{21}$$

$$t_{kt} \equiv \text{Entry point if } \mu_j(t_{kt}^-) = 0,\ \mu_j(t_{kt}^+) \geq 0,\ k = 1, 2, \ldots, n_{kt} \tag{22}$$

$$t_{kx} \equiv \text{Exit point if } \mu_j(t_{kx}^+) = 0,\ \mu_j(t_{kx}^-) \geq 0,\ k = 1, 2, \ldots, n_{kx} \tag{23}$$

$$\text{For at least one } j = 1, 2, \ldots, q \tag{24}$$

JUNCTION CONDITIONS (ENTRY POINT)

$$0 = g_i^j(x(t_{kt}), u(t_{kt})),\ j = 0, \ldots, \hat{l}_i - 1 \tag{25}$$

$$0 = g_i^{\hat{l}_i}(x(t_{kt}^+), u(t_{kt}^+)),\ k = 1, 2, \ldots n_{kt},\ i = 1, \ldots, q \tag{26}$$

JUMP CONDITIONS (ENTRY POINT - EXIT POINT)

$$\lambda(t_{kt}^+) = \lambda(t_{kt}^-) + \sum_{i=1}^{q} \sum_{j=0}^{\hat{l}_i - 1} \left(\frac{\partial g_i^j(x(t_{kt}), u(t_{kt}))}{\partial x(t_{kt})} \right)^T \cdot \varphi_{j,i}(t_{kt}) \tag{27}$$

$$H(t_{kt'}^+) = H(t_{kt'}^-),\ k = 1, 2, \ldots, n_{kt} \tag{28}$$

$$\lambda(t_{kx}^+) = \lambda(t_{kx}^-) \tag{29}$$

$$H(t_{kx'}^+) = H(t_{kx'}^-),\ k = 1, 2, \ldots, n_{kx} \tag{30}$$

$$H(t) = \dot{x}(t)\lambda(t) + g(x(t), u(t))^T \cdot \mu(t) \tag{31}$$

$$t_{k(t,x)} = \{\min(t_{k(t,x)'}, t_f) \vee \max(t_{k(t,x)'}, t_0)\} \tag{32}$$

where $\lambda(t) \in \mathbb{R}^n$ is the vector of adjoint (co-state) variables, $\mu(t) \in \mathbb{R}^q$ is the vector of Lagrange multipliers associated with the path constraints, $\nu(t) \in \mathbb{R}^{Q_g}$ is the vector of Lagrange multipliers of the end-point constraints, $\varphi_i \in \mathbb{R}^{\hat{l}_i}$, $i = 1, \ldots, q$ are the Lagrange multipliers linked with the jump conditions and $H(t)$ is the Hamiltonian function of the system. The time points where the jump conditions apply are called *corners* or *switching points*. The time intervals $t \in [t_k, t_{k+1}]$, $k = 1, \ldots, (n_{kt} + n_{kx})$ between two consecutive corners are termed as *constrained* or *boundary arcs* if at least one constraint is active or *unconstrained arcs* otherwise, where n_{kt} is the maximum number of entry points that may exist in the problem and n_{kx} is the maximum number of exit points.

Remark 1. For a zeroth order constraint, equations (25),(26) are omitted, (27), (28) are written as $\lambda(t_{kt}^{+}) = \lambda(t_{kt}^{-})$ and $H(t_{kt}^{+}) = H(t_{kt}^{-})$ respectively and $\varphi = 0$.

The following assumption is necessary for the analysis that will follow.

Assumption 3.1. *There exist an analytical solution to the differential algebraic equation (DAE) system arising from* (12), (18) *and* (19) *with boundary conditions the equations in* (13), (14), (25), (26), (27) *and* (29).

If the above assumption holds then $x(t, t^k, x_0)$, $\lambda(t, t^k, x_0)$, $\mu(t, t^k, x_0)$, $u(t, t^k, x_0)$ and $\xi(t^k, x_0) = [x_f^T \; \lambda_0^T \; \mu^T(t_1) \; \ldots \; \mu^T(t_{n_{kt}}) \; \varphi^T(t_1) \; \ldots \; \varphi^T(t_{n_{kt}}) \; \nu^T]$ are explicit, non-linear functions of time t, the switching points $t^k = \{t_1 \; t_2 \; \ldots \; t_{n_{kt}} + t_{n_{kx}}\}$ $\equiv \{t_{1t} \; t_{1x} \; t_{2t} \; \ldots \; t_{n_{kx}}\}$ and the initial condition x_0. This allows the derivation of the optimal profiles of the control inputs in terms of x_0 and the determination of the compact regions in the space of the initial conditions where these functions hold.

In order to obtain the optimal control profiles the following algorithm can be followed:

Algorithm 3.1

1: Define an initial region CR^{IG} in which problem (11) is going to be solved
2: Select a realization in the parameter space of x_0 and compute the optimal number of switching points and (constrained and/or unconstrained) arcs for these points by solving the DO problem (12)-(32).
3: Given the sequence of switching points and considering x_0 as a free parameter, solve analytically the DAE system arising from (12), (18) and (19) with boundary conditions the equations in (13), (14), (25), (26), (27) and (29).to obtain, first $\hat{\xi}(t^k, x_0)$ and then the differential states $\lambda(t, \hat{t^k}, x_0)$, $\hat{x}(t, t^k, x_0)$, $\hat{\mu}(t, t^k, x_0)$ and finally the algebraic variables $\hat{u}(t, t^k, x_0)$.
4: Substitute the values of $\hat{\xi}(t^k, x_0)$, $\lambda(t, \hat{t^k}, x_0)$, $\hat{x}(t, t^k, x_0)$, $\hat{\mu}(t, t^k, x_0)$ and $\hat{u}(t, t^k, x_0)$ in the equations (28), (30) and (32) and solve the new system of non-linear, algebraic equations to obtain t^k as an explicit function of the free parameter x_0 and call it $t^k(x_0)$.
5: Substitute $t^k(x_0)$ into the expression of $u(t, t^k(x_0), x_0)$ to obtain the optimal parametric control profile.
6: Compute the critical region CR where the optimal parametric control profile is valid.
7: If CR is not empty then select a new initial condition x_0 outside CR and go to Step 2 else stop

The algorithm starts with the definition of the space CR^{IR} of initial conditions x_0, in which the mp-DO problem is going to be solved. In step 2 the switching points and the corresponding arcs and active constraints are obtained by solving the DO (12)-(32) for a fixed value of x_0. In step 3 the DAE system that consists of the system's dynamic model and the optimality conditions corresponding to the switching points and active constraints, derived in step 2, is solved symbolically to obtain the optimal profiles of $\hat{\xi}(t^k, x_0)$, $\lambda(t, \hat{t^k}, x_0)$, $\hat{x}(t, t^k, x_0)$, $\hat{\mu}(t, t^k, x_0)$ and $\hat{u}(t, t^k, x_0)$. The vector $t^k(x_0)$ is calculated in the step 4 by solving symbolically the non-linear, algebraic equalities of the Jump conditions (28), (30) and (32). In step 5 the optimal parametric control profile is obtained by substituting $t^k(x_0)$ into $\hat{u}(t, t^k, x_0)$. Finally the critical region in which the optimal control profile is valid, is calculated in step 6, following the procedure which will be described in the following. The algorithm then repeats the procedure until the whole initial region CR^{IR} is covered.

A critical region CR in which the optimal control profiles are valid, is the region of initial conditions x_0 where the active and inactive constraints, obtained in step 2 of algorithm 3.1, remain unaltered ([20]). Define the set of inactive constraints $\breve{g}$, the active constraints $\tilde{g}$ and $\tilde{\hat{\mu}} > 0$ the Lagrange multipliers associated with the active constraints $\tilde{g}$; obviously the Lagrange multipliers μ associated with the inactive constraints are 0. The critical region CR is then identified by the following set of inequalities

$$CR \triangleq \{x_0 \in \mathbb{R}^n \mid \breve{g}(\hat{x}(t, t^k(x_0), x_0), \hat{u}(t, t^k(x_0), x_0)) < 0 \ , \ \tilde{\hat{\mu}}(t, t^k(x_0), x_0) > 0 \\ \tilde{\nu}(t, t^k(x_0), x_0) > 0\} \tag{33}$$

In order to characterize CR one has to obtain the boundaries of the set described by inequalities (33). These boundaries obviously are obtained when each of the linear inequalities in (33) is critically satisfied. This can be achieved by solving the following parametric programming problems, where time t is the variable and x_0 is the parameter.

- Take first the inactive constraints through the complete time horizon and derive the following parametric expressions:

  $$\breve{G}_i(x_0) = \max_t \{\breve{g}_i(\hat{x}(t, t^k(x_0), x_0), \hat{u}(t, t^k(x_0), x_0)) | t \in [t_0, t_f]\}, \ i = 1, \ldots, \breve{q} \tag{34}$$

  where $\breve{q}$ is the number of inactive constraints.
- Take the path constraints that have at least one constrained arc $[t_{i,\tilde{k}t}, t_{i,\tilde{k}x}]$ and obtain the following parametric expression

  $$\tilde{G}_i(x_0) = \max_t \{\tilde{g}_i(\hat{x}(t, t^k(x_0), x_0), \hat{u}(t, t^k(x_0), x_0)) | t \in [t_0, t_f]\} \wedge \{t \not\in [t_{i,\tilde{k}t}, t_{i,\tilde{k}x}]]\} \tag{35}$$

  $$k = 1, 2, \ldots, n_{i,\tilde{k}t}, \ i = 1, 2, \ldots, \tilde{q}$$

  where $n_{i,\tilde{k}t}$ is the total number of entry points associated with the ith active constraint and $\tilde{q}$ is the number of active constraints.
- Finally, take the multipliers of the active constraints and obtain the following parametric expressions

  $$\breve{\mu}(x_0) = \min_t \{\tilde{\hat{\mu}}(t, t^k(x_0), x_0) | t = t_{i,kt} = t_{i,kx}, k = 1, 2, \ldots, n_{i,kt}\}, \ i = 1, 2, \ldots, \tilde{q} \tag{36}$$

One should notice that the multipliers assume their minimum value when the corresponding constraint is critically satisfied, hence, the path constraint reduces to a point constraint. This property is captured in the equality constraint $t = t_{i,kt} = t_{i,kx}$.

In each of the above problems the critical time, where each of the inequalities $(\breve{g}_i(\hat{x}(t, t^k(x_0), x_0), \hat{u}(t, t^k(x_0), x_0))$, $\tilde{g}_i(\hat{x}(t, t^k(x_0), x_0), \hat{u}(t, t^k(x_0), x_0))$, $\tilde{\mu}(t, t^k (x_0), x_0))$ is critically satisfied, is obtained as an explicit function of x_0 and then is replaced in the inequality to obtain a new inequality $(\breve{G}_i(x_0)$, $\tilde{G}_i(x_0)$, $\breve{\mu}(x_0))$ in terms of x_0. The critical region in which $\hat{u}(t, t^k(x_0), x_0)$ is valid, is given as follows

$$CR = \{\breve{G}(x_0) > 0\ ,\ \tilde{G}(x_0) > 0\ ,\ \tilde{\mu}(x_0) > 0\ ,\ \tilde{\nu}(x_0) > 0\} \cap CR^{IG} \tag{37}$$

It is obvious the critical region CR is defined by a set of compact, non-linear inequalities. The boundaries of CR are represented by parametric non-linear expressions in terms of x_0. Moreover, (34), (35) and (36) imply that in every region calculated in Step 6 of the proposed algorithm, a different number and sequence of switching points and arcs holds.

Although, the optimal control profile $\hat{u}(t, t^k(x_0), x_0)$ constitutes an open-loop control policy, its implementation can be performed in a MPC fashion, thus resulting to a closed loop optimal control policy. More specifically, this is achieved by treating the current state $x(t^*) \equiv x_0$ as an initial state, where t^* is the time when the state value becomes available. The control action $\hat{u}(t, t^k(x(t^*)), x(t^*))$ is then applied for the time interval $[t^*, t^* + \Delta t]$, where Δt denotes the plants sampling time, and in the next time instant $t^* + \Delta t$ the state is updated and the procedure is repeated. Hence, this implementation results to the control law $u(x(t^*)) = \{\hat{u}(t, t^k(x(t^*)), x(t^*)) | t^* \leq t \leq t^* + \Delta t\}$.

4 Example

We are going to illustrate the methods discussed above for the constrained Brachistochrone problem in which a beam slides on a frictionless wire between a point and a vertical plane $1m$ on the right of this point ([3]). The coordinates of the beam on every point on the wire satisfy the following system of differential equations

$$\dot{x} = (2gy)^{1/2} \cos\gamma \tag{38}$$

$$\dot{y} = (2gy)^{1/2} \sin\gamma \tag{39}$$

where x is the horizontal distance, y is the vertical distance (positive downwards), g is the acceleration due to gravity and γ is the angle the wire forms with the horizontal direction. The goal is to find the shape of the wire that will produce a minimum-time path between the two positions, while satisfying the inequality $y - 0.5x - 1 \leq 0$. The above problem is of the form (11) where $\phi(x(t_f), t_f) = t_f$, (11b) is replaced by (38) and (39), $g(x(t), u(t)) = y - 0.5x - 1$

and $\psi^g(x(t_f)) = -x(t_f) + 1$. The last expression represents that at final time the beam should be positioned at a point where $x(t_f) \geq 1$. We also assume that $t_0 = 0$. Although, the problem has already been solved for a fixed initial point $x_0 = [0 \;\; 0]^T$ (as for example in [3]), here the optimal solution is derived for the first time as a function of the initial point coordinates.

The problem is first dealt in discrete-time as described in Section 2. The continuous-time system is turn into a discrete-time system assuming a sampling time Δt such that $[t_0, t_f]$ is divided in three equally spaced time intervals of Δt i.e. $[t_0, t_f] = 3\Delta t$. The discrete-time problem

$$\begin{aligned} &\min_{\gamma_k} \Delta t \\ &x_{k+1} = x_k + (2gy_k)^{-1/2}\, cos\gamma_k\, \Delta t \;,\; k = 0,1,2 \\ &y_{k+1} = y_k + (2gy_k)^{-1/2}\, sin\gamma_k\, \Delta t \;,\; k = 0,1,2 \\ &y_k - 0.5x_k - 1 \leq 0 \;,\; k = 0,1,2,3 \\ &x_3 \geq 1 \end{aligned}$$

is then solved by transforming the above problem in (5) and solving the mp-LP problem to acquire the PWA solution. The continuous-time is solved next following Algorithm 3.1. The results for both the discrete-time case and the continuous-time case together with a simulation for $x_0 = [0 \quad 0]^T$ are shown in Figure 1 and 2. The straight line in both diagrams represents the boundary of the linear constraint $y - 0.5 - 1 \leq 0$. There are three control laws for the continuous-time case, depending in which region the initial state is. The control law in the unconstrained region (Figure 2.) is obtained by solving the following system of algebraic equalities with respect to c_1 and γ

$$\begin{aligned} 0 &= c_1^2 - \arccos(c_1\sqrt{y}) - xc_1^2 + c_1\sqrt{y}\sin\arccos c_1\sqrt{y} \\ \gamma &= -1/2(2g)^{1/2}c_1 t + \arccos c_1\sqrt{y} \end{aligned}$$

In the constrained region the control law is obtained as following. First, the following system of equalities is solved

$$\begin{aligned} &x(\tau'') - \frac{(2g)^{\frac{1}{2}}}{2c_1}\tau'' + \frac{1}{2c_1^2}\sin(2g)^{\frac{1}{2}}c_1(t_f - \tau'') = 1 - \frac{(2g)^{\frac{1}{2}}}{2c_1}t_f \\ &t_f = \frac{\arccos c_1\sqrt{y} + 0.5(2g)^{\frac{1}{2}}c_1\tau''}{0.5(2g)^{\frac{1}{2}}c_1} \;,\; x(\tau'') = 0.1989g\tau''^2 + (2g^{\frac{1}{2}})0.896\sqrt{y_0}\tau'' + x_0 \\ &y(\tau'') = \left(0.222(2g)^{\frac{1}{2}}\tau'' + \sqrt{y_0}\right)^2 \;,\; 0.46 = 0.5(2g)^{\frac{1}{2}}c_1\,(t_f - \tau'') \end{aligned}$$

which is a system of five equations with five unknowns $t_f, \tau'', c_1, x(\tau''), y(\tau'')$. Then, the control to be applied is given as

If $t \leq \tau''$ then $\gamma = 0.46 = \arctan(0.5)$ **Else If** $t \geq \tau''$ then $\gamma = 0.5(2g)^{\frac{1}{2}}c_1(t_f - t)$

As it can be observed from Fig. 1 the approximating, discrete-time, PWA solution is not the optimal one comparing to the optimal solution as it is given

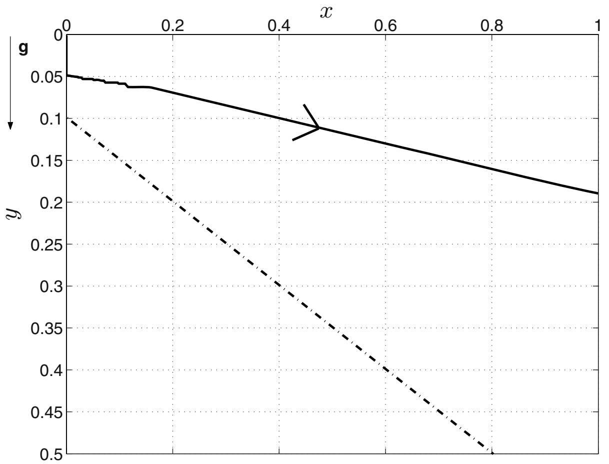


Fig. 1. Discrete-time Brachistochrone Problem

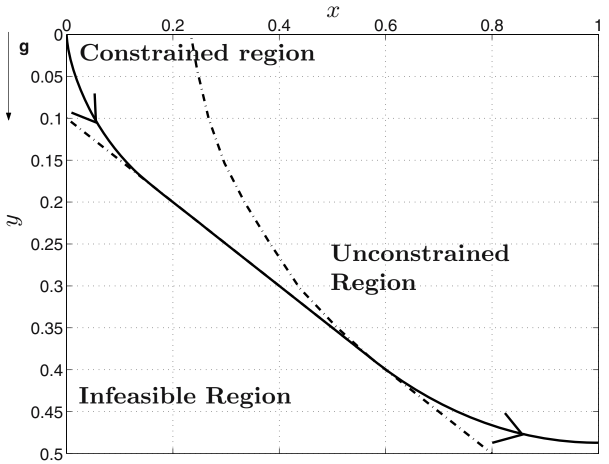


Fig. 2. Continuous-time Brachistochrone Problem

in [3], due to approximation error. On the other hand, the multi-parametric optimal control law illustrated in Fig. 2, is the optimal solution for each initial condition contained in the constrained and unconstrained regions.

5 Conclusions

In this paper the discrete-time MPC problem as well as the continuous-time optimal control problem were examined. A method was presented for obtaining a linear, PWA approximation to the discrete-time MPC problem where the objective and control are obtained as linear, PWA functions of the initial condition. Then, an algorithm was presented that solves the mp-DO problem arising from the non-linear, continuous-time, optimal control problem with state and input constraints, where the objective is a non-linear function of the state at the final time. It was shown that the optimal control profile is a non-linear function of the time and the state variables.

Acknowledgements

The financial support of the Engineering and Physical Sciences Research Council (EPSRC) is greatfully acknowledged.

References

[1] Augustin D, Maurer H (2001) Ann Oper Res 101:75–99
[2] Bemporad A, Morrari M, Dua V, Pistikopoulos EN (2002) Automatica 38(1):3–20
[3] Bryson AE, Ho Y (1975) Applied Optimal Control. Taylor & Francis, New York
[4] Bryson AE, Dehnam WF, Dreyfus SE (1963) AIAA J 1(11):2544–2550
[5] Chen H, Allgower F (1998) Automatica 34:1205–1217
[6] Diehl M, Bock HG, Schloder JP, Findeisen R., Nagy Z, Allgower F (2002) J Proc Con 12:577–585
[7] Dua V, Papalexandri KP, Pistikopoulos EN (2004) J Global Optim 30:59–89
[8] Dua V, Pistikopoulos EN, (1998) Comp Chem Eng 22:S955–S958
[9] Floudas CA, Akrotirianakis IG, Caratzoulas S, Meyer CA, Kallrath J (2005)Comp Chem Eng 29:1185–1202
[10] Hartl RF, Sethi SP, Vickson RG (1995) SIAM Rev 37(2):181–218
[11] Johansen A (2002) Proc IEEE 41st Conf Dec Con, Las Vegas, Nevada USA, 2768–2773
[12] Johansen A (2004) Automatica 40:293–300
[13] Manousiouthakis V, Chmielewski D (2002) Chem Eng Sci 57:105–114
[14] Malanowski K, Maurer H,(2001) Ann Oper Res 101:43–73
[15] Mayne DQ, Rawlings JB, Rao CV, Scokaert PO (2000) Automatica 36:789–814
[16] McCormick GP (1976) Math Prog 10:147–175
[17] Pistikopoulos EN, Dua V, Bozinis NA, Bemporad A, Morrari M (2002) Comp Chem Eng 26(2):175–185
[18] Rawlings JB, Meadows ES, Muske KR (1994) ADCHEM'94 Proceedings, Kyoto, Japan, 185–197
[19] Sakizlis V, Perkins JD, Pistikopoulos EN (2002) Recent developments in optimization and control in Chemical Engineering. In: Luus R (ed) Multiparametric Dynamic Optimization of Linear Quadratic Optimal Control Problems: Theory and Applications. Research Sign-post. Keral, India
[20] Sakizlis V, Perkins JD, Pistikopoulos EN (2005) IEE Proc Con Th Appl 152:443–452
[21] Smith EMB, Pantelides CC (1996) Global optimisation of general process models. In: I.E. Grossmann (ed) Global optimization in engineering design. Dordrecht, Kluwer

Interior-Point Algorithms for Nonlinear Model Predictive Control

Adrian G. Wills[1] and William P. Heath[2]

[1] School of Elec. Eng. & Comp. Sci., Uni. of Newcastle, Australia
Adrian.Wills@newcastle.edu.au

[2] Control Systems Centre, School of Electrical and Electronic Engineering, The University of Manchester, United Kingdom
William.Heath@manchester.ac.uk

Summary. In this contribution we present two interior-point path-following algorithms that solve the convex optimisation problem that arises in recentred barrier function model predictive control (MPC), which includes standard MPC as a limiting case. However the optimisation problem that arises in nonlinear MPC may not be convex. In this case we propose sequential convex programming (SCP) as an alternative to sequential quadratic programming. The algorithms are appropriate for the convex program that arises at each iteration of such an SCP.

1 Introduction

It is often the case in nonlinear model predictive control (NMPC) that the system dynamics are nonlinear in the states and inputs while the constraint sets for both state and input sequences are assumed to be convex [6]. The nonlinear dynamics can, and often do, make the associated control optimisation problem non-convex, and hence more difficult to solve.

One popular strategy for solving non-convex problems is sequential quadratic programming (SQP) where an iterative search procedure is used and the search directions are computed via a quadratic program [11]. Specific adaptations of this SQP approach have been developed for nonlinear model predictive control (NMPC) which take advantage of the sparse structure typical to these problems [4].

An emergent alternative approach for solving non-convex optimisation problems, particularly favoured in topology optimisation, is sequential convex programming (SCP) [16]. SCP is similar in concept to SQP but the search direction is obtained by solving a more general convex programme (CP) in place of the less general quaratic programme (QP). While it has been recognised that this generalisation pays dividends for some mechanical engineering problems [16], the benefits of using SCP for NMPC are as yet undetermined. A thorough evaluation of the relative merits of SQP and SCP for nonlinear MPC is beyond the scope of this contribution.

Rather, in this contribution we consider two interior-point algorithms useful for solving quite general convex programming problems, for example, of the type that

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 207–216, 2007.
springerlink.com

arise at each iteration of an SCP approach. These algorithms are almost standard except that they are geared towards solving convex optimisation problems with a weighted barrier function appearing in the cost. This slight generalisation allows a parsimonious treatment of both barrier function based model predictive control [14] and "standard" model predictive control, which can be identified as a special limiting case. The benefit of including a weighted barrier function is that iterations stay strictly inside the boundary and fewer iterations are needed to converge. This barrier approach is called r-MPC and has been successfully applied to an industrial edible oil refining process as discussed in [15].

Note that due to page limitations all proofs have been omitted and can be found in [13].

2 Nonlinear Model Predictive Control

In what follows we describe NMPC and formulate an optimisation problem which is convex except for the nonlinear equality constraints that represent the system dynamics. This motivates a very brief discussion of SCP which leads to the main theme of this contribution being the two algorithms in Sections 3 and 4. The problem may be formulated as follows. Consider the following discrete-time system with integer k representing the current discrete time event,

$$x(k+1) = f(x(k), u(k)). \tag{1}$$

In the above, $u(k) \in \mathbb{R}^m$ is the system input and $x(k) \in \mathbb{R}^n$ is the system state. The mapping f is assumed to be differentiable and to satisfy $f(0,0) = 0$. Given some positive integer N let $\mathbf{u}$ denote a sequence of control moves given by $\mathbf{u} = \{u(0), u(1), \ldots, u(N-1)\}$ and let $\mathbf{x}$ denote a state sequence given by $\mathbf{x} = \{x(0), x(1), \ldots, x(N)\}$.

For the purposes of this contribution we require that the input *sequence* $\mathbf{u}$ should lie within a compact and convex set $\mathbb{U}$ while the state *sequence* $\mathbf{x}$ should lie in the closed and convex set $\mathbb{X}$. Let $V_N(\mathbf{x}, \mathbf{u})$ denote the objective function associated with prediction horizon N. We assume that V_N is a convex function. The control strategy for NMPC may be described as follows: at each time interval k, given the state $x(k)$, compute the following and apply the first control move to the system.

$$(\mathcal{MPC}): \quad \min_{\mathbf{x},\mathbf{u}} V_N(\mathbf{x}, \mathbf{u}), \text{ s.t. } x_0 = x(k), \quad x_{i+1} = f(x_i, u_i), \quad \mathbf{x} \in \mathbb{X}, \quad \mathbf{u} \in \mathbb{U}.$$

We can associate with the sets $\mathbb{X}$ and $\mathbb{U}$, respectively, *gradient recentred self-concordant barrier functions* B_x and B_u [14]. This allows $(\mathcal{MPC})$ to be expressed as the limiting case when $\mu \to 0$ of the following class of optimisation problems [3].

$$(\mathcal{MPC}_\mu): \quad \min_{\mathbf{x},\mathbf{u}} V_N(\mathbf{x}, \mathbf{u}) + \mu B_x(\mathbf{x}) + \mu B_u(\mathbf{u}) \text{ s.t. } x_0 = x(k), \quad x_{i+1} = f(x_i, u_i).$$

The above class of optimisation problems $(\mathcal{MPC}_\mu)$ have, by construction, a convex cost function and nonlinear equality constraints. If these equality constraints are modelled locally by a linear approximation, then the resulting problem is convex and more readily soluble. This is the impetus for using SCP; at each iteration of the method a local linear approximation to the nonlinear equality constraints is formed and the corresponding CP is solved and the solution provides a search direction, which is then used in a simple line-search method to reduce a merit function. It is beyond the scope of this contribution to provide a detailed SCP algorithm, but standard texts on SQP offer the main themes (see e.g. Chapter 18 from [11]).

We turn our attention to solving ($\mathcal{MPC}_\mu$) where the nonlinear equalities have been linearised, thus resulting in a convex optimisation problem. To this end we present two algorithms in Sections 3 and 4, based on path-following interior-point methods. The first algorithm is a two stage long-step path-following algorithm for the case where the convex constraint set is closed and bounded with non-empty interior. The second algorithm is based on a primal-dual path-following method which is less general but more efficient since it is suitable for the case where the convex constraint set is a self-scaled cone with non-empty interior.

3 Barrier Generated Path-Following Algorithm

Disregarding previous notation, consider the following convex optimisation problem ($\mathcal{P}$) and its closely related class of barrier generated problems ($\mathcal{P}_\mu$).

$$(\mathcal{P}): \quad \min_x f(x) \quad \text{s.t.} \quad x \in G, \qquad\qquad (\mathcal{P}_\mu): \quad \min_x \tfrac{1}{\mu} f(x) + F(x).$$

In the above, $f : \mathbb{R}^n \to \mathbb{R}$ is assumed to be a twice continuously differentiable convex function, G is a closed and bounded convex subset in $\mathbb{R}^n$ with non-empty relative interior denoted G^o, and F is a ν-self-concordant barrier function for G [7].

Restricting the feasible domain G to be closed and bounded means that the barrier function F is strictly convex and attains its unique minimum over the interior of G (see Proposition 2.3.2 in [7]). Hence ($\mathcal{P}_\mu$) has a unique minimiser for all $\mu > 0$, which we denote by $x(\mu)$, and the set $\mathcal{C} \triangleq \{x(\mu) : \mu > 0\}$ of solutions to ($\mathcal{P}_\mu$) is known as the central-path of ($\mathcal{P}$). In fact, the solution $x(\mu)$ coincides with the solution to ($\mathcal{P}$) in the limit as $\mu \to 0$.

However, we are interested in solving ($\mathcal{P}_\mu$) for the case where μ may be chosen as a fixed and relatively large positive number (e.g. for $\mu = 0.1$). This means that standard interior-point algorithms are not directly applicable since they are geared towards solving ($\mathcal{P}_\mu$) for $\mu \to 0$. Nevertheless, as demonstrated in this section, straightforward generalisations of standard algorithms allow for the case of solving ($\mathcal{P}_\mu$) with $\mu \gg 0$.

In terms of the algorithm structure itself, the intuition is as follows. At the k'th iteration, given a point x_k and value $\mu_k > 0$ such that x_k is close to $x(\mu_k)$ the algorithm selects $\mu_{k+1} < \mu_k$ and generates a new point x_{k+1} that is close to $x(\mu_{k+1})$. This process is repeated until the barrier weighting μ_k converges to some prescribed constant value $\mu_c > 0$ with the corresponding x_k's converging to the point $x(\mu_c)$. A damped Newton method is used to generate the new point x_{k+1}, and, we employ a long-step approach, so that aggressive reductions in μ_k are allowed. This has the consequence of increasing theoretical complexity bounds but tends to be favoured for practical algorithms since these bounds are usually conservative.

With this in mind, it remains to find an initial point x_0 and value μ_0 such that x_0 is close to $x(\mu_0)$. This is the subject of Section 3.1 which discusses an initialisation algorithm.

Further to this, however, is a minor technical issue. The algorithm presented in this section requires that the objective function f is linear, but we are interested in a more general choice of f (typically quadratic). Nevertheless, it is straightforward to embed problem $(\mathcal{P})$ into a slightly larger formulation which does have a linear objective. In fact, we do this now using the epigraphic form [2].

$$(\mathcal{A}): \quad \min_{(t,x)} t \quad \text{s.t. } f(x) \leq t, \;\; x \in G, \qquad\qquad (\mathcal{A}_\mu): \;\; \min_{(t,x)} \tfrac{1}{\mu}t - \ln(t - f(x)) + F(x),$$

where $t \in \mathbb{R}$ and $-\ln(t - f(x))$ is assumed to be a ν_f-self-concordant barrier function for $f(x) \leq t$ (e.g. in the case where f is quadratic then this assumption is satisfied with $\nu_f = 2$). Note that $(\mathcal{A}_\mu)$ has a unique minimiser, denoted by $(t(\mu), x(\mu))$. The following lemma shows that if $(t(\mu), x(\mu))$ minimises $(\mathcal{A}_\mu)$ then $x(\mu)$ minimises $(\mathcal{P}_\mu)$ and it therefore appears reasonable to solve problem $(\mathcal{A}_\mu)$ and obtain the minimiser for problem $(\mathcal{P}_\mu)$ directly.

Lemma 1. *Let $(t(\mu), x(\mu))$ denote the unique minimiser of problem $(\mathcal{A}_\mu)$ for some $\mu > 0$, then $x(\mu)$ also minimises problem $(\mathcal{P}_\mu)$ for the same value of μ.*

3.1 Initialisation Stage

The purpose of an initialisation stage is to generate a point (t_0, x_0) and a value μ_0 such that (t_0, x_0) is close to $(t(\mu_0), x(\mu_0))$ (the minimiser of $(\mathcal{A}_\mu)$), which enables the main stage algorithm to progress as described above. The usual approach is to minimise the barrier function $F_{\mathcal{A}}(t, x) \triangleq -\ln(t - f(x)) + F(x)$, which in turn provides a point at the "centre" of the constraint set and close to the central-path of $(\mathcal{A})$ for μ large enough.

However, by construction $F_{\mathcal{A}}(t, x)$ is unbounded below as $t \to \infty$. Thiss problem is overcome, in the literature, by including a further barrier on the maximum value of t resulting in the combined barrier function

$$F_{\mathcal{B}}(t, x) \triangleq -\ln(R - t) + F_{\mathcal{A}}(x), \qquad F_{\mathcal{A}}(t, x) \triangleq -\ln(t - f(x)) + F(x). \tag{2}$$

The unique minimiser of the barrier function $F_{\mathcal{B}}$ and the central path of problem $(\mathcal{A})$ are related in the following important way.

Lemma 2. *Let (t^*, x^*) denote the unique minimiser of $F_{\mathcal{B}}(t, x)$. Then (t^*, x^*) coincides with a point on the central path of problem $(\mathcal{A})$ identified by $\mu = R - t^*$.*

Therefore, minimising the barrier function $F_{\mathcal{B}}$ actually provides a point on the central path of $(\mathcal{A})$, which is precisely the goal of the initialisation stage.

Given $x \in G^o$ and $f(x) < t < R$, let $z_0 = (t, x)$ and choose $\kappa \in (0, 1)$, $\gamma \in (0, 1)$ and $\beta \in (2 - 3^{1/2}, 1)$. Define the function $\phi_\eta(z) \triangleq \eta\langle -\nabla F_{\mathcal{B}}(z_0), z\rangle + F_{\mathcal{B}}(z)$. Let $\eta = 1$ and iterate the following steps.

1. If $\lambda(F_{\mathcal{B}}, z) \leq \beta$ then let $z^* = z$ and stop.
2. If $\lambda(\phi_\eta, z) \leq \gamma$ then update η via $\eta \leftarrow \kappa\eta$.
3. Update z according to damped Newton step $z \leftarrow z - \frac{1}{1+\lambda(\phi_\eta,z)}[\nabla^2\phi_\eta(z)]^{-1}\nabla\phi_\eta(z)$.

Given (t_0, x_0) and μ_0 such that $\lambda(f_{\mu_0}, (t_0, x_0)) \leq \beta$ (where β is chosen in Algorithm 3.1), then choose $\epsilon \in (0, 2 - 3^{1/2})$ and $\kappa \in (0, 1)$ and let $\mu = \mu_0$ and $z = (t_0, x_0)$ and iterate the following steps.

1. If $\mu = \mu_c$ and $\lambda(f_\mu, z) \leq \epsilon$ then stop.
2. If $\mu_c \leq \mu$ then let $\mu \leftarrow \max\{\mu_c, \kappa\mu\}$, otherwise let $\mu \leftarrow \min\{\mu_c, \mu/\kappa\}$.
3. If $\mu = \mu_c$ then let $\beta = \epsilon$.
4. Iterate $z \leftarrow z - \frac{1}{1+\lambda(f_\mu,z)}[\nabla^2 f_\mu(z)]^{-1}\nabla f_\mu(z)$ until $\lambda(f_\mu, z) \leq \beta$.

Furthermore, Nesterov and Nemirovskii's initialisation algorithm (with associated complexity bounds – see Section 3.2.3 in [7]) is directly applicable in this case. Their approach may be described as follows.

Let $\lambda(h, z)$ denote the Newton decrement for a 1-strongly self-concordant nondegenerate function h at the point z defined as (see Section 2.2.1 in [7]),

$$\lambda(h, z) = \left(\nabla^T h(z) \left[\nabla^2 h(z)\right]^{-1} \nabla h(z)\right)^{1/2}.$$

The following proposition shows that (t^*, x^*) combined with a careful choice of initial weighting parameter provide suitable initial conditions for the main stage algorithm.

Proposition 1. *Let* $(t_0, x_0) = (t^*, x^*)$ *and* $\mu_0 = -(e^T Pe)/(e^T Pg)$ *where* $g \triangleq \nabla F_{\mathcal{A}}(z^*)$ *and* $P \triangleq \left[\nabla^2 f_\tau^{\mathcal{A}}(z^*)\right]^{-1}$ *and* $e \triangleq [1,\ 0, \ldots, 0]^T \in \mathbb{R}^{n+1}$. *Then* (t_0, x_0) *and* μ_0 *are suitable initial conditions for the main stage algorithm.*

3.2 Main Stage

Consider the point on the central path of $(\mathcal{A})$ corresponding to some $\mu_0 > 0$, denoted as usual by $(t(\mu_0), x(\mu_0))$. Given some pair (t_0, x_0) close to $(t(\mu_0), x(\mu_0))$, the main stage algorithm follows $(t(\mu), x(\mu))$ towards the point $(t(\mu_c), x(\mu_c))$, where the constant $\mu_c > 0$ is pre-specified by the user.

We are interested in the total number of Newton iterations taken in Step 4 in the above algorithm. In the following proposition we bound this number using results from [7].

Proposition 2. *The total number of Newton iterations required in Algorithm 3.2 is bounded from above by*

$$\mathcal{N}_T = \lfloor \log_\kappa(\mu_c/\mu_0) \rfloor \mathcal{N}(\beta) + \mathcal{N}(\lambda_*) + \lceil \log_2((2 - 3^{1/2})/\epsilon) \rceil.$$

where $\mathcal{N}(\beta) \triangleq \lceil O(1)(1 + |\kappa - 1|\nu_{\mathcal{A}}^{1/2} + \nu_{\mathcal{A}}(\kappa - 1 - \ln\kappa))\rceil$ *and* $\nu_{\mathcal{A}} \triangleq \nu + \nu_f$ *and* $O(1)$ *depends on* β *only.*

4 Self-scaled Cones

The conic form optimisation problem introduced by [7] involves minimisation of a linear objective function over the intersection of an affine subspace and a closed and pointed convex cone. [8] define a class of cones and associated barrier functions which are called *self-scaled*. This class has special properties which allow for efficient interior-point algorithms to be developed [9, 10, 12]. In this section, three types of self-scaled cones are treated, namely the standard non-negative orthant, the second-order cone, and the cone of positive semi-definite symmetric matrices.

The primal-dual path-following algorithm presented below in Section 4.3 is based on [8, 9]. The algorithm finds the point on the primal-dual central path corresponding to the positive real number μ_c. The sections preceding this (namely Sections 4.1 and 4.2) give background material and definitions used in association with the algorithm and notation is as follows. Euclidean space with inner product $\langle\cdot,\cdot\rangle$ and let K denote a self-scaled cone in E. The standard primal and dual conic form optimisation problems considered in the remainder of this section are given by

$$(\mathcal{PC}) : \min_x \langle c, x\rangle, \text{ s.t. } Ax{=}b,\ x \in K, \qquad (\mathcal{DC}) \text{: } \max_{y,s} \langle b, y\rangle, \text{ s.t. } A^*y{+}s{=}c,\ s \in K_*.$$

In the above, $c \in E$, $b \in \mathbb{R}^m$ and $A : E \to \mathbb{R}^m$ is a surjective linear operator and K_* denotes the cone dual to K (which is K itself). Let K^o denote the interior of K.

We are interested in finding the point on the central path of $(\mathcal{PC})$ corresponding to the positive scalar μ_c. This may be achieved by solving the following primal-dual central path minimisation problem for $\mu = \mu_c$.

$$(\mathcal{PD}_\mu) : \quad \min_{x,y,s} \tfrac{1}{\mu}\langle s, x\rangle + F(x) + F_*(s), \quad \text{s.t.} \quad Ax = b,\ A^*y + s = c$$

In the above, F is a ν-self-scaled barrier function for the cone K and F_* is a sign modified Fenchel conjugate of F for the cone K_* – see [8]. Denote the set of minimisers of $(\mathcal{PD}_\mu)$ for $\mu \in (0, \infty)$ by $\mathcal{C}_{\mathcal{PD}}$; this set is typically called the primal-dual central path. Let $S^o(\mathcal{PD})$ denote the set of strictly feasible primal-dual points for $(\mathcal{PD})$ given by

$$S^o(\mathcal{PD}) = \{(x, s, y) \in E \times E \times \mathbb{R}^m : Ax = b,\ A^*y + s = c,\ x \in K^o,\ s \in K^o\}.$$

The algorithm presented in section 4.3 uses a predictor-corrector path-following strategy. Such strategies typically start from a point close to the primal-dual central path and take a predictor step which aims for $\mu = 0$. This direction is followed until the new points violate some proximity measure of the central path, at which time a series of centring steps are taken. These steps aim for

Given a strictly feasible initial point $(x_0, s_0, y_0) \in S^o(\mathcal{PD})$, let $(x, s, y) = (x_0, s_0, y_0)$ and define the Newton iterates as follows.

1. If termination conditions are satisfied then stop.
2. Form (d_x, d_y, d_s) by solving (3) with (x, y, s).
3. Update (x, y, s) according to $x \leftarrow x + \alpha d_x,\ s \leftarrow s + \alpha d_s,\ y \leftarrow y + \alpha d_y$, where α is given at each iteration by $\alpha = (\mu(x,s)\,\sigma_s(\nabla F(x)) + \bar{\sigma})^{-1}$ and $\bar{\sigma} \triangleq \max\{\sigma_x(d_x),\ \sigma_s(d_s)\}$.

the “closest” point on the central path and cease when suitable proximity is restored. There exist many proximity measures for the central path, but [9] use a so-called functional proximity measure which is a global measure in the sense that it has meaning everywhere on $S^o(\mathcal{PD})$, and is defined as

$$\gamma(x,s) = F(x) + F_*(s) + \nu \ln(\mu(x,s)) + \nu, \qquad \mu(x,s) = \tfrac{1}{\nu}\langle s, x\rangle.$$

A region of the central path, denoted $\mathcal{F}(\beta)$, that uses this measure is defined by

$$\mathcal{F}(\beta) = \{(x,y,s) \in S^o(\mathcal{PD}) : \gamma(x,s) \leq \beta\}.$$

4.1 Centring Direction

Given a strictly feasible point $(x, s, y) \in S^o(\mathcal{PD})$ and $\omega \in K^o$ such that $\nabla^2 F(\omega)x = s$, the centring direction (d_x, d_s, d_y) is defined as the solution to the following.

$$\nabla^2 F(\omega) d_x + d_s = -\frac{1}{\mu(s,x)} s - \nabla F(x), \quad A d_x = 0, \quad A^* d_y + d_s = 0. \qquad (3)$$

Let $u \in E$ and $v \in K^o$. Define $\sigma_v(u) = \frac{1}{\alpha}$, where $\alpha > 0$ is the maximum possible value such that $v + \alpha u \in K$. It is convenient to define a centring algorithm [9].

4.2 Affine Scaling Direction

Given a strictly feasible point $(x, s, y) \in S^o(\mathcal{PD})$ and $\omega \in K^o$ such that $\nabla^2 F(\omega)x = s$, the affine scaling direction (p_x, p_s, p_y) is defined as the solution to the following.

$$\nabla^2 F(\omega) p_x + p_s = -s, \quad A p_x = 0, \quad A^* p_y + p_s = 0. \qquad (4)$$

Note that $\langle p_s, p_x\rangle = 0$ from the last two equations of (4). Furthermore, $\langle s, p_x\rangle + \langle p_s, x\rangle = \langle s, x\rangle$ since from the first equation in (4),

$$-\langle s, x\rangle = \langle \nabla^2 F(\omega) p_x + p_s, x\rangle = \langle s, p_x\rangle + \langle p_s, x\rangle.$$

Thus,

$$\langle s + \alpha p_s\ ,\ x + \alpha p_x\rangle = \langle s, x\rangle + \alpha(\langle s, p_x\rangle + \langle p_s, x\rangle) + \alpha^2 \langle p_s, p_x\rangle, = (1-\alpha)\langle s, x\rangle.$$

Therefore, α can be chosen such that $\langle s + \alpha p_s, x + \alpha p_x\rangle = \nu \mu_c$, i.e. $\alpha = 1 - \mu_c/\mu(s,x)$. In general, it is not always possible to take a step in direction (p_x, p_s, p_y) with step size α calculated here, since this may result in an infeasible

Choose ϵ, β and $\varDelta$ such that $\epsilon > 0, \quad 0 < \beta < 1 - \ln(2), \quad \beta < \varDelta < \infty$. Given a problem in the form of $(\mathcal{PD})$, an initial point $(x_0, s_0, y_0) \in \mathcal{F}(\beta)$ and a value $\mu_c > 0$, let $(x, s, y) = (x_0, s_0, y_0)$ and iterate the following steps.

1. While $\mu(s, x) \neq \mu_c$ and $\gamma(s, x) > \beta$ iterate the following steps.
   a) Compute the affine scaling direction (p_x, p_s, p_y) by solving (4).
   b) Update λ with $\lambda \leftarrow 1 - \mu_c/\mu(x, s)$. If $\lambda > 0$ then find $\eta > 0$ such that $\gamma(x + \eta p_x, s + \eta p_s) = \varDelta$ and update α with $\alpha \leftarrow \min\{\eta, \lambda\}$.
   Otherwise find $\eta < 0$ such that $\gamma(x + \eta p_x, s + \eta p_s) = \varDelta$ and $\alpha \leftarrow \max\{\eta, \lambda\}$.
   c) Update the predictor point (x^+, s^+, y^+) using $x^+ \leftarrow x + \alpha p_x, \quad s^+ \leftarrow s + \alpha p_s, \quad y^+ \leftarrow y + \alpha p_y$.
   d) Update the iterate (x, s, y) using the Newton process 4.1 starting from (x^+, s^+, y^+) and stopping as soon as a point in $\mathcal{F}(\beta)$ is found.
2. Update the iterate (x, s, y) using Algorithm 4.1 starting from (x, s, y) and stopping as soon as a point in $\mathcal{F}(\epsilon)$ is found.

point. However, if it is possible to take the full step of size α then the duality gap is equal to $\nu\mu_c$.

4.3 Path-Following Algorithm

The structure of the algorithm is described as follows: starting from a strictly feasible initial point $(x_0, s_0, y_0) \in S^o(\mathcal{PD})$, then firstly a predictor step is computed which aims at reducing the distance between $\mu(x, s)$ and μ_c. The step size is computed in order to maintain iterates within a certain region of the central path If it is possible to take a step that reduces the gap $|\mu(x, s) - \mu_c|$ to zero, whilst remaining inside the allowed region of the central path, then this step will be taken. After computing the intermediate *predictor* point, the algorithm proceeds to *correct* the iterates towards the central path until they are sufficiently close. This process sets the scene for a new predictor step in which further reduction of the gap $|\mu(x, s) - \mu_c|$ may be achieved. From Theorem 7.1 in [9], the number of corrector steps n_c in each iteration is bounded by

$$n_c \leq \varDelta\ (\tau - \ln(1 + \tau)), \qquad \tau = 0.5\,(3\beta/(1 + \beta))^{1/2}. \tag{5}$$

The number of final corrector steps n_{fc} is given by the same relation but with $\beta = \epsilon$.

Also from Theorem 7.1 in [9], if $\mu_0 \geq \mu_c$, then the number of predictor steps $n_{p,1}$ is bounded from above by

$$n_{p,1} \leq \left\lceil \frac{\ln(\mu_0/\mu_c)}{\ln(1/(1 - \delta_1))} \right\rceil, \quad \delta_1 = -\frac{1}{2\nu} c(\varDelta, \beta) + \tfrac{1}{2} \left(\left(\frac{1}{\nu} c(\varDelta, \beta) \right)^2 + 4 \frac{1}{\nu} c(\varDelta, \beta) \right)^{1/2}, \tag{6}$$

where $c(\varDelta, \beta)$ is a positive constant that depends on $\varDelta$ and β only. Furthermore, in [10] it is noted that Theorem 7.1 from [9] holds for negative values of α. Hence, if $\mu_0 \leq \mu_c$, then the number of predictor steps $n_{p,2}$ is bounded from above by

$$n_{p,2} \le \left\lceil \frac{\ln(\mu_c/\mu_0)}{\ln(1-\delta_2)} \right\rceil, \quad \delta_2 = -\frac{1}{2\nu}c(\Delta,\beta) - \tfrac{1}{2}\left(\left(\frac{1}{\nu}c(\Delta,\beta)\right)^2 + 4\frac{1}{\nu}c(\Delta,\beta)\right)^{1/2}. \tag{7}$$

Proposition 3. *The total number of predictor and corrector steps n_{pc} in the above algorithm is bounded from above by*

$$n_{pc} \le \begin{cases} n_{p,1}n_c + n_{fc} & \text{if } \mu_0 \ge \mu_c, \\ n_{p,2}n_c + n_{fc} & \text{if } \mu_0 \le \mu_c. \end{cases} \tag{8}$$

5 Conclusion

The two algorithms presented in this contribution are useful for solving convex programming problems where the solution is generalised to be a specific point on the central-path. This finds immediate application to r-MPC which employs a gradient recentred self-concordant barrier function directly into the cost. More generally however, it would appear that these algorithms are useful within a sequential convex programming approach for solving nonlinear model predictive control problems. This latter point is the subject of further research activity.

References

[1] Albuquerque, J. and V. Gopal and G. Staus and L. T. Biegler and B. E. Ydstie. “Interior point SQP strategies for structured process optimization problems”. *Computers & Chemical Engineering* **2**, S853–S859, (1997).

[2] Boyd, S. and L. Vandenberghe. *Convex optimization.* Cambridge University Press. Cambridge UK (2004).

[3] Fiacco, A. V. and G. P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques.* John Wiley & Sons Inc. New York, (1968).

[4] Gopal, V. and L. T. Biegler. “Large scale inequality constrained optimization and control”. *IEEE Control Systems Magazine* **18**(6), 59–68, (1998).

[5] Kailath, T. *Linear Systems.* Prentice-Hall. Englewood Cliffs, New Jersey, (1980).

[6] Mayne, D. Q., Rawlings, J. B., Rao, C. V. and Scokaert, P. O. M., “Constrained Model Predictive Control: Stability and optimality”, *Automatica.*, **36**, 789–814, (2000).

[7] Nesterov, Y. and A. Nemirovskii. *Interior-point Polynomial Algorithms in Convex Programming.* SIAM. Philadelphia, (1994).

[8] Nesterov, Y. E. and M. J. Todd. “Self-scaled cones and interior–point methods in nonlinear programming”. *Mathematics of Operations Research* **22**, 1–42, (1997).

[9] Nesterov, Y. E. and M. J. Todd. “Primal–dual interior–point methods for self–scaled cones”. *SIAM Journal on Optimization* **8**, 324–364, (1998).

[10] Nesterov, Y. E., M. J. Todd and Y. Ye. “Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems”. *Mathematical Programming* **84**, 227–267, (1999).

[11] Nocedal, J. and S. J. Wright. *Numerical Optimization*. Springer-Verlag. New York, (1999).

[12] Renegar, J. *A Mathematical View of Interior-Point Methods in Convex Optimization*. SIAM. Philadelphia, (2001).

[13] Wills, A. G. Barrier Function Based Model Predictive Control. PhD thesis. School of Electrical Engineering and Computer Science, University of Newcastle, Australia, (2003).

[14] Wills, A. G., Heath, W. P., "Barrier function based model predictive control", *Automatica.*, **40**, 1415–1422, (2004).

[15] Wills, A. G., Heath, W. P., "Application of barrier function based model predictive control to an edible oil refining process", *Journal of Process Control.*, **15**, 183–200, (2005).

[16] Zillober, C., and K. Schittkowski and K. Moritzen. "Very large scale optimization by sequential convex programming". *Optimization Methods and Software* **19**(1), 103–120, (2004).

Hard Constraints for Prioritized Objective Nonlinear MPC

Christopher E. Long and Edward P. Gatzke

University of South Carolina, Department of Chemical Engineering, Columbia, SC, USA 29208
gatzke@sc.edu

Summary. This paper presents a Nonlinear Model Predictive Control (NMPC) algorithm that uses hard variable constraints to allow for control objective prioritization. Traditional prioritized objective approaches can require the solution of a complex mixed-integer program. The formulation presented in this work relies on the feasibility and solution of a relatively small logical sequence of purely continuous nonlinear programs (NLP). The proposed solution method for accomodation of discrete control objectives is equivalent to solution of the overall mixed-integer nonlinear programming problem. The performance of the algorithm is demonstrated on a simulated multivariable network of air pressure tanks.

1 Introduction

Model Predictive Control (MPC) technology is most notable for its ability to control complex multivariable industrial systems. The model-based control scheme relies on the online solution of an optimization problem for the optimal control sequence that minimizes a cost function which evaluates the system over some prediction horizon. The typical cost function accounts for numerous control objectives spanning different levels of relative importance including those stemming from equipment limits and safety concerns, product quality specifications, as well as economic goals. Traditional formulations penalize violations of soft constraints to achieve the desired performance. However, they often rely on *ad hoc* tuning to determine the appropriate trade-off between the various control objectives. Moreover, the tuning becomes less intuitive for systems of increasing complexity.

Recent studies have focused on ensuring that control objective prioritization is handled effectively [3, 4, 7, 8, 13, 14, 18, 19]. Mixed integer methods utilize propositional logic [17] and binary variables to define whether discrete control objectives have been met and whether they are met in order of priority. Terms are included in the cost function to penalize failure to meet the prioritized objectives and the resulting mixed integer program is solved to determine the appropriate input move(s). Nevertheless, the mixed-integer programs are inherently combinatorial in nature and can prove to be computationally demanding, making real-time application difficult. This is of particular concern in nonlinear

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 217–228, 2007.
springerlink.com

formulations with logic constraints that require the solution of the nonconvex Mixed-Integer Nonlinear Program (MINLP).

Here, a nonlinear MPC formulation is presented that utilizes hard constraints to allow for control objective prioiritization. This avoids the need to solve the complex mixed-integer programming problem. By imposing hard variable constraints corresponding to process control objectives such as variable upper and lower bounds, the feasibility of the resulting nonlinear program (NLP) can be evaluated to determine whether a solution exists that will satisfy the control objective. Beginning with the highest priority objective, the hard constraints can be appended to the traditional MPC formulation and the feasibility can be tested in an orderly fashion to determine the optimal control sequence while addressing objective prioritization. This framework is equivalent to the mixed-integer formulation with geometrically weighted control objectives, in which a given control objective is infinitely more important that subsequent objectives. Instead of solving the combinatorial MINLP, the hard constraint approach provides an efficient and logical progression through the binary tree structure. This requires solution of a minimal number of NLP's.

The nonconvex nature of each NLP poses interesting problems in this NMPC formulation with regard to the determination of both global feasibility and global optimality. Use of local solution techniques lacks the ability to provide any indication of problem feasibility and leaves one susceptible to suboptimal solutions. A globally convergence stochastic approach is used to provide indication of global infeasibility and to pursue the global optimum. Deterministic approaches to guarantee global optimality are also considered, as they have been used previously in NMPC formulations[10].

2 Mixed Integer Control Formulation

In traditional Model Predictive Control approaches, control objectives are managed through penalizing violations of soft constraints. The typical objective function to be minimized at each time step is of the form:

$$\Phi = \sum_{i=1}^{p} e(i)^T \Gamma_e e(i) + \sum_{i=1}^{m} \Delta u(i)^T \Gamma_{\Delta u} \Delta u(i) \tag{1}$$

Here m and p are the move and prediction horizon. The error vector, e, represents the difference in the model predicted value and the desired reference for each of the controlled variables, while Δu is a vector describing the level of actuator movement. A vector of weights, or penalties (Γ_e), consists of constant scalar elements that represent the cost of violating each control objective. It is here that control objectives are assigned a relative importance or priority. Likewise, the elements of $\Gamma_{\Delta u}$ are utilized to suppress unnecessary control moves. The downfall of this approach is that it relies on *ad hoc* tuning to determine the appropriate trade-off between meeting various control objectives. For example, it is often difficult to infer how a controller will choose between violating a given constraint by a large amount for a short period of time, or violating a given constraint by a small amount for a more substantial period of time. This becomes particularly troublesome as the complexity of the system increases.

Mixed integer formulations have been used in the MPC framework in an attempt to make controller tuning more intuitive and to insure that higher priority control objectives are accommodated before the less important objectives [8, 9]. This is accomplished by adding additional terms to the traditional objective function and by introducing binary variables into the optimization problem. The premise of this approach is that errors associated with each control objective are explicitly defined and the traditional control objectives can be discretized using propositional logic and a "big M" constraint of the form:

$$e(i) \leq M(1 - O_j) \quad \forall j = 1..p \tag{2}$$

Here the O_j is a binary flag defining whether a given control objective can be met absolutely and M is a large value. Thus, if the error associated with a given control objective is zero, implying the control objective can be met at each point across the prediction horizon, the binary flag can take on a value of 1. Otherwise, O_j is forced to a value of zero in order to relax the constraint. Additional constraints can then be used to indicate whether a control objective is met in order of its priority. A binary flag, P_j is used to indicate if this is indeed the case. Constraints of the form $P_j \leq O_j$ are required to insure that the objectives are met before being flagged as met in order according to their relative priority. A set of constraints of the form $P_{j+1} \leq P_j$ are used to force higher priority objectives to be met first. The objective function in such a formulation is of the form:

$$\Phi = \Gamma_O O + \Gamma_P P + \sum_{i=1}^{p} e(i)^T \Gamma_e e(i) + \sum_{i=1}^{m} \Delta u(i)^T \Gamma_{\Delta u} \Delta u(i) \tag{3}$$

Here O is a vector of the binary flags defining if the discrete objectives have been met and P is a vector of binary flags defining whether or not the control objectives have been met in order of priority. Γ_O and Γ_P are vectors of weights that reward meeting the discretized objectives in order or priority. Typically the values of Γ_P are chosen to be much larger than the elements of Γ_O, which are orders of magnitude larger than the traditional MPC penalties. This approach has been demonstrated in a number of cases, including in the inferential control of unmeasured states in which a state space model is employed to explicitly define the unmeasured states to be constrained [8].

One drawback of the mixed integer MPC approaches is the combinatorial nature of the resulting mixed integer optimization problem. The resulting mixed-integer linear programming (MILP) problem can require the solution of up to 2^N LPs where N is the number of binary variables incorporated into the problem. Fortunately, the computational demand is often mitigated by the relatively few number of true decision variables associated with the MPC problem. The controller need only specify the m moves for each of the n_u process inputs, and the remaining variables (modeled process states, errors, etc) are then subsequently defined based on their relationship to the chosen input sequences, the model, and process data. From a practical standpoint, the approach can also be limited by problem dimensionality as penalties spanning numerous orders of magnitudes raise issues with solver tolerances.

To date, these mixed integer approaches for prioritized objective control have focused on linear formulations. With the availability of efficient MILP solvers [6], the methods prove to be viable for real-time control. This is particularly true for control of chemical processes, which typically have time constants on the order of minutes. However, many industrial processes are sufficiently nonlinear to motivate the consideration of nonlinear formulations. The use of a nonlinear model provides improved closed-loop performance but at the expense of increased computational demand as the nonlinear formulation inherently relies on the solution of a more difficult nonconvex nonlinear problem (NLP). As the prioritization of control objectives requires the solution of an MILP in the linear case instead of an LP, the nonlinear mixed integer formulation requires the solution of a difficult mixed integer nonlinear program (MINLP) instead of a single NLP.

3 Hard Constraints Formulation for Objective Prioritization

Assume that for a particular control problem n control objectives are to be handled appropriately based on their perceived relative priority. In the mixed integer formulation, this would require in the worst case the solution of $2^{2n+1}-1$ LP relaxation nodes in a traditional branch-and-bound search. This stems from the $2n$ binary variables. Efficient MILP solvers exist [6] and have been shown to be viable for real-time implementation in linear MPC formulations. However, for nonlinear dynamic formulations that consider solution of nonconvex MINLP's, the optimization problem can prove to be too computationally demanding. The motivation of the hard constraint formulation is a reduction in the computational demand to make prioritized objective NMPC possible for real-time control.

This algorithm uses hard variable constraints as a means to avoid the need to solve the complex MINLP for control objective prioritization in nonlinear MPC formulations. Consider the nonlinear control problem with n prioritized control objectives. Provided that each control objective is infinitely more important than subsequent objectives, these constraints can be handled using a logical progression through a reduced binary tree structure. Initially, a purely continuous constrained NLP is formulated. A traditional objective function (as in Equation 1) is used with explicitly defined errors for violations associated with each control objective constraint at each point in the prediction horizon. Soft constraint penalty weights are defined as in typical MPC methods. These dictate the controller performance in cases in which a control objective cannot be met for all points in p. Note that as with all soft constraint formulations, this NLP is inherently feasible.

Starting with the highest priority objective, the ability to meet each individual control objective is considered. First, hard constraints forcing the errors associated with the highest priority objective to zero are incorporated into the optimization problem. The constraints are of the form: $e_i(k) = 0 \; \forall k = 1..p$ where i corresponds to the objective. The feasibility of the modified problem is then examined, but the actual ***global solution is not required***. If feasible, it is the known that the control objective can be met and the constraints are left in the problem. However, if infeasible, the corresponding hard constraints are removed (relaxed). Note that at this point, violations of this unachievable control objective will be ultimately minimized based on their appearance in the traditional objective function as soft penalties. The next highest priority objective is then considered. Appropriate hard constraints are again added to the problem and the feasibility is again tested, with two possible results: The second problem considered will involve hard constraints corresponding to the two highest priority control objectives if the initial problem was feasible. However, if the initial problem was not feasible, the second problem will only consider the hard constraints associated with the second control objective. All subsequent control objectives are considered one at a time according to this procedure. This will define a single NLP that represents the final node from the binary tree that would have yielded the optimal solution. Again, the traditional weights associated with each error variable are still necessary. These values will define how the controller will handle cases in which the hard constraints associated with particular prioritized control objectives cannot be met absolutely over the whole prediction horizon. When it has been determined that a hard constraint cannot be met without sacrificing higher priority objectives the control algorithm will fall back to the traditional weights as it minimizes soft constraint violation. This NLP is then solved for the optimal input sequence which can then be implemented. Pseudo-code of this algorithm is presented in Algorithm 1.

Ultimately, this approach requires only that the feasibility of a maximum of n problems be assessed and then only the solution of a single NLP. Effectively, each individual feasibility check represents the binary flag from the mixed integer formulation that defines whether or not the discretized control objective can be met. Checking the numerous control objectives individually in the order of priority replaces the need for the binary variables and additional propositional logic constraints associated with meeting the objectives in order. Note that this framework is exactly equivalent to the mixed-integer implementation with geometrically weighted control objectives in which a given control objective is infinitely more important than subsequent objectives.

As presented, this NMPC formulation judiciously handles n prioritized control objectives through the consideration of a maximum of n NLPs. The feasibility of up to n problems must be determined and the appropriate NLP is then solved to global optimality. This maximum number of problems is encountered in the instance where the feasibility of the problem associated with each control objective is considered individually beginning with the highest priority objective. However, a number of heuristics can be utilized to further reduce the computational demands. For example, under relatively normal operating conditions, it is to be expected that a large number of the control objectives associated with

```
Formulate traditional soft constraint MPC problem with explicit errors for each control objective. ($Eq.1$)
Rank control objectives in order from highest to lowest priority.
FOR $i$ from 1 to # of Prioritized Objectives
    Impose hard constraints with highest priority control objective $i$ not yet considered:
        ($e_i(k) = 0\ \forall k = 1..p$)
    Check problem feasibility (Solve deterministically until infeasible or feasible)
    IF Problem is feasible
        Retain corresponding hard constraints.
    ELSE
        Remove corresponding hard constraints.
    END
END For all objectives
Solve resulting NLP with hard constraints corresponding to achievable objectives to global optimality.
Implement optimal control sequence.
```

Algorithm 1. Pseudo-Code for the Hard Constraint Formulation for Prioritized Objective Control

a given process can be met, particularly those of high priority. Under this assumption, it could be beneficial in the optimal case where all objective can be met. If feasible, the remaining problems (feasibility checks) can be ignored. If all control objectives are indeed feasible, the algorithm needs only to consider and solve a single NLP.

4 Nonconvex Optimization

An integral piece of any MPC algorithm is the optimization problem solution method. For nonlinear formulations this is of particular importance. Local solution methods can be used. This can leave the strategy susceptible to suboptimal solutions. Gradient-based methods can become trapped in local minima, thus the optimality of the solution may be dependent on the initial value. In an attempt to address the nonconvex problem, globally convergent stochastic methods can be employed. In this work, the feasible solution space is randomly searched and probabilistic arguments support the convergence of the algorithm to solution.

NLP solution serves two purposes in this NMPC formulation. The first function of this tool is to check the feasibility of a given NLP to determine whether a control objective can be met in order of priority. In this instance, an optimal solution is not needed. Here, the search need only provide any feasible solution (upper bound) or sufficiently search the solution space and exhaust the possibility that a feasible solution exists. For the stochastic search, as the number of points considered approaches infinity, the global feasibility can be guaranteed. This semblance of global optimality provided by the random search is important, as local methods can only be relied upon to indicate that a problem is feasible. Indication of local infeasibility fails to provide information of global feasibility.

The last step in the algorithm is to determine the global optimum of the final NLP that best accounts for the control objective prioritization. In this case, the stochastic approach is used to solve this single problem to “global optimality”. For this particular NLP, the solution space is again randomly searched. The best solution found in the random search is used as the starting point for a local gradient based solution. This effectively determines the global solution, provided that the solution space is adequately sampled. However, no guarantee can be made for samples of finite size. In this context, the solution provides an optimal control sequence that assures that all control objectives are logically handled in order of their priority and those that cannot be met absolutely have their violations minimized based on their appearance in the traditional objective function.

Alternatively, established deterministic methods that provide a rigorous guarantee on global optimality can be considered. Deterministic methods for global optimization typically rely on the generation of convex relaxations of the original nonconvex problem. These convex relaxations are constructed in a number of ways. One such approach, the αBB method [2], handles general twice-differentiable nonconvex functions. This method relies on the determination of the minimum eigenvalue for the Hessian of the nonconvex function over the region of interest, however does not require additional variables or constraints in the formulation. An alternative approach [11, 16] generates convex functions using the known convex envelopes of simple nonlinear functions. The original nonconvex problem is reformulated to a standard form with constraints involving simple nonlinear functions by the introduction of new variables and new constraints. This analysis is achieved by recursively simplifying terms in the function tree expression by introduction of new simple nonlinear expressions. The new simple nonlinear functions explicitly define new variables in terms of other variables. A resulting nonconvex equality constraint can then be replaced by convex inequality constraints. A detailed explanation of these methods and some comparison of the methods is given in [5].

Upon creation of the linear relaxation for the nonconvex nonlinear problem, the branch-and-reduce method [15] can be implemented. This is an extension of the traditional branch-and-bound method with bound tightening techniques for accelerating the convergence of the algorithm. Within this branch-and-reduce algorithm, infeasible or suboptimal parts of the feasible region can be eliminated using range reduction techniques such as optimality-based and feasibility-based range reduction tests [1, 15, 16] or interval analysis techniques [12]. These techniques help to derive tighter variable bounds for a given partition in the search tree. The algorithm terminates when the lower bounds for all partitions either exceed or are sufficiently close (within specified tolerances) to the best upper bound. At this point, a global optimum has been found. This approach was applied in a NMPC framework [10].

5 Case Study

The proposed prioritized objective nonlinear model predictive control algorithm is demonstrated on a simulated multivariable network of air pressure tanks. A complete description of the system and closed-loop results are presented below. Consider a simulated multivariable network of air pressure tanks.

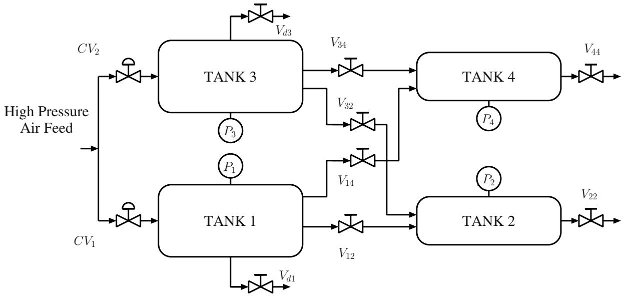


Fig. 1. Schematic of the Network of Pressure Tanks

Supply air is fed to the system at 60 psig through two control valves which act as the manipulated variables (u_1 and u_2) in the control problem. Pressure gradients drive the flow of air through the system. The air flows through the four tanks that are interconnected, while the numerous valves throughout the system dictate the direction of the flow. After traveling through the system, the air ultimately exits the downstream tanks to the atmosphere. It is assumed that the pressure in each of the four tanks can be measured. These will act as the process outputs to be controlled. The non-square nature of this configuration (2×4) lends itself well to demonstrating the ability of this specific controller as all four measurements cannot be maintained at setpoint using only two manipulated variables, thus forcing the controller to decide the appropriate trade-off based on the prioritized objectives.

For this work, it is assumed that the flow of air across a given valve (v_i) can be defined as:

$$f_i(k) = c_i \sqrt{\Delta P_i(k)} \tag{4}$$

where f_i is a molar flow, k is the sample time, c_i is a proportionality constant related to the valve coefficient, and ΔP_i is the pressure drop across the valve. For this study, it is assumed that there is no reverse flow across a valve, forcing ΔP_i non-negative. Under ideal conditions, the discrete time governing equations defining the pressures in each tank are taken as:

$$\begin{aligned}
P_1(k+1) &= \tfrac{hP_1(k)}{V_1}\left(u_1 f_{CV1} - \gamma_1 f_{12} - (1-\gamma_1) f_{14}\right) + P_1(k) \\
P_2(k+1) &= \tfrac{hP_2(k)}{V_2}\left(\gamma_1 f_{12} + (1-\gamma_2) f_{32} - f_{22}\right) + P_2(k) \\
P_3(k+1) &= \tfrac{hP_3(k)}{V_3}\left(u_2 f_{CV2} - \gamma_2 f_{34} - (1-\gamma_2) f_{32}\right) + P_3(k) \\
P_4(k+1) &= \tfrac{hP_4(k)}{V_4}\left(\gamma_2 f_{34} + (1-\gamma_1) f_{14} - f_{44}\right) + P_4(k)
\end{aligned} \tag{5}$$

where γ_1 and γ_2 define the fractional split of air leaving the upstream tanks. Here, this set of equations represents both the nonlinear process and the model used for control purposes. The sampling period (h) used was 3 minutes. This is important as it defines the time limit in which each optimization problem must be solved for real-time operation. The parameter values used in this study are summarized in Table 1.

Table 1. Model Parameters for the Simulated Network of Pressure Tanks

$V_1 = 8$	$V_3 = 8$	$c_{CV1} = 0.25$	$c_{12} = 0.02$	$c_{22} = 0.06$	$c_{44} = 0.06$	$\gamma_1 = 0.5$
$V_2 = 5$	$V_4 = 5$	$c_{CV2} = 0.25$	$c_{14} = 0.05$	$c_{34} = 0.02$	$c_{32} = 0.05$	$\gamma_2 = 0.3$

6 Closed-Loop Results

The performance of the proposed control algorithm is tested on the simulated pressure tank network. Its ability to appropriately handle control objective prioritization through a number of reference transitions and in the presence of disturbance loads is demonstrated. A number of control objectives are defined and assigned a relative priority. These are summarized in Table 2. Assume that for safety concerns, it is important that the pressure in the upstream tanks are kept below a pressure of 60 psig. These constraints are given highest priority. A secondary goal is the regulation of the pressure in second tank (P_2). It is desirable for this tank pressure to closely track setpoint, and thus a setpoint constraint as well as tight upper and lower bounds (±2 psig from setpoint) are imposed. Note that the lower and upper bounds are assigned to be priority 3 and 4 respectively, while the setpoint constraint is not considered for objective prioritization and use of hard constraints. Subsequent control objectives include a lower bound on the pressure in tank 1 and bounds on the pressure in tank 4. The pressure in tank 3 is left unconstrained. All constraints include a 15 minute delay for enforcement.

Table 2. Summary of Prioritized Control Objectives (* Note that a hard constraint corresponding to the setpoint control objective is not used.)

Relative Priority	Variable Constrained	Constraint Type	Constraint Value	Relative Priority	Variable Constrained	Constraint Type	Constraint Value
1	P_1	UB	60	6	P_4	UB	30
2	P_3	UB	60	7	P_4	LB	20
3	P_2	LB	25/20/25	8	P_4	LB	23
4	P_2	UB	29/24/29	9*	P_2	SP	27/22/27
5	P_1	LB	55				

The controller is tuned to with $m = 2$ and $p = 20$. Each control objective is assigned a weight of $\Gamma_e = 100$ and the input movements are not penalized, $\Gamma_u = 0$. These weights are used to determine the tradeoffs between soft constraint violations of the various control objectives for which the hard constrained problem cannot be solved. For this example, at each time step, the appropriate NLP is solved using a stochastic approach followed by a local gradient based search. Specifically, 1000 points in the solution space are considered, the best of which is used as the starting point for the gradient-based solution. This stochastic and gradient-based solution process is repeated 3 times and the best solution is taken as the optimal control sequence to be implemented.

At $t = 150$ minutes, a setpoint change for the pressure in tank 2 steps from its initial value of 27 psig to 22 psig. The controller recognizes the change and

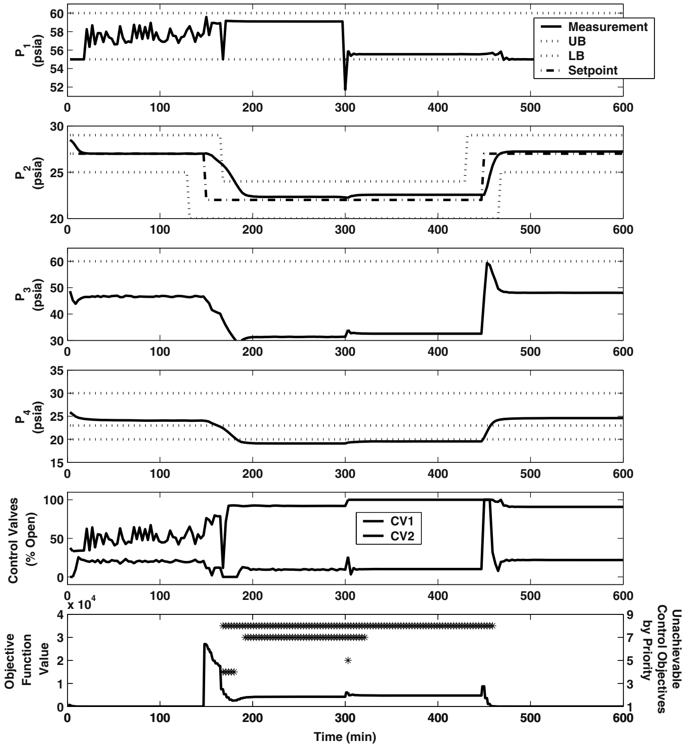


Fig. 2. Closed-loop Results of the Simulated Network of Pressure Tanks being Controlled by the Hard Constraint formulation for Prioritized Objective NMPC, Objective Function Values, and Unachievable Control Objectives

begins moving the system to accommodate it (Figure 2). Note that the bounds associated with P_2 are of high priority (priority 3 and 4). The move to satisfy these constraints requires the controller to violate the lower bounds on the pressure in tank 4 (priority 7 and 8). This means that imposing hard constraints corresponding to these control objectives renders the associated NLP infeasible. The control objectives that lead to NLP infeasibility and the objective function of the solution at each time step is seen in Figure 2. During this transition, the controller is also unable to move the system fast enough to avoid briefly violating the upper bound on the pressure in tank 2 (priority 4). The controller does however, track the new setpoint.

At $t = 300$ minutes, a disturbance is imposed on the system by simulating a leak in the first tank. The controller cannot respond to the disturbance fast enough to avoid the lower bound on the pressure in this tank (priority 5). However the controller is able to quickly return the pressure within the limiting values. This action saturates one of the inputs. Again, P_2 is maintained within its limits but is unable to track the setpoint in an offset free position. At $t = 450$ minutes, while still operating under a disturbance, P_2 is returned to its original setpoint value of 27 psig. The controller recovers the ability to meet the all control objectives associated with P_4. However, the controller does ride the lower bound constraint associated with P_1.

Note that the indication of feasibility in this formulation is based on the ability for the controller to find control moves that can maintain the model predicted outputs within their constraint limits by driving the error to zero. However, the model predicted values are not always necessarily equivalent to the true process values (measurements). This plant-model mismatch is inherent in the accuracy of models developed through the identification process and is often exaggerated in the presence of an unmeasured disturbance. For this reason, more conservative bounds should be used to insure that the desired limits are enforced.

7 Conclusions

A Nonlinear Model Predictive Control (NMPC) algorithm that utilizes hard variable constraints for control objective prioritization has been proposed. The formulation requires the solution of only a minimal number of NLP's as opposed to a complex MINLP. A stochastic approach is utilized to check problem feasibility and to find the optimum of the resulting nonconvex NLP's. This alleviates the shortcomings of purely local gradient based methods as it better searches the solution space for the global optimum. However, optimality can only be rigorously guaranteed using existing deterministic methods. The controller was shown to be effective in appropriately handling control objectives of varying importance in a simulated multivariable network of pressure tanks.

Acknowledgments

The authors would like to acknowledge financial support from the American Chemical Society Petroleum Research Foundation grant #38539-G9 and the National Science Foundation Early Career Development grant CTS-038663.

References

[1] C. S. Adjiman, I. P. Androulakis, and C. A. Floudas. Global Optimization of Mixed-Integer Nonlinear Problems. *AIChE J.*, 46(9):1769–1797, 2000.

[2] C. S. Adjiman, S. Dalliwig, C. A. Floudas, and A. Neumaier. A Global Optimization Method, αBB, for General Twice-Differentiable Constrained NLPs - I Theoretical Advances. *Comput. Chem. Eng.*, 22(9):1137–1158, 1998.

[3] A. Bemporad and M. Morari. Control of Systems Integrating Logic, Dynamics, and Constraints. *Automatica*, 35:407–427, 1999.

[4] E. P. Gatzke and F. J. Doyle III. Multi-Objective Control of a Granulation System. *Journal of Powder Technology*, 121(2):149–158, 2001.

[5] E. P. Gatzke, J. E. Tolsma, and P. I. Barton. Construction of Convex Function Relaxations Using Automated Code Generation Techniques. *Optimization and Engineering*, 3(3):305–326, 2002.

[6] ILOG. *ILOG CPLEX 9.1: User's Manual.* Mountain View, CA, April 2005.

[7] E. C. Kerrigan, A. Bemporad, D. Mignone, M. Morari, and J. M. Maciejowski. Multi-objective Prioritisation and Reconfiguration for the Control of Constrained Hybrid Systems. In *Proceedings of the American Controls Conference*, Chicago, Illinois, 2000.

[8] C. E. Long and E. P. Gatzke. Globally Optimal Nonlinear Model Predictive Control. In *DYCOPS 7, the 7th International Symposium on Dynamics and Control of Process Systems*, Boston, MA, 2004.

[9] C. E. Long and E. P. Gatzke. Model Predictive Control Algorithm for Prioritized Objective Inferential Control of Unmeasured States Using Propositional Logic. *Ind. Eng. Chem. Res.*, 44(10):3575–3584, 2005.

[10] C. E. Long, P. K. Polisetty, and E. P. Gatzke. Nonlinear Model Predictive Control Using Deterministic Global Optimization. *J. Proc. Cont.*, accepted, 2005.

[11] G. P. McCormick. Computability of Global Solutions to Factorable Nonconvex Programs: Part I - Convex Underestimating Problems. *Mathematical Programming*, 10:147–175, 1976.

[12] R. E. Moore. *Methods and Applications of Interval Analysis.* SIAM, Philadelphia, 1979.

[13] R. S. Parker, E. P. Gatzke, and F. J. Doyle III. Advanced Model Predictive Control (MPC) for Type I Diabetic Patient Blood Glucose Control. In *Proc. American Control Conf.*, Chicago, IL, 2000.

[14] A. N. Reyes, C. S. Dutra, and C. B. Alba. Comparison of Different Predictive Controllers with Multi-objective Optimization: Application to An Olive Oil Mill. In *Proceedings of the 2002 IEEE International Conference on Control Applications and International Symposium on Computer Aided Control Systems Designs*, pages 1242–1247, Glasgow, UK, 2002.

[15] H. S. Ryoo and N. V. Sahinidis. Global Optimization of Nonconvex NLPS and MINLPs with Application to Process Design. *Comput. Chem. Eng.*, 19(5):551–566, 1995.

[16] E. M. B. Smith. *On the Optimal Design of Continuous Processes.* PhD thesis, Imperial College, London, 1996.

[17] M. L. Tyler and M. Morari. Propositional Logic in Control and Monitoring Problems. *Automatica*, 35:565–582, 1999.

[18] J. Vada, O. Slupphaug, and T. A. Johansen. Optimal Prioritized Infeasibility Handling in Model Predictive Control: Parameteric Preemptive Multiobjective Linear Programming Approach. *Journal of Optimization Theory and Applications*, 109(2):385–413, 2001.

[19] J. Vada, O. Slupphaug, T. A. Johansen., and B. A. Foss. Linear MPC with Optimal Prioritized Infeasibility Handling: Application, Computational Issues, and Stability. *Automatica*, 37:1835–1843, 2001.

A Nonlinear Model Predictive Control Framework as Free Software: Outlook and Progress Report

Andrey Romanenko and Lino O. Santos

GEPSI-PSE Group, Dep. de Engenharia Química, Universidade de Coimbra,
Pinhal de Marrocos, Pólo II, 3030-290 Coimbra, Portugal
andrew,lino@eq.uc.pt

Summary. Model predictive control (MPC) has been a field with considerable research efforts and significant improvements in the algorithms. This has led to a fairly large number of successful industrial applications. However, many small and medium enterprises have not embraced MPC, even though their processes may potentially benefit from this control technology. We tackle one aspect of this issue with the development of a nonlinear model predictive control package NEWCON that will be released as free software. The work details the conceptual design, the control problem formulation and the implementation aspects of the code. A possible application is illustrated with an example of the level and reactor temperature control of a simulated CSTR. Finally, the article outlines future development directions of the NEWCON package.

1 Introduction

Model predictive control has been a field with considerable research efforts and significant improvements in the algorithms [3]. Also, the number of commercial MPC offerings on the market has increased in the last years. Clearly, there are a number of industrial areas where MPC use is prominent because of the great economical benefit of this advanced control technology: refineries, petrochemical, and chemical [20].

It should be noted, however, that the scope of companies that use MPC solutions is rather limited. In order to implement a successful solution, not only is it necessary to be able to make a very significant investment in expensive proprietary products on the market, but also it is important to have in-house technical and engineering staff able to apply and maintain them and the management to realize the benefits [7]. Because of these two factors, small and medium enterprises (SME), that play a very important role in some economies, may not know about the existence of MPC or may not realize the potential or, finally, may not afford it.

Lately, the free and open-source software (OSS) development paradigm has been gaining wide acceptance. It is mainly characterized by the rights to use, make modifications, and redistribute software subject to certain limitations. Some free software packages enjoy a big user community resulting in a fast development pace.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 229–238, 2007.
springerlink.com

The advantages of such development model for academia and research are evident, especially for computing intensive fields [25]. Octave [6], R [12], and ASCEND [19] are examples of successful free software projects.

Besides, new "open source" business models for commercial companies have been outlined [10, 14]: distribution and technical support, in which a set of OSS packages are customized and tuned to a specific platform; education, with OSS used as pedagogical material; hardware sale, in which OSS adds value to the hardware being marketed; custom development, in which OSS is tailored for the needs of a particular user; proprietary-open source mixture, in which a proprietary enhancement to an open source program is available for a fee.

This type of software has a great potential in the field of process control, especially if some of its disadvantages are overcome [21].

As an alternative to proprietary software for nonlinear model predictive control (NMPC) and as a way to enable SMEs to use NMPC, we have been developing newcon, a solution based on open-source and free software. The details of the underlying technology may be found elsewhere [23]. For the core elements of the NMPC framework, the ODE solver with sensitivity analysis capabilities and an optimizer, we use highly efficient third party libraries developed as open-source and free software [9, 11]. Once ready for an initial release, this NMPC framework will be available for educational institutions for teaching and research entities for testing and further improvement. There is a technological spin-off in the process of creation whose role is to promote the development of the package and to deploy MPC application in the free software paradigm. Besides, the code will be available for download for anybody, subject to a free software license.

Other related software packages in the field of NMPC include the Octave NMPC package [26], Omuses [8], a robust NMPC package [15], the optimal control package MUSCOD-II [4], and the automatic code generation system for nonlinear receding horizon control AutoGenU [16].

A description of the implemented control formulation along with a design overview of newcon is given in Section 2. Also, information on the ODE and optimizer solvers is provided in Section 2.2. The application of newcon is illustrated with a nonlinear example in Section 3. Finally, some remarks and future directions are pointed out in Section 4.

2 Nonlinear MPC Framework

The nonlinear MPC framework newcon proposed here is based on the Fortran package developed by [23]. It implements a control formulation with a multiple shooting strategy to perform the NMPC predictions as described in [22], and is based on the Newton-type control formulation approach established by [17, 18].

One of the main concerns in the design and the implementation phases of newcon is to make it modular so that existing components and software, such as a regulatory control and automation layers, could be integrated in the resulting advanced control system. The whole point of deploying newcon in small and

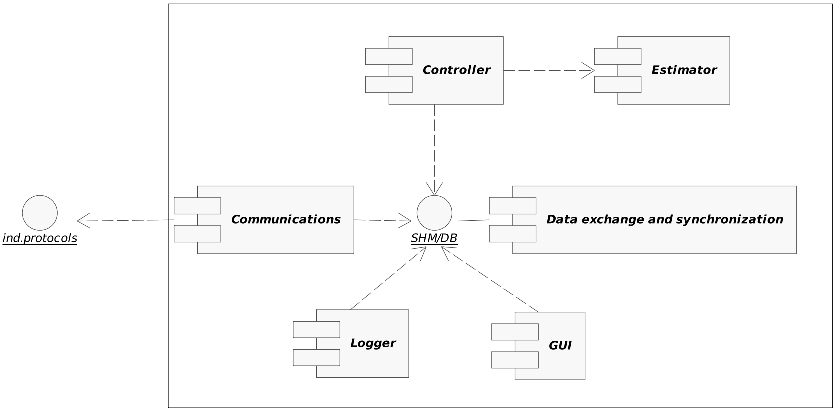


Fig. 1. NEWCON Component diagram

medium companies is not to replace existing systems and components, but rather to build upon them.

The conceptual design of NEWCON (Figure 1) incorporates several modules with distinct features.

The *Controller* block contains code necessary for the control problem formulation (as described in Section 2.1) and interface routines to an ODE and a QP solvers. Besides, in order to reduce the communication overhead, the system model also makes part of this module. Moreover, an *Estimator* component may be used to update the model states and parameters, hence reducing model-plant mismatch.

The purpose of the *Data exchange and synchronization* component is to provide the means for consistent dataflows and to ensure correct timing in the overall system. This is achieved by the use of POSIX shared memory and synchronization mechanisms. Alternatively, when the timing requirements are not strict, this module may be based on a database engine.

The function of the *Communications* block is to interface NEWCON to the existing regulatory control system of the plant using open industrial communication protocols. Because of the widespread use and low cost of the hardware, open TCP/IP based protocols running over Ethernet hardware will be favored. This module may implement capabilities/protocols necessary in a particular application, adding to the flexibility of the overall system and reducing its size.

The measurements of the plant, the controller states, setpoints, outputs, performance indices, as well as other relevant information is recorded by the *logger* module. This module currently supports data saving in plain format in several text datafiles. Its functionality will be expanded to include database support.

The graphical user interface (GUI) module provides a user friendly way to control the system, to monitor graphically important variable trends and performance indicators. In order to distribute computing resources evenly, and to prevent information “flooding”, two GUI modules are considered, one for the plant (be it real or simulated) and the other for the controller itself.

NEWCON is being developed as a package in the Linux operating system using its powerful development tools, such as the Gnu Compiler Collection (GCC) and Autotools.

2.1 Control Problem Formulation

The NEWCON framework requires a mechanistic model of the process to control of the form:

$$\dot{x} = f(x, u, d; \theta) \tag{1}$$

$$y = g(x; \theta) \tag{2}$$

with f and g twice continuously differentiable, where $x \in \mathbb{R}^{n_\mathrm{s}}$ is the state vector, $u \in \mathbb{R}^{n_\mathrm{m}}$ is the control vector, $d \in \mathbb{R}^{n_\mathrm{d}}$ is the disturbance vector, $\theta \in \mathbb{R}^{n_\theta}$ is the parameter vector and $y \in \mathbb{R}^{n_\mathrm{o}}$ is the vector of output variables. A multiple shooting formulation with different output and input predictive horizon lengths (denoted by p and m respectively, with $p \geqslant m$) is used to solve the model (1-2) over the predictive horizon p, where the state equations are integrated inside each sampling interval [22]. This method is also referred to as direct multiple shooting [2].

The predictive control formulation features state, output and control constraints. Moreover, it can handle output terminal constraints, control move rate constraints, and state, output, input and control move rate constraint relaxation. This leads to the following control problem formulation to solve at every time index i [23]:

$$\min_{X,U,\epsilon} \Upsilon_i\big(Y, U, \epsilon\big) = \Psi_i\big(Y, U\big) + P_i(\epsilon) \tag{3}$$

$$\text{s.t. } u_{i+k} = u_{i+m-1}, \quad k = m, \cdots, p-1 \tag{4}$$

$$\bar{x}_{i+k} - \phi(\bar{x}_{i+k-1}, \bar{u}_{i+k-1}) = 0, \quad k = 1, \ldots, p \tag{5}$$

$$y_{\mathrm{sp},i+p} - y_{i+p} = 0 \tag{6}$$

$$X_\mathrm{L} - \epsilon_\mathrm{x} \leqslant X \leqslant X_\mathrm{U} + \epsilon_\mathrm{x} \tag{7}$$

$$Y_\mathrm{L} - \epsilon_\mathrm{y} \leqslant Y \leqslant Y_\mathrm{U} + \epsilon_\mathrm{y} \tag{8}$$

$$U_\mathrm{L} - \epsilon_\mathrm{u} \leqslant U \leqslant U_\mathrm{U} + \epsilon_\mathrm{u} \tag{9}$$

$$\varDelta U_\mathrm{min} - \epsilon_{\Delta\mathrm{u}} \leqslant \varDelta U \leqslant \varDelta U_\mathrm{max} + \epsilon_{\Delta\mathrm{u}} \tag{10}$$

$$\epsilon \geqslant 0 \tag{11}$$

where the subscripts $_\mathrm{sp}$, $_\mathrm{L}$ and $_\mathrm{U}$ stand for *setpoint*, *lower* and *upper* bound, respectively. The objective function (3) is defined with two terms: a quadratic cost term, $\Psi_i\big(Y, U\big)$, and a penalty (exact or quadratic) term, $P_i(\epsilon)$. The quadratic cost is given by

$$\Psi_i\big(Y,U\big)=\sum_{k=1}^{p} e_{i+k}^{\mathrm{T}} Q_{\mathrm{y}k} e_{i+k} + \sum_{k=1}^{m}\big(u_{i+k-1}-u_{\mathrm{r},i+k-1}\big)^{\mathrm{T}} Q_{\mathrm{u}k}\big(u_{i+k-1}-u_{\mathrm{r},i+k-1}\big)$$

where the subscript $_\mathrm{r}$ stands for *reference* [17, 18], $Q_{\mathrm{u}k}$ and $Q_{\mathrm{y}k}$ are weighting diagonal matrices, and $e_{i+k} = y_{\mathrm{sp},i+k} - y_{i+k}$. The penalty term is used only when constraint relaxation is requested, and ϵ is a measure of the original constraint violations on the states, outputs, inputs and control move rates, defined by

$$\epsilon = \left[\epsilon_{\mathrm{x}}^{\mathrm{T}}\ \epsilon_{\mathrm{y}}^{\mathrm{T}}\ \epsilon_{\mathrm{u}}^{\mathrm{T}}\ \epsilon_{\Delta\mathrm{u}}^{\mathrm{T}}\right]^{\mathrm{T}} .$$

The problem formulation is coded such that it can handle either an exact or a quadratic penalty formulation. For instance, if the penalty term is defined according to the exact penalty formulation, it follows that $P_i(\epsilon) = r^{\mathrm{T}}\epsilon$, where r is the vector of penalty parameters of appropriate size defined by: $r = [\rho \cdots \rho]^{\mathrm{T}}, \rho \in \mathbb{R}^+$. The *augmented* vectors X, Y, U and ΔU are defined by

$$X = \begin{bmatrix} x_{i+1} \\ \vdots \\ x_{i+p} \end{bmatrix},\ Y = \begin{bmatrix} y_{i+1} \\ \vdots \\ y_{i+p} \end{bmatrix},\ U = \begin{bmatrix} u_{i} \\ \vdots \\ u_{i+m-1} \end{bmatrix} \quad \text{and} \quad \Delta U = \begin{bmatrix} \Delta u_{i} \\ \Delta u_{i+1} \\ \vdots \\ \Delta u_{i+m-1} \end{bmatrix},$$

where $\Delta u_{i+k} = u_{i+k} - u_{i+k-1}$, $k = 2, \ldots, m-1$. Vectors $\Delta U_{\min}$ and $\Delta U_{\max}$ in (10) are defined as follows:

$$\Delta U_{\min} = \left[\Delta u_{\min}^{\mathrm{T}} \cdots \Delta u_{\min}^{\mathrm{T}}\right]^{\mathrm{T}}, \quad \Delta U_{\max} = \left[\Delta u_{\max}^{\mathrm{T}} \cdots \Delta u_{\max}^{\mathrm{T}}\right]^{\mathrm{T}},$$

with $\Delta u_{\min}$, $\Delta u_{\max} \in \mathbb{R}^{n_{\mathrm{m}}}$. Although in this representation it is assumed that vectors $\Delta U_{\min}$ and $\Delta U_{\max}$ are constant over the entire input predictive horizon, the implementation of a variable profile is straightforward. Equality constraints (5) result from the multiple shooting formulation and are incorporated into the optimization problem such that after convergence the state and output profiles are continuous over the predictive horizon. Note that $\phi(\bar{x}_{i+k-1}, \bar{u}_{i+k-1})$, that is, x_{i+k}, is obtained through the integration of (1) inside the sampling interval $t \in [t_{i+k-1}, t_{i+k}]$ only, using as initial conditions the initial nominal states and controls, $\bar{x}_{i+k-1}$ and $\bar{u}_{i+k-1}$ respectively. Equation (6) is the the output terminal equality constraint.

Finally, the actual implementation of the control formulation includes integral action to eliminate the steady-state offset in the process outputs resulting from step disturbances and to compensate to some extent the effect due to the model-plant mismatch. This is achieved by adding in the discrete linearized model the state equations [17, 18]

$$z_{i+k} = z_{i+k-1} + K_{\mathrm{I}}\big(y_{i+k} - y_{\mathrm{sp},i+k}\big), \quad k = 1, \ldots, p \tag{12}$$

with $z_i = z_0$, where $z_i \in \mathbb{R}^{n_{\mathrm{o}}}$, $K_{\mathrm{I}} \in \mathbb{R}^{n_{\mathrm{o}} \times n_{\mathrm{o}}}$, and z_0 is the accumulated value of steady state offset over all the past and present time instants. The constant diagonal matrix K_I determines the speed of the response of the integrator element. This feature requires an appropriate extension of the formulation (3-11). A detailed description of the derivation of the multiple shooting approach using integral action is presented in [23].

The control problem formulation is presently implemented in a computational framework (NEWCON) coded into Fortran and C^{++}. The NEWCON code features setup flags to be defined by the user such that the following features are optional: output terminal constraints, integral action, constraint relaxation (exact or quadratic penalty), and control move rate constraints.

2.2 ODE and QP Solvers

For the core elements of the NEWCON framework, the ODE solver with sensitivity analysis capabilities and the optimizer, we use highly efficient third party libraries developed as open-source and free software.

The integration of (1) to perform the predictions and to obtain sensitivity information is done using the code CVODES [11]. The code CVODES is a solver for stiff and nonstiff initial value problems for systems of ordinary differential equations. It has forward and adjoint sensitivity analysis capabilities. CVODES is part of a software family called SUNDIALS: SUite of Nonlinear and DIfferential/ALgebraic equation Solvers. It is noteworthy, that SUNDIALS is built upon generic vectors. The suite provides a serial vector implementation as well as a parallel one based on Message Passing Interface (MPI) communication protocol. A more detailed description of this code can be found in [11].

The resulting nonlinear programming problem (3–11) is solved using a successive quadratic programming (SQP) method with a line search algorithm based upon a procedure by [1]. Here the Quadratic Programming (QP) problem is solved at every iteration using a quadratic programming solver code taken from the SQP-type solver HQP for large-scale optimization problems. A more detailed description of this optimizer can be found in [9].

3 Illustrative Nonlinear Example

To illustrate the application of NEWCON we consider the simulation of a continuous pilot reactor where an exothermic zero-order reaction, A → B, occurs. This nonlinear example is taken from [23, 24], and a brief summary of the mathematical model is provided here. The total reactor mass balance gives

$$\frac{\mathrm{d}V}{\mathrm{d}t} = F_0 - F\,, \tag{13}$$

where V is the reactor liquid volume, F_0 is the inlet flow and F is the outlet flow. The mass balance to the reactant A is given by

$$\frac{\mathrm{d}C_\text{A}}{\mathrm{d}t} = \frac{F_0}{V}\,(C_{\text{A}0} - C_\text{A}) - k_0\,\mathrm{e}^{-E_\text{a}/(R\,T_\text{r})}\,. \tag{14}$$

Table 1. Model data

C_{A0}	10.	mol/l	T_{j0}	26.0	°C
C_p , C_{pj}	4184.	$\mathrm{J\,kg^{-1}\,K^{-1}}$	U	900.	$\mathrm{W\,m^{-2}\,K^{-1}}$
F_0 , F	4.0	l/min	V_j	0.014	$\mathrm{m^3}$
E_a/R	10080.	K	α_j	7.0×10^5	J/K
k_0	6.20×10^{14}	$\mathrm{mol\,m^{-3}\,s^{-1}}$	$(-\Delta H_r)$	33488.	J/mol
T_0	21.0	°C	ρ , ρ_j	1000.	$\mathrm{kg/m^3}$

Table 2. Typical steady states

Steady states	**lower**	**upper**	
h	0.30	0.30	m
C_A	7.82	4.60	mol/l
T_r	31.5	40.1	°C
T_j	28.0	28.0	°C
F_j	14.0	48.8	l/min

The reactor temperature dynamics is described by

$$\frac{\mathrm{d}T_r}{\mathrm{d}t} = \frac{F_0}{V}(T_0 - T_r) - \frac{U\,A}{\rho\,C_p\,V}(T_r - T_j) + \frac{(-\Delta H_r)}{\rho\,C_p}\,k_0\,\mathrm{e}^{-E_a/(R\,T_r)}\ , \tag{15}$$

and the jacket temperature dynamics is described by

$$\frac{\mathrm{d}T_j}{\mathrm{d}t} = \frac{1}{\rho_j\,C_{pj}\,V_j + \alpha_j}\Big[\rho_j\,C_{pj}\,F_j\,(T_{j0} - T_j) + U\,A\,(T_r - T_j)\Big]\ , \tag{16}$$

where C_{pj} is the specific heat capacity of the coolant, and F_j is the coolant flow rate. The heat transfer area is calculated from $A = \pi(r^2 + 2\,r\,h)$ with $r = 0.237\,\mathrm{m}$. Finally, the coefficient α_j in (16) stands for the contribution of the wall and spiral baffle jacket thermal capacitances. A summary of the data model is given in Table 1. Two typical steady states of this system, one stable at a lower temperature and one unstable at an upper temperature, are given in Table 2. Further details on this model are provided in [23, 24].

3.1 Simulation Results

The output variables are the reactor level and the temperature, $y^T = [\,h\ T_r\,]$, and the controls are the coolant flow rate and the outlet flow rate, $u^T = [\,F_j\ F\,]$. The following operating limits on the outputs and the controls are considered: $0.08 \leqslant h \leqslant 0.41\,\mathrm{m}$; $T_r \geqslant 0$; $0 \leqslant F_j \leqslant 76\,\mathrm{l/min}$; and $0 \leqslant F \leqslant 12\,\mathrm{l/min}$.

The results presented in Figure 2 were obtained assuming that the model is perfect and that all the state variables are measured. The output terminal constraints, integral action, control move rate constraints and constraint relaxation were turned off. These results were obtained using predictive horizons $(p, m) = (20, 5)$, a sampling time of 30 s, and diagonal weighting matrices $Q_{yk} = \mathrm{diag}(5 \times 10^2, 10^5)$ and $Q_{uk} = \mathrm{diag}(10^{-1}, 10^{-3})$, $k = 1, \cdots, p$.

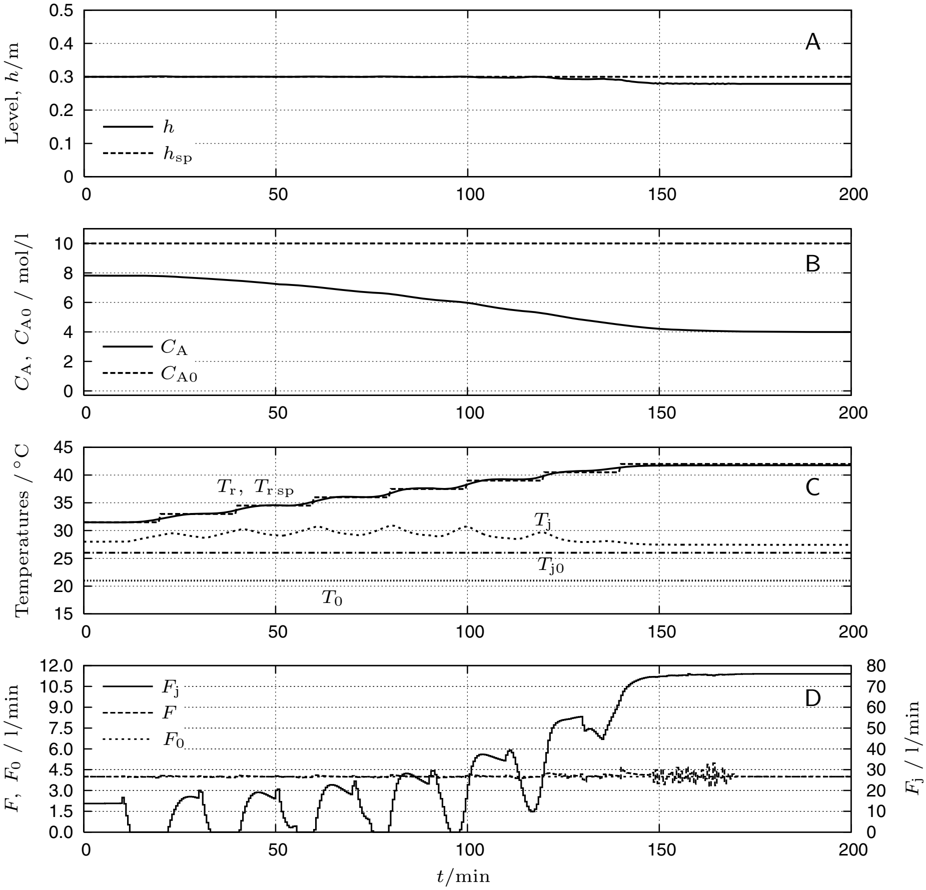


Fig. 2. Reactor closed loop response to a sequence of step changes in the reactor temperature set-point

Figure 2 shows the reactor closed loop response to a sequence of reactor temperature setpoint step changes. Note that the predictive setpoint profiles are updated in accordance to the *operator* scheduled setpoint changes. The reactor is driven to the operating conditions around the unstable steady-state (Figure 2C), to get a higher rate of conversion of reactant A (Figure 2B). One observes that the coolant flow rate reaches its upper operating constraint, 76 l/min, around $t \simeq 150$ min (Figure 2D). At this point there is no more cooling capacity available to sustain in a stable way any reactor temperature rise. To compensate for this the NMPC controller stabilizes the reactor by reducing the residence time, manipulating the outlet flow rate to drive the level to a value below the level setpoint (Figure 2A).

4 Final Remarks and Future Work

In this article we have outlined the conceptual design and the current implementation of the nonlinear model predictive control framework NEWCON as an open-source software package. An illustrative example by simulation is provided.

However, the package may benefit substantially from the following improvements that are of high priority in its development. Although the QP solver from the HQP package utilizes sparse linear algebra, the original NEWCON formulation used dense matrices. The conversion from dense to sparse matrices implies a sizable overhead. This overhead should be eliminated by formulating the optimization problem using sparse linear algebra.

Currently, the controller, together with the simulated plant, run as a single Linux process. However, following the multitasking paradigm of Linux, it is possible to use the available computing power more efficiently if the package is broken up into several independent processes, especially on multiprocessor systems.

The future work directions should include performance tests of NEWCON on real large-scale problems such as those presented in [5, 27] and the development of a state and parameter estimator, e.g., the unscented Kalman filter [13].

References

[1] Biegler, L. T., and J. E. Cuthrell (1985). “Improved Infeasible Path Optimization for Sequential Modular Simulators – II: The Optimization Algorithm”, *Computers & Chemical Engineering*, **9**(3), 257–267.

[2] Bock, H. G., and K. J. Plitt (1984). “A multiple shooting algorithm for direct solution of optimal control”, In *Proc. 9th IFAC World Congress*, Budapest, Pergamon Press, 242–247.

[3] Cannon, M. (2004). “Efficient nonlinear model predictive control algorithms”, *Annual Reviews in Control*, **28**(2), 229–237.

[4] Diehl, M., D. B. Leineweber, and A. S. Schäfer (2001). “MUSCOD-II Users’ Manual”, IWR Preprint 2001-25, University of Heidelberg.

[5] Diehl, M., R. Findeisen, S. Schwarzkopf, I. Uslu, F. Allgöwer, H. G. Bock, and J. P. Schlöder (2003). ”An efficient approach for nonlinear model predictive control of large-scale systems. Part II: Experimental evaluation for a distillation column”, *Automatisierungstechnik*, **51**(1), 22–29.

[6] Eaton, J. W. (2002) “GNU Octave Manual”, Network Theory Limited.

[7] Eder, H. H. (2003). “Advanced process control: Opportunities, benefits, and barriers”, *IEE Computing & Control Engineering*, **14**(5), 10–15.

[8] Franke, R., and E. Arnorld (1996). “Applying new numerical algorithms to the solution of discreet-time optimal control problems”, In *Proc. 2nd IEEE European Workshop on Computer Intensive Methods in Control and Signal Processing*, Budapest, 67–72.

[9] Franke, R. (1998). “Omuses a tool for the Optimization of multistage systems and HQP a solver for sparse nonlinear optimization, Version 1.5”, Technical report, Technical University of Ilmenau, Germany.

[10] Hawkins, R. E. (2004). “The economics of Open Source Software for a Competitive Firm”, *NetNomics*, **6**(2), 103–117.

[11] Hindmarsh, A. C., and R. Serban (April 2005). “User Documentation for CVODES v2.2.0”, Technical report UCRL-SM-208111, Center for Applied Scientific Computing Lawrence Livermore National Laboratory.

[12] Ihaka, R., and R. Gentleman (1996). “R: A Language for Data Analysis and Graphics”, *Journal of Computational and Graphical Statistics*, **5**(3), 299–314.

[13] Julier, S. J., and J. K. Uhlmann (2004). “Unscented Filtering and Nonlinear Estimation”, *IEEE Transactions on Control Systems Technology*, **2**(3), 169-182.

[14] Lerner, J., and J. Tirole (2002). “Some Simple Economics of Open Source”, *The Journal of Industrial Economics*, **50**(2), 197–234.

[15] Nagy, Z., and R. D. Braatz (2003). “Robust Nonlinear Model Predictive Control of Batch Processes”, *AIChE Journal*, **49**(7), 1776–1786.

[16] Ohtsuka, T. (2004). ”A continuation/GMRES method for fast computation of nonlinear receding horizon control”, *Automatica*, **40**(4), 563–574.

[17] Oliveira, N. M. C. (1994). “Constraint Handling and Stability Properties of Model Predictive Control”, PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania.

[18] Oliveira, N. M. C., and L. T. Biegler (1995). “Newton-type Algorithms for Nonlinear Process Control. Algorithm and Stability Results”, *Automatica*, **31**(2), 281–286.

[19] Piela, P. C., T. G. Epperly, K. M. Westerberg, and A. W. Westerberg (1991). “ASCEND: An Object Oriented Computer Environment for Modeling and Analysis — The Modeling Language”, *Computers & Chemical Engineering*, **15**(1), 53–72.

[20] Qin, S. J., and T. A. Badgwell (2003). “A survey of industrial model predictive control technology”, *Control Engineering Practice*, **11**(7), 733-764.

[21] Romanenko, A. (2003). “Open-source software solutions in chemical process engineering – present status and perspectives”, *Proceedings of the ISA EXPO 2003 Technical Conference, Houston, USA*, 21–23.

[22] Santos, L. O., N. M. C. Oliveira, and L. T. Biegler (1995). “Reliable and Efficient Optimization Strategies for Nonlinear Model Predictive Control”, *Proc. of DYCORD+’95, Helsingør, Denmark* (J. B. Rawlings, Ed.), Elsevier Science, Oxford, 33–38.

[23] Santos, L. O. (2001). “Multivariable Predictive Control of Chemical Processes”, PhD thesis, Faculdade de Ciências e Tecnologia, Universidade de Coimbra, Coimbra, Portugal.

[24] Santos, L. O., P. A. F. N. A. Afonso, J. A. A. M. Castro, N. M. C. Oliveira, and L. T. Biegler (2001). “On-Line Implementation of Nonlinear MPC: An Experimental Case Study”, *Control Engineering Practice*, **9**, 847–857.

[25] Spedding, V. (2002). “Open Doors for Open Source”, *Scientific Computing World*, **66**, 11–13.

[26] Tenny, M. J., S. J. Wright, and J. B. Rawlings (2004). “Nonlinear model predictive control via feasibility-perturbed sequential quadratic programming”, *Computational Optimization and Applications*, **28**(1), 87–121.

[27] Young, R. E., R. D. Bartusiak and R. W. Fontaine (2001). “Evolution of an industrial nonlinear model predictive controller”, In *J. B. Rawlings, B. A. Ogunnaike, and J.W. Eaton, editors, Chemical Process Control VI: Sixth International Conference on Chemical Process Control, Tucson, Arizona*, AIChE Symposium Series, **98**(326), 342–351.

Robustness and Robust Design of MPC for Nonlinear Discrete-Time Systems

Lalo Magni[1] and Riccardo Scattolini[2]

[1] Dipartimento di Informatica e Sistemistica, University of Pavia, Italy
lalo.magni@unipv.it

[2] Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy
riccardo.scattolini@elet.polimi.it

1 Introduction

In view of the widespread success of Model Predictive Control (MPC), in recent years attention has been paid to its robustness characteristics, either by examining the robustness properties inherent to stabilizing MPC algorithms, or by developing new MPC methods with enhanced robustness properties.

By restricting attention to nonlinear systems, this paper presents in a unified framework some of the robustness results available for nonlinear MPC. Specifically, the first part of the paper is concerned with the introduction of the main definitions and of the general results used in the sequel as well as with the description of a “prototype” nominal MPC algorithm with stability. Then, the considered class of model uncertainties and disturbances are defined.

In the second part of the paper, the inherent robustness properties of MPC algorithms designed on the nominal model are reviewed under the main assumption that the problem is unconstrained and feasibility is always guaranteed. The results reported rely on the decreasing property of the optimal cost function [5], [6], [42], [25]. Further robustness characteristics can be derived by showing that unconstrained MPC is inversely optimal, and as such has gain and phase margins [34].

The last part of the paper is devoted to present the approaches followed so far in the design of MPC algorithms with robustness properties for uncertain systems. A first method consists in minimizing a nominal performance index while imposing the fulfillment of constraints for each admissible disturbance, see [24]. This calls for the inclusion in the problem formulation of tighter state, control and terminal constraints and leads to very conservative solutions or even to unfeasible problems. With a significant increase of the computational burden, an alternative approach consists in solving a min-max optimization problem. Specifically, in an open-loop formulation the performance index is minimized with respect to the control sequence and maximized with respect to the disturbance sequence over the prediction horizon. However, this solution is still unsatisfactory, since the minimization with respect to a single control profile does not solve the feasibility problem. This drawback can be avoided as in [3], where the

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 239–254, 2007.
springerlink.com

MPC control law is applied to an already robust stable system. Alternatively, the intrinsic feedback nature of every Receding Horizon (*RH*) implementation of *MPC* can be exploited by performing optimization with respect to closed-loop strategies, as discussed in [8], [29], [30], [33] where robust algorithms have been proposed for systems with perturbations vanishing at the origin.

2 Notations and Basic Definitions

We use Z_+ to denote the set of all nonnegative integers. Euclidean norm is denoted simply as $|\cdot|$. For any function $\phi : Z_+ \to R^n$, $\|\phi\| = \sup\{|\phi(k)| : k \in Z_+\} \leq \infty$. B_r is the closed ball of radius r, i.e. $B_r = \{x \in R^n | \, |x| \leq r\}$.

A continuous function $\alpha(\cdot) : R_+ \to R_+$ is a $\mathcal{K}$ function if $\alpha(0) = 0$, $\alpha(s) > 0$ for all $s > 0$ and it is strictly increasing. A continuous function $\beta : R_+ \times Z_+ \to R_+$ is a $\mathcal{KL}$ function if $\beta(s,t)$ is a $\mathcal{K}$ function in s for any $t \geq 0$ and for each $s > 0$ $\beta(s,\cdot)$ is decreasing and $\beta(s,t) \to 0$ as $t \to \infty$. $\mathcal{M}_\Omega$ is the set of signals in some subset Ω.

Definition 1 (Stability). *[22], [23] Given the discrete-time dynamic system*

$$x(k+1) = f(x(k)), k \geq t, \;\; x(t) = \bar{x} \tag{1}$$

with $f(0) = 0$ and a set $\Xi \subseteq R^n$ with the origin as an interior point:

1. *the origin is an asymptotically stable equilibrium in Ξ if $\forall \varepsilon > 0$ $\exists \delta > 0$ such that $\forall \bar{x} \in \Xi$ with $|\bar{x}| \leq \delta$, $|x(k)| < \varepsilon$, $k \geq t$, and $lim_{k\to\infty} |x(k)| \to 0$;*
2. *the origin is a locally exponentially stable equilibrium point if there exist positive constants δ, α and $\rho < 1$ such that for any $\bar{x} \in B_\delta$, $|x(k)| < \alpha |\bar{x}| \rho^{k-t}$, $k \geq t$;*
3. *the origin is an exponentially stable equilibrium point in Ξ if there exist positive constants α and $\rho < 1$ such that for any $\bar{x} \in \Xi$, $|x(k)| < \alpha |\bar{x}| \rho^{k-t}$, $k \geq t$.*

Definition 2 (Lyapunov function). *[23] A function $V(\cdot)$ is called a Lyapunov function for system (1) if there exist two sets Ξ_1 and Ξ_2 with $\Xi_1 \subseteq \Xi_2$ and $\mathcal{K}$ functions α_1, α_2, and α_3 such that*

$$\begin{aligned} V(x) &\geq \alpha_1(|x|), \forall x \in \Xi_2 \\ V(x) &\leq \alpha_2(|x|), \forall x \in \Xi_1 \\ \Delta V(x) = V(f(x)) - V(x) &\leq -\alpha_3(|x|), \forall x \in \Xi_2 \end{aligned} \tag{2}$$

Lemma 1. *[23]Let Ξ_2 be a positive invariant set for system (1) that contains a neighborhood Ξ_1 of the origin and let $V(\cdot)$ be an associated Lyapunov function. Then:*

1. *the origin is an asymptotically stable equilibrium in Ξ_2;*
2. *if $\alpha_1(|x|) = \alpha_1 |x|^p$, $\alpha_2(|x|) = \alpha_2 |x|^p$, $\alpha_3(|x|) = \alpha_3 |x|^p$, for some real positive α_1, α_2, α_3 and p, the origin is locally exponentially stable. Moreover, if the inequality (2) holds for $\Xi_1 = \Xi_2$, then the origin is exponentially stable in Ξ_2.*

Definition 3 (Output admissible set). *Consider the system*

$$x(k+1) = f(x(k), u(k)),\ k \geq t,\ x(t) = \bar{x} \tag{3}$$

where k is the discrete time index, $x(k) \in R^n$, $u(k) \in R^m$, and $f(0,0) = 0$. The state and control variables are required to fulfill the following constraints

$$x \in X, u \in U \tag{4}$$

where X and U are compact subsets of R^n and R^m, respectively, both containing the origin as an interior point. Consider the control law

$$u = \kappa(x). \tag{5}$$

Then, the term output admissible set [11], referred to the closed-loop system (3), (5) denotes a positively invariant set $\bar{X} \subseteq X$ which is a domain of attraction of the origin and such that $\bar{x} \in \bar{X}$ implies $\kappa(x(k)) \in U$, $k \geq t$.

3 Nominal Model Predictive Control

Given the system (3) and the state and control constraints (4), we assume that $f(\cdot,\cdot)$ is a C^1 function with Lipschitz constant L_f, $\forall x \in X$ and $\forall u \in U$.

To introduce the MPC algorithm, first let $u_{t_1,t_2} := [u(t_1)\ u(t_1+1) \ldots u(t_2)]$, $t_2 \geq t_1$, then define the following finite-horizon optimization problem.

Definition 4 (FHOCP). *Consider a stabilizing auxiliary control law $\kappa_f(\cdot)$ and an associated output admissible set X_f. Then, given the positive integer N, the stage cost $l(\cdot,\cdot)$ and the terminal penalty $V_f(\cdot)$, the Finite Horizon Optimal Control Problem ($FHOCP$) consists in minimizing, with respect to $u_{t,t+N-1}$, the performance index*

$$J(\bar{x}, u_{t,t+N-1}, N) = \sum_{k=t}^{t+N-1} l(x(k), u(k)) + V_f(x(t+N)) \tag{6}$$

subject to

(i) the state dynamics (3) with $x(t) = \bar{x}$;
(ii) the constraints (4), $k \in [t, t+N-1]$;
(iii) the terminal state constraint $x(t+N) \in X_f$.

It is now possible to define a "prototype" *Nonlinear Model Predictive Control* ($NMPC$) algorithm*:* at every time instant t, define $\bar{x} = x(t)$ and find the optimal control sequence $u^o_{t,t+N-1}$ by solving the $FHOCP$. Then, according to the Receding Horizon approach, define

$$\kappa^{MPC}(\bar{x}) = u^o_{t,t}(\bar{x}) \tag{7}$$

where $u^o_{t,t}(\bar{x})$ is the first column of $u^o_{t,t+N-1}$, and apply the control law

$$u = \kappa^{MPC}(x) \tag{8}$$

In order to guarantee the stability of the origin of the closed-loop system (3), (8), many different choices of the stabilizing control law $\kappa_f(\cdot)$, of the terminal set X_f and of the terminal cost function V_f have been proposed in the literature, see [37], [40], [2], [6], [28], [32], [7], [16], [14], [20]. Irrespective of the specific algorithm applied, a general result can be stated under the following assumptions which will always be considered in the sequel.

Assumption 3.1. *$l(x,u)$ is Lipschitz with Lipschitz constant L_l and is such that $\alpha_l(|x|) \leq l(x,u) \leq \beta_l(|(x,u)|)$ where α_l and β_l are $\mathcal{K}$ functions.*

Assumption 3.2. *Let $\kappa_f(\cdot)$, $V_f(\cdot)$, X_f be such that*

1. *$X_f \subseteq X$, X_f closed, $0 \in X_f$*
2. *$\kappa_f(x) \in U$, $\forall x \in X_f$*
3. *$\kappa_f(x)$ is Lipschitz in X_f with Lipschitz constant $L_{\kappa f}$*
4. *$f(x, \kappa_f(x)) \in X_f$, $\forall x \in X_f$*
5. *$\alpha_{V_f}(|x|) \leq V_f(x) \leq \beta_{V_f}(|x|)$, α_{V_f} and β_{V_f} $\mathcal{K}$ functions*
6. *$V_f(f(x,\kappa_f(x))) - V_f(x) \leq -l(x,\kappa_f(x))$, $\forall x \in X_f$*
7. *V_f is Lipschitz in X_f with Lipschitz constant L_{Vf}*

Theorem 1. *Let $X^{MPC}(N)$ be the set of the states such that a feasible solution for the $FHOCP$ exists. Given an auxiliary control law κ_f, a terminal set X_f, a terminal penalty V_f and a cost $l(\cdot,\cdot)$ satisfying Assumptions 3.1, 3.2, the origin is an asymptotically stable equilibrium point for the closed-loop system formed by (3) and (8) with output admissible set $X^{MPC}(N)$ and $V(\bar{x},N) := J(\bar{x}, u^o_{t,t+N-1}, N)$ is an associated Lyapunov function. Moreover if $\alpha_l(|x|) = \alpha_l |x|^p$, $\beta_{V_f}(|x|) = \beta_{V_f} |x|^p$, $p > 0$, then the origin is an exponentially stable equilibrium point in $X^{MPC}(N)$.*

Proof of Theorem 1. First note that

$$V(x,N) := J(x, u^o_{t,t+N-1}, N) \geq l(x, \kappa^{MPC}(x)) \geq \alpha_l(|x|) \tag{9}$$

Moreover, letting $u^o_{t,t+N-1}$ be the solution of the $FHOCP$ with horizon N at time t, in view of Assumption 3.2

$$\tilde{u}_{t,t+N} = [u^o_{t,t+N-1}, \kappa_f(x(t+N))]$$

is an admissible control sequence for the $FHOCP$ with horizon $N+1$ with

$$J(x, \tilde{u}_{t,t+N}, N+1) = V(x,N) - V_f(x(t+N)) + V_f(x(t+N+1))$$
$$+ l(x(t+N), \kappa_f(x(t+N))) \leq V(x,N)$$

so that

$$V(x,N+1) \leq V(x,N), \quad \forall x \in X^{MPC}(N) \tag{10}$$

with $V(x,0) = V_f(x)$, $\forall x \in X_f$. Then

$$V(x,N+1) \leq V(x,N) \leq V_f(x) \leq \beta_{V_f}\left(|x|\right), \quad \forall x \in X_f \tag{11}$$

Finally

$$\begin{aligned} V(x,N) &= l(x,\kappa^{MPC}(x)) + J(f(x,\kappa^{MPC}(x)), u^o_{t+1,t+N-1}, N-1) \\ &\geq l(x,\kappa^{MPC}(x)) + V(f(x,\kappa^{MPC}(x)), N) \\ &\geq \alpha_l(|x|) + V(f(x,\kappa^{MPC}(x)), N), \quad \forall x \in X^{MPC}(N) \end{aligned} \tag{12}$$

Then, in view (9), (11) and (12) $V(x,N)$ is a Lyapunov function and in view of Lemma 1 the asymptotic stability in $X^{MPC}(N)$ and the exponential stability in X_f are proven. In order to prove exponential stability in $X^{MPC}(N)$, let B_ρ be the largest ball such that $B_\rho \in X_f$ and $\bar{V}$ be a constant such that $V(x,N) \leq \bar{V}$ for all $x \in X^{MPC}(N)$. Now define

$$\bar{\alpha}_2 = \max\left(\frac{\bar{V}}{\rho^p}, \beta_{V_f}\right),$$

then it is easy to see [27] that

$$V(x,N) \leq \bar{\alpha}_2 \left|x\right|^p, \quad \forall x \in X^{MPC}(N) \tag{13}$$

4 Robustness Problem and Uncertainty Description

Let the uncertain system be described by

$$x(k+1) = f(x(k),u(k)) + g(x(k),u(k),w(k)),\ k \geq t,\ x(t) = \bar{x} \tag{14}$$

or equivalently

$$x(k+1) = \tilde{f}(x(k),u(k),w(k)),\ k \geq t,\ x(t) = \bar{x} \tag{15}$$

In (14), $f(x,u)$ is the nominal part of the system, $w \in \mathcal{M}_{\mathcal{W}}$ for some compact subset $\mathcal{W} \subseteq R^p$ is the disturbance and $g(\cdot,\cdot,\cdot)$ is the uncertain term assumed to be Lipschitz with respect to all its arguments with Lipschitz constant L_g.

The perturbation term $g(\cdot,\cdot,\cdot)$ allows one to describe modeling errors, aging, or uncertainties and disturbances typical of any realistic problem. Usually, only partial information on $g(\cdot,\cdot,\cdot)$ is available, such as an upper bound on its absolute value $\left|g(\cdot,\cdot,\cdot)\right|$.

For the robustness analysis the concept of Input to State Stability (ISS) is a powerful tool.

Definition 5 (Input-to-state stability). *The system*

$$x(k+1) = f(x(k),w(k)), k \geq t,\ x(t) = \bar{x} \tag{16}$$

with $w \in \mathcal{M}_{\mathcal{W}}$ is said to be ISS in Ξ if there exists a $\mathcal{KL}$ function β, and a $\mathcal{K}$ function γ such that

$$\left|x(k)\right| \leq \beta(\left|\bar{x}\right|, k) + \gamma\left(\|w\|\right),\ \forall k \geq t, \forall \bar{x} \in \Xi$$

Definition 6 (ISS-Lyapunov function). *A function $V(\cdot)$ is called an ISS-Lyapunov function for system (16) if there exist a set Ξ, $\mathcal{K}$ functions α_1, α_2, α_3, and σ such that*

$$\begin{aligned} V(x) &\geq \alpha_1(|x|), \forall x \in \Xi \\ V(x) &\leq \alpha_2(|x|), \forall x \in \Xi \\ \Delta V(x,w) = V(f(x,w)) - V(x) &< -\alpha_3(|x|) + \sigma(|w|), \forall x \in \Xi, \forall w \in \mathcal{M}_{\mathcal{W}} \end{aligned} \tag{17}$$

Note that if the condition on ΔV is fulfilled with $\sigma(\cdot) = 0$, then the origin is asymptotically stable for any considered disturbance w.

Lemma 2. *[21] Let Ξ be a positive invariant set for system (16) that contains the origin and let $V(\cdot)$ be a ISS-Lyapunov function for system (16), then the system (16) is ISS in Ξ.*

5 Inherent Robustness of Nominal *MPC*

In this section, the robustness properties of nominal MPC algorithms are reviewed under the fundamental assumption that the presence of uncertainties and disturbances do not cause any loss of feasibility. This holds true when the problem formulation does not include state and control constraints and when any terminal constraint used to guarantee nominal stability can be satisfied also in perturbed conditions.

5.1 Inverse Optimality

It is well known that the control law solving an unconstrained optimal Infinite Horizon (IH) problem guarantees robustness properties both in the continuous and in the discrete time cases, see [12], [43], [10], [1]. Hence, the same robustness characteristics can be proven for MPC regulators provided that they can be viewed as the solution of a suitable IH problem. For continuous time systems, this has been proven in [34], while in the discrete time case, from the optimality principle we have

$$V(x,N) = \bar{l}(x(k), \kappa^{MPC}(x(k))) + V(f(x, \kappa^{MPC}(x)), N)$$

with

$$\begin{aligned} \bar{l}(x(k), \kappa^{MPC}(x(k))) : &= l(x(k), \kappa^{MPC}(x(k))) - V(f(x, \kappa^{MPC}(x)), N) \\ &+ V(f(x, \kappa^{MPC}(x)), N-1) \end{aligned}$$

Then $\kappa^{MPC}(x(k))$ is the solution of the Hamilton-Jacobi-Bellman equation for the IH optimal control problem with stage cost $\bar{l}(x,u)$. In view of Assumption 3.2 and (10) it follows that

$$\bar{l}(x(k), \kappa^{MPC}(x(k))) > l(x(k), \kappa^{MPC}(x(k)))$$

so that the stage cost is well defined and robustness of IH is guaranteed. Specifically, under suitable regularity assumptions on V, in [5] it has been shown that MPC regulators provide robustness with respect to gain perturbations due to actuator nonlinearities and additive perturbations describing unmodeled dynamics. Further results on gain perturbations can be achieved as shown in [1].

5.2 Robustness Provided by the *ISS* Property

The robustness analysis provided by ISS, see also [21], can be summarized by the following result.

Theorem 2. *Under Assumptions 3.1 and 3.2, if $V(x,N)$ is Lipschitz with Lipschitz constant L_V, the closed-loop system (14), (8) is ISS in $X^{MPC}(N)$ for any perturbation $g(x,u,w)$ such that $|g(x,u,0)| < \frac{\rho}{L_V}\alpha_l(|x|)$ where $0<\rho<1$ is an arbitrary real number.*

Proof. Note that (9) and (13) still hold. Moreover

$$\begin{aligned}
&V(\tilde{f}(x,\kappa^{MPC}(x),w),N) - V(f(x,\kappa^{MPC}(x),N)\\
&\le L_V\left|g(x,\kappa^{MPC}(x),w)\right| \le L_V\left|g(x,\kappa^{MPC}(x),0)\right| + L_V L_g\,|w|\\
&\le \rho\alpha_l(|x|) + L_V L_g\,|w|
\end{aligned}$$

Hence

$$\begin{aligned}
&V(\tilde{f}(x,\kappa^{MPC}(x),w),N)\\
&\le V(x,N) - (1-\rho)\alpha_l(|x|) + L_V L_g\,|w|
\end{aligned}$$

Remark 1. If $w=0$ the result is equivalent to the one on robust stability reported in [6]. On the contrary if $w \neq 0$ then ISS guarantees that the system evolves towards a compact set which size depends on the bound on w. A way to estimate this size is given in [31]. Further results on the robustness with bounded and exponentially decaying disturbances are reported in [42], [25].

Remark 2. All the above results assume some regularity of the MPC control law and of the value function, see also [18], [19], [26]. It is well known that the MPC control law could be even discontinuous [38]. In [13], some examples of the loss of robustness have been presented. For a specific discussion on robustness of discontinuous MPC see [9].

6 Robust *MPC* Design with Restricted Constraints

The development of MPC algorithms robust with respect to persistent disturbances has received a great deal of attention both for linear systems, see e.g. [4], and in the nonlinear case. An approach to overcome the feasibility and stability problems consists in minimizing a nominal performance index while imposing the constraints fulfillment for any admissible disturbance. This implies the use of tighter and tighter state, control and terminal constraints, so leading to very conservative solutions or even to unfeasible problems. Algorithms with these characteristics have been described in [39] for continuous-time and in [24] for discrete time systems. The technique presented in [24] is now briefly summarized. To this aim, the following assumption must be introduced to allow for the analysis of the (worst-case) effects of the disturbance.

Assumption 6.1. *The uncertain term in (14) is bounded by* γ, *that is* $|g(\cdot,\cdot,\cdot)| \leq \gamma$ *for any* x *and* u *satisfying (4) and* $w \in \mathcal{M}_{\mathcal{W}}$.

In order to guarantee that at any future time instant in the prediction horizon the disturbance does not cause the state constraints violation, first introduce the following definition.

Definition 7 (Pontryagin difference). *Let* $A, B \subset R^n$, *be two sets, then the Pontryagin difference set is defined as* $A \sim B = \{x \in R^n | x + y \in A, \forall y \in B\}$.

Consider now the following sets $X_j = X \sim B_\gamma^j$ where B_γ^j is defined as

$$B_\gamma^j = \left\{ z \in R^n : |z| \leq \frac{L_f^j - 1}{L_f - 1}\gamma \right\}.$$

Definition 8 (NRFHOCP). *Consider a stabilizing auxiliary control law* $\kappa_f(\cdot)$ *and an associated output admissible set* X_f. *Then, given the positive integer* N, *the stage cost* $l(\cdot,\cdot)$ *and the terminal penalty* $V_f(\cdot)$, *the Nominal Robust Finite Horizon Optimal Control Problem (NRFHOCP) consists in minimizing, with respect to* $u_{t,t+N-1}$,

$$J(\bar{x}, u_{t,t+N-1}, N) = \sum_{k=t}^{t+N-1} l(x(k), u(k)) + V_f(x(t+N))$$

subject to:

(i) the state dynamics (3) with $x(t) = \bar{x}$;
(ii) the constraints $u(k) \in U$ *and* $x(k) \in X_{k-t+1}$, $k \in [t, t+N-1]$, *where* X_{k-t+1} *are given in Definition 7;*
(iii) the terminal state constraint $x(t+N) \in X_f$.

From the solution of the $NRFHOCP$, the Receding Horizon control law

$$u = \kappa^{MPC}(x) \tag{18}$$

is again obtained as in (7) and (8). Concerning the stability properties of the closed-loop system, the following hypothesis substitutes Assumption 3.2.

Assumption 6.2. *Let* $\kappa_f(\cdot)$, $V_f(\cdot)$, X_f *such that*

1. $\Phi_f := \{x \in R^n : V_f(x) \leq \alpha\} \subseteq X$, Φ_f *closed,* $0 \in \Phi_f$, α *positive constant*
2. $\kappa_f(x) \in U$, $\forall x \in \Phi_f$

3. $f(x, \kappa_f(x)) \in \Phi_f$, $\forall x \in \Phi_f$
4. $V_f(f(x, \kappa_f(x))) - V_f(x) \leq -l(x, \kappa_f(x))$, $\forall x \in \Phi_f$
5. $\alpha_{V_f}(|x|) \leq V_f(x) \leq \beta_{V_f}(|x|)$, α_{V_f}, β_{V_f} *are* $\mathcal{K}$ *functions*
6. $V_f(\cdot)$ *is Lipschitz in* Φ_f *with a Lipschitz constant* L_{Vf}
7. $X_f := \{x \in R^n : V_f(x) \leq \alpha_v\}$ *is such that for all* $x \in \Phi_f$, $f(x, \kappa_f(x)) \in X_f$, α_v *positive constant*

Then, the final theorem can be stated.

Theorem 3. *[24]Let* $X^{MPC}(N)$ *be the set of states of the system where there exists a solution of the* $NRFHOCP$*. Then the closed loop system (14), (18) is* ISS *in* $X^{MPC}(N)$ *if Assumption 6.1 is satisfied with*

$$\gamma \leq \frac{\alpha - \alpha_v}{L_{Vf} L_f^{N-1}}$$

The above robust synthesis method ensures the feasibility of the solution through a wise choice of the constrains (ii) and (iii) in the $NRFHOPC$ formulation. However, the solution can be extremely conservative or may not even exist, so that less stringent approaches are advisable.

7 Robust *MPC* Design with Min-Max Approaches

The design of MPC algorithms with robust stability has been first placed in an H_∞ setting in [44] for linear unconstrained systems. Since then, many papers have considered the linear constrained and unconstrained case, see for example [41]. For nonlinear continuous time systems, H_∞-MPC control algorithms have been proposed in [3], [30], [8], while discrete-time systems have been studied in [29], [15], [17], [33], [36]. In [29] the basic approach consists in solving a min-max problem where an H_∞-type cost function is maximized with respect to the admissible disturbance sequence, i.e. the "nature", and minimized with respect to future controls over the prediction horizon. The optimization can be solved either in open-loop or in closed-loop. The merits and drawbacks of these solutions are discussed in the sequel.

7.1 Open-Loop Min-Max *MPC*

Assume again that the perturbed system is given by

$$x(k+1) = \tilde{f}(x(k), u(k), w(k)),\ k \geq t,\ x(t) = \bar{x} \tag{19}$$

where now $\tilde{f}(\cdot, \cdot, \cdot)$ is a known Lipschitz function with Lipschitz constant $L_{\tilde{f}}$ and $\tilde{f}(0, 0, 0) = 0$. The state and control variables must satisfy the constraints (4), while the disturbance w is assumed to fulfill the following hypothesis.

Assumption 7.1. *The disturbance* w *is contained in a compact set* $\mathcal{W}$ *and there exists a* $\mathcal{K}$ *function* $\gamma(\cdot)$ *such that* $|w| \leq \gamma(|(x, u)|)$.

Letting $w_{t_1,t_2} := [w(t_1)\ w(t_1+1)\ \ldots\ w(t_2)]$, $t_2 \geq t_1$, the optimal min-max problem can now be stated.

Definition 9 (FHODG). *Consider a stabilizing auxiliary control law $\kappa_f(\cdot)$ and an associated output admissible set X_f. Then, given the positive integer N, the stage cost $l(\cdot,\cdot) - l_w(\cdot)$ and the terminal penalty $V_f(\cdot)$, the Finite Horizon Open-loop Differential Game (FHODG) problem consists in minimizing, with respect to $u_{t,t+N-1}$ and maximizing with respect to $w_{t,t+N-1}$ the cost function*

$$J(\bar{x}, u_{t,t+N-1}, w_{t,t+N-1}) = \sum_{k=t}^{t+N-1} \{l(x(k), u(k)) - l_w(w(k))\} + V_f(x(t+N))$$

subject to:

(i) the state dynamics (19) with $x(t) = \bar{x}$;
(ii) the constraints (4), $k \in [t, t+N-1]$;
(iii) the terminal state constraint $x(t+N) \in X_f$.

Once the *FHODG* is solved and the optimal control sequence $u^o_{t,t+N-1}$ is available, according to the *RH* principle the feedback control law is again given by (7) and (8). To achieve robustness the idea could be to use a terminal set and a terminal penalty satisfying the following “robust” version of the sufficient conditions reported in Assumption 3.2.

Assumption 7.2. *Let $\kappa_f(\cdot)$, $V_f(\cdot)$, X_f such that*

1. $X_f \subseteq X$, X_f *closed,* $0 \in X_f$
2. $\kappa_f(x) \in U$, $\forall x \in X_f$
3. $\tilde{f}(x, \kappa_f(x), w) \in X_f, \forall x \in X_f, \forall w \in \mathcal{W}$
4. $\alpha_{V_f}(|x|) \leq V_f(x) \leq \beta_{V_f}(|x|)$, α_{V_f} *and* β_{V_f} $\mathcal{K}$ *functions*
5. $V_f(\tilde{f}(x, \kappa_f(x), w)) - V_f(x) \leq -l(x,u) + l_w(w)$, $\forall x \in X_f, \forall w \in \mathcal{W}$
6. V_f *is Lipschitz in* X_f *with Lipschitz constant* L_{Vf}

Along this line, one could argue again that the value function $V(x) = J(\bar{x}, u^o_{t,t+N-1}, w^o_{t,t+N-1})$ is a candidate to prove the stability of the closed-loop system. However, the following fundamental feasibility problem arises. Suppose that at time t an optimal (hence admissible) control sequence $u^o_{t,t+N-1}$ for the *FHODG* is known. In other words, irrespective of the specific realization of w, this sequence steers the state x to X_f in N steps or less; hence, the abbreviated control sequence $u^o_{t+1,t+N-1}$ steers the state $x(t+1)$ to X_f at most in $N-1$ steps. Now, the major difficulty is to obtain a feasible control sequence $\tilde{u}_{t+1,t+N} := \left[u^o_{t+1,t+N-1}, v\right]$ required to complete the stability proof (see the proof of Theorem 1). In fact, Assumption 7.2 does not ensure the existence of a signal v with this property since the auxiliary control law $\kappa_f(x(t+N))$ can only provide a control value depending on $x(t+N)$, which in turn is a function of the particular realization of the disturbance w.

One way to avoid this impasse is given in [3] where the *MPC* approach is applied to an already robust stable system, so that Assumption 7.2 is satisfied with $\kappa_f(\cdot) \equiv 0$. In this case a feasible control sequence is

$$\tilde{u}_{t+1,t+N} := \left[u^o_{t+1,t+N-1}, 0\right]$$

In order to obtain a system with a-priori robustness properties with respect to the considered class of disturbances, in [3] it has been suggested to precompensate the system under control by means of an inner feedback loop designed for example with the H_∞ approach.

7.2 Closed-Loop Min-Max *MPC*

The limitations of the open-loop min-max approach can be overcome by explicitly accounting for the intrinsic feedback nature of any *RH* implementation of *MPC*, see e.g. [41] for the linear case and [29] for nonlinear systems. In this approach, at any time instant the controller chooses the input u as a function of the current state x, so as to guarantee that the effect of the disturbance w is compensated for any choice made by the "nature". Hence, instead of optimizing with respect to a control sequence, at any time t the controller has to choose a sequence of control laws $\kappa_{t,t+N-1} = [\kappa_0(x(t))\ \kappa_1(x(t+1)\ \ldots\ \kappa_{N-1}(x(t+N-1)]$. Then, the following optimal min-max problem can be stated.

Definition 10 (FHCDG). *Consider a stabilizing auxiliary control law $\kappa_f(\cdot)$ and an associated output admissible set X_f . Then, given the positive integer N, the stage cost $l(\cdot,\cdot) - l_w(\cdot)$ and the terminal penalty $V_f(\cdot)$, the Finite Horizon Closed-loop Differential Game (FHCDG) problem consists in minimizing, with respect to $\kappa_{t,t+N-1}$ and maximizing with respect to $w_{t,t+N-1}$ the cost function*

$$J(\bar{x}, \kappa_{t,t+N-1}, w_{t,t+N-1}, N) = \sum_{k=t}^{t+N-1} \{l(x(k), u(k)) - l_w(w(k))\} + V_f(x(t+N))$$

subject to:

(i) the state dynamics (19) with $x(t) = \bar{x}$;
(ii) the constraints (4), $k \in [t, t+N-1]$;
(iii) the terminal state constraint $x(t+N) \in X_f$.

Finally, letting $\kappa^o_{t,t+N-1}$, $w^o_{t,t+N-1}$ the solution of the *FHCDG* the feedback control law $u = \kappa^{MPC}(x)$ is obtained by setting

$$\kappa^{MPC}(x) = \kappa^o_0(x) \tag{20}$$

where $\kappa^o_0(x)$ is the first element of $\kappa^o_{t,t+N-1}$.

In order to derive the main stability and performance properties associated to the solution of *FHCDG*, the following assumption is introduced.

Assumption 7.3. *$l_w(\cdot)$ is such that $\alpha_w(|w|) \le l_w(w) \le \beta_w(|w|)$ where α_w and β_w are $\mathcal{K}$ functions.*

Then, the following result holds.

Theorem 4. *Let $X^{MPC}(N)$ be the set of states of the system where there exists a solution of the $FHCDG$ and $\kappa_{t,t+N-1}$ a vector of Lipschitz continuous control policies. Under Assumptions 7.1-7.3 the closed loop system Σ^{MPC} given by (19)-(20) is ISS with robust output admissible set $X^{MPC}(N)$, moreover if $\gamma(\cdot)$ is such that $\beta_w\left(\gamma\left(\left|x,\kappa^{MPC}(x)\right|\right)\right) - \alpha_l(|x|) < -\delta(|x|)$, where δ is a $\mathcal{K}$ function, the origin of the closed loop system Σ^{MPC} given by (19)-(20) is robustly asymptotically stable.*

Proof. First note that in view of Assumption 7.2, given $\tilde{w}_{t,t+N-1} = 0$, for every admissible $\kappa_{t,t+N-1}$

$$\begin{aligned}&J(\bar{x},\kappa_{t,t+N-1},0,N)\\ &= \sum_{k=t}^{t+N-1}\{l(x(k),u(k))\} + V_f(x(t+N)) > 0, \forall x \in X^{MPC}(N)/\{0\}\end{aligned}$$

so that

$$\begin{aligned}V(x,N) \ :&= J(\bar{x},\kappa^o_{t,t+N-1},w^o_{t,t+N-1},N) \geq \min_{\kappa_{t,t+N-1}} J(\bar{x},\kappa_{t,t+N-1},0,N)\\ &> l(x,\kappa^{MPC}(x)) > \alpha_l(|x|), \quad \forall x \in X^{MPC}(N)\end{aligned} \tag{21}$$

In view of the Lipschitz assumption on $\kappa_{t,t+N-1}$ and Assumption 7.1, one can show that there exists a $\mathcal{K}$ function $\alpha_2(|x|)$ such that (17) is fulfilled for any $x \in X^{MPC}(N)$. Suppose now that $\kappa^o_{t,t+N-1}$ is the solution of the $FHCDG$ with horizon N and consider the following policy vector for the $FHCDG$ with horizon $N+1$

$$\tilde{\kappa}_{t,t+N} = \begin{cases} \kappa^o_{t,t+N-1} & t \leq k \leq t+N-1 \\ \kappa_f(x(t+N)) & k = t+N \end{cases}$$

Correspondingly

$$\begin{aligned}&J(\bar{x},\tilde{\kappa}_{t,t+N},w_{t,t+N},N+1)\\ &= V_f(x(t+N+1)) - V_f(x(t+N))\\ &\quad + l(x(t+N),u(t+N)) - l_w(w(t+N))\\ &\quad + \sum_{k=t}^{t+N-1}\{l(x(k),u(k)) - l_w(w(k))\} + V_f(x(t+N))\end{aligned}$$

so that in view of Assumption 7.2

$$\begin{aligned}&J(\bar{x},\tilde{\kappa}_{t,t+N},w_{t,t+N},N+1)\\ &\leq \sum_{k=t}^{t+N-1}\{l(x(k),u(k)) - l_w(w(k))\} + V_f(x(t+N))\end{aligned}$$

which implies

$$\begin{aligned} V(x,N+1) &\leq \max_{w\in\mathcal{M}_\mathcal{W}} J(\bar{x},\tilde{\kappa}_{t,t+N-1},w_{t,t+N-1},N+1) \\ &\leq \max_{w\in\mathcal{M}_\mathcal{W}} \sum_{k=t}^{t+N-1} \{l(x(k),u(k))-l_w(w(k))\}+V_f(x(t+N)) \\ &= V(x,N) \end{aligned} \tag{22}$$

which holds $\forall x \in X^{MPC}(N), \forall w \in \mathcal{M}_\mathcal{W}$. Moreover

$$\begin{aligned} V(x,N) = {} & V(\tilde{f}(x,\kappa^{MPC}(x),w),N-1) \\ & +l(x,\kappa^{MPC}(x))-l_w(w) \\ \geq {} & V(\tilde{f}(x,\kappa^{MPC}(x),w),N)+l(x,\kappa^{MPC}(x))-l_w(w) \end{aligned}$$

$\forall x \in X^{MPC}(N), \forall w \in \mathcal{M}_\mathcal{W}$ and

$$V(\tilde{f}(x,\kappa^{MPC}(x),w),N)-V(x,N) \leq -l(x,\kappa^{MPC}(x))+l_w(w)$$

and the ISS is proven. Note also that in view of (22)

$$V(x,N) \leq V(x,N-1) \leq V(x,0)=V_f(x) \leq \beta_{V_f}(|x|), \quad \forall x \in X_f \tag{23}$$

so that if $X_f = X^{MPC}(N)$ the Lipschitz assumption on $\kappa_{t,t+N-1}$ can be relaxed.

Finally, in view of Assumption 7.1 with $\gamma(\cdot)$ such that $\beta_w\left(\gamma\left(\left|x,\kappa^{MPC}(x)\right|\right)\right)-\alpha_l(|x|) < -\delta(|x|)$

$$\begin{aligned} V(\tilde{f}(x,\kappa^{MPC}(x),w),N)-V(x,N) &\leq -\alpha_l(|x|)+\beta_w\left(\gamma\left(\left|x,\kappa^{MPC}(x)\right|\right)\right) \\ &\leq -\delta(|x|), \forall x \in X^{MPC}(N), \forall w \in \mathcal{M}_\mathcal{W} \end{aligned}$$

and robust asymptotic stability is derived.

Remark 3. The major drawback of the closed-loop min-max approach is due to the need to perform optimization over an infinite dimensional space. However two comments are in order. First, one can resort to a finite dimensional parametrization of the control policies, see e.g. [35], [29], [8]. In this case, it is necessary that also the auxiliary control law shares the same structural properties. Second, similar results can be achieved using different prediction (N_p) and control (N_c) horizons, with $N_c \ll N_p$, see [29]. In this case, optimization has to be performed only with respect to N_c policies, while from the end of the control horizon onwards the auxiliary control law can be applied.

Remark 4. By means of the same kind of reasoning followed in the proof of Theorem 1 to derive an upper bound of V in $X^{MPC}(N)$, one can relax the hypothesis on Lipschitz continuity of $\kappa_{t,t+N-1}$ [27]. However, since in practice this sequence of control laws must be parametrized a priori, the continuity assumption in this case can be explicitly verified.

Remark 5. The computation of the auxiliary control law, of the terminal penalty and of the terminal inequality constraint satisfying Assumption 3.2, is not trivial at all. In this regard, a solution has been proposed for affine system in [29], where it is shown how to compute a non linear auxiliary control law based on the solution of a suitable H_∞ problem for the linearized system under control.

Acknowledgement. The authors thank a reviewer for many suggestions which helped to improve the paper. The authors acknowledge the financial support of the MIUR projects *Advanced Methodologies for Control of Hybrid Systems* and *Identification and Adaptive Control of industrial systems.*

References

[1] V. S. Chellaboina and W. M. Haddad. Stability margins of discrete-time nonlinear-nonquadratic optimal regulators. In *IEEE CDC*, pages 1786–1791, 1998.

[2] H. Chen and F. Allgöwer. A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34:1205–1217, 1998.

[3] H. Chen, C. W. Scherer, and F. Allgöwer. A game theoretical approach to nonlinear robust receding horizon control of constrained systems. In *American Control Conference '97*, 1997.

[4] L. Chisci, J. A. Rossiter, and G. Zappa. Systems with persistent disturbances: Predictive control with restricted constraints. *Automatica*, 37:1019–1028, 2001.

[5] G. De Nicolao, L. Magni, and R. Scattolini. On the robustness of receding-horizon control with terminal constraints. *IEEE Trans. Automatic Control*, 41:451–453, 1996.

[6] G. De Nicolao, L. Magni, and R. Scattolini. Stabilizing receding-horizon control of nonlinear time-varying systems. *IEEE Trans. on Automatic Control*, AC-43:1030–1036, 1998.

[7] F. A. C. C. Fontes. A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42:127–143, 2001.

[8] F. A. C. C. Fontes and L. Magni. Min-max model predictive control of nonlinear systems using discontinuous feedbacks. *IEEE Trans. on Automatic Control*, 48:1750–1755, 2003.

[9] F. A. C. C. Fontes, L. Magni, and E. Gyurkovics. Sampled-data model predictive control for nonlinear time-varying systems: Stability and robustness. In *International Workshop on Assessment and Future Directions of Nonlinear Model Predictive Control, Freudenstadt-Lauterbad, Germany August 26-30.*, 2005.

[10] J. C. Geromel and J. J. Da Cruz. On the robustness of optimal regulators for nonlinear discrete-time systems. *IEEE Trans. on Automatic Control*, AC-32:703–710, 1987.

[11] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: the theory and application of maximal output admissible sets. *IEEE Transaction on Automatic Control*, AC-36:1008–1020, 1991.

[12] S. T. Glad. Robustness of nonlinear state feedback- a survey. *Automatica*, 23:425–435, 1987.

[13] G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel. Examples when nonlinear model predictive control is nonrobust. *Automatica*, 40:1729–1738, 2004.

[14] G. Grimm, M. J. Messina, S. E. Tuna, and A. R. Teel. Model predictive control: For want of a local control Lyapunov function, all is not lost. *IEEE Transactions on Automatic Control*, 50:546–558, 2005.

[15] E. Gyurkovics. Receding horizon H_∞ control for nonlinear discrete-time systems. *IEE Proceedings - Control Theory and Applications*, 149:540–546, 2002.

[16] E. Gyurkovics and A. M. Elaiw. Stabilization of sampled-data nonlinear systems by receding horizon contyrol via discrete-time approximations. *Automatica*, 40:2017–2028, 2004.

[17] E. Gyurkovics and T. Takacs. Quadratic stabilisation with h_∞-norm bound of non-linear discrete-time uncertain systems with bounded control. *Systems & Control Letters*, 50:277–289, 2003.

[18] B. Hu and A. Linnemann. Toward infinite-horizon optimality in nonlinear model predictive control. *IEEE Trans. on Automatic Control*, 47:679–682, 2002.

[19] A. Jadbabaie and J. Hauser. Unconstrained receding-horizon control of nonlinear systems. *IEEE Trans. on Automatic Control*, 5:776–783, 2001.

[20] A. Jadbabaie and J. Hauser. On the stability of receding horizon control with a general terminal cost. *IEEE Transactions on Automatic Control*, 50:674–678, 2005.

[21] Z.-P. Jiang and Y. Wang. Input-to-state stability for discrete-time nonlinear systems. *Automatica*, 37:857–869, 2001.

[22] H. K. Khalil. *Nonlinear systems*. Prentice Hall, 1996.

[23] M. Lazar, W. Heemels, A. Bemporad, and S. Weiland. On the stability and robustness of non-smooth nonlinear model predictive control. In *Int. Workshop on Assessment and Future Directions of NMPC, Freudenstadt-Lauterbad, Germany*, pages 327–334, 2005. Also submitted for Springer Book.

[24] D. Limón, T. Alamo, and E. F. Camacho. Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties. In *IEEE CDC*, pages 4619–4624, 2002.

[25] D. Limón, T. Alamo, and E. F. Camacho. Stability analysis of systems with bounded additive uncertainties based on invariant sets: Stability and feasibility of MPC. In *ACC02*, pages 364–369, 2002.

[26] D. Limon, T. Alamo, and E. F. Camacho. Stable constrained MPC without terminal constraint. In *American Control Conference*, pages 4893 – 4898, 2003.

[27] D. Limón, T. Alamo, and E. F. Camacho. Robust stability of min-max MPC controllers for nonlinear systems with bounded uncertainties. In *MTNS*, 2004.

[28] L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini. A stabilizing model-based predictive control for nonlinear systems. *Automatica*, 37:1351–1362, 2001.

[29] L. Magni, G. De Nicolao, R. Scattolini, and F.Allgöwer. Robust model predictive control of nonlinear discrete-time systems. *International Journal of Robust and Nonlinear Control*, 13:229–246, 2003.

[30] L. Magni, H. Nijmeijer, and A. J. van der Schaft. A receding-horizon approach to the nonlinear H_∞ control problem. *Automatica*, 37:429–435, 2001.

[31] L. Magni, D. M. Raimondo, and R. Scattolini. Regional input-to-state stability for nonlinear model predictive control. *IEEE Transactions on Automatic Control,* AC51, pp. 1548–1553, 2006.

[32] L. Magni and R. Scattolini. Model predictive control of continuous-time nonlinear systems with piecewise constant control. *IEEE Trans. on Automatic Control*, 49:900–906, 2004.

[33] L. Magni and R. Scattolini. Control design for nonlinear systems: Trading roubustness and performance with the model predictive control approach. *IEE Proceedings - Control Theory & Application*, pages 333–339, 2005.

[34] L. Magni and R. Sepulchre. Stability margins of nonlinear receding horizon control via inverse optimality. *Systems & Control Letters*, 32:241–245, 1997.

[35] D. Q. Mayne. Nonlinear model predictive control: Challenges and opportunities. In F. Allgoẅer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 23–44. Progress in Systems and Control Theory, Birkhauser Verlag, 2000.

[36] D. Q. Mayne. Control of constrained dynamic systems. *European Journal of Control*, 7:87–99, 2001.

[37] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[38] E. S. Meadows, M. A. Henson, J. W. Eaton, and J. B. Rawlings. Receding horizon control and discontinuous state feedback stabilization. *Int. J. Control*, 62:1217–1229, 1995.

[39] H. Michalska and D. Q. Mayne. Robust receding horizon control of constrained nonlinear systems. *IEEE Trans. on Automatic Control*, 38:1623–1633, 1993.

[40] T. Parisini and R. Zoppoli. A receding-horizon regulator for nonlinear systems and a neural approximation. *Automatica*, 31:1443–1451, 1995.

[41] P. O. M. Scokaert and D. Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Trans. on Automatic Control*, 43:1136–1142, 1998.

[42] P. O. M. Scokaert, J. B. Rawlings, and E. S. Meadows. Discrete-time stability with perturbations: Application to model predictive control. *Automatica*, 33:463–470, 1997.

[43] R. Sepulchre, M. Jankovic, and P. V. Kokotovic. *Constructive Nonlinear Control.* Springer-Verlag, 1996.

[44] G. Tadmor. Receding horizon revisited: An easy way to robustly stabilize an LTV system. *Systems & Control Letters*, 18:285–294, 1992.

MPC for Stochastic Systems

Mark Cannon, Paul Couchman, and Basil Kouvaritakis

Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK
mark.cannon@eng.ox.ac.uk

Summary. Stochastic uncertainty is present in many control engineering problems, and is also present in a wider class of applications, such as finance and sustainable development. We propose a receding horizon strategy for systems with multiplicative stochastic uncertainty in the dynamic map between plant inputs and outputs. The cost and constraints are defined using probabilistic bounds. Terminal constraints are defined in a probabilistic framework, and guarantees of closed-loop convergence and recursive feasibility of the online optimization problem are obtained. The proposed strategy is compared with alternative problem formulations in simulation examples.

1 Introduction

The success of a Model Predictive Control (MPC) strategy depends critically on the choice of model. In most applications the plant model necessarily involves uncertainty, either endemic (e.g. due to exogenous disturbances) or introduced into the model to account for imprecisely known dynamics. It is usual in robust MPC to assume that uncertainty is bounded, or equivalently that it is random and uniformly distributed, and to adopt a worst case approach (e.g. [1, 2]). This is often considered to be overly pessimistic, even though it can be made less conservative through the use of closed-loop optimization [3, 4], albeit at considerable computational cost.

A more realistic approach, especially when uncertainty is known to be random but is not uniform, is to identify the distributions of uncertain model parameters and use these to solve a stochastic MPC problem. In many applications distributions for uncertain parameters can be quantified (e.g. as part of the model identification process), and some of the constraints are soft and probabilistic in nature (e.g. in sustainable development applications). Ignoring this information (by employing worst case performance indices and invoking constraints over all possible realizations of uncertainty) results in conservative MPC laws. This motivates the development of stochastic MPC formulations, which have been proposed for the case of additive disturbances (e.g. [5, 6, 7]) and for models incorporating multiplicative disturbances (e.g. [8, 9]).

Information on the distributions of stochastic parameters can be exploited in an optimal control problem by defining the performance index as the expected

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 255–268, 2007.
springerlink.com

value of the usual quadratic cost. This approach is the basis of unconstrained LQG optimal control, and has more recently been proposed for receding horizon control [7, 10]. Both [10] and [7] consider input constraints, with [10] performing an open-loop optimization while [7] uses Monte Carlo simulation techniques to optimize over feedback control policies. This paper also considers constraints, but an alternative cost is developed based on bounds on predictions that are invoked with specified probability. The approach allows for a greater degree of control over the output variance, which is desirable for example in sustainable development, where parameter variations are large and the objective is to maximize the probability that the benefit associated with a decision policy exceeds a given aspiration level.

Probabilistic formulations of system constraints are also common in practice. For example an output may occasionally exceed a given threshold provided the probability of violation is within acceptable levels; this is the case for economic constraints in process control and fatigue constraints in electro-mechanical systems. Probabilistic constraints are incorporated in [11] through the use of statistical confidence ellipsoids, and also in [9], which reduces conservatism by applying linear probabilistic constraints directly to predictions without the need for ellipsoidal relaxations. The approach of [9] assumes Moving Average (MA) models with random coefficients, and achieves the guarantee of closed-loop stability through the use of an equality stability terminal constraints. The method is extended in [12] to more general linear models in which the uncertain parameters are contained in the output map of a state-space model, and to incorporate less restrictive inequality stability constraints.

The current paper considers the case of uncertain time-varying plant parameters represented as Gaussian random variables. This type of uncertainty is encountered for example in civil engineering applications (e.g. wind-turbine blade pitch control) and in financial engineering applications, where Gaussian disturbance models are common. Earlier work is extended in order to account for uncertainty in state predictions, considering in particular the definition of cost and terminal constraints to ensure closed-loop convergence and feasibility properties. For simplicity the model uncertainty is assumed to be restricted to Gaussian parameters in the input map, since this allows the distributions of predictions to be obtained in closed-form, however the design of cost and constraints extends to more general model uncertainty. After discussing the model formulation in section 2 and the stage cost in section 3, sections 4 and 5 propose a probabilistic form of invariance for the definition of terminal sets and define a suitable terminal penalty term for the MPC cost. Section 6 describes closed-loop convergence and feasibility properties, and the advantages over existing robust and stochastic MPC formulations are illustrated in section 7.

2 Multiplicative Uncertainty Class

In many control applications, stochastic systems with uncertain multiplicative parameters can be represented by MA models:

$$y_i(k) = \sum_{m=1}^{n_u} g_{im}^T(k)\tilde{u}_m(k-1), \quad \tilde{u}_m(k-1) = [u_m(k-n)\dots u_m(k-1)]^T \tag{1}$$

where $u_m(k)$, $m = 1,\dots,n_u$, $y_i(k)$, $i = 1,\dots,n_y$, are input and output variables respectively, and the plant parameters $g_{im}(k)$ are Gaussian random variables. For convenience we consider the case of two outputs ($n_y = 2$): y_1 is taken to be *primary* (in that a probabilistic measure of performance on it is to be optimized) whereas y_2 is subject to probabilistic performance constraints and is referred to as *secondary.*

As a result of the linear dependence of the model (1) on uncertain plant parameters, the prediction of $y_i(k+j)$ made at time k (denoted $y_i(k+j|k)$) is normally distributed. Therefore bounds on $y_i(k+j|k)$ that are satisfied with a specified probability p can be formulated as convex (second-order conic) constraints on the predicted future input sequence. Bounds of this kind are used in [9] to derive a probabilistic objective function and constraints for MPC. These are combined with a terminal constraint that forces predictions to reach a pre-computed steady-state at the end of an N-step prediction horizon to define a stable receding horizon control law. Subsequent work has applied this methodology to a sustainable development problem using linear time-varying MA models [13].

Though often convenient in practice, MA models are non-parsimonious, and an alternative considered in [12] is given by the state space model:

$$x(k+1) = Ax(k) + Bu(k), \qquad y_i(k) = c_i^T(k)\,x(k), \quad i = 1,2 \tag{2}$$

where $x(k) \in \mathbb{R}^n$ is the state (assumed to be measured at time k), $u(k) \in \mathbb{R}^{n_u}$ is the input, and A, B are known constant matrices. The output maps $c_i(k) \in \mathbb{R}^n$, $i = 1,2$ are assumed to be normally distributed: $c_i(k) \sim \mathcal{N}(\bar{c_i}, \Theta_{c,i})$, with $\{c_i(k), c_i(j)\}$ independent for $k \neq j$. The stability constraints of [9] are relaxed in [12], which employs less restrictive inequality constraints on the N step-ahead predicted state.

This paper considers a generalization of the model class in order to handle the case that the future plant state is a random variable. For simplicity we restrict attention to the case of uncertainty in the input map:

$$x(k+1) = Ax(k) + B(k)u(k), \;\; B(k) = \bar{B} + \sum_{r=1}^{L} q_r(k)B_r, \;\; y_i(k) = c_i^T x(k), \; i = 1,2 \tag{3}$$

where A, $\bar{B}$, B_i, c_i are known and $q(k) = [q_1(k) \;\; \cdots \;\; q_L(k)]^T$ are Gaussian parameters. We assume that $q(k) \sim \mathcal{N}(0, I)$ since it is always possible to define the model realization (A, B, C) so that the elements of $q(k)$ are uncorrelated, and that $\{q(k), q(j)\}$ are independent for $k \neq j$. Correlation between model parameters at different times could be handled by the paper's approach, but the latter assumption simplifies the expressions for the predicted covariances in section 5 below. The state $x(k)$ is assumed to be measured at time k. The paper focuses on the design of the MPC cost and terminal constraints so as to ensure closed-loop stability (for the case of soft constraints) and recursive feasibility with a pre-specified confidence level.

3 Performance Index and Constraints

The control objective is to regulate the expected value and variance of the primary output while respecting constraints on inputs and secondary outputs. We define the receding horizon cost function to be minimized online at time k as

$$J = \sum_{j=0}^{N-1} l(k+j|k) + L(k+N|k) \tag{4}$$

where

$$l(k+j|k) = \bar{y}_1^2(k+j|k) + \kappa_1^2 \sigma_1^2(k+j|k) \tag{5}$$

with $\bar{y}_1(k+j|k) = \mathbb{E}_k y_1(k+j|k)$ and $\sigma_1^2(k+j|k) = \mathbb{E}_k\big[y_1(k+j|k) - \bar{y}_1(k+j|k)\big]^2$ denoting the mean and variance of $y_1(k+j|k)$ given the measurement $x(k)$ (we denote the expectation of a variable z given the measurement $x(k)$ as $\mathbb{E}_k z$).

This form of stage cost is used in preference to the more usual expectation MPC cost (e.g. [7, 10, 11]) because it enables the relative weighting of mean and variance to be controlled directly via the parameter κ_1, which can be interpreted in terms of probabilistic bounds on the prediction $y_1(k+j|k)$. To see this, let t_{lower} and t_{upper} be lower and upper bounds on $y_1(k+j|k)$ with a given probability p_1:

$$\begin{aligned} \Pr\big(y_1(k+j|k) \geq t_{\text{lower}}(k+j|k)\big) &\geq p_1 \\ \Pr\big(y_1(k+j|k) \leq t_{\text{upper}}(k+j|k)\big) &\geq p_1 \end{aligned} \tag{6}$$

then, since the predictions generated by (3) are normally distributed, it is easy to show that the stage cost (5) is equivalent to

$$l(k+j|k) = \tfrac{1}{2} t_{\text{lower}}^2(k+j|k) + \tfrac{1}{2} t_{\text{upper}}^2(k+j|k)$$

provided κ_1 satisfies $\mathfrak{N}(\kappa_1) = p_1$, where $\mathfrak{N}$ is the normal distribution function: $\Pr(z \leq Z) = \mathfrak{N}(Z)$ for $z \sim \mathcal{N}(0,1)$.

An important property of the stage cost is that it allows closed-loop stability under the MPC law to be determined by considering the optimal value of J as a stochastic Lyapunov function. The analysis (which is summarized in Section 6 below) is based on the following result.

Lemma 1. *If $\kappa_1 \geq 1$, then for any given input sequence $\{u(k), u(k+1), \ldots, u(k+j-1)\}$, the expectation of $l(k+j|k+1)$ conditional on time k satisfies:*

$$\mathbb{E}_k l(k+j|k+1) \leq l(k+j|k). \tag{7}$$

Proof. Re-writing (5) as $l(k+j|k) = \mathbb{E}_k y_1^2(k+j|k) + (\kappa_1^2 - 1)\sigma_1^2(k+j|k)$, and noting that

$$\begin{aligned} \mathbb{E}_k\big(\mathbb{E}_{k+1} y_1^2(k+j|k+1)\big) &= \mathbb{E}_k y_1^2(k+j|k) \\ \mathbb{E}_k \sigma_1^2(k+j|k+1) &= \sigma_1^2(k+j|k+1) \end{aligned}$$

we have

$$\mathbb{E}_k l(k+j|k+1) = l(k+j|k) - (\kappa_1^2 - 1)\big(\sigma_1^2(k+j|k) - \sigma_1^2(k+j|k+1)\big).$$

The required bound therefore holds if $\kappa_1 \geq 1$ since $\sigma_1^2(k+j|k) \geq \sigma_1^2(k+j|$ $k+1)$. ■

Remark 1. In accordance with Lemma 1 it is assumed below that $\kappa_1 \geq 1$, or equivalently that the bounds (6) are invoked with probability $p_1 \geq 84.1\%$ (to 3 s.f.). With $\kappa_1 = 1$, this formulation recovers the conventional expectation cost: $l(k+j|k) = \mathbb{E}_k y_1^2(k+j|k)$ for regulation problems.

Consider next the definition of constraints. Since output predictions are Gaussian random variables, we consider probabilistic (as opposed to hard) constraints:

$$\Pr\big(y_2(k+j|k) \leq Y_2\big) \geq p_2 \tag{8}$$

where Y_2 is a constraint threshold. Input constraints are assumed to have the form:

$$|u(k+j|k)| \leq U \tag{9}$$

where $u(k+j|k)$ is the predicted value of $u(k+j)$ at time k.

4 Terminal Constraint Set

Following the conventional dual mode prediction paradigm [14], predicted input trajectories are switched to a linear terminal control law: $u(k+j|k) = Kx(k+j|k)$, $j \geq N$ after an initial N-step prediction horizon. For the case of uncertainty in the output map (2), an ellipsoidal terminal constraint can be computed by formulating conditions for invariance and satisfaction of constraints (8),(9) under the terminal control law as LMIs [12]. However, in the case of the model (3), the uncertainty in the predicted state trajectory requires that a probabilistic invariance property is used in place of the usual deterministic definition of invariance when defining a terminal constraint set. We therefore impose the terminal constraint that $x(k+N|k)$ lie in a terminal set Ω with a given probability, where Ω is designed so that the probability of remaining within Ω under the closed-loop dynamics $x(k+1) = \Phi(k)x(k)$ is at least p_Ω, i.e.

$$\Pr(\Phi x \in \Omega) \geq p_\Omega \quad \forall x \in \Omega. \tag{10}$$

If constraints on the input and secondary output are satisfied everywhere within Ω, then this approach can be used to define a receding horizon optimization which is feasible with a specified probability at time $k+1$ if it is feasible at time k. Given that the uncertain parameters of (3) are not assumed bounded, this is arguably the strongest form of recursive feasibility attainable.

For computational convenience we consider polytopic terminal sets defined by $\Omega = \{x : v_i^T x \leq 1,\ i = 1, \ldots, m\}$. Denote the closed-loop dynamics of (3) under $u = Kx$ as

$$x(k+1) = \Phi(k)x(k), \quad \Phi(k) = \bar{\Phi} + \sum_{i=1}^{L} q_i(k)\Phi_i, \quad q(k) \sim \mathcal{N}(0, I) \tag{11}$$

(where $\bar{\Phi} = A + \bar{B}K$ and $\Phi_i = B_i K$), then confidence ellipsoids for q can be used to determine conditions on the vertices x_j, $j = 1, \ldots, M$ of Ω so that Ω is invariant with a given probability. Specifically, the condition $v_i^T \Phi x_j \leq 1$ is equivalent to $x_j^T [\Phi_1^T v_i \ \cdots \ \Phi_L^T v_i] q \leq 1 - x_j^T \bar{\Phi}^T v_i$, and, since $\|q\|^2$ is distributed as χ^2 with L degrees of freedom, it follows that $v_i^T \Phi x_j \leq 1$ with probability p_Ω if

$$r_\Omega \left\| x_j^T \left[\Phi_1^T v_i \cdots \Phi_L^T v_i\right] \right\|_2 \leq 1 - x_j^T \bar{\Phi} v_i \tag{12}$$

where r_Ω satisfies $\Pr(\chi^2(L) < r_\Omega^2) = p_\Omega$.

Lemma 2. *Ω is invariant under (11) with probability p_Ω, i.e.*

$$\Pr(v_i^T \Phi x \leq 1, \ i = 1, \ldots, m) \geq p_\Omega, \quad \forall x \in \Omega \tag{13}$$

if (12) is satisfied for $i = 1, \ldots, m$ and $j = 1, \ldots, M$.

Proof. If (12) holds for given j and $i = 1, \ldots, m$, then x_j necessarily satisfies $\Pr(\Phi x_j \in \Omega) \geq p_\Omega$ (since $\|q\|_2 \leq r_\Omega$ with probability p_Ω). Furthermore, invoking this condition for each vertex x_j implies (13), since (12) is convex in x_j. ■

The problem of maximizing Ω subject to (10) and the conditions that input constraints (9) and the secondary output constraint $y_2 \leq Y_2$ are met everywhere within the terminal set can be summarized as:

$$\begin{aligned} \text{maximize} \ \ & \text{vol}(\Omega) \\ \text{subject to} \ \ & r_\Omega \left\| x_j^T \left[\Phi_1^T v_i \cdots \Phi_L^T v_i\right] \right\|_2 \leq 1 - x_j^T \bar{\Phi} v_i, \\ & |K x_j| \leq U \\ & c_2 x_j \leq Y_2 \end{aligned} \tag{14}$$

in variables $\{v_i, i = 1, \ldots, m\}$ and $\{x_j, j = 1, \ldots, M\}$. This is a nonconvex problem, but for fixed $\{v_i\}$ the constraints are convex in $\{x_j\}$, enabling a sequence of one-step sets of increasing volume to be computed via convex programming. Therefore a (locally) optimal point for (14) can be found using a sequential approach similar to that of [15]. Furthermore, if Ω is defined as a symmetric low-complexity polytope (i.e. $\Omega = \{x : \|Wx\|_\infty \leq 1\}$, for full-rank $W \in \mathbb{R}^{n \times n}$), then the linear feedback gain K can be optimized simultaneously with Ω by including the vertex controls, u_j, $j = 1, \ldots, n$ as additional optimization variables in (14), where $u_j = K x_j$.

5 Terminal Penalty

To allow a guarantee of closed-loop stability, we define the terminal penalty in (4) as the cost-to-go over all $j \geq N$ under the terminal control law $u(k+j|k) = Kx(k+j|k)$. This section derives the required function $L(k+N|k)$ as a quadratic form based on the solution of a pair of Lyapunov equations, and shows that the Lyapunov-like property:

$$\mathbb{E}_k\big[L(k+N+1|k+1)+l(k+N|k+1)\big] \leq L(k+N|k) \tag{15}$$

holds whenever predictions at time $k+1$ are generated by the sequence

$$\mathbf{u}(k+1)=\{u(k+1|k),\dots,u(k+N-1|k),Kx(k+N|k+1)\} \tag{16}$$

where $\mathbf{u}(k)=\{u(k|k),u(k+1|k),\dots,u(k+N-1|k)\}$ is the predicted input sequence at time k and $u(k)=u(k|k)$.

To simplify notation, let $x_\delta = x-\bar{x}$, where $\bar{x}(k+j|k)=\mathbb{E}_k x(k+j|k)$, and define

$$Z_1(k+j|k)=\mathbb{E}_k\big[x(k+j|k)x^T(k+j|k)\big],$$
$$Z_2(k+j|k)=\mathbb{E}_k\big[x_\delta(k+j|k)x_\delta^T(k+j|k)\big].$$

Lemma 3. *If the terminal penalty in (4) is defined by*

$$L(k+N|k)=\mathrm{Tr}\big(Z_1(k+N|k)S_1\big)+(\kappa_1^2-1)\mathrm{Tr}\big(Z_2(k+N|k)S_2\big) \tag{17}$$

where $S_1=S_1^T\succ 0$ and $S_2=S_2^T\succ 0$ are the solutions of the Lyapunov equations

$$\bar{\Phi}^T S_2\bar{\Phi}+c_1c_1^T=S_2 \tag{18a}$$

$$\bar{\Phi}^T S_1\bar{\Phi}+\sum_{i=1}^{L}\Phi_i^T\big(S_1+(\kappa_1^2-1)S_2\big)\Phi_i+c_1c_1^T=S_1 \tag{18b}$$

then $L(k+N|k)$ is the cost-to-go: $L(k+N|k)=\sum_{j=N}^{\infty}l(k+j|k)$ for the closed-loop system formed by (3) under the terminal control law $u(k+j|k)=Kx(k+j|k)$.

Proof. With $u(k+j|k)=Kx(k+j|k)$, it is easy to show that, for all $j\geq N$:

$$Z_1(k+j+1|k)=\bar{\Phi}Z_1(k+j|k)\bar{\Phi}^T+\sum_{i=1}^{L}\Phi_iZ_1(k+j|k)\Phi_i^T \tag{19a}$$

$$Z_2(k+j+1|k)=\bar{\Phi}Z_2(k+j|k)\bar{\Phi}^T+\sum_{i=1}^{L}\Phi_iZ_2(k+j|k)\Phi_i^T. \tag{19b}$$

Using these expressions and (18a,b) to evaluate $L(k+j+1|k)$, we obtain

$$L(k+j+1|k)+l(k+j|k)=L(k+j|k), \tag{20}$$

which can be summed over all $j\geq N$ to give

$$L(k+N|k)-\lim_{j\to\infty}L(k+j|k)=\sum_{j=N}^{\infty}l(k+j|k),$$

but $x(k+1)=\Phi x(k)$ is necessarily mean-square stable [16] in order that there exist positive definite solutions to (18a,b), and it follows that $\lim_{j\to\infty}L(k+j|k)=0$. ∎

Remark 2. For any $j \geq 1$, Z_1 and Z_2 can be computed using

$$Z_2(k+j|k) = \sum_{i=0}^{j-1} \Psi_i \Psi_i^T, \quad \Psi_i = A^{j-1-i}\big[B_1 u(k+i|k) \cdots B_L u(k+i|k)\big] \tag{21a}$$

$$Z_1(k+j|k) = \bar{x}(k+j|k)\bar{x}^T(k+j|k) + Z_2(k+j|k) \tag{21b}$$

Therefore $L(k+N|k)$ is a quadratic function of the predicted input sequence:

$$L(k+N|k) = \mathbf{u}^T(k)H\mathbf{u}(k) + 2g^T\mathbf{u}(k) + \gamma$$

where $\mathbf{u}(k) = [u^T(k|k) \cdots u^T(k+N-1|k)]^T$ and H, g are constants.

Theorem 1. *If $L(k+N|k)$ is given by (17) and $L(k+N+1|k+1), l(k+N+1|k+1)$ correspond to the predictions generated by the input sequence (16), then (15) is satisfied if $\kappa_1 \geq 1$.*

Proof. From (19), (18), and $u(k+N|k+1) = Kx(k+N|k+1)$ it follows that $L(k+N+1|k+1) + l(k+N|k+1) = L(k+N|k+1)$. Furthermore, from (21a,b) we have

$$\mathbb{E}_k Z_2(k+N|k+1) = Z_2(k+N|k+1), \quad \mathbb{E}_k Z_1(k+N|k+1) = Z_1(k+N|k).$$

Combining these results, the LHS of (15) can be written

$$\mathbb{E}_k\big[L(k+N+1|k+1) + l(k+N|k+1)\big] = \mathrm{Tr}\big(Z_1(k+N|k)S_1\big) + (\kappa_1^2 - 1)\mathrm{Tr}\big(Z_2(k+N|k+1)S_2\big)$$

and therefore (15) holds if $\kappa_1 \geq 1$ since (21a) implies $Z_2(k+N|k+1) \preceq Z_2(k+N|k)$. ■

6 MPC Strategy and Closed-Loop Properties

The stage cost, terminal cost and terminal constraints are combined in this section to construct a receding horizon strategy based on the online optimization:

$$\underset{\mathbf{u}(k)}{\text{minimize}} \;\; J_k = \sum_{j=0}^{N-1} l(k+j|k) + L(k+N|k) \tag{22a}$$

subject to the following constraints, invoked for $j = 1, \ldots, N-1$:

$$|u(k+j|k)| \leq U \tag{22b}$$

$$\Pr\big(y_2(k+j|k) \leq Y_2\big) \geq p_2 \tag{22c}$$

$$\Pr\big(x(k+N|k) \in \Omega\big) \geq p_2 \tag{22d}$$

The MPC law is defined as $u(k) = u^*(k|k)$, where $\mathbf{u}^*(k) = \{u^*(k|k), \ldots, u^*(k+N-1|k)\}$ is optimal at time k, computed on the basis of the measured $x(k)$. From the plant model and parameter distributions, the constraints (22c) on y_2 can be written

$$\kappa_2\big(c_2^T Z_2(k+j|k)c_2\big)^{1/2} \le Y_2 - c_2^T \bar{x}(k+j|k) \tag{23}$$

where κ_2 satisfies $\mathfrak{N}(\kappa_2) = p_2$. Similarly, making use of confidence ellipsoids for $q(k)$, the terminal constraint (22d) can be expressed

$$r_1\big(v_i^T Z_2(k+j|k)v_i\big)^{1/2} \le 1 - v_i^T \bar{x}(k+j|k), \quad i = 1,\dots,m \tag{24}$$

where r_1 is defined by $\Pr\big(\chi^2(NL) \le r_1^2\big) = p_2$. It follows that (22) is convex, and has the form of a second-order cone program (SOCP), enabling solution via efficient algorithms [17]. The stability properties of the MPC law can be stated as follows.

Theorem 2. *Assume that (22) is feasible at all times $k = 0,1,\dots$. Then $y_1(k) \to 0$, and $\|x(k)\|_2$ converges to a finite limit with probability 1 if (A, c_1) is observable.*

Proof. From Lemmas 1 and 3, the cost, $\tilde{J}_{k+1}$, for the suboptimal sequence $\tilde{\mathbf{u}}(k+1) = \{u^*(k+1|k),\dots,Kx^*(k+N|k)\}$ at time $k+1$ satisfies $\mathbb{E}_k\tilde{J}_{k+1} \le J_k^* - y_1^2(k)$, where J_k^* is the optimal value of (22a). After optimization at $k+1$ we have

$$\mathbb{E}_k J_{k+1}^* \le \mathbb{E}_k \tilde{J}_{k+1} - y_1^2(k) \le J_k^* - y_1^2(k) \tag{25}$$

It follows that J_k converges to a lower limit and $y_1(k) \to 0$ with probability 1 [18]. Furthermore the definitions of stage cost (5) and terminal penalty (17) imply that

$$J_k = \sum_{j=0}^{\infty} c_1^T \bar{x}(k+j|k)\bar{x}^T(k+j|k)c_1 + \kappa_1^2 c_1^T Z_2(k+j|k)c_1$$

and, since $\sum_{j=0}^{\infty} c_1^T Z_2(k+j|k)c_1$ is positive definite in $\mathbf{u}(k)$ if (A, c_1) is observable, it follows that J_k is positive definite in $x(k)$ if (A, c_1) is observable. Under this condition therefore, $\|x(k)\|_2$ converges to a finite limit with probability 1. ■

Note that the derivation of (25) assumes a pre-stabilized prediction model; the same convergence property can otherwise be ensured by using a variable horizon N.

The constraints (22b-d) apply only to predicted trajectories at time k, and do not ensure feasibility of (22) at future times. For example, at time $k+1$, (23) requires

$$\kappa_2\big(c_2^T Z_2(k+j|k+1)c_2\big)^{1/2} \le Y_2 - c_2^T \bar{x}(k+j|k+1), \quad j = 1,\dots,N$$

where $\bar{x}(k+j|k+1)$ is a Gaussian random variable at time k, with mean $\bar{x}(k+j|k)$ and variance $c_2^T A^{j-1} Z_2(k+1|k) A^{j-1\,T} c_2$. Therefore (23) is feasible at $k+1$ with probability p_2 if

$$\kappa_2\big(c_2^T Z_2(k+j|k+1)c_2\big)^{1/2} + \kappa_2\big(c_2^T A^{j-1} Z_2(k+1|k) A^{j-1\,T} c_2\big)^{1/2} \le Y_2 - c_2^T \bar{x}(k+j|k)$$

holds for $j = 1,\dots,N$ at time k; this condition is necessarily more restrictive than (23) since $Z_2(k+j|k) = Z_2(k+j|k+1) + A^{j-1} Z_2(k+1|k) A^{j-1\,T}$. In order to provide a recursive guarantee of feasibility we therefore include additional constraints in the online optimization, as summarized in the following result.

Theorem 3. *If (23) and (24) are replaced in the MPC online optimization (22) by*

$$\sum_{l=0}^{j-1} \kappa_2 \big(c_2^T A^{j-1-l} Z_2(k+l+1|k+l) A^{j-1-l^T} c_2\big)^{1/2} \leq Y_2 - c_2^T \bar{x}(k+j|k) \quad \text{(26a)}$$

$$\sum_{l=0}^{j-2} \kappa_2 \big(v_i^T A^{j-1-l} Z_2(k+l+1|k+l) A^{j-1-l^T} v_i\big)^{1/2}$$

$$+ r_j \big(v_i^T Z_2(k+N|k+j-1) v_i\big)^{1/2} \leq 1 - v_i^T \bar{x}(k+N|k) \quad \text{(26b)}$$

for $j = 2, \ldots, N$, where r_j is defined by $\Pr(\chi^2((N+1-j)L) \leq r_j) = p_2/p_\Omega^{j-1}$, then feasibility of (22) at time k implies feasibility at time $k+1$ with probability p_2.

Proof. Condition (26a) ensures that: (i) $\Pr(y(k+j|k) \leq Y_2) \geq p_2$ for $j = 1, \ldots, N$; (ii) $\Pr(y(k+j|k+1) \leq Y_2) \geq p_2$, $j = 2, \ldots, N$, is feasible at $k+1$ with probability p_2; and (iii) the implied constraints are likewise feasible with probability p_2 when invoked at $k+1$. Here (iii) is achieved by requiring that the constraints $\Pr(y(k+l|k+j) \leq Y_2) \geq p_2$ be feasible with probability p_2 when invoked at $k+j$, $j = 2, \ldots, N-1$. Condition (26b) ensures recursive feasibility of (22d) with probability p_2 through the constraint that $\Pr(x(k+N|k+j) \in \Omega) \geq p_2/p_\Omega^j$, $j = 0, \ldots, N-1$ (and hence also $\Pr(x(k+N+j|k+j) \in \Omega) \geq p_2$) should be feasible with probability p_2. ■

Incorporating (26a,b) into the receding horizon optimization leads to a convex online optimization, which can be formulated as a SOCP. However (26) and the constraint that Ω should be invariant with probability $p_\Omega > p_2^{1/(N-1)} \geq p_2$ are more restrictive than (22c,d), implying a more cautious control law.

Remark 3. The method of computing terminal constraints and penalty terms described in sections 4 and 5 is unchanged in the case that A contains random (normally distributed) parameters. However in this case state predictions are not linear in the uncertain parameters, so that the online optimization (22) could no longer be formulated as a SOCP. Instead computationally intensive numerical optimization routines (such as the approach of [7]) would be required.

Remark 4. It is possible to extend the approach of sections 4 and 5 to nonlinear dynamics, for example using linear difference inclusion (LDI) models. In the case that uncertainty is restricted to the linear output map, $y_j(k) = C_j(k)x(k)$ predictions then remain normally distributed, so that the online optimization, though nonconvex in the predicted input sequence, would retain some aspects of the computational convenience of (22).

7 Numerical Examples

This section uses two simulation examples to compare the stochastic MPC algorithm developed above with a generic robust MPC algorithm and the stochastic MPC approach of [12].

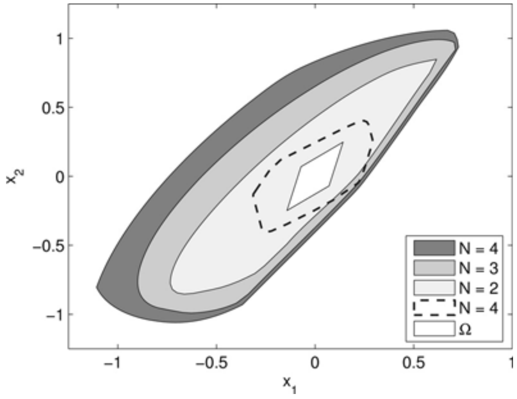


Fig. 1. Feasible initial condition sets for stochastic MPC ($p_2 = 0.85$) with varying N. Dashed line: feasible set of robust MPC based on 85% confidence level for $N = 4$.

First consider the plant model with

$$A = \begin{bmatrix} 1.04 & -0.62 \\ 0.62 & 1.04 \end{bmatrix} \; \bar{B} = \begin{bmatrix} 0 \\ 2 \end{bmatrix} \; B_1 = \begin{bmatrix} -0.12 \\ 0.02 \end{bmatrix} \; B_2 = \begin{bmatrix} 0.04 \\ -0.06 \end{bmatrix} \; c_1^T = \begin{bmatrix} 0 \\ -4.4 \end{bmatrix} \; c_2^T = \begin{bmatrix} 3 \\ 2.3 \end{bmatrix}$$

$U = 1$, $Y_2 = 1$, and $p_1 = p_2 = 0.85$. The offline computation for stochastic MPC involves maximizing a low-complexity polytopic set Ω subject to $\Pr(\Phi x \in \Omega) \geq p_2$ for all $x \in \Omega$; for this example the maximal Ω has an area of 0.055.

An alternative approach to MPC is to determine bounds on plant parameters corresponding to a confidence level of, say, p_2 by setting

$$B(k) = \bar{B} + \sum_{i=1}^{L} q_i(k) B_i, \quad |q(k)| \leq \mathfrak{N}^{-1}(p_2) \tag{27}$$

in (3), and then to implement a robust MPC law based on this approximate plant model. For a confidence level of $p_2 = 0.85$, the maximal low-complexity set Ω', which is robustly invariant for bounded parameter variations (27), is similar in size (area = 0.048) to Ω. The similarity is to be expected since the assumption of bounded uncertainty implies that the probability that $\Phi x \in \Omega'$ under the actual plant dynamics for any $x \in \Omega'$ is p_2.

A robust (min-max) MPC law employing open-loop predictions based on the parameter bounds of (27) is, however, significantly more conservative than the stochastic MPC law of (22) for the same confidence level. This can be seen in Fig. 1, which compares the feasible sets for the two control laws for $p_2 = 0.85$ (the feasible set for robust MPC decreases with increasing N for $N > 4$ since the plant is open-loop unstable). Closed-loop performance is also significantly worse: the closed-loop cost for robust MPC (based on the parameter bounds (27) with $p_2 = 0.85$), averaged over 10 initial conditions and 200 uncertainty realizations, is 63% greater than that for stochastic MPC (with $p_1 = p_2 = 0.85$). Figures 2 and 3 compare the closed-loop responses for a single initial condition and 20

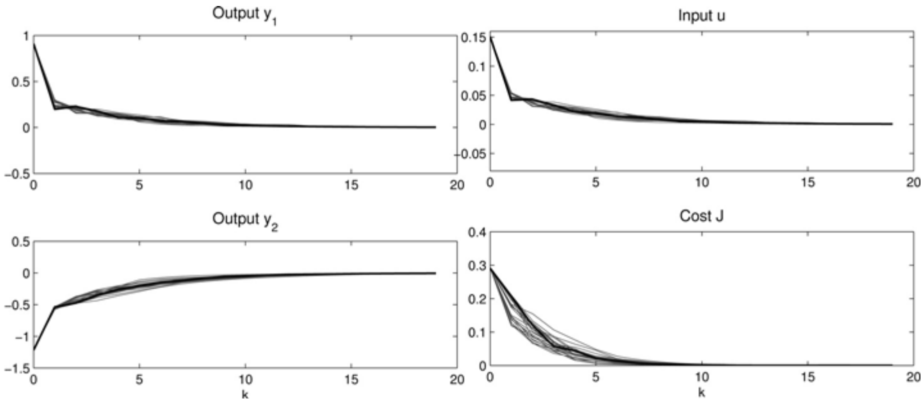


Fig. 2. Stochastic MPC closed-loop responses for $p_2 = 0.85$ and 20 uncertainty realizations (dark lines show responses for a single uncertainty realization)

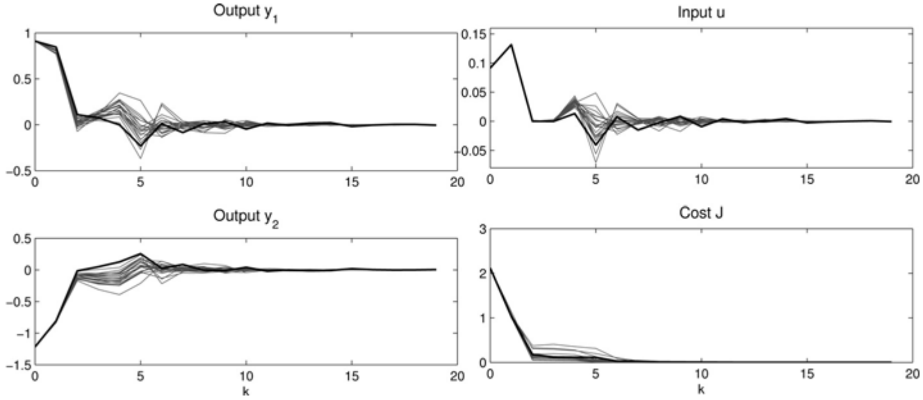


Fig. 3. Robust MPC closed-loop responses for 85% confidence levels and the same set of uncertainty realizations as in Fig. 2

uncertainty realizations. The higher degree of conservativeness and greater variability in Fig. 3 is a result of the robust min-max strategy, which attempts to control worst-case predictions based on the confidence bounds of (27), whereas the stochastic MPC strategy (Fig. 2) has direct control over the statistics of future predictions at each sampling instant.

Consider next the effects of approximating uncertainty in the input map as output map uncertainty. Modelling uncertainty in plant parameters as output map uncertainty simplifies MPC design since state predictions are then deterministic, but can result in a higher degree of suboptimality. Thus for the 3rd order plant model:

$$A = \begin{bmatrix} -0.33 & 0.31 & -0.14 \\ 0.31 & -0.53 & 0.07 \\ -0.13 & 0.07 & -0.04 \end{bmatrix} \quad \bar{B} = \begin{bmatrix} 1.61 \\ -0.12 \\ -3.31 \end{bmatrix} \quad B_1 = \begin{bmatrix} 1.80 \\ 1.20 \\ -0.80 \end{bmatrix} \quad B_2 = \begin{bmatrix} 1.40 \\ 0.20 \\ 1.60 \end{bmatrix}$$

$$c_1 = \begin{bmatrix} 0.80 \; 3.30 \; -3.20 \end{bmatrix} \; c_2 = \begin{bmatrix} 2.60 \; 0.80 \; 1.20 \end{bmatrix},$$

with $U = 0.5$, $Y_2 = 2$, an approximate model realization involving only output map uncertainty can be constructed by identifying the means and variances of a pair of MA models. However, a stochastic MPC law for output map uncertainty designed using the approach of [12] (using 6th order MA models) gives an average closed-loop cost (over 100 initial conditions) of 114, whereas the average cost for (22) for the same set of initial conditions is 36.1.

References

[1] M.V. Kothare, V. Balakrishnan, and M. Morari. Robust constrained model predictive control using linear matrix inequalities. *Automatica*, 32:1361–1379, 1996.

[2] J.C. Allwright and G.C. Papvasiliou. Linear programming and robust model-predictive control using impulse-responses. *Systems & Control Letters*, 18:159–164, 1992.

[3] P.O.M. Scokaert and D.Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Trans. Autom. Control*, 43:1136–1142, 1998.

[4] A. Bemporad, F. Borrelli, and M. Morari. Min-max control of constrained uncertain discrete-time linear systems. *IEEE Trans. Automatic Control*, 48(9):1600–1606, 2003.

[5] D.W. Clarke, C. Mothadi, and P.S. Tuffs. Generalized predictive control. *Automatica*, 23:137–160, 1987.

[6] P. Li, M. Wendt, and G. Wozny. A probabilistically constrained model predictive controller. *Automatica*, 38(7):1171–1176, 2002.

[7] I. Batina. *Model predictive control for stochastic systems by randomized algorithms.* PhD thesis, Technische Universiteit Eindhoven, 2004.

[8] A.J. Felt. Stochastic linear model predictive control using nested decomposition. In *Proc. American Control Conf.*, pages 3602–3607, 2003.

[9] B. Kouvaritakis, M. Cannon, and P. Couchman. MPC as a tool for sustainable development integrated policy assessment. *IEEE Trans. Autom. Control*, 51(1):145–149, 2006.

[10] J.H. Lee and B.L. Cooley. Optimal feedback control strategies for state-space systems with stochastic parameters. *IEEE Trans. Autom. Control*, 43(10):1469–1475, 1998.

[11] D. van Hessem, C.W. Scherer, and O.H. Bosgra. LMI-based closed-loop economic optimization of stochastic process operation under state and input constraints. In *Proc. 40th IEEE Conf. Decision Control, Orlando*, 2001.

[12] P. Couchman, M. Cannon, and B. Kouvaritakis. Stochastic MPC with inequality stability constraints. *Automatica*, 2006. Submitted.

[13] P. Couchman, B. Kouvaritakis, and M. Cannon. LTV models in MPC for sustainable development. *Int. J. Control*, 79(1):63–73, 2006.

[14] D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000.

[15] M. Cannon, B. Kouvaritakis, and V. Deshmukh. Enlargement of polytopic terminal region in NMPC by interpolation and partial invariance. *Automatica*, 40:311–317, 2004.

[16] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory.* SIAM, 1994.

[17] M. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284:193–228, 1998.

[18] B.T. Polyak. *Introduction to optimization.* Opt. Software Inc., 1987.

NMPC for Complex Stochastic Systems Using a Markov Chain Monte Carlo Approach

Jan M. Maciejowski[1], Andrea Lecchini Visintini[2], and John Lygeros[3]

[1] Cambridge University Engineering Dept, Cambridge CB2 1PZ, England
{jmm,al394}@eng.cam.ac.uk
[2] Department of Engineering, University of Leicester, Leicester LE1 7RH, UK
alvl@leicester.ac.uk
[3] Automatic Control Laboratory, ETH Zurich, CH-8092, Switzerland
lygeros@ee.upatras.gr

Summary. Markov chain Monte Carlo methods can be used to make optimal decisions in very complex situations in which stochastic effects are prominent. We argue that these methods can be viewed as providing a class of nonlinear MPC methods. We discuss decision taking by maximising expected utility, and give an extension which allows constraints to be respected. We give a brief account of an application to air traffic control, and point out some other problem areas which appear to be very amenable to solution by the same approach.

1 Introduction

Model Predictive Control (MPC) is characterised by the following features [14]:

1. An explicit internal model used to generate predictions,
2. Online, real-time optimisation to determine the control signals (manipulated variables),
3. Use of repeated measurements and re-optimisation to obtain feedback,
4. Use of a receding horizon (optional),
5. Explicit consideration of constraints (optional).

In this chapter we will introduce a control method which appears very different from MPC, as it is usually understood. But it has all the features listed here, so we claim that it should be considered to be an MPC method. Furthermore, it can be used with nonlinear models of arbitrary 'nastiness', the only requirement being that predictions can be generated by simulating the model behaviour. Our method is definitely a *nonlinear* MPC method. It is even a *robust* nonlinear MPC method, because it can cope with uncertainty in the model.

On the other hand, there are some restrictions:

1. The uncertainty has to be probabilistic in nature for the method to make sense.
2. The criterion of optimality must be the expectation of some utility or performance function. We shall argue in section 2 that this is a very mild restriction.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 269–281, 2007.
springerlink.com

3. Constraints are respected in the sense that the probability of constraint violations is kept below some user-defined threshold. This threshold can be arbitrarily small (at the expense of computational complexity) but hard constraints are not enforced, strictly speaking.
4. Computational complexity is very high, so that the time required to find a solution at each step may be of the order of several hours. In section 6 we shall identify some applications for which this is not a problem. These applications arise in diverse areas such as batch process control, financial investment policy, and environment management.

Our method relies on stochastic optimisation. It makes use of a Monte Carlo Markov Chain technique for optimal decision-taking which was pioneered by Müller [16] and rediscovered independently by Doucet *et al* [8]. Our contribution to the method is a way of incorporating constraints into the problem formulation, which is presented in section 3. There is no requirement for convexity of the objective function, convergence (in probability) resulting from the very mild conditions that are required for the convergence of a homogeneous Markov chain. The technical details of the method are given in section 4.

Kouvaritakis *et al.* have previously proposed a stochastic MPC formulation for solving a problem in the area of sustainable development policy [12]. Their method involves the solution of a convex optimisation problem, with constraints in the form of thresholds for the probabilities of obtaining certain outcomes — as in our approach. The convexity requirement imposes limitations on the model, objective function, and constraints. Our approach is free of these limitations, but at the expense of greater computational complexity.

2 Maximising Expected Utility or Performance

The notion of rational decision-making under uncertainty as the maximisation of the statistical expectation of a utility function has a long history, dating back to Bernoulli [4] and Bentham [3]. Although it has gone into and out of fashion as a foundational axiom for economics, its place has been firmly established since its use (and formalisation) by von Neumann and Morgenstern in Game Theory [21], and extended to other applications of decision theory by various authors [1, 11, 18, 20].

Suppose that a set of possible decisions Ω is given, from which one decision $\omega \in \Omega$ must be chosen. For each such ω, let the outcome be a random variable X, and let its probability distribution be $P_\omega(x)$. Suppose that to each decision-outcome pair (ω, x) we can attach a real-valued *utility function* $u(\omega, x)$, such that the pair (ω_1, x_1) is preferred to the pair (ω_2, x_2) if $u(\omega_1, x_1) > u(\omega_2, x_2)$. Then the principle of maximising expected utility states that one should choose the decision optimally as follows:

$$\omega^* = \arg\max_{\omega} E_X u(\omega, x) = \arg\max_{\omega} \int u(\omega, x) dP_\omega(x) \tag{1}$$

In familiar finite-horizon LQG control theory, the utility is the negative cost

$$u(\omega, x) = -\sum_{k=0}^{N} \left(\xi_{k+1}^T Q \xi_{k+1} + \eta_k^T R \eta_k\right) \tag{2}$$

with $x = (\xi_1, \ldots, \xi_{N+1})$ being the state sequence and $\omega = (\eta_0, \ldots, \eta_N)$ being the control sequence.[1] Since we will abandon the luxury of linear models and Gaussian distributions, there will be no analytical advantage in retaining the quadratic structure, and we will use more general forms of utility function. In most applications the distribution $P_\omega(x)$ will be very complicated, and the only way to elicit it will be to draw samples from a simulator based on the model. The only way to estimate the integral in (1) will be by Monte Carlo methods. Thus the only requirement on the form of $u(\omega, x)$ will be sufficient regularity for such estimation to be possible.

This allows many possibilities. For example, if a single outcome x_0 is desired, then setting $u(\omega, x) = \delta(x - x_0)$ (the Dirac impulse) gives $E_X u(\omega, x) = P_\omega(x_0)$, and hence ω^* is a decision which maximises the likelihood of outcome x_0.

In the sequel, and in our related publication [13], we use the term *performance* instead of *utility.*

3 Satisfying Constraints

An important consideration for us is that we wish to satisfy constraints, with a higher priority than maximising the expected performance. That is, we wish to solve problems of the form

$$\omega^* = \arg\max_{\omega} E_X \mathrm{perf}(\omega, x) \quad \text{subject to} \quad Pr\{x^* \in \mathbf{X_f}\} > 1 - \epsilon, \quad (0 < \epsilon < 1) \tag{3}$$

where x^* is the outcome resulting from decision ω^* and $\mathbf{X_f}$ is a set of allowed (feasible) outcomes. The approach we take is of the penalty-function type, approximating this constrained problem by an alternative unconstrained problem. First we scale or otherwise transform the performance function $\mathrm{perf}(\omega, x)$ so that[2]

$$0 \le \mathrm{perf}(\omega, x) \le 1 \quad \text{for all} \quad \omega, x \tag{4}$$

Then we define

$$u(\omega, x) = \begin{cases} \mathrm{perf}(\omega, x) + \Lambda & \text{if} \quad x \in \mathbf{X_f} \\ \\ 1 & \text{if} \quad x \notin \mathbf{X_f}, \end{cases} \tag{5}$$

where $\Lambda > 1$ is a parameter which represents a reward for constraint satisfaction. We then solve problem (3) approximately by solving problem (1).

[1] The algorithm which we will present in section 4 requires u to be positive, so (2) could not be used with this algorithm without modification.

[2] This is a mild restriction on perf(.,.), that such a transformation exists. Usually the initial definition of perf(.,.) is such that (4) holds.

The quality of the approximation is characterised by the following results, where

$$\text{PERF}_{max|\epsilon} = \sup_{\omega} E_X \text{perf}(\omega, x) \quad \text{subject to} \quad Pr\{x \notin \mathbf{X_f}\} < \epsilon \tag{6}$$

Theorem 1

$$Pr\{x^* \notin \mathbf{X_f}\} \leq \left(1 - \frac{1}{\Lambda}\right) \inf_{\omega} \left[Pr\{x \notin \mathbf{X_f}\}\right] + \frac{1}{\Lambda} \tag{7}$$

$$E_X \text{perf}(\omega^*, x) \geq \text{PERF}_{max|\epsilon} - (\Lambda - 1)\left(\epsilon - \inf_{\omega} Pr\{x \notin \mathbf{X_f}\}\right) \tag{8}$$

The bound (7) suggests that if a decision ω exists such that the probability of violating the constraints is very small, then $\Lambda \approx 1/\epsilon$ is appropriate. More precisely we have:

Corollary 1. *If* $\inf_{\omega} Pr\{x \notin \mathbf{X_f}\} \leq \epsilon/2$ *and* $\Lambda = 2/\epsilon$ *then*

$$Pr\{x^* \notin \mathbf{X_f}\} \leq \epsilon - \frac{\epsilon^2}{4} < \epsilon \tag{9}$$

However, (8) shows that a large Λ may result in an unnecessary loss of performance. Compromise choices of Λ are discussed in [13].

4 Optimisation Using an MCMC Approach

The Markov Chain Monte Carlo (MCMC) optimisation procedure works as follows. Although the decision to be found is not a random variable, we consider it to be a random variable Ω. Suppose for the moment that we know its distribution, and that we can extract random samples ω from this distribution. For each such sample, we can run a model-based simulation J times, and obtain J realisations of the (random variable) outcome: $(x_1, x_2, \ldots, x_J)$. We thus have a realisation $(\omega, x_1, x_2, \ldots, x_J)$ of the joint random variable $(\Omega, X_1, X_2, \ldots, X_J)$, in which the X_i's all have the same marginal distribution. Now suppose that we do this repeatedly according to some procedure, and let $(\Omega, X_1, X_2, \ldots, X_J)_n$ denote the random variable at the n'th step. (If the procedure were to always make independent extractions from the same joint distribution then this random variable would not depend on n.) We can evaluate the performance of each of the J outcomes at each step by computing the values $u(\omega, x_i), i = 1, \ldots, J$, and then score the decision ω by the value $\tilde{u}_J = \prod_{i=1}^{J} u(\omega, x_i)$. This value $\tilde{u}_J$ is the realisation of a random variable, which we denote by $\tilde{U}_J$. We then decide randomly whether to 'accept' the pair $(\omega, \tilde{u}_J)$ or to keep the value we obtained at the $(n-1)$'th step, and we update the distribution of Ω. Let $(\Omega, U_J)_n$ denote the joint random variable obtained at the n'th step in this way. The clever part is to make this random decision, and the update of the distribution, such as to obtain the following properties:

1. The sequence of random variables $(\Omega, U_J)_n$ should be a *Markov chain.* That is, for each n its probability distribution should depend only on the distribution of $(\Omega, U_J)_{n-1}$.
2. Furthermore, it should be a *homogeneous* Markov chain, namely it should have an equilibrium distribution, independent of n. This is so that the chain converges to its equilibrium distribution as $n \to \infty$. (In [17] this requirement is relaxed.)
3. When the Markov chain is at its equilibrium distribution, the marginal distribution of Ω should be (proportional to) $[E_X u(\omega, x)]^J$. If J is large enough, this distribution will be concentrated close to the optimal value(s) of ω, and most random samples generated from it will be close to the optimal solution.

This is achieved by the following algorithm. We start with a distribution $g(\omega)$, which is known as the *instrumental* (or *proposal*) distribution and is (almost) freely chosen by the user.

Algorithm 2 (MCMC Algorithm)

```
initialization:
Extract a sample $\omega(0)$ of the random variable $\Omega(0)$ from
    the distribution $g(\omega)$.
Extract $J$ independent samples $x_j(0)$ of the random variable $X_j(0)$ from
    the (unknown) distribution $p_{\Omega(0)}(x)$, by running the simulator $J$
    times.
Compute $U_J(0) = \prod_{j=1}^{J} u(\omega(0), x_j(0))$.
Set $k = 0$.
repeat
Extract a sample $\tilde{\omega}$ of the random variable $\tilde{\Omega} \sim g(\omega)$.
Extract $J$ independent samples $\tilde{x}_j$ of the random variable $\tilde{X}_j \sim p_{\tilde{\Omega}}(x)$,
    $j = 1, \ldots, J$, by running the simulator $J$ times.
Compute $\tilde{U}_J = \prod_{j=1}^{J} u(\tilde{\omega}, \tilde{x}_j)$
Set $\rho = \min\left\{1, \frac{\tilde{U}_J}{u_J(k)} \frac{g(\omega(k))}{g(\tilde{\omega})}\right\}$
Set $[\Omega(k+1), U_J(k+1)] = \begin{cases} [\tilde{\omega}, \tilde{U}_J] & \text{with probability } \rho \\ [\omega(k), U_J(k)] & \text{with probability } 1-\rho \end{cases}$
Set $k = k + 1$.
until True
```

In the initialization step the state $[\omega(0), U_J(0)]$ is always accepted. In subsequent steps the new extraction $[\tilde{\omega}, \tilde{U}_J]$ is accepted with probability ρ otherwise it is rejected and the previous state of the Markov chain $[\omega(k), u_J(k)]$ is maintained. In practice, the algorithm is executed until a certain number of extractions (say 1000) have been accepted. Because we are interested in the equilibrium distribution of the Markov chain, the first few (say 10%) of the accepted states are discarded to allow the chain to reach its equilibrium distribution ("burn in period").

Let $h(\omega, x_1, x_2, \ldots, x_J)$ denote the joint distribution of $(\Omega, X_1, X_2, X_3, \ldots, X_J)$. If

$$h(\omega, x_1, x_2, \ldots, x_J) \propto \prod_{j=1}^{J} u(\omega, x_j) p_\Omega(x_j) \tag{10}$$

then the marginal distribution of Ω, also denoted by $h(\omega)$ for simplicity, satisfies

$$h(\omega) \propto \left[\int u(\omega, x) p_\Omega(x) dx \right]^J = [E_X u(\omega, x)]^J \tag{11}$$

which is the property we specified as property 3 above.

The random *accept/reject* mechanism, governed by the probability ρ, is a particular formulation of the *Metropolis-Hastings* algorithm, which is a general algorithm for generating a homogeneous Markov chain [19] . For a desired (target) distribution given by $h(\omega, x_1, x_2, \ldots, x_J)$ and proposal distribution given by

$$g(\omega) \prod_j p_\Omega(x_j)$$

the acceptance probability for the standard Metropolis-Hastings algorithm is

$$\min \left\{ 1, \frac{h(\tilde{\omega}, \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_J)}{h(\omega, x_1, x_2, \ldots, x_J)} \frac{g(\omega) \prod_j p_\Omega(x_j)}{g(\tilde{\omega}) \prod_j p_{\tilde{\Omega}}(\tilde{x}_j)} \right\}$$

By inserting (10) in this expression one obtains the probability ρ as defined in the algorithm. Under minimal assumptions, the Markov Chain generated by the $\Omega(k)$ is uniformly ergodic with equilibrium distribution $h(\omega)$ given by (11). Therefore, after a burn in period, the extractions $\Omega(k)$ accepted by the algorithm will concentrate around the modes of $h(\omega)$, which, by (11) coincide with the optimal points of $U(\omega)$. Results that characterize the convergence rate to the equilibrium distribution can be found, for example, in [19].

5 Using MCMC for NMPC

The original successful idea of MPC was to use general-purpose optimisation algorithms to solve a sequence of open-loop problems, with feedback being introduced by resetting the initial conditions for each problem on the basis of updated measurements. Since the algorithm which we have described is an optimisation algorithm, we can use it in exactly this way to implement nonlinear MPC. But it is desirable to modify it in some ways, so as to make it more suitable for solving MPC problems. Increased suitability has at least two aspects:

1. Addressing the theoretical concerns that have been raised for MPC, in particular stability, feasibility, and robust versions of these.
2. Reducing the computational complexity, to allow a wider range of problems to be tackled by the MCMC approach.

In many of the application areas that we envisage being suitable for the application of the MCMC algorithm, performance is often associated with achievements at the *end* of a prediction horizon — quality of a batch at the end of processing, value of an asset portfolio at a maturity date, total emissions of pollutants by 2020, etc. There is no fundamental problem with this, but as is pointed out in [12], it could be dangerous to have an objective at the end of the horizon only, and then implement a receding-horizon strategy, since very unsatisfactory trajectories could result which were never 'seen' by the performance criterion. As in conventional MPC, one can expect that such problems can be avoided by having sufficiently many points in the horizon at which the performance is assessed.[3] An alternative might be to avoid such problems by imposing suitable constraints on trajectories during the horizon. Much of MPC stability theory relies on using the 'cost-to-go' as a Lyapunov function. A stochastic version of this approach may be applicable to the MCMC approach, although this would impose a restriction on the performance criterion (which would need to be increasing with time under certain conditions).

Since our interest is in causal dynamic models and time trajectories of variables, we expect that the computational complexity of the MCMC approach for MPC can be reduced by exploiting *sequential* Monte Carlo techniques [7]. These techniques are roughly analogous to recursive filtering methods, but for nonlinear systems and non-Gaussian processes. Apart from exploiting the temporal structure, it should be noted that there is considerable scope for speed-up by parallelisation, since each of the J simulations that are required at each step of the MCMC algorithm can be executed in parallel.

Our current and planned research is addressing these issues. But it is important to emphasise that applications of the MCMC approach to NMPC problems do not need to wait for the outcome of this and similar research by others. As in the early days of MPC (and as still practiced more often than not) it is possible to take a 'just do it' approach in the face of a dearth of theoretical results,[4] and to respond to computational complexity by reducing the degrees of freedom.[5]

6 Applications

It should be stated at the outset that this section will not provide a list of successful applications of the proposed MCMC-based approach to NMPC. So far we have applied the approach to only one application, which will be outlined in section 6.1, and the development of the approach even to this application is still far from complete. Our main purpose here is to draw attention to the fact that there are many other applications, particularly outside the areas normally

[3] 'Coincidence points' in conventional MPC jargon.

[4] Although the situation is a little worse than with early MPC, which did at least have the possibility of checking nominal closed-loop stability *post-hoc* for unconstrained linear models.

[5] Early MPC implementations typically used control horizons of only 1 or 2 steps.

considered by control engineers, which appear to be amenable to solution by our approach.

6.1 Air Traffic Control

Our original motivation for developing the MCMC method for MPC came from a problem in civilian air traffic control (ATC) [13]. The current workload of air traffic controllers is such that they must be provided with some kind of automated assistance, if projected increases in air traffic are to be safely accommodated [5, 10]. We have examined two types of scenarios. In the first, a potential conflict between two aircraft during a prediction horizon (of typical length 20 minutes) is detected, and instructions must be given to the two aircraft for maneouvres which will resolve the conflict. In the second, a number of aircraft have entered a *terminal manoeuvering area* (TMA), which is the region in which aircraft descend from cruising altitude to the beginning of their final approach to the runway. Instructions must be given to each of them, to maintain safe separation between them, and to achieve desirable objectives such as minimising fuel use or landing as quickly as possible.

The aircraft are modelled realistically, both as regards their dynamics, and as regards their standard flight profiles [9] — for example, the aircraft speed is scheduled on its altitude in normal operation, the schedule being dependent on both the aircraft type and the airline. A major source of uncertainty is the effect of wind; realistic spatial and temporal correlation has been included in the model. Further uncertainty is in the parameters of individual aircraft, precise values of masses and moments of inertia being unknown to air traffic control. There is also some uncertainty in the response time between an instruction being issued by a controller and the aircraft's response. All of these uncertainties are modelled stochastically in our simulator. Their combined effect is surprisingly large — an aircraft descending from a given point at 35000 feet to 10000 feet may arrive at the lower altitude anywhere in an interval of some tens of kilometres — see Figure 1.

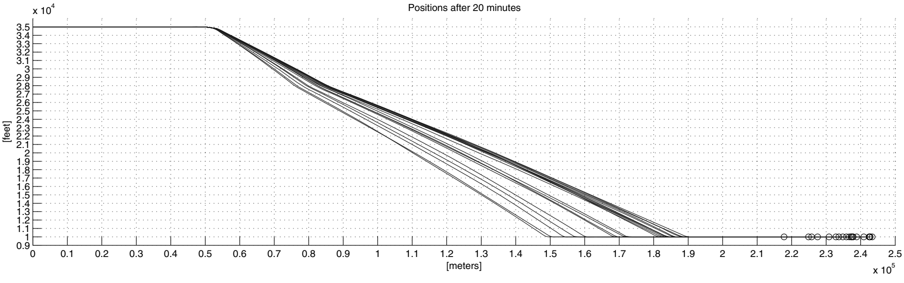


Fig. 1. A typical set of trajectories for an aircraft descending from 35000 feet to 10000 feet

A difference from usual control engineering is that the decision variables (control inputs) do not follow arbitrary trajectories in time. Particularly in the terminal manoeuvering area, there is a finite number of possible standard maneouvres, such as 'descend', 'circle while descending', 'turn right through 90 degrees', etc, and an expected sequence of these maneouvres. The available decisions are (real-valued) parameters of these manoeuvres, such as the initiation time, diameter of the circle, rate of descent, etc.

In both the scenarios defined above, there are safety-related constraints (minimum horizontal and vertical separation between aircraft), legislative constraints (eg not entering certain urban zones at night), and desirable objectives (minimum disruption of scheduled flight-plan, landing aircraft X as soon as possible, etc). The objectives are liable to change with time — for example, an aircraft might enter the TMA which has already been delayed *en route*, and should therefore be given priority over aircraft which are already in the area. It is therefore important to have a solution methodology that can accept a wide range of performance functions to be maximised. This is not to suggest that formulations of such performance functions should be done in real time; a set of performance functions should be devised for a set of scenarios, and their suitability evaluated carefully before being put into use. Then the choice of a suitable function from the set could be made in real time.

The time available for making a decision in the ATC context is of the order of a few minutes. (Emergency manoeuvres required for avoiding imminent collision require much faster decisions of course, but these are handled locally by the affected aircraft, without involving ATC.) Our current implementation of the MCMC approach is much slower than this, even when only two decision variables are involved. A speed-up of about one order of magnitude should result from more efficient coding at an elementary level (much of the time is currently lost by very inefficient hand-overs between the *Java*-based simulator and the *Matlab*-based MCMC algorithm), but more sophisticated algorithm development will be required to obtain some further speed-up.

A typical scenario involves two aircraft, one of which is following a fixed set of instructions, while the other will fly a straight course to a waypoint which needs to be selected, after which it will fly to a fixed waypoint. The performance objective is that they should arrive at a final waypoint (glide-slope capture) separated in time by 300 sec; this is represented by the objective function $\exp\{-a|(|T_1 - T_2|) - 300|\}$, where T_1 and T_2 are the arrival times of the two aircraft at the final waypoint, and $a > 0$. The constraint is that their minimum separation should be 5 nautical miles horizontally and 1000 feet vertically at all times, and the probability of violating this constraint should be smaller than $\epsilon = 0.1$. As formulated, this simple scenario is a finite-time problem, which may be re-solved as time proceeds, but it is a 'shrinking-horizon' rather than a receding-horizon scenario. This problem was solved by initially choosing the instrumental distribution $g(\omega)$ uniform over the possible parameter space (a rectangle in $\mathbb{R}^2$), and $J = 10$. This gave two rather large 'clouds' of possible solutions, but all of them safe ones. Another instrumental distribution $g(\omega)$ was obtained as a sum-of-Gaussians, fitted to the solutions obtained from the first run, and the MCMC algorithm was run with $J = 50$. Finally this procedure was repeated with $J = 100$. After the Markov chain has reached equilibrium (approximately), 1000 accepted states still form a 'cloud' which is not very concentrated — see Figure 2. However, this is not necessarily a problem: air traffic controllers have told us that being presented with such a choice of solutions, with the assurance that each one is safe, is for them a very acceptable outcome, because it leaves them some freedom in making the final decision. The number of simulations required for $J = (10, 50, 100)$ was 40740, 161764, and 366666, respectively. Note that the number of simulations required with the larger values of J would have been much larger if good instrumental distributions had not been available from previous runs.

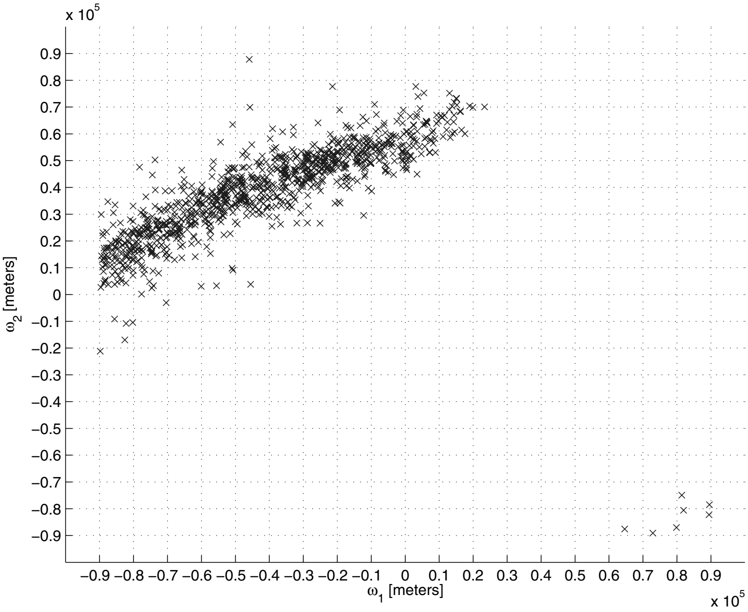


Fig. 2. The 'cloud' of 1000 accepted solutions with $J = 100$

6.2 Batch Process Control

Batch process control is often characterised by very nonlinear models, which are often quite simple and with very uncertain parameters (which may be described by probability distributions). The performance criterion often depends on product quality at the end of the process, although time-integral criteria (such as total energy used) also appear. The number of degrees of freedom is often quite small (a temperature at which to run one step of the process, the duration of a process stage, etc), and the update interval between decisions can be very large (hours, days). All these characteristics indicate that the MCMC approach should be very suitable for many batch processes.

6.3 Environment Management

In [12] a version of stochastic MPC is developed for a problem of investment decisions into R&D on alternative sustainable technologies. A stochastic model is available which describes the effects of R&D expenditure into alternative technologies, on various indicators of sustainability, such as CO_2 emissions and energy use, over a horizon of 30 years. The objective is to decide on the allocation over time of expenditure into the competing technologies, subject to a budgetary constraint. The performance criterion is the probability that a particular indicator exceeds some threshold value, and the constraints are minimum probabilities of other indicators exceeding their threshold values. Here is an application in which 'real-time' means updating the solution once a year, so that the computational complexity of the MCMC approach is not an issue.

There are many other environmental problems, such as water resources management, fishery harvesting policy, atmospheric ozone regulation, river quality management, etc, that require decision-making over time on the basis of dynamic models, under conditions of considerable uncertainty [2]. All of these appear to be suitable applications of the MCMC approach.

6.4 Financial Management

The problem of optimal portfolio allocation is often posed as a problem of stochastic control, and has attracted very sophisticated solutions [6]. This is the problem of distributing (and re-distributing from time to time) a fixed investment budget into various asset classes, so as to maximise the value of the portfolio, or the income, etc [15]. Very approximate, inherently stochastic models, are available for the future trajectory of value appreciation and/or income from each asset class, and constraints may be present, such as a limit on the probability of losses exceeding some specified value. Update intervals may range from hours to months, depending on the kinds of assets considered. Once again, these problems appear to be amenable to the MCMC approach.

7 Conclusions

We have presented a very powerful and general approach to solving optimisation problems in the form of maximisation of the expected value of a performance criterion, subject to satisfying a set of constraints with a prescribed probability. The only essential requirement for implementation of the approach is a model-based stochastic simulator. We have argued that the form of the criterion is not restrictive, and that this approach is applicable to a wide variety of stochastic control problems. Since the approach depends on the availability of a model, and can be applied repeatedly in real-time, updated by the latest measurements, we believe that it qualifies to be considered as an MPC method.

The two approaches can be contrasted as follows:

1. Conventional NMPC typically assumes that disturbances and other sources of uncertainty are bounded but otherwise unknown. MCMC requires a stochastic description of uncertainty.

2. Conventional NMPC does not guarantee convergence to the global optimum. MCMC guarantees convergence to a near-optimal solution.
3. Conventional NMPC interprets constraints as hard bounds. MCMC gives a pre-defined probability of not violating constraints.
4. Conventional NMPC uses hill-climbing optimization. MCMC uses stochastic optimization, which requires much more computational effort and time.
5. Satisfactory behaviour of conventional NMPC over time can be achieved using known techniques. The behaviour of MCMC when used repeatedly in a receding-horizon context has not yet been investigated.

The application of the MCMC approach to air traffic control problems has been described. Some speculations on other possible applications have also been given. It is expected that many such applications will be found outside traditional engineering areas, where nonlinear uncertain models are available, and enough time is available to complete the required computations 'on-line'.

Research is currently under way to make the MCMC algorithm more suitable for MPC applications. But some applications can be tackled now, without waiting for the results of further research.

References

[1] K.J. Arrow. *Social Choice and Individual Values.* Yale University Press, 2nd edition, 1963.

[2] M.B. Beck, editor. *Environmental Foresight and Models: a Manifesto.* Elsevier, 2002.

[3] J. Bentham. *Principles of Morals and Legislation.* Oxford University Press, 1823 (originally published 1789).

[4] D. Bernoulli. Specimen theoriae novae de mensura sortis, Commentarii Academiae Scientiarum Imperialis Petropolitanae (5, 175-192, 1738). *Econometrica*, 22:23–36, 1954. Translated by L. Sommer.

[5] H. Blom and J. Lygeros. *Stochastic Hybrid Systems: Theory and Safety Critical Applications.* Springer-Verlag, 2006.

[6] M.H.A. Davis and M. Zervos. A problem of singular stochastic control with discretionary stopping. *Annals of Applied Probability*, 4:226–240, 1994.

[7] A. Doucet, N. de Freitas, and N. Gordon, editors. *Sequential Monte Carlo Methods in Practice.* Statistics for Engineering and Information Science. Springer, New York, 2001.

[8] A. Doucet, S.J. Godsill, and C. Robert. Marginal maximum a posteriori estimation using Markov chain Monte Carlo. *Statistics and Computing*, 12:77–84, 2002.

[9] W. Glover and J. Lygeros. A stochastic hybrid model for air traffic control simulation. In R. Alur and G. Pappas, editors, *Hybrid Systems: Computation and Control*, number 2993 in LNCS, pages 372–386. Springr-Verlag, 2004.

[10] HYBRIDGE: Distributed Control and Stochastic Analysis of Hybrid Systems Supporting Safety Critical Real-Time Systems Design. `http://www.nlr.nl/public/hosted-sites/hybridge/`, 2005. European Commission FP5 project, contract IST-2001-32460.

[11] R.C. Jeffrey. *The Logic of Decision.* University of Chicago Press, 2nd edition, 1983.

[12] B. Kouvaritakis, M. Cannon, and V. Tsachouridis. Recent developments in stochastic MPC and sustainable development. In J.J. Gertler, editor, *Annual Reviews in Control*, volume 28, pages 23–35. Elsevier, 2004.

[13] A. Lecchini Visintini, W. Glover, J. Lygeros, and J.M. Maciejowski. Monte Carlo optimization strategies for air traffic control. In *Proc. AIAA Guidance, Navigation and Control Conference*, San Francisco, August 2005.

[14] J.M. Maciejowski. *Predictive Control with Constraints.* Prentice-Hall, Harlow UK, 2002.

[15] H. Markowitz. Portfolio selection. *Journal of Finance*, 7:77–91, 1952.

[16] P. Müller. Simulation based optimal design. In J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, editors, *Bayesian Statistics 6*, pages 459–474. Oxford University Press, 1999.

[17] P. Müller, B. Sansó, and M. De Iorio. Optimal Bayesian design by inhomogeneous Markov chain simulation. *J. American Statistical Association*, 99(467):788–798, 2004.

[18] H. Raiffa. *Decision Analysis: Introductory Lectures on Choices Under Uncertainty.* Addison-Wesley, Reading, MA, 1968.

[19] C.P. Robert and G. Casella. *Monte Carlo Statistical Methods.* Springer-Verlag, New York, 1999.

[20] L.J. Savage. *The Foundations of Statistics.* Wiley, New York, 1954.

[21] J. von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior.* Princeton University Press, 1944.

On Disturbance Attenuation of Nonlinear Moving Horizon Control

Hong Chen[1], Xingquan Gao[1,⋆], Hu Wang[1], and Rolf Findeisen[2]

[1] Department of Control Science and Engineering, Jilin University, Changchun, PR China,
chenh@jlu.edu.cn

[2] Institute for Systems Theory in Engineering, University of Stuttgart, Germany
findeise@ist.uni-stuttgart.de

Summary. This paper addresses the disturbance attenuation problem in nonlinear moving horizon control. Conceptually a minimax formulation with a general dissipation constraint is suggested and theoretical results on closed-loop dissipation, $\mathcal{L}_2$ disturbance attenuation and stability are discussed. The implementation issue is attacked with respect to tracking a reference trajectory in the presence of external disturbances and control constraints, and a computationally tractable algorithm is given in the framework of LMI optimization. Simulation and comparisons of setpoint tracking control of a CSTR are presented.

1 Introduction

The academic research of moving horizon control, mostly referred to as model predictive control (MPC), has achieved significant progresses with respect to the stability and robustness issues [ABQ99, BM99, MRR00]. It is however rarely addressed how to guarantee disturbance attenuation in MPC, some schemes with additive bounded disturbances see for example [SM98, BBM01]. Possibilities to take uncertainties or disturbances are either game theoretic approaches to moving horizon control [LK94, CSA97, MNS03] with the purpose of incorporating well-known robustness guarantees through $\mathcal{H}_\infty$ constraints into MPC schemes, or moving horizon approaches to the time-varying or nonlinear $\mathcal{H}_\infty$ control problems (e.g. [BB99, MNS01]). Previous works can be for example found in [Tad92, LK94] for linear time-varying systems and recently in [BB99, MNS01, MNS02, Gyu02] for nonlinear systems, where time-domain constraints are not taken into account except for [Gyu02]. These formulations perform implicitly an infinite (or quasi-infinite) horizon feedback prediction. As a benefit of neglecting time-domain constraints, a prescribed constant disturbance attenuation level can be strived. In the presence of time-domain constraints, [CSA97] uses a feedback controller as precompensator to guarantee that the terminal region is rendered robust invariant, and an open-loop optimization problem is then solved for the feedback pre-compensated system to obtain the MPC action. When taking disturbances into account, the dissipation inequality

[⋆] Currently with Jilin Institute of Chemical Technology, Jilin, PR China.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 283–294, 2007.
springerlink.com

of the moving horizon system seems to play an important role. For the finite-horizon moving horizon formulations, this is achieved by choosing the terminal penalty function to satisfy the Hamilton-Jacobi-Isaacs inequality locally in a positive invariant terminal region defined as a value set of the terminal function. In [CSA97, MNS01], the positive invariance of the terminal region is sufficiently guaranteed by restricting the amplitude of the disturbance to $\|w(t)\| \leq \delta \|z(t)\|$ (or the discrete form in [MNS03]), whereas in [Gyu02] it is included in primal assumptions. Removing the rather restrictive hypotheses on the size of disturbances, it is shown in [CS03] that closed-loop dissipation might fail, even if dissipation inequalities are satisfied at each optimization step. This is first observed in [SCA02] with respect to switching between $\mathcal{H}_\infty$ controllers, where a condition is derived to recover dissipation. This condition is called dissipation condition in [CS03, CS04] and introduced into the on-line optimization problem to enforce dissipation for the moving horizon system.

This paper extends the results in [CS03, CS04] to address the disturbance attenuation issue of nonlinear moving horizon control. Section 2 presents a conceptual minimax moving horizon control formulation for nonlinear constrained systems and the theoretical results on closed-loop dissipation, $\mathcal{L}_2$ disturbance attenuation and stability. In Section 3, the implementation issue of the suggested formulation is addressed with respect to tracking in the presence of disturbances and control constraints. Simulation and comparison results of setpoint tracking control of a CSTR are given in Section 4.

2 Moving Horizon Control with Disturbance Attenuation

Consider a nonlinear system described by

$$\begin{aligned} &\dot{x}(t) = f\left(x(t), w(t), u(t)\right), \, x(t_0) = x_0, \\ &z_1(t) = h_1\left(x(t), w(t), u(t)\right), \;\; z_2(t) = h_2\left(x(t), u(t)\right), \end{aligned} \tag{1}$$

with time-domain constraints

$$|z_{2j}(t)| \leq z_{2j,\, max}, j = 1, 2, \cdots, p_2, \; t \geq t_0, \tag{2}$$

where $x \in \mathbb{R}^n$ is the state, $w \in \mathbb{R}^{m_1}$ is the external disturbance, $u \in \mathbb{R}^{m_2}$ is the control input, $z_1 \in \mathbb{R}^{p_1}$ is the performance output and $z_2 \in \mathbb{R}^{p_2}$ is the constrained output. It is assumed that the vector fields $f : \mathbb{R}^n \times \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \to \mathbb{R}^n$, $h_1 : \mathbb{R}^n \times \mathbb{R}^{m_1} \times \mathbb{R}^{m_2} \to \mathbb{R}^{p_1}$ and $h_2 : \mathbb{R}^n \times \mathbb{R}^{m_2} \to \mathbb{R}^{p_2}$ are sufficiently smooth and satisfy $f(0,0,0) = 0$, $h_1(0,0,0) = 0$ and $h_2(0,0) = 0$.

With respect to disturbance attenuation, we strive to solve the following minimax optimization problem for the system (1) with the initial state $x(t_0)$ in moving horizon fashion:

$$\min_{u \in \mathcal{U}} \max_{w \in \mathcal{W}} \int_{t_0}^{\infty} \|z_1(t)\|^2 - \gamma^2 \|w(t)\|^2 dt. \tag{3}$$

Here $\mathcal{U}$ denotes the set of all admissible controls such that time-domain constraints are respected; $\mathcal{W}$ represents the set of all admissible disturbances. In the following we assume that there exists an admissible (optimal) control for the minimization part of (3) with the initial condition $x(t_0) = x$, *i.e.*,, $u_0^* \in \mathcal{U}$ for which

$$V(x) := \max_{w \in \mathcal{W}} \int_{t_0}^{\infty} \|z_1(t)\|^2 - \gamma^2 \|w(t)\|^2 dt \tag{4}$$

with a given (or minimized) γ. We assume that V is locally continuous in x, $0 < V(x) < \infty$ for $x \neq 0$ and $V(0) = 0$. By dynamic programming, we obtain from (4) the integral dissipation inequality

$$V(x(t_1)) - V(x(t_0)) \leq - \int_{t_0}^{t_1} \|z_1(t)\|^2 - \gamma^2 \|w(t)\|^2 dt \tag{5}$$

for the system (1) with u_0^* and any $w \in \mathcal{W}$.

Remark 1. *Note that u_0^* can be for example given by $u_0^* := \{\kappa_0(x), \kappa_1(x), \cdots\}$, if an open-loop u_0^* is too conservative due to the existence of external disturbances. In the following, we discuss conceptually how to achieve $\mathcal{L}_2$ disturbance attenuation of NMPC, i.e. we do not care how u_0^* is obtained, although this is in general difficult and important for the implementation. We will propose in the next section how to find suitable feedbacks and inputs.*

Considering the principle of MPC, the optimization problem (3) will be solved repeatedly at each time $t_k \geq t_0$, updated by the actual state $x(t_k)$. The obtained control action is injected into the system until the next sampling time, *i.e.*,,

$$u(t) := u_k^*(t),\ t \in [t_k, t_{k+1}), t_k \geq t_0. \tag{6}$$

By the existence and boundedness assumption for each time instant, we obtain a sequence of pairs $(V_k, \gamma_k), k = 0, 1, 2, \cdots$ satisfying (5). As shown in [SCA02], however, it might not be possible to guarantee dissipation for the closed-loop system after $k \geq 1$. A quadratic switching condition is there derived to recover the closed-loop dissipation and hence $\mathcal{L}_2$ disturbance attenuation. In the context of MPC, this condition (called as dissipation constraint) is introduced into the on-line solved optimization problem to enforce the dissipation property of the moving horizon system [CS03, CS04]. We propose to extend the dissipation constraint, that will be added to (3) as a constraint for all $k \geq 1$, in a general non-quadratic form by

$$p_0 - p_{k-1} + V_{k-1}\left(x(t_k)\right) - V_k\left(x(t_k)\right) \geq 0, \tag{7}$$

where $p_0 = V_0(x_0)$ and p_k is recursively computed by

$$p_k := p_{k-1} - \left[V_{k-1}(x(t_k)) - V_k(x(t_k))\right] \ . \tag{8}$$

Hence, at each sampling time $t_k \geq t_1$ with the actual state $x(t_k)$, the moving horizon control defined in (6) is obtained via (existence assumed)

$$u_k^* := \arg\min_{u \in \mathcal{U}} \max_{w \in \mathcal{W}} \int_{t_k}^{\infty} \|z_1(t)\|^2 - \gamma^2 \|w(t)\|^2 dt \text{ subject to (7).} \tag{9}$$

For the closed-loop system with (6), if we define the piecewise continuous function $V(x(t), t)$ as

$$V(x(t), t) := V_k(x(t)), \; t \in [t_k, t_{k+1}), t_k \geq t_0 \tag{10}$$

we can state the following result:

Proposition 1. *For any $\tau > t_0$, the moving horizon closed-loop system given by (1) and (6) is dissipative in the sense of*

$$V(x_0, t_0) + \int_{t_0}^{\tau} \gamma(t)^2 \|w(t)\|^2 - \|z_1(t)\|^2 dt \geq V(x(\tau), \tau) \tag{11}$$

where $\gamma(t)$ is piecewise constant and defined as

$$\gamma(t) := \gamma_i, \, \forall t \in [t_i, t_{i+1}), t_i \geq t_0. \tag{12}$$

Proof: We denote the sequence of the sampling times as $t_0, t_1, t_2, \cdots$ and consider that $t_k < \tau$ coincides with the sampling time closest to τ. By a simple addition of the sequent inequalities (5) that are satisfied by the pairs (V_k, γ_k) for $k = 0, 1, 2, \cdots$, we arrive at

$$V(x(\tau), \tau) - V(x_0, t_0) \leq \sum_{i=1}^{k} V_i(x(t_i)) - V_{i-1}(x(t_i)) \\ - \int_{t_0}^{\tau} \|z_1(t)\|^2 - \gamma(t)^2 \|w(t)\|^2 dt, \tag{13}$$

where (10) and (12) are used. Substituting (8) into (7) recursively, we conclude that the dissipation constraint enforces $\sum_{i=1}^{k} V_i(x(t_i)) - V_{i-1}(x(t_i)) \leq 0$. Combining it with (13) leads then to (11), as required. □

The following result is a direct consequence of Proposition 1.

Teorema 2.1. *The moving horizon closed-loop system admits an $\mathcal{L}_2$ -gain from the disturbance w to the performance output z_1 less than $\bar{\gamma}$ given by*

$$\bar{\gamma} := \max_{t \geq t_0} \gamma(t). \tag{14}$$

Teorema 2.2. *If the system (1) with the output z_1 is zero-state detectable and the disturbance has finite energy, then, the moving horizon closed-loop system is asymptotically stable; furthermore, if the disturbance amplitude is bounded in the form of*

$$\int_{t_i}^{t_{i+1}} \|w(t)\|^2 dt \le \int_{t_i}^{t_{i+1}} \frac{\|z_1(t)\|^2}{\gamma_i^2} dt, \ \forall t_i \ge t_0 \tag{15}$$

then, the moving horizon closed-loop system is stable in the sense of Lyapunov.

Proof: For finite energy disturbances, due to $V(x,t) \ge 0$, it follows from (11) that $\int_{t_0}^{\infty} \|z_1(t)\|^2 dt \le x(t_0)^T P_0 x(t_0) + \int_{t_0}^{\infty} \gamma(t)^2 \|w(t)\|^2 dt < \infty$, which implies $x(t) \to 0$ as $t \to \infty$ by the zero-state detectability. If the disturbances satisfy (15), we obtain from (11) that $V(x(\tau),\tau) \le V(x_0,t_0)$, $\forall \tau \ge t_0$. This implies that the closed-loop system is Lyapunov stable (e.g. [Kha92]). □

3 Implementation with Respect to the Tracking Problem

While the approach outlined in the previous section is theoretically appealing, the question arises whether it can be implemented at all, due to the computational complexity. This section addresses the implementation issue. For constrained linear system, [CS04, CS05] presents a computationally tractable formulation in the framework of LMI optimization. In order to reduce conservatism involving in the ellipsoid evaluation of time-domain constraints and avoid infeasibility, [CGW06] suggests an improved LMI optimization problem in terms of Lagrange duality. We now provide a tractable implementation for the suggested minimax formulation with respect to the tracking problem in the presence of external disturbances and time-domain constraints, considering the time-varying linearized error system.

3.1 Proposed Algorithm

For simplicity, we consider only control constraints of the form

$$|u_j(t)| \le u_{j,\max}, \ j = 1, 2, \cdots, m_2, \ t \ge t_0. \tag{16}$$

Given a (time-varying, pre-known) reference trajectory (z_d, x_d, u_d) consistent with the unperturbed nonlinear system, let us define the errors $x_e := x - x_d$, $u_e := u - u_d$, $z_e := z_1 - z_d$ and linearize (1) about the reference trajectory at each sampling time t_k. The error dynamics system can be approximated as

$$\begin{aligned} \dot{x}_e(t) &= A_k x_e(t) + B_{1k} w(t) + B_{2k} u_e(t), \ t \ge t_k \ge t_0 \\ z_e(t) &= C_{1k} x_e(t) + D_{1k} w(t) + D_{2k} u_e(t), \end{aligned} \tag{17}$$

with $\Omega_k := \begin{pmatrix} A_k & B_{1k} & B_{2k} \\ C_{1k} & D_{1k} & D_{2k} \end{pmatrix} = \left. \begin{pmatrix} \frac{\partial f}{\partial x} & \frac{\partial f}{\partial w} & \frac{\partial f}{\partial u} \\ \frac{\partial h_1}{\partial x} & \frac{\partial h_1}{\partial w} & \frac{\partial h_1}{\partial u} \end{pmatrix} \right|_{(x_d(t_k), u_d(t_k), 0)}$. Note that the control inputs in the error system (17) are constrained by $|u_{e,j}(t)| \leq u_{j,\max} - |u_{d,j}(t)|, \forall t \geq t_k,\ j = 1, 2, \cdots, m_2$. We observe that at each fixed sampling time t_k, (17) is an LTI system. This simplifies significantly finding a solution to the optimization problem (9). For this purpose, consider $V(x) := x^T P x$ with a symmetric $P > 0$. It is then easy to show the equivalence of (5) to the following LMI with $Q = P^{-1}$ and $Y = KQ$:

$$\begin{pmatrix} A_k Q + Q A_k^T + B_{2k} Y + Y^T B_{2k}^T & * & * \\ B_{1k}^T & -\gamma^2 I & * \\ C_{1k} Q + D_{2k} Y & D_{1k} & -I \end{pmatrix} < 0. \tag{18}$$

The feasibility of (5) implies that $x_e(t_k)^T P x_e(t_k) \geq \max_{w \in \mathcal{W}} \int_{t_k}^{\infty} \|z_e(t)\|^2 - \gamma^2 \|w(t)\|^2 dt$. Thus, the defined quadratic V provides an upper bound for (4). Hence, we suggest to solve the following LMI optimization problem instead of the minimax problem (9) with the smallest possible γ:

$$\min_{r, \gamma^2, Q = Q^T > 0, Y} q_1 r + q_2 \gamma^2 \quad \text{subject to (18) and} \tag{19a}$$

$$\begin{pmatrix} r & x_e(t_k)^T \\ x_e(t_k) & Q \end{pmatrix} \geq 0,\ r \leq r_c(1+\epsilon), \tag{19b}$$

$$\begin{pmatrix} \frac{u_{jk,\max}^2}{r_c} & e_j^T Y \\ * & Q \end{pmatrix} \geq 0, \quad j = 1, 2, \ldots, m_2, \tag{19c}$$

$$\begin{pmatrix} p_0 - p_{k-1} + x_e(t_k)^T P_{k-1} x_e(t_k) & x_e(t_k)^T \\ x_e(t_k) & Q \end{pmatrix} \geq 0 \tag{19d}$$

for a given $r_c > 0$, where $u_{jk,\max} := \min_{t \in [t_k, t_{k+1})} (u_{j,\max} - |u_{d,j}(t)|)$ and (q_1, q_2) are weights. The LMI (19c) is introduced to guarantee the satisfaction of the control constraints and (19d) is the LMI formulation of the dissipation constraint (7) for the quadratic V considered. We omit the detailed derivation and refer to [CGW06]. Feasibility of the optimization problem (19) at each sampling time is crucial for the implementation. Hence, we provide a condition that sufficiently renders (19) feasible, if the linearized models about the reference belong to a polytope with finite vertices of the form

$$\Omega_k \in Co\left\{ \begin{pmatrix} A_i & B_{1,i} & B_{2,i} \\ C_{1,i} & D_{1,i} & D_{2,i} \end{pmatrix},\ i = 1, 2, \cdots, L \right\}, \forall k = 0, 1, 2, \cdots. \tag{20}$$

Lemma 3.1. *Suppose that*

- *there exists a triple (γ_o, Q_o, Y_o) satisfying (18) for all vertices in (20);*
- *the initial error state $x_e(t_0)$ is bounded in the sense of $\|x_e(t_0)\| < \infty$;*
- *the amplitude of the disturbance is bounded for all $t \geq t_k \geq t_0$.*

Then, the optimization problem (19) is feasible at each $t_k \geq t_0$ for some $r_c > 0$ and $\epsilon \geq 0$.

Proof: By the first assumption, the triple (γ_o, Q_o, Y_o) renders (18) feasible for all $k \geq 0$ and (19d) feasible for all $k \geq 1$. Moreover, at each $t_k \geq t_0$, we can always find $r_c > 0$ satisfying (19c). At time t_0, if $x_e(t_0)$ is bounded, we can define $r_0 := x_e(t_0)^T Q_o^{-1} x_e(t_0)$ satisfying (19b) for some $\epsilon \geq 0$. The feasibility of (18) leads to (5) with the pair $(x_e^T Q_o^{-1} x_e, \gamma_o)$, which implies that $x(1)$ is bounded if the disturbance is bounded in the amplitude. By induction, we can conclude that there exist $r_c > 0$ and $\epsilon \geq 0$ such that $(r_k, \gamma_o, Q_o, Y_o)$ construct a feasible solution to (19) at each $t_k \geq t_0$, where $r_k := x_e(t_k)^T Q_o^{-1} x_e(t_k)$ □

We now give the following moving horizon algorithm for the tracking problem considered, which is computationally tractable:

Step 1. Initialization. Choose r_c and (q_1, q_2).

Step 2. At time t_0. Get $x_e(t_0), u_{j0,\max}$ and Ω_0. Take $\epsilon = 0$ and solve (19) without (19d) to obtain $(r_0, \gamma_0, Q_0, Y_0)$. If the problem is not feasible, increase $\epsilon > 0$. Set $K_0 = Y_0 Q_0^{-1}$, $P_0 = Q_0^{-1}$, $p_0 = V_0(x_e(t_0))$ and go to Step 4.

Step 3. At time $t_k > t_0$. Get $x_e(t_k), u_{jk,\max}$ and Ω_k. Take $\epsilon = 0$ and solve (19) to obtain $(r_k, \gamma_k, Q_k, Y_k)$. If the problem is not feasible, increase $\epsilon > 0$. Set $K_k = Y_k Q_k^{-1}$, $P_k = Q_k^{-1}$ and prepare for the next time instant according to (8).

Step 4. Compute the closed-loop control as

$$u_e(t) = K_k x_e(t), \forall t \in [t_k, t_{k+1}). \tag{21}$$

Replace t_k by t_{k+1} and continue with Step 3.

3.2 Closed-Loop Properties

The above algorithm provides an (almost) optimal solution to the optimization problem (19) at each sampling time $t_k \geq t_0$, denoted by $(r_k, \gamma_k, Q_k, Y_k)$. The closed-loop control is then given by (21) with $K_k = Y_k Q_k^{-1}$. We first discuss the satisfaction of the control constraints (16). The result is obvious: if the following inequality

$$\left| e_j^T \left(K_k x_e(t) + u_d(t) \right) \right| \leq u_{j,\max}, \forall t \in [t_k, t_{k+1}),\ j = 1, 2, \cdots, m_2 \tag{22}$$

is satisfied, then, the control constraints are respected. According to the algorithm, the on-line optimization procedure shapes first the state ellipsoid to meet the control constraints. If it fails, the ellipsoid will be enlarged by some $\epsilon > 0$. After the successful optimization, the satisfaction of the control constraints can be checked by (22). The conservatism involved in the ellipsoid evaluation of the control constraints is to some extent reduced, since the control constraints are in general given in a polytopic form [CGW06]. Hence, we can state the following results according to the discussion in Section 2.

Corollary 1. *The system (17) with the moving horizon control (21) achieves the following properties:*

1. *the disturbance is attenuated in the sense of the $\mathcal{L}_2$ -gain from w to z_e less than $\bar{\gamma}$, where $\bar{\gamma}$ is defined by (14);*
2. *the closed loop is asymptotically stable, if the system (17) is uniformly zero-state detectable and the disturbance has finite energy;*
3. *the closed loop is stable in the Lyapunov sense, if the disturbance is bounded in the amplitude by (15).*

Proof: The proof is an immediate consequence of Theorem 2.1 and Theorem 2.2, considering $V(x) = x^T Px$. □

Note that the above results are for the linearized error system. The complete feedback control law for system (1) is given by $u = \kappa(x) := K_k (x - x_d) + u_d$, where $\kappa(x_d) = u_d$. The dynamics of the tracking error is then given by

$$\begin{aligned} \dot{x}_e(t) &= f_e (x_e(t), w(t)) \ , t \geq t_0 \\ z_e(t) &= h_e (x_e(t), w(t)) \end{aligned} \tag{23}$$

with $f_e(x_e, w) := f(x_e + x_d, \kappa(x_e + x_d), w) - \dot{x}_d$, and $h_e(x_e, w) := h_1(x_e + x_d, \kappa(x_e + x_d), w) - z_d$. Since (z_d, x_d, u_d) satisfies the unperturbed nonlinear system, *i.e.*,, $\dot{x}_d(t) = f(x_d(t), u_d(t), 0)$, $z_d(t) = h_1 (x_d(t), u_d(t), 0)$, the point $(0, 0)$ is then an equilibrium of the nonlinear error system (23). Therefore, we can state the following local $\mathcal{L}_2$ disturbance attenuation property.

Teorema 3.1. *If the system (1) with the performance output z_1 is zero-state detectable, then, the nonlinear moving horizon tracking system (23) achieves disturbance attenuation in the sense of the local $\mathcal{L}_2$ -gain from w to z_e less than $\bar{\gamma}$, where $\bar{\gamma}$ is defined by (14).*

Proof: Due to the zero-state detectability, it follows from the result (2) of Theorem 1 that the linearized system of (23) at the equilibrium $(0, 0)$ is asymptotically stable. Hence, we can apply Corollary 8.3.4 in [Sch00] to conclude that there exists a neighborhood of the equilibrium such that the error system (23) admits an $\mathcal{L}_2$ -gain less than $\bar{\gamma}$. □

4 Example: Reference Tracking of a CSTR

To demonstrate the suggested moving horizon tracking algorithm, we consider a continuous stirred tank reactor, in which the following reactions take place: $A \xrightarrow{k_1} B \xrightarrow{k_2} C$, $2A \xrightarrow{k_3} D$. A more detailed description can be found in [KEK94]. The control objective is to asymptotically track a given reference trajectory despite disturbances: the inflow concentration and inflow temperature. As state variables, we consider the concentrations of the initial reactant A and the product B in the reactor, denoted as c_A and c_B, the temperatures in the reactor and in the cooling jacket, represented by ϑ and ϑ_K. The normalized flow rate q to the reactor and the heat removal power j_Q from the cooling jacket are considered as control inputs, that are assumed to be saturated as

$$3\frac{1}{\mathrm{h}} \leq q \leq 35\frac{1}{\mathrm{h}}, \quad -9000\frac{\mathrm{kJ}}{\mathrm{h}} \leq j_Q \leq 0\frac{\mathrm{kJ}}{\mathrm{h}}\,. \tag{24}$$

Since the main product of the CSTR is the substance B, we choose $z_e = (Hx_e \ Eu_e)^T$ with $H = \mathrm{diag}(0.1, 1, 0.5, 0.1)$ and $E = \mathrm{diag}(0.35, 0.35)$ as performance output. The proposed moving horizon tracking scheme is implemented with a sampling time of $T = 20$s, and the tuning parameters in (19) are chosen as $r_c = 11$, $q_1 = 0.1$ and $q_2 = 1$.

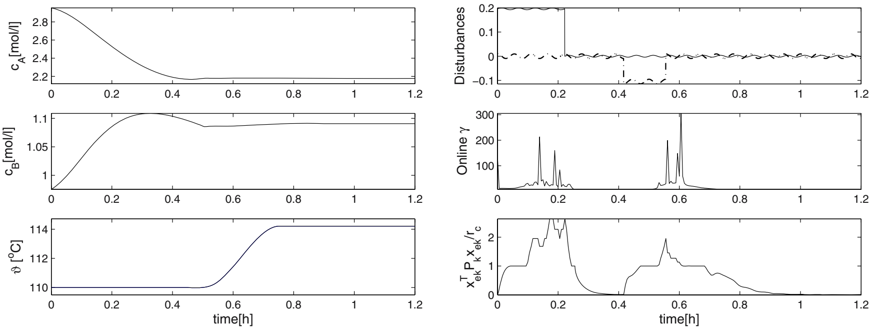


Fig. 1. Reference trajectory (left) and disturbances (right,top): normalized inflow concentration (——) and temperature (—·—); profiles of $\gamma(t)$ and $\frac{x_e(t)^T P x_e(t)}{r_c}$ (right,middle and bottom)

The left plot of Fig. 1 shows a reference trajectory generated by flatness technique [RRZ96], that is consistent with the unperturbed CSTR model and feasible for control constraints to drive the CSTR from the setpoint of $\vartheta = 110^o C$ and $c_B = 0.975\frac{\mathrm{mol}}{\mathrm{l}}$ to the maximal yield point of $\vartheta = 114.07^o\mathrm{C}$ and $c_B = 1.09\frac{\mathrm{mol}}{\mathrm{l}}$. In the presence of disturbances shown in the top right of Fig. 1, the tracking errors are plotted in Fig. 2. As a comparison, we design two fixed $\mathcal{H}_\infty$ controllers by solving the following LMI optimization problem

$$\min_{\gamma^2, Q=Q^T>0, Y} \gamma^2 \ \text{ subject to (18) and (19c)} \tag{25}$$

with $r_c = 11$ (denoted as A) and $r_c = 40$ (denoted as B), where the subscript k in (18) is replaced by i with $i = 1, 2$ corresponding to the linearized models of the CSTR at the two setpoints. We stress that the control actions injected into the CSTR will be clipped if the values exceed the bounds given in (24). This happens for the fixed controller A during the both disturbances (see the dashed lines in Fig. 3), and for the B during the second disturbance (see the dash-dotted lines in Fig. 3). A clear performance degeneration duo to clipping can be seen in Fig. 2. For the suggested moving horizon tracking controller, we summarize the

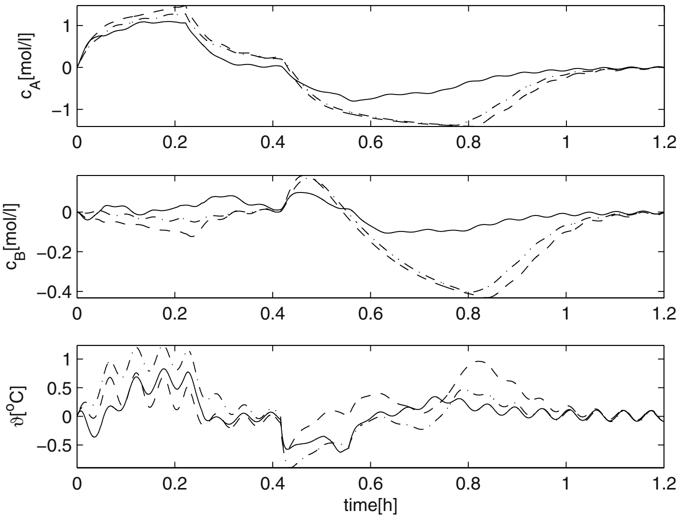


Fig. 2. Tracking errors for moving horizon controller with $r_c = 11$ (—) and fixed controllers with $r_c = 11$ (−−) and $r_c = 40$ (− · −)

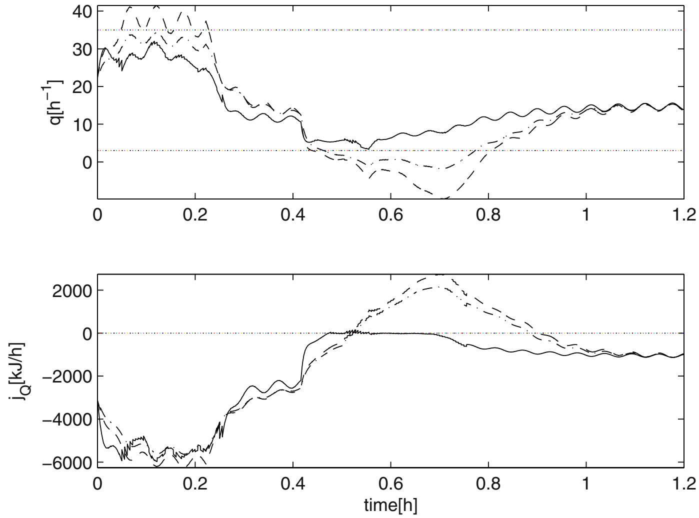


Fig. 3. Control inputs for moving horizon controller with $r_c = 11$ (—) and fixed controllers with $r_c = 11$ (−−) and $r_c = 40$ (− · −)

following points: Control constraints are satisfied; it achieves better performance by avoiding saturation and making the best of the allowable controls; it avoids actuator saturation by relaxing the performance level and recovers a higher performance level when large disturbances inclusive of reference changes vanish. Moreover, we observe the clear and large violation of $x_e(t)^T P_k x_e(t) \leq r_c$ in the bottom right of Fig. 1. This confirms the advantages of the suggested moving horizon tracking algorithm in avoiding infeasibility and reducing conservatism in handling time-domain constraints.

5 Conclusions

In this paper, the disturbance attenuation issue of nonlinear moving horizon control has been addressed. By extending the dissipation constraint in a general non-quadratic form, a conceptual minimax moving horizon formulation is suggested and theoretical results on closed-loop dissipation, $\mathcal{L}_2$ disturbance attenuation and stability are discussed. The implementation issue is attacked with respect to tracking a reference trajectory in the presence of external disturbances and control constraints. A computationally tractable algorithm is given in the framework of LMI optimization and applied to the reference tracking control of the CSTR. As revealed in the title, the results of this paper might be preliminary. Further works are required either to build a rigorous theoretical basis or to achieve non-conservative and computationally tractable algorithms.

Acknowledgement

The work is supported by the National Nature Science Foundation China (No 60374027) and by Program for New Century Excellent Talents in University.

References

[ABQ99] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright, "Nonlinear predictive control and moving horizon estimation - An introductory overview," in *Advances in Control, Highlights of ECC'99* (P. Frank, ed.), pp. 391–449, Springer Verlag, (1999).

[BM99] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in Identification and Control, Lecture Notes in Control and Information Sciences, vol. 245* (A. V. A. Garulli, A. Tesi, ed.), pp. 207–226, Springer-Verlag, (1999).

[MRR00] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, no. 6, pp. 789–814, (2000).

[SM98] P. O. M. Scokaert and D. Q. Mayne, "Min-max feedback model predictive control for constrained linear systems," *IEEE Trans. Automat. Contr.*, vol. 43, no. 8, pp. 1136–1142, (1998).

[BBM01] A. Bemporad, F. Borrelli, and M. Morari, "Robust model predictive control: piecewise linear explicit solution," in *Proc. European Control Conf.*, (Porto, Portugal), pp. 939–944, (2001).

[LK94] S. Lall and K. Glover, "A game theoretic approach to moving horizon control," in *Advances in Model-Based Predictive Control* (D. Clarke, ed.), Oxford University Press, (1994).

[CSA97] H. Chen, C. W. Scherer, and F. Allgöwer, “A game theoretic approach to nonlinear robust receding horizon control of constrained systems,” in *Proc. Amer. Contr. Conf.*, (Albuquerque), pp. 3073–3077, (1997).

[MNS03] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer, “Robust model predictive control for nonlinear discrete-time systems,” *Int. J. of Robust and Nonlinear Control*, vol. 13, no. 3-4, pp. 229–246, (2003).

[BB99] R. Blauwkamp and T. Basar, “A receding-horizon approach to robust output feedback control for nonlinear systems,” in *Proc. 38th IEEE Conf. Decision Contr.*, (Phonex, AZ), pp. 4879–4884, (1999).

[MNS01] L. Magni, H. Nijmeijer, and A. J. van der Schaft, “A receding horizon approach to the nonlinear H_∞ control problem,” *Automatica*, vol. 37, pp. 429–435, (2001).

[Tad92] G. Tadmor, “Receding horizon revisited: An easy way to robustly stabilize an LTV system,” *Syst. Contr. Lett.*, vol. 18, pp. 285–294, (1992).

[MNS02] L. Magni, G. De Nicolao, R. Scattolini, and F. Allgöwer, “Robust receding horizon control for nonlinear discrete-time systems,” in *Proc. 15th IFAC World Congress*, (Barcelona, Spain), (2002).

[Gyu02] E. Gyurkovics, “Receding horizon H_∞ control for nonlinear discrete-time systems,” *Control Theory and Applications, IEE Proceedings*, vol. 149, no. 6, pp. 540 –546, (2002).

[CS03] H. Chen and C. W. Scherer, “Disturbance attenuation with actuator constraints by moving horizon H_∞ control,” in *IFAC ADCHEM’03*, (HongKong, China), pp. 447–452, (2003).

[SCA02] C. W. Scherer, H. Chen, and F. Allgöwer, “Disturbance attenuation with actuator constraints by hybrid state-feedback control,” in *Proc. 41th IEEE Conf. Decision Contr.*, pp. 4134–4139, (2002).

[CS04] H. Chen and C. W. Scherer, “An LMI based model predictive control scheme with guaranteed H_∞ performance and its application to active suspension,” in *Proc. Amer. Contr. Conf.*, (Boston, USA), pp. 1487–1492, (2004).

[Kha92] H. Khalil, *Nonlinear Systems.* New York: Macmillan Publishing Company, (1992).

[CS05] H. Chen and C. W. Scherer, “Moving horizon H_∞ control with performance adaptation for constrained linear systems.” accepted for Automatica, (2005).

[CGW06] H. Chen, X.-Q. Gao, and H. Wang, “An improved moving horizon H_∞ control scheme through Lagrange duality.” *Int. J. Control*, vol. 79, no. 3, pp. 239–248, (2006).

[Sch00] A. van der Schaft, *L_2-Gain and Passivity Techniques in Nonlinear Control.* London: Springer Verlag, (2000).

[KEK94] K.-U. Klatt, S. Engell, A. Kremling, and F. Allgöwer, “Testbeispiel: Rührkesselreaktor mit Parallel- und Folgereaktion,” in *Entwurf Nichtlinearer Regelungen* (S. Engell, ed.), pp. 425–432, München: Oldenbourg Verlag, (1995).

[RRZ96] R. Rothfuss, J. Rudolph, and M. Zeitz, “Flateness based control of a nonlinear chemical reactor,” *Automatica*, vol. 32, no. 10, pp. 1433–1439, (1996).

Chance Constrained Nonlinear Model Predictive Control

Lei Xie[1,3], Pu Li[2], and Günter Wozny[3]

[1] State Key Laboratory of Industrial Control Technology, Institute of Advanced Process Control, Zhejiang University, Hangzhou 310027, China
LeiX@iipc.zju.edu.cn

[2] Institute of Automation and Systems Engineering, Ilmenau University of Technology, P.O. Box 100565, Ilmenau 98684 Germany
Pu.Li@tu-ilmenau.de

[3] Department of Process Dynamics and Operation, Berlin University of Technology, Sekr. KWT 9, Berlin 10623, Germany
Guenter.Wozny@tu-berlin.de

Summary. A novel robust controller, chance constrained nonlinear MPC, is presented. Time-dependent uncertain variables are considered and described with piecewise stochastic variables over the prediction horizon. Restrictions are satisfied with a user-defined probability level. To compute the probability and its derivatives of satisfying process restrictions, the inverse mapping approach is extended to dynamic chance constrained optimization cases. A step of probability maximization is used to address the feasibility problem. A mixing process with both an uncertain inflow rate and an uncertain feed concentration is investigated to demonstrate the effectiveness of the proposed control strategy.

1 Introduction

Model predictive control (MPC) refers to a family of control algorithms which utilize an explicit model to calculate the manipulated variables that optimize the future plant behaviour. The inherent advantages of MPC, including its capability of dealing with multivariate variable problems as well as its capability of handling constraints, make it widely used in the process industry.

Due to the nature of process uncertainty, a robust MPC is desired to obtain satisfactory control performances. Including uncertainty in control system design will enhance the robustness of MPC. Generally speaking, there are three basic approaches to address uncertainty. The constant approach which assumes the model mismatch is unchanged during the prediction horizon [1] leads to an aggressive control strategy. In contrary, the Min-Max approach in which the boundaries of the uncertain variables are taken into account [2] is too conservative. The third one is the stochastic approach, or chance constrained MPC [3], [4], in which uncertain variables in the prediction horizon are described as stochastic variables with known probability distribution functions (PDF). Restrictions are to be satisfied with a user-defined probability level. Due to the fact that using this method a desired compromise between the optimal function

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 295–304, 2007.
springerlink.com

and the reliability of holding the constraints can be chosen, the derived control strategy can be neither aggressive nor conservative.

Linear chance constrained MPC have been previously studied in ref [10]. In the present study, we extend this approach to nonlinear systems. The major obstacle towards realizing chance constrained nonlinear MPC (CNMPC) lies in the computation of the probability and its deviations of satisfying process restrictions. To address this problem, an inverse mapping approach proposed by Wendt *et al.* [5] is extended to dynamic chance constrained optimization. In addition, a step of maximization is proposed to address the feasibility problem of CNMPC.

The paper is divided into the following sections. Section 2 gives a general formulation of CNMPC considering both parameter and disturbance uncertainties. Section 3 analyzes some computational aspects of CNMPC. The effectiveness of CNMPC is illustrated in Section 4 by controlling a mixing process. Finally, some concluding remarks of this work are given in Section 5.

2 Chance Constrained Nonlinear MPC

It has been recognized that problems in process system engineering (PSE) are almost all confronted with uncertainties [7], [13]. In the industrial practice, uncertainties are usually compensated by using conservative design as well as conservative operating strategies, which may lead to considerably more costs than necessary. To overcome this drawback, the authors have recently developed a chance constrained programming (CCP) framework for process optimization and control [3], [5], [10], [11], [12]. In this framework, the uncertainty properties, obtained from the statistical analysis of historical data, are included in the problem formulation explicitly.

Chance constrained nonlinear MPC (CNMPC) employs a nonlinear model to predict future outputs, based on the current states, past controls as well as uncertain variables. The optimal control sequence is obtained at every sampling instant by optimizing some objective functions and ensuring the chance constraints for the outputs.

The general CNMPC problem to be solved at sampling time k is formulated as follows:

$$\begin{aligned}
&\text{Min}\quad J = E\{f\} + \omega D\{f\}\\
&s.t.\\
&f = \sum_{i=1}^{P} \|\mathbf{y}(k+i|k) - \mathbf{y}_{ref}\|_{\mathbf{Q}_i}\\
&\qquad\qquad + \sum_{i=0}^{M-1} \{\|\mathbf{u}(k+i|k) - \mathbf{u}_{ref}\|_{\mathbf{R}_i} + \|\Delta\mathbf{u}(k+i|k)\|_{\mathbf{S}_i}\} \qquad (1)\\
&\mathbf{x}(k+i+1|k) = \mathbf{g}_1(\mathbf{x}(k+i|k), \mathbf{u}(k+i|k), \xi(k+i))\\
&\mathbf{y}(k+i|k) = \mathbf{g}_2(\mathbf{x}(k+i|k), \xi(k+i))
\end{aligned}$$

$$\begin{aligned}
&\Delta\mathbf{u}(k+i|k) = \mathbf{u}(k+i|k) - \mathbf{u}(k+i-1|k)\\
&\mathbf{u}_{\min} \leq \mathbf{u}(k+i|k) \leq \mathbf{u}_{\max}, i = 0, \ldots, M-1.\\
&\Delta\mathbf{u}_{\min} \leq \Delta\mathbf{u}(k+i|k) \leq \Delta\mathbf{u}_{\max}, i = 0, \ldots, M-1.\\
&\mathcal{P}\{\mathbf{y}_{\min} \leq \mathbf{y}(k+i|k) \leq \mathbf{y}_{\max}\} \geq \alpha, i = 1, \ldots, P.
\end{aligned}$$

where P and M are the length of prediction and control horizon, ξ represents the uncertain variables with known PDF, $\mathcal{P}\{\cdot\}$ represents the probability to satisfy the constraint $\mathbf{y}_{\min} \leq \mathbf{y}(k+i|k) \leq \mathbf{y}_{\max}$ and $0 \leq \alpha \leq 1$ is the predefined confidence level. States $\mathbf{x}$, outputs $\mathbf{y}$ and controls $\mathbf{u}$ are all doubly indexed to indicate values at time $k+i$ given information up to and including time k. Q_i, R_i, and S_i are weighting matrices in the objective function. E and D are the operators of expectation and variation, respectively.

Since the outputs have been confined in the chance constraints, the objective function f in Eq.(1) may exclude the quadratic terms on outputs for the sake of simplicity [10]. The simplified CNMPC objective function can be described as follows:

$$\text{Min} \quad J = \sum_{i=1}^{M-1} \{\|\mathbf{u}(k+i|k) - \mathbf{u}_{ref}\|_{\mathbf{R}_i} + \|\Delta\mathbf{u}(k+i|k)\|_{\mathbf{S}_i}\} \tag{2}$$

This problem can be solved by using a nonlinear programming algorithm. The key obstacle towards solving the CNMPC problem is how to compute $\mathcal{P}\{\cdot\}$ and its gradient with respect to the controls. In the next section, the computational aspects of CNMPC to address this problem as well as the feasibility analysis will be discussed.

3 Computational Aspects of CNMPC

In process engineering practice, uncertain variables are usually assumed to be normally distributed due to the central limit theory. However, a normal distribution means that the uncertain variable is boundless, which is not true for some parameters with physical meanings, e.g. the molar concentration in a flow should be in the range of [0, 1]. In order to describe the physical limits of the uncertainty parameters, it is preferable to employ truncated normal distribution which has been used extensively in the fields of economic theory [9]. The basic definition of truncated normal distribution is given as follows:

Definition 1. *Let z be a normally distributed random variable with the following PDF:*

$$\rho(z) = \frac{1}{\sigma\sqrt{2\pi}} \exp\{-\frac{(z-\mu)^2}{2\sigma^2}\} \tag{3}$$

Then the PDF of ξ, the truncated version of z on $[a_1, a_2]$ is given by:

$$\rho(\xi) = \begin{cases} \frac{1}{\sigma\sqrt{2\pi}(\Phi(a_2)-\Phi(a_1))} \exp\{-\frac{(\xi-\mu)^2}{2\sigma^2}\}, \ a_1 \leq \xi \leq a_2 \\ 0, \xi \leq a_1 \ or \ a_2 \leq \xi \end{cases} \tag{4}$$

where $\Phi(\cdot)$is the cumulative distribution function of z.

Detailed discussion about the properties of the truncated normal distribution can be found in [9] and it is easy to extend Definition 1 to the multivariate case. In the following, a truncated normally distributed ξ with mean μ, covariance matrix $\mathbf{\Sigma}$ and truncated points $\mathbf{a}_1$, $\mathbf{a}_2$, denoted as $\xi \sim TN(\mu, \mathbf{\Sigma}, \mathbf{a}_1, \mathbf{a}_2)$, is considered.

3.1 Inverse Mapping Approach to Compute the Probability and Gradient

If the joint PDF of the output $\mathbf{y}(k+i|k)$ is available, the calculation of $\mathcal{P}\{\mathbf{y}_{\min} \leq \mathbf{y}(k+i|k) \leq \mathbf{y}_{\max}\}$ and its gradient to $\mathbf{u}$ can be cast as a standard multivariate integration problem [8]. But unfortunately, depending on the form of $\mathbf{g}_2$, the explicit form of the output PDF is not always avaliable. To avoid directly using the output PDF, an inverse mapping method has been recently proposed for situations in which the monotone relation exists between the output and one of the uncertain variables [5].

Without loss of generality, let $y = F(\xi_S)$ denotes the monotone relation between a single output y and one of the uncertain variables ξ_S in ξ=[ξ_1, $\xi_{2,}\ldots$, $\xi_S]^T$. Due to the monotony, a point between the interval of $[y_{min}, y_{max}]$ can be inversely mapped to a unique ξ_S through $\xi_S = F^{-1}(y)$:

$$\mathcal{P}\{y_{\min} \leq y \leq y_{\max}\} \Leftrightarrow \mathcal{P}\{\xi_S^{\min} \leq \xi_S \leq \xi_S^{\max}\} \tag{5}$$

It should be noted that the bounds $\xi_S^{\min}$, $\xi_S^{\max}$ depends on the realization of the individual uncertain variables ξ_i, $(i = 1, \cdots, S-1)$ and the value of input u, i.e.

$$[\xi_S^{\min}, \xi_S^{\max}] = F^{-1}(\xi_1, \cdots, \xi_{S-1}, y_{\min}, y_{\max}, u) \tag{6}$$

and this leads to the following representation

$$\mathcal{P}\{y_{\min} \leq y \leq y_{\max}\} = \int\limits_{-\infty}^{\infty} \cdots \int\limits_{-\infty}^{\infty} \int\limits_{\xi_S^{\min}}^{\xi_S^{\max}} \rho(\xi_1, \cdots, \xi_{S-1}, \xi_S) d\xi_S d\xi_{S-1} \cdots d\xi_1 \tag{7}$$

From (6) and (7), u has the impact on the integration bound of ξ_S. Thus the following equation can be used to compute the gradient of $\mathcal{P}\{y_{\min} \leq y \leq y_{\max}\}$ with respect to the control variable u:

$$\begin{aligned} \tfrac{\partial \mathcal{P}\{y_{\min} \leq y \leq y_{\max}\}}{\partial u} = \int\limits_{-\infty}^{\infty} \cdots \int\limits_{-\infty}^{\infty} \{\rho(\xi_1, \cdots, \xi_{S-1}, \xi_S^{\max}) \frac{\partial \xi_S^{\max}}{\partial u} - \\ \rho(\xi_1, \cdots, \xi_{S-1}, \xi_S^{\min}) \tfrac{\partial \xi_S^{\min}}{\partial u} \} d\xi_{S-1} \cdots d\xi_1 \end{aligned} \tag{8}$$

A numerical integration of (7) is required when taking a joint distribution function of ξ into account. Note that the integration bound of the last variable in (7) is not fixed. A novel iterative method based on the orthogonal collocation on finite elements was proposed in ref [5] to accomplish the numerical integration in the unfixed-bounded region.

Extending inverse mapping to the dynamic case. If a monotone relation also exists between $\mathbf{y}(k+i|k)$ and $\xi(k+i)$ for $i = 1, \ldots, P$ in $\mathbf{g}_2$ of Eq.(1), the inverse mapping method is readily extended to obtain the value and the gradient of $\mathcal{P}\{\mathbf{y}_{\min} \leq \mathbf{y}(k+i|k) \leq \mathbf{y}_{\max}\}$. For the sake of simplifying notations, a SISO system is considered in the present study, and it is not difficult to generalize the following conclusions. With the monotone relation between $y(k+i|k)$ and $\xi(k+i)$, we have

$$y_{\min} \leq y(k+i|k) \leq y_{\max} \Leftrightarrow \xi^{\min}_{k+i} \leq \xi(k+i) \leq \xi^{\max}_{k+i} \tag{9}$$

Due to the propagation of the uncertainty through the dynamic system, $y(k+i|k)$ is influenced not only by $\xi(k+i)$, $u(k+i-1|k)$, but also by previous $\xi(k)$ to $\xi(k+i-1)$ and $u(k|k)$ to $u(k+i-2|k)$. Therefore, the bounds $\xi^{\min}_{k+i}$ and $\xi^{\max}_{k+i}$ are determined based on the realization of the uncertain variables and controls from the time interval k to $k+i$, namely,

$$[\xi^{\min}_{k+i}, \xi^{\max}_{k+i}] = F^{-1}(\xi(k+i-1), \cdots, \xi(k), u(k+i-1|k), \cdots, u(k|k), y_{\min}, y_{\max}) \tag{10}$$

So the joint outputs chance constraint over the prediction horizon can be reformulated as

$$\begin{aligned}
&\mathcal{P}\{y_{\min} \leq y(k+i|k) \leq y_{\max}, i = 1, 2, \ldots, P\} \\
&= \mathcal{P}\{\xi^{\min}_{k+i} \leq \xi(k+i) \leq \xi^{\max}_{k+i}, i = 1, 2, \ldots, P\} \\
&= \int_{-\infty}^{\infty} \int_{\xi^{\min}_{k+1}}^{\xi^{\max}_{k+1}} \cdots \int_{\xi^{\min}_{k+P}}^{\xi^{\max}_{k+P}} \rho(\xi(k), \xi(k+1) \cdots, \xi(k+P)) d\xi(k+P) \cdots d\xi(k+1) d\xi(k)
\end{aligned} \tag{11}$$

where ρ is the joint PDF of the future uncertain variables.

The gradient computation of $\mathcal{P}\{\cdot\}$ is more complicated due to the complex relation between the integration bounds and the controls. With the assumption of a same control and prediction horizon, M=P, the gradient with respect to $u(k+i|k)$ can be determined as follows

$$\begin{aligned}
&\partial \mathcal{P}\{y_{\min} \leq y(k+i|k) \leq y_{\max}, i = 1, 2, \ldots, P\} / \partial u(k+i|k) = \\
&\sum_{j=i+1}^{P} \{ \int_{-\infty}^{\infty} \int_{\xi^{\min}_{k+1}}^{\xi^{\max}_{k+1}} \cdots \int_{\xi^{\min}_{k+j-1}}^{\xi^{\max}_{k+j-1}} \\
&\{ \frac{\partial \xi^{\max}_{k+j}}{\partial u(k+i|k)} \int_{\xi^{\min}_{k+j+1}}^{\xi^{\max}_{k+j+1}} \cdots \int_{\xi^{\min}_{k+P}}^{\xi^{\max}_{k+P}} \rho(\xi(k), \cdots, \xi^{\max}_{k+j}, \cdots \xi(k+P)) d\xi(k+P) \cdots d\xi(k+j+1) \\
&- \frac{\partial \xi^{\min}_{k+j}}{\partial u(k+i|k)} \int_{\xi^{\min}_{k+j+1}}^{\xi^{\max}_{k+j+1}} \cdots \int_{\xi^{\min}_{k+P}}^{\xi^{\max}_{k+P}} \rho(\xi(k), \cdots, \xi^{\min}_{k+j}, \cdots \xi(k+P)) d\xi(k+P) \cdots d\xi(k+j+1)\} \\
&d\xi(k+j-1) \cdots d\xi(k+1) d\xi(k)\}
\end{aligned} \tag{12}$$

Note that if the predictive horizon length P is too large, the integration in (12) will lead to considerable computing time. Thus a value of P less than 10 is suggested in practice.

3.2 Feasibility Analysis

Feasibility analysis concerns the problem of whether the chance constraint $\mathcal{P}\{\mathbf{y}_{\min} \leq \mathbf{y}(k+i|k) \leq \mathbf{y}_{\max}\} \geq \alpha$ is feasible. This is an important issue for the chance constrained problems, since it is likely that the predefined level α is higher than reachable. In this case the optimization routine can not find a feasible solution. A straightforward way to address this problem is to compute the maximum reachable probability before doing the optimization. As a result, the original objective function in (1) will be replaced with

$$\text{Max}\,\mathcal{P}\{y_{\min} \leq y(k+i|k) \leq y_{\max}, i=1,2,\ldots,P\} \tag{13}$$

The maximum reachable α can be obtained by solving the corresponding optimization problem.

4 Application to a Mixing Process

The discretized model of the tank mixing process under study with unit sampling time interval is:

$$\begin{aligned} V(k+1) &= V(k) + q(k) - u(k) \\ C(k+1) &= C(k) + \tfrac{q(k)}{V(k+1)}[C_0(k) - C(k)] \end{aligned} \tag{14}$$

where V and C are the volume and product mass concentration in the tank, q, u are the feed and outlet flow rates and C_0 is the feed mass concentration, respectively. The control objective is, under the inlet uncertain flow rate and composition, to obtain a possibly flat outlet flow rate while holding the outlet concentration and tank volume in specified intervals. Based on (14), the future process outputs are predicted as

$$V(k+i|k) = V(k) + \sum_{j=0}^{i-1}(q(k+j) - u(k+j|k)) \tag{15}$$

$$\begin{aligned} C(k+i|k) = &\prod_{j=0}^{i-1} \tfrac{V(k+j|k)-u(k+j|k)}{V(k+j+1|k)} C(k) \\ &+ \sum_{j=0}^{i-1} \Big(\prod_{s=j+1}^{i-1} \tfrac{V(k+s|k)-u(k+s|k)}{V(k+s+1|k)} \Big) \tfrac{q(k+j)}{V(k+j+1|k)} C_0(k+j) \end{aligned} \tag{16}$$

With the above prediction model, the nonlinear CNMPC problem at sampling instant k can be formulated as:

$$\begin{aligned} &\text{Min} \;\; \varDelta\mathbf{u}^T \varDelta\mathbf{u} \\ &s.t. \\ &(15) \text{ and } (16) \\ &u_{\min} \leq u(k+i|k) \leq u_{\max}, i = 0,\ldots,M-1. \\ &\mathcal{P}\{V_{\min} \leq V(k+i|k) \leq V_{\max}, i=1,\ldots,P\} \geq \alpha_1 \\ &\mathcal{P}\{C_{\min} \leq C(k+i|k) \leq C_{\max}, i=1,\ldots,P\} \geq \alpha_2 \end{aligned} \tag{17}$$

Li *et al.* [3] studied the linear case which only concerns the volume constraints and the outlet flow rate u only affects the mean value of the output V. In contrast, for the nonlinear model in (16), u affects both the mean and covariance of the distribution of the outlet concentration C.

With the assumption that the feed flow rate $q(k+i)$ and feed concentration $C_0(k+i)$ follow a positive truncated normal distribution, namely, the low truncating point a_1 in (4) is positive, the following monotone relation can be found

$$\begin{aligned} &q(k+i) \uparrow \Rightarrow V(k+i+1|k) \uparrow \\ &C_0(k+i) \uparrow \Rightarrow C(k+i+1|k) \uparrow \end{aligned} \tag{18}$$

Thus the chance constraints in (17) can be transformed into

$$\begin{aligned} &\mathcal{P}\{V_{\min} \le V(k+i|k) \le V_{\max}, i=1,\ldots,P\} \\ &\Rightarrow \mathcal{P}\{q_{k+i-1}^{\min} \le q(k+i-1) \le q_{k+i-1}^{\max}, i=1,\ldots,P\} \\ &\mathcal{P}\{C_{\min} \le C(k+i|k) \le C_{\max}, i=1,\ldots,P\} \\ &\Rightarrow \mathcal{P}\{C_{0(k+i-1)}^{\min} \le C_0(k+i-1) \le C_{0(k+i-1)}^{\max}, i=1,\ldots,P\} \end{aligned} \tag{19}$$

Therefore (11) and (12) can be used to compute $\mathcal{P}\{\cdot\}$ and its gradient.

The proposed CNMPC controller is applied to the mixing process. The initial values of tank volume and product concentration are $V(0) = 160\ l$ and $C(0) = 50g/l$, respectively. The inlet flow $q(k)$ and concentration $C_0(k)$ are assumed to be multivariate truncated normal sequences with the truncating intervals of [0, 20] and [46, 56]. The mean profiles of $q(k)$ and $C_0(k)$ within a period of 20 minutes are shown in Fig.1 and 2. In each time interval, they have the stand deviation values of 0.70 and 1.0. In addition, both $q(k)$ and $C_0(k)$ at different intervals are assumed to be independent. The dashed lines in Fig.1 and 2 are 10 realizations of the disturbance from random samples and it is shown that the uncertainty is considerable. The prediction and control horizon of CNMPC is fixed at $P = M = 5$ and the lower and upper bounds of the output variables, V and C, are [130, 170] and [49, 51], respectively. The probability level of α_1 and α_2 are both given as 0.9. The control results are illustrated in Fig.3 to Fig.6. In Fig.3, it can be seen that the control variable $u(k)$ is more flat than the inlet and thus the disturbance to the downstream unit is thus decreased. As shown in

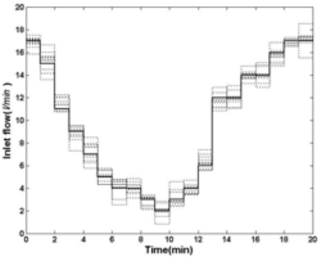


Fig. 1. Inlet flow disturbance profile

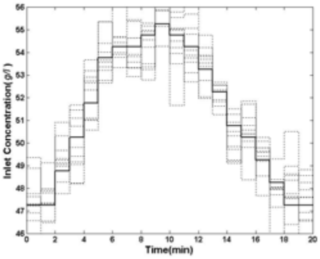


Fig. 2. Inlet concentration disturbance profile

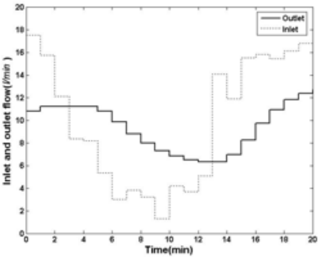


Fig. 3. Inlet $q(k)$ and outlet flow $u(k)$

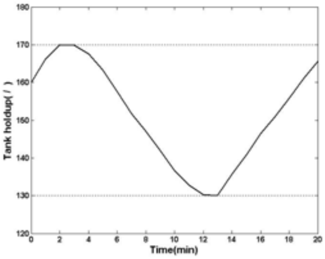


Fig. 4. Tank volume $V(k)$

Fig.4 and Fig.5, the tank volume $V(k)$ and outlet product concentration $C(k)$ are strictly restricted in the predefined bounds. In addition, oscillations also occur in the controlled variables profiles, which means that the controller takes the advantage of the freedom available to keep the control action as flat as possible. The feasibility analysis of production concentration chance constraint is also performed and the maximum reachable possibility of $\mathcal{P}\{C_{\min} \le C(k+i|k) \le C_{\max}\}$ in each interval is depicted in Fig.6. It can be seen that the maximum reachable probabilities are all greater than the predefined value (α_2=0.90), which implies that the corresponding CNMPC problem is feasible. Note that at the 4^{th} and 13^{th} minute when the concentration approaches its limits, the maximum probability reaches its minimum value.

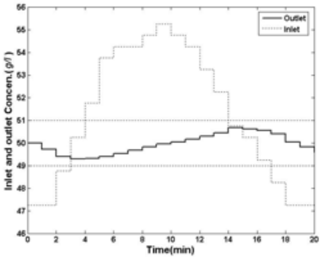


Fig. 5. Inlet and outlet concentration $C(k)$

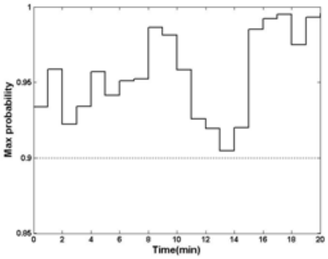


Fig. 6. Max reachable probabilities of the $C(k)$ chance constraint

5 Conclusions

In this work, a robust nonlinear model predictive controller, chance constrained NMPC, is proposed. To compute the probability and derivatives of holding inequality constraints inverse mapping approach is extended to dynamic nonlinear situations. To find a monotone relation between the uncertain variable and the output which is necessary for the inverse mapping approach, truncated normal distribution is considered to describe uncertainty variables. CNMPC is illustrated to be effective by controlling a mixing process with both uncertain feed flow rate and feed concentration.

Acknowledgements. Lei Xie is grateful for the financial support through the Sandwich Program from the Berlin University of Technology. Financial Support from the Deutsche Forschungsgemeischaft (DFG) under the contract LI 806/8-1 is also gratefully acknowledged.

References

[1] Mayne D Q, Rawlings J B, Rao C V, Scokaert P O M (2000) Constrained model predictive control: Stability and optimality. Automatica 36:789-814

[2] Lee J H, Yu Z (1997) Worst-case formulations of model predictive control for systems with bounded parameters. Automatica 33:763-781

[3] Li P, Wendt M, Wozny G (2002) A probabilistically constrained model predictive controller. Automatica 38:1171-1176

[4] Schwarm A T, Nikolaou M (1999) Chance-constrained model predictive control. AIChE Journal 45:1743-1752

[5] Wendt M, Li P, Wozny G (2002) Nonlinear chance-constrained process optimization under uncertainty. Ind. Eng. Chem. Res. 41:3621-3629

[6] Birge J R, Louveaux F (1997) Introduction to Stochastic Programming. Springer, Berlin.

[7] Biegler L T, Grossmann I E (2004) Retrospective on optimization. Computers and Chemical Engineering 28:1169-1192

[8] Diwekar U M, Kalagnanam J R (1997) Efficient sampling technique for optimization under uncertainty. AIChE Journal 43:440-447

[9] Horrace W C (2005) Some results on the multivariate truncated normal distribution. Journal of Multivariate Analysis 94:209-221

[10] Li P, Wendt M, Wozny G (2000) Robust model predictive control under chance constraints. Computers and Chemical Engineering 24:829-834

[11] Li P, Wendt M, Arellano-Carcia H, Wozny G (2002) Optimal operation of distillation process under uncertain inflows accumulated in a feed tank. AIChE. Journal 48:1198-1211

[12] Li P, Wendt M, Wozny G (2004) Optimal production planning for chemical processes under uncertain market conditions. Chemical Engineering Technology 27:641-651

[13] Sahinidis N V (2004) Optimization under uncertainty: state-of-the-art and opportunities. Computers and Chemical Engineering 28: 971-983

Close-Loop Stochastic Dynamic Optimization Under Probabilistic Output-Constraints

Harvey Arellano-Garcia, Moritz Wendt, Tilman Barz, and Guenter Wozny

Department of Process Dynamics and Operation, Berlin University of Technology, Germany
arellano-garcia@tu-berlin.de

Summary. In this work, two methods based on a nonlinear MPC scheme are proposed to solve close-loop stochastic dynamic optimization problems assuring both robustness and feasibility with respect to output constraints. The main concept lies in the consideration of unknown and unexpected disturbances in advance. The first one is a novel *deterministic* approach based on the *wait-and-see* strategy. The key idea is here to anticipate violation of output hard-constraints, which are strongly affected by instantaneous disturbances, by backing off of their bounds along the moving horizon. The second method is a new *stochastic* approach to solving nonlinear chance-constrained dynamic optimization problems under uncertainties. The key aspect is the explicit consideration of the stochastic properties of both exogenous and endogenous uncertainties in the problem formulation (*here-and-now* strategy). The approach considers a *nonlinear* relation between the uncertain input and the constrained output variables.

1 Introduction

Due to its ability to directly include constraints in the computation of the control moves, nonlinear model predictive control offers advantages for the optimal operation of transient chemical plants. Previous works on robust MPC have focused on output constrained problems under model parameter uncertainty, in particular, worst-case performance analysis over a specified uncertainty range [4, 8]. The drawback of this worst-case formulation – min-max approach – is that the resulting control strategy will be overly conservative. In this work, we extended our previous work in [3, 5, 7] to a new chance-constrained optimization approach for NMPC. Unlike the linear case, for nonlinear (dynamic) processes the controls have also an impact on the covariane of the outputs. The new approach also involves efficient algorithms so as to compute the probabilities and, simultaneously, the gradients through integration by collocation in finite elements. However, in contrast to all our previous works (see e.g. [7]), the main novelty here is also that the chance-constrained approach is now also applicable for those cases where a monotic relationship between constrained output and uncertain input can not be asurred. In addition, due to the consideration of a first principle model with a several number of uncertain variables, the problem can conditionally become too computationally intensive for an online

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 305–315, 2007.
springerlink.com

application. Thus, we propose alternatively a dynamic adaptive back-off strategy for a NMPC scheme embedded in an online re-optimization framework. The performance of both proposed approaches is assessed via application to a runaway-safe semi-batch reactor under safety constraints.

2 Problem Formulation

A strongly exothermic series reaction conducted in a non-isothermal fed-batch reactor is considered. The reaction kinetics are second-order for the first reaction producing B from A, and an undesirable consecutive first-order reaction converting B to C. The intermediate product B is the desired product.

$$2A \xrightarrow{K_1} B \xrightarrow{K_2} C \tag{1}$$

A detailed first-principles model of the process is given by a set of DAEs based on mass balances:

$$\begin{aligned} \dot{n}_A &= -\nu_a k_{01} \frac{n_A^2}{V} e^{-\frac{E_{A1}}{\mathbb{R}T}} + feed; \quad \dot{n}_B = -k_{02}\, n_B\, e^{-\frac{E_{A2}}{\mathbb{R}T}} + k_{01} \frac{n_A^2}{V} e^{-\frac{E_{A1}}{\mathbb{R}T}}; \\ \dot{n}_C &= +k_{02}\, n_B\, e^{-\frac{E_{A2}}{\mathbb{R}T}}, \end{aligned} \tag{2}$$

the energy balance:

$$\dot{\bar{T}}_{cool} = \frac{\dot{V}_{cool} \cdot \rho_{cool} \cdot c_{p,cool} \cdot (T_{cool,in} - \bar{T}_{cool}) - \dot{Q}_{cool}^{HT}}{V_{cool} \cdot \rho_{cool} \cdot c_{p,cool}}; \; \dot{T} = \frac{\dot{Q}_{reac} + \dot{Q}_{feed} + \dot{Q}_{cool}}{n_S \sum_i (c_{pi} x_i)} \tag{3}$$

and constitutive algebraic equations:

$$\begin{aligned} \dot{Q}_{reac} &= -\sum (h_i \dot{n}_i) = (h_{0A} + c_{pA}(T - T_0))\dot{n}_A + (h_{0B} + c_{pB}(T - T_0))\dot{n}_B \\ &\quad + (h_{0C} + c_{pC}(T - T_0))\dot{n}_C \\ \dot{Q}_{feed} &= (h_{0A} + c_{pA}(T - T_0)) \cdot feed \\ \dot{Q}_{cool}^{HT} &= -k_{HT} A(T - \bar{T}_{cool}) = -k_{HT}(0.25\pi d^2 + 4Vd^{-1})(T - \bar{T}_{cool}) \\ n_S &= n_A + n_B + n_C; \quad n_S \sum_i (c_{pi} x_i) = c_{pA} n_A + c_{pB} n_B + c_{pC} n_C \\ V &= \frac{n_A \tilde{M}_A + n_B \tilde{M}_B + n_C \tilde{M}_C}{n_A \bar{\tilde{\rho}}_A + n_B \bar{\tilde{\rho}}_B + n_C \bar{\tilde{\rho}}_C} n_S. \end{aligned} \tag{4}$$

In these equations V denotes the varying volume, n_i the molar amount of component i, T, T_F, $\bar{T}_{cool}$, T_{cool}, the reactor, dosing, jacket and cooling medium temperatures, respectively. h_{0i} are the specific standard enthalpies, k_{HT} the heat transfer coefficient, d the scaled reactor diameter, A the heat exchange surface, $\tilde{M}_i$ molecular weights, ρ_i densities and c_{pi} are heat capacities. Besides, since the heat removal is limited, the temperature is controlled by the feed rate of the reactant $A(feed)$, and the flow rate of the cooling liquid $\dot{V}_{cool}$ in the nominal operation. The reactor is equipped with a jacket cooling system.

The developed model considers both the reactor and the cooling jacket energy balance. Thus, the dynamic performance between the cooling medium flow rate as manipulated variable and the controlled reactor temperature is also included in the model equations. The open-loop optimal control is solved first for the successive optimization with moving horizons involved in NMPC. The objective function is to maximize the production of B at the end of the batch CB_f while minimizing the total batch time t_f with $\beta = 1/70$:

$$\min_{\Delta t,\, \dot{V}_{cool},\, feed} (-CB_f + \beta \cdot t_f) \tag{5}$$

subject to the equality constraints (process model equations (2) – (4)) as well as path and end point constraints. First, a limited available amount of A to be converted by the final time is fixed to $\int_{t_0=0}^{t_f} n_A(t)dt = 500mol$. Furthermore, so as to consider the shut-down operation, the reactor temperature at the final batch time must not exceed a limit $(T(t_f) \leq 303\,K)$. There are also path constraints for the maximal reactor temperature and the adiabatic end temperature T_{ad}. The latter is used to determine the temperature after failure. This is a safety restriction to ensure that even in the exterme case of a total cooling failure no runaway will occur $(T(t) \leq 356\,K; \quad T_{ad}(t) \leq 500\,K)$ [1]. Additionally, the cooling flow rate changes from interval to interval are restricted to an upper bound: $\left\|\dot{V}_{cool}(t+1) - \dot{V}_{cool}(t)\right\| \leq 0.05$. The decision variables are the feed flow rate into the reactor, the cooling flow rate, and the length of the different time intervals. A multiple time-scale strategy based on the orthogonal collocation method on finite elements is applied for both discretization and implementation of the optimal policies according to the controller's discrete time intervals (6 – 12 s; 600 – 700 intervals). The resulting trajectories of the reactor temperature and the adiabatic end temperature (safety constraint) for which constraints have been formulated are depicted in Fig. 1. It can be observed that during a large part of the batch time both states variables evolve along their upper limits i.e. the constraints are active. The safety constraint (adiabatic end temperature), in particular, is an active constraint over a large time period (Fig. 1 right). Although operation at this nominal optimum is desired, it typically cannot be ahieved with simultaneous satisfaction of all contraints due to the influence of uncertainties and/or external disturbances. However, the safety hard-constraints should not be violated at any time point.

3 Dynamik Adaptive Back–Off Strategy

Based on the open-loop optimal control trajectories of the critical state variables, in this section, a deterministic NMPC scheme for the online optimization of the fed-batch process is proposed. Furthermore, the momentary criteria on the restricted controller horizon with regard to the entire batch operation is however insufficient. Thus, the original objective of the nominal open-loop optimization

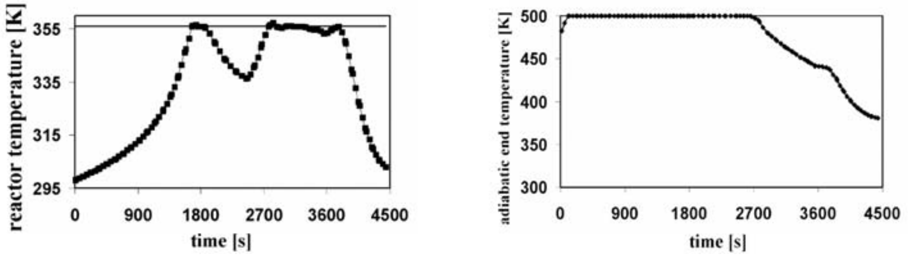


Fig. 1. Path constraints: Optimal reactor temperature (left) and adiabatic end temperature (right)

problem is substituted by a tracking function which can then be evaluated on the local NMPC prediction horizon:

$$\min_{\dot{V}_{cool}} J(N_1,\ N_2,\ N_U) = \sum_{j=N_1}^{N_2} \delta(j)\cdot[\hat{y}(t+j\mid t) - w(t+j)]^2 + \sum_{j=1}^{N_U} \lambda(j)\cdot[\mathit{\Delta} u(t+j-1)]^2 \tag{6}$$

The first term of the function stands for the task of keeping as close as possible to the calculated open loop optimal trajectory of the critical variables $\hat{y}$ (e.g. the reactor temperature, which can easily be measured online), whereas the second term corresponds to control activity under the consideration of the systems restriction's described above. N_1, N_2 denote the number of past, and future time intervals, respectively. N_U stands for the number of controls. The prediction T_P and control horizon T_C comprises 8 intervals, respectively. Furthermore, $\lambda = 3000$ and $\delta(j) = 0.7^{(T_P - j)}$ are the variation and offset weighting factor, respectively. In order to guarantee robustness and feasibility with respect to output constraints despite of uncertainties and unexpected disturbances, an adaptive dynamic back-off strategy is introduced into the optimization problem to guarantee that the restrictions are not violated at any time point, in particular, in case of sudden cooling failure [1]. For this purpose, it is necessary to consider the impact of the uncertainties between the time points for re-optimization and the resulting control re-setting by setting, in advance, the constraint bounds much more severe than the physical ones within the moving horizon. Thus, as shown in Fig. 2 left, the key idea of the approach is based on backing-off of these bounds with a decreasing degree of severity leading then to the generation of a trajectory which consist of the modified constraint bounds along the moving horizon. For the near future time points within the horizon, these limits (bounds) are more severe than the real physical constraints and will gradually be eased (e.g. logarithmic) for further time points. The trajectory of these bounds is dependent on the amount of measurement error and parameter variation including uncertainty.

As previously illustrated in Fig. 1, the true process optimum lies on the boundary of the feasible region defined by the active constraints. Due to the uncertainty in the parameters and the measurement errors, the process optimum and

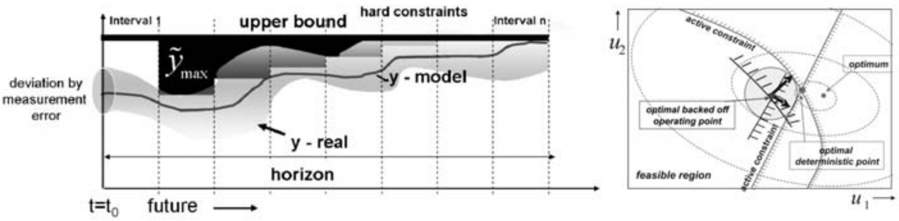


Fig. 2. Back-off strategy within moving horizon; back-off from active constraints

the set-point trajectory would be infeasible. By introducing a back-off from the active constraints in the optimization, the region of the set-point trajectory is moved inside the feasible region of the process to ensure, on the one hand, feasible operation, and to operate the process, on the other hand, still as closely to the true optimum as possible. By this means, the black-marked area in Fig. 2 illustrates the corrected bounds $\tilde{y}_{max}$ of the hard constraints. Here, it should however be noted that due to the severe bound at the computation of the previous horizon, the initial value at t_0 is rather far away from the constraint limit in the feasible area. Thus, in the first intervall of the current moving horizon, the bound is set at the original physical limit to avoid infeasibility. The back-off adjustment starts from the second interval, i.e. from the time point on, where the next re-optimization begins. The size of $\tilde{y}_{max}$ strongly depends on parametric uncertainty, disturbances, and the deviation by measurement errors. Thus, the constraints in (8) within the moving horizon (8 intervals) are now reformulated as follows with $j = 2, ..., 8$, $\alpha = 0.5$, $\tilde{T}_{max} = 4\,K$ and $\tilde{T}_{ad,\,max} = 3\,K$:

$$T(j) \leq 356\,K - \tilde{T}_{max} \cdot \alpha^{(j-2)}; \quad T_{ad}(j) \leq 500\,K - \tilde{T}_{ad,\,max} \cdot \alpha^{(j-2)} \qquad (7)$$

The decision variable is the cooling flow rate. In order to test robustness characteristics of the controler, the performances of the open-loop nominal solution, the nominal NMPC, and the NMPC with the proposed adaptive back-off approach are compared under different disturbances, namely: catalyst activity mismatch and fluctuations of the reactor jacket cooling fluid temperature. Additionally, all measurements are corrupted with white noise e.g. component amount 8% and temperature 2%.

3.1 Dynamik Real–Time Optimization

The size of the dynamic operating region around the optimum (see Fig. 2 right) is affected by fast disturbances. These are, however, efficiently buffered by the proposed regulatory NMPC-based approach. On the other hand, there are, in fact, slowly time-varying non-zero mean disturbances or drifting model parameters which change the plant optimum with time. Thus, a online re-optimization i.e. dynamic real-time optimization (D-RTO) may be indispensable for an optimal operation. When on-line measurement gives access to the system state,

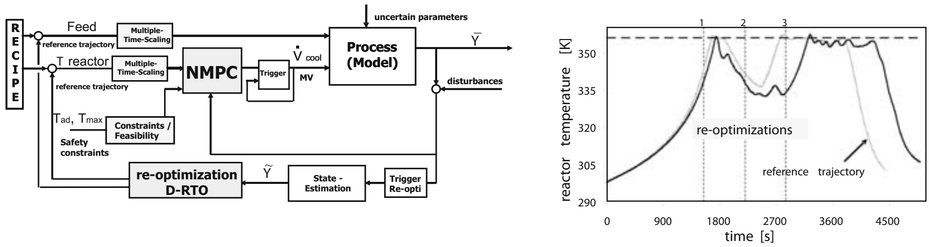


Fig. 3. Integration of NMPC and D-RTO; On-line re-optimization

its promises considerably improvement. Moreover, additional constraints can be integrated. Simulation results are shown in Fig. 3 right.

In order to compensate slow disturbances, the on-line re-optimization problem is automatically activated three times along the batch process time according to a trigger defined as the bounded above difference between the reactor temperature and the temperature reference trajectory (Fig. 3 right). New recipes resulting from this are then updated as input to the on-line framework. Due to the different trigger time-points, the current D-RTO problem progressively possesses a reduced number of variables within a shrinking horizon [6]. As a result, the total batch time increases. But, despite the large plant mismatch and the absence of reliable kinetic knowledge a very good control is accomplished. Thus, the resulting NMPC scheme embedded in the on-line re-optimization framework is viable for the optimization of the semi-batch reactor recipe while simultaneously guaranteeing the constraints compliance, both for nominal operation as well as for cases of large disturbances e.g. failure situation. The proposed scheme yields almost the same profit as the one of the off-line optimization operational profiles (see Tab. 1) where CB_f and CC_f are the final total amount of B and C.

Table 1. Simulation results

	CB_f [mol]	CC_f [mol]	t_f [s]
Nominal open-loop optimization	152.5	37.8	4297
NMPC w. uncertainty + dyn. back-off	127.9	12.8	4297
NMPC w. uncertainty + dyn. back-off + D-RTO	148.8	36.8	4892

4 Robust NMPC Under Chance Constraints

Since the prediction of future process outputs within an NMPC moving horizon is based on a process model involving the effects of manipulated inputs and disturbances on process outputs, the compliance with constraints on process outputs is more challenging than these on process inputs. Moreover, as the model involves uncertainty, process output predictions are also uncertain. This results in output constraints violation by the close-loop system, even though predicted outputs over the moving horizon might have been properly constrained. Consequently, a method of incorporating uncertainty explicit into the output constraints of the online optimization is needed. Thus, in this work, a robust NMPC that uses a close-loop model considering the uncertainty in future process outputs due to stationary and non-stationary stochastic disturbances is presented. The new controller solves a chance-constrained nonlinear dynamic problem at each execution in order to determine the set of control moves that will optimize the expected performance of the system while complying with the constraints. The controller deals with the model uncertainty and disturbances by replacing deterministic constraints in the NMPC formulation of the form $y_{min} \leq y \leq y_{max}$, here Eq. (8), with chance constraints of the form:

$$\boldsymbol{Pr}\left\{y_{min} \leq y \leq y_{max}\right\} \geq \boldsymbol{\alpha} \tag{8}$$

The main challenge lies in the computation of the probability and its gradients. To address this problem, we propose in [7] an inverse mapping approach where the monotonic relationship of the constrained output y^{bound} to at least one uncertain input ξ_S is employed. Due to the monotony, the constrained bound value y^{bound} in the output region corresponds to a limit value ξ_S^L for ξ_S in the uncertain input region. The basic idea is to map the probabilistic constrained output region back to a bounded region of the uncertain inputs. Hence, the output probabilities and, simultaneously, their gradients can be calculated through multivariate integration of the density function of the uncertain inputs by collocation on finite elements with an optimal number of collocation points and intervals.

$$\begin{aligned} P\left\{y \leq y^{bound}\right\} &= P\left\{\xi_S \leq \xi_S^L,\, \xi_k \subseteq \mathbb{R}^K, s \neq k\right\} \\ &= \int\limits_{-\infty}^{\infty} \cdots \int\limits_{-\infty}^{\xi_S^L} \cdots \int\limits_{-\infty}^{\infty} \rho(\xi)\, d\xi_l \cdots d\xi_S \cdots d\xi_K \quad l = 1, ..., n \end{aligned} \tag{9}$$

where the $\rho(\xi)$ is the unified distribution function of ξ. The solution strategy is however not dependent on the distribution of the uncertain variables. The probability computation procedure can straightforwardly be extended to multiple single probabilistic constraints with different confidence levels. To compute the probability values of (13), a multivariate integration in the region of uncertain inputs is required. Numerical integration is needed, especially in cases of correlated uncertain variables. We refer to Wendt et al. (2001) for a method based on orthogonal collocation for correlated uncertain variables with normal distributions. Following this idea, we extend the approach to dealing with nonlinear dynamic optimization problems [2, 3]. In this contribution, a new framework is proposed also for such stochastic dynamic optimization problems where *no monotonic* relation between constrained output and any uncertain input variable can be guaranteed. This approach also involves efficient algorithms for the computation of the required (mapping) reverse projection. To decompose the problem, the proposed approach uses a two-stage computation framework(see Fig. 4 left). The upper stage is a superior optimizer following the sequential strategy. Inside the simulation layer, there is a two-layer structure to compute the probabilistic constraints. One is the superior layer, where the probabilities and their gradients are finally calculated by multivariate integration. The main novelty is contained in the other, the sub-layer, and is the key to the computation of the chance constraints with non-monotonous relation. The main principal is that for the multivariate integration the bounds of the constrained output y and those for the selected uncertain variables ξ reflecting the feasible area concerning y are computed at temporarily given values of both the decision and the other uncertain variables. Thus, all local minima und maxima of the function reflecting y are first detected (see Fig. 4 right). The computation of the required points of $[\min y(\xi)]$ and $[\max y(\xi)]$ is achieved by an optimization step in the sub-layer. With the help of those significant points, the entire space of ξ can be divided into monotonous sections in which the bounds of the subspaces of feasibility can be computed through a reverse projection by solving the model equations in the following step of this inferior layer. The bounds of feasibility are supplied to the superior multivariate integration layer, where the necessary probabilities (Eqs. 14) and the gradients are computed by adding all those feasible fractions together (Fig. 4 right).

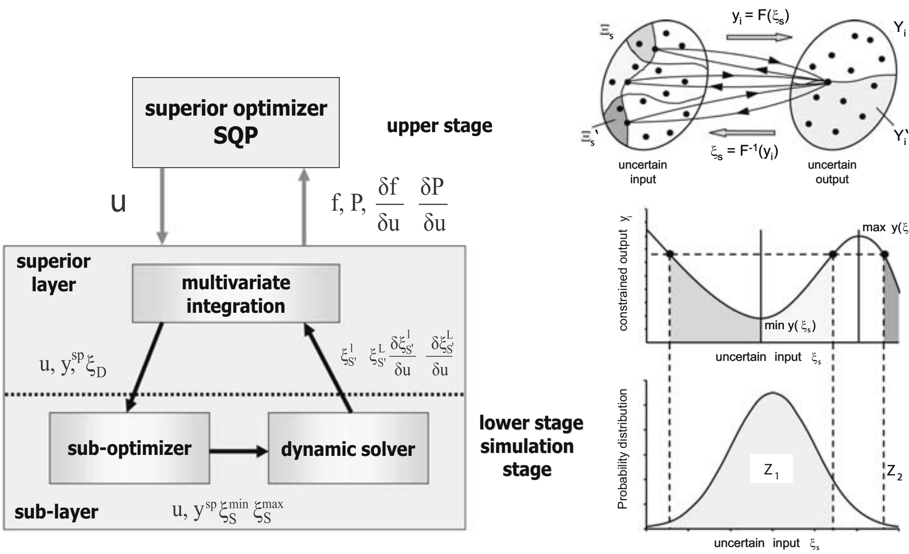


Fig. 4. Structure of optimization framework (left) and mapping of feasible reagions (right)

$$\boldsymbol{Pr} = \sum \boldsymbol{Pr}(z_i); \quad \boldsymbol{Pr}(z_i) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\xi_S^{l,i}}^{\xi_S^{L,i}} \varphi\left(\xi_i,\ R\right) d\xi_S\, d\xi_{S-1}\cdots d\xi_1 \tag{10}$$

where R denotes the covariance matrix. Arising changes of the integration limits are verified for every monotone section. In case of variation, a reverse projection of the constrained output leads to new integration limits, which are, then, employed to compute the probability by multivariate integration. The general chance constrained NMPC problem which is solved at each sampling time k can be formulated as follows:

$$\begin{aligned} &\min \sum_{i=1}^{N_u} \left[\boldsymbol{u}(k+i) - \boldsymbol{u}(k+i-1)\right]^2 \\ s.t.\ &\boldsymbol{x}(k+i+1 \mid k) = \boldsymbol{g}_1[\boldsymbol{x}(k+i \mid k),\ \boldsymbol{u}(k+i \mid k),\ \boldsymbol{\xi}(k+i)] \\ &\boldsymbol{y}(k+i \mid k) = \boldsymbol{g}_2[\boldsymbol{x}(k+i \mid k),\ \boldsymbol{u}(k+i \mid k),\ \boldsymbol{\xi}(k+i)] \\ &\boldsymbol{Pr}\left\{\boldsymbol{y}_{min} \le \boldsymbol{y}(k+i \mid k) \le \boldsymbol{y}_{max}\right\} \ge \alpha; \quad i = 1, ..., n \\ &\boldsymbol{u}_{min} \le \boldsymbol{u}(k+i \mid k) \le \boldsymbol{u}_{max}; \quad i = 0, ..., m-1 \\ &\Delta\boldsymbol{u}_{min} \le \Delta\boldsymbol{u}(k+i \mid k) = \boldsymbol{u}(k+i \mid k) - \boldsymbol{u}(k+i-1 \mid k) \le \Delta\boldsymbol{u}_{max} \end{aligned} \tag{11}$$

Where $\boldsymbol{g}_1$ are the first-principle model equations describing the dynamic changes of the state variables $\boldsymbol{x}$, while $\boldsymbol{g}_2$ describe the state of the constrained variables $\boldsymbol{y}$ depending on the control variables $\boldsymbol{u}$ and the uncertain parameters $\boldsymbol{\xi}$, and $\alpha = 96.7\%$. The efficiency of the chance-constrained approach is proved through application to the same scenario of the fed-batch reactor under safety constraints. The resulting NMPC scheme is also embedded in the same on-line optimization framework. Moreover, the relationship between the probability levels and the corresponding values of the objective function can be used for a suitable trade-off decision between *profitability* and *robustness*. Tuning the value of $\boldsymbol{\alpha}$ is also an issue of the relation between *feasibility* and profitability. The solution of a defined problem (Eq. 15), however, is only able to arrive at a maximum value α^{max} which is dependent on the properties of the uncertain inputs and the restriction of the controls. The value of α^{max} can be computed through a previous probability maximization step. The use of this strategy with the consideration of uncertainties in advance has for those NMPC-Problems a great impact in those periods, where the reference trajectory is very close to a defined upper bound of the constraint output. However, a comparison between the stochastic approach and the deterministic dynamic adaptive back-off strategy is meaningful in order to find further improvement of operation policies due to the stochastic approach. Thus, the reference trajectory of the reactor temperature in Equation (10) is set as a constant which is close to the upper bound of the reactor temperature.

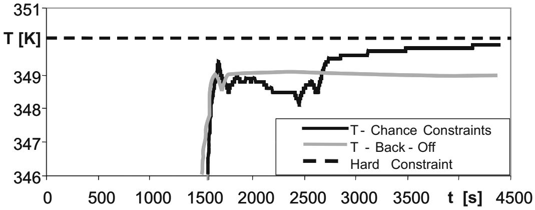


Fig. 5. Comparison back-off strategy and chance constraints

The resulting trajectories of the reactor temperature concerning both strategies are illustrated in Fig. 5. The figure shows, that the reactor temperature resulted by the back-off strategy reaches very early a stationary value caused by fixed bounds of the temperature formulated in the corresponding optimization problem. The temperature curve of the stochastic approach shows more drastical changes with lower values of temperatures in earlier parts of the diagram and higher values later. This is caused by the fact, that with the consideration of uncertainties in advance, also the change of sensitivities of uncertain parameters towards the reactor temperature can be taken into consideration by the stochastic approach. At the beginning in the diagram, the stochastic approach realizes the matching of a more conservative strategy to higher sensitivities, and thus the operation achieves more robustness than the one achieved by the back-off strategy. At the end of the curves, the decrease of sensitivities is used for a closer approach to the maximum temperature and thus leads to a better objective value. Therefore, the strategy leads to an improvement of both, robustness and the objective value.

5 Conclusions

The chance constrained optimization framework has been demonstrated to be promising to address optimization and control problems under uncertainties. Feasibility and robustness with respect to input and output constraints have been achieved by the proposed approach. Thus, the solution of the problem has the feature of prediction, robustness and being closed-loop. The resulting NMPC scheme embedded in the on-line re-optimization framework is viable for the optimization of the reactor recipe while simultaneously guaranteeing the constraints compliance, both for nominal operation as well as for cases of large disturbances e.g. failure situation. In fact, the approach is relevant to all cases when uncertainty can be described by any kind of joint correlated multivariate distribution function. The authors gratefully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG).

References

[1] O. Abel and W. Marquardt. Scenario-integrated on-line optimization of batch reactors. *Journal of Process Control*, 13:703–715, 2003.

[2] H. Arellano-G., W. Martini, M. Wendt, P. Li, and G. Wozny. Chance constrained batch distillation process optimization under uncertainty. In I.E. Grossmann and C.M. McDonald, editors, *FOCAPO 2003 - fourth International Conference on Foundations of Computer-Aided Process Operations*, pages 609–612. CACHE, CAST Division, AICHE, 2003.

[3] H. Arellano-Garcia, W. Martini, M. Wendt, and G. Wozny. A New Optimization Framework for Dynamic Systems under Uncertainty. *Computer-Aided Chemical Engineering*, 18:553–559, 2004.

[4] J. H. Lee and Z. H. Yu. Worst-case formulations of model predictive control for systems with bounded parameters. *Automatica*, 33:765–781, 1997.

[5] P. Li, M. Wendt, and G. Wozny. Robust Model Predictive Control under Chance Constraints. *Computers & Chemical Engineering*, 24:829–833, 2000.

[6] Z. Nagy and R. D. Braatz. Open-loop and close-loop robust optimal control of batch processes using distributional and worst-case analysis. *Journal of Process Control*, 14:411–422, 2004.

[7] M. Wendt, P. Li, and G. Wozny. Nonlinear Chance-Constrained Process Optimization under Uncertainty. *Ind. Eng. Chem. Res.*, 41:3621–3629, 2002.

[8] E. Zafirou. Robust model predictive control of process with hard constraints. *Computers & Chemical Engineering*, 14:359–371, 1990.

Interval Arithmetic in Robust Nonlinear MPC

D. Limon[1], T. Alamo[1], J.M. Bravo[2], E.F. Camacho[1], D.R. Ramirez[1],
D. Muñoz de la Peña[1], I. Alvarado[1], and M.R. Arahal[1]

[1] Departamento de Ingeniería de Sistemas y Automática, Universidad de Sevilla, Sevilla, Spain
{limon, alamo, eduardo, danirr, davidmps, alvarado, arahal} @cartuja.us.es

[2] Departamento de Ingeniería Electrónica, Sistemas Informáticos y Automática. Universidad de Huelva, Huelva, Spain
caro@uhu.es

Summary. This paper shows how interval arithmetic can be used to design stabilizing robust MPC controllers. Interval arithmetic provides a suitable framework to obtain a tractable procedure to calculate an outer bound of the range of a given nonlinear function. This can be used to calculate a guaranteed outer bound on the predicted sequence of reachable sets. This allows us to consider the effect of the uncertainties in the prediction and to formulate robust dual-mode MPC controllers with ensured admissibility and convergence. Interval arithmetic can also be used to estimate the state when only outputs are measurable. This method provides a guaranteed outer bound on the set of states consistent with the output measurements. Generalizing the controllers based on reachable sets, a novel robust output feedback MPC controller is also proposed.

1 Introduction

Consider a process described by an uncertain nonlinear time-invariant discrete time model

$$x^+ = f(x, u, w) \tag{1}$$

$$y = g(x, u, v) \tag{2}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the current control vector, $y \in \mathbb{R}^p$ is the measured output, $w \in \mathbb{R}^{n_w}$ is the disturbance input which models the uncertainty, $v \in \mathbb{R}^{p_v}$ is the measurement noise and x^+ is the successor state. x_k, u_k, y_k, w_k and v_k denote the state, input, output, uncertainty and noise of the plant at sampling time k, respectively.

It is assumed that the uncertainty is bounded and contained in a compact set,

$$w \in W \tag{3}$$

which contains the origin and the noise is bounded in the compact set

$$v \in V. \tag{4}$$

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 317–326, 2007.
springerlink.com

The system is subject to constraints on both the state and the control input. These constraints are given by

$$u \in U, \quad x \in X \tag{5}$$

where X is a closed set and U a compact set, both of them containing the origin.

The objective of this paper is to present some robust MPC controllers which are able to robustly stabilize this system. These controllers deal with the uncertainty by means of interval arithmetic with an affordable increment of the computational burden with respect to the nominal problem. In the following section, a brief introduction to interval arithmetic is presented.

2 A Brief Introduction to Interval Arithmetic

An interval $X = [a, b]$ is the set $\{\ x \ : \ a \leq x \leq b\ \}$. The unitary interval is $\mathbf{B} = [-1, 1]$. The set of real compact intervals $[a, b]$, where $a, b \in \mathbb{R}$ and $a \leq b$, is denoted as I. A box is an interval vector. A unitary box, denoted as $\mathbf{B}^m$, is a box composed by m unitary intervals. Given a box $Q = ([a_1, b_1], \ldots, [a_n, b_n])^\top$: $mid(Q)$ denotes its center and $\text{diam}\,(Q) = (b_1 - a_1, \ldots, b_n - a_n)^\top$. The Minkowski sum of two sets X and Y is defined by $X \oplus Y = \{\ x + y \ : \ x \in X, y \in Y\ \}$. Given a vector $p \in \mathbb{R}^n$ and a matrix $H \in \mathbb{R}^{n \times m}$, the set:

$$p \oplus H\mathbf{B}^m = \{\ p + Hz \ : \ z \in \mathbf{B}^m\ \}$$

is called a zonotope of order m. Given a continuous function $f(\cdot, \cdot)$ and sets $X \subset \mathbb{R}^n$ and $W \subset \mathbb{R}^{n_w}$, $f(X, W)$ denotes the set $\{\ f(x, w) \ : \ x \in X,\ w \in W\ \}$. Interval arithmetic and Kühn's method provides two approaches to obtain outer bounds of set $f(X, W)$.

The interval arithmetic is based on operations applied to intervals. An operation *op* can be extended from real numbers to intervals as: for a given $A, B \in I$, $A\ op\ B = \{\ a\ op\ b \ : \ a \in A, b \in B\ \}$. The four basic interval operations are defined in [1], where the sum is $[a, b] + [c, d] = [a + c, b + d]$, and the product is $[a, b] * [c, d] = [\min(ac, ad, bc, bd), \max(ac, ad, bc, bd)]$, for instance. The interval extension of standard functions {*sin, cos, tan, arctan, exp, ln, abs, sqr, sqrt*} is possible too. A guaranteed bound of the range of a non-linear function $f : \mathbb{R}^n \to \mathbb{R}$ can be obtained by means of the natural interval extension, that is, replacing each occurrence of each variable by the corresponding interval variable, by executing all operations according to interval operations and by computing ranges of the standard functions.

Theorem 4. *[1] A natural interval extension $\Box(f)$ of a continuous function $f : \mathbb{R}^n \to \mathbb{R}$ over a box $X \subseteq \mathbb{R}^n$ satisfies that $f(X) \subseteq \Box(f(X))$.*

The natural interval extension is a particular and efficient way to compute an interval enclosure. However, natural interval extension may lead to unnecessary overestimation when a variable appears several times in the same expression (multi-occurrence). To reduce this overestimation, Kühn's method can be used [2].

Kühn's method is a procedure to bound the orbits of discrete dynamical systems. In Kühn's method, the evolution of the system is approximated by a zonotope and sub-exponential overestimation is proven [2]. First the zonotope inclusion operator is introduced in the following theorem:

Theorem 1 (Zonotope inclusion). *Consider a family of zonotopes represented by* $Z = p \oplus \mathbf{M}\mathbf{B}^m$ *where* $p \in \mathbb{R}^n$ *is a real vector and* $\mathbf{M} \in I^{n \times m}$ *is an interval matrix. A zonotope inclusion, denoted by* $\diamond(Z)$*, is defined by:*

$$\diamond(Z) = p \oplus [mid(\mathbf{M}) \quad G] \begin{bmatrix} \mathbf{B}^m \\ \mathbf{B}^n \end{bmatrix} = p \oplus J\mathbf{B}^{m+n}$$

where $G \in \mathbb{R}^{n \times n}$ *is a diagonal matrix that satisfies:*

$$G_{ii} = \sum_{j=1}^{m} \frac{\mathrm{diam}\,(\mathbf{M}_{ij})}{2}, \quad i = 1, \ldots, n.$$

Under these definitions it results that: $Z \subseteq \diamond(Z)$*.*

The following theorem is a generalization of Kühn's method for a function which depends on an unknown but bounded vector of parameters [3].

Theorem 5. *Given a function* $f(x, w) : \mathbb{R}^n \times \mathbb{R}^{n_w} \to \mathbb{R}^n$*, a zonotope* $X = p \oplus H\mathbf{B}^m$ *and a zonotope* $W = c_w \oplus C_w\mathbf{B}^{s_w}$*, consider the following interval extensions:*

- *A zonotope* $q \oplus S\mathbf{B}^d$ *such that* $f(p, W) \subseteq q \oplus S\mathbf{B}^d$*.*
- *An interval matrix* $\mathbf{M} = \Box(\nabla_x f(X, W))H$*.*
- *A zonotope* $\Psi(X, W) = q \oplus S\mathbf{B}^d \oplus \diamond(\mathbf{M}\mathbf{B}^m) = q \oplus H_q\mathbf{B}^l$ *with* $l = d + n + m$

Under the previous assumptions it results that $f(X, W) \subseteq \Psi(X, W)$

Note that the zonotope $q \oplus S\mathbf{B}^d$ of theorem 5 can be obtained by means of a natural interval extension of $f(p, W)$. It is worth remarking that this method increases the dimension of the obtained zonotope, and hence its complexity. In order to reduce this effect, a lower order zonotope which bounds the obtained zonotope can be used instead. This bound can be obtained by means of the procedure proposed in [3].

In the following section it is shown how an outer estimation of the reachable set can be obtained by means of the presented interval arithmetic methods.

3 Guaranteed Approximation of the Reachable Set

For a system in absence of uncertainties and for a given sequence of control inputs, the prediction of the evolution of the system is a trajectory. However, when uncertainties or noises are present, there exists a trajectory for each realization of the uncertainties and noise. The set of these trajectories forms a tube that will be denoted as the sequence of reachable sets. a precise definition is the following:

Definition 1 (Sequence of reachable sets). *Consider a system given by* (1), *consider also that the set of states at sample time k is X_k and that a sequence of control inputs $\{u(k+i|k)\}$ is given, then the reachable sets $\{\mathbb{X}(k|k), \mathbb{X}(k+1|k), \ldots, \mathbb{X}(k+N|k)\}$ are obtained from the recursion: $\mathbb{X}(k+j|k) = f(\mathbb{X}(k+j-1|k), u(k+j-1|k), W)$ where $\mathbb{X}(k|k) = X_k$.*

The set of states X_k might be either a singleton x_k, in the case that the state is certainly known, or a set, in the case that this is unknown but bounded (this is what happen when the state is not fully measurable).

It is clear that the computation of this sequence of sets is not possible in general; fortunately, a guaranteed estimation of this sequence can be obtained if for a given control input u, a guaranteed estimator of $f(X, u, W)$, $\psi(X, u, W)$ is used for the computation. The guaranteed estimation must satisfy that $\psi(X, u, W) \supseteq f(X, u, W)$, for all X, u, W. Thus the sequence of guaranteed estimation of the reachable set is given by the following recursion:

$$\hat{\mathbb{X}}(k+j|k) = \psi(\hat{\mathbb{X}}(k+j-1|k), u(k+j-1|k), W)$$

with $\hat{\mathbb{X}}(k|k) = X_k$. In order to emphasize the parameters of $\hat{\mathbb{X}}(k+j|k)$, this will be denoted as $\mathbb{X}(k+j|k) = \Psi(j; \mathbb{X}_k, \mathbf{u}, W)$.

In the previous section, two methods to obtain a guaranteed estimator of a function were introduced: the interval extension of the function $f(X, u, W)$ and the one based on Kühn's method. Both procedures can be used to approximate the sequence of reachable sets.

The interval extension method provides a guaranteed box considering that X and W are boxes. Its main advantages are that this procedure is easy to implement and its computational burden is similar to the one corresponding to evaluate the model. Moreover, the bounding operator based on interval extension is monotonic, that is, if $A \subseteq B$, then $\psi(A, u, W) \subseteq \psi(B, u, W)$. Its main drawback is that it may be very conservative.

The procedure based on Kühn's method calculates zonotopic approximations, assuming that X and W are zonotopes. This method provides more accurate approximation than the one obtained by the interval extension at expense of a more involved procedure with a bigger computational burden. Moreover this procedure may not be a monotonic procedure.

The monotonicity property of the estimation procedure $\psi(\cdot, \cdot, \cdot)$ provides an interesting property to the estimated reachable set: consider the sequence of estimated reachable sets $\hat{\mathbb{X}}(k+j|k) = \Psi(j; X_k, \mathbf{u}, W)$. Consider a set $X_{k+1} \subseteq \hat{\mathbb{X}}(k+1|k)$, then for all $j \geq 1$, $\Psi(j-1; X_{k+1}, \mathbf{u}, W) \subseteq \Psi(j; X_k, \mathbf{u}, W)$. This property will be exploited in the following sections.

Based on this estimation of the sequence of reachable sets, a robust MPC controller is presented in the following section.

4 Robust MPC Based on Guaranteed Reachable Sets

While in a nominal framework of MPC, the future trajectories are obtained by means of the nominal model of the system, when uncertainties are present, the predictions depends on the future realization of the uncertainties. This effect of the uncertainties can be considered in the controlled design by replacing the sequence of predicted states by the sequence of reachable sets. As it was commented before, the exact computation of this sequence is very difficult and tractable methods to compute a guaranteed estimation of this sequence can be used instead.

In order to enhance the accuracy of the estimation of the reachable sets, a pre-compensation of the system can be used. This consists in parametrizing the input as $u_k = K{\cdot}x_k + v_k$ where v_k is the new (artificial) control input. By doing this, system (1) can be rewritten as $f(x, K{\cdot}x + v, w) = f_K(x, v, w)$. This is a practical way to provide some amount of feedback to the predictions as well as a technique to enhance the structure of the system to obtain better interval or zonotopic approximations.

The proposed MPC is based on the solution of an optimization problem such that a performance cost of the predicted nominal trajectory is minimized (although other cost functions depending on the uncertainties might be considered, such as the cost of the worst case in the min-max framework). This cost function for a given sequence of N control inputs $\mathrm{v}(k) = \{v(k|k), \cdots, v(k+N-1|k)\}$, is given by

$$J_N(x_k, \mathrm{v}) = \sum_{j=0}^{N-1} L(\hat{x}(k+j|k), v(k+j|k)) + V(\hat{x}(k+N|k)) \tag{6}$$

where $\hat{x}(k+j+1|k) = f_K(\hat{x}(k+j|k), v(k+j|k), 0)$, with $\hat{x}(k|k) = x_k$. The stage cost function $L(\cdot,\cdot)$ is a positive definite function of the state and input and the terminal cost function $F(\cdot)$ is typically chosen as a Lyapunov function of system $x^+ = f_K(x, 0, 0)$. Thus, the optimization problem $P_N(x_k)$ to solve is

$$\begin{aligned}
\min_{\mathrm{v}} \quad & J_N(x_k, \mathrm{v}) \\
s.t.\ & \hat{\mathbb{X}}(k+j|k) = \Psi_K(j; x_k, \mathrm{v}, W) \quad j = 1, \cdots, N \\
& v(k+j|k) \oplus K\hat{\mathbb{X}}(k+j|k) \subseteq U, \quad j = 0, \cdots, N-1 \\
& \hat{\mathbb{X}}(k+j|k) \subseteq X \quad j = 0, \cdots, N \\
& \hat{\mathbb{X}}(k+N|k) \subseteq \Omega
\end{aligned}$$

where Ω is an admissible robust invariant set for the system $x^+ = f_K(x, 0, w)$, and $\psi_K(\cdot,\cdot,\cdot)$ denotes a guaranteed estimator of $f_K(\cdot,\cdot,\cdot)$.

The stabilizing design of the controller arises two questions: the admissibility of the obtained trajectories and the convergence to a neighborhood of the origin.

Admissibility: this is ensured if the initial state is feasible. In effect, if the estimation operator $\psi_K(\cdot,\cdot,\cdot)$ is monotonic (as for instance, the one based on interval extension of the model), then the optimization problem is feasible all the time [6] and hence the closed loop system is admissible.

If the monotonicity is not ensured (as may occur, for instance, when Kühn's method approach is used), then feasibility of the optimization problem may be lost. In this case, admissibility can be ensured applying the tail of the last computed optimal sequence. [5].

Convergence: this is achieved by means of a dual mode strategy ensuring that the MPC steers the system to the terminal set Ω and then switching to the local robust control law $u_k = K{\cdot}x_k$. The convergence to the terminal set can be guaranteed by using two different techniques: the first one, used in [6], consists in reducing the prediction horizon at each sampling time. Hence, the system reaches Ω in N steps or less.

A drawback of this approach is that the reduction of the prediction horizon may provide a worse closed loop behavior than when the prediction horizon is constant. In order to mitigate this and to provide convergence for a constant prediction horizon, an stabilizing constraint can be added to the problem. This stabilizing constraint is based on function

$$J_E(k) = \sum_{i=0}^{N-1} \|\hat{\mathbb{X}}(k+i|k)\|_{\beta\Omega}$$

where β is a parameter contained in $(0,1)$ and $\|A\|_B$ denotes a measure of the maximum distance between sets A and B. Clearly, if $A \subseteq B$, then this measure is zero. The proposed stabilizing constraint is [5]

$$\sum_{i=0}^{N-1} \|\hat{\mathbb{X}}(k+i|k)\|_{\Omega} - J_E(k-1) < -\frac{1-\beta}{\beta} \tag{7}$$

which does not reduce the feasibility region and ensures that $J_E(k)$ tends to zero, or equivalently, x_k tends to Ω.

This controller requires that the state is fully measurable to be implemented. If this is not possible, then an output feedback approach could be used. In the following section an output feedback MPC is presented.

5 Robust Output Feedback MPC Based on Guaranteed Estimation

In this section we present a robust MPC controller based on the measurement of the outputs. This controller computes the control input considering an estimation of the set of states calculated at each sampling time by an interval arithmetic based algorithm. This procedure is presented in what follows.

5.1 Guaranteed State Estimation

Consider an uncertain non-linear discrete-time system (1) such that the applied control inputs are known and only the outputs are measurable. The impossibility to calculate accurately the states from the outputs together with the uncertainties and noises present in the system makes that only a region where the actual state is confined is estimated. Thus, based on the measured outputs, an estimation of the set of possible states at each sample time is obtained assuming that the initial state is confined in a known compact sets $x_o \in \mathbb{X}_0$.

For a measured output y_k and input u_k, the consistent state set at time k is defined as $\mathbb{X}_{y_k} = \{\, x \in \mathbb{R}^n \; : \; y_k \in g(x, u_k, V) \,\}$. Thus, the set of possible states at sample time k, $\mathbb{X}_k$, is given by the recursion $\mathbb{X}_k = f(\mathbb{X}_{k-1}, u_{k-1}, W) \bigcap \mathbb{X}_{y_k}$ for a given bounding set of initial states $\mathbb{X}_0$ and for a given sequence of inputs u and outputs y. It is clear that the exact computation of these sets is a difficult task for a general nonlinear system. Fortunately, these sets can be estimated by using a guaranteed estimator $\psi(\cdot,\cdot,\cdot)$ of the model $f(\cdot,\cdot,\cdot)$ [3]. The estimation is not obtained by merely using $\psi(\cdot,\cdot,\cdot)$ instead of $f(\cdot,\cdot,\cdot)$ in the recursion, and some problems must be solved.

The first issue stems from the fact that consistent state set $\mathbb{X}_{y_k}$ is typically difficult to compute. Therefore, this can be replaced by a tractable outer approximation $\bar{\mathbb{X}}_{y_k}$. In [3] an ad-hoc procedure is proposed to obtain an outer approximation set $\bar{\mathbb{X}}_{y_k}$ as the intersection of the p strip-type sets. This procedure requires the measured output y_k and a region $\bar{\mathbb{X}}_k$ where $\mathbb{X}_{y_k}$ is contained; thus this will be denoted as $\bar{\mathbb{X}}_{y_k} = \Upsilon(y_k, \bar{\mathbb{X}}_k)$.

The second problem to solve is derived from the fact that the procedure $\psi(A, u, W)$ requires that A has an appropriate shape. For instance, if the natural interval extension of $f(\cdot,\cdot,\cdot)$ is being used, then A must be a box, while if it is based on Kühn's method, then A must be a zonotope. Assume that $\mathbb{X}_k$ has the appropriate shape, a zonotope for instance, then the set $\psi(\mathbb{X}_k, u_k, W)$ is also a zonotope, but the intersection $\psi(\mathbb{X}_k, u_k, W) \cap \bar{\mathbb{X}}_{y_{k+1}}$ may be not a zonotope. Then, it is compulsory to obtain a procedure to calculate a zonotope which contains this intersection. This procedure is such that for a given zonotope (box) A, and a given set B, defined as the intersection of strips, calculates a zonotope (box) $C = \Theta(A, B)$ such that $C \supseteq A \cap B$. An optimized procedure for zonotopes is proposed in [3]. In case of boxes, a simple algorithm can also be obtained based on this.

Considering these points, the following algorithm to obtain a sequence of guaranteed state estimation sets $\mathbb{X}_k$ is proposed:

1. For $k = 0$ and for a given zonotope $\mathbb{X}_0$, take $\hat{\mathbb{X}}_0 = \mathbb{X}_0$.
2. For $k \geq 1$, make
   a) $\bar{\mathbb{X}}_k = \psi(\hat{\mathbb{X}}_{k-1}, u_{k-1}, W)$.
   b) $\bar{\mathbb{X}}_{y_k} = \Upsilon(y_k, \bar{\mathbb{X}}_k)$.
   c) $\mathbb{X}_k = \bar{\mathbb{X}}_k \cap \bar{\mathbb{X}}_{y_k}$
   d) $\hat{\mathbb{X}}_k = \Theta(\bar{\mathbb{X}}_k, \bar{\mathbb{X}}_{y_k})$.

From this algorithm it is clear that both $\mathbb{X}_k$ and $\hat{\mathbb{X}}_k$ are guaranteed state estimation sets. Based on this algorithm, a robust output feedback MPC is proposed in the following section.

5.2 Robust Output Feedback MPC

In the previous section, some robust MPC controllers based on a guaranteed estimation of the reachable sets were presented. This estimation is achieved by using a guaranteed estimator $\psi(\cdot,\cdot,\cdot)$ of the model function $f(\cdot,\cdot,\cdot)$. These controllers are able to robustly steer the system to a target set under assumption that the full state variables are measurable. In the case that only the output signals are measurable, then a set of estimated states can be calculated by means of the proposed guaranteed state estimator estimator. Then the state feedback MPC controller can be modified to deal with the output feedback case by considering that the predicted sequence of reachable sets starts from a given set instead of a single state initial state.

Consider that the control law $u = h(x)$ is an admissible robustly stabilizing control law for system (1) in a neighborhood of the origin. Assume that system (1) is locally detectable for the corresponding dynamic output feedback controller

$$\begin{aligned} \hat{x}_{k+1} &= \kappa(\hat{x}_k, u_k, y_k) \\ u_k &= h(\hat{x}_k) \end{aligned} \tag{8}$$

in such a way that the closed loop system

$$x_{k+1} = f(x_k, h(\hat{x}_k), w_k) \tag{9}$$

$$\hat{x}_{k+1} = \kappa(\hat{x}_k, h(\hat{x}_k), g(x_k, h(\hat{x}_k), v_k)) \tag{10}$$

is robustly stable in $(x,\hat{x}) \in \Gamma$, where Γ is a polyhedral robust invariant set for system (9), i.e. $\forall (x_k,\hat{x}_k) \in \Gamma$, $x_k \in X$, $h(\hat{x}_k) \in U$, and $(x_{k+1},\hat{x}_{k+1}) \in \Gamma$, $\forall w \in W, v \in V$. It is clear that Γ contains the origin in its interior.

Assume that there is available a procedure to implement the proposed estimation algorithm that provides $\hat{\mathbb{X}}_k$ at each sampling time from the measurement y_k. Then the robust output feedback MPC controller first estimates the set of states $\hat{\mathbb{X}}_k$ and then calculates the control input based on this set by minimizing an optimization problem $P_N^o(\hat{\mathbb{X}}_k)$.

The cost to minimize $J_N(\hat{\mathbb{X}}_k, \mathrm{u})$ must be calculated from the set of initial states $\hat{\mathbb{X}}_k$. This can be done for instance based on the nominal prediction considering as the initial state the center of the zonotope $\hat{\mathbb{X}}_k$ or calculating the sequence of reachable sets and computing the worst case cost, in a similar way to the min-max paradigm. Thus, the optimization problem $P_N^o(\hat{\mathbb{X}}_k)$ is given by

$$\begin{aligned} \min_{\mathrm{u},\hat{x}} \quad & J_N(\hat{\mathbb{X}}_k, \mathrm{u}) \\ s.t.\ & \hat{\mathbb{X}}(k+j|k) = \Psi(j; \hat{\mathbb{X}}_k, \mathrm{u}, W) \quad j = 1, \cdots, N \\ & u(k+j|k) \in U, \quad j = 0, \cdots, N-1 \\ & \hat{\mathbb{X}}(k+j|k) \subseteq X \quad j = 0, \cdots, N \\ & (x,\hat{x}) \in \Gamma, \quad \forall x \in \hat{\mathbb{X}}(k+N|k) \end{aligned}$$

The extra decision variable $\hat{x}$ has been added to force that for any state contained in $\hat{\mathbb{X}}(k+N|k)$, the dynamic controller (8) stabilizes the system considering $\hat{x}$ as initial dynamic state. This constraint can be easily implemented thanks to the polyhedral nature of Γ. Assume that $\hat{X}(k+N|k) = p \oplus H\mathrm{B}^N$ and $\Gamma = \{(x, \hat{x}) : T_1 x + T_2 \hat{x} \leq t\}$ then the terminal constraint can be posed as $T_2\hat{x} \leq t - T_1 p - \|T_1 H\|_1$, where $\|T_1 H\|_1$ denotes the vector which $i-$th component is the 1-norm of the $i-$th row of the matrix $T_1 H$.

The admissibility of the controller can be ensured by means of the methods used for the case of full-state measurement presented in the previous section. However, the addition of the state estimator may introduce feasibility loss of the optimization problem due to the fact that the set $\hat{X}_{k+1}$ may be not contained in the set $\psi(\hat{X}_k, u_k, W)$ because of the outer approximation used in the computation of $\hat{X}_{k+1}$. The probability that this happens is low but if so, the admissibility of the problem can be ensured by solving the problem $P_N(\hat{\mathbb{X}}(k+1|k))$ instead or by merely applying $u^*(k+1|k)$. Convergence of the real state to the set $\Omega = Proj_x(\Gamma)$ can be ensured by the previously proposed method: shrinking the prediction horizon or considering an stabilizing constraint. Once that $\hat{X}_k \subset \Omega$, then the controller switches to the local dynamic controller considering as the initial dynamic state $\hat{x}^*$.

6 Conclusions

This paper summarizes some results on interval arithmetic applied to the design of robust MPC controllers. First it is shown how the keystone is the procedure to bound the range of a function and how this can be used to approximate the sequence of reachable sets. This sequence can be used to design the robust MPC controller in a natural way replacing the predicted trajectory by the sequence of reachable sets. Admissibility and convergence of this controller can be guaranteed by several methods.

The bounding procedure allows us to present an algorithm to estimate the set of states based on the measure of the outputs. Based on this, a robust MPC controller based on output feedback is presented. This controller ensures the admissible evolution of the system to a neighborhood of the origin under the existence of a local detector.

References

[1] Moore, R.E. “Interval Analysis”, *Prentice-Hall, Englewood Cliffs, NJ.*, (1966).

[2] Kühn, W. “Rigorous computed orbits of dynamical systems without the wrapping effect”, *Computing*, **61**, 47-67, (1998).

[3] Alamo, T. and Bravo, J.M. and Camacho, E.F. , “Guaranteed state estimation by zonotopes”, *Automatica*, **41**, 1035-1043, (2005).

[4] Bravo, J.M. and Limon, D. and Alamo, T. and Camacho, E.F. , “Robust MPC of constrained descrete-time nonlinear systems based on zonotopes”, *European Control Conference*, (2003).

[5] Bravo, J.M. and and Alamo, T. and Camacho, E.F. , “Robust MPC of constrained discrete-time nonlinear systems based on approximated reachable sets ”, *Submitted to Automatica*, 42 (2005), pp. 1745–1751.

[6] Limon, D. and Bravo, J.M. and Alamo, T. and Camacho, E.F. , “Robust MPC of constrained nonlinear systems based on interval arithmetic”, *IEE Proceedings - Control Theory and Applications*, **152**, (2005).

[7] Messina, M. J. and Tuna, S. Z. and Teel A. R., “Discrete-time certainty equivalence output feedback: allowing discontinuous control laws including those from Model Predictive Control”, *Automatica*, **41**, 617-628, (2005).

Optimal Online Control of Dynamical Systems Under Uncertainty

Rafail Gabasov[1], Faina M. Kirillova[2], and Natalia M. Dmitruk[2]

[1] Belarussian State University, 4 Nezavisimosti av., Minsk 220050, Belarus
[2] Institute of Mathematics, 11 Surganov str., Minsk 220072, Belarus
kirill@nsys.minsk.by, dmitruk@im.bas-net.by

Summary. A problem of synthesis of optimal measurement feedbacks for dynamical systems under uncertainty is under consideration. An online control scheme providing a guaranteed result under the worst-case conditions is described.

1 Introduction

An up-to-date methodology for control of constrained systems is model predictive control (MPC) [1]. MPC feedback strategies are constructed as a result of open-loop optimization of a nominal model for a given state. However, closed-loop performance can be poor due to uncertainties present in the system such as disturbances or modeling inaccuracies, and moreover, the exact measurement of the state can be unavailable. For these reasons, in recent research attention has been given to output feedback [2] and robust MPC techniques design [3]. In the latter two approaches can be distinguished. One method is to optimize a nominal system subject to tightened constraints [4], while a game approach leads to min-max formulations of MPC [5, 6].

This paper deals with the optimal synthesis problem for dynamical systems under uncertainty of set-membership type which output is available with a limited accuracy. In this case the feedback constructed is rather a measurement feedback [7] than an output feedback [2], and a problem of set-membership estimation arises. According to classical formulation, feedbacks have to be constructed in advance for all possible future positions of the system. Due to enormous computational burden such closed-loop strategies are rarely calculated even for linear determined problems. In this paper we implement receding horizon control principle and construct an optimal feedback as a result of repeated online optimization under the worst-case uncertainty realization. For linear systems two types of problems are solved in the course of the control process: a) optimal observation problem and b) optimal control problem for a determined control system. Numerical methods for problems a) and b) elaborated by the authors are briefly discussed.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 327–334, 2007.
springerlink.com

2 Optimal Closed-Loop Control Under Uncertainty

On the time interval $T = [t_*, t^*]$ consider the dynamical system

$$\dot{x}(t) = f(x(t), u(t), w(t), t), \tag{1}$$

where $x(t) \in R^n$ is a state of system (1) at an instant t; $u(t) \in R^r$ is a control; $w(t) \in R^p$ is a disturbance; $f : R^n \times R^r \times R^p \times T \to R^n$ is a given function such that equation (1) has a unique solution for any control and disturbance.

The control $u(t)$, $t \in T$, is a discrete function with a sampling period $h = (t^* - t_*)/N$: $u(t) \equiv u(\tau),\ t \in [\tau, \tau + h[,\ \tau \in T_h = \{t_*, t_* + h, \ldots, t^* - h\}$, taking values in a bounded set $U \subset R^r$.

The uncertainties are the disturbance w and the initial state $x(t_*)$. Both are assumed to be bounded: $w(t) \in W \subset R^p$, $t \in T$; $x(t_*) \in X_0 \subset R^n$.

Online information on the behaviour of system (1) arrives at discrete instants $t \in T_h$ through measurements of the sensor

$$y(t) = g(x(t), t) + \xi(t), \tag{2}$$

with continuous function $g : R^n \times T \to R^q$, and bounded measurement errors $\xi(t)$: $\xi(t) \in \Xi \subset R^q$, $t \in T_h$.

Let Y_τ be a totality of all signals $y_\tau(\cdot)$ that can be obtained in system (1), (2) by the instant τ.

A vector functional

$$u = u(\tau, y_\tau(\cdot)),\ y_\tau(\cdot) \in Y_\tau(u),\ \tau \in T_h, \tag{3}$$

depending on positions $(\tau, y_\tau(\cdot))$ is called a *feedback* for problem (1), (2).

Let $X(t, u, y)$, $t \in T$, be a totality of all trajectories of system (1), closed by feedback (3), such that they are consistent with a signal $y(\cdot) = (y(t), t \in T_h)$. Let $X(\tau, u) = \bigcup X(\tau, u, y)$, $y(\cdot) \in Y_\tau(u)$, $\tau \in T_h$.

Feedback (3) is called *admissible* if $u(\tau, y_\tau(\cdot)) \in U$, $X(\tau, u) \neq \emptyset$, $\tau \in T_h$, and $X(t^*, u) \subset X^*$, where $X^* \subset R^n$ is a given terminal set. Thus, the admissible feedback guarantees that system (1) reaches the set X^* at the moment t^* despite uncertainties in (1), (2).

Let the cost of the admissible feedback (3) be evaluated as $J(u) = \min c'x$, $x \in X(t^*, u)$. The admissible feedback $u^0(\tau, y_\tau(\cdot)), y_\tau(\cdot) \in Y_\tau,\ \tau \in T_h$, is called *optimal* if $J(u^0) = \max J(u)$, where maximum is calculated over all admissible feedbacks.

Control u^0 is the optimal closed-loop control strategy. It gives the best result under the worst conditions (optimal guaranteed result). Synthesis problem in classical formulation implies the construction of optimal feedback $u^0(\tau, y_\tau(\cdot))$ for all possible positions $(\tau, y_\tau(\cdot))$. This problem is computationally intractable even for linear optimal control problems for determined systems. One approach to overcome this difficulty [8] is to apply receding horizon (or, more precisely, decreasing horizon) technique. Repeated online solution of optimal control problems formulated for a current position $(\tau, y^*_\tau(\cdot))$ of (1), (2) allows to construct a feedback along a realized path $(\tau, y^*_\tau(\cdot))$, $\tau \in T_h$. Effort on treating admissible but not realized positions $(\tau, y_\tau(\cdot))$ is not spent.

In the sequel we concentrate on a linear optimal control problem with parametric uncertainty. The feedback will be constructed on the base of optimal open-loop control strategies. Note that closed-loop formulation as in [5, 6] for the problem under consideration (on finite-time interval T) results again in construction of the functional $u^0(\tau, y_\tau(\cdot))$ for all $y_\tau(\cdot) \in Y_\tau$, $\tau \in T_h$.

3 Optimal Online Control with Open-Loop Strategies

Consider a linear time-varying control system and a linear sensor:

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) + w(t), \quad y(t) = C(t)x(t) + \xi(t) \tag{4}$$

with piecewise continuous matrix functions $A(t) \in R^{n\times n}$, $B(t) \in R^{n\times r}$, $C(t) \in R^{q\times n}$, $t \in T$.

Let the disturbance $w(t)$, $t \in T$, and the initial state $x(t_*)$ have the form

$$w(t) = M(t)v, t \in T; \quad x(t_*) = x_0 + Gz,$$

where $M(t) \in R^{p\times l}$, $t \in T$, is a piecewise continuous matrix function, $x_0 \in R^n$, $G \in R^{n\times k}$; $v \in R^l$ and $z \in R^k$ are unknown bounded parameters:

$$v \in V = \{v \in R^l : w_* \leq v \leq w^*\}; \; z \in Z = \{z \in R^k : \; d_* \leq z \leq d^*\}.$$

Drawing analogy with stochastic uncertainty, we call the sets Z, V *a priori* distributions of the initial state and the disturbance parameters. A set $\Gamma = Z \times V$ is called the *a priori* distribution of the parameters $\gamma = (z, v)$.

Let $U = \{u \in R^r : u_* \leq u \leq u^*\}$, $\Xi = \{\xi \in R^q : \xi_* \leq \xi \leq \xi^*\}$, $X^* = \{x \in R^n : g_{*i} \leq h_i' x \leq g_i^*, i = \overline{1,m}\}$, where $h_i \in R^n$, $g_{*i} < g_i^*$.

Now we describe an optimal online control procedure for a particular control process where a signal $y^*(\cdot)$ realizes.

Consider an arbitrary (current) instant $\tau \in T_h$. Suppose that by the instant $\tau \in T_h$ the control function $u^*(t)$, $t \in [t_*, \tau[$, has been fed into the input of (4) and the signal $y^*_{\tau-h}(\cdot)$ has been recorded. At the instant τ the measurement $y^*(\tau)$ is obtained, therefore the signal $y^*_\tau(\cdot)$ is known. This signal contains additional information about the parameter vector $\gamma^* = (z^*, v^*)$ realized in the process. This information is described by an *a posteriori* distribution $\hat{\Gamma}(\tau) = \hat{\Gamma}(\tau; y^*_\tau(\cdot))$ that is a set of all vectors γ consistent with $y^*_\tau(\cdot)$.

In a more rigorous formulation the *a posteriori* distribution $\hat{\Gamma}(\tau)$ consists of all $\gamma \in \Gamma$ for which there exist an initial state $x(t_*) = x_0 + Gz$, a disturbance $w(t) = M(t)v$, $t \in [t_*, \tau[$, and measurement errors $\xi(t) \in \Xi$, $t \in T_h \bigcap [t_*, \tau]$, such that (4) with the control $u^*(t)$, $t \in [t_*, \tau[$, produces the signal $y^*_\tau(\cdot)$.

Let us construct the (*current*) optimal open-loop control $u^0(t|\tau, y^*_\tau(\cdot))$, $t \in [\tau, t^*]$, such that: 1) it transfers system (4) on the terminal set X^* at the instant t^* for every $\gamma \in \hat{\Gamma}(\tau)$; 2) it delivers maximum to the objective functional $J(u)$.

The first condition gives admissibility of the control strategy while the second one ensures its optimality.

It can be shown that a control $u(t|\tau, y^*_\tau(\cdot))$, $t \in [\tau, t^*]$, is admissible if together with $u^*(t)$, $t \in [t_*, \tau[$, it steers the nominal system

$$\dot{x}_0(t) = A(t)x_0(t) + B(t)u(t),\ x_0(t_*) = x_0$$

at the instant t^* on the terminal set

$$X^*_0(\tau) = \{x \in R^n:\ g_{*i} - \hat{\beta}_i(\tau) \leq h'_i x \leq g^*_i - \hat{\alpha}_i(\tau),\ i = \overline{1,m}\}.$$

Here the tightened terminal constraints are determined by the estimates

$$\hat{\alpha}_i(\tau) = \max h'_i x,\ x \in \hat{X}_*(\tau);\quad \hat{\beta}_i(\tau) = \min h'_i x,\ x \in \hat{X}_*(\tau);\quad i = \overline{1,m}, \tag{5}$$

of the *a posteriori* distribution $\hat{X}_*(\tau) = \hat{X}_*(\tau, y^*_\tau(\cdot))$ of terminal states $x_*(t^*)$ of the uncertain system

$$\dot{x}_*(t) = A(t)x_*(t) + w(t),\ x_*(t_*) = Gz,\quad (z, v) \in \hat{\Gamma}(\tau).$$

The (*current*) optimal open-loop control $u^0(t|\tau, y^*_\tau(\cdot))$, $t \in [\tau, t^*]$, therefore, is a solution to the determined optimal control problem

$$\begin{gathered} c'x(t^*) \to \max,\ \dot{x}(t) = A(t)x(t) + B(t)u(t),\ x(\tau) = x_0(\tau), \\ x(t^*) \in X^*_0(\tau),\ u(t) \in U, t \in [\tau, t^*]. \end{gathered} \tag{6}$$

Extremal problems (5) are called the *optimal observation problems* accompanying the optimal control problem under uncertainty. Problem (6) is called *the accompanying optimal control problem.*

The optimal open-loop control is fed into the system until the next measurement is processed: $u^*(t) = u^0(t|\tau, y^*_\tau(\cdot)),\ t \in [\tau + s(\tau), \tau + h + s(\tau + h)[$. Here $s(\tau) < h$ is a delay caused by computational needs. It represents time required to solve $2m$ accompanying optimal observation problems (5) and one accompanying optimal control problem (6).

Remark. On the interval $[t_*, t_* + s(t_*)[$ system (4) is controlled with the (*a priori*) optimal open-loop control $u^0(t)$, $t \in T$, constructed as above on the base of the *a priori* distribution Γ.

If the delay $s(\tau)$ is neglected then $u^*(t)$, $t \in T$, is an optimal control among all feedbacks constructed on the base of open-loop strategies.

Optimal control problem (6) in the class of discrete controls is equivalent to a linear program that can be solved by any standard method. However, for satisfactory performance of the online control procedure the delay $s(\tau)$ must be as small as possible. The reduction of the optimal control problem (6) to linear program and the use of standard methods in this case are not effective. Specially designed algorithms that take into account the repeated optimization performed for every $\tau \in T_h$ are desirable. The authors designed fast numerical methods for optimization of linear and nonlinear control systems [9, 10]. These methods are based on a special parametrization of the solutions of (5), (6).

One can notice that problems (6) solved at the previous instant $\tau - h$ and at the current instant τ are close in the sense that switching points of their optimal open-loop controls $u^0(t|\tau - h, y^*_{\tau-h}(\cdot))$, $t \in [\tau - h, t^*]$, and $u^0(t|\tau, y^*_\tau(\cdot))$, $t \in [\tau, t^*]$, are close. This property resulted in parametrization of the optimal open-loop controls by their switching points. Indirect methods for optimal control problems require solution of boundary value problems, therefore primal and adjoint equations are integrated on the whole control interval and on each iteration. Parametrization proposed in [9] allows integration of adjoint systems only on short intervals where the switching points move. As a result the method performs fast corrections of the switching points of the optimal open-loop control $u^0(t|\tau - h, y^*_{\tau-h}(\cdot))$, $t \in [\tau - h, t^*]$, to switching points of $u^0(t|\tau, y^*_\tau(\cdot))$, $t \in [\tau, t^*]$ (see [9] for the details and a numerical example in section 5 for complexity estimate). Similarly, a special parametrization of solutions of optimal observation problems with a small number of parameters and fast algorithms for their solution were proposed in [11].

4 Two-Mode Algorithm for Problems with Large Uncertainty

It is well known that feedbacks based on optimal open-loop strategies may fail to provide a feasible solution for positions where optimal closed-loop controls exist. Here we discuss one approach to overcome the feasibility problem when using the procedure described in section 3.

We take into account that uncertainty reduces in the course of the control process, i.e. that $\hat{\alpha}_i(\tau) - \hat{\beta}_i(\tau)$, $\tau \in T_h$, is a decreasing function. This fact obviously follows from observation that $\hat{X}_*(\tau) = \hat{X}_*(\tau - h) \cap \{x \in R^n : x = x(t^*)$ consistent with $y^*(\tau)\}$. Thus, $\hat{X}_*(\tau) \subseteq \hat{X}_*(\tau - h)$ and $\hat{\alpha}_i(\tau - h) \geq \hat{\alpha}_i(\tau)$, $\hat{\beta}_i(\tau - h) \leq \hat{\beta}_i(\tau)$, $i = \overline{1, m}$. Consequently $X^*_0(\tau) \supseteq X^*_0(\tau - h)$.

Let $\tau_* \in T_h$ be such that $\hat{\alpha}_i(\tau) - \hat{\beta}_i(\tau) > g^*_i - g_{*i}$ for all $\tau < \tau_*$ and $\hat{\alpha}_i(\tau_*) - \hat{\beta}_i(\tau_*) \leq g^*_i - g_{*i}$. If accompanying optimal control problem (6) for $\tau = \tau_*$ has a solution, then as follows from the discussion above, problem (6) has admissible (and optimal) controls for any $\tau > \tau_*$. Thus, starting from the moment τ_* the control process can be performed as described in section 3.

The following two-mode algorithm is proposed to design a control strategy for the whole control interval T. At the *first mode* ($\tau < \tau_*$) find a minimal neighborhood of the set $X^*_0(\tau)$ that can be reached by the instant t^* despite the uncertainty. To this end the following extremal problem is solved

$$\rho \to \min,\ \dot{x}(t) = A(t)x(t) + B(t)u(t),\ x(\tau) = x_0(\tau), \qquad (7)$$
$$x(t^*) \in X^*_\rho(\tau),\ u(t) \in U, t \in [\tau, t^*].$$

Here $X^*_\rho(\tau) = \{x \in R^n : g_{i*} - \hat{\beta}_i(\tau) - \rho \leq h'_i x \leq g^*_i - \hat{\alpha}_i(\tau) + \rho, i = \overline{1, m}\}$ is a terminal set with boundaries adjusted by a parameter ρ.

Problem (7) always has a solution $\rho^0(\tau)$, $u^0_\rho(t|\tau, y^*_\tau(\cdot))$, $t \in [\tau, t^*]$, resulting in existence of admissible controls in an auxiliary optimal control problem with relaxed terminal constraints

$$c'x(t^*) \to \max,\ \dot{x}(t) = A(t)x(t) + B(t)u(t),\ x(\tau) = x_0(\tau), \qquad (8)$$
$$x(t^*) \in X^*_{\rho+\varepsilon}(\tau),\ u(t) \in U, t \in [\tau, t^*].$$

Here $\rho = \rho^0(\tau)$ and $\varepsilon > 0$ is a small parameter.

The optimal open-loop control $u^0_{\rho+\varepsilon}(t|\tau, y^*_\tau(\cdot))$, $t \in [\tau, t^*]$, of problem (8) is fed to the input of system (4) on the interval $[\tau + s(\tau), \tau + h + s(\tau + h)[$.

If $\rho^0(\tau) \leq 0$, then $\tau = \tau_*$ and accompanying optimal control problem (6) has a solution. The controller switches to the *second mode* and the process follows the procedure described in section 3.

Remark. Another approach to avoid feasibility problems and incorporate closed-loop strategies can be proposed as a generalization of a closable feedback introduced in [12] for uncertain linear control problems with exact measurements of the states. The closable feedback is a combination of open-loop and closed-loop predictions; it takes into account the fact that in the future more measurements will be available, but considers only a small number of them to avoid enormous computational burden. The closable measurement feedback for problem under consideration will be developed elsewhere.

5 Example

On the time interval $T = [0, 15]$ consider the following system (half-car model):

$$\ddot{x} = -2.1x + 0.31\varphi - u_1 + u_2 + w_1, \qquad (9)$$
$$\ddot{\varphi} = 0.93x + 6.423\varphi + 1.1u_1 + 0.9u_2 + w_2.$$

with known states $x(0) = 0.1$, $\varphi(0) = 0$; unknown velocities $\dot{x}(0) = z_1$, $\dot{\varphi}(0) = z_2$; and disturbances $w_1(t) = v_1 \sin(4t)$, $w_2(t) = v_2 \sin(3t)$, $t \in T$; where $|z_1| \leq 0.1$, $|z_2| \leq 0.33$, $|v_i| \leq 0.01$, $i = 1, 2$.

Let the sensor at $t \in T_h = \{0, h, \ldots, 15 - h\}$, $h = 0.02$, measure values

$$y_1 = -x + 1.1\varphi + \xi_1,\ y_2 = x + 0.9\varphi + \xi_2,$$

where $\xi_i = \xi_i(t)$, $|\xi_i(t)| \leq 0.01$, $t \in T_h$, $i = 1, 2$, are bounded errors.

The aim of the control process is to steer system (9) on the sets $X^* = \{x \in R^2 : |x_1| \leq 0.05, |x_2| \leq 0.1\}$; $\Phi^* = \{\varphi \in R^2 : |\varphi_1| \leq 0.05, |\varphi_2| \leq 0.2\}$; using bounded controls $0 \leq u_i(t) \leq 0.02$, $t \in T$, $i = 1, 2$, while minimizing

$$J(u) = \int_0^{15} (u_1(t) + u_2(t))dt.$$

Let a particular control process be generated by the following parameters and errors: $z_1^* = -0.1$, $z_2^* = 0.33$; $v_1^* = -0.005$, $v_2^* = 0.01$; $\xi_1^*(t) = 0.01\cos(2t)$, $\xi_2^*(t) = -0.01\cos(4t)$, $t \in T_h$.

In the control process under consideration the optimal value of the objective function was equal to 0.1046290478. Figure 1 presents the estimates $\hat{\alpha}_i(\tau)$, $\hat{\beta}_i(\tau)$, $i = \overline{1,4}$; $t \in [0, 2]$. Figure 2 shows the optimal control function $u^*(t)$, $t \in T$, and the projections on the phase planes $x\dot{x}$ and $\varphi\dot{\varphi}$ of the corresponding optimal trajectories.

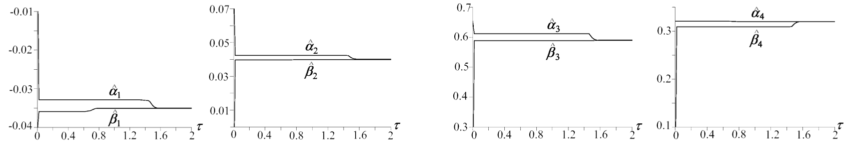


Fig. 1. Linear estimates $\hat{\alpha}_i(\tau)$, $\hat{\beta}_i(\tau)$, $\tau \in T_h$, $i = \overline{1,4}$

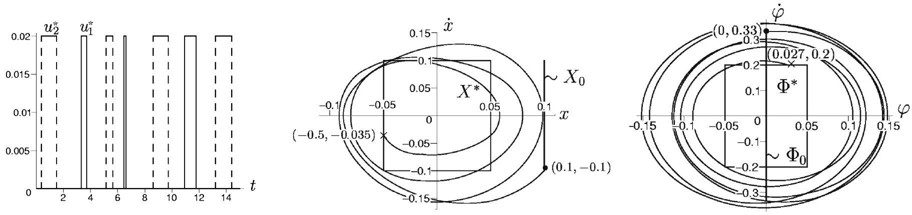


Fig. 2. Optimal control and projections of the optimal trajectories

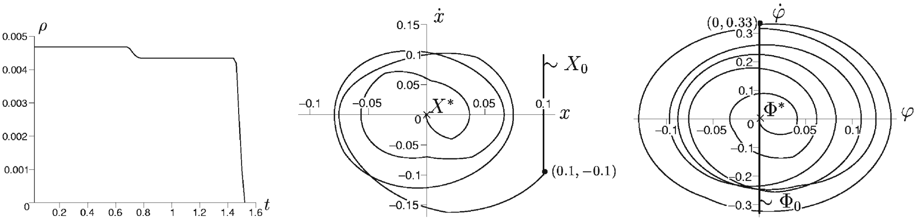


Fig. 3. Two-mode algorithm: estimate $\rho(t)$, $t \in T_h$, and optimal trajectories

To estimate the delay $s(\tau)$ assume that σ is maximal time required to integrate primal or adjoint equations on the whole interval T. We used methods from [9, 11] to perform online corrections of solutions of optimal control and observation problems. The maximal length of the intervals, where differential equations were integrated, was $l_{\max} = 0.043t^*$ (4.3% of the control interval). If σ is such that $\sigma l_{\max} < h$, then system (9) can be controlled in real time.

To demonstrate the two-mode method new (substantially smaller) terminal sets $X^* = \{x \in R^2 : |x_i| \le 10^{-3}, i = 1, 2\}$, $\Phi^* = \{\varphi \in R^2 : |\varphi_i| \le 10^{-3}, i = 1, 2\}$ were chosen as well as new bounds on controls: $0 \le u_i(t) \le 0.05$, $t \in T$, $i = 1, 2$. Under these conditions accompanying optimal control problem (6) on the interval [0,1.52] had no solution and the first mode was performed. The results are presented in Figure 3.

6 Conclusion

In this paper we have considered a problem of synthesis of optimal feedbacks for dynamical systems under uncertainty. For a linear time-varying system with parametric uncertainty and inexact measurements of output an online control algorithm based on open-loop strategies has been presented. It has been shown that in the course of a control process problems of two types are solved: a) optimal observation problem that consists in set-membership estimation of terminal states of an uncertain dynamical system without control and b) optimal control problem for a nominal control system. Numerical methods of their solution has been discussed. One approach to tackle a feasibility problem arising as a result of open-loop controls has been presented.

Acknowledgements

This research was partially supported by the State Program for Applied Research of Belarus. The third author would like to thank Prof. F. Allgöwer for invitation and support for International Workshop on Assessment and Future Directions of NMPC.

References

[1] Mayne DQ, Rawlings JB, Rao CV, Scokaert POM (2000) Constrained model predictive control: stability and optimality. Automatica 26:789–814

[2] Findeisen R, Imsland L, Allgöwer F, Foss BA (2003) State and output feedback nonlinear model predictive control: An overview. Eur J of Control 9:190–206

[3] Magni L, Scattolini R (2005) Robustness and robust design of MPC for nonlinear discrete-time systems. In: this volume

[4] Chisti L, Rossiter JA, Zappa G (2001) Systems with persistent disturbancies: predictive control with restristed constraints. Automatica 37:1119–1028

[5] Lee JH, Yu Z (1997) Worst-case formulation of model predictive control for systems with bounded disturbances. Automatica 33:763–787

[6] Scokaert POM, Mayne DQ (1998) Min-max model predictive control for constrained linear systems. IEEE Trans on Automatic Control 43:1136–1142

[7] Kurzhanski AB (2003) The Principle of optimality in measurement feedback control for linear systems. In Rantzer A, Byrnes CI (eds) LNCIS 286: 193–202

[8] Gabasov R, Kirillova FM, Prischepova SV (1995) Optimal Feedback Control. In: Thoma M (ed) LNCIS 208. Springer, Berlin Heidelberg New York

[9] Gabasov R, Kirillova FM, Balashevich NV (2000) Open-loop and Closed-loop Optimization of Linear Control Systems. Asian Journal of Control 2:155 – 168

[10] Balashevich NV, Gabasov R, Kalinin AI, Kirillova FM (2002) Optimal Control of Nonlinear Systems. Comp Math and Math Physics 42:931 – 956

[11] Gabasov R, Dmitruk NM, Kirillova FM (2002) Optimal Observation of Nonstationary Dynamical Systems. J of Comp and Syst Sci Int 41:195–206

[12] Gabasov R, Kirillova FM, Balashevich NV (2004) Guaranteed On-line Control for Linear Systems under Disturbances. Functional Diff Eq 11:341 – 361

State Estimation Analysed as Inverse Problem

Luise Blank

NWF I - Mathematik, University of Regensburg, Germany
luise.blank@mathematik.uni-regensburg.de

Summary. In dynamical processes states are only partly accessible by measurements. Most quantities must be determined via model based state estimation. Since in general only noisy data are given, this yields an ill-posed inverse problem. Observability guarantees a unique least squares solution. Well-posedness and observability are qualitative behaviours. The quantitative behaviour can be described using the concept of condition numbers. which we use to introduce an observability measure. For the linear case we show the connection to the well known observability Gramian. For state estimation regularization techniques concerning the initial data are commonly applied in addition. However, we show that the least squares formulation is well-posed, avoids otherwise possibly occuring bias and that the introduced observability measure gives a lower bound on the conditioning of this problem formulation.

Introducing possible model error functions we leave the finite dimensional setting. To analyse in detail the influence of the regularization parameters and of the coefficients of the model, we study, as a start, linear state equations as constraints, which appear nearly always as a subproblem of the nonlinear case. We show that state estimation formulated as optimization problem omitting regularization of the initial data leads to a well-posed problem with respect to L_2- and L_∞- disturbances. If the introduced measure of observability is low, the arising condition numbers with respect to the L_2-norm can be arbitrarily large. Nevertheless, for the probably in praxis more relevant L_∞-norm perturbations yield errors in the initial data bounded independently of the system matrix.

1 Introduction

In application the state of a process has to be estimated given noisy data over a past time horizon. These data correspond only to a few state functions, so called output functions. The coupling with all remaining states is given by model equations. This inverse problem is in general ill-posed, since the measurements are noisy and the corresponding continuous signals do not fulfill the model equations. Hence, the existence requirement for well-posedness in the sense of Hadamard is violated. Considering the least squares solution the uniqueness is guaranteed by the observability of the system given by the model equation. The third requirement of well-posedness, namely stability, depends crucially on the norms, which are chosen to measure the disturbances. For state estimation stability may be present in some cases. However, as soon as model error functions are introduced

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 335–346, 2007.
springerlink.com

this is not any longer true. Additional regularization is required. Assuming now well-posedness, there arises the next question: how 'well' does the solution behave in case of disturbances? The corresponding question for the system itself is, how 'well' is the system observable?Both questions ask for the relation of the output error to the input error. Condition numbers is a general formulated mathematical concept for operators answering this question. Hence, we use this concept to derive a definition of observability measure. Alltogether, this paper discusses for state estimation the well-posedness of the regularized least squares problem formulation, the conditioning, i.e. error propagation, and the influencing observability measure. For all three issues the chosen norms play an important role.

The structure of the paper is the following. First we resume the definitions of well-posedness, the possibilities to overcome ill-posedness and the concept of condition numbers. We discuss for the system observability and introduce an observability measure based on condition numbers. For linear model equations and the L_2-norm this observability measure depends on the observability Gramian. Then the well-posedness of the least squares formulation for linear estimation is shown requiring only observability.Additional regularization of the initial value yields unnecessary bias. In the following section we extend the model equations linearly by possible model error functions. The least squares problem formulation, now necessarily regularized with respect to the error functions, gives an optimization problem, for which we state the first order necessary condition. Then we restrict the analysis to linear state equations omitting inequality constraints. They appear usually as subproblems solving the nonlinear problem and their analysis enhances already some of the main features we face also for the nonlinear case. In particular, we study the influence of the measure of observability and of the regularization parameter. Omitting regularization of the initial data we derive well-posedness for the optimization formulation with respect to the L_2-norm and with respect to the L_∞-norm. However, while for one state only we see that the problem is well-conditioned with respect to the L_∞-norm, a low observability measure may result into an ill-conditioned problem with respect to the L_2-norm independent of the regularization parameter. In the last section we draw conclusions and emphasize the issue of the appropriate choice of norms concerning data errors and state errors.

2 Well-Posedness, Condition Number and Observability Measure

Typically the noisy measurements $z(t_i) \in \mathbb{R}^{n_y}$ at discrete times are preprocessed. Most algorithms are based on the assumption to have an underlying function corresponding to the discrete data, e.g. for many filtering techniques the Fourier transformation is used at some stage. Hence, it is appropriate to assume a preprocessing of the data on a horizon $[t_0, t_0 + H]$ to a still noisy function $z \in L_2([t_0, t_0 + H], \mathbb{R}^{n_y})$. These data z correspond to a few states, called output functions which we denote with y. The output functions y are coupled with all states x and their initial values $x_0 = x(t_0) \in \mathbb{R}^{n_x}$ by model equations. Usually the number of output functions n_y is far less than the number of state functions n_x. Given the model equations one can determine from the initial values the states and therefore the outputs. Hence they define an operator $K : x_0 \to y$. In general we have $z \notin \mathcal{R}(K)$ (the range of K). This violates the first condition of well-posedness in the sense of Hadamard [8]:

2.1 Ill-Posed Problems, Regularization and Condition Numbers

Definition 1. *Given an operator $K : X \to Y$ where X and Y are normed spaces, then the equation $Kx = y$ is well-posed in the sense of Hadamard iff*
1. Existence: there exists for all $y \in Y$ a solution $x \in X$; (K surjective).
2. Uniqueness: there is at most one solution $x \in X$; (K injective).
3. Stability: x depends continuously on y, i.e.
$\|Kx_n - Kx\|_{\boldsymbol{Y}} \to 0 \Rightarrow \|x_n - x\|_{\boldsymbol{X}} \to 0$*; ($K^{-1}$ continuous).*
The equation is ill-posed if one of these properties does not hold.

It is important to specify the spaces as well as the topologies of the spaces, i.e. the norms $\|\cdot\|_{\boldsymbol{X}}$ and $\|\cdot\|_{\boldsymbol{Y}}$. The problem can be well-posed using one set of norms and ill-posed in another set of norms. If the problem is ill-posed there are several remedies, of which we recall only some relevant in our context. Assume that X and Y are Hilbert-spaces (i.e. there exists a scalar product; e.g. the space L_2) and $K : X \to Y$ is linear and compact then x is called *least-squares solution* (best fit) if x is the solution of $\min_{\boldsymbol{x}\in\boldsymbol{X}} \|Kx - z\|_{\boldsymbol{Y}}$. Moreover, it holds: x is the least squares solution if and only if the *normal equation* $K^*Kx = K^*z$ holds, where K^* denotes the adjoint operator. In case of a finite-dimensional space X this ansatz overcomes the failure of existence. Uniqueness is not necessarily an issue in our context since we require observability of the system given by the model equations (see later). However, otherwise one can use the Moore-Penrose inverse, also called *generalized inverse*, which is the least squares solution of minimal norm [8], if there exists a least squares solution. For finite-dimensional X the generalized inverse is given by $K^\dagger := (K^*K)^{-1}K^*$. However, the generalized inverse does not overcome the lack of continuity in general. Regularization techniques have to be applied. Here, we can distinguish roughly speaking three kinds of approaches, namely the Tikhonov regularization, which we consider here, iterative regularization methods and regularization by discretization (projection). An application of the latter in state estimation can be found i.e. in [3]. Tikhonov regularization shifts the spectrum of K^*K and leads to the *regularized generalized inverse* $R_d := (dI + K^*K)^{-1}K^*$, which is bounded, with a regularization parameter $d > 0$. Solving $R_d x = z$ is equivalent to the minimization problem

$$\min_{\boldsymbol{x}\in\boldsymbol{X}} \|Kx - z\|_{\boldsymbol{Y}}^2 + d\|x\|_{\boldsymbol{X}}^2.$$

The operator R_dK should converge pointwise to the identity for $d \to 0$. Moreover, the choice of the regularization parameter d should give the best compromise between data and regularization error, i.e. let $\|z - z^\delta\| \le \delta$ and $x^{d,\delta} = R_d z^\delta$ then $\|x^{d,\delta} - x\| \le \|R_d(z^\delta - z)\| + \|R_d z - x\| \le \|R_d\|\delta + \|R_d z - x\|$, should be minimal. The first term is called data error and the second regularization error.

This is a non trivial task but will not be discussed in this paper, instead we refer to the literature, e.g. [8].

Now let us assume that $Kx = y$ is a well-posed problem. Then K^{-1} exists and is bounded with respect to the chosen norms. That means, the equation is stable, which is a qualitative statement. The mathematical concept of condition number is quantitative. It measures the possible error propagation with respect to the absolute or relative error [7].

Definition 2. *Considering the problem given y determining the solution x of $Kx = y$ and let $\|\tilde{y} - y\|_{\boldsymbol{Y}} \to 0$:*
1. the absolute condition number is the smallest number $\kappa_{abs}(y) > 0$ with
$$\|\tilde{x} - x\|_{\boldsymbol{X}} = \|K^{-1}\tilde{y} - K^{-1}y\|_{\boldsymbol{X}} \leq \kappa_{abs}(y)\|\tilde{y} - y\|_{\boldsymbol{Y}} + o\left(\|\tilde{y} - y\|_{\boldsymbol{Y}}\right),$$
2. the relative condition number is the smallest number $\kappa_{rel}(y) > 0$ with
$$\|\tilde{x} - x\|_{\boldsymbol{X}}/\|x\|_{\boldsymbol{X}} \leq \kappa_{rel}(y)\|\tilde{y} - y\|_{\boldsymbol{Y}}/\|y\|_{\boldsymbol{Y}} + o\left(\|\tilde{y} - y\|_{\boldsymbol{Y}}/\|y\|_{\boldsymbol{Y}}\right).$$

The problem is called well-conditioned if κ is small and ill-conditioned for large κ. For linear K we have
$$\kappa_{abs}(y) \leq \|K^{-1}\|_{\boldsymbol{Y}\to\boldsymbol{X}} \text{ and } \kappa_{rel}(y) \leq \|K\|_{\boldsymbol{X}\to\boldsymbol{Y}}\|K^{-1}\|_{\boldsymbol{Y}\to\boldsymbol{X}}.$$
If K is a matrix, the condition number is defined as the latter namely $cond(K) := \|K\|\|K^{-1}\|$, where commonly the l_2-norms are used.

2.2 Observability Measure

For state estimation on the horizon $[t_0, t_0 + H]$ the operator $K : x_0 \longmapsto y$ is given by the *model equations*:

$$\text{State equations:} \qquad G\dot{x} - f(x, u, p) = 0, \quad x(t_0) = x_0 \tag{1}$$
$$\text{Output equations:} \qquad y - Cx = 0 \tag{2}$$

The system (1)-(2) is called *observable*, if for any given u and p the initial state x_0 can be uniquely determined from the output y [11]. Hence, $K : x_0 \longmapsto y$ is injective for fixed u, p and K^{-1} exists on $\mathcal{R}(K)$. The space X is the finite-dimensional space $\mathbb{R}^{n_x}$. Observability is the qualitative behaviour that a difference in the states shall be seen in the outputs. The observability measure shall quantify this statement, hence we consider
$$\|y - \tilde{y}\| \geq c\|x_0 - \tilde{x}_0\|$$
or a relative measurement independent of the scaling
$$\|y - \tilde{y}\|/\|y\| \geq c\|x_0 - \tilde{x}_0\|/\|x_0\|.$$

As larger c as better the observability measure. This suggest the use of the condition number $\kappa = 1/c$ of the problem given y determining the solution of $Kx_0 = y$. The evaluation of the conditioning is mentioned also in [1] in preference to the yes/no answer of observability.

Definition 3. *The absolute and the relative measure of observability of x_0 are defined as $1/\kappa_{abs}$ and $1/\kappa_{rel}$.The system is called well observable for x_0, if $\kappa = 1/c$ is small, and has a low observability measure for large κ.*

For linear model equations the corresponding operator K is affine. Let us first consider linear K, i.e. the model equations are linear and p and $u = 0$. Without loss of generality we consider in the rest of the paper only the case $t_0 = 0$. Thus we have with

$$\dot{x} - Ax = 0, \quad x(0) = x_0, \quad y - Cx = 0 \tag{3}$$

$$\Rightarrow Kx_0 = Ce^{\boldsymbol{A}t}x_0 \tag{4}$$

Choosing now as norm on Y the $L_2([0,H])^{n_y}$-norm we obtain

$$\|Kx_0\|_{L_2}^2 = x_0^T \int_0^H (e^{\boldsymbol{A}t})^T C^T C e^{\boldsymbol{A}t}\, dt\, x_0 = x_0^T \mathcal{G}(H) x_0$$

where the matrix $\mathcal{G}(H) \in \mathbb{R}^{n_x \times n_x}$ is the known finite time observability Gramian (e.g. [6, 11, 12]).

Lemma 1. *Let the system be observable, then:*
a.) The observability Gramian $\mathcal{G}(H) = \int_0^H (e^{\boldsymbol{A}t})^T C^T C e^{\boldsymbol{A}t}\, dt \in \mathbb{R}^{n_x \times n_x}$ is symmetric positive definite, and therefore invertible.
b.) Let v be a normed real eigenvalue of A to an eigenvalue $\alpha \in \mathbb{R}$. Then $\|\mathcal{G}(H)\|_2$ is large for large α and for a long horizon $[0,H]$, while $\|\mathcal{G}(H)^{-1}\|_2$ is large if $-\alpha$ is large or $\|Cv\|_{l_2}$ is small or if the horizon is short.

Proof: a.) Symmetry is obvious. Given $v \neq 0$ then $y(t) = Ce^{\boldsymbol{A}t}v \not\equiv 0$ since the system is observable. Hence, $v^T\mathcal{G}v = \int_0^H y^T(t)y(t)\, dt = \|y\|_{L_2}^2 > 0$.
b.) Let v and α fulfill the assumption, then

$$v^T\mathcal{G}v = \|e^{\alpha t}\|_{L_2(0,H)}^2 \|Cv\|_{l_2}^2 = \frac{e^{2\alpha H} - 1}{2\alpha}\|Cv\|_{l_2}^2. \tag{5}$$

With $||\mathcal{G}||_2 = \max_{\boldsymbol{v}\in\mathbb{R}^{n_x}}(v^T\mathcal{G}v)/(v^Tv)$, $\|\mathcal{G}^{-1}\|_2 = \max_{\boldsymbol{v}\in\mathbb{R}^{n_x}}(v^Tv)/(v^T\mathcal{G}v)$ follows the assertion. ■

Using the l_2-norm for $X = \mathbb{R}^{n_x}$ it follows for K:

$$\|K\|_{l_2 \to L_2}^2 = \sup_{\boldsymbol{x}_0 \in \mathbb{R}^{n_x}} \frac{\|Kx_0\|_{L_2}^2}{\|x_0\|_{l_2}^2} = \sup_{\boldsymbol{x}_0 \in \mathbb{R}^{n_x}} \frac{x_0^T\mathcal{G}x_0}{x_0^T x_0} = ||\mathcal{G}||_2 \tag{6}$$

$$\|(K\,|_{\mathcal{R}(\boldsymbol{K})})^{-1}\|_{L_2 \to l_2}^2 = \sup_{\boldsymbol{x}_0 \in \mathbb{R}^{n_x}} \frac{\|x_0\|_{l_2}^2}{\|Kx_0\|_{L_2}^2} = ||\mathcal{G}^{-1}||_2 \tag{7}$$

For linear systems with not necessarily p and $u = 0$ we need to consider for the condition numbers $\|K\tilde{x_0} - Kx_0\|_{L_2}^2 = (\tilde{x_0} - x_0)^T\mathcal{G}(\tilde{x_0} - x_0)$. Hence, having $\sqrt{\|\mathcal{G}^{-1}\|_2}\|y - \tilde{y}\|_{L_2} \geq \|x_0 - \tilde{x}_0\|_{l_2}$ for all x_0, we can define for linear systems -like condition numbers for matrices- a observability measure independent of the state x_0:

Definition 4. *The absolute, respectively the relative observability measure with respect to the l_2 and L_2-norms for linear systems is given by*

$$1/\sqrt{||\mathcal{G}^{-1}||_2} \quad \textit{respectively} \quad 1/\sqrt{cond(\mathcal{G})}.$$

This definition is in agreement with the Gramian based measure in [13], where one considers a infinite horizon. There, several measures are proposed and compared, which are all rather based on the various tests for observability than motivated by error propagation.

Lemma 1 immediately shows that long horizons are better for observability reasons. It also reassures that the eigenvectors of system matrix A should not be close to the null-space of the output matrix C. For rapidly decaying systems it is confirmed that it is difficult to determine the initial value exactly, while for the forward problem K the value at the end point is sensitive to $\alpha \gg 0$.

2.3 State Estimation as a Least Squares Problem

Going back to the inverse problem of state estimation we obtain

Theorem 1. *For a observable system the problem formulation on* $[0, H]$

$$\min \|y - z\|_{L_2}^2 + d\|x_0 - x_0^{ref}\|_{l_2}^2 \;\; s.t.\; \dot{x} - Ax = u, x(0) = x_0, y - Cx = 0 \qquad (8)$$

is well-posed for all $d \geq 0$*, and the solution is given by*

$$x_0 = (\mathcal{G}(H) + dI)^{-1}[\int_0^H e^{\boldsymbol{A}^T t} C^T \{z(t) - Ce^{\boldsymbol{A}t} \int_0^t e^{-As} u(s)\, ds\}\, dt + dx_0^{ref}]. \quad (9)$$

Regularization of the initial data ($d \neq 0$) *is not necessary, and leads to bias if inexact reference values are used.*

Proof: Setting $\hat{z} = z - C\int_0^t e^{\boldsymbol{A}(t-s)} u(s)\, ds$ we have

$$\min \|y - z\|_{L_2} + d\|x_0 - x_0^{ref}\|_{l_2}^2 = \min \|\hat{y} - \hat{z}\|_{L_2} + d\|x_0 - x_0^{ref}\|_{l_2}^2$$

where $\hat{y}$ fulfills the model equations with $\hat{u} = 0$. Then, dropping for convenience the $\hat{}$ we have the equivalence of (8) to the normal equation: $(K^*K + dI)x_0 = K^*z + dx_0^{ref}$ with $K^*z = \int_0^H e^{\boldsymbol{A}^T t} C^T z(t) dt$ since $(\xi_0, K^*y)_{l_2} = (K\xi_0, y)_{L_2} = \xi_0^T \int_0^H (e^{\boldsymbol{A}t})^T C^T y(t)\; dt$. Hence $\mathcal{G} = K^*K$ which is invertible for observable systems, and we obtain (9). Moreover, since $\mathcal{G}$ is finite-dimensional it has a bounded inverse. Therefore, (8) is a well-posed problem even for $d = 0$, i.e. without regularizing the initial value. Given a noise free signal z, there exists a unique x_0^{exact} s.t. $Kx_0^{exact} = z$. We obtain

$$x_0 = x_0^{exact} + (\mathcal{G} + dI)^{-1} d(x_0^{ref} - x_0^{exact}) \qquad (10)$$

which answers the question of bias. ■

(Remark: the result concerning the bias is not in contrast to the probabilistic ansatz leading to the Kalman filter, since there one would choose also $d = 0$ for noise free signals [10]. If one may wish to include a priori knowledge about a linear process appropriate priors and consequently possible choices for d with a small influence on the solution can be found e.g. in [5]. For $d = 0$ see also [6] for extension to time varying systems and the connection to the Kalman filter.)

While the least squares formulation is well-posed, it is not necessarily well-conditioned.

Corollary 1. *The condition number of (8) with $d = 0$, i.e. of the least squares problem $K^\dagger = \mathcal{G}^{-1}K^*$ with respect to the l_2 and L_2-norms obeys*

$$\kappa_{abs} \geq \|(K\left|_{\mathcal{R}(\boldsymbol{K})}\right)^{-1}\|^2_{L_2 \to l_2} = \sqrt{\|\mathcal{G}^{-1}\|_2} \quad \textit{and} \quad \kappa_{rel} \geq \sqrt{cond(\mathcal{G})}.$$

Proof: Let v be the normed eigenvector to the smallest eigenvalue of $\mathcal{G}$, then $\|\mathcal{G}^{-1}\|_2 = \|\mathcal{G}^{-1}v\|_2$. For $z(t) = kCe^{\boldsymbol{A}t}\mathcal{G}^{-1}v$ and $u = 0$ we have $x_0 = k\mathcal{G}^{-1}v$. Furthermore, $\|z\|^2_{L_2} = k^2\|\mathcal{G}^{-1}\|_2$ and therefore $\|x_0\| = \|z\|_{L_2}\sqrt{\|\mathcal{G}^{-1}\|_2}$. With (6) and (7) the assertion holds. ■

Consequently, a low observability measure leads to an ill-conditioned least squares formulation, even though observability provides well-posedness. E.g. we may face large error propagation if the assumptions of Lemma 1b.) hold.

3 Inclusion of Model Error Functions

3.1 Optimality Conditions

In the following we include linearly possible model error functions w in the model equations. With this step we leave the finite dimensional setting. Considering only the least squares solution does not guarantee stability any longer, as the example of the signal $z(t) = \delta \sin \frac{n}{\delta} t$ with the model equations $\dot{x} = ax + w$ and $y = x$ shows for $n \to \infty$. Regularization with respect to w is necessary. As regularization parameters we employ now matrices instead of scalars. For the mathematical consideration it is at this point no issue to distinguish the given parameters p and the controls u. We summarize them to u. Equation (1) for the initial condition can be omitted, since it does not contain any information. Additional inequality constraints reflect safety constraints as well as model verification. Summarized we consider in the following the Tikhonov-type regularized least squares solution of:

$$\min \int_0^H (y-z)^T Q (y-z) + w^T R_{\boldsymbol{w}} w \; dt + \|D^{1/2}(x(0) - x_0^{ref})\|^2_{l_2} \tag{11}$$

$$\text{s.t.} \quad G\dot{x} - f(x,u) - Ww = 0, \quad y - Cx = 0, \quad c(x,u) \geq 0. \tag{12}$$

Obviously, we can substitute y by Cx and reduce the system by y, the output equations and avoid Lagrange multipliers for these. In addition, setting up the necessary first order equations we see w can be eliminated by the Lagrange multiplier with resp. to the DAE's, namely $w = R_{\boldsymbol{w}}^{-1}W^T\lambda$. This is a major reduction in size since $w(t)$ may be in $\mathbb{R}^{n_x}$. Defining $R := WR_{\boldsymbol{w}}^{-1}W^T$ we obtain the following *necessary conditions* (for details see [4], for the linear case without inequality constraints see [10], where also the connection to the Kalman filter is given):

$$G\dot{x} - f(x,u) - R\lambda = 0 \tag{13}$$

$$-\frac{d}{dt}(G^T\lambda) - (\frac{\partial}{\partial \boldsymbol{x}} f(x,u))^T\lambda + C^TQCx + (\frac{\partial}{\partial \boldsymbol{x}} c(x,u))^T\nu = C^TQz \tag{14}$$

$$(G^T\lambda)(0) = D(x(0) - x_0^{ref}) \quad \text{and} \quad (G^T\lambda)(H) = 0 \tag{15}$$

$$\nu^T c(x,u) = 0 \quad c(x,u) \leq 0 \quad \nu \geq 0. \tag{16}$$

In case of ODE's as state constraints with no regularization of the initial state, this yields $w \in H^1$, where H^1 denotes the Sobolev-space of weakly differentiable functions in L_2, and $w = 0$ at the end points. If model error functions are present in all state equations, the Lagrange parameter λ can be eliminated too with $\lambda = R^{-1}\left(G\dot{x} - f(x,u)\right)$. Then, the necessary conditions reduce to a second order DAE system with mixed boundary constraints for the state x only and the equations (16) for the inequality constraints.

3.2 Analysis for Linear ODE's

In the solution process for a nonlinear problem with inequality constraints nearly always optimization problems with linear model equations and without inequality constraints (or an equivalent linear equation system) appear as subproblems. Although for the linear case, in particular with regularization of the initial data, efficient methods as the Kalman filter are well established [10, 11], these methods and its extensions cannot be applied or do not perform sufficiently in the presence of nonlinearity and inequality constraints [9]. However, the study of linear model equations without inequality constraints enhances already some of the main features we face for the treatment of the optimization formulation of the nonlinear state estimation. The goal of the study here is to iluminate the influence of the eigenvalues of the system matrix, the influence of the observability measure and the influence of the regularization parameters. As a start we assume possible model error functions in all state equations. We consider the following problem:

$$\min \int_0^H (Cx - z)^T Q (Cx - z) + w^T R_{\boldsymbol{w}} w \, dt + \|D^{1/2}(x(0) - x_0^{ref})\|_{l_2}^2 \tag{17}$$
$$\text{s.t.} \qquad \dot{x} - Ax - w = u.$$

A detailed analysis of this problem and its results can be found in [4], where they are presented for $H = 1$ and $D = 0$. With a few modifications they can be extended to any $H > 0$. In this paper we summarize some of the main results. To study the properties we choose one of the following three equivalent formulation, which derive from elimination of the error function w using the state equation or from the necessary conditions eliminating the Lagrange parameter λ. The boundary value problem and its weak formulation are not only necessary but also sufficient condition, which is shown later (Theorem 4):

Optimization problem:

$$\min_{\boldsymbol{x} \in H^1} \left\{ \|Q^{\frac{1}{2}}\left(Cx - z\right)\|_{L_2}^2 + \|R^{-\frac{1}{2}}\left(\dot{x} - Ax - u\right)\|_{L_2}^2 + \|D^{\frac{1}{2}}(x(0) - x_0^{ref})\|_{l_2}^2 \right\}. \tag{18}$$

Second order BVP:

$$-R^{-1}\ddot{x} + \left(R^{-1}A - A^TR^{-1}\right)\dot{x} + \left(A^TR^{-1}A + R^{-1}\dot{A} + C^TQC\right)x$$
$$= C^TQz - R^{-1}\dot{u} - A^TR^{-1}u \tag{19}$$

with boundary conditions $\dot{x}(0) - (A + RD)x(0) = u(0) - RDx_0^{ref}$ and $\dot{x}(H) - Ax(H) = u(H)$.

Weak Formulation:

$$\langle\dot{\zeta} - A\zeta, R^{-1}(\dot{x} - Ax)\rangle + \langle C\zeta, QCx\rangle + \zeta^T(0)D(x(0) - x_0^{ref}) \tag{20}$$
$$= \langle C\zeta, Qz\rangle + \langle\dot{\zeta} - A\zeta, R^{-1}u\rangle \quad \text{for all } \zeta \in H^1.$$

In the following we concentrate only on the case $D = 0$, i.e. on the least squares formulation without regularization of the initial data. Among other things we show its well-posedness. With regularization of the initial data this can be studied in the framework of Tikhonov regularization.

For one state function only $(A = \alpha, C = \delta, Q = q, R^{-1} = r)$ one can use the BVP to derive an explicit formula for the solution. Analysing this solution we obtain the following theorem.

Theorem 2

1. *The regularized problem formulation (17) (with $D = 0$), i.e. given z determining x_0 and w, is well-posed.*
2. *Small perturbation of z in the L_2-norm may lead to large error propagation in the initial data x_0 independently of r and q if $-\alpha$ is large.*
3. *For perturbations of z in the L_∞-norm we have bounds for the errors in x_0 and w independently of α.*

For several state functions we use the weak formulation to show well-posedness. As a first step we again study first the case of one state function where observability is not an issue and extend then the result to several state functions. Let us define the symmetric bilinear form $a : H^1(0,H) \times H^1(0,H) \to \mathbb{R}$

$$a(\zeta, x) = \langle\dot{\zeta} - A\zeta, R^{-1}(\dot{x} - Ax)\rangle + \langle C\zeta, QCx\rangle \tag{21}$$

If $A, C \in \mathbb{R}$ then a is positive definite if $R^{-1}, Q > 0$. Hence it defines an operator $\mathcal{S} : H^1(0,H) \longrightarrow (H^1(0,H))'$ with $(\zeta, \mathcal{S}x) := a(\zeta, x)$ and the weak formulation (20) is equivalent to $(\zeta, \mathcal{S}x) = \langle C\zeta, Qz\rangle + \langle\dot{\zeta} - A\zeta, R^{-1}u\rangle$ for all $\zeta \in H^1$.We would like to remark that, not only for the well-posedness it is of interest to study the properties of $\mathcal{S}$ but also for the numerical solution approaches. If we use a Galerkin discretization of (17) then the properties of $\mathcal{S}$ govern to a large extend also the numerical method. For example the condition number of the discretization matrix is influenced by the condition number of $\mathcal{S}$. Using methods of functional analysis one can show the following result in the case of one state function:

Theorem 3. *$\mathcal{S}$ is a linear isomorphism and with the real values $A = \alpha, C = \delta$, $Q = q, R^{-1} = r$ we have*

$$\min(4r, q\delta^2) \leq \quad \|\mathcal{S}\|_{H^1 \to (H^1)'} \quad \leq 2r \max(1, \alpha^2) + q\delta^2, \tag{22}$$

$$\frac{c(\alpha)}{q\delta^2} \leq \|\mathcal{S}^{-1}\|_{(H^1)' \to H^1} \leq \max\left\{\frac{2}{r}, \frac{2\alpha^2+1}{q\delta^2}\right\}, \tag{23}$$

with $c(\alpha) \approx |\alpha|^{3/2}\sqrt{\frac{e^{2H}-1}{2(e^{2H}+1)}}$ for large $|\alpha|$. Hence for fixed regularization parameters r and q cond($\mathcal{S}$) $= \|\mathcal{S}\|_{H^1 \to (H^1)'} \|\mathcal{S}^{-1}\|_{(H^1)' \to H^1}$ is bounded but tends to infinity with $|\alpha| \to \infty$.

The exact value of $c(\alpha) := \|\exp(\alpha\cdot)\|_{H^1} / \|\exp(\alpha\cdot)\|_{(H^1)'}$ is

$$c^2(\alpha) = \frac{(1+\alpha^2)(\alpha+1)^2(\alpha-1)^2(e^H - e^{-H})(e^{2\alpha H}-1)}{(2\alpha+1)(\alpha-1)^2(e^H - e^{-H})(e^{2\alpha H}-1) + 4\alpha^3 e^{-H}(e^{\alpha H} - e^H)^2}.$$

Considering several state functions one has to take into account observability to show positive definiteness and herewith continuity and coercivity of a. We shortly sketch this step while refering for the other arguments to [4]. Given $a(x, x) = 0$ then x is the solution of the system $\dot{x} - Ax \equiv 0$, $x(0) = x_0$ and $(y \equiv) Cx \equiv 0$. Since the system is observable $y \equiv 0$ yields $x_0 = 0$ and consequently $x \equiv 0$. Hence, $a(x, x) > 0$ for $x \not\equiv 0$. Then continuity and coercivity of a yield

Theorem 4. *For any $z, u \in L_2$ the solution x of the weak formulation (20) determines the unique solution of the minimization problem (18).*

Hence for well-posedness only the question of stability has still to be answered. Using the Riesz representation theorem [2] and considering like for one state only and Lemma 1b.) the exponential function we obtain

Theorem 5. *$\mathcal{S} : H^1 \to (H^1)'$ is bounded, has a bounded inverse and*

$$0 < 2/\|R\| \leq \|\mathcal{S}\|_{H^1 \to (H^1)'} \leq 2\|R\|^{-1} \max(1, \|A\|^2) + \|C^T QC\|, \tag{24}$$

$$\max\{c(\alpha)/\|C^T QCv\|_{l_2}\} \leq \|\mathcal{S}^{-1}\|_{(H^1)' \to H^1}. \tag{25}$$

for all normed $v \in \mathbb{R}^{n_x}$ eigenvectors of A with real eigenvalue α. In (24) the lower bound is valid if there exists an $\alpha^2 > 1$. Observability guarantees $Cv \neq 0$.

As a consequence of Theorem 5 and Lemma 1b.) we have:

Corollary 2. *For any fixed regularization cond($\mathcal{S}$) is large, if there exists an in modulo large real eigenvalue of A or if there exists a real eigenvector v of A which is close to the null space of C. If this is the case then the observability measure is low too.*

Nevertheless, the boundedness of the inverse $\mathcal{S}^{-1}$ and the compact inbedding $H^1 \hookrightarrow C^0$ yields

Corollary 3. *Linear state estimation formulated as least squares problem*

$$\min \tfrac{1}{2}\int_0^H (y - z)^T Q(y - z) + w^T R_{\boldsymbol{w}} w \; dt$$

$$s.t. \qquad \dot{x} - Ax - w = u \qquad y - Cx = 0$$

is well-posed, i.e. $\|x - x^\delta\|_{H^1} \leq c\|z - z^\delta\|_{(H^1)'}$ *and, consequently, with a generic constant c,* $\max\{|x_0 - x_0^\delta|, \|w - w^\delta\|_{L_2}\} \leq c\|z - z^\delta\|_{L_2}$ *and* $\|x - x^\delta\|_{C^0} \leq c\|z - z^\delta\|_{L_2} \leq c\|z - z^\delta\|_{L_\infty}$.

4 Conclusions and Questions Concerning the Appropriate Norms

Observability is like well-posedness a qualitative property. Condition numbers quantify the error propagation. We used this concept to define an observability measure. For linear systems we derived the use of the inverse of observability Gramian $\mathcal{G}$. With $\mathcal{G}^{-1}$ one can estimate the minimal difference in the outputs which can be seen for given different initial data. Also we showed that regularization of initial data is not necessary for stability, hence the least squares formulation for state estimation is well-posed. Regularization of initial data would lead to bias. However, a low observability measure leads to an ill-conditioned least squares problem.

Introducing linearly model error functions we leave the finite dimensional setting. For this case we stated the first order necessary condition and reduced them by several variables and equations. Analysing them for linear systems we showed that the least squares problem formulation without regularizing the initial data is well-posed not only with respect to the L_2-norm but also for the L_∞-norm. However, the error propagation with respect to L_2-errors may be large for low observability measures independent of the regularization parameter for the model errors. Considering L_∞-errors the behaviour is different. Then, in case of one state only, we have bounds independent of the stiffness of state equations.

As seen it is fundamental to discuss in which norms the data errors are bounded and what output is of interest. In my opinion one has not only the L_2-norm of the data error bounded, but one can assume $\|z^\delta\|_{L_2(t_0,t_0+H)} \leq \delta\sqrt{H}$ and $\|z^\delta\|_\infty \leq c$. Hence, one has additional information about the error which should be taken into account, and the error would depend on the length of the horizon. The question concerning the outputs depends on the application of state estimation. Which output is of interest should be stated together with the problem formulation. For example, employing state estimation to obtain the current state required for the main issue of controling a process should have the focus on $x(t_0 + H)$, the filtered state. Then the L_2-norm of the state on $[t_0, t_0 + H]$ is less adequate than measuring the error of $x(t_0 + H)$. If one is interested on the state over the whole horizon, also called smoothed state, one may consider a weighted L_2-norm putting more weight on the current state than on the past state. Or, is the L_∞- norm over the whole horizon more adequate than the L_2-norm? The answers of these question do not only influence the theoretical analysis but should also affect the numerical studies. In particular, as soon as adaptivity concerning the underlying discretization grid is introduced it is of greatest importance to know which error shall be finally small to obtain greatest efficency.

References

[1] Allgöwer, F. and Badgwell, T.A. and Qin, J.S. and Rawlings, J.B. and Wright, S.J., “Nonlinear Predictive Control and Moving Horizon Estimation - An Introductory Overview”, *Advances in Control: Highlights of ECC 1999*, ed. Frank, P.M., Springer-Verlag, 391-449, (1999).

[2] Alt, H.W., “Lineare Funktionalanalysis”, *Springer*, Berlin, Heidelberg, (1999).

[3] Binder, T. and Blank, L. and Dahmen, W. and Marquardt, W., “On the Regularization of Dynamic Data Reconciliation Problems”, *Journal of Process Control*, **12**, 557-567, (2002).

[4] Blank, L., “State Estimation without Regularizing the Initial Data”, *Inverse Problems*, **20**,5, 1357-1370, (2004).

[5] Box, G.E.P. and Tiao, G.C., “Bayesian Inference in Statistical Analysis”, *Addison-Wesley*, Reading, (1973).

[6] Corless, M.J. and Frazho, A.E., “Linear Systems and Control, An Operator Perspective”, *Marcel Dekker, Inc.*, New York, Basel, (2003).

[7] Deuflhard, P. and Hohmann, A., “Numerische Mathematik I. Eine algorithmisch orientierte Einführung”, *Walter de Gruyter*, Berlin, NewYork, (2002).

[8] Engl, H.W. and Hanke, M. and Neubauer, A., “Regularization of Inverse Problems”, *Kluwer*, Dordrecht, The Netherlands, (1996).

[9] Haseltine, E.L. and Rawlings, J.B., “Critical evaluation of extended Kalman filtering and moving horizon estimation”, *Ind. Eng. Chem. Res.*,**44**,8, 2451-2460, (2005).

[10] Jazwinski, A.H., “Stochastic Processes and Filtering Theory”, *Academic Press*, New York, (1970).

[11] Kailath, T., “Linear Systems”, *Prentice Hall*, Englewood Cliffs, New Jersey, (1980).

[12] Muske, K.R. and Edgar, T.F., “Nonlinear state estimation”, *Nonlinear Process Control*, eds. Henson, M.A. and Seborg, D.E., 311-711, (1997).

[13] Waldraff, W. and Dochain, D. and Bourrel, S. and Magnus, A., ”On the Use of Observability Measures for Sensor Location in Tubular Reactors”, *Journal of Process Control*, **8**, 497-505, (1998).

Minimum-Distance Receding-Horizon State Estimation for Switching Discrete-Time Linear Systems

Angelo Alessandri[1], Marco Baglietto[2], and Giorgio Battistelli[3]

[1] Department of Production Engineering, Thermoenergetics, and Mathematical Models, DIPTEM–University of Genoa, P.le Kennedy Pad. D, 16129 Genova, Italy
alessandri@diptem.unige.it

[2] Department of Communications, Computer and System Sciences, DIST–University of Genoa, Via Opera Pia 13, 16145 Genova, Italy
mbaglietto@dist.unige.it

[3] Dipartimento di Sistemi e Informatica, DSI-Universit'a di Firenze, Via S. Marta 3, 50139, Firenze Italy
battistelli@dsi.unifi.it

Summary. State estimation is addressed for a class of discrete-time systems that may switch among different modes taken from a finite set. The system and measurement equations of each mode are assumed to be linear and perfectly known, but the current mode of the system is unknown and is regarded as a discrete state to be estimated at each time instant together with the continuous state vector. A new computationally efficient method for the estimation of the system mode according to a minimum-distance criterion is proposed. The estimate of the continuous state is obtained according to a receding-horizon approach by minimizing a quadratic least-squares cost function. In the presence of bounded noises and under suitable observability conditions, an explicit exponentially converging sequence provides an upper bound on the estimation error. Simulation results confirm the effectiveness of the proposed approach.

1 Introduction

The literature on state estimation for systems that may undergo switching among various modes includes, among others, methods based on the use of banks of filters and hidden finite-state Markov chains [5]. In this contribution, a different approach is presented that is based on the idea of using only a limited amount of the most recent information and is usually referred to as *receding-horizon* or *moving-horizon*.

Recently, after the success of model predictive control [8], many researches on receding-horizon state estimation appeared [1, 3, 6, 9]. The first investigations on such techniques date back to the late sixties (see, e.g., [7]), when it was proposed to reduce the effects of the uncertainties by determining estimates that depend only on a batch of the most recent measurements.

In this contribution, we focus on a receding-horizon state estimator with a lower computation effort with respect to that required by the method described

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 347–358, 2007.
springerlink.com

in [2] for the same class of hybrid systems. More specifically, following the lines of [2], we use a generalized least-squares approach that consists in minimizing a quadratic estimation cost function defined on a moving window including the most recent measurements. To account for the occurrences of switches, the proposed approach relies on the estimation of both the discrete state and the continuous one. At every time step, first an estimate of the discrete state is determined on the basis of the observations vector over the most recent time instants, then such an estimate is used to determine the quadratic loss function that has to be minimized in order to estimate the continuous state. The main novelties with respect to the approach presented in [2] concern: i) the development of a new method for the estimation of the system mode, based on a minimum-distance criterion, which turns out to be more computationally efficient; ii) the choice of a simpler quadratic loss function that allows one to derive a closed-form expression for the optimal estimate of the continuous state variables, thus avoiding the necessity of resorting to heavy on-line computations. Furthermore, it is important to remark that the proposed estimation scheme can be always applied regardless of the form of the sets to which the system and measurement noises belong.

Likewise in [2], an explicit relationship is established between the observability and the convergence properties of the estimation error. More specifically, in the presence of bounded noises and under suitable observability conditions, an exponentially converging sequence can be obtained that provides an upper bound on the estimation error for the continuous state vector (even when the system dynamics is unstable). Simulation results show the effectiveness of the proposed approach in comparison with the one proposed in [2]. As expected, the new estimation method leads to a great reduction in computation time at the price of a small decay in performance. The proofs are omitted for the sake of brevity (the interested reader can contact the authors).

Let us introduce some definitions that will be useful in the following. Given a vector v, $\|v\|$ denotes its Euclidean norm. For a generic time-varying vector v_t, let us define $v_{t-N}^t \triangleq \text{col}\,(v_{t-N}, v_{t-N+1}, \ldots, v_t)$. Given a matrix M, we denote by $\underline{\sigma}(M)$ and $\bar{\sigma}(M)$ its minimum and maximum singular values, respectively. Furthermore, $M^\top$ is the matrix transpose of M and $\|M\| = \bar{\sigma}(M)$ is its norm. Given n square matrices $M_1, M_2, \ldots, M_n$, $\prod_{i=1}^{n} M_i \triangleq M_1 M_2 \cdots M_n$ is the ordered product of such matrices.

2 A Minimum-Distance Criterion for the Estimation of the Discrete State

Let us consider a class of switching discrete-time linear systems described by

$$\begin{aligned} x_{t+1} &= A(\lambda_t)\, x_t + w_t \\ y_t \; &= C(\lambda_t)\, x_t + v_t \end{aligned} \tag{1}$$

where $t = 0, 1, \ldots$ is the time instant, $x_t \in \mathbb{R}^n$ is the continuous state vector (the initial continuous state x_0 is unknown), $\lambda_t \in \mathcal{L} \triangleq \{1, 2, \ldots, L\}$ is the system mode or discrete state, $w_t \in \mathcal{W} \subset \mathbb{R}^n$ is the system noise vector, $y_t \in \mathbb{R}^m$ is the vector of the measures, and $v_t \in \mathcal{V} \subset \mathbb{R}^m$ is the measurement noise vector. $A(\lambda)$ and $C(\lambda)$, $\lambda \in \mathcal{L}$, are $n \times n$ and $m \times n$ matrices, respectively. We assume the statistics of x_0, w_t, and v_t to be unknown as well as the law governing the evolution of the discrete state.

In this section, a minimum-distance criterion is proposed for the estimation of the discrete state of system (1). More specifically, given the noisy observations vector y_{t-N}^t over a given time interval $[t-N, t]$, such a criterion allows one to estimate the *switching pattern* $\pi_t \triangleq \lambda_{t-N}^t$ (or at least a portion of it [2]). Since system (1) is time-invariant with respect to the extended state (x_t, λ_t), in the following of this section, for the sake of simplicity and without loss of generality, we shall always consider the interval $[0, N]$.

If the evolution of the discrete state is completely unpredictable, the switching pattern $\pi_N \triangleq \lambda_0^N$ can assume any value in the set $\mathcal{L}^{N+1}$. However, in many practical cases, the a-priori knowledge of the system may allow one to consider a restricted set of "admissible" switching patterns [4]. Think, for example, of the case in which the discrete state is slowly varying, i.e., there exists a minimum number τ of steps between one switch and the following one. Of course, such a-priori knowledge may make the task of estimating the discrete state from the measures y_0^N considerably simpler. As a consequence, instead of considering all the possible switching patterns belonging to $\mathcal{L}^{N+1}$, we shall consider a restricted set $\mathcal{P}_N \subseteq \mathcal{L}^{N+1}$ of all the *admissible switching patterns*, i.e., of all the switching patterns consistent with the a-priori knowledge of the evolution of the discrete state.

Let us consider a generic switching pattern $\pi \triangleq \mathrm{col}\left(\lambda^{(0)}, \ldots, \lambda^{(N)}\right)$ and define the matrices $F(\pi)$ and $H(\pi)$ as

$$F(\pi) \triangleq \begin{bmatrix} C(\lambda^{(0)}) \\ C(\lambda^{(1)})A(\lambda^{(0)}) \\ \vdots \\ C(\lambda^{(N)}) \prod_{i=1}^{N} A(\lambda^{(N-i)}) \end{bmatrix},$$

$$H(\pi) \triangleq \begin{bmatrix} 0 & 0 & \cdots & 0 \\ C(\lambda^{(1)}) & 0 & \cdots & 0 \\ C(\lambda^{(2)})A(\lambda^{(1)}) & C(\lambda^{(2)}) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ C(\lambda^{(N)}) \prod_{i=1}^{N-1} A(\lambda^{(N-i)}) & C(\lambda^{(N)}) \prod_{i=1}^{N-2} A(\lambda^{(N-i)}) & \cdots & C(\lambda^{(N)}) \end{bmatrix}.$$

Then the observations vector y_0^N can be written as

$$y_0^N = F(\pi_N)\, x_0 + H(\pi_N)\, w_0^{N-1} + v_0^N\,. \tag{2}$$

For the sake of clarity, let us first recall some results on the observability of the discrete state in the absence of noises. Towards this end, let us consider the noise-free system

$$\begin{aligned} x_{t+1} &= A(\lambda_t)\, x_t \\ y_t &= C(\lambda_t)\, x_t\,. \end{aligned} \tag{3}$$

In this case, since the observations vector can be expressed as $y_0^N = F(\pi_N)\, x_0$, the set $\bar{\mathcal{S}}(\pi)$ of all the possible vectors of observations in the interval $[0, N]$ associated with a switching pattern π corresponds to the linear subspace

$$\bar{\mathcal{S}}(\pi) \overset{\triangle}{=} \left\{ \bar{y} \in \mathbb{R}^{m(N+1)} : \bar{y} = F(\pi)x\,,\ x \in \mathbb{R}^n \right\}\,,\ \pi \in \mathcal{P}_N\,.$$

The following notion of *distinguishability* between two switching patterns in the noise-free case can be introduced.

Definition 1. *For system* (3)*, two switching patterns* $\pi, \pi' \in \mathcal{P}_N$ *with* $\pi \neq \pi'$ *are said to be* distinguishable *if* $F(\pi)\, x \neq F(\pi')\, x'$ *for all* $x, x' \in \mathbb{R}^n$ *with* $x \neq 0$ *or* $x' \neq 0$.

As shown in [10], the *joint observability matrix* $\left[\, F(\pi)\ F(\pi')\, \right]$ plays a key role in determining the distinguishability of two switching patterns π and π'. More specifically, the following lemma holds.

Lemma 1. *Let us consider two generic switching patterns* $\pi \neq \pi' \in \mathcal{P}_N$. *Then* π *is distinguishable from* π' *if and only if* π *and* π' *are* jointly observable*, i.e.,* $\mathrm{rank}\left(\left[\, F(\pi)\ F(\pi')\, \right] \right) = 2n$.

In the light of Lemma 1, if the joint-observability condition were satisfied for every couple of switching patterns $\pi \neq \pi' \in \mathcal{P}_N$, then it would be possible to uniquely determine the switching pattern π_N on the basis of the observations vector y_0^N, provided that the initial continuous state x_0 is not null. Unfortunately, as shown in [2], unless the number of measures available at each time step is at least equal to the number of continuous state variables (i.e., $m \geq n$), in general it is not possible to satisfy the joint observability condition for all $\pi \neq \pi' \in \mathcal{P}_N$ (this happens because it is not possible to detect switches that occur in the last or in the first instants of an observations window). As a consequence, even in the absence of noises, in general it is not possible to uniquely determine the whole switching pattern π_N.

In order to overcome such a drawback, following the lines of [2], we shall look for two integers, α and ω, with $\alpha, \omega \geq 0$ and $\alpha + \omega \leq N$, such that it is possible to uniquely determine the discrete state λ_t in the restricted interval $[\alpha, N - \omega]$ on the basis of the observations vector y_0^N. Towards this end, given a switching pattern π in the interval $[0, N]$, let us denote as $r^{\alpha,\omega}(\pi)$ the restriction of π to the interval $[\alpha, N - \omega]$. Thus, the following notion of mode observability in the restricted interval $[\alpha, N - \omega]$ can be introduced.

Definition 2. *System* (3) *is said to be* (α,ω)-mode observable *in* $N+1$ *steps if, for every couple* $\pi, \pi' \in \mathcal{P}_N$ *such that* $r^{\alpha,\omega}(\pi) \neq r^{\alpha,\omega}(\pi')$ *,* π *is distinguishable from* π' *(or, equivalently,* π *and* π' *are jointly observable).*

According to Definition 2, if system (3) is (α,ω)-mode observable, then different switching patterns in the interval $[\alpha, N-\omega]$ generate different observations vector in the interval $[0,N]$, provided that the initial continuous state is not null. As a consequence, the switching pattern $r^{\alpha,\omega}(\pi_N)$ can be determined uniquely from the observations vector y_0^N. In fact, there could be more than one switching pattern π such that $y_0^N \in \bar{\mathcal{S}}(\pi)$; however, they all correspond to the same switching pattern in the restricted interval $[\alpha, N-\omega]$.

With these observability results in mind, let us now focus on the noisy system (1). Clearly, if the noise vectors are not identically null, in general the noisy observations vector y_0^N does not belong to the linear subspace $\bar{\mathcal{S}}(\pi_N)$. However, if the noise vectors are "small," it is reasonable to think that y_0^N is "close" (in some sense) to such a set. This simple intuition leads us to adopt a *minimum-distance criterion* for the estimation of the switching pattern. Towards this end, given a generic switching pattern $\pi \in \mathcal{P}_N$, let us denote as $d(y_0^N, \pi)$ the distance between the observations vector y_0^N and the linear subspace $\bar{\mathcal{S}}(\pi)$. Clearly, $d(y_0^N, \pi)$ can be obtained as

$$d(y_0^N, \pi) = \left\| \, [I - P(\pi)] \, y_0^N \, \right\|$$

where $P(\pi)$ is the matrix of the orthogonal projection on $\bar{\mathcal{S}}(\pi)$, Then we shall consider as an estimate of π_N the switching pattern $\hat{\pi}_N$ such that

$$\hat{\pi}_N = \arg \min_{\pi \in \mathcal{P}_N} d(y_0^N, \pi) \,. \tag{4}$$

It is important to note that such a criterion can be always applied regardless of the form of the sets $\mathcal{W}$ and $\mathcal{V}$ to which the system and measurement noises belong. Moreover, an exact knowledge of the form of such sets is not required.

Of course, it would be interesting to know whether, under suitable assumptions, the estimate $\hat{\pi}_N$ coincides with the true switching pattern π_N at least in the restricted interval $[\alpha, N-\omega]$. With this respect, by defining the quantities

$$\delta_{\max}(\pi, \pi') \overset{\triangle}{=} \sup_{\bar{w} \in \mathcal{W}^N \, ; \, \bar{v} \in \mathcal{V}^{N+1}} \left\| \, [I - P(\pi)] \, [H(\pi')\bar{w} + \bar{v}] \, \right\| , \quad \pi, \pi' \in \mathcal{P}_N \,,$$

the following lemma can be stated.

Lemma 2. *Suppose that the sets* $\mathcal{W}$ *and* $\mathcal{V}$ *are bounded and consider a switching pattern* $\pi \in \mathcal{P}_N$ *(with* $\pi \neq \pi_N$ *) such that* π *and* π_N *are jointly observable. If the initial continuous state* x_0 *satisfies the condition*

$$\|x_0\| > \frac{\delta_{\max}(\pi_N, \pi_N) + \delta_{\max}(\pi, \pi_N)}{\underline{\sigma}\,\{[I - P(\pi)]F(\pi_N)\}} \,, \tag{5}$$

then we have

$$d(y_0^N, \pi) > d(y_0^N, \pi_N) \,.$$

In the light of Lemma 2, provided that the initial continuous state x_0 is "far enough" from the origin, if one applies the minimum-distance criterion (4), the actual switching pattern π_N cannot be confused with another switching pattern π that is distinguishable from π_N according to Definition 1. Note that the boundedness of the sets $\mathcal{W}$ and $\mathcal{V}$ ensures the finiteness of all the scalars $\delta_{\max}(\pi,\pi')$. Furthermore, the satisfaction of the joint observability condition ensures that the minimum singular value $\underline{\sigma}\{[I-P(\pi)]F(\pi_N)\}$ is strictly greater than 0. It is important to note that such a value represents a measure of the separation between the linear subspaces $\bar{\mathcal{S}}(\pi)$ and $\bar{\mathcal{S}}(\pi_N)$, i.e., the greater is the angle between such subspaces the greater is the value of $\underline{\sigma}\{[I-P(\pi)]F(\pi_N)\}$.

Recalling the notion of observability given in Definition 2, Lemma 2 leads in a straightforward way to the following theorem.

Theorem 1. *Suppose that the sets $\mathcal{W}$ and $\mathcal{V}$ are bounded and that the noise-free system* (3) *is (α,ω)-mode observable in $N+1$ steps. If the initial continuous state x_0 satisfies the condition*

$$\|x_0\| > \rho_x \overset{\triangle}{=} \max_{\substack{\pi,\pi' \in \mathcal{P}_N \\ r^{\alpha,\omega}(\pi) \neq r^{\alpha,\omega}(\pi')}} \frac{\delta_{\max}(\pi,\pi)+\delta_{\max}(\pi',\pi)}{\underline{\sigma}\{[I-P(\pi')]F(\pi)\}},$$

then $r^{\alpha,\omega}(\hat{\pi}_N) = r^{\alpha,\omega}(\pi_N)$.

Thus, provided that the initial continuous state x_0 is "far enough" from the origin, the minimum-distance criterion (4) leads to the exact identification of the discrete state in the interval $[\alpha, N-\omega]$.

3 A Receding-Horizon State Estimation Scheme

In this section, the previous results are applied to the development of a receding-horizon scheme for the estimation of both the discrete and the continuous state.

In Section 2, it has been shown that under suitable assumptions, given the observations vector y_{t-N}^{t}, it is possible to obtain a "reliable" estimate of the discrete state in the restricted interval $[t-N+\alpha, t-\omega]$. As a consequence, at any time instant $t=N,N+1,\ldots$, the following estimation scheme can be adopted: i) estimate the switching pattern in the restricted interval $[t-N+\alpha, t-\omega]$ on the basis of the observations vector in the extended interval $[t-N,t]$; ii) estimate the continuous state in the restricted interval $[t-N+\alpha, t-\omega]$ by minimizing a certain quadratic cost involving the estimated discrete state.

Let us first consider step i). Towards this end, let us denote as $\gamma_t \overset{\triangle}{=} r^{\alpha,\omega}(\pi_t)$ the switching pattern in the restricted interval $[t-N+\alpha, t-\omega]$. Furthermore, let us denote by $\hat{\lambda}_{t-N,t},\ldots,\hat{\lambda}_{t,t}$, $\hat{\pi}_{t,t}$, and $\hat{\gamma}_{t,t}$ the estimates (made at time t) of $\lambda_{t-N},\ldots,\lambda_t$, π_t, and γ_t, respectively. In order to take into account the possibility of a time-varying a-priori knowledge on the discrete state, let us consider the set $\mathcal{P}_t$ of all the admissible switching patterns at time t, i.e., the set of all the switching patterns in the observations window $[t-N,t]$ consistent with the a-priori knowledge of the evolution of the discrete state. Furthermore, let us denote by $\mathcal{G}_t$ the set of admissible switching patterns in the restricted interval $[t-N+\alpha, t-\omega]$. For the sake of simplicity, let us suppose that such a-priori knowledge does not diminish with time, i.e., $\mathcal{P}_{t+1} \subseteq \mathcal{P}_t$ for $t = N, N+1, \ldots$, or, less restrictively, $\mathcal{P}_t \subseteq \mathcal{P}_N$. In accordance with the minimum-distance criterion proposed in Section 2, at every time instant $t = N, N+1, \ldots$, we shall address the minimization of the loss function

$$d(y_{t-N}^t, \hat{\pi}_{t,t}) = \left\| \left[I - P(\hat{\pi}_{t,t})\right] y_{t-N}^t \right\| . \tag{6}$$

Let us now consider step ii). At any time $t = N, N+1, \ldots$, the objective is to find estimates of the continuous state vectors $x_{t-N+\alpha}, \ldots, x_{t-\omega}$ on the basis of the measures collected in an observations window $[t-N+\alpha, t-\omega]$, of a "prediction" $\bar{x}_{t-N+\alpha}$, and of the estimate $\hat{\gamma}_{t,t}$ of the switching pattern γ_t obtained in step i). Let us denote by $\hat{x}_{t-N+\alpha,t}, \ldots, \hat{x}_{t-\omega,t}$ the estimates (to be made at time t) of $x_{t-N+\alpha}, \ldots, x_{t-\omega}$, respectively. We assume that the prediction $\bar{x}_{t-N+\alpha}$ is determined via the noise-free state equation by the estimates $\hat{x}_{t-N+\alpha-1,t-1}$ and $\hat{\lambda}_{t-N+\alpha-1,t-1}$, that is,

$$\bar{x}_{t-N+\alpha} = A(\hat{\lambda}_{t-N+\alpha-1,t-1})\, \hat{x}_{t-N+\alpha-1,t-1}, \quad t = N+1, N+2, \ldots . \tag{7}$$

The vector $\bar{x}_\alpha$ denotes an a-priori prediction of x_α.

A notable simplification of the estimation scheme can be obtained by determining the estimates $\hat{x}_{t-N+\alpha+1,t}, \ldots, \hat{x}_{t-\omega,t}$ from the first estimate $\hat{x}_{t-N+\alpha,t}$ via the noise-free state equation, that is,

$$\hat{x}_{i+1,t} = A(\hat{\lambda}_{i,t})\, \hat{x}_{i,t}, \quad i = t-N+\alpha, \ldots, t-\omega-1. \tag{8}$$

By applying (8), it follows that, at time t, only the estimate $\hat{x}_{t-N+\alpha,t}$ has to be determined, whereas the estimates $\hat{x}_{t-N+\alpha+1,t}, \ldots, \hat{x}_{t-\omega,t}$ can be computed via (8).

As we have assumed the statistics of the disturbances and of the initial continuous state to be unknown, a natural criterion to derive the estimator consists in resorting to a least-squares approach. Towards this end, following the lines of [1, 3], at any time instant $t = N, N+1, \ldots$ we shall address the minimization of the following quadratic cost function:

$$\begin{aligned} J\left(\hat{x}_{t-N+\alpha,t}, \bar{x}_{t-N+\alpha}, y_{t-N+\alpha}^{t-\omega}, \hat{\gamma}_{t,t}\right) &= \mu \left\| \hat{x}_{t-N+\alpha,t} - \bar{x}_{t-N+\alpha} \right\|^2 \\ &+ \sum_{i=t-N+\alpha}^{t-\omega} \left\| y_i - C(\hat{\lambda}_{i,t})\, \hat{x}_{i,t} \right\|^2 \end{aligned} \tag{9}$$

where μ is a non-negative scalar by which we express our belief in the prediction $\bar{x}_{t-N+\alpha}$ as compared with the observation model. It is worth noting that μ could be replaced with suitable weight matrices, without involving additional conceptual difficulties in the reasoning reported later on. Note that, by applying (8), cost (9) can be written in the equivalent form

$$\begin{aligned} J\left(\hat{x}_{t-N+\alpha,t}, \bar{x}_{t-N+\alpha}, y_{t-N+\alpha}^{t-\omega}, \hat{\gamma}_{t,t}\right) &= \mu \left\| \hat{x}_{t-N+\alpha,t} - \bar{x}_{t-N+\alpha} \right\|^2 \\ &+ \left\| y_{t-N+\alpha}^{t-\omega} - F(\hat{\gamma}_{t,t})\, \hat{x}_{t-N+\alpha,t} \right\|^2 . \end{aligned} \tag{10}$$

Summing up, the following receding-horizon estimation procedure has to be applied at any time instant $t = N, N+1, \ldots$.

Procedure 1

1. *Given the observations vector* y_{t-N}^{t}, *compute the optimal estimate* $\hat{\pi}_{t,t}^{\circ}$ *that minimizes the distance measure* (6), *i.e.,*

$$\hat{\pi}_{t,t}^{\circ} = \arg \min_{\hat{\pi}_{t,t} \in \mathcal{P}_t} d(y_{t-N}^{t}, \hat{\pi}_{t,t}) \,.$$

2. *Set* $\hat{\gamma}_{t,t}^{\circ} = r^{\alpha,\omega}(\hat{\pi}_{t,t}^{\circ})$.
3. *Given the optimal estimate* $\hat{\gamma}_{t,t}^{\circ}$, *the observations vector* $y_{t-N+\alpha}^{t-\omega}$, *and the prediction* $\bar{x}_{t-N+\alpha}$, *compute the optimal estimate*

$$\hat{x}_{t-N+\alpha,t}^{\circ} = \arg \min_{\hat{x}_{t-N+\alpha,t}} J\left(\hat{x}_{t-N+\alpha,t}\,,\, \bar{x}_{t-N+\alpha}\,,\, y_{t-N+\alpha}^{t-\omega}\,,\, \hat{\gamma}_{t,t}^{\circ}\right)$$

that minimizes cost (9) *under the constraints* (8).

4. *Given the optimal estimates* $\hat{x}_{t-N+\alpha,t}^{\circ}$ *and* $\hat{\lambda}_{t-N+\alpha,t}^{\circ}$, *compute the prediction* $\bar{x}_{t-N+\alpha+1}$ *as*

$$\bar{x}_{t-N+\alpha+1} = A(\hat{\lambda}_{t-N+\alpha,t}^{\circ})\, \hat{x}_{t-N+\alpha,t}^{\circ} \,.$$

The procedure is initialized at time $t = N$ with an a-priori prediction $\bar{x}_{\alpha}$.

It is important to note that the form of the set $\mathcal{P}_t$ plays a central role in the possibility of computing the minimum in step 1 in a reasonable time. In fact, if the cardinality of the set $\mathcal{P}_t$ grows very rapidly with the size N of the observations window or with the number L of possible discrete states, such a computation may become too time-demanding (this happens, for example, when the system can switch arbitrarily at every time step). Such issues can be avoided if the a-priori knowledge on the evolution of the discrete state leads to a considerable reduction of the number of admissible switching patterns. This is the case, for example, when the size $N+1$ of the observations window is smaller than the minimum admissible number of steps between one switch and the following one. In fact, under such an assumption, the cardinality of the set $\mathcal{P}_t$ is $L[(L-1)N+1]$ (see [2]).

As to step 3, since cost (9) depends quadratically on the estimate $\hat{x}_{t-N+\alpha,t}$, by applying the first order optimality condition a closed-form expression can be derived for the optimal estimate $\hat{x}_{t-N+\alpha,t}^{\circ}$. More specifically, along the lines of [1], where non-switching linear systems were considered, the following proposition can be easily proved.

Proposition 1. *Suppose that* $\mu > 0$ *or that* $\operatorname{rank}\left\{F(\hat{\gamma}_{t,t}^{\circ})\right\} = n$. *Then cost* (9) *has a unique minimum point given by*

$$\hat{x}_{t-N+\alpha,t}^{\circ} = \left[\mu I + F(\hat{\gamma}_{t,t}^{\circ})^{\top} F(\hat{\gamma}_{t,t}^{\circ})\right]^{-1} \left[\mu\, \bar{x}_{t-N+\alpha} + F(\hat{\gamma}_{t,t}^{\circ})^{\top} y_{t-N+\alpha}^{t-\omega}\right] .$$

Let us now define the following quantities

$$f_{\min} \triangleq \min_{\gamma \in \mathcal{G}_N} \underline{\sigma}\left[F(\gamma)\right], \quad f \triangleq \max_{\gamma \in \mathcal{G}_N} \|F(\gamma)\|, \quad h \triangleq \max_{\gamma \in \mathcal{G}_N} \|H(\gamma)\|,$$
$$\bar{f} \triangleq \max_{\gamma,\gamma' \in \mathcal{G}_N} \|F(\gamma) - F(\gamma')\|, \quad \rho_w \triangleq \sup_{w \in \mathcal{W}} \|w\|, \quad \rho_v \triangleq \sup_{v \in \mathcal{V}} \|v\|,$$
$$a \triangleq \max_{\lambda \in \mathcal{L}} \|A(\lambda)\|, \quad \bar{a} \triangleq \max_{\lambda,\lambda' \in \mathcal{L}} \|A(\lambda) - A(\lambda')\|.$$

Note that, for the sake of compactness, the dependence of $f_{\min}$, h, f, and $\bar{f}$ on the size N of the observations window and on the scalars α and ω has been omitted.

In order to show the convergence properties of the proposed estimator, the following assumptions are needed.

A1. $\mathcal{W}$ and $\mathcal{V}$ are bounded sets.
A2. System (3) is (α,ω)-mode observable in $N+1$ steps.
A3. For any $\gamma \in \mathcal{G}_N$, we have $\operatorname{rank}\{F(\gamma)\} = n$.

Clearly, Assumption A1 ensures that $\rho_w < +\infty$ and $\rho_v < +\infty$. As to Assumption A3, it ensures that the considered system is observable in the restricted window $[t-N+\alpha, t-\omega]$ with respect to the continuous state for any switching pattern γ_t and hence that $f_{\min} > 0$.

We are now ready to state the following theorem.

Theorem 2. *Suppose that Assumptions A1, A2, and A3 are satisfied. Then the norm of the estimation error* $e_{t-N+\alpha} \triangleq x_{t-N+\alpha} - \hat{x}^{\circ}_{t-N+\alpha,t}$ *is bounded above as*

$$\|e_{t-N+\alpha}\| \le \zeta_{t-N+\alpha} \quad , \quad t = N, N+1, \ldots \,.$$

The sequence $\{\zeta_t\}$ *is defined recursively as*

$$\begin{aligned} \zeta_\alpha &= d_\alpha\,, \\ \zeta_t &= c\,\zeta_{t-1} + d\,, \quad t = \alpha+1, \alpha+2, \ldots \,. \end{aligned} \tag{11}$$

where

$$c = \frac{\mu\, a}{\mu + f_{\min}^2}\,,$$
$$d = \frac{1}{\mu + f_{\min}^2}\left\{ \left(\mu\,\bar{a} + f\,\bar{f}\right) \left(a^\alpha\,\rho_x + \frac{a^\alpha - 1}{a-1}\,\rho_w\right) + \mu\,\rho_w \right.$$
$$\left. + f\,\left(h\,\sqrt{N-\alpha-\omega}\;\rho_w + \sqrt{N-\alpha-\omega+1}\;\rho_v\right) \right\},$$
$$d_\alpha = \frac{1}{\mu + f_{\min}^2}\left\{ \mu\,\|x_\alpha - \bar{x}_\alpha\| + f\,\bar{f}\,\left(a^\alpha\,\rho_x + \frac{a^\alpha - 1}{a-1}\,\rho_w\right)\right.$$
$$\left. + f\,\left(h\,\sqrt{N-\alpha-\omega}\;\rho_w + \sqrt{N-\alpha-\omega+1}\;\rho_v\right] \right\}.$$

Moreover, if the scalar weight μ *has been selected such that* $c < 1$*, then the sequence* $\{\zeta_t\}$ *converges exponentially to the asymptotic value* $e_\infty(\mu) \overset{\triangle}{=} d/(1-c)$*.*

Note that, since $f_{\min} > 0$, condition $c < 1$ can be easily verified for any value of a through a suitable choice of μ.

4 Simulation Results

Let us consider the switching system, also investigated in [2], described by means of equations (1) with

$$A(1) = \begin{bmatrix} 0 & 1.2 \\ -0.6 & 0 \end{bmatrix},\ A(2) = \begin{bmatrix} 0 & 0.5 \\ -1.5 & 0 \end{bmatrix},\ C(1) = \begin{bmatrix} 1 & 1 \end{bmatrix},\ C(2) = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

We assumed that such system has a minimum dwell time (i.e., the minimum number τ of steps between one switch and the next) equal to 7. Moreover, we assumed w_t and v_t to belong to the polytopic compact sets $\mathcal{W} = [-0.01, 0.01]^2$ and $\mathcal{V} = [-0.1, 0.1]$, respectively. Note that with such choices the trajectories of the continuous state turn out to be bounded. It is immediate to verify that, in this case, by choosing $\alpha = 1$, $\omega = 2$, and $N = 6$, Assumptions A2 and A3 are satisfied, hence one can use the receding-horizon estimation scheme proposed in Section 3.

In the following, for the sake of brevity, we shall refer to the estimator obtained by iteratively applying Procedure 1 as the *Minimum-Distance Receding-Horizon Filter* (MDRHF). In order to evaluate the ability of the proposed estimation scheme to deal with unknown switches in the discrete state, we compared the proposed filter with the receding-horizon filter obtained by an exact knowledge of the discrete state, i.e., by minimizing cost (10) with $\hat{\gamma}_{t,t} = \gamma_t$. Such an estimator will be called the *Receding-Horizon Filter with Perfect Information* (RHFPI). Furthermore, the proposed filter will be also compared with the *Constrained Receding-Horizon Filter* (CRHF) of [2]. For the sake of comparison, we supposed x_0 to be a Gaussian-distributed zero-mean random variable with covariance $\mathrm{diag}(1, 1)$, and w_t and v_t, $t = 0, 1, \ldots$, to be independent random variables uniformly distributed in the sets $\mathcal{W}$ and $\mathcal{V}$, respectively. As to the cost, we chose $\mu = 10$.

Fig. 1 shows the plots of the *Root Mean Square Error* (RMSEs), computed over 10^4 randomly chosen simulations, for the considered filters. As expected, the best asymptotic behavior is provided by the RHFPI, thanks to its exact knowledge of the discrete state. The MDRHF shows a small decay in performance if compared with the CRHF. However, each iteration of the proposed estimation scheme required on average just 0.3 ms, while the mean time required by an iteration of the CRHF turned out to be 37.0 ms (simulations were performed on an AMD Athlon XP 1700+ PC and each minimization problem for the CRHF was solved on line by means of the standard Matlab routine for quadratic programming). As a consequence, one may conclude that in this case the MDRHF provides

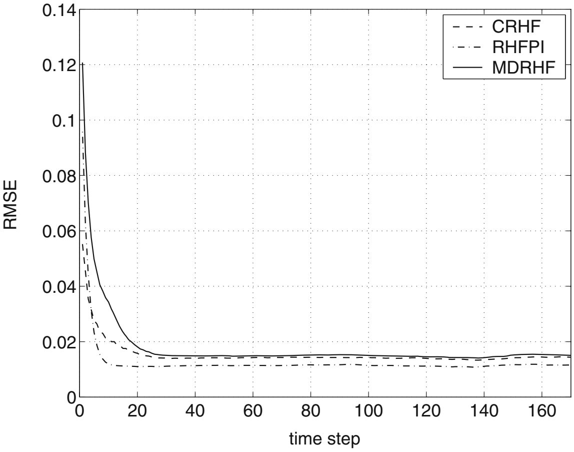


Fig. 1. Plots of the RMSEs for the considered filters

a reasonable tradeoff between the performance and the computational burden. It is important to point out that the simulation framework considered in this example represents a best-case scenario for the CRHF, as the sets $\mathcal{W}$ and $\mathcal{V}$ are known and polytopic, thus ensuring the applicability of such an approach, as well as its numerical tractability. On the contrary, the MDRHF could be applied regardless of the form of the sets $\mathcal{W}$ and $\mathcal{V}$ and no exact knowledge of such sets is required.

References

[1] Alessandri, A., Baglietto, M., and Battistelli, G., "Receding-horizon estimation for discrete-time linear systems," *IEEE Trans. on Automatic Control*, **48**, pp. 473–478, (2003).

[2] Alessandri, A., Baglietto, M., and Battistelli, G., "Receding-horizon estimation for switching discrete-time linear systems", *IEEE Trans. on Automatic Control*, **50**, pp. 1736–1748, (2005).

[3] Alessandri, A., Baglietto, Parisini, T., and Zoppoli, R., "A neural state estimator with bounded errors for nonlinear systems," *IEEE Trans. on Automatic Control*, **44**, pp. 2028–2042, (1999).

[4] Balluchi, A., Benvenuti, L., Di Benedetto, M.D., and Sangiovanni- Vincentelli, A., "Design of observers for hybrid systems," *Hybrid Systems: Computation and Control*, ser. Lecture Notes in Computer Science, C.J. Tomlin and M.R. Greenstreet, Eds., Springer, pp. 76–89, (2002).

[5] Bar-Shalom, Y. and Li, X., *Estimation and Tracking*, Artech House, Boston-London, 1993.

[6] Ferrari-Trecate, G., Mignone, D., and Morari, M., "Moving horizon estimation for hybrid systems," *IEEE Trans. on Automatic Control*, **47**, pp. 1663–1676, (2002).

[7] Jazwinski, A.H., "Limited memory optimal filtering," *IEEE Trans. on Automatic Control*, **13**, pp. 558–563, (1968).

[8] Mayne D.Q., Rawlings, J.B., Rao, C.V., and Scokaert, P.O.M., "Constrained model predictive control: stability and optimality," *Automatica*, **36**, pp. 789–814, (2000).

[9] Rao, C.V., Rawlings, J.B., and Mayne, D.Q., "Constrained state estimation for nonlinear discrete-time systems: stability and moving horizon approximations," *IEEE Trans. on Automatic Control*, **48**, pp. 246–257, (2003).

[10] Vidal, R., Chiuso, A., and Soatto, S.,"Observability and identifiability of jump linear systems," *Proc. of the 41st IEEE Conference on Decision and Control*, Las Vegas, Nevada, pp. 3614–3619, (2002).

New Extended Kalman Filter Algorithms for Stochastic Differential Algebraic Equations

John Bagterp Jørgensen[1], Morten Rode Kristensen[2], Per Grove Thomsen[1], and Henrik Madsen[1]

[1] Informatics and Mathematical Modelling, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
{jbj,pgt,hm}@imm.dtu.dk

[2] Department of Chemical Engineering, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
mrk@kt.dtu.dk

Summary. We introduce stochastic differential algebraic equations for physical modelling of equilibrium based process systems and present a continuous-discrete paradigm for filtering and prediction in such systems. This paradigm is ideally suited for state estimation in nonlinear predictive control as it allows systematic decomposition of the model into predictable and non-predictable dynamics. Rigorous filtering and prediction of the continuous-discrete stochastic differential algebraic system requires solution of Kolmogorov's forward equation. For non-trivial models, this is mathematically intractable. Instead, a suboptimal approximation for the filtering and prediction problem is presented. This approximation is a modified extended Kalman filter for continuous-discrete systems. The modified extended Kalman filter for continuous-discrete differential algebraic systems is implemented numerically efficient by application of an ESDIRK algorithm for simultaneous integration of the mean-covariance pair in the extended Kalman filter [1, 2]. The proposed method requires approximately two orders of magnitude less floating point operations than implementations using standard software. Numerical robustness maintaining symmetry and positive semi-definiteness of the involved covariance matrices is assured by propagation of the matrix square root of these covariances rather than the covariance matrices themselves.

1 Introduction

The objective of state estimation in nonlinear model predictive control is to reconstruct the current state from past and current measurements. This state estimate is called the filtered state and is used as initial condition for prediction of the mean evolution in the dynamic optimization part of nonlinear model predictive control. While there is little or no difference in the way the predictions are accomplished, extended Kalman filtering (EKF) and moving horizon estimation (MHE) approaches have been suggested to compute the filtered state estimate for systems described by index-1 differential algebraic equations. However, mainly discrete-time stochastic systems or deterministic continuous-time systems with stochastics appended in an *ad hoc* manner have been applied to index-1 differential algebraic systems.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 359–366, 2007.
springerlink.com

In this contribution, we propose an extended Kalman filter (EKF) for stochastic continuous-time systems sampled at discrete time. The system evolution is described by stochastic index-1 differential algebraic equations and the output measurements at discrete times are static mappings contaminated by additive noise. In the implementation, the special structure of the resulting EKF equations are utilized such that the resulting algorithm is computationally efficient and numerically robust. By these features, the proposed EKF algorithm can be applied for state estimation in NMPC of large-scale systems as well as in algorithms for grey-box identification of stochastic differential-algebraic systems [3].

2 Extended Kalman Filters

The extended Kalman filter has been accepted as an *ad hoc* filter and predictor for nonlinear stochastic systems. It is *ad hoc* in the sense that it does not satisfy any optimality conditions but adopts the equations for the Kalman filter of linear systems, which is an optimal filter. In this section, we present the extended Kalman filter for a stochastic difference-algebraic system (discrete time) and show how this method is adapted to a stochastic differential-algebraic system (continuous-discrete time).

2.1 Discrete-Time System

Consider the stochastic difference-algebraic system

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{z}_k, \boldsymbol{w}_k) \tag{1a}$$
$$0 = g(\boldsymbol{x}_k, \boldsymbol{z}_k) \tag{1b}$$
$$\boldsymbol{y}_k = h(\boldsymbol{x}_k, \boldsymbol{z}_k) + \boldsymbol{v}_k \tag{1c}$$

in which $\boldsymbol{w}_k \sim N(0, Q_k)$, $\boldsymbol{v}_k \sim N(0, R_k)$ and $\boldsymbol{x}_0 \sim N(0, P_{0|-1})$. (1a) and (1b) represent the system dynamics while (1c) is a measurement equation. Assume further that $\frac{\partial g}{\partial z}$ is non-singular. Then according to the implicit function theorem, the algebraic variables, $\boldsymbol{z}_k$, are an implicit function of the state variables, $\boldsymbol{x}_k$, i.e. $\boldsymbol{z}_k = \chi(\boldsymbol{x}_k)$. Consequently, the stochastic difference-algebraic system (1) may be represented as a stochastic difference system with a measurement equation

$$\boldsymbol{x}_{k+1} = F(\boldsymbol{x}_k, \boldsymbol{w}_k) = f(\boldsymbol{x}_k, \chi(\boldsymbol{x}_k), \boldsymbol{w}_k) \tag{2a}$$
$$\boldsymbol{y}_k = H(\boldsymbol{x}_k) + \boldsymbol{v}_k = h(\boldsymbol{x}_k, \chi(\boldsymbol{x}_k)) + \boldsymbol{v}_k \tag{2b}$$

This is the stochastic difference system to which the discrete-time extended Kalman filter applies. The extended Kalman filter for (2) consists of the equations described in the following [4]. The filter part of the extended Kalman filter consists of the innovation-computation

$$e_k = y_k - \hat{y}_{k|k-1}, \tag{3a}$$

the feedback gain, $K_{fx,k}$, computation

$$C_k = \frac{\partial H}{\partial x}(\hat{x}_{k|k-1}) \tag{3b}$$

$$R_{k|k-1} = R_k + C_k P_{k|k-1} C_k' \tag{3c}$$

$$K_{fx,k} = P_{k|k-1} C_k' R_{k|k-1}^{-1} \tag{3d}$$

and the filtered mean estimate and covariance

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_{fx,k} e_k \tag{3e}$$

$$P_{k|k} = P_{k|k-1} - K_{fx,k} R_{k|k-1}^{-1} K_{fx,k}' \tag{3f}$$

As $\boldsymbol{w}_k \perp \boldsymbol{v}_k$ by assumption, $\hat{w}_{k|k} = 0$ and $Q_{k|k} = Q_k$. The one-step ahead prediction equations of the extended Kalman filter are the mean-covariance evolution equations

$$\hat{x}_{k+1|k} = F(\hat{x}_{k|k}, \hat{w}_{k|k}) \tag{4a}$$

$$P_{k+1|k} = A_k P_{k|k} A_k' + B_k Q_{k|k} B_k' \tag{4b}$$

in which $A_k = \frac{\partial F}{\partial x}(\hat{x}_{k|k}, \hat{w}_{k|k})$ and $B_k = \frac{\partial F}{\partial w}(\hat{x}_{k|k}, \hat{w}_{k|k})$. The one-step ahead prediction of the measurements is

$$\hat{y}_{k+1|k} = H(\hat{x}_{k+1|k}) \tag{4c}$$

2.2 Continuous-Discrete Time SDE System

Most physical systems are modelled in continuous-time using conservation equation. For deterministic systems this gives rise to a system of ordinary differential equations, while it for stochastic systems gives rise to a system of stochastic differential equations [5, 6].When measurements at the discrete-times $\{t_k : k = 0, 1, \ldots\}$ are added to that system we have a continuous-discrete time stochastic system

$$d\boldsymbol{x}(t) = F(\boldsymbol{x}(t))dt + \sigma(t) d\boldsymbol{\omega}(t) \tag{5a}$$

$$\boldsymbol{y}(t_k) = H(\boldsymbol{x}(t_k)) + \boldsymbol{v}(t_k) \tag{5b}$$

In this notation $\{\boldsymbol{\omega}(t)\}$ is a standard Wiener process, i.e. a Wiener process with incremental covariance Idt, the additive measurement noise is distributed as $\boldsymbol{v}(t_k) \sim N(0, R_k)$, and the initial states are distributed as $\boldsymbol{x}(t_0) \sim N(\hat{x}_{0|-1}, P_{0|-1})$.

The filter equations in the extended Kalman filter for the continuous-discrete time system (5) are equivalent to the filter equations for the discrete-time system (2), i.e. (3) constitutes the filter equations. The mean and covariance of the one-step ahead prediction, $\hat{x}_{k+1|k} = \hat{x}_k(t_{k+1})$ and $P_{k+1|k} = P_k(t_{k+1})$, are obtained by solution of the mean-covariance system of differential equations

$$\frac{d\hat{x}_k(t)}{dt} = F(\hat{x}_k(t)) \tag{6a}$$

$$\frac{dP_k(t)}{dt} = \left(\frac{\partial F}{\partial x}(\hat{x}_k(t))\right) P_k(t) + P_k(t) \left(\frac{\partial F}{\partial x}(\hat{x}_k(t))\right)' + \sigma(t)\sigma(t)' \tag{6b}$$

with initial conditions $\hat{x}_k(t_k) = \hat{x}_{k|k}$ and $P_k(t_k) = P_{k|k}$. The one-step ahead prediction of the measurements is

$$\hat{y}_{k+1|k} = \hat{y}_k(t_{k+1}) = H(\hat{x}_k(t_{k+1})) = H(\hat{x}_{k+1|k}) \tag{7}$$

The mean-covariance pair may be solved by standard ODE solvers in which the lower triangular part of the covariance matrix differential equation (6b) is appended to the mean differential equation (6a). For stiff systems in which implicit ODE solvers are needed, assuming that a dense linear algebra solver is used, the computational costs of solving this system has complexity $O(m^3)$ in which $m = n + n(n+1)/2$ and n is the state dimension. For large scale systems this corresponds to computational complexity $O(n^6)$. Even if sparse solvers for the linear algebra are applied, the mean-covariance pair cannot be solved in reasonable time by standard ODE solvers [2]. As a consequence, the extended Kalman filter for continuous-discrete time systems is not applicable to large-scale systems using a standard implicit ODE solver for solution of (6). It can easily be demonstrated [2] that solution of (6) is equivalent to solution of the system of differential equations

$$\frac{d\hat{x}_k(t)}{dt} = F(\hat{x}_k(t)) \qquad \hat{x}_k(t_k) = \hat{x}_{k|k} \tag{8a}$$

$$\frac{d\Phi(t,s)}{dt} = \left(\frac{\partial F}{\partial x}(\hat{x}_k(t))\right)\Phi(t,s) \qquad \Phi(s,s) = I \tag{8b}$$

along with the integral equation

$$P_k(t) = \Phi(t,t_k)P_{k|k}\Phi(t,t_k)' + \int_{t_k}^{t} \Phi(t,s)\sigma(s)\sigma(s)'\Phi(t,s)'ds \tag{8c}$$

The advantage of the formulation (8) is that very efficient solvers exist [1, 7, 8]for integration of the states (8a) along with the state sensitivities (8b). Subsequent computation of the covariance (8c) by quadrature is relatively cheap computation. Extended Kalman filters for continuous-discrete time systems based on solution of (8) rather than (6) are more than two orders of magnitude faster for a system with 50 states [2].

2.3 Continuous-Discrete Time SDAE System

Many systems in the process industries are modelled by systems of index-1 differential algebraic equations rather than systems of ordinary differential equations. Index-1 differential algebraic equations arise when some physical phenomena are described as equilibrium processes. This is the case for phase-equilibrium models of separation processes, e.g. distillation columns. The extension of filtering and prediction in continuous-discrete time stochastic differential equation (SDE) systems (5) to filtering and prediction in continuous-discrete time stochastic index-1 differential algebraic (SDAE) systems is the main contribution of this paper. A continuous-discrete time SDAE system can be written as

$$d\boldsymbol{x}(t) = f(\boldsymbol{x}(t), \boldsymbol{z}(t))dt + \sigma(t)d\boldsymbol{\omega}(t) \tag{9a}$$

$$0 = g(\boldsymbol{x}(t), \boldsymbol{z}(t)) \tag{9b}$$

$$\boldsymbol{y}(t_k) = h(\boldsymbol{x}(t_k), \boldsymbol{z}(t_k)) + \boldsymbol{v}(t_k) \tag{9c}$$

in which the evolution of the systems is modelled by a stochastic differential equation (9a) and an algebraic equation (9b). The system is observed at discrete times $\{t_k : k = 0, 1, \ldots\}$ through the measurement equation (9c) which is corrupted by additive measurement noise, $\boldsymbol{v}(t_k) \sim N(0, R_k)$. The initial state is distributed as $\boldsymbol{x}(t_0) \sim N(\hat{x}_{0|-1}, P_{0|-1})$ and the algebraic states are assumed to be consistent with the states and (9b). Furthermore, $\frac{\partial g}{\partial z}$ is assumed to be non-singular, i.e. the system is assumed to be of index-1.

By the implicit function theorem, the algebraic states, $\boldsymbol{z}(t)$, may formally be expressed as a function of the states, $\boldsymbol{x}(t)$, i.e. $\boldsymbol{z}(t) = \chi(\boldsymbol{x}(t))$. This implies that we may formally regard (9) as a continuous-discrete time SDE system (5) with $F(\boldsymbol{x}(t)) = f(\boldsymbol{x}(t), \chi(\boldsymbol{x}(t)))$ and $H(\boldsymbol{x}(t_k)) = h(\boldsymbol{x}(t_k), \chi(\boldsymbol{x}(t_k)))$. Consequently, if the algebraic states are eliminated using the implicit function theorem, the extended Kalman filter equations for the continuous-discrete time SDE system (5) apply to the continuous-discrete time SDAE system (9) as well. While the transformation of index-1 DAE systems to ODE systems is technically correct, it is not computationally efficient to solve index-1 DAE systems by this procedure. Instead of computing the mean evolution by (8a) we solve the following differential algebraic system for the mean evolution

$$\frac{d\hat{x}_k(t)}{dt} = f(\hat{x}_k(t), \hat{z}_k(t)) \qquad \hat{x}_k(t_k) = \hat{x}_{k|k} \tag{10a}$$

$$0 = g(\hat{x}_k(t), \hat{z}_k(t)) \tag{10b}$$

Similarly, the state sensitivities, $\Phi_{xx}(t,s) = \frac{\partial \hat{x}_k(t)}{\partial \hat{x}_k(s)}$ and $\Phi_{zx}(t,s) = \frac{\partial \hat{z}_k(t)}{\partial \hat{x}_k(s)}$, are computed as the corresponding sensitivities for index-1 DAE systems [8]

$$\frac{d\Phi_{xx}(t,s)}{dt} = \left(\frac{\partial f}{\partial x}(\hat{x}_k(t), \hat{z}_k(t))\right)\Phi_{xx}(t,s) + \left(\frac{\partial f}{\partial z}(\hat{x}_k(t), \hat{z}_k(t))\right)\Phi_{zx}(t,s)$$

$$\Phi_{xx}(s,s) = I \tag{10c}$$

$$0 = \left(\frac{\partial g}{\partial x}(\hat{x}_k(t), \hat{z}_k(t))\right)\Phi_{xx}(t,s) + \left(\frac{\partial g}{\partial z}(\hat{x}_k(t), \hat{z}_k(t))\right)\Phi_{zx}(t,s) \tag{10d}$$

rather than by solution of (8b). Finally, the state covariance, $P_k(t)$, is computed by quadrature

$$P_k(t) = \Phi_{xx}(t,t_k)P_{k|k}\Phi_{xx}(t,t_k)' + \int_{t_k}^{t} \Phi_{xx}(t,s)\sigma(s)\sigma(s)'\Phi_{xx}(t,s)'ds \tag{10e}$$

using the state sensitivities, $\Phi_{xx}(t,s)$. In conclusion, the one-step ahead predicted state mean, $\hat{x}_{k+1|k} = \hat{x}_k(t_{k+1})$, and covariance, $P_{k+1|k} = P_k(t_{k+1})$, in continuous-discrete time SDAE systems (9) is accomplished by solution of solution of (10). The one-step ahead prediction of the measurement is

$$\hat{y}_{k+1|k} = \hat{y}_k(t_{k+1}) = h(\hat{x}_k(t_{k+1}), \hat{z}_k(t_{k+1})) = h(\hat{x}_{k+1|k}, \hat{z}_{k+1|k}) \tag{11}$$

in which $\hat{z}_{k+1|k} = \hat{z}_k(t_{k+1})$. The filter equations of the extended Kalman filter for (9) are identical to the filter equations (3) for the discrete-time system (1). In particular, it should be noted that C_k is computed using the implicit function theorem, i.e.

$$C_k = \frac{\partial H}{\partial x}(\hat{x}_{k|k-1}) = \left[\left(\frac{\partial h}{\partial x}\right) - \left(\frac{\partial h}{\partial z}\right)\left(\frac{\partial g}{\partial z}\right)^{-1}\left(\frac{\partial g}{\partial x}\right)\right]_{(\hat{x}_{k|k-1}, \hat{z}_{k|k-1})} \tag{12}$$

3 Numerical Implementation

The efficiency of the numerical implementation of the extended Kalman filter for stochastic continuous-discrete time differential algebraic equations stems from efficient integration of the mean-covariance pair describing the evolution of the mean and covariance of the system.

3.1 ESDIRK Based Mean-Covariance Integration

The system (10) is integrated using an ESDIRK method with sensitivity computation capabilities [1, 8]. The ESDIRK method for integration of (10a)-(10b) consists of solution of the equations

$$X_1 = x_n \qquad Z_1 = z_n \qquad g(X_1, Z_1) = 0 \tag{13a}$$

$$R(X_i, Z_i) = \left(\begin{bmatrix} X_i \\ 0 \end{bmatrix} - \tau_n \gamma \begin{bmatrix} f(X_i, Z_i) \\ g(X_i, Z_i) \end{bmatrix} \right) - \left(\begin{bmatrix} x_n \\ 0 \end{bmatrix} + \sum_{j=1}^{i-1} \tau_n a_{ij} \begin{bmatrix} f(X_j, Z_j) \\ 0 \end{bmatrix} \right) = 0 \tag{13b}$$

with $X_i = x(T_i)$, $Z_i = z(T_i)$, $T_i = t_n + \tau_n c_i$, and $i = 2, 3, 4$. It is assumed that (x_0, z_0) is consistent, i.e. $g(x_0, z_0) = 0$. Note that in the applied ESDIRK method: $t_n = T_1$, $t_{n+1} = T_4$ and $x_{n+1} = x_n + \tau_n \sum_{j=1}^{4} b_j f(X_j, Z_j)$ with $b_j = a_{4j}$ for $j = 1, 2, 3, 4$. The method is stiffly accurate implying that $x_{n+1} = X_4$ and $z_{n+1} = Z_4$. By construction, the pair (x_{n+1}, z_{n+1}) is consistent with the algebraic relation, i.e. $g(x_{n+1}, z_{n+1}) = 0$. The integration error estimate, e_{n+1}, used by the step-length controller in the ESDIRK integration method is

$$e_{n+1} = \sum_{j=1}^{4} \tau_n d_j f(X_j, Z_j) \tag{13c}$$

Assuming that the initial pair (x_0, z_0) is consistent with the algebraic relation, i.e. $g(x_0, z_0) = 0$, the differential-algebraic equations (10a)-(10b) are integrated by solution of (13b) using a modified Newton method:

$$M \begin{bmatrix} \varDelta X \\ \varDelta Z \end{bmatrix} = R(X_i, Z_i), \quad M = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix} - \tau_n \gamma \begin{bmatrix} \frac{\partial f}{\partial x}(x_n, z_n) & \frac{\partial f}{\partial z}(x_n, z_n) \\ \frac{\partial g}{\partial x}(x_n, z_n) & \frac{\partial g}{\partial z}(x_n, z_n) \end{bmatrix} \tag{14a}$$

$$\begin{bmatrix} X_i \\ Z_i \end{bmatrix} \leftarrow \begin{bmatrix} X_i \\ Z_i \end{bmatrix} - \begin{bmatrix} \varDelta X \\ \varDelta Z \end{bmatrix}, \tag{14b}$$

Each step, $(\varDelta X, \varDelta Z)$, in the modified Newton method is generated by solution of a linear system using an approximate iteration matrix, M. Assuming, that the step-length controller selects the steps, τ_n, such that the Jacobian of (13b) and thus M are constant in each accepted step, the state sensitivities (10c)-(10d) with $s = t_n$ may be computed using a staggered direct approach [1, 8]

$$\begin{bmatrix} \varPhi_{xx}(T_1, t_n) \\ \varPhi_{zx}(T_1, t_n) \end{bmatrix} = \begin{bmatrix} I \\ -\left(\frac{\partial g}{\partial z}(x_n, z_n)\right)^{-1} \left(\frac{\partial g}{\partial x}(x_n, z_n)\right) \end{bmatrix} \tag{15a}$$

$$M \begin{bmatrix} \varPhi_{xx}(T_i, t_n) \\ \varPhi_{zx}(T_i, t_n) \end{bmatrix} = \begin{bmatrix} I \\ 0 \end{bmatrix} + \begin{bmatrix} \frac{\partial f}{\partial x}(x_n, z_n) & \frac{\partial f}{\partial z}(x_n, z_n) \\ \frac{\partial g}{\partial x}(x_n, z_n) & \frac{\partial g}{\partial z}(x_n, z_n) \end{bmatrix} \sum_{j=1}^{i-1} \tau_n a_{ij} \begin{bmatrix} \varPhi_{xx}(T_j, t_n) \\ \varPhi_{zx}(T_j, t_n) \end{bmatrix} \tag{15b}$$

with $i = 2, 3, 4$. The linear equations (15b) are solved reusing the LU-factorization of the iteration matrix, M, from the Newton steps in the integration of the index-1 DAE system itself. The assumption of a constant Jacobian in each accepted step implies that the state covariance (10e) may be computed as [2]

$$\begin{aligned} P_k(t_{n+1}) &= \varPhi_{xx}(T_4, t_n) P_k(t_n) \varPhi_{xx}(T_4, t_n)' \\ &\quad + \sum_{j=1}^{4} \tau_n b_j \varPhi_{xx}(T_j, t_n) \sigma(t_{n+1} - c_j \tau_n) \sigma(t_{n+1} - c_j \tau_n)' \varPhi_{xx}(T_j, t_n)' \end{aligned} \tag{16}$$

in which the quadrature formula of the ESDIRK method is used for derivation of the equation. This procedure for the state covariance evolution is initialized with $P_k(t_0) = P_{k|k}$.

4 Discussion and Conclusion

The proposed continuous-discrete time extended Kalman filter algorithm has direct applications in nonlinear model predictive control for state estimation. Given the stochastic model (9) and arrival of a new measurement, y_k, the filtered states, $\hat{x}_{k|k}$, and algebraic variables, $\hat{z}_{k|k}$, are computed using the extended Kalman filter. The regulator part of the nonlinear model predictive controller applies (10a)-(10b) as predictor with $\hat{x}_{k|k}$ and $\hat{z}_{k|k}$ as consistent initial conditions [7, 8]. For such an NMPC application numerical robustness and efficiency of the extended Kalman filter is of course important. However, numerical robustness and efficiency of the extended Kalman filter is even more significant when it is applied in systematic grey-box modelling of stochastic systems[3]. In such applications, numerical computation of e.g. the one-step ahead maximum-likelihood parameter estimate requires repeated evaluation of the negative log-likelihood function for parameters, θ, set by a numerical optimization algorithm during the course of an optimization. Compared to currently practiced grey-box identification in stochastic models, the algorithm proposed in this paper is significantly (more than two orders of magnitude for a system with 50 states) faster and has been extended to continuous-discrete time stochastic differential-algebraic systems (9). The proposed systematic estimation of the deterministic and stochastic part of the EKF-predictor represents an alternative to output-error estimation of the drift terms and covariance matching for the process and measurement noise covariance.

References

[1] Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G. & Jørgensen, S. B. An ESDIRK method with sensitivity analysis capabilities. *Computers and Chemical Engineering* **28**, 2695–2707 (2004).

[2] Jørgensen, J. B., Kristensen, M. R., Thomsen, P. G. & Madsen, H. Efficient numerical implementation of the continuous-discrete extended kalman filter. *Submitted to Computers and Chemical Engineering* (2006).

[3] Kristensen, N. R., Madsen, H. & Jørgensen, S. B. Parameter estimation in stochastic grey-box models. *Automatica* **40**, 225–237 (2004).

[4] Kailath, T., Sayed, A. H. & Hassibi, B. *Linear Estimation* (Prentice Hall, 2000).

[5] Åström, K. J. *Introduction to Stochastic Control Theory* (Academic Press, 1970).

[6] Jazwinski, A. H. *Stochastic Processes and Filtering Theory* (Academic Press, 1970).

[7] Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G. & Jørgensen, S. B. Efficient sensitivity computation for nonlinear model predictive control. In Allgöwer, F. (ed.) *NOLCOS 2004, 6th IFAC-Symposium on Nonlinear Control Systems, September 01-04, 2004, Stuttgart, Germany*, 723–728 (IFAC, 2004).

[8] Kristensen, M. R., Jørgensen, J. B., Thomsen, P. G., Michelsen, M. L. & Jørgensen, S. B. Sensitivity analysis in index-1differential algebraic equations by ESDIRK methods. In *16th IFAC World Congress 2005* (IFAC, Prague, Czech Republic, 2005).

NLMPC: A Platform for Optimal Control of Feed- or Product-Flexible Manufacturing

R. Donald Bartusiak

Core Engineering Process Control Department, ExxonMobil Chemical Company,
4500 Bayway Drive, Baytown, TX 77522, USA
don.bartusiak@exxonmobil.com

Summary. Nonlinear model predictive controllers (NLMPC) using fundamental dynamic models and online nonlinear optimization have been in service in ExxonMobil Chemical since 1994. The NLMPC algorithm used in this work employs a state space formulation, a finite prediction horizon, a performance specification in terms of desired closed loop response characteristics for the outputs, and costs on incremental manipulated variable action. The controller can utilize fundamental or empirical models. The simulation and optimization problems are solved simultaneously using sequential quadratic programming (SQP). In the paper, we present results illustrating regulatory and grade transition (servo) control by NLMPC on several industrial polymerization processes. The paper outlines the NLMPC technology employed, describes the current status in industry for extending linear model predictive control to nonlinear processes or applying NLMPC directly, and identifies several needs for improvements to components of NLMPC.

1 Introduction

The motivation for ExxonMobil Chemical's work in nonlinear model predictive control (NLMPC) was to control a first-of-a-kind polymerization process that was started up in 1990. A 2-year research program was established that included collaborations with the University of Maryland for polymerization modeling [1] and with Georgia Tech for a controller [2]. The model developed during the collaboration was eventually used for closed loop control after additional modifications by ExxonMobil Chemical. The controller – a quasilinearized MPC approach – was not used in practice for reasons explained in Section 2.

Internal development of a nonlinear controller continued guided by the experience and exceptional skills of my colleague Robert W. Fontaine, reference system synthesis ideas [3], and numerical approaches to address constraints [4]. The objective was to develop a general-purpose nonlinear controller with the specific motivation of regulatory and grade transition control of polymerization systems. The end result of the internal development was an NLMPC controller that used the NOVA software (Plant Automation Solutions, Inc.) to solve the optimization problem. The first on-process application of the controller was commissioned in mid-1994. ExxonMobil Chemical patented the technology [5], and then licensed it to Dynamic Optimization Technology Products, Inc. (now Plant Automation Services).

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 367–381, 2007.
springerlink.com

During the past decade, ExxonMobil Chemical has concentrated its efforts in NLMPC on implementing applications. During this period, the general state of the art has progressed from a sense that NLMPC was computationally infeasible [6], through demonstrations of practicality by the research community [7], to the point where several process control vendors now offer nonlinear control products [8] [9].

In this paper, Section 2 outlines the control algorithm as licensed. Section 3 discusses modeling and parameter estimation. Section 4 provides closed loop results from several industrial processes. Section 5 discusses the current state of industrial practice and highlights needs for improvement of the technology.

2 Control Algorithm

The moving horizon controller that was developed solves the following nonlinear optimization problem online. In the objective function (Eqn. 1), J_1 is a cost on deviations of the controlled variables from their reference trajectories, J_2 is an economic cost associated with input, output, or state variables, J_3 is a cost on incremental moves of the manipulated variables, and μ_i are weights. The objective function is detailed below in Eqns. 10, 11, and 12.

$$\min_{u}(\mu_1 J_1 + \mu_2 J_2 + \mu_3 J_3) \tag{1}$$

subject to

$$\begin{aligned} 0 &= f(\dot{x}, x, u, v, p) \\ 0 &= g(x, u, v, p) \\ 0 &= y - h(x, u, v, p) \end{aligned} \tag{2}$$

$$x(0) = x_0 \quad and \quad y(0) = y_0 \tag{3}$$

$$\dot{y} = (SP_h - (y + b))/\tau_c + P_h - S_h \tag{4}$$

$$\dot{y} = (SP_l - (y + b))/\tau_c + S_l - P_l \tag{5}$$

$$P_h,\ S_h,\ P_l,\ S_l \geq 0 \tag{6}$$

$$b = y_{meas} - y \tag{7}$$

$$u_{LB} \leq u \leq u_{UB} \tag{8}$$

$$|u_k - u_{k-1}| \leq \Delta u_B \tag{9}$$

Equations 2 and 3 define the process model in differential algebraic equation (DAE) form. x are the states, u are the manipulated inputs, v are the measured disturbances or disturbance model variables, p are parameters, and y are the outputs. In this problem statement x, u, v, and y are in absolute, not deviation, terms.

Equations 4 and 5 are examples of specifications for the reference trajectories for the desired closed loop behavior of the outputs. In this example, a first-order response is requested with a time constant of τ_c. SP_h and SP_l are the high and low setpoint (target) values for the outputs that defines the allowed settling zone. P_h and P_l are 1-norm penalty variables on deviations of the output from the reference trajectory. S_h and S_l are the corresponding slack variables (i.e. not costed in the objective function).

Output feedback is incorporated in additive form in Eqn. 7. For scaling reasons, we sometimes incorporate output feedback in multiplicative form.

Equations 8 and 9 impose absolute and incremental bounds on the manipulated variables. The subscripts k and $k-1$ refer to adjacent zero-order hold values of the manipulated variables across the entire control horizon.

Objective Function

In the objective function, the J_1 term is a weighted 1-norm of errors from the desired closed loop output trajectory over the prediction horizon.

$$J_1 = \frac{1}{n_p}\sum_{i=1}^{n_y}\sum_{k=1}^{n_p}(w_{h_i}P_{h_{i,k}} + w_{l_i}P_{l_{i,k}}) \tag{10}$$

Conceptually, this can be depicted as a conic section of unpenalized trajectories with linear penalties assigned to trajectories outside of the section. n_p is the length of the prediction horizon. n_y is the number of outputs. The significance of the 1-norm is a distinct relaxation of soft constraints from lowest ranked to highest ranked. This one-at-a-time relaxation is more aligned with industrial expectations as opposed to 2-norm behavior which spreads error across multiple outputs when constraints become active [10].

The J_2 term is an economic cost whose mean value over the prediction horizon is minimized to specify where within the allowed zone the system will settle. A useful alternative explanation is that $-J_2$ is a net income (product value - cost of feed and control) to be maximized.

$$J_2 = \frac{1}{n_p}(\sum_{i=1}^{n_y}\sum_{k=1}^{n_p}c_{y_i}y_{i,k} + \sum_{m=1}^{n_u}\sum_{k=1}^{n_p}c_{u_m}u_{m,k} + \sum_{j=1}^{n_v}\sum_{k=1}^{n_p}c_{v_i}v_{j,k}) \tag{11}$$

n_u is the number of manipulated inputs. n_v is the number of measured disturbance variables.

The J_3 term is the cost of incremental moves of the manipulated variables.

$$J_3 = \sum_{m=1}^{n_u}\sum_{l=1}^{n_c}c_{\Delta u_m}\left|\Delta u_{m,l}\right| \tag{12}$$

Solution Method, Initialization, and State Estimation

The optimization problem is solved using a sequential quadratic program (SQP). The DAE system is discretized by the optimization code using orthogonal collocation. Piecewise linear moves are calculated across a finite control horizon. By a simple configuration, the prediction horizon can be extended beyond the control horizon.

The prediction, control, and optimization problems are solved simultaneously. The discretized DAE system and resultant nonlinear program (NLP) are solved using an SQP algorithm. The previous solution of the controller is used as the initial condition array.

During development of this NLMPC algorithm, we found that the simultaneous solution approach was computationally faster than a shooting method approach [12] in which the prediction and control move calculations are separated and solved recursively.

The prediction, control, and optimization problems are solved simultaneously using an initial condition array from the previous solution. A shooting method is not used.

Regarding state estimation and initialization, the controller has an embedded full-order observer. The system can be initialized internally by the solution of the steady state DAE system (Eqn. 13). Alternatively, the initial condition can be supplied externally.

$$\begin{aligned} 0 &= f(0, x_0, u, v, p) \\ 0 &= g(x_0, u, v, p) \\ 0 &= y_0 - h(x_0, u, v, p) \end{aligned} \tag{13}$$

State feedback and input disturbance models can be incorporated in the DAE system during model development. The state estimation schemes used to date capture some of the benefit of incorporating feedback as input disturbances [13] [14], but are less comprehensive than an extended Kalman filter or moving horizon state estimator [15].

The controller used to date does not incorporate a terminal state condition [16].

Characteristics of Closed Loop Performance

The objective function (Eqn. 1) differs from that of the linear quadratic regulator (LQR). J_1 in Eqn. 1 in effect incorporates a reference trajectory $y_{ref}(t)$ instead of a fixed target in the calculation of the output error. This is a reference system synthesis or pole placement design method that results in closed loop output dynamics that are invariant throughout the nonlinear operating space when constraints are not active.

A pole placement performance specification is not sufficient when there is an excess of manipulated inputs compared to outputs. Analogous to LQR, the cost on the manipulated variable action provided by J_3 serves to uniquely determine the control law, at least for the linear case.

Avoiding operating point-specific tuning – alternatively stated, achieving invariance of the closed loop output dynamics – was an important design basis for our nonlinear controller development project. The move suppression based tuning specification of LQR and traditional MPC results in output dynamics that are a function of the operating point for the nonlinear case. This is illustrated in Figure 1 from a simulation study detailed in [17]. Invariance of the output dynamics was a primary reason why the quasilinearized MPC formulation was not pursued during the development project.

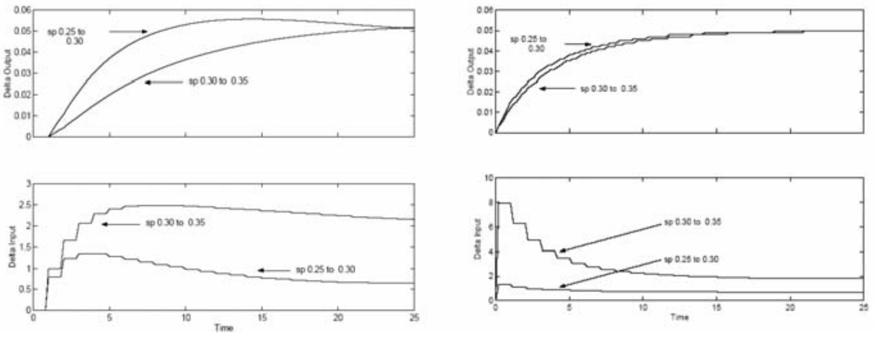


Fig. 1. Operating point dependent output dynamics with quasilinearized MPC formulation. (left) Invariant output dynamics with the NLMPC formulation using Eqn. (1) objective function. (right)

3 Models and Parameter Estimation

The majority of the models used to date in ExxonMobil Chemical's NLMPC applications consists primarily of first-principles elements with some empirical elements. For example, a polymerization model that uses fundamentals to predict the statistical moments of molecular weight distribution may use regressions for polymer end use properties as a function of the moments, the comonomer composition, etc. A minority of the NLMPC models consists primarily of empirical elements.

The process model will have various parameters that must be specified, such as equipment volumes, physical property constants, kinetic constants, coefficients in empirical models, and tuning parameters in embedded models of regulatory controls. Most of the fitting work is directed at kinetic constants and coefficients in empirical models. The tuning parameters in the embedded models of regulatory controls can be calculated directly, or estimated from step tests.

The primary means of estimating parameters, p, is weighted least squares minimization with steady state data collected from process history.

$$\min_p \left(\sum_{j=1}^{N_{OBS}} \sum_{i=1}^{N_Y} w_i^Y \left(\frac{y_{i,j} - y_{i,j}^{MEAS}}{y_{i,j}^{SCALE}} \right)^2 \right) \tag{14}$$

y are the model predicted values of the outputs, y^{MEAS} are the measured values, y^{SCALE} are scaling factors. w^y are weighting factors. N_y is the number of outputs. N_{OBS} is the number of observations in the dataset being processed. p are the parameters to be estimated.

It is important to know what parameters are identifiable given the structure of the model and the available data. It is important to have a balanced dataset that spans the operating space.

Our experience has been that more art than most engineers expect is required to estimate and validate parameter estimates. Process knowledge has been required to improve initial conditions and to shape the estimation process (e.g. define relationships among parameters, fit subsets of parameters at a time) to yield good results. There are opportunities to improve the determination of identifiable parameters, data mining, and global optimization applied to this task.

4 Results from Industrial Processes

ExxonMobil Chemical has concentrated on polymerization reactor control for the early use of this technology. To date, NLMPC applications have been implemented on five different classes of polyolefin polymerization processes summarized in Table 1. There are multiple commissioned applications for each of the processes in Table 1, except for "A".

Table 1. Characteristics of NLMPC applications on five different polyolefin processes in ExxonMobil Chemicals. (DAE is number of differential algebraic equations (Eqns. 2), CV is controlled variables, MV is manipulated variables, and FF is feedforward (measured disturbance) variables.)

Process	DAE	CV	MV	FF
A	21	2	2	3
B	42	8	5	7
C	128	4	4	22
D	21	2	2	16
E	2300	6	3	31

These applications are deployed on a variety of HP Alpha servers running OpenVMS, or Dell servers running Windows. The controller scan times are in the range of 3 to 6 minutes. Note this statement of controller scan times includes Process E with 2300 DAE equations before discretization. Typical control horizon lengths are in the range of 5 to 10 scan periods.

Nonlinearity

For the first set of results from industrial processes, let's isolate on one input/output relationship that clearly illustrates the process nonlinearity. The

Table 2. Steady state gains at two operating points in a grade transition scenario

	$\frac{\partial CV_1}{\partial MV_1}$	$\frac{\partial CV_1}{\partial MV_2}$	$\frac{\partial CV_2}{\partial MV_1}$	$\frac{\partial CV_2}{\partial MV_2}$
$CV_1 = 125$	2.68	6.09	0.033	1.08
$CV_1 = 1.2$	0.0238	0.0696	0.012	0.556
Fractional change of gain	113	87.5	2.75	1.94

case is a polymer grade transition where the steady state gain changes by two orders of magnitude.

Figure 2 illustrates the CV_1 and MV_1 transients for a transition in CV_1 from 125 to 1.2. The lower curve in the CV_1 plot is the one being controlled. It is easy to see the increasing amount of manipulated variable action required to achieve the controlled variable trajectory requested.

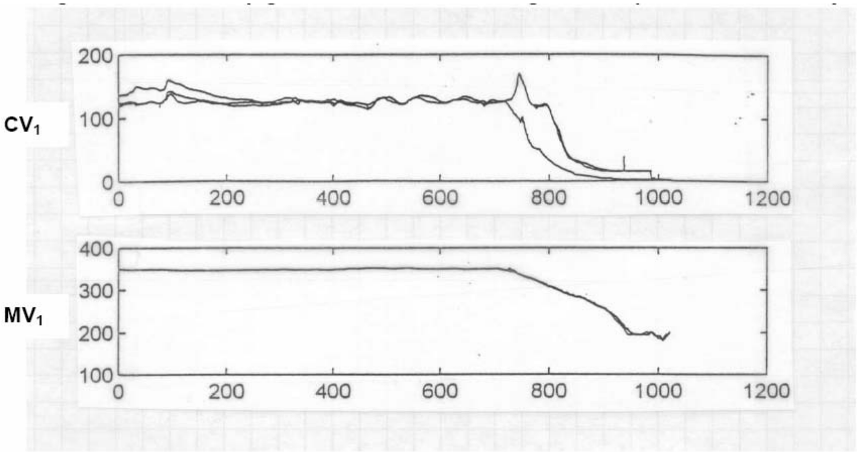


Fig. 2. Closed loop grade transition illustrating effect of process nonlinearity

Constraint Handling

Handling the nonlinearity alone is not sufficient justification for MPC. The second set of results, presented in Figure 3, illustrates the importance of constraint handling and the MPC formulation. The case is a grade transition with a 5 MV by 8 CV application. In the transition, we want to change CV_1 and keep CV_3 constant.

Figure 3 shows that two of the manipulated variables are saturated during most of the transition. This application also has a constraint-controlled variable (CV_2) – one that must be kept within a range, such as a temperature limit or a fouling condition – which also becomes active during the transition. With all of these constraints active, the controller holds the higher priority CV_3 at its setpoint, gives up on matching the requested CV_1 trajectory, and does the best it can by making moves when it can, for example in MV_1, whenever CV_2 becomes unconstrained.

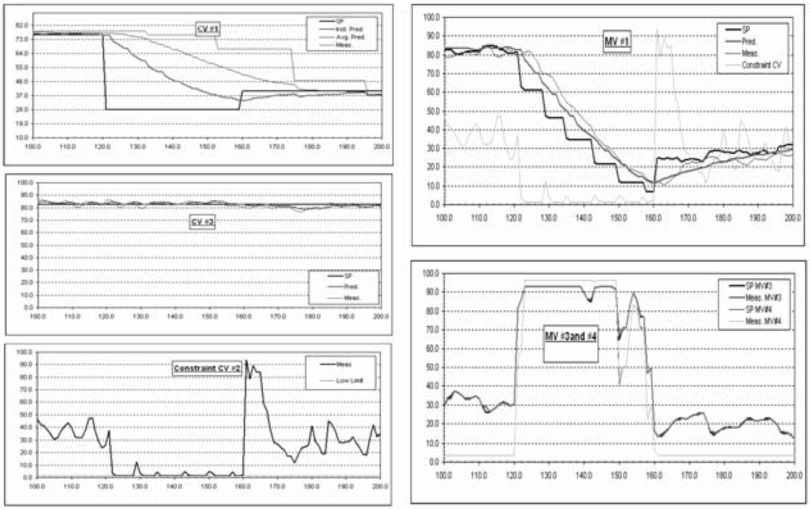


Fig. 3. Closed loop grade transition illustrating handling of process constraints

Dynamic Decoupling

The third set of results illustrates decoupling and the value of the dynamic nonlinear model. Three grade transitions occur during the time series presented in Figure 4.

In the first transition, only one of the CV's is transitioned. The other CV's are regulated within their target range.

In the second and third transitions, setpoints for both of the CV's depicted are changed. The controller makes moves in the MV's such that each CV follows its requested trajectory with negligible interactions.

Grade Transitions: Variance Reduction

The final set of results illustrates grade transition data before- and after installing the controller.

Note that it is easy to speed up transition times with a controller when the manual transitions were done without pushing the unit to constraints, such as minimum or maximum flows of reactants or utilities streams. It's a more challenging problem to justify the investment in a controller for a unit that already has aggressive, constraint-pushing manual procedures that have been refined during years of operation. This is the situation addressed in the next set of results.

Figure 5 depicts half-a-dozen each of manual and NLMPC grade transitions. Because the manual transitions were already pushing to limiting constraints, you would have to modify the facilities to significantly speed up the best transition. What we've seen in practice, however, is depicted in the charts on the right. Closed loop control can significantly improve grade transition performance by reducing product quality variance in even the best-run units.

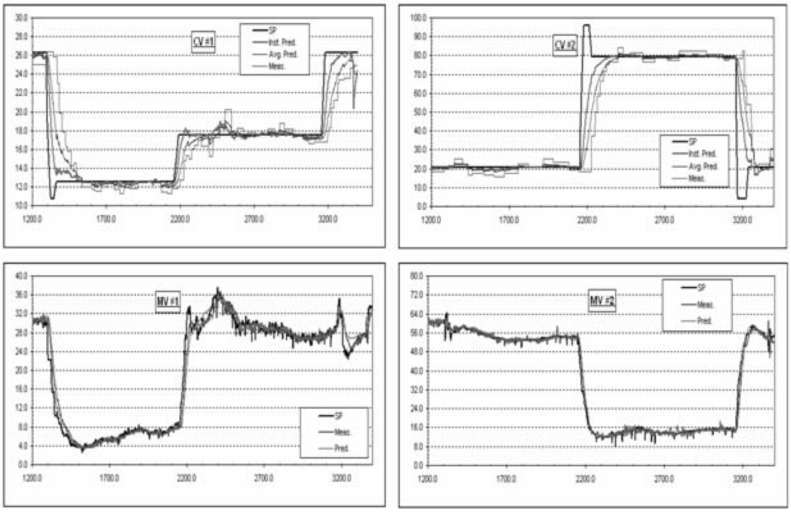


Fig. 4. Dynamic decoupling during multivariable grade transitions

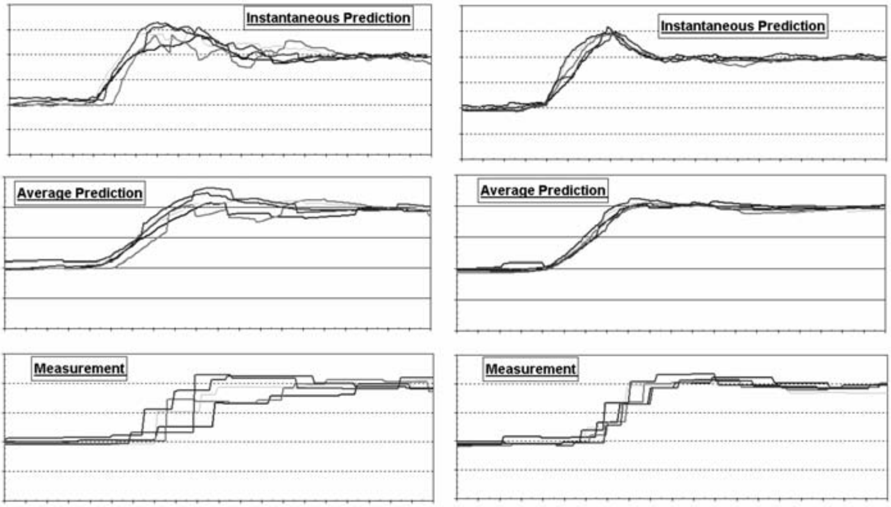


Fig. 5. Grade transition reproducibility before (left) and after (right) closed loop control

Other Benefits

The NLMPC technology and fundamental models have provided several additional benefits, as follows:

- Ability to transition to and operate reactors on grades that were never manufactured before or included in the parameter estimation dataset.

- Can transfer applications from one unit to another, or modify applications after facilities expansion projects without much effort. It is a straightforward matter to revise equipment-related parameters. Also, we've been able to transfer kinetic parameter values, although clearly caution is advised and one must always plan to refit data.
- Deliver the NLMPC applications close to start-up of new facilities.
- Precise and rapid detection of instrument faults via insights gained from large residual errors in individual model equations.

5 Concluding Remarks

This section provides a brief outlook on industrial practice for extending existing model predictive control technology to nonlinear systems, a summary of the NLMPC work described in the paper, and outlook remarks regarding the evolution of NLMPC.

5.1 Extensions of Existing MPC and Real-Time Optimization Technologies

Various techniques have been used or are emerging in industrial practice to extend the applicability of LMPC technologies to nonlinear systems.

For more than a decade, small-scale nonlinearities such as flow rate to valve position effects or composition dependencies in distillation towers have been effectively handled with mathematical transforms.

Within the past 5 to 10 years, LMPC's applicability in the face of nonlinearities has been extended via gain updating or gain scheduling approaches. Infrequent, event-driven gain update approaches are more common than continual gain scheduling. Gain updating is typically used for feed-flexible plants where there are relatively few permutations of feed types that affect the process response.

Current continuous improvement efforts on the model identification task, e.g., automated plant test tools [11] [18], may increase the applicability of LMPC further, at least for regulatory control. The state of the art is moving in the direction of human-supervised adaptation. However, it is unlikely that adaptation – supervised or not – will further enable LMPC technology to solve the servo (grade transition) problem for nonlinear systems.

What industry calls real time optimization (RTO) represents another means of implementing nonlinear compensation in constrained multivariable control. Specifically, RTO implies the use of a steady state model and a nonlinear optimization program to set output (and possibly input) targets for an underlying layer of LMPC's or other regulators. Introducing RTO into this discussion is relevant because of its architectural similarity to current ideas for implementing NLMPC that involve separating the target setting function from the regulator function [19] [20].

RTO is an old idea – arguably the driving force for computer process control almost 50 years ago [21]. The abiding motivation is to achieve market price driven economic optimization of the process on a large scale, e.g. an entire olefins plant. Original motivations as a multivariable regulator have been made obsolete by LMPC. While it may be an old idea, RTO's impact has increased during the past decade as a result of enormous improvements in the software used, and by the accumulation of practical engineering experience. As described in [22], enterprise-wide optimization, including RTO, continues to be an important goal for industry.

In general, RTO is used for continuous or infrequently changed semi-continuous processes – not for the polymerization processes discussed in Section 4. Where RTO is used, it is most common that it only supplies targets to LMPC's or other linear regulators that are not compensated for process nonlinearities. Less common is the case where gain updating is used to compensate the LMPC's, as discussed in the preceding section. Possible with the current state of the art, but even more rare, is the case where the nonlinear RTO model is used to continually update the gains in the LMPC's.

It is beyond the scope of this paper to comment further on RTO. However, it is important to acknowledge the following three contributions from RTO work that were significant in enabling ExxonMobil Chemical's reduction of nonlinear model predictive control theory to industrial practice: (1) developments of online nonlinear programming technology, (2) hardware and software tools, and (3) the practical experience we gained doing RTO applications [23].

5.2 NLMPC Summary and Outlook

NLMPC using fundamental models and online solution of the nonlinear optimization problem across the prediction horizon is practical in industrial service. The technology is being used to automate state (product grade) transitions (servo control) and to achieve operating point-independent regulatory control of processes highly nonlinear for LMPC.

The size of the applications implemented so far is small by input/output count compared to current practices for LMPC applications. The scope and size of the NLMPC applications is increasing as the technology matures and the experience base grows.

Algorithm

The objective function used in this work combines the minimization of a net operating cost, output errors against reference trajectories, and cost of incremental manipulated variable movement. This objective function is different from LQR particularly regarding the use of the 1-norm for the output error and the use of a reference trajectory over the prediction horizon as a primary means of specifying closed loop performance.

The solution method used in this work is to simultaneously solve the simulation and optimization problem using an SQP code.

On the general question of the NLMPC algorithm, one key technical question is whether it is necessary to retain the nonlinear relationships across the prediction horizon, or is linearization at one or more points satisfactory. Even for the finite horizon NLMPC presented in this paper, Figure 1 illustrates a case where significant process gain changes occur within the span of the prediction horizon. There is a trade-off to be made among accuracy of the control moves calculated on each scan, computational effort, and the number of "moving parts" in the controller software (more for quasilinearization).

A second technical question pertains to the value of invariance of the closed loop speed of response. Operating point independent tuning was an essential requirement that led to the algorithm used in this work. In practice, it has been easy to choose the tuning values for the desired closed loop response using a basic knowledge of the process dynamics. When required, specifying the costs on the incremental MV moves has involved guesswork and required more time for simulations.

Models

Regarding models, either fundamental or empirical or a combination can be used with the NLMPC method described in this paper. For ExxonMobil Chemical's the initial set of applications, fundamental models were preferred and developed – justified by their capability to be extrapolated and ported to multiple facilities.

On the general question about models for NLMPC, Lee provides an excellent breakdown of the issues and research requirements for fundamental and empirical models, order reduction, etc [24]. Industry's likely path forward is as follows. Where possible and profitable, libraries of fit-for-purpose process models will be developed and continuously improved. Where library models are not possible or profitable, empirical modeling tools will be used. Hybrid modeling approaches will likely be used in any given application – with empirical elements used to simplify where possible or to provide the "lubrication" for required effects that are too costly to model precisely. There is no substitute for experience for deciding how to simplify. As computationally more-efficient models are developed, the scope of applications attempted will increase.

A common approach for empirical nonlinear dynamic models used in industry is to combine a static nonlinear part with a linear dynamic part (Hammerstein model). For the purposes of this discussion, it does not matter whether the nonlinearity is represented by equations, neural nets, etc. Such a modeling basis in NLMPC will indeed work for many practical problems, but may be vulnerable to nonlinear dynamics induced by changes in plant throughput or even when trying to control startups and shutdowns.

An additional motivation for using fundamental models in MPC is the promise of lower lifecycle costs of applications versus empirical models in the face of "clonable" units and on-going facilities changes. An example of the clonable unit challenge is distillation tower control. At least for nonexotic towers, the notion is that configurable controllers could be developed via a preloaded simulation model together with order reduction and state estimation schemes. If dynamic simulation to closed loop controllers is starting to happen in practice [25], why not configurable MPC's? For an example of low cost revision after a facility change, we have already successfully revised an NLMPC application after a debottleneck project simply by changing the parameters for volumes of vessels and piping, the location of feed injection locations, etc. without any refitting to plant data.

Parameter Estimation

One of the advantages of fundamental models over empirical dynamic ones is that fewer data are required to fit the adjustable parameters. Indeed, this has been our experience. Relatively few steady-state datasets have been sufficient to estimate parameter values that yielded satisfactory closed loop control performance. Nevertheless, after fundamental modeling itself, the NLMPC development task that has been characterized by the most art was estimation of parameters in the fundamental models.

To the casual reader of global optimization literature, it seems that estimation of parameters in nonlinear models for NLMPC is a tractable problem. We'll leave this statement here without further elaboration. The needs for fundamental technology improvements and new computer tools should be obvious.

State Estimation

State estimation is an essential part of the NLMPC described in this paper. The initial applications used the state estimator only to propagate unmeasured state values forward. In time, we started to incorporate input disturbance models into NLMPC applications with generally favorable but not "game breaking" results.

The incorporation of state estimators with postulated disturbance models to improve the disturbance rejection properties of commercial LMPC products is still a recent, and not ubiquitous, development. It's interesting to note Eastman Chemical's incorporation of state space and infinite horizon features in its inhouse MPC well in advance of commercial LMPC products [26].

The outlook is that optimal state estimators and input disturbance models will be incorporated into the NLMPC toolset. Contrasting the extended Kalman filter (EKF) and the moving horizon estimator (MHE) approaches [27], we note a preference for EKF based on the lesser computational requirement. MHE is likely to be particularly good for the problem of infrequent, uncertain lab measurements, but it should be a second option based on our experience to date.

Implementation and Sustainment

The predictability and efficiency of controller implementation projects and the service factors of the commissioned applications are functions of the quality and usability of the software tools used. This is worth noting, but this paper is not the proper forum to detail the requirements for software features, user interfaces, etc.

The NLMPC engineer must be skilled at properly diagnosing and correcting convergence or singularity problems in nonlinear programs, particularly when the model is presented with the potentially inconsistent real-time process data. There is little evidence that these skills are acquired in the university, and little hope of dramatic breakthroughs for software tools for precise diagnostics. On-the-job training has been the best bet.

Operators and process engineers must have sufficient understanding and trust in the NLMPC application for it to be sustained at full effectiveness. This need is particularly acute when the NLMPC is controlling a unit where only SISO controls existed before. Pre-commissioning training; timely, process-relevant explanations of controller behavior; and what-if simulation studies are some measures that can increase understanding and trust.

Closing Remarks

NLMPC is practical in industrial service. We have no doubt that the process control community – academics, vendors, and operating companies – will continuously improve on the technology described or surveyed in this paper. Dogma on the choice between fundamental and empirical models used across-the-board for NLMPC is doomed to be erroneous. NLMPC is set to be the platform for optimal control of feed- or product flexible manufacturing processes characterized by significant nonlinearities.

Acknowledgements

The principal contributors to the work described in this paper include Robert W. Fontaine, Carl O. Schwanke, and Arul R. Gomatam. Other contributors to Exxon-Mobil Chemicals' NLMPC work are listed in [28]. The author is grateful to William F. Floyd for comments that helped converge the topics to be included in this paper, and to Christopher E. Long for highlighting points requiring clarification.

References

[1] Choi KY, Ray WH (1985) Chem Eng Sci (1985) 40:2261-2279.
[2] Peterson T, Hernandez E, Arkun Y, Schork J (1992) Chem Eng Sci. 47:737-753.
[3] Bartusiak RD, Georgakis CG, Reilly MJ (1988) Chem Eng Sci 44:1837-1851.
[4] Li WC, Biegler LT (1988) Ind & Eng Chem Res. 27:1421-1433.
[5] Bartusiak RD, Fontaine RW (1997) Feedback method for controlling nonlinear processes. U.S. Patent 5,682,309.
[6] Bequette BW (1991) Ind Eng Chem Res 30:1391-1413.
[7] Findeisen RH, Allgower F(2000) Nonlinear model predictive control for index-one DAE systems. In: Allgower F, Zheng A (eds) Nonlinear Model Predictive Control. Birkhauser Verlag, Basel.
[8] Qin SJ, Badgwell TA (2000) An overview of nonlinear model predictive control applications. In: Allgower F, Zheng A (eds) Nonlinear Model Predictive Control. Birkhauser Verlag, Basel.
[9] Qin SJ, Badgwell TA (2003) Control Eng Practice 11:733-764.
[10] Saffer DR, Doyle FJ (2004) Comp & Chem Eng. 41:2749-2793.
[11] Snow WP, Emigholz KF, Zhu Y (2001) Petroleum Technology Quarterly. 02:97-101.
[12] Martinsen F, Biegler LT, Foss BA (2004) J Proc Control. 14:853-865.
[13] Lee JH, Morari M, Garcia CE (1994) Automatica. 30:707-717.

[14] Lundstrom P, Lee JH, Morari M, Skogestad S (1995) Comp & Chem Eng. 19:409-421.

[15] Robertson DG, Lee JH (1995) J Proc Cont 5:291-299.

[16] Rawlings JB, Muske KR (1993) IEEE Trans Auto Control. 38:1512-1516.

[17] Young RE, Bartusiak RD, Fontaine RW (2002). Evolution of an industrial nonlinear model predictive controller. In: Rawlings JB, Ogunnaike BA, Eaton JW (eds) Sixth International Conference on Chemical Process Control. AIChE Symposium Series No. 326 Vol. 98. CACHE, AIChE.

[18] Harmse M, Hokanson DA (2005) Experiences in using SmartStep. AspenTech AC&O User Group meeting, League City TX.

[19] Bindlish R, Rawlings JB (2003) AIChE J 49:2885-2899.

[20] Helbig A, Abel O, Marquardt W (2000) Structural concepts for optimization based control of transient processes. In: Allgower F, Zheng A (eds) Nonlinear Model Predictive Control. Birkhauser Verlag, Basel.

[21] Astrom KJ, Wittenmark B (1984) Computer Controlled Systems Theory and Design. Prentice-Hall, Englewood Cliffs.

[22] Grossmann I (2005) AIChE J. 51:1846-1857.

[23] Gallun SE, Luecke RH, Scott DE Morshedi AM (1992) Hydrocarbon Proc. 71:78-82.

[24] Lee JH (2000) Modeling and identification in nonlinear model predictive control: Requirements, current status and future research needs. In: Allgower F, Zheng A (eds) Nonlinear Model Predictive Control Birkhauser Verlag, Basel.

[25] Vragolic S (2004) Alternate methods of process model identification: Benzene distillation. Proceedings of the 2004 Ethylene Producers Conference, AIChE.

[26] Vogel EF, Downs JJ (2002). Industrial experience with state space model predictive control. In: Rawlings JB, Ogunnaike BA, Eaton JW (eds) Sixth International Conference on Chemical Process Control. AIChE Symposium Series No. 326 Vol. 98. CACHE, AIChE.

[27] Heseltine EL, Rawlings JB (2005) Ind & Eng Chem Res. 44:2451-2460.

[28] Bartusiak RD, Fontaine RW, Schwanke CO, Gomatam AR (2003) Nonlinear model predictive control of polymerization reactors. AIChE Annual Meeting, San Francisco CA.

Experiences with Nonlinear MPC in Polymer Manufacturing

Kelvin Naidoo, John Guiver, Paul Turner, Mike Keenan, and Michael Harmse

Aspen Technology Inc., 2500 City West Blvd, Suite 1600, Houston,
Texas 77042, USA
john.guiver@aspentech.com

Summary. This paper discusses the implementation of nonlinear model predictive control on continuous industrial polymer manufacturing processes. Two examples of such processes serve to highlight many of the practical issues faced and the technological solutions that have been adopted. An outline is given of the various phases of deploying such a solution, and this serves as a framework for describing the relevant modeling choices, controller structures, controller tuning, and other practical issues

1 Introduction

Starting with a pilot implementation in 2001, Aspen Technology has gained a large amount of experience in the field in implementing fully non-linear MPC on several different types of continuous polymer manufacturing processes. There are many benefits obtained by putting MPC on these industrial units, but one of the main goals is to minimize production of off-spec material both in steady state operation and when transitioning from one product grade to another. Prior to 2001, implementations using empirical models were typically done with some form of gain adaptation or gain scheduling, which, though suitable for steady state operation, is sub-optimal for transitions. This is due to the fact that process gains often change by an order of magnitude or more over a relatively short period of time, and the non-linearities involved interact in a multivariate manner and cannot be removed by univariate transforms. Even if the gains are scheduled in a non-linear manner across the transition horizon [5], this does not take into account these interactions, and no optimized path can be calculated. A breakthrough came with the development of Bounded Derivative Network technology [6], which allowed the building of empirical, fast-executing, control-relevant models that could be embedded directly in a nonlinear control law, removing the need for gain scheduling completely.

There are many practical matters that impact the success of a control project including technology, process understanding, best-practice methodology, the use of reliable software with suitable functionality, and developing a deep understanding of practical operational requirements. The purpose of this paper is to touch on many of these issues and share best practice concepts around how

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 383–398, 2007.
springerlink.com

polymer control projects should be executed, what practical issues have to be faced, and to describe the solutions that have been adopted.

This paper starts with a brief description of some example industrial processes that will be referred to later in the paper. The remaining sections in the paper outline, in chronological order, the various phases of deploying one of these solutions. At each stage, technological and practical points of interest are discussed.

2 Process Descriptions

In this section we will briefly describe some typical polymer processes, which will serve to illuminate some of the later discussion. In the interests of space, we will limit the discussion to two specific polymer processes, but much of the discussion is common to a wide range of continuous polymer manufacturing technologies on which Aspen Technology has implemented nonlinear MPC, including Dow UNIPOL Polyethylene (PE) and Polypropylene (PP), Basell Spheripol Polypropylene (PP), Polystyrene, BP Innovene PE and PP, and High Pressure Tube Low Density Polyethylene (LDPE).

2.1 Dow UNIPOL PolyPropylene

The Dow UNIPOL PP process is a gas phase, dual series reactor process capable of making impact and random copolymer and homopolymer grades. The process consists of several sections including Raw Material Purification, Reactors, Resin degassing, Vent Recovery, Pelletizer, Blending and Bagging, Compounding, etc. MPC is usually applied to the reaction and related systems, but can be extended to the vent recovery unit. Figure 1 shows a schematic of the first reactor section.

On the reactors, there are both quality and composition variables requiring control. Quality variables relate to the powder properties, while composition

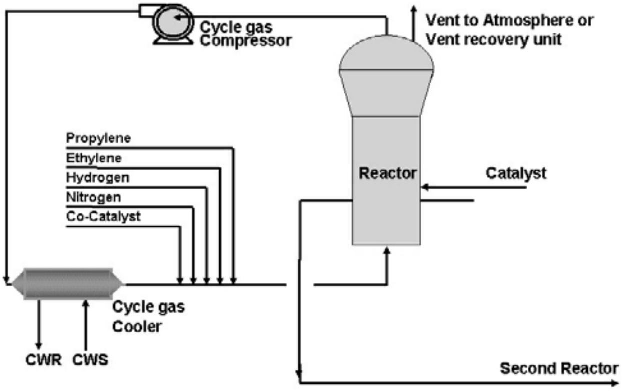


Fig. 1. First Reactor Section of Dow's UNIPOL PP Process

variables refer to the feed components in the reactor and to general reactor condition variables like temperature and pressure.

Perhaps the most critical controlled variable is the reactor temperature. This is too fast for conventional MPC and usually has to be a non-linear PID loop. If the temperature is unstable or poorly controlled, all other variables on the reactor, both quality and compositions will be difficult, if not impossible to maintain.

For MPC purposes, the controller design has to focus on those compositions and reactor conditions that have an impact on the various product quality attributes. To do this, the MPC will manipulate the feed and additive flow rates. To control the product quality, the controller has to determine the correct compositions and reactor conditions and control to these set-points.

Typically these processes are constrained by cooling capacity, powder discharge system limits or downstream extrusion constraints. The physical constraint set will differ from plant to plant. It is up to the MPC system to ensure that the process is always riding these constraints in order to maximize production or maintain powder qualities.

2.2 Polyethylene High Pressure Tube Reactor

One method of manufacturing low-density polyethylene is via a high-pressure tube reaction system. Figure 2 presents a schematic of a typical LDPE high-pressure tube process.

Ethylene is pressurised to over 2000 bar in two stages. A chain transfer agent is then injected which promotes longer chain molecules. A cocktail of initiator is injected at various points along the tube producing an exothermic reaction that peaks in temperature and then decays again until the polymer reaches the next initiator injection point. Increasing the injection of initiator increases the reaction resulting in a higher peak temperature and increased conversion. There

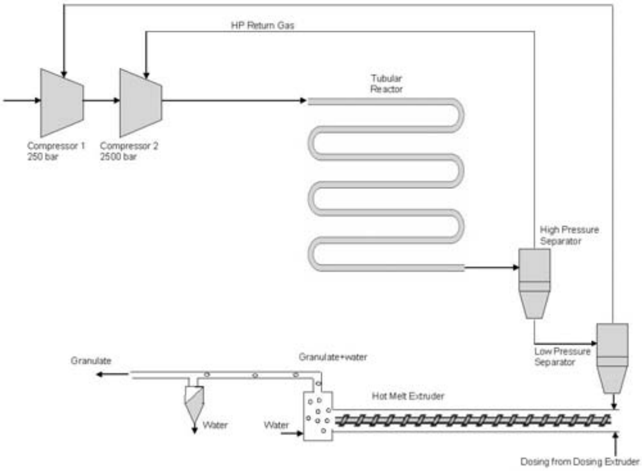


Fig. 2. High Pressure Tube LDPE Process

are two key quality parameters that are controlled. Melt Flow Index is a measure of polymer viscosity and increases as the molecular chain length increases. Another key parameter is gloss. The chain transfer agent and the temperatures are typical manipulated variables for product qualities.

At the end of the reaction the polymer is extruded and cut into fine pellets. These plants are capable of producing many different grades of polymer. The production schedule is determined by a Planning and Scheduling system to ensure that the entire supply chain operation is optimized. As a result, many polymer production lines have an intensive grade transition schedule requiring several transitions per week. MPC that is reliable across all product grades and a wide range of transitions is essential for providing the necessary degrees of freedom to the supply chain solution

3 Instrumentation Review and Pre-test

At the start of each project, a visit is made to the plant to assess the instrumentation, analyzers, valves, PID loops, and DCS issues, so as to establish a set of preliminary remedial actions. Ensuring that all the control valves, critical instruments and analyzers work properly, and tuning the PID loops properly is a pre-requisite for implementing the solution.

4 Controller Functional Design

The Controller Functional Design Specification (FDS) provides the preliminary controller design, which specifies:

1. Control objectives
2. Process constraints
3. Controller structure
4. Transition and recipe scope
5. System architecture

Every plant is different even within the same technology. The following issues have to be accounted for in the FDS to ensure that the controller design is suitable for the particular polymer line:

1. Different operation methodologies
2. Different process constraints
3. Different levels of operating expertise
4. Different feedstock qualities and different specifications
5. Different maintenance standards

It is essential to understand the specific process technology, as well as the process constraints that may be unique for a particular process unit, and to understand the unique operations requirements for the site.

4.1 General Solution Architecture

A typical architecture of a polymer APC solution is shown in Figure 3. Typically the controller system will use a cascaded structure with a master Quality Controller providing set-points for the slave Composition Controller. The advantages of a cascaded structure are highlighted in the next section.

The purpose of the upper tier controller is to control various quality measures. Feedback for these comes in the form of lab analysis - typically every 4 hours or so, with a significant delay. This lab information is managed by the Lab Update system, which compares the lab result with the continuous inferred quality value from the time that the lab sample was taken. Both the inferred quality value calculations and the controller use the same non-linear quality models, though one uses process values as input and the other uses set-points, as described in the next section. The Lab Update system calculates a bias, which is then applied to the controller quality model.

During transitions, lab feedback is typically not made available, so the quality models need to be very accurate to be able to drive the transitions.

The lower tier controller receives concentration ratio set-points from the upper tier controller and adjusts flow set-points, pressure and temperature set-points in the DCS. The feedback for the lower-tier model typically comes from fast online analyzers, though in some cases equation based calculations may be available for some process quantities.

Recipe data such as quality and process constraints are provided by the Transition Management module which is configured for the full set of products and product transitions for the particular production line.

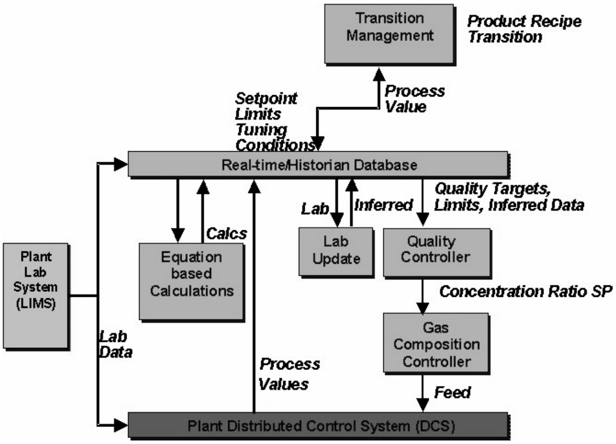

Transition Management
Product Recipe Transition
Setpoint Limits Tuning Conditions
Process Value
Real-time/Historian Database
Calcs
Lab
Inferred
Quality Targets, Limits, Inferred Data
Plant Lab System (LIMS)
Equation based Calculations
Lab Update
Quality Controller
Concentration Ratio SP
Lab Data
Gas Composition Controller
Process Values
Feed
Plant Distributed Control System (DCS)

Fig. 3. Typical Solution Architecture

4.2 Cascaded Structure

The cascaded structure is an important strategy in implementing polymer MPC systems. There are several practical benefits that a cascaded structure provides:

1. It greatly simplifies model maintenance. If a new product grade, or a new CV needs to be added, or the process changes in some way, it is not necessary to overhaul the composition controller.
2. It allows the top tier controller to be taken off-line while still running the lower-tier controller.
3. It facilitates a gas composition overshoot strategy, which significantly reduces the grade transition time.
4. It improves disturbance handling. If any of the feed composition variables experience a major process disturbance, the gas composition controller will immediately take action to reject it before the polymer product quality metrics are affected. If this were not done, the controller would only respond after the quality has moved, and usually this would only become apparent after a laboratory result is available. The cascaded structure, similar to cascaded PID loops, effectively gives the control solution an extra degree of freedom for disturbance rejection.

One potential problem with a cascaded strategy is that there is no guarantee that the lower tier slave controller will meet the targets required by the master controller. This is similar to a cascaded PID control scheme where the slave loop will saturate when the control valve goes fully open. The implication for the polymer APC system is that the master quality controller will continue to ramp (causing so-called wind-up), unless proper anti-windup detection is provided.

The solution for the cascaded MPC system is to configure two sets of product quality predictions at the upper tier. The first set of quality predictions are based on the concentration ratio targets (the gas concentration set-points), while the second set of predictions are based on the actual gas concentration process values - using the same models in each case. The latter are used to estimate any bias (or offset) between the upper tier property models, and the actual product quality feedback provided by the laboratory analysis. This bias is added to both sets of property predictions. The bias-updated target- based predictions (based on the gas composition set-point) are then used as feedback for the property controller. The controller will see any such bias as an unmeasured disturbance and will respond accordingly. In addition to this, anti-windup flags are used. If the steady state value of the lower tier controller's CV (the gas concentration ratio) is not equal (within tolerance limits) to the steady state target value provided by the property controller, a wind-up flag is set to the corresponding MV in the upper tier controller. If there are other handles available, the controller will then use those until the wind-up condition has passed.

4.3 Overshoot Strategy

The cascaded control structure facilitates a safe method for implementing an overshoot strategy on the plant. Polymers qualities, especially in the case of a UNIPOL gas phase reactor (this is effectively a continuously stirred tank reactor with a large inventory), have very slow process dynamics. During product quality transitions, it is essential that the closed loop responses be faster than these slow open loop process dynamics - otherwise, there will be several dozen tons of product of intermediate quality that is effectively off-spec. Furthermore, the longer the reactor spends making transition quality products, the less time there is for in-grade saleable products. As a result, the net annual production rate will be lower. In order to ensure more profitable process operation, the controller has to provide for the fastest possible grade transitions, which require an overshoot strategy to be employed where the flow targets (setpoints) need to dynamically overshoot their final steady state value by a large amount.

One of the limitations of replicating this strategy in a single controller is that the controller will complete all of its moves in a very small space of time and leave no room for inaccuracy in the models. The risk of this is that the actual product quality may overshoot its target and exceed upper/lower limits of the new grade, which extends the grade transition time rather than shortening it. An even riskier scheme is for a quality controller to overshoot the concentration targets in order to force a lower level PID flow controller to overshoot.

The ideal solution is for the lower tier concentration MPC controller to be tuned aggressively in order to cause the flow set-point to overshoot its final value by a large amount, but still ensuring that the gas concentration does *not* overshoot. Cascaded to this is the top tier quality controller. It is tuned for an over-damped response to ensure that the concentration set-points will not overshoot the final value. This provides the desired overshoot strategy (which is required in order to minimize the grade transitions) but guarantees that the gas concentration will not overshoot its final value, and thus reduces the risk of overshooting the product quality targets (which could extend the transition time by a large margin and create a lot of off-specification product).

With older linear technologies that utilised gain scheduling on the quality models, the overshoot strategy was implemented by utilizing two CVs per quality. The first being the real quality predictions with the real process dynamics (known as the bed average qualities) and the second being a quality prediction without dynamics or dead times - this is a steady state prediction (known as the instantaneous quality prediction). This approach was sometimes referred to as the analyzer-predictor method.

In this older approach, the bed average prediction is given the final quality target and the instantaneous prediction is given a target much higher or lower (depending on whether the quality is increasing or decreasing) than the final target. This forces the controller to move the concentration ratios more aggressively than it would normally since it is trying to meet the instantaneous quality target as well as the bed average. Once a minimum time has passed or the bed average prediction has reached a key point (this time or value is determined from simulation), the instantaneous CV is turned off and the controller then aims only for the bed average quality. If the instantaneous quality is not turned off, the controller will try to trade off the bed average and the instantaneous CVs since both CVs are using the same MVs. This means that the bed average will have moved further than anticipated and the instantaneous less than needed.

However, with the advances in non-linear control, the need for the instantaneous quality has been removed. The fact that the non-linear dynamic model is directly embedded in the control law means that the move plan is consistent from cycle to cycle. The speed of the transition and the overshoot in the manipulated variables can be controlled purely by how aggressive we make the move plan. By using low move suppression on the MVs and high penalization of deviation from a CV trajectory, we can induce the MVs to overshoot significantly. Before a transition, the tuning of the controller is changed by the transition management software to implement this and once the transition is over, the tuning is set back to more conservative values for in-grade control.

4.4 Controller Structure for High Pressure Tube Process

The controller structure for a high pressure tube process will typically incorporate a lower tier chain transfer agent (CTA) concentration controller and an upper tier quality controller that cascades both to the concentration controller and to the peak temperature control loops on the tube. Since the height of the peaks has an impact on the product quality the valley temperatures are utilized as disturbances to the quality controller. The controlled variables in the quality controller would typically be MFI and Gloss.

4.5 Transition Management

Although a detailed description is beyond the scope of this paper, transition management is an important part of the total solution. The transition from one product grade to another may range from simple (for example changing the set-point on a single quality variable) to complex (for example changing from production of homopolymer to copolymer). Transition management consists of managing the Recipe data that defines a particular product grade, managing the sequence of steps that need to be taken before and after a transition occur, and managing any open loop processing that is required during the transition. The non-linear controller implements the actual transition. In order to implement an overshoot strategy, the transition manager may be configured to download more aggressive tuning during the transition stage.

5 Model Building and Simulation

Model building for polymer MPC projects is significantly different from traditional MPC projects. In addition, requirements for control models are different than those for process design or simulation, for example. It is important to build an understandable model - as simple as possible to make future maintenance as easy as possible - yet with accurate and reliable gains. Some of the requirements for a model to be suitable for control are:

1. A flexible modelling strategy that takes into account process specific knowledge. Model replication from one site to another is rarely successful. One size does not fit all!

2. Safe, intelligent and maintainable extrapolation and interpolation - i.e. guaranteed gain behaviour at all operating points. Specifically, gain sign inversion has to be prevented unless it is genuinely present.
3. Support non-linear dynamics, e.g. different up and down responses, or rate of change dependent dynamics.
4. Minimal data requirements - it is difficult to step test a polymer plant when the unit is not allowed to produce off-specification product.
5. Support multivariate non-linearities (e.g. multiple catalysts / donors), i.e., model structures where the true process characteristic is represented by a curved surface in N-dimensional space. This implies that the current gain value depends on more than one input variable. Models with these characteristics cannot be linearized by using simple input or output linearization curves.
6. Fast reliable execution.
7. Ability to fine-tune the model while commissioning the controller.

Some of the details of the modelling now follow

1. Quality models for polymer reactors are characterized by low order dynamics, but strong non-linearities. Extensive experience shows that these can be very accurately modelled with a low order linear dynamic model feeding an empirically identified gain-constrained BDN (Bounded Derivative Network) model. These steady state non-linear BDN models can be calibrated on normal historical operating data, or sometimes, even recipe data, thus precluding the need for a plant test in most cases. Figure 4 shows the structure of a BDN and the types of surface that it synthesizes. These surfaces are shown as curves, but more generally they can be thought of as multi-dimensional sheets with a single bend. The superscripts in the equations represent layer indexing in a network-like structure. The natural interpolation and extrapolation achieved by synthesizing these surfaces precludes the need for large amounts of data. This contrasts with a traditional neural network (for example a Multi-layer Perceptron) where the natural tendency is for gain information to interpolate and extrapolate to 0 - an ill-suited and unsafe characteristic for models embedded in a control law. In addition, analytic constraints can be put on the calibration of these models so that global gain guarantees are satisfied, thus allowing process knowledge (potentially derived from rigorous chemical engineering models) to be imposed on these empirical structures. Figure 5 shows this calibration procedure. Calibrating a BDN model utilizes a back propagation algorithm (a general form of chain rule) to efficiently calculate analytic derivatives. This was originally developed for the field of neural networks in order to provide the analytic derivatives for simple delta-rule iterative learning algorithms in Multi-Layer Perceptron structures. For BDNs, it provides the analytic derivatives for the general constrained non-linear solver.
2. Concentration ratio models map flows to concentration ratios of the primary flows into the reactor. Feedback for these variables typically is available from fast online analyzers. These models are non-linear (due to the ratio

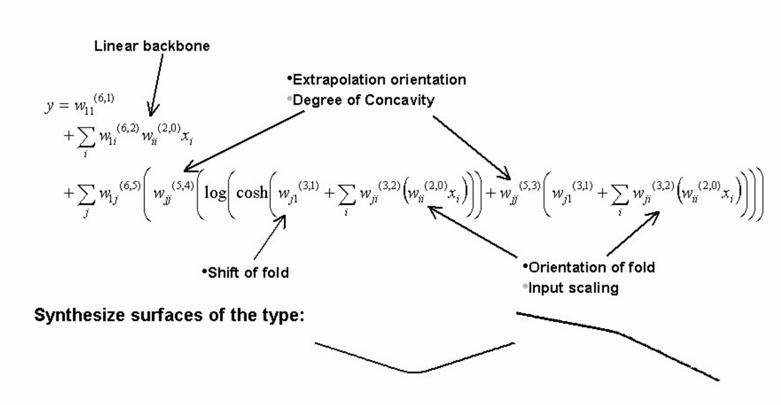

Linear backbone
•Extrapolation orientation
•Degree of Concavity
$y = w_{11}^{(6,1)} + \sum_i w_{1i}^{(6,2)} w_{ii}^{(2,0)} x_i + \sum_j w_{1j}^{(6,5)} \left(w_{jj}^{(5,4)} \left(\log\left(\cosh\left(w_{j1}^{(3,1)} + \sum_i w_{ji}^{(3,2)} \left(w_{ii}^{(2,0)} x_i \right) \right) \right) + w_{jj}^{(5,3)} \left(w_{j1}^{(3,1)} + \sum_i w_{ji}^{(3,2)} \left(w_{ii}^{(2,0)} x_i \right) \right) \right) \right)$
•Shift of fold
•Orientation of fold
•Input scaling
Synthesize surfaces of the type:

Fig. 4. Bounded Derivative Network

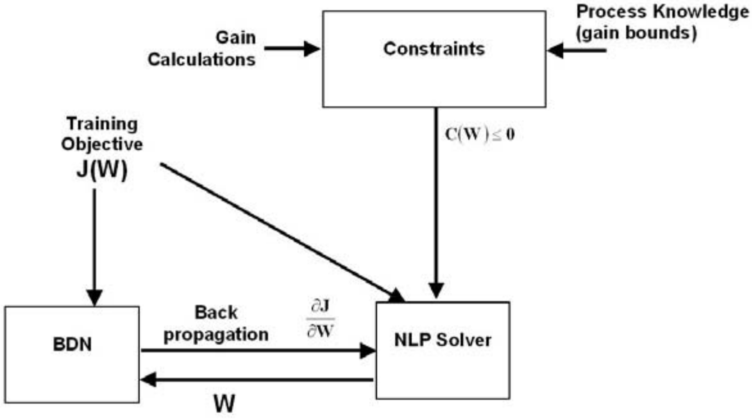


Fig. 5. Calibrating a BDN

calculation) but also may have very different dynamic behaviour depending on the vent valve position. The software provides a way of utilizing non-linear difference equations to provide these models, which also only need minimal data for calibration.

3. Quality models have different behaviours for different catalysts, but from the non-linear controller's perspective, they should be pieced together into one model in a way that allows continuous transition control from one catalyst to another.
4. Integrating models (also called 'Ramp' models) are needed for modelling pressure, partial pressure, and level variables. These are typically very simple, well-understood linear models with a single integrating state and possibly additional stable dynamics. Model-plant mismatch is modelled by assuming a stochastic disturbance on the integrating state.

The control engineer will put together an integrated simulation for validating controller structure and models, and for setting up much of the controller tuning. This simulation will also incorporate any transition management.

Fine-tuning of the models and recalibration goes on during a commissioning phase.

Example: Quality Models for High Pressure Tube Process

The high-pressure tube process provides an interesting aspect to the empirical modelling challenge. The dynamics of the process response to temperature changes are typically second order plus dead time at a given operating point. In addition to this the temperature peaks are highly correlated with each other which requires that the modelling architecture enforces this correlation at all times. This is achieved by feeding the temperature peaks into the same two states representing a second order dynamic system, where process data is used to calibrate the contributions of each temperature peak to these states. These two states then get combined into a single state (i.e. representing the output of second-order dynamic system), which is fed along with other states (representing the dynamic states of other inputs) into a Wiener structure utilizing a Bounded Derivative Network. The advantages of this approach are that (a) the temperatures are guaranteed to have a correlated effect within the model, and (b) the nonlinear model is forced to accept the second order dynamic rather than having to deal with the two individual states that make up the second order response. This guarantees a reliable second order response at all operating points.

6 Controller Tuning

6.1 Estimation

Plant-model mismatch is handled by assuming a stochastic disturbance entering each of the non-linear MISO models that make up the total controller model. This augments each model with an additional disturbance state.

For stable CVs this disturbance state acts as a bias on the output of the model. For ramp CVs this disturbance state partly enters the core integrating states and partly acts as a bias on the output as apportioned by a 'rotation factor'. A Kalman Filter is used to estimate the stochastic states online.

The rotation factor determines how quickly the model adapts to error in ramp rate. One practical issue is that changing the rotation factor online (which is essential during commissioning) changes the augmented model, and without additional logic, will cause an artificial disturbance. It is important that logic is in place to effect a bumpless change of this tuning parameter.

6.2 Optimization

Business requirements drive the controller objectives and therefore the controller design. Such requirements may rank some objectives as much more important than others. For example:

1. Maintain process safety ...
2. ... while satisfying the product quality objectives ...
3. ... at the required production rate ...
4. ... and minimizing energy consumption.

The controller uses two different schemes for this multi-objective optimization: lexicographic ordering of constraints, and weighting of constraints. Lexicographic ordering implies that one constraint can be made infinitely more important than a lower "ranked" constraint. Several constraints can also be assigned to the same rank, and weighting is then used for constraints at the same priority (rank).

Lexicographic ordering is achieved by solving a series of nonlinear optimization (or feasibility) problems. This can be written in general form as:

$$\begin{aligned} \min \quad & r_k^T W r_k \\ s.t. \quad & g(x) - r_k \leq d \\ & h(x) \leq e \end{aligned}$$

where W is a weighting matrix, $g(x)$ are the constraints at rank k (soft constraints) k is the current rank, and $h(x)$ are the constraints at ranks higher than k (hard constraints). Let $r_k(opt)$ be the solution to rank k. Then the next rank (k+1) can be written as:

$$\begin{aligned} \min \quad & r_{k+1}^T W r_{k+1} \\ s.t. \quad & g_k(x) \leq d + r_k(opt) \\ & g_{k+1}(x) - r_{k+1} \leq d_k \\ & h(x) \leq e \end{aligned}$$

Note that the soft constraints from rank k have been promoted to hard constraints.

This resulting scheme has been used for a number of years in the industrial community and has recently been attracting academic interest [1].

A full optimization problem must be solved at each rank. A sequential quadratic programming approach is used. Trust regions are placed on the inputs. A primal-dual interior point algorithm is used to solve the QP sub-problems [8]. Artificial variables are added to the QP sub-problems to ensure feasibility.

Usually 3 to 5 QP sub-problem iterations are sufficient. Although there is no proof of convergence to a local optimum, the cyclical nature of the controller ensures that a local optimum will eventually be reached. Options are available within the controller to activate other algorithms with formal convergence proofs, but these algorithms do not appear to be as consistent.

The limits, rankings, and economic values are the main tuning values for the steady state optimization.

6.3 Ranking for High Pressure Tube example

In the high-pressure tube example the highest priority objective is to maintain the melt index and gloss within their limits. However a secondary objective is to maximize conversion. It is extremely important that this secondary objective is not weighted against the melt index and gloss targets as the conversion optimization is only important when the gloss and melt index are within their limits. Explicit ranking guarantees this. Without ranking it would be possible that under certain circumstances that the controller may calculate that slightly violating limits was less costly than maximizing peak temperatures and this would be unacceptable.

6.4 Move Plan

There are severe requirements for the MV move plan. It needs to be executing reliably at least once a minute, for some applications less than a minute. Typical controller sizes vary between 5 × 5 and 30 × 30 for each tier of the cascaded structure, with a control horizon that may be several hours, particularly on gas phase processes. Typically all the controllers in a cascaded architecture will run on the same box. A feasible path algorithm is necessary so that sub-optimal solutions can be used in the (very unlikely) event that cycle time is insufficient to converge. In practice we typically see times of a few seconds for solving these types of problem during transitions, much less for in-grade operation. The practical means to solve these problems include intelligent blocking of moves and coincident points along the CV horizons. The general form of the move plan objective is:

$$\Phi = \Phi^{CVTrajectoryPenalty} + \Phi^{MVTrajectoryPenalty} + \Phi^{CVConstraintHandling} \\ + \Phi^{MVMoveSuppresion}$$

The first two terms penalize deviation from the CV and MV trajectories, respectively, that join the current point to the calculated steady state target. These trajectories can be tuned online and provide a handle for controlling the aggressiveness of the controller on a per-CV and per-MV basis.

One of the important practical requirements for doing polymer transitions, especially in the case of a UNIPOL gas phase process for which the qualities have very slow dynamics, is for the control law to support overshoot strategies as described earlier in this paper. This type of strategy is implemented by aggressive CV trajectories, high CV trajectory penalties, low MV trajectory penalties, and low MV move suppression.

CV constraints are handled by means of L1 penalty treatment as described in [3] and [7]. In L1 penalty treatment, as opposed to L2 penalty treatment, constraints are exactly satisfied if there is a feasible solution, and the constrained system has stability characteristics identical to the corresponding unconstrained case.

The objective is optimized using a Multi-step Newton-type algorithm as described in [2] and [4].

Another practical issue occurs when the calculated steady state solution lies outside the operating limits - particularly in the presence of large disturbances. In these situations it is undesirable for the controller to aggressively pursue this infeasible steady state solution and the control law must be aware of these types of situation and act appropriately.

7 Commissioning

Commissioning is usually phased over several visits to a site. The aim of the first visit is to put the inferential quality models online so they can be observed over several operating conditions to ensure accuracy and models can be improved if necessary.

Follow up visits will involve commissioning of the controllers. Some practical issues:

1. Slave (gas composition) controllers are commissioned prior to master (quality) controllers.
2. Controllers are commissioned incrementally adding one or two handles at a time - the software must support this incremental workflow.
3. Controllers are initially commissioned for in-grade control. Only when this is satisfactory are transitions commissioned.
4. Windows of opportunity for a given grade may be very short - sometimes a window of only a day or two within a period of several weeks. This has several implications:
   - Commissioning must be well planned and well prepared to make the most of these opportunities.
   - Models must allow for fine tuning online so as to make the most of the short time window
5. Valve properties may require some practical approaches
   - Accumulate several moves before implementing them
   - When writing valve outputs directly, it is better to explicitly separate out valve non-linearities from the model structure
6. Identify disturbances at the source and address them early - robust control is still necessary, but not sufficient
   - Examples of disturbances in a UNIPOL PP process:
     - Cooling water inlet temperature variance.
     - Catalyst batch changes
     - Day/night feed quality changes
     - Bed instability (monitor bed levels, bed weight)
     - Powder discharge system problems
   - For example, one might check on the feasibility of including cooling water inlet temperature control to minimize ambient effects.

8 Operating the Solution

Both operators and plant engineers have their own interface to the online solution. The operator interacts with the system through a simplified interface which allows monitoring of key quality and composition variables, allows changing of operating limits, in-grade set-point changes, validation of lab results, triggering of transition sequences, and the ability to turn the controller off and change to manual control.

The engineering interface is much richer, allowing all the operator interactions, as well as allowing changes to controller tuning. Both operator and engineering interfaces are required to have role-based security to avoid unauthorized personnel making changes that affect the plant.

9 Benefits

Demonstrated project results show:

1. Capacity Increase: $2 - 10\%$
2. Off-spec reduction during transition: $25 - 50\%$
3. Off-spec reduction during steady state operation: $50 - 100\%$

The direct quantitative benefits are significant and are typically in the region of $\$400,000$ to $\$1,000,000$ per line per year. Qualitative benefits include:

1. Minimizing product transition times
2. Minimizing variability in quality
3. Maximizing production capacity
4. Reducing raw material consumptions
5. Reducing downtime and maintenance cost
6. Reducing safety stocks and slow-moving inventory

10 Conclusions

This paper has examined some of the technological and practical issues faced in implementing nonlinear control on industrial continuous polymer manufacturing processes. Descriptions of modelling technology and controller technology have been given with emphasis on practical solutions. The importance of suitable models has been emphasized, and it is surely a fruitful area of research for the academic community to find more and better ways to impose process and fundamental knowledge onto simple control-suitable models structures – to bridge the gap between complex continuous-time rigorous models and simple discrete-time models with well-understood gain characteristics. In addition, it would be very useful to develop control theoretic results on stability, observability, controllability, problem convexity, etc. within the context of these reduced scope models.

References

[1] Eric C. Kerrigan and Jon M. Maciejowski, “Designing model predictive controllers with prioritised constraints and objectives.”, *Proceeding of the IEEE Conference on CACSD, Glasgow*, pages 309-315, (2002).

[2] W.C. Li and L.T. Biegler, “Multistep, Newton-type control strategies for constrained nonlinear processes.”, *Chemical Engineering Research Design*, pages 562-577, (1989).

[3] N.M.C. Oliveira and L.T. Biegler, "Constraint handling and stability properties of model-predictive control.", *AIChe Journal*, pages 1138-1155, (1994).

[4] N.M.C. Oliveira and L.T. Biegler, "An extension of Newton-type algorithms for nonlinear process control", *Automatica*, pages 281-286 (1995).

[5] S. Piche., B. Sayyar-Rodsari, D. Johnson, and M. Gerules, " Nonlinear model predictive control using neural networks", *IEEE Control Systems Magazine*, pages 53-62, (2000).

[6] P. Turner, J. Guiver, "Introducing the bounded derivative network - superceding the application of neural networks in control", *Journal of Process Control*, pages 407-415 (1997).

[7] H. Zhao, J. Guiver , R. Neelakantan and L. Biegler, "Nonlinear Industrial Model Predictive Controller Using Integrated PLS and Neural Net State Space Model", *IFAC World Congress, Beijing, China*, (1999).

[8] S.J. Wright, "Primal-Dual Interior-Point Methods.", *SIAM*, (1997).

Integration of Advanced Model Based Control with Industrial IT

Rüdiger Franke[1] and Jens Doppelhamer[2]

[1] ABB AG, Power Generation, Mannheim, Germany
Ruediger.Franke@de.abb.com
[2] ABB Corporate Research, Ladenburg, Germany
Jens.Doppelhamer@de.abb.com

Summary. Advanced model based control is a promising technology that can improve the productivity of industrial processes. In order to find its way into regular applications, advanced control must be integrated with the industrial control systems. Modern control systems, on the other hand, need to extend the reach of traditional automation systems – beyond control of the process – to also cover the increasing amount of information technology (IT) required to successfully operate industrial processes in today's business markets. The Industrial IT System 800xA from ABB provides a scalable solution that spans and integrates loop, unit, area, plant, and interplant controls.

This paper introduces the 800xA and the underlying Aspect Object technology. It is shown how model knowledge and optimization solver technology are integrated into the 800xA framework. This way, advanced model based control solutions can be set up in an open and modularly structured way. New model and solver aspects can be combined with available aspects covering standard functionality like process connectivity, management of process data, trend&history data and application data, as well as operator graphics.

A Nonlinear Model-based Predictive Controller (NMPC) for power plant start-up is treated as example. This paper discusses how NMPC can be integrated with a modern control system so that standard concepts are re-used for this advanced model based control concept.

1 Introduction

During the last decades, several advanced control technologies have been developed, including adaptive control, fuzzy control and neuro control. While each of these technologies offers advantages over classical control methods, PID controllers still dominate the vast majority of industrial applications.

One reason for the lack of mainstream use of advanced control technologies is seen in the fact that they require specialized engineering knowledge and tools. Normally, specialized experts are required to apply advanced control methods. A better integration of advanced control technologies with regular control systems is seen as a key factor for improved acceptance.

Nonlinear model based control (NMPC) has received much attention during the last years. The technology has several advantages from a control point of view: it accommodates nonlinear, multi-variable problems with state constraints.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 399–406, 2007.
springerlink.com

Important achievements have been made to treat the computationally challenging task of formulating and solving large-scale nonlinear optimization problems on-line [2, 4, 6]. Moreover, NMPC has the advantage that the technology is more open, compared to other advanced control methods. Models do represent the behavior of a plant and standard optimization algorithms are used to apply the models to control. This openness improves the acceptance of NMPC on the one hand side.

On the other side, still special purpose tools are required to implement model based control. This implies that concepts which are readily available in a standard control system need to be specifically interfaced or even redeveloped for applications of model based control, including e.g. signal exchange with sensors, actuators and low level controls, operator graphics, trend&history display, signaling of alarms and events, as well as system maintenance. This is seen as an important burden for both: acceptance and cost of NMPC.

2 The Industrial IT System 800xA

2.1 System Overview

The Industrial IT System 800xA seamlessly integrates traditionally isolated plant devices and systems, extending the reach of the automation system to all plant areas. The result is a simplified, software representation of the plant, from simple on/off-type switches and valves to smart field devices, dedicated control subsystems, and PC-based supervisory systems [1].

The framework for the 800xA system architecture is built upon ABB's Aspect Object technology. Aspect Objects relate plant data and functions – the aspects, to specific plant assets – the objects. Aspect objects represent real objects, such as process units, devices and controllers. Aspects are informational items, such as I/O definitions, engineering drawings, process graphics, reports and trends that are assigned to the objects in the system.

Aspect Objects are organized in hierarchical structures that represent different views of the plant. One object may be placed multiple times in different structures. Examples for different types of structures are:

Functional Structure: Shows the plant from the process point of view.

Location Structure: Shows the physical layout of what equipment is located where in the plant.

Control Structure: Shows the control network in terms of networks, nodes, fieldbuses, and stations.

The idea of placing the same object in multiple structures is based on the IEC standard 1346 [3, 9].

The Plant Explorer is the main tool used to create, delete, and organize Aspect Objects and aspects. It is based on a structural hierarchy, similar to Windows Explorer, as demonstrated in Figure 1. The object hierarchy is visible on the left hand side of the window. The upper right pane shows the aspects of an object and the lower right pane views a selected aspect.

2.2 Integration of Model Based Control

A new Model aspect has been developed so that mathematical model information can be added to an Aspect Object. The model has the form of a hybrid differential algebraic equation system (hybrid DAE)

$$\begin{aligned}
\mathbf{0} &= \mathbf{F}[\mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{m}(t), \mathbf{u}(t), \mathbf{z}(t), \mathbf{y}(t), \mathbf{p}, t], && (1)\\
&\quad \mathbf{F} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_m} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_p} \times \mathbb{R}^1 \mapsto \mathbb{R}^{n_x} \\
\mathbf{m}(t) &:= \mathbf{G}[\mathbf{x}(t), \mathbf{m}(t), \mathbf{u}(t), \mathbf{z}(t), \mathbf{y}(t), \mathbf{p}, t], && (2)\\
&\quad \mathbf{G} : \mathbb{R}^{n_x} \times \mathbb{R}^{n_m} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_z} \times \mathbb{R}^{n_y} \times \mathbb{R}^{n_p} \times \mathbb{R}^1 \mapsto \mathbb{R}^{n_m}.
\end{aligned}$$

Here $\mathbf{x}$ denote continuous-time states, $\mathbf{m}$ are discrete modes, $\mathbf{u}$ and $\mathbf{z}$ are controlled and not-controlled inputs, respectively, $\mathbf{y}$ are outputs and $\mathbf{p}$ are model parameters. Discrete modes are variables that change their values only at discrete time instants, so called event instants t_e. See [10] for more information on the treated hybrid DAE.

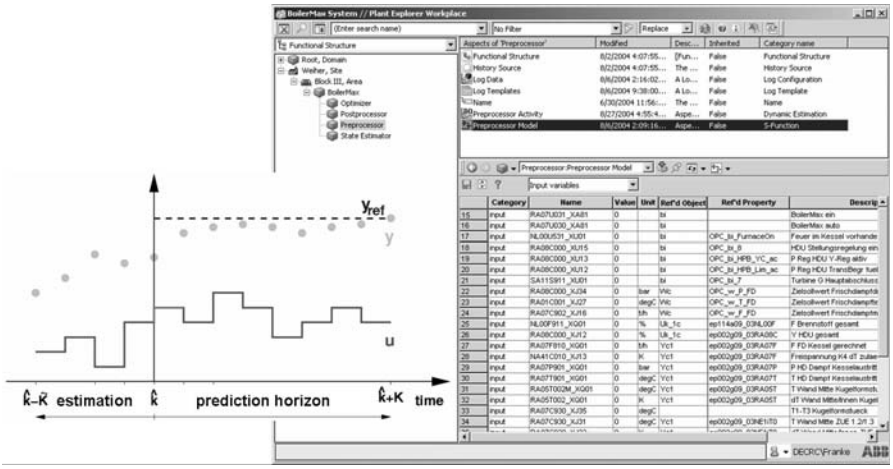
The Model aspect holds information related to the model, including

- Declaration of model variables in categories (Parameter, Input, Output, State, Generic),
- Values for model variables, e.g. for parameters,
- References to process signals, e.g. for inputs and outputs,
- Structural information for hierarchical sub-model structure,
- Reference to the implementation of the model.

The Model aspect does not provide any functionality nor does it deal with implementation details. Instead it references an external implementation. In this way available modeling tools can be applied and expensive re-implementation is avoided.

A model can be used to perform one or more model-based activities. A second aspect, the Dynamic Optimization aspect has been developed to interface a numerical solver, hold the solver configuration, and to exchange data between the solver and the control system. The exchanged data includes: configuration data, current process values (like sensor values and controller set-points), and history logs. Predictions are written back to the control system as history logs with future time stamps.

The integrated solver HQP is primarily intended for structured, large-scale nonlinear optimization [7]. It implements a Sequential Quadratic Programming algorithm that treats nonlinear optimization problems with a sequence of linear-quadratic sub-problems. The sub-problems are formed internally by simulating the model and by analyzing sensitivities. They are solved with an interior point method that is especially suited for a high number of inequality constraints, e.g. resulting from the discretization of path constraints. See [6], [8], and [7] for more details about the solver.



Aspects of 'Preprocessor'	Modified	Desc...	Inherited	Category name
Functional Structure	8/2/2004 4:07:55...	[Fun...	False	Functional Structure
History Source	8/2/2004 4:07:55...	The ...	False	History Source
Log Data	8/6/2004 2:16:02...	A Lo...	False	Log Configuration
Log Templates	8/6/2004 9:38:00...	A Lo...	False	Log Template
Name	6/30/2004 11:56:...	The ...	False	Name
Preprocessor Activity	8/27/2004 4:55:4...	Aspe...	False	Dynamic Estimation
Preprocessor Model	8/6/2004 2:09:16...	Aspe...	False	S-Function

	Category	Name	Value	Unit	Ref'd Object	Ref'd Property	Descrip
15	input	RA07U031_XA81	0		bi		BoilerMax ein
16	input	RA07U030_XA81	0		bi		BoilerMax auto
17	input	NL00U531_XU01	0		bi	OPC_bi_FurnaceOn	Feuer im Kessel vorhande
18	input	RA08C000_XU15	0		bi	OPC_bi_8	HDU Stellungsregelung ein
19	input	RA08C000_XU13	0		bi	OPC_bi_HPB_YC_ac	P Reg HDU Y-Reg aktiv
20	input	RA08C000_XU12	0		bi	OPC_bi_HPB_Lim_ac	P Reg HDU TransBegr fuel
21	input	SA11S911_XU01	0		bi	OPC_bi_7	Turbine 0 Hauptabschluss
22	input	RA08C000_XJ34	0	bar	Wc	OPC_w_P_FD	Zielsollwert Frischdampfd
23	input	RA01C001_XJ27	0	degC	Wc	OPC_w_T_FD	Zielsollwert Frischdampfte
24	input	RA07C902_XJ16	0	t/h	Wc	OPC_w_F_FD	Zielsollwert Frischdampfm
25	input	NL00F911_XQ01	0	%	Uk_1c	ep114a09_03NL00F	F Brennstoff gesamt
26	input	RA08C000_XJ12	0	%	Uk_1c	ep002g09_03RA08C	Y HDU gesamt
27	input	RA07F810_XQ01	0	t/h	Yc1	ep002g09_03RA07F	F FD Kessel gerechnet
28	input	NA41C010_XJ13	0	K	Yc1	ep002g09_03RA07F	Freispannung K4 dT zulae
29	input	RA07P901_XQ01	0	bar	Yc1	ep002g09_03RA07P	P HD Dampf Kesselaustritt
30	input	RA07T901_XQ01	0	degC	Yc1	ep002g09_03RA07T	T HD Dampf Kesselaustritt
31	input	RA05T002M_XQ01	0	degC	Yc1	ep002g09_03RA05T	T Wand Mitte Kugelformst.
32	input	RA05T002_XQ01	0	K	Yc1	ep002g09_03RA05T	dT Wand Mitte/innen Kugel
33	input	RA07C930_XJ35	0	degC			T1-T3 Kugelformstueck
34	input	RA07C930_XJ31	0	degC	Yc1	ep002g09_03NE1/T0	T Wand Mitte ZUE 1.2/1.3

Fig. 1. Plant Explorer showing the Functional Structure of NMPC for boiler start-up (BoilerMax)

The treated model based activities include

- Initial value simulation for specified initial states $\mathbf{x}(t_0)$ and model inputs,
- Estimation of model parameters and initial states,
- Nonlinear optimal control with constraints on model inputs and outputs,
- Steady-state simulation, estimation and optimization at one time instant.

An initial-value simulation covers hybrid DAEs (1),(2). However, optimization and estimation problems can currently only be solved for a simplified hybrid DAE $\mathbf{F}$, $\mathbf{G}'$ of the form:

$$\begin{aligned} \mathbf{m}(t) &:= \mathbf{G}'[\mathbf{m}(t), \mathbf{z}(t), t], \\ \mathbf{G}' &: \mathbb{R}^{n_m} \times \mathbb{R}^{n_z} \times \mathbb{R}^1 \mapsto \mathbb{R}^{n_m}, \end{aligned} \tag{3}$$

where discrete modes do not depend on states or optimized variables.

Figure 1 shows how the functional structure is set up for an NMPC using Aspect Object technology. Different Aspect Objects represent the major processing activities of the NMPC algorithm.

- The Preprocessor reads current measurements from the underlying control system, validates the data and generates a guess for the model state. Furthermore a short term history is assembled.
- The State Estimator estimates the initial states based on the short-term history collected by the Preprocessor.
- The Optimizer predicts the optimal control into the future, starting from the estimated initial state

- The Postprocessor checks optimization results and communicates set points to the underlying control system.
- The Scheduler periodically triggers the other activities and supervises their successful completion.

The object-oriented, physical modeling technology Modelica is used to build the models [10]. A physical plant model is built on available model libraries [5]. It is used by both: state estimator and optimizer. Moreover, specific preprocessor and the postprocessor models are formulated as computational algorithms in Modelica. The scheduler model is formulated as state graph [12].

Based on the models, the activities are formulated as estimation (State Estimator), optimization (Optimizer) or initial-value simulation (Preprocessor, Postprocessor, Scheduler).

3 Application Example

A Nonlinear Model-based Predictive Controller (NMPC) for power plant start-up serves as example. The start-up problem is challenging as it is highly nonlinear in the covered large range of operation. Thermal stress occurring in thick walled components needs to be kept in given limits. Multiple manipulated variables must be coordinated. A long prediction horizon is required to fulfill the constraints during a start-up.

Figure 2 shows a process diagram of a power plant. Feed water goes through pre-heaters and the economizer into the evaporator, as seen in the lower left

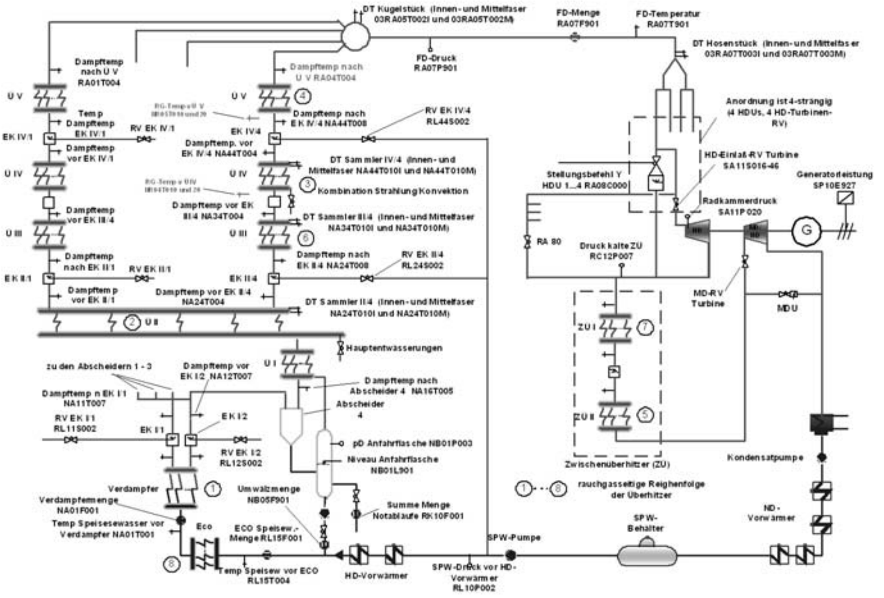
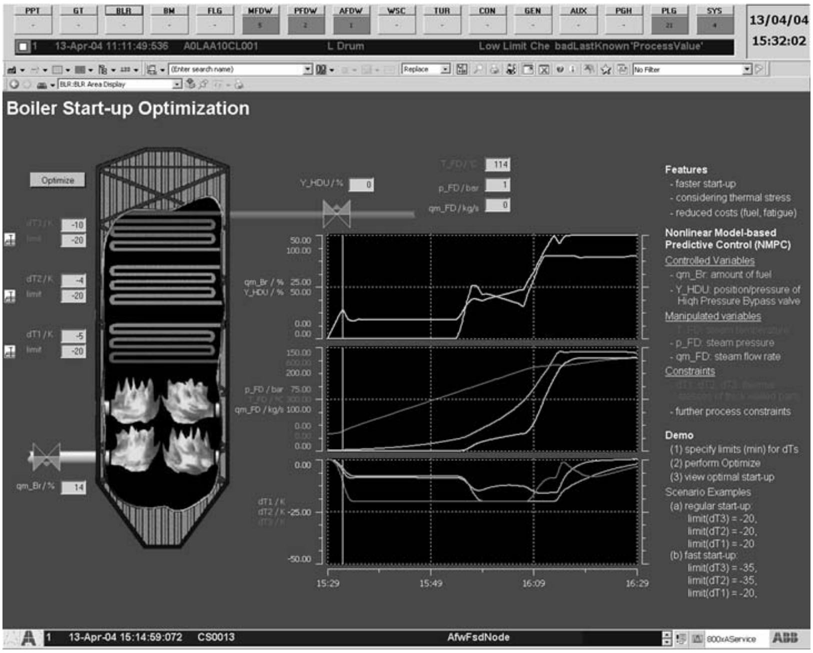

DT Kugelstück (Innen- und Mittelfaser 03RA05T002I und 03RA05T002M)
FD-Menge RA07F901
FD-Temperatur RA07T901
DT Hosenstück (Innen- und Mittelfaser 03RA07T003I und 03RA07T003M)
FD-Druck RA07P901
Dampftemp nach U V RA01T004
Dampftemp nach U V RA04T004
U V
U V
4
Temp Dampftemp EK IV/1
EK IV/1
RV EK IV/1
EK IV/4
Dampftemp nach EK IV/4 NA44T008
RV EK IV/4 RL44S002
Dampftemp vor EK IV/1
Dampftemp. vor EK IV/4 NA44T004
DT Sammler IV/4 (Innen- und Mittelfaser NA44T010I und NA44T010M)
U IV
U IV
3
Kombination Strahlung Konvektion
Dampftemp vor EK III/4
Dampftemp vor EK III/4 NA34T004
DT Sammler III/4 (Innen- und Mittelfaser NA34T010I und NA34T010M)
U III
U III
6
Dampftemp nach EK II/1
RV EK II/1
EK II/1
EK II/4
Dampftemp nach EK II/4 NA24T008
RV EK II/4 RL24S002
Dampftemp vor EK II/1
Dampftemp vor EK II/4 NA24T004
DT Sammler II/4 (Innen- und Mittelfaser NA24T010I und NA24T010M)
2
U II
Hauptentwässerungen
U I
zu den Abscheidern 1 - 3
Dampftemp vor EK I/2 NA12T007
Dampftemp n EK I/1 NA11T007
Dampftemp nach Abscheider 4 NA16T005
Abscheider 4
RV EK I/1 RL11S002
EK I/1
EK I/2
RV EK I/2 RL12S002
pD Anfahrflasche NB01P003
Niveau Anfahrflasche NB01L901
Verdampfer
1
Verdampfermenge NA01F001
Temp Speisesewasser vor Verdampfer NA01T001
Umwälzmenge NB05F901
Summe Menge Notablaufe RK10F001
Eco
ECO Speisew.-Menge RL15F001
8
Temp Speisew vor ECO RL15T004
HD-Vorwärmer
SPW-Pumpe
SPW-Druck vor HD-Vorwärmer RL10P002
Anordnung ist 4-strängig (4 HDUs, 4 HD-Turbinen-RV)
HD-Einlaß-RV Turbine SA11S016-46
Stellungsbefehl Y HDU 1...4 RA08C000
Generatorleistung SP10E927
Radkammerdruck SA11P020
G
RA 80
Druck kalte ZU RC12P007
MD-RV Turbine
MDU
ZU I
7
ZU II
5
Zwischenüberhitzer (ZU)
Kondensatpumpe
1 - - 8
rauchgasseitige Reighenfolge der Überhitzer
ND-Vorwärmer
SPW-Behälter

Fig. 2. Simplified process diagram of a power plant



Boiler Start-up Optimization

Features
- faster start-up
- considering thermal stress
- reduced costs (fuel, fatigue)

Nonlinear Model-based Predictive Control (NMPC)

Controlled Variables
- qm_Br: amount of fuel
- Y_HDU: position/pressure of High Pressure Bypass valve

Manipulated variables
- T_FD: steam temperature
- p_FD: steam pressure
- qm_FD: steam flow rate

Constraints
- dT1, dT2, dT3: thermal stresses of thick walled parts
- further process constraints

Demo
(1) specify limits (min) for dTs
(2) perform Optimize
(3) view optimal start-up

Scenario Examples
(a) regular start-up:
limit(dT3) = -20,
limit(dT2) = -20,
limit(dT1) = -20
(b) fast start-up:
limit(dT3) = -35,
limit(dT2) = -35,
limit(dT1) = -20,

Fig. 3. Operator display showing the optimal start-up predicted by the NMPC, in addition to current process values and history logs

section of the diagram. Saturated steam leaving the evaporator is super-heated in several super-heater stages. The example uses five super-heater stages and four parallel streams, as seen in the upper left section of the diagram. The live steam leaving the boiler goes to the turbine. The example uses two turbine sections. In the turbine, thermal energy is transformed to mechanical energy, driving the generator. Afterwards the steam is condensed and water flows back to the feed water tank, as seen in the lower right section of the diagram.

A boiler model was built using the Modelica technology [5]. The model needs to be carefully designed so that it expresses the relationship between optimized control actions (fuel flow rate and valve positions) and constrained process values (pressures, temperatures and thermal stresses). In the example described here, a system of differential-algebraic equations (DAE) with 940 variables was built, using measurements of about 150 process values. The Dynamic Optimization aspect system was used off-line to identify model parameters based on data logs available for historical start-ups.

During a run of the NMPC, an optimization problem is solved on-line every minute. The model is adapted to the process based on 36 on-line signals. 18 values are communicated back to the process, including three controller set points and additional signals for switch conditions and operator displays. The time horizon for prediction and control is 90 minutes in the example. It gets divided into 90 sample periods. The optimized manipulated variables are parameterized piecewise linear. All other model variables are evaluated at the sample time points. This means that overall 85540 variables are present in the on-line optimization problem. The solution time is about five minutes for a cold start of the solver and about 40 seconds for a subsequent solver run. Please see [8] for details about the numerical formulation and solution of the optimization problem.

Figure 3 shows an operator display for boiler start-up optimization. The trend plot displays the manipulated variables in the upper pane, the main process variables (live steam parameters) in the middle pane and constrained thermal stresses in the lower pane. As a result of the optimization, the process is driven along the allowed limits for thermal stresses.

Traditionally an operator display shows current process values and history logs. As a by-product of model predictive control, the operator can additionally see the prediction of the future behavior of the plant. As the NMPC runs integrated with the control system, this display can easily be configured.

Using the NMPC, the start-up time could be reduced by about 20 minutes and the start-up costs by 10% as compared to a well tuned classical control.

4 Conclusions

Nonlinear Model-based Predictive Control (NMPC) is a promising control technology. Due to advances in computational algorithms during recent years, it is now possible formulate and solve the underlying large-scale nonlinear optimization problems on-line under real-time conditions. The example discussed here was developed in detail in [8].

For a successful application of NMPC it is equally important to appropriately integrate the method with the control system. This paper discusses how this is done with the Industrial IT System 800xA by ABB. Based on international standards for control systems engineering and software, the System 800xA architecture and the Aspect Object technology allow a flexible integration of model knowledge and model based applications. Two new aspects have been developed in the Dynamic Optimization system extension. The new aspects can be combined with other available aspects, e.g. for controller connectivity, history logs and process graphics.

The NMPC runs on an application server that is integrated as additional node with the system. Installation and maintenance are identical to other nodes, like data servers and display clients.

This paper uses the start-up of a power plant as example. Batch processes are another promising application area, as described in [11].

References

[1] ABB Automation Technologies. Industrial IT System 800xA – System Architecture Overview. http://www.abb.com, Document Id: 3BUS092080R0101, 2005.

[2] Lorenz T. Biegler. Efficient solution of dynamic optimization and NMPC problems. In Frank Allgöwer and Alex Zheng, editors, *Nonlinear Model Predictive Control*, pages 219–243. Birkhäuser, Basel, 2000.

[3] L.G. Bratthall, R. van der Geest, H. Hoffmann, E. Jellum, Z. Korendo, R. Martinez, M. Orkisz, C. Zeidler, and J. S Andersson. Integrating hundred's of products through one architecture – the Industrial IT architecture. In *International Convverence on Software Engineering.* Orlando, Florida, USA, 2002.

[4] M. Diehl, R. Findeisen, S. Schwarzkopf, I. Uslu, F. Allgöwer, H.G. Bock, and J.P. Schlöder. An efficient algorithm for optimization in nonlinear model predictive control of large-scale systems. *at – Automatisierungstechnik*, 50(12), 2002.

[5] H. Elmqvist, H. Tummescheit, and M. Otter. Modeling of thermo-fluid systems – Modelica.Media and Modelica.Fluid. In *Proceedings of the 3rd International Modelica Conference.* Modelica Association, Linköping, Sweden, November 2003.

[6] R. Franke and E. Arnold. Applying new numerical algorithms to the solution of discrete-time optimal control problems. In K. Warwick and M. Kárný, editors, *Computer-Intensive Methods in Control and Signal Processing: The Curse of Dimensionality*, pages 105–118. Birkhäuser Verlag, Basel, 1997.

[7] R. Franke, E. Arnold, and H. Linke. HQP: a solver for nonlinearly constrained large-scale optimization. http://hqp.sourceforge.net.

[8] R. Franke, K. Krüger, and M. Rode. Nonlinear model predictive control for optimized startup of steam boilers. In *GMA-Kongress 2003.* VDI-Verlag, Düsseldorf, 2003. VDI-Berichte Nr. 1756, ISBN 3-18-091756-3.

[9] International Electrotechnical Commission. Industrial systems, installations and equipment and industrial products – structuring principles and reference designations. IEC Standard 61346, 1996.

[10] Modelica Association. Modelica – A Unified Object-Oriented Language for Physical Systems Modeling, Version 2.2. http://www.modelica.org, 2005.

[11] Z.K. Nagy, B. Mahn, R. Franke, and F. Allgöwer. Nonlinear model predictive control of batch processes: an industrial case study. In *16th IFAC World Congress.* Prague, Czech Republic, July 2005.

[12] M. Otter, J. Årzén, and A. Schneider. StateGraph – a Modelica library for hierarchical state machines. In *Proceedings of the 4th International Modelica Conference.* Modelica Association, Hamburg-Harburg, Germany, March 2005.

Putting Nonlinear Model Predictive Control into Use

Bjarne A. Foss[1] and Tor S. Schei[2]

[1] Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway
Bjarne.Foss@itk.ntnu.no

[2] Cybernetica AS, Norway
Tor.S.Schei@cybernetica.no

Summary. We will in this paper highlight our experience with NMPC. In our context NMPC shall mean the use of a nonlinear mechanistic model, state estimation, and the solution of an online constrained nonlinear optimisation problem. Our reference base is a number of applications of NMPC in a variety of processes.

We discuss the use of mechanistic models in NMPC applications and in particular the merits and drawbacks of applying such models in online applications. Further, we focus on state estimation, and the use of Kalman filters and moving horizon estimation. Finally, we consider the design of the optimization problem itself and implementation issues.

1 Introduction

Nonlinear model predictive control (NMPC) opens for the use of MPC in more demanding applications than has normally been the case for linear MPC. In particular NMPC lends itself to nonlinear systems which exhibit large variations in operating conditions and which are critically dependent on the use of a dynamic nonlinear model to gain sufficient performance. A nice overview of NMPC can be found in [9].

NMPC is not a well defined term in the sense that NMPC may be used for controllers ranging from a slight variation of linear MPC to the online solution of a constrained nonlinear optimisation problem. One example of a slight modification to account for nonlinearities is the use of multiple linear models in such a way that the current working point defines which model should be active at a given time instant. Hence, the QP-problem frequently encountered in linear MPC will change as the active model changes. In our context NMPC shall mean the use of a nonlinear mechanistic model, state estimation, and the solution of an online constrained nonlinear optimisation problem.

The scope of this paper is to pinpoint critical issues when applying NMPC by drawing on our experience within the process industries since 2000. To ensure a sound level of credibility we first present our most important application areas and accompanying control challenges in some detail. Thereafter we address four

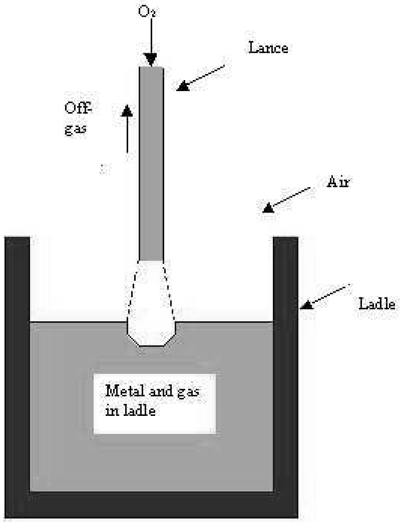
R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 407–417, 2007.
springerlink.com

critical areas: modelling, state estimation, formulation of the control problem itself and implementation issues. The paper ends with some conclusions.

2 Reference Base

The reference base contains a number of industrial applications of NMPC in a variety of processes. Some examples are:

- A system for optimization of suspension PVC polymerization processes has been implemented on two large (140 m3) autoclaves located at Hydro Polymers' plant in Porsgrunn, Norway. The system is characterized as follows: It contains a rather detailed nonlinear model of the polymerization reactor. The reactor model includes reaction kinetics, thermodynamic calculations for the four-phase suspension process, quality parameters and energy balances for the suspension and cooling systems. The application optimizes the temperature reference trajectory and the amount of initiators charged to the reactor in order to initiate the polymerization process. The optimization is based on an economic criterion which includes the batch time as well as cost of initiators. The process is highly exothermic, and the purpose of the optimization is to minimize the batch time without exceeding available cooling capacity and without using an inhibitor to slow down the polymerization process. The optimization is performed once for each batch cycle, and the temperature profile, which consists of approximately 80 "straight line segments", is optimised for the entire batch under a number of constraints imposed by the quality specifications of the various PVC products. Based on logged data a few model parameters are estimated as functions of conversion. The system is implemented using Cybernetica's NMPC and batch optimisation platform.
- Three NMPC applications for stabilization and quality control of the Borealis polypropolyne plant in Schwechat, Austria have been developed and implemented in cooperation with Borealis' personnel. The implementations are based on nonlinear first-principles models of the polyolefine plant (including three different polymerization reactors), and on Borealis in-house system for model predictive control (BorAPC). The system is characterized as follows: The three MPC applications comprise a multivariable control system with all together 11 control inputs and 19 controlled outputs. The nonlinear model consists of 77 states which are estimated on-line together with a few model parameters.
- NMPC of a base-catalyzed phenol-formaldehyde batch polymerization process has been implemented. The system is based on a rigorous model of the polymerization reactor. The system is implemented at Dynea's plant in Springfield, Oregon, USA. The model for this condensation polymerization process includes reaction kinetics, thermodynamics, population balances for functional groups and energy balance for the reactor and cooling system. Safety is an important issue and one driving force for implementing NMPC.



ladle1

Fig. 1. A sketch of the metal refining process

Another important consideration is reduced batch time. A second application on a twin reactor is presently being developed. The system is implemented using Cybernetica's NMPC platform.

Two additional applications will be presented in some more detail.

2.1 Manganese Metal Refining

This case is a metal refining process for removing carbon from manganese metal. The process is sketched in Figure 1. It consists of a ladle which is filled with liquid-phase high-carbon manganese metal. This implies that about 7% of the metal bath consists of carbon. In addition there is some iron and MnO in the metal bath. Carbon is removed by blowing O_2 into the ladle. The main overall reaction is

$$C + O_2 \rightarrow CO_2$$

The refining process produces different products with a carbon content in the range $0.5\% - 1.5\%$. Downstream the refining process the metal is casted, crushed and screened before it is packed and shipped to customers.

In addition to the main reaction there are intermediate reactions as well as side reactions. One important side reaction is evaporation of manganese metal.

$$Mn_{(l)} \rightarrow Mn_{(g)}$$

Fumes generated during the batch are collectected in an off-gas system and routed to a filter-system for removing dust.

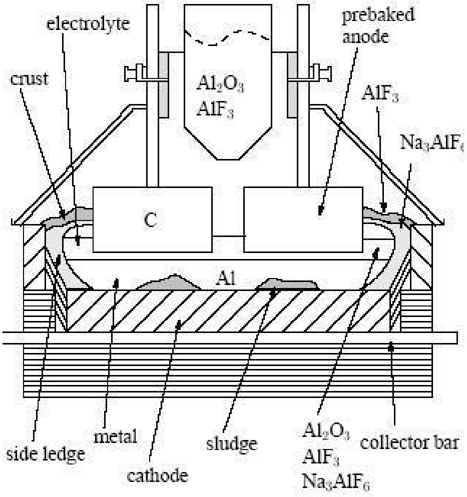


Fig. 2. Sketch of an aluminum electrolysis cell

The metal refining process is operated as a fed-batch process. The length of a batch sequence is in the order of $1 - 2$ hours. The same ladle is used from one batch to the next. It is replaced when the inner lining becomes too thin. The economic insentive for improved control is minimizing metal loss due to evaporation, while satisfying an upper limit on the concentration of carbon at the end of the batch. The control problem is challenging since this batch process is highly nonlinear and operational constraints are critical. An NMPC application was implemented in 2003 using Cybernetica's NMPC platform. More information on Mn decarburation can be found in [3].

2.2 Aluminum Electrolysis Cell

The Hall-Heroult process - a continuous process - is dominating worldwide in the production of aluminum [5]. The fundamentals of the process are to dissolve Al_2O_3 in molten cryolite, and electrically reduce complex aluminum containing ions to pure aluminum. The overall electro-chemical reaction in the electrolyte is

$$2Al_2O_3 + 3C \longrightarrow 4Al + 3CO_2$$

where carbon is fed to the reaction as consumable anodes. By the use of various additives, in particular AlF_3, the operating temperature of the electrolyte can be lowered from $1010C$ to approximately $960C$. Both decreased temperature and increased excess AlF_3 is believed to be beneficial for the current efficiency and the energy consumption. As molten cryolite is very corrosive, the only component of an acceptable cost presently capable of coexisting with it over time is frozen cryolite. It is therefore necessary to maintain a layer of frozen cryolite (side ledge) to prevent the carbon walls from eroding. In order to maintain the side ledge there has to be a substantial heat loss through the side ledge and the carbon walls of the cell.

The cell voltage applied is typically $4.5V$, and the electric current through the cell is typically $150 - 200kA$. A sketch of a cell is shown in Figure 2. In real life 100-200 cells are placed and connected in series.

There are three control inputs to the process, anode beam adjustments (controlling energy input), addition of AlF_3 and addition of Al_2O_3, and three controlled variables, bath temperature, concentration of AlF_3 and concentration of Al_2O_3. A cell is regularly perturbed since liquid aluminium is tapped and one of the anode blocks is changed on a daily basis. This induces severe disturbances in the energy balance, and it implies that the operating conditions will vary significantly and hence provoke nonlinear cell effects. The process has strong internal couplings, for instance between the mass and energy balance through the side ledge. Control of an aluminum electrolysis cell is a challenging problem [2], particularly as cell capacity increases beyond $200kA$. An NMPC application is presently under development.

3 Modeling

A common denominator for the applications referenced above is the use of a nonlinear first principles model. Empirical models have traditionally dominated the arena of MPC applications. A meaningful question would therefore be: “Why use mechanistic models?” The answer can be divided into two parts, and the first part can be found outside the model itself. In some of the applications a mechanistic model existed prior to the NMPC project initiation, and substantial resources had been used to develop and validate such a model. The fact that a model existed and considerable resources had been spent on development and validation are important reasons for extending the use of a model. Further, added use increases the odds for long-term survival of a model within a company. In the aluminum electrolysis case a model which had been developed over several years existed. This model had to be adjusted, only slightly however, to fit the NMPC application. No dynamic model existed prior to the NMPC project for the Mn metal refining reactor. The company, however, viewed the development of a mechanistic model in itself as important since such a model acts as a knowledge repository. One consequence of this was that the model development and validation phase was run as a joint activity between Cybernetica and the customer.

Second, in our experience a major advantage of first principles models is the reduced need for plant experimentation. As a matter of fact in the bulk of the above cases models have been developed purely on the basis of data from regular operation, ie. no dedicated measurement campaigns have been necessary. This implies that model structures have been selected, parameters have been estimated and models have been validated without resorting to often costly plant experiments. To further substantiate this, personnel from Borealis state “The model can be identified without doing plant experiments at all” [1] when discussing their proprietary NMPC technology and its application to polyolefine reactors. The need for reduced plant experimentation becomes particularly apparent when applying NMPC to several similar reactors as was the case for the suspension PVC polymerization reactors and the phenol-formaldehyde batch polymerization reactors referenced earlier.

Despite the above, the modeling process is a complex task and may be a bottleneck in the development of advanced computer-based applications, see eg. [4]. Without delving into this issue it should be noted that the process data used for offline tuning of parameters must, in some cases, include supplementary information to apply them in an appropriate manner. The reason for this is the fact that the process data does not necessarily contain sufficient information to uniquely define the process conditions applicable to the model. Examples of this may be changes in low level instrumentation and control loops due to maintenance, or merely the fact that different shifts use different operational strategies. One shift may for instance prefer to run a low level feeder control loop in manual as opposed the others running the same loop in automatic mode. The implication of the need for added information is that close ties between developers and key process personnel is important for efficient model development, in particular to swiftly provide the additional information if and when necessary.

In addition to prediction accuracy a model used for optimization-based control should be smooth to facilitate the search algorithm for solving the online constrained nonlinear optimization problem. Hence, it is not necessarily to develop a model with good prediction accuracy. The model should also be smooth with respect to the control inputs eligible for optimization. Our experience is that this has been a key issue to obtain robust and computationally efficient performance of the optimization algorithm both in the reference cases on metal refining and in the suspension PVC-application.

To elaborate on the metal refining case the basic kinetics models and thermodynamics are non-smooth. The non-smooth function were changed by applying sigmoid-functions. To illustrate assume the following kinetic model for the reaction rate r for the reaction $B \rightarrow A$.

$$r = \begin{cases} a(p_B - p_{equil}) & \text{if } p_B > p_{equil} \\ 0 & \text{if } p_B \leq p_{equil} \end{cases}$$

$a > 0$ is some constant, p_B is the partial pressure of (gas) component B, and p_{equil} is the equilibrium partial pressure. A smooth approximate model for the reaction rate, which, however, does allow negative reaction rates, is

$$r = h(p_B, p_{equil}) \cdot [a(p_B - p_{equil})]$$
$$\text{where } h(p_B, p_{equil}) = \frac{1}{1 + e^{-\alpha(p_B - p_{equil})}}, \; \alpha > 0$$

4 State Estimation

The aim of the state estimator is to provide a robust and reliable estimate of the current state at all times. This is a challenging problem since the process data vector in our experience often is dynamic in the sense that data points routinely are biased, delayed or even missing. Examples of this are delayed or missing measurements at the startup of the metal refining batch, and delayed temparature and composition measurements in the aluminum electrolysis cell. The former typically happens on a sporadic basis while as the latter occurs regularly. The time delays in the process data from the aluminum electrolysis cell, however, may vary significantly from one sample to another.

The dynamic process data vector constitutes a challenge since an application requires a robust and reliable estimate of the current state at all times. We have applied two methods for state estimation, an extended and augmented Kalman filter (EAKF) with some modifications and recently a moving horizon estimator (MHE), see eg. [10]. The estimation software includes handling of asynchronous measurements with arbitrary sampling intervals and varying measurement delays.

It is our experience the Kalman filter has proved to work very well in several demanding applications, even if this simple estimation algorithm provides a crude approximative solution to the underlying nonlinear stochastic estimation problem. The performance of the EAKF for a specific application depends, however, crucially on the modelling of the stochastic process disturbances and on the choice of which model parameters to estimate recursively in addition to the model states. In Kalman filtering the process disturbances are modelled as filtered white noise, and this disturbance model should reflect how the true process disturbances and uncertainties are anticipated to influence the real process. Special attention should be directed towards fulfilling basic mass and energy balance requirements. It is, however, a shortcoming of the Kalman filter that these balances will generally not be exactly fulfilled even if the process disturbances are properly modelled. This is due to the linearization approximations involved in the calculation of model state updates from measurement prediction deviations.

The choice of which parameters to estimate in the EAKF should be guided by an identifiability analysis. Usually we cannot assume that the process excitations fulfil certain persistency requirements in order to ensure convergence of parameter estimates. Hence, we normally choose a set of parameters which is identifiable from stationary data, and which do not require any particular excitations in order to obtain convergence. By carefully selecting the set of model parameters to estimate, we can usually obtain zero steady-state deviations in measurement predictions.

The MHE has several advantages compared to the Kalman filter. Evident advantages are the ability to handle varying measurement delays as well as constraints in a consistent manner. As mentioned above varying delays occur in some of our applications. The ability to include constraints is also important since a nonlinear mechanistic model by definition includes physically related states and parameters, variables which often can be limited by a lower and upper bound.

Other advantages of the MHE are related to the increased accuracy in solving the underlying stochastic estimation problem. The stochastic estimation problem is solved exactly over the length of the horizon. Hence, as the estimation horizon increases towards infinity, the MHE estimate approaches the true solution of the underlying nonlinear estimation problem.

The main disadvantage with the MHE as compared to the Kalman filter is the increased computational requirements for the MHE. The estimation of stochastic process disturbances over a long estimation horizon, when compared to the length of the sampling interval, may lead to a nonlinear programming problem of untractable size. Hence, in practical applications it might be necessary to restrict the length of the horizon or to parameterize the process disturbances with a limited number of parameters over the estimation horizon. These modifications will generally reduce the accuracy of the MHE.

The bulk of our experimence is based on the use of the EAKF. Because of the advantages of the MHE scheme, despite its drawback from a computational point of view, we foresee a shift towards this estimation scheme in future applications.

The state estimator in itself often provides interesting information about the process conditions. Hence, commissioning the state estimator, assuming that it provides reliable estimates, before the actual NMPC application is in our experience a favourable option. This provides at least three positive effects. First, the state estimator, a critical component of the NMPC application, is tested in its real environment. Such an environment will always provide some challenges not present in a testing environment. Second, the state estimates may provide important information to plant personnel. This is for instance the case for the aluminum electrolysis cell where estimates of internal cell states are highly interesting. Finally, the estimator builds trust and interest in the future NMPC application.

5 Control Formulation and Online Optimization

Formulating the control problem, ie. the online optimization problem, tends to be simpler than the modeling and estimation tasks described above. The online problem for an NMPC application does not in principle differ from the linear MPC case. The objective function will in some sense be related to economic conditions. For a batch reactor, in which batch capacity limits production, minimizing the batch time is in most cases equivalent to optimizing an economic criterion. This was the case both for the metal refining case, the PVC polymerization reactors and the phenol-formaldehyde batch polymerization process. The constraints will limit variables linked to safety and quality. A typical safety constraint is the net cooling capacity in the (exothermic) PVC polymerization reactors while the end point carbon content is an important quality constraint in the metal refining reactor. The choice of control inputs and controlled outputs is again a problem where the issues in linear MPC and NMPC are similar and will hence not be discussed further herein.

Online optimization in NMPC applications is a very different issue than the convex QP-problem normally encountered in linear MPC. A robust and reliable algorithm is critical. In our experience such an algorithm can only be developed by merging insight into nonlinear programming techniques with extensive trials. The NMPC algorithm is based on the Newton-type algorithm developed by Biegler and co-workers [8]. Their approach is to linearize a nonlinear state space model around a nominal trajectory determined by the input sequence computed at the previous sampling time. A new input sequence is computed by solving a quadratic program, once over the time horizon, followed by a line search where the quadratic optimization criterion is computed based on the nonlinear model. Through the line search, global convergence of the method is enforced as long as the objective function exhibits descent directions. Sufficient conditions for global convergence and stability are developed by Li and Biegler [6]. Their development assumes that the states are available, hence state estimation is not considered in the referenced paper. The algorithm we use extends and modifies the Newton-type algorithm proposed by Biegler in several ways. The algorithm is based on a linearization of the nonlinear model around nominal input and output trajectories, which are computed at each time step. The linearization is usually performed once at each time step. The optimization criterion is quadratic and the constraints are linear in the process outputs and inputs. The outputs are, however, arbitrary nonlinear functions of the states and the inputs. Another extension of Biegler's algorithm includes more flexible parameterizations of inputs and outputs; each input and output variable is parameterized independently.

Input constraints are hard constraints in the optimization. Output constraints are handled as soft exact penalty type constraints as outlined by Oliveira and Biegler [7].

6 Implementation

Putting NMPC into industrial use requires competence and systems beyond NMPC theory and the algorithms themselves. This includes a project development plan which does not differ from a typical project plan for implementing other advanced controllers. A project will include a functional design specification task which describes the functionality and details the specifications for the delivery. Thereafter the application is developed, integrated into the existing control system, and finally formally accepted by the customer through a Site Acceptance Test.

A difference in the development of applications based on first principles models compared to data driven models is that the time spent at the plant, performing process experiments and application commissioning and testing, is shortened by the use of mechanistic models. The reason is that less experiments are required for the model and state estimation development and tuning. Usually, we have on-line secure internet connection to the application computer. Then the estimator can be tested on-line and also the NMPC application can be tested in open loop before closed loop testing.

In the Cybernetica system an NMPC application will consist of one component, a model component, which is developed specifically for the type of process unit or process section to be controlled. The other parts of the system, such as the EAKF, MHE and NMPC algorithms as well as configuration interface and communication interface to the basic control system, consist of generic components. Usually, the operator interface is integrated into the same operator interface system as used by the basic control system. Hence, from the operators point of view, the introduction of an NMPC application is seen as an extension of the existing control system.

A close dialogue with key personnel at the plant has been critical in many of our reference projects. There are several reasons for this. First, as remarked earlier, personnel have access to information and they possess knowledge which is vital in specifying an application and in the model development stage. Second, the operating personnel may have the privilege to choose between an existing operating strategy as an alternative to a (new) NMPC application. In such a situation it is important that the operators understand the application so as to gain confidence in the new application. Further, insight and motivation definitely helps in a situation where an application needs to be modified due to some unforeseen problems.

In most of the reference cases the NMPC application replaces an existing application. Usually it is necessary to upgrade other parts of the system in conjunction with an NMPC project simply because the NMPC application normally requires more accurate information than what was the case prior to its installation. Improvements typically include upgrading of instrumentation and data collection routines, and retuning of low-level control loops.

7 Conclusions

This paper discusses issues that arise when implementing NMPC in the sense of a nonlinear mechanistic model, state estimation, and the solution of an online constrained nonlinear optimisation problem. Even though this technology presently is in the development stage several demanding applications have been developed with good industrial acceptance.

References

[1] K. S. Andersen, M. Hillestad and T. Koskelainen, Model Predictive Control of a BORSTAR Polyethylene Process, *In: 1st European Conference on the Reaction Engineering of Polyolefins (ECOREP)*, France, 2000.

[2] T. Drengstig, D. Ljungquist and B. A. Foss, "On the AlF_3 and Temperature Control of an Aluminum Electrolysis Cell," *IEEE Trans. of Control System Technology,* vol. 6, no. 2, pp. 157–171, 1998.

[3] W. Dresler, "Limitations to carburization of high carbon ferromanganese", *In: Proceedings 1989 Steelmaking conference*, 1989.

[4] B. A. Foss, B. Lohmann and W. Marquardt, "A field study of the industrial modeling process," *Journal of Process Control,* vol. 8, no. 5-6, pp. 325–338, 1997.

[5] K. Grjotheim and H. Kvande, eds., "Introduction to aluminium electrolysis". *Aluminum-Verlag*, 1993.

[6] W. C. Li and L. T. Biegler, "Multistep, Newton-type control strategies for constrained nonliner processes", *Chem. Eng. Res. Des.*, vol.67, pp. 562-577, 1989.

[7] N. M. C. de Oliveria and L. T. Biegler, "Constraint handling and stability properties of model-predictive control," *AIChE Journal*, vol. 40, pp. 1138-1155, 1994.

[8] N. M. C. de Oliveria and L. T. Biegler, "A finite-difference method for linearization in nonlinear estimation algorithms", *Automatica*, vol.31, pp. 281-286, 1995.

[9] S. J. Qin and T. A. Badgwell, "An Overview of Nonlinear Model Predictive Control", *In: Nonlinear Model Predictive Control*, F. Allgower and A. Zheng Birkhauser, eds., 2000.

[10] C. V. Rao and J. B. Rawlings, "Nonlinear Moving Horizon Estimation", *In: Nonlinear Model Predictive Control*, F. Allgower and A. Zheng Birkhauser, eds., 2000.

Integration of Economical Optimization and Control for Intentionally Transient Process Operation

Jitendra V. Kadam* and Wolfgang Marquardt

Lehrstuhl für Prozesstechnik, RWTH Aachen University, Turmstr. 46, D-52064
Aachen, Germany
* ExxonMobil Chemical Company, Canadastraat 20, B-2070 Zwijndrecht, Belgium
{kadam,marquardt}@lpt.rwth-aachen.de

Summary. This paper summarizes recent developments and applications of dynamic real-time optimization (D-RTO). A decomposition strategy is presented to separate economical and control objectives by formulating two subproblems in closed-loop. Two approaches (model-based and model-free at the implementation level) are developed to provide tight integration of economical optimization and control, and to handle uncertainty. Simulated industrial applications involving different dynamic operational scenarios demonstrate significant economical benefits.

1 Introduction

Increasing competition coupled with a highly dynamic economic environment in the process industry require a more agile plant operation in order to increase productivity under flexible operating conditions while decreasing the overall production cost [1]. The polymer industry is an illustrative example of this development. While on the one hand the product specifications for high-value products become tighter and tighter, on the other hand many of the specialty polymers are becoming commodities resulting in lower profit margins, thus requiring an efficient and cost-effective production [6]. Multi-product and multi-purpose plants have become common. Therefore, transient operational tasks involving sudden changes in production load, product grade (usually triggered by market conditions) are routinely performed. These scenarios demand integrated economical optimization of the overall plant operation.

Today's plant operation requires *real-time business decision making (RT-BDM)* tasks at different levels integrating planning, scheduling, optimization and control tasks. Figure 1 depicts a typical decision making and automation hierarchy. Due to a wide range of process dynamics, different time-scales are involved at each level such as fractions of seconds for base layer control, minutes for advanced control, hours for set-point/trajectory optimization, days for planning and scheduling, and months or even years for strategic corporate planning.

[*] Current address.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 419–434, 2007.
springerlink.com

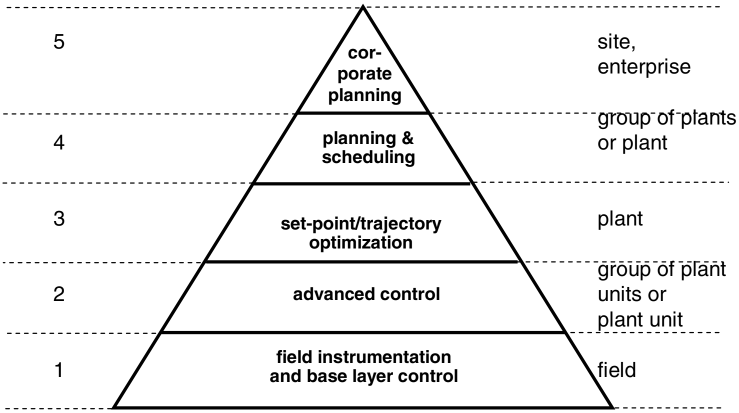


Fig. 1. Real-time business decision making and automation hierarchy

Accordingly, RT-BDM involves multiple decision making levels each with a different objective reflecting the natural time scales. Despite the decomposition in the implementation, there is a single overall objective for the complete structure, namely maximization of profitability and flexibility of plant operation.

In the last decades, technologies have been developed to solve operational problems at different levels of the automation hierarchy. However, most of them are segregated techniques, each one targeting a single problem independently and exclusively. For example, model predictive control technology using linear, nonlinear or empirical models [16, 17] is used to reject disturbances and to control the process at given target set-points (level 2 in Figure 1). The set-points are often the result of a *stationary real-time optimization* [15] using steady-state process models (level 3 in Figure 1). Alternatively, nonlinear model predictive control (NMPC) with an economical objective (referred to as *direct approach* in [9]; Figure 2) has more recently been suggested for transient processes [5] to solve the tasks on level 2 and 3 in Figure 1. On a moving horizon, NMPC repetitively solves a dynamic optimization problem with a combined economical and control objective. On a given time horizon $[t^j, t_f^j]$ with a sampling interval Δt, the corresponding dynamic optimization problem (denoted by the superscript j) reads as:

$$\min_{\boldsymbol{u}^j(t)} \Phi(\boldsymbol{x}(t_f)) \tag{P1}$$

$$\text{s.t.} \quad \dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}^j(t), \hat{\boldsymbol{d}}^j(t)) \,, \quad \boldsymbol{x}(t^j) = \hat{\boldsymbol{x}}^j \,, \tag{1}$$

$$\boldsymbol{0} \geq \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{y}(t), \boldsymbol{u}^{\boldsymbol{j}}(t)), \quad t \in [t^j, t_f^j], \quad t_f^j := t_f^{j-1} + \Delta t, \tag{2}$$

$$\boldsymbol{0} \geq \boldsymbol{e}(\boldsymbol{x}(t_f^j)) \,. \tag{3}$$

$\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ are the state variables with the initial conditions $\hat{\boldsymbol{x}}^j$; $\boldsymbol{y}(t) \in \mathbb{R}^{n_y}$ are the algebraic output variables. The dynamic process model (1) is formulated in $\boldsymbol{f}(\cdot)$. The time-dependent input variables $\boldsymbol{u}^j(t) \in \mathbb{R}^{n_u}$ and possibly the

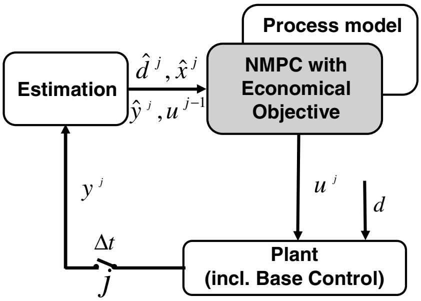


Fig. 2. NMPC with economical objective (or direct approach) using a process model

final time are the decision variables for optimization. Furthermore, equations (2) and (3) denote path constraints $\boldsymbol{h}(\cdot)$ on input and state variables, and end-point constraints $\boldsymbol{e}(\cdot)$ on state variables, respectively. Uncertainties of different time-scales $\boldsymbol{d}^j(t) \in \mathbb{R}^{n_d}$ (e.g. fast changing model parameters, disturbances, and relatively slow changing external market conditions) are also included in the formulation. In NMPC, measurements $(\boldsymbol{y}^j)$ are used to estimate on-line the current states $(\hat{\boldsymbol{x}}^j)$, outputs $(\hat{\boldsymbol{y}}^j)$ and uncertainties $(\hat{\boldsymbol{d}}^j)$. The inputs $(\hat{\boldsymbol{u}})$ are updated subsequently by an on-line solution of the dynamic optimization problem P1. For large-scale industrial applications, the NMPC problem is computationally expensive to solve though significant progress has been made in recent years (e.g. [2, 5, 18]). Due to the considerable computational requirements, larger sampling intervals (Δt) are required, which may not be acceptable due to uncertainty.

Functional integration can, alternatively, be achieved by a cascaded feedback structure maintaining the automation hierarchy that has been evolved in the process industry with the base layer control (level 1 in Figure 1) being the most inner and the corporate planning (level 5 in Figure 1) the most outer loop. The operational problem formulation (objective, constraints etc.) at each level should be *consistently* derived from its upper level. This is in contrast to the existing technologies used today in the automation hierarchy, where inconsistencies in objectives, constraints and process models exist at each of the different levels. Furthermore, uncertainties due to process disturbances, plant-model mismatch and changes in external market conditions need to be efficiently tackled. Though NMPC could be tailored to deal with the requirements, it is not a cascaded feedback control system which respects the established time-scale decomposition in the automation hierarchy. Furthermore, NMPC lacks functional transparency which complicates human interaction and engineering. Due to these concerns, the acceptance of such a monolith solution in industry is limited. Rather, a cascaded feedback optimizing control strategy is preferred, because it is less complex and computationally better tractable in real-time, but provides approximate control profiles of sufficient quality. In summary, *the overall problem of economical optimization and control of dynamic processes should be decomposed into consistent and simple subproblems, and subsequently re-integrated using* *efficient techniques to handle uncertainty.* This contribution reviews some of useful concepts to address the above mentioned requirements, and presents their application to simulated industrial case studies.

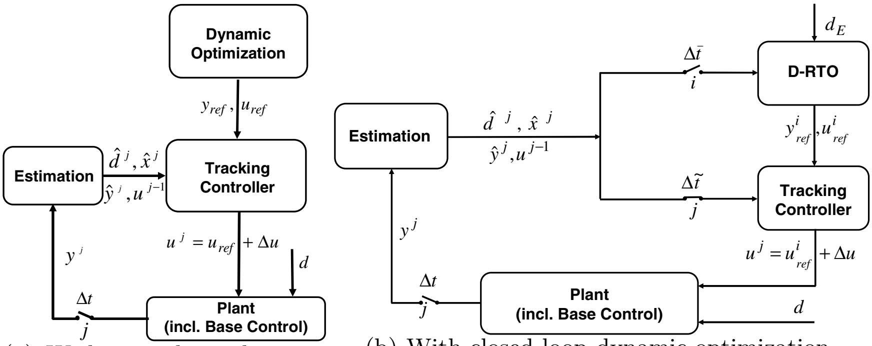
The paper is organized as follows: In Section 2, a two-level optimization and control strategy is presented. To handle uncertainty and tightly integrate the economical optimization and control levels, two strategies are presented in Section 3 and 4. In the first approach in Section 3, a strategy for a fast update of reference tracking trajectories with possible changes in the active constraints set (due to uncertainty) is presented. When the active constraint set is constant, an NCO tracking control approach (in Section 4) can be used, which does not require on-line solution of the dynamic optimization problem and uses only available measurements or estimates of the process variables. The two-level dynamic optimization and control strategy along with fast update and NCO tracking forms a cascaded optimizing control strategy that implements close-to-optimal plant operation. In each section, a simulated industrial application involving different types of transitions is presented.

2 A Two-Level Optimization and Control Strategy

2.1 Concept

For an integration of economical optimization and control, we consider the *two-level* strategy introduced in [9] and modified in [12]. Problem P1 is decomposed into an upper level economical dynamic optimization problem and a lower level tracking control problem, as shown in Figures 3(a) and 3(b). The dynamic optimization in the approach depicted in Figure 3(a) does not involve measurements feedback to update the model. Hence no re-optimization has to be performed online, but suboptimal behavior is unavoidable. Therefore, it is referred to as the two-level approach with open-loop dynamic optimization. In contrast, the approach shown in Figure 3(b) involves feedback and hence on-line re-optimization (D-RTO), but can cope with uncertainty. Consequently, it is referred to as the two-level approach with closed-loop dynamic optimization. Any controller, for example, a PID controller or a predictive controller using a linear, possibly time-variant, or even a nonlinear model-based controller may be used at the lower level to track the reference trajectories of the outputs $\boldsymbol{y}_{ref}$ and the controls $\boldsymbol{u}_{ref}$ which results from a solution of the D-RTO problem at the upper level. The concept of providing reference trajectories for tracking is similar to the calculation of constant targets of controls and outputs used in MPC [17]. Note that economical optimization is considered for the nominal model, at the D-RTO level in the simplest case only, while uncertainty is accounted for on the control level only. Hence, the process model used for the optimization has to have sufficient prediction quality and should cover a wide range of process dynamics. Therefore, a fundamental process model is a natural candidate.

This decomposition has two different time-scales, a slow time-scale denoted by $\bar{t}$ on the D-RTO level and a fast time-scale $\tilde{t}$ on the control level, with the corresponding sampling times $\Delta\bar{t}$ and $\Delta\tilde{t}$, respectively. As shown in Figure 3,



(a) With open-loop dynamic optimization

(b) With closed-loop dynamic optimization

Fig. 3. Two-level dynamic optimization and control strategies

the solution of the upper level dynamic optimization problem determines optimal trajectories $\boldsymbol{u}_{ref}$, $\boldsymbol{y}_{ref}$ for all relevant process variables to minimize an economical objective function. The sampling time $\Delta\tilde{t}$ (the time interval between two successive re-optimizations performed in the approach in Figure 3(b)) has to be sufficiently large to capture the process dynamics, yet small enough to make flexible economic optimization possible. Depending on whether uncertainty affects the reference trajectories, the two-level approach can be implemented with open-loop (with $\Delta\bar{t} = \infty$) or closed-loop (with $\Delta\bar{t} = \Delta\bar{t}_0$) dynamic optimization depending on the requirements of the application at hand.

On the lower level, the control problem is solved in a delta mode to track the optimal reference trajectories (see Figure 3). The tracking controller calculates only updates $\Delta\boldsymbol{u}$ to $\boldsymbol{u}_{ref}$ (provided by the upper level as feed-forward part of the control) at every sampling time $\tilde{t}_j$ to minimize the deviation from $\boldsymbol{y}_{ref}$. Hence, the degree of optimality achieved by employing the two-level approach depends upon the reference trajectories provided by dynamic optimization at the upper level. The set of tracked variables in $\boldsymbol{y}_{ref}$ is selected from the important output variables available in the plant. The sampling interval $\Delta\tilde{t}$ has to be reasonably small to handle the fast, control relevant process dynamics. The values of the initial conditions $\hat{\boldsymbol{x}}^j$ and disturbances $\hat{\boldsymbol{d}}^j$ for the control problem are estimated from measurements by a suitable estimation procedure such as an extended Kalman filter or a moving horizon estimator.

2.2 Optimal Load Change of an Industrial Polymerization Process

An industrial polymerization process is considered. The problem has been introduced by Bayer AG as a test case during the research project INCOOP [11].

Process description: The flowsheet of this large-scale continuous polymerization process is shown in Figure 8. The exothermic polymerization involving multiple reactions takes place in a continuously stirred tank reactor (CSTR) equipped with an evaporative cooling system. The reactor is operated at an open-loop unstable operating point corresponding to a medium level of conversion. It is followed by a separation unit for separating the polymer from unreacted monomer and solvent. Unreacted monomer and solvent are recycled back to the reactor via a recycle tank, while the polymer melt is sent to downstream processing and blending units. For this process, the following measurements (or estimates) are considered to be available: Flowrates of recycle and fresh monomers, $F_{M,R}$ and $F_{M,in}$, flowrate of reactor outlet $F_{R,out}$, recycle tank holdup V_{RT}, reactor solvent concentration C_S, reactor conversion μ, polymer molecular weight M_W. The reactor holdup V_{RT} is maintained at a desired setpoint using a proportional control that manipulates the reactor outlet flowrate $F_{R,out}$. A rigorous dynamic process model consisting of about 2500 differential and algebraic equations is available from previous studies at Bayer AG [6].

Results: The following scenario is a typical example for an intentionally dynamic mode of operation. Due to changed demand from the downstream processing unit, the polymer load needs to be instantaneously changed from 50% load to 100% load and back to 50% load after a given time interval. It is desired, if possible at all, to produce on-spec polymer during the transition and thereafter. Otherwise, the total amount of off-spec polymer produced during the transition should be minimized. At the end of the transition and thereafter, the process is required to be at the given steady-state operating point. The polymer quality variables, reactor conversion and polymer molecular weight, are allowed to vary in a band of ±2% around their specifications. Three input variables $\boldsymbol{u}$ are available: Flowrate of fresh monomer $F_{M,in}$, catalyst feed stream $F_{C,in}$ and flowrate of recycled monomer $F_{M,R}$. Path and end-point constraints on five process variables need to be respected during the load change operation. Various uncertainties and disturbances in the form of unknown solvent concentration and initial conditions, measurement errors need to be considered during the transition.

The two-level strategy with open-loop dynamic optimization and control ($\Delta\bar{t} = \infty$; cf. Figure 3(a)) has been implemented in a software environment and applied to the simulated polymerization process for the load change scenario. For this transitional scenario, off-line optimization studies have shown that the prevalent uncertainties and disturbances have an insignificant effect on the optimal reference trajectories. Only representative results from the closed-loop control simulation are reported in Figure 4 (see [6] for further details). The solid lines in the plots show the optimal reference trajectories which are calculated by solving a dynamic optimization problem that employs the nominal process model. The lower level of the two-level strategy involving estimation and control was run in a closed-loop simulation in order to verify its capabilities to follow the reference trajectories in the presence of the various process disturbances. A linear time-variant model derived repetitively on-line along the reference trajectories is employed in the tracking controller. The optimization of the load transitions led to significantly improved operation of the plant, when compared to the conventional strategies used by the operators. The transition time is drastically reduced, and the production of off-spec material can be completely avoided, which also could not be ensured in conventional operation.

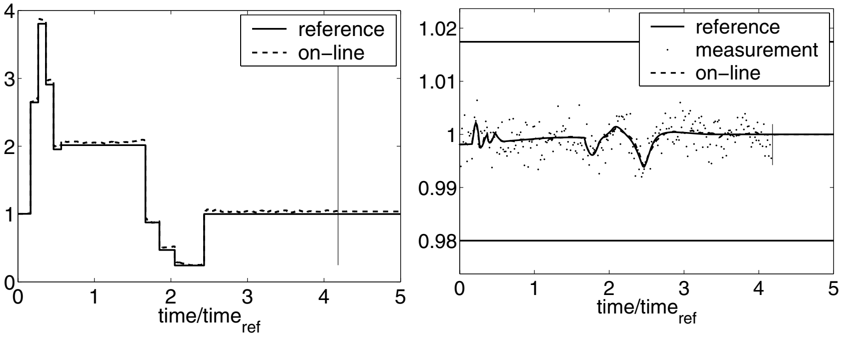


Fig. 4. Results using the two-level strategy with open-loop dynamic optimization: Fresh monomer flowrate $F_{M,in}$ (left) and polymer molecular weight M_W (right)

2.3 Tight Integration and Uncertainty Handling

The two-level strategy is essentially a *cascaded optimizing feedback control system* which generalizes steady-state RTO and advanced predictive control of intentionally dynamic processes. In this approach, the overall problem is *decomposed* into two sub-problems (with *consistent* objectives) that need to be subsequently *re-integrated* in closed-loop. Furthermore, to consider effects of uncertainty, the tracking reference trajectories can be updated by repetitive re-optimization using the feedback (state and eventually model update) provided at a constant time interval $\Delta\bar{t}$. However, a repetitive re-optimization is not always necessary. Rather, it can be systematically triggered by analyzing the optimal reference trajectories based on the disturbance dynamics and its predicted effect on the optimality of P1 if needed. Two strategies are proposed for uncertainty handling and tighter integration of the two-levels of dynamic optimization and control subsequently. In the first approach introduced in Section 3, a neighboring extremal control approach is used for linear updates of the reference trajectories even in case of active inequality constraints. In the second approach presented in Section 4, a solution model is derived from a nominal optimal solution of the dynamic optimization problem. The resulting solution model is used to implement a decentralized supervisory control system to implement a controller with close-to-optimal performance even in case of uncertainty.

3 Sensitivity-Based Update of Reference Trajectories

3.1 Concept

Due to uncertainty, the reference trajectories of the inputs and outputs need to be updated. So far, the update is done via repetitive re-optimization, which can be computationally expensive. Furthermore this may not be necessary as the updated solution and the predicted benefits (objective function) may not be significantly different from the reference solution. Parametric sensitivity analysis

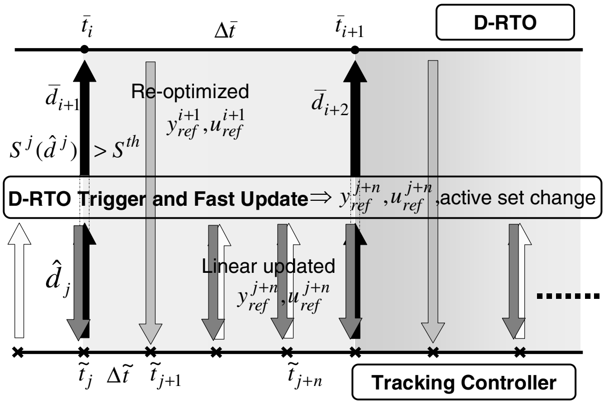


Fig. 5. Schematic of D-RTO trigger and fast update strategy

[7] is a strong tool to analyze an optimal solution for perturbations of parameter values. Consequently, this analysis has been extensively used in steady-state and dynamic optimization for calculating updates due to parametric perturbations or parametric uncertainty (cf. [4]) because it demands only negligible computational time. The applicability of parametric sensitivity techniques, also referred to as *neighboring extremal control*, depends upon the strong assumption that the active constraint set does not change with perturbations, which is often quite restrictive. The assumption is only valid for sufficiently small perturbations entering the optimization problem.

A trigger strategy is suggested in the two-level strategy in Figure 3(b) to initiate a solution of the D-RTO problem only if necessary, otherwise it provides linear updates to $\boldsymbol{u}_{ref}, \boldsymbol{y}_{ref}$ based on the neighboring extremal control with the handling of possible changes in the active constraints set. A schematic of the D-RTO trigger and fast update strategy is given in Figure 5. The reader is referred to [10] for algorithmic details. An optimal solution is available at the nominal values of uncertainty parameters from the previous optimization at time $\bar{t}_i$ and updates at time $\tilde{t}_j$. At each sampling time $\tilde{t}_j$, reference trajectories of the controls are updated as u_{ref}^{j+1}, and the changed active constraint set is calculated using the neighboring extremal control strategy with inequality constraints [10]. Simultaneously, sensitivities S^j of the Lagrange L^j of P1 are evaluated as $S^j = \frac{\partial L^j(u_{ref}^{j+1}, \hat{d}^j)}{\partial u}$, where L^j is calculated for the updated controls u_{ref}^{j+1} and the uncertainty estimate $\hat{d}^j$. A D-RTO trigger criteria ($S^j > S^{th}$ with S^{th} as threshold value) is defined to analyze the updated control for optimality of P1. If the criteria is met, a linear update is not sufficient and a re-optimization is performed to calculate new reference trajectories $\boldsymbol{u}_{ref}^{i+1}, \boldsymbol{y}_{ref}^{i+1}$.

3.2 Productivity Maximization of a Semi-batch Reactor

Problem description: A semi-batch reactor is considered here, which is derived from the continuous Williams-Otto benchmark reactor [8]. The following reactions are taking place in the reactor: $A{+}B \xrightarrow{k_1} C$, $C{+}B \xrightarrow{k_2} P{+}E$, $P{+}C \xrightarrow{k_3} G$. The reactor is fed initially with a fixed amount of reactant A; reactant B is fed continuously. The first-order reactions produce the desired products P and E. Product G is a waste. As the heat generated by the exothermic reactions is removed through the cooling jacket by manipulating the cooling water temperature. During reactor operation, path constraints on the feed rate of reactant B ($F_{B_{in}}$), reactor temperature (T_r), hold-up (V) and cooling water temperature (T_w) have to be respected. $F_{B_{in}}$ and T_w are the manipulated variables. The operational objective is to maximize the yield of the main products at the end of batch. A measurable disturbance ΔT_{in} affects the feed temperature at $t =$ 250 sec during batch operation. Furthermore, the parameter b_1 in the reaction kinetic equation $k_1{=}a_1 \exp(\frac{b_1}{T_r+273.15})$ is assumed to vary about ±25% from its nominal value $b_1 = 6666.7 \text{ sec}^{-1}$.

Results: The economical optimization problem is solved using DyOS [19] to obtain the optimal solution for nominal values of the uncertain parameters. The nominal optimal control and constraint profiles are depicted in Figure 6 by solid lines. These profiles have different arcs corresponding to active and inactive parts of the path constraints, which are characterized as follows: $F_{B_{in}}$ is initially kept at its *upper bound* and then switched to its *lower bound* when the *reactor volume* (V) reaches its *upper bound.* The second control variable T_w is manipulated to move the *reactor temperature* (T_r) to its lower bound at $t{=}140$ sec and keep it there. At the switching time $t{=}360$ sec, T_r is moved away from its lower bound by *manipulating* T_w in a bang-bang profile with the switching times computed implicitly by optimization. Note that T_w is at its lower bound at $t{=}0$ sec and quickly switched to its upper bound.

The profiles shown by a solid line with dots in Figure 6 depict the response of the neighboring extremal control update and D-RTO trigger strategy in the presence of uncertainty and disturbances. Only once a re-optimization was triggered in this episode. It can be observed in the figures that the closed-loop linearly updated solution is almost identical to the synchronously re-optimized solution (depicted by dashed lines). Note that the structure of the true optimal solution under uncertainty and disturbances is drastically different from that of the nominal solution. Most interestingly, $F_{B_{in}}$ is stopped at $t{=}282$ sec, and again switched back to its upper bound at $t{=}656$ sec until the reactor hold-up reaches its upper bound. Furthermore, the reactor temperature is never at either of its bounds, while T_w is at its lower bound throughout the operation. These changed active sets are correctly and timely detected by the sensitivity based-update strategy, and the batch operation is optimized in real-time. It is shown that by using the D-RTO trigger and the linear fast update in two-level integrated dynamic optimization economical and control, large uncertainty and disturbances can be effectively handled.

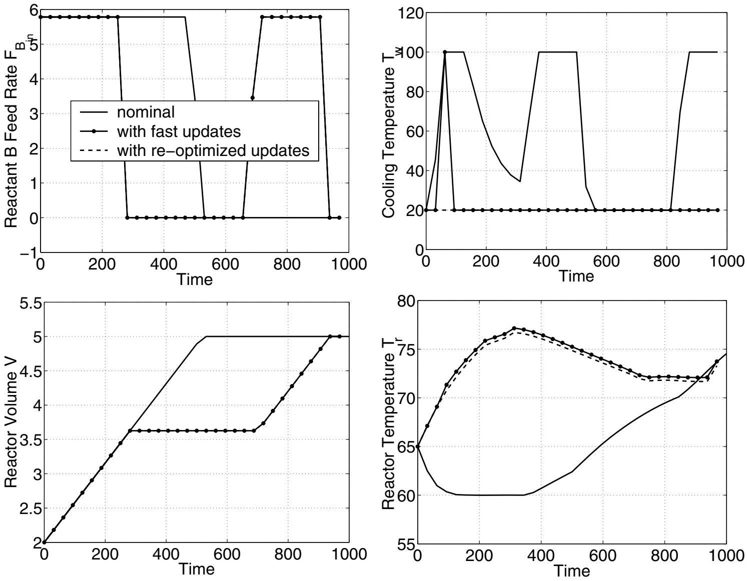


Fig. 6. Nominal and closed-loop optimization profiles of controls and constraints

4 Solution Model-Based NCO Tracking

4.1 Concept

Instead of using *uncertainty-variant* tracking reference trajectories as presented in Sections 2 and 3, a *combination of uncertainty-variant and uncertainty-invariant* arcs of the optimal solution is deduced for the tracking control problem. The approach is termed *NCO tracking* [20] as it adjusts the inputs by means of a decentralized control scheme in order to track the necessary conditions of optimality (NCO) of problem (P1) (cf. Table 1 [3]) in the presence of uncertainty. As shown in Figure 7, measurements ($\boldsymbol{y}$) are employed to directly update the inputs ($\boldsymbol{u}$) using a parameterized solution model obtained from off-line numerical solution of problem (P1) [21]. This way, nearly optimal operation is implemented via feedback control without the need for solving a dynamic optimization problem in real-time. The real challenge lies in the fact that four different objectives (Table 1) are involved in achieving optimality. These path and terminal objectives are linked to active constraints (row 1 of Table 1) and sensitivities (row 2 of Table 1). Hence, it becomes important to appropriately parameterize the inputs using time functions and scalars, and assign them to the different objectives. There results a solution model, i.e. a decentralized self-optimizing control scheme, that relates the available decision variables (seen as inputs) to the NCO (seen as measured or estimated outputs).

Table 1. Separation of the NCO into four distinct parts

	Path objectives	Terminal objectives
Constraints	$\boldsymbol{\mu}^T \boldsymbol{h} = 0$	$\boldsymbol{\nu}^T \boldsymbol{e} = 0$
Sensitivities	$\frac{\partial H}{\partial \boldsymbol{u}} = \mathbf{0}$	$H(t_f) + \left.\frac{\partial \Phi}{\partial t}\right\|_{t_f} = 0$

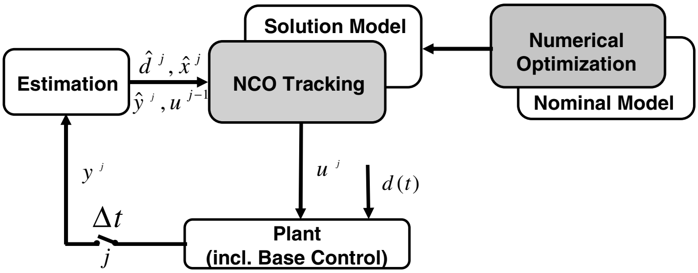


Fig. 7. D-RTO via numerical optimization of a nominal model and NCO tracking

The generation of a solution model includes two main steps:

- *Input dissection:* Using the structure of the optimal solution provided by off-line numerical optimization, this step determines the so-called fixed (uncertainty-invariant) and free (uncertainty-variant) arcs of the inputs. In some of the arcs, the inputs are independent of the prevailing uncertainty, e.g. in arcs where the inputs are at their bounds, and thus can be applied in an open-loop fashion. Hence, the corresponding input elements can be considered as fixed in the solution model. In other arcs, the inputs are affected by uncertainty and need to be adjusted for optimality based on measurements. All the input elements affected by uncertainty constitute the *decision variables* of the optimization problem.

  Input dissection is based on off-line numerical optimization using a nominal process model. The resulting optimal solution consists of various arcs or intervals [3]. The information on the type of arcs can be deduced from the numerical solution of (P1). Schlegel and Marquardt [18] have proposed a method that automatically detects the control switching structure even for large-scale problems with multiple manipulated variables as well as path and endpoint constraints. The structure detection algorithm also provides the dissected optimal input profiles that are re-parameterized with a small number of parameters: $\boldsymbol{u}(t) = \mathcal{U}(\boldsymbol{\eta}(t), \boldsymbol{\mathcal{A}}, \boldsymbol{\tau})$, where $\boldsymbol{\eta}(t) \in \mathbb{R}^L$ are the time-variant arcs, $\boldsymbol{\tau} \in \mathbb{R}^L$ the switching times, and L the total number of arcs. The set of decision variables is comprised of $\boldsymbol{\eta}(t)$ and $\boldsymbol{\tau}$. The boolean set $\boldsymbol{\mathcal{A}}$ of length L describes the type of each particular arc, which can be of the type $\{u_{min}, u_{max}, u_{state}, u_{sens}\}$ depending on whether the corresponding input u_i

is at its lower or upper bound, determined by a state constraint or such that it is adjusted to minimize the objective function.

- *Linking the decision variables to the NCO:* The next step is to provide a link between every decision variable and each element of the NCO as given in Table 1. The active path and terminal constraints fix some of the time functions $\boldsymbol{\eta}(t)$ and scalar parameters $\boldsymbol{\tau}$, respectively. The remaining degrees of freedom are used to meet the path and terminal sensitivities. Note that the pairing is not unique. An important assumption here is that the set of active constraints is correctly determined and does not vary with uncertainty. Fortunately, this restrictive assumption can be relaxed by considering a *superstructure of the solution model* and process insight, which takes into account foreseen changes in the nominally active constraints set.

A designed solution model in the form of input-output pairing provides the basis for adapting the decision variables by employing appropriate controllers and measurements or estimates of its related NCO element as feedback. On-line implementation requires reliable on-line measurements of the corresponding NCO parts. In most applications, measurements of the constrained variables are available on-line. When on-line measurements of certain NCO parts are not available (e.g. sensitivities and terminal constraints), a model can be used to predict them. Otherwise, a run-to-run implementation that uses measurements at the end of the run becomes necessary.

4.2 Optimal Grade Transition of an Industrial Polymerization Process

The same polymerization process presented in Section 2.2 is used to produce different grades of polymer. Therefore, grade changes are routinely performed in this process. The optimization of grade transition is considered in this study. The task is to perform a change from polymer grade A of molecular weight $\bar{M}_{W,A} = 0.727 \pm 0.014$ to grade B of molecular weight $\bar{M}_{W,B} = 1.027 \pm 0.027$ in minimum time. During the transition, operational constraints are enforced on the state and input variables. Additionally, there are endpoint constraints on the reactor conversion μ and the polymer molecular weight $M_{W,B}$ that are more strict than those enforced on these quantities during the transition. For a detailed discussion on this case study and the complete set of results, the reader is referred to [13].

The optimal grade change problem is solved numerically using the dynamic optimizer DyOS [18]. To find an accurate optimal solution with an identifiable control structure, a wavelet-based adaptive refinement method combined with an automatic control structure detection algorithm [18, 19] is applied. The nominal optimal solution and its automatically detected structure are characterized, and a solution model linking inputs to parts of the NCO is derived. As certain nominally inactive path constraints can become active in the presence of model and process uncertainties, a superstructure solution model (to consider *foreseen* changes in nominally active constraints set) is developed. The input-output links

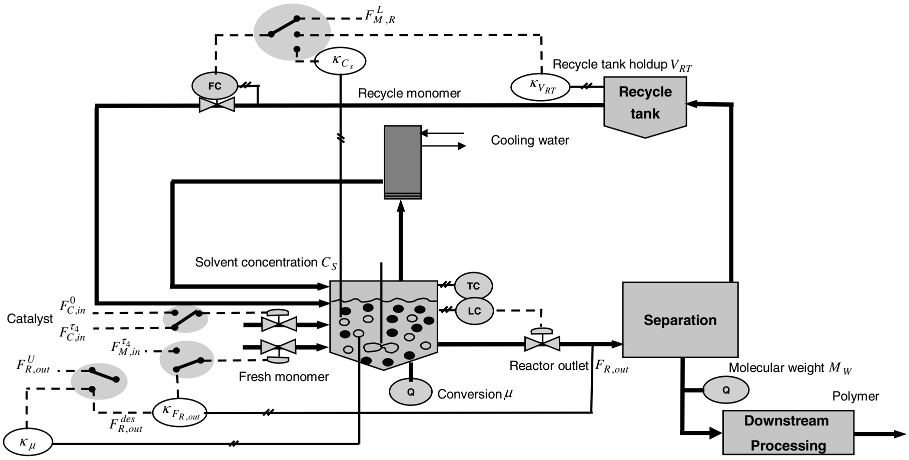


Fig. 8. Process schematic with the NCO tracking controllers and triggers

in the solution model are implemented using the controllers $\mathcal{K}$ as depicted in Figure 8. In this study, PI-type controllers are used employing the nominal input profiles as feedforward terms. Advanced controllers could also be used for improved tracking performance (cf. Section 2.2 and [6]). In the designed control superstructure, depending upon the state of the process, one controller *overrides* the other. In the classic process control terminology, this type of control structure is referred to as *overriding* or *signal-select* controller [14]. Reliable on-line measurements or estimates of the constrained variables are necessary for implementing the NCO tracking strategy using the superstructure solution model.

A considerable amount of uncertainty due to different than nominal initial conditions and reactor solvent concentration is present in practice. The proposed NCO tracking superstructure for optimal grade transition is tested for its performance in the presence of uncertainty using the simulated plant model. The PI controllers are tuned for the nominal case. The simulated NCO tracking profiles of fresh monomer flowrate $F_{M,in}$ and polymer molecular weight M_W are depicted by dash-dotted lines in Figure 9. The transition time t_f for the uncertainty case is considerably larger than that for the nominal case, which is calculated on-line in simulation by using the solution model. The performance of the NCO tracking solution is compared to a robust solution and optimization with known uncertainty in Table 2. The robust solution (column 2 of Table 2) represents a single strategy computed off-line which is feasible for both the nominal and perturbed cases. Such an approach is often used in industrial practice to avoid real-time optimization. The NCO tracking approach (column 3 of Table 2) is computed using the decentralized control structure presented in Figure 9. Finally, the numerical optimal solution (column 4 of Table 2) corresponds to the best possible solution that can be computed using full knowledge of the uncertainty. Table 2 shows that the robust solution is rather poor. In

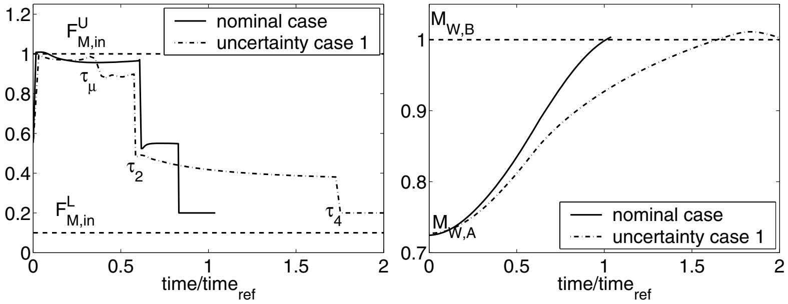


Fig. 9. NCO tracking solution profiles for nominal and uncertainty case 1: Fresh monomer flowrate $F_{M,in}$ (left) and polymer molecular weight M_W (right)

Table 2. Transition times using different optimization strategies; two distinct case of uncertainty are considered, each with different initial conditions corresponding to different solvent concentrations

Case	Robust solution (uncertainty known)	NCO tracking (uncertainty unknown)	Num. optimization (uncertainty known)
nominal	≥ 5	1.008	1.0
uncertainty 1	≥ 5	2.03	1.81
uncertainty 2	≥ 5	0.938	0.915

contrast, NCO tracking comes very close to the best possible solution, without knowledge of the uncertainty but at the expense of on-line measurements and (possibly) state estimations.

The results have demonstrated that a simple decentralized control strategy using a solution model and measurements can implement a complex grade transition. It must be re-emphasized that the generation of the solution model as well as its superstructure requires the optimal solution for the nominal case and process insights that help to simplify it. However, the economic benefits in terms of transition time reduction, and thus the amount of off-spec material, is quite significant compared to the conventional approach practiced in the plant. For limited grade transitions, nominal optimal solutions can be calculated off-line and implemented on-line using NCO tracking controllers. However, reliable on-line measurements or model-based estimates of certain variables are required. Furthermore, the solution model has to be tested and validated for different realization of uncertainty as an online re-optimization is not considered.

5 Conclusions

In this contribution, it is emphasized that the scope of NMPC needs to be broadened from its classic roles of set-point tracking and disturbance rejection. In the context of real-time business decision making (RT-BDM) implemented in the automation hierarchy, a cascaded optimizing feedback control strategy is necessary for economical and agile plant operation. A two-level decomposition strategy of dynamic optimization and control of transient processes is suggested. The overall objectives of profitability and flexibility with respect to scheduled or un-scheduled transitions are maintained consistently at two optimization and control levels. For tighter integration and effective uncertainty handling, two approaches based 1) on neighboring extremal control with inequality constraints and 2) on decentralized control for tracking the necessary conditions of optimality of the economical optimization problem are used. The simulated industrial applications for different transitions have shown significant economical benefits. The case studies show the potential of the suggested approaches. Obviously, there are many opportunities for the further development of an integrated dynamic optimization and control system implemented in multiple levels that consistently solves simple level-specific problems as part of the automation hierarchy.

References

[1] T. Backx, O. Bosgra, and W. Marquardt. Towards intentional dynamics in supply chain conscious process operation. In *FOCAPO 98, Snowbird, Utah.* www.lpt.rwth-aachen.de/Publication/Techreport/1998/LPT-1998-25.html, 1998.

[2] L.T. Biegler, A.M. Cervantes, and A. Wächter. Advances in simultaneous strategies for dynamic process optimization. *Chem. Eng. Sci.*, 24:39–51, 2002.

[3] A.E. Bryson and Y.-C. Ho. *Applied Optimal Control.* Taylor & Francis, Bristol, PA, 1975.

[4] C. Büskens and H. Maurer. Sensitivity analysis and real-time optimization of parametric nonlinear-programming problems. In M. Grötschel, S. O. Krumke, and J. Rambau, editors, *Online Optimization of Large Scale Systems*, pages 3–16. Springer, 2002.

[5] M. Diehl, H.G. Bock, J.P. Schloder, R. Findeisen, Z. Nagy, and F. Allgower. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control*, 12(4):577–585, 2002.

[6] G. Dünnebier, D. van Hessem, J.V. Kadam, K.-U. Klatt, and M. Schlegel. Prozessführung und Optimierung von Polymerisationsprozessen. *Chemie Ingenieur Technik*, 76(6):703–708, 2004.

[7] A.V. Fiacco. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming.* Academic Press, New York, 1983.

[8] J.F. Forbes. *Model structure and adjustable parameter selection for operations optimizations.* PhD thesis, McMaster University, 1994.

[9] A. Helbig, O. Abel, and W. Marquardt. Structural concepts for optimization based control of transient processes. In F. Allgöwer and A. Zheng, editors, *Nonlinear Model Predictive Control*, pages 295–311. Birkhäuser, Basel, 2000.

[10] J.V. Kadam and W. Marquardt. Sensitivity-based solution updates in close-loop dynamic optimization. In S.L. Shah and J.F. MacGregor, editors, *Proceedings of the DYCOPS 7 conference*, 2004.

[11] J.V. Kadam, W. Marquardt, M. Schlegel, O. H. Bosgra T. Backx, P.-J. Brouwer, G. Dünnebier, D. van Hessem, A. Tiagounov, and S. de Wolf. Towards integrated dynamic real-time optimization and control of industrial processes. In I. E. Grossmann and C. M. McDonald, editors, *Proc. of FOCAPO 2003*, pages 593–596, 2003.

[12] J.V. Kadam, W. Marquardt, M. Schlegel, R.L. Tousain, D.H. van Hessem, J. van den Berg, and O. H. Bosgra. A two-level strategy of integrated optimization and control of industrial processes: a case study. In J. Grievink and J. v. Schijndel, editors, *European Symposium on Computer Aided Process Engineering - 12*, pages 511–516. Elsevier, 2002.

[13] J.V. Kadam, B. Srinivasan, D. Bonvin, and W. Marquardt. Optimal grade transition in industrial polymerization processes via NCO tracking. *Submitted to: AICHE J.*, 2005.

[14] M.L. Luyben and W.L. Luyben. *Essentials of Process Control.* McGraw-Hill, New York, USA, 1997.

[15] T.E. Marlin and A.N. Hrymak. Real-time optimization of continuous processes. In *Chemical Process Control-V: Proceedings of the 5^{th} International Conference of Chemical Process Control*, volume 93 of *AIChE Symposium Series: 316*, pages 156–164, 1997.

[16] S.J. Qin and T.A. Badgwell. A survey of industrial model predictive control technology. *Control Engineering Practice*, 11:733–764, 2003.

[17] J.B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, 20(3):38–52, 2000.

[18] M. Schlegel and W. Marquardt. Detection and exploitation of the control switching structure in the solution of dynamic optimization problems. *Journal of Process Control*, 16:275–290, 2006.

[19] M. Schlegel, K. Stockmann, T. Binder, and W. Marquardt. Dynamic optimization using adaptive control vector parameterization. *Comp. Chem. Eng.*, 29:1731–1751, 2005.

[20] B. Srinivasan and D. Bonvin. Dynamic optimization under uncertainty via NCO tracking: A solution model approach. In C. Kiparissides, editor, *Proc. BatchPro Symposium 2004*, pages 17–35, 2004.

[21] B. Srinivasan, D. Bonvin, E. Visser, and S. Palanki. Dynamic optimization of batch processes II. Role of measurements in handling uncertainty. *Comp. Chem. Eng.*, 27:27–44, 2003.

Controlling Distributed Hyperbolic Plants with Adaptive Nonlinear Model Predictive Control

José M. Igreja[1], João M. Lemos[2], and Rui Neves da Silva[3]

[1] INESC-ID/ISEL , R. Alves Redol 9, 1000-029 Lisboa, Portugal
jose.igreja@deq.ipl.pt
[2] INESC-ID/IST, R. Alves Redol 9, 1000-029 Lisboa, Portugal
jlml@inesc-id.pt
[3] FCT, UNL, 2829-516 Caparica, Portugal
rns@fct.unl.pt

1 Introduction

A number of plants of technological interest include transport phenomena in which mass, or energy, or both, flow along one space dimension, with or without reactions taking place, but with neglected dispersion. This type of processes are described by hyperbolic partial differential equations [4] and is receiving an increasing attention in what concerns the application of Predictive Control [6]. Two examples considered are distributed collector solar fields [3, 10] and tubular bioreactors [5]. In both cases the manipulated variable is assumed to be the flow. For lack of space, only the first example is considered hereafter.

1.1 Distributed Collector Solar Fields

In simple terms, a distributed collector solar field [3, 10] consists of a pipe located at the focus of parabolical concentrating mirrors. Inside the pipe flows an oil which is to be heated. The manipulated variable is the oil speed (proportional to oil flow) and the aim consists in regulating the outlet oil temperature. The main disturbances are the solar radiation intensity and the inlet oil temperature. For the purposes of control design the field may be modelled by the following hyperbolic PDE resulting from an energy balance:

$$\frac{\partial T(z,t)}{\partial t} + \frac{u(t)}{L}\frac{\partial T(z,t)}{\partial z} = \alpha\ R(t) \tag{1}$$

Here, *T(z,t)* is the oil temperature at normalized position z measured along the field and at time t, u is the oil velocity and R is the intensity of solar radiation, assumed to depend only on time t. The parameter L is the pipe length. The actual space coordinate (measured in $[meters]$) is given by zL. The parameter α is unknown and it will be called ”efficiency” since it is related to mirror efficiency, although it also depends on other factors, such as the oil specific heat, that is a nonlinear function of temperature.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 435–441, 2007.
springerlink.com

1.2 Paper Contributions

The class of plants considered presents significant levels of uncertainty, thereby motivating the use of adaptive control techniques. The contribution of this paper consists in showing how the approach of [1] can be used to yield adaptive nonlinear model predictive control algorithms for distributed hyperbolic plants using the solar collector field (eq. 1) to illustrate this fact.

2 Orthogonal Collocation

In order to design the controller, the distributed parameter model (1) is first approximated by a lumped parameter model by using the Orthogonal Collocation Method (OCM) [5]. For this sake, it is assumed that the temperature along the pipe $T(z,t)$ is represented by the weighted sum

$$T(z,t) = \sum_{i=0}^{N+1} \varphi_i(z)\, T_i(t) \tag{2}$$

where the functions $\varphi_i(z)$ are Lagrange interpolation polynomials, orthogonal at the so called interior collocation points z_i for i=1,..., N and at the boundary collocation points z_0 and z_{N+1}.

Inserting (2) into (1) results in an ordinary differential equation verified by the time weights $T_i(t)$. By making $j = 1, \ldots, N+1$, *i. e.* by considering all the collocation points apart from the first, the PDE (1) is therefore approximated by n=N+1 ordinary differential equations (ODE), reading in matrix form

$$\dot{x} = -\frac{u}{L}\ (Ax + BT_0) + C\ \alpha\ R(t) \tag{3}$$

where $x = \begin{bmatrix} T_1\ T_2 \cdots T_{N+1} \end{bmatrix}^T$ with $T_i(t) \equiv T(z_i,\, t)$, the matrices A, B and C depend on $\varphi_j^{'}(z_i) \equiv \frac{d\varphi_j(z)}{dz}\ |_{z=z_i}$ and $T_0 = T(0,\, t)$ is the boundary condition. The use of a lumped parameter approximation relying on the finite fifference method (FDM) also leads to a model with the same structure as (3). However, experience shows that the same degree of approximation attained by the FDM with a grid having 100 points (corresponding to 100 states) is yielded by the OCM with just 5 points (corresponding to 5 states).

3 The Control Algorithm

The control algorithm comprises three parts: A receding horizon controller, a state observer and a parameter estimator (adaptation law).

3.1 The Receding Horizon Controller

The receding horizon controller (RHC) is now established such as to regulate the state around an equilibrium point, together with a condition which ensures stability in closed loop. For that sake, and inspired by [2], define the control law given by

$$u^* = \frac{\alpha\, R^*\, L}{r^* - T_0} \tag{4}$$

where r^* is the set-point of the outlet oil temperature x_n^*. Let x^* be the equilibrium state corresponding to u^* and consider the dynamics of the error $e = x - x^*$, given by

$$\dot{e} = \frac{-A}{L}\, u^*\, e + \frac{-A\, e - A\, x^* - B\, x_0}{L}\, \tilde{u} + C\, \alpha\, \tilde{R} \tag{5}$$

where $\tilde{u} = u - u^*$ and $\tilde{R} = R - R^*$. As shown by applying the Gronwall-Bellman inequality, for $\tilde{u} = 0$ and $\tilde{R} = 0$ the error dynamics is stable whenever the matrix $\bar{A} = -\frac{A}{L}$ is stable. It is not easy to prove a general result concerning the stability of the matrix $\overline{A}$ generated by the OCM and hence its stability must be checked for each application.

Define the RHC for the error dynamics by

$$\min_u J = \int_t^{t+H} \left(e^T(\tau) P e(\tau) + \rho\, \tilde{u}^2(\tau) \right) d\tau \tag{6}$$

where $\rho \geq 0$, $H > 0$, and subject to

$$\dot{e} = \frac{-A}{L}\, u_*\, e + \frac{-A\, e - A\, x^* - B\, T_0}{L}\, \tilde{u} \tag{7}$$

$$V_0(t+H) \geq V_{rhc}(t+H) \tag{8}$$

in which r^* is given by (4), $V_0(H) = e^T(H) \mid_{\tilde{u}=0} Pe(H) \mid_{\tilde{u}=0}$ and $V_{rhc}(H) = e^T(H) P e(H)$ where P is an arbitrary symmetric positive definite matrix.

The constraint (8) is equivalent to impose to the RHC that, at each iteration, the norm of the error at the end of the optimal sequence is bounded by the same norm resulting from the error when $u = u^*$. The existence of a control law, defined for $u = u^*$, which stabilizes the closed loop, allows to interpret V_0 as a Control Lyapunov Function [11] and, assuming complete plant knowledge, is a sufficient condition to ensure Global Asymptotic Stability of the loop closed by the RHC, when the controller is applied to (3) [8]. The constraint (3) is therefore a sufficient condition for stability. It has been observed in the simulations performed that this condition is active in the initial period, depending on the initial conditions. The rationale for minimizing (7) under the constraint (8) consists in increasing the performance while ensuring stability (by imposing the constraint).

3.2 State Observer

To (3) associate the state estimator with output error re-injection:

$$\dot{\hat{x}} = -\frac{u}{L}\left(A\hat{x} + Bx_0\right) + C\hat{\alpha}R(t) + K(t)D(x - \hat{x}) \qquad \hat{y} = D\hat{x} = \begin{bmatrix} 0 \; 0 \cdots 1 \end{bmatrix} \hat{x} \tag{9}$$

The error dynamics $e_1 := x - \hat{x}$ is given by:

$$\dot{e_1}(t) = A_e e_1 + C\tilde{\alpha}R(t) \tag{10}$$

where $A_e := -\frac{u}{L}A - K(t)D$ with $K(t)$ the observer gain.

3.3 Lyapunov Adaptation Law

Consider the candidate Lyapunov function

$$V_1 = e_1^T Q e_1 + \frac{1}{\gamma}\,\tilde{\alpha}^2 \tag{11}$$

where $\gamma > 0$ is a parameter, Q is a positive definite matrix and the parameter estimation error $\tilde{\alpha}$ is defined as $\tilde{\alpha}(t) := \alpha - \hat{\alpha}(t)$ where $\hat{\alpha}$ is the estimate of α. Its derivative is given by:

$$\dot{V}_1 = e_1^T(A_e^T Q + QA_e)e_1 + 2\tilde{\alpha}R(t)C^T Q e_1 + \frac{2}{\gamma}\tilde{\alpha}\,\dot{\tilde{\alpha}} \tag{12}$$

Stability holds if

$$-M(t) = (A_e^T Q + QA_e) < 0 \ \text{ and } \ \dot{\tilde{\alpha}} = -\gamma(CR(t))^T Q e_1$$

from which the following adaptation law follows:

$$\dot{\hat{\alpha}} = \gamma(CR(t))^T Q e_1 \tag{13}$$

It is possible to prove that $M(t) > 0$ is ensured by the following choice of the observer gain:

$$K(t) = \frac{u}{L}K_0 \tag{14}$$

with the matrix M_0 given by

$$M_0 = -[(-A - K_0 D)^T Q + Q(-A - K_0 D)] \tag{15}$$

that exists if the pair $(A,\, D)$ is observable and choosing K_0 such that $-A - K_0 D$ is stable. With this choice, and remarking that $u > 0$:

$$\dot{V}_1 = -e_1^T M(t) e_1 = -\frac{u}{L}e_1^T M e_1 \leq -\frac{u_{max}}{L}\, e_1^T M e_1 \leq 0 \tag{16}$$

and, by La Salle's Invariance Principle, it follows that $\lim_{t\to 0} e_1(t) = 0$. The parameter estimation error $\tilde{\alpha}(t)$ will tend to zero if u satisfies a persistency of excitation condition.

3.4 RHC Computational Algorithm

A computational efficient version of the the adaptive RHC is obtained by constraining u in (6) to be a staircase function with N_u steps $u = seq\{u_1,\, \ldots,\, u_{N_u}\}$ and using x and α replaced by their estimates. The estimate of u^* is given by:

$$\hat{u}^* = \frac{\hat{\alpha}(t)\; R(t)\; L}{r(t) - T_0(t)} \tag{17}$$

Here, $\hat{x}$ and $\hat{\alpha}$ are obtained using the state estimator (9), the adaptation law (13), $u(\bar{t})$ is a sequence of step functions with amplitude u_i $(i = 1, \ldots, N_u)$ and duration $\frac{T}{N_u}$. The variable $\bar{t}$ represents time during the minimization horizon $\bar{t} \in [0, H[$. The initial condition $\hat{\alpha}(0)$ for the estimate of α is provided by the designer.

Once the minimization result $u(\bar{t})$ is obtained, according to a receding horizon scheme u_1 is applied to the plant at $t + \delta$ and the whole process is restarted, δ being an interval of time which is at least the time needed to compute the solution. It is assumed that δ is much smaller than the sampling period. There are classes of plants, such as switched nonlinear systems [7], for which the choice of δ has a bearing on the stability properties of the predictive controller. In the case at hand, no such problems were found.

The minimization is always feasible since $\tilde{u} = 0$ (corresponding to $u = u^*$) preserves the closed loop stability while satisfying the constraint (8) with $V_0 = V_{rhc}$.

4 Simulation Results

Simulation results of the proposed RHC have been performed in a detailed model of the solar field obtained from first physical principles, according to [3], and calibrated with plant data. Experimental sequences for $R(t)$ and $T_0(t)$ are used. The reduced model (3) uses 3 interior collocation points $z = [0.113 \;\; 0.500 \;\; 0.887]$ and leads to a matrix $-A/L$ whose eigenvalues have all strictly negative part.

In order to configure the controller, the following parameter choices have been made: $\gamma = 1 \times 10^5$, $K_0 = [15\,15\,15\,15]$, $\rho = 1.5 \times 10^{-10}$, $H = 180\,s$ and $N_u = 26$. Figures (1) through (3) show the results. Fig. (1) shows the time evolution of the reference (easily recognizable by being a sequence of steps) and of the temperature

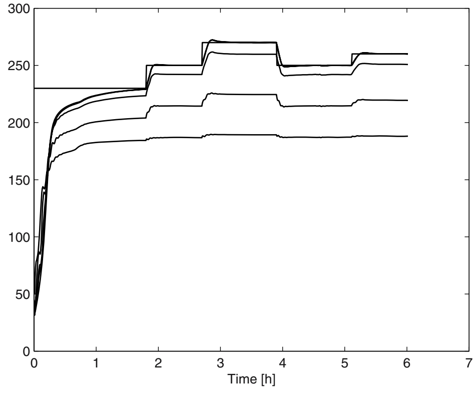


Fig. 1. Solar field with RHC. Outlet oil temperature and reference and temperature estimates at the collocation points $[{}^oC]$.

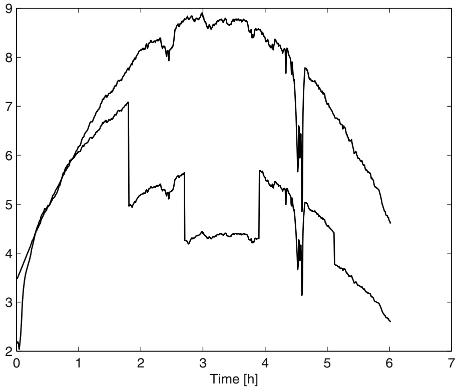


Fig. 2. Solar field with RHC. Radiation (disturbance – above) ($\times 10^{-2}$) $[W/m^2]$ and oil flow (manipulated variable – below) $[l/s]$.

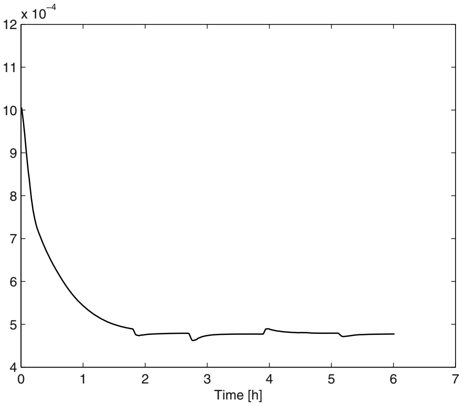


Fig. 3. Solar field with RHC. Mirror efficiency estimate, $\hat{\alpha}$.

at the collocation points. The highest temperature at the collocation points corresponds to the oil outlet temperature (variable to be controlled). As it is seen, after an initial transient corresponding to the start-up of the field in which the temperature raises progressively (lasting for about 1 hour), the plant output tracks the reference with without static error , with a small overshoot, and with a raise time of about 10 minutes. When the oil temperature is to be raised to track a positive reference step, the oil flow suddenly decreases. Otherwise, it changes to compensate for the daily evolution of solar radiation (lower curve in fig. 2). As seen in fig. 2), at about 4h30, a strong disturbance caused by a sudden drop of radiation acted on the system. The controller reduced the oil flow so that the energy accumulated by unit volume of the oil leaving the pipe remains constant, an action that resulted in an almost perfect rejection of the disturbance.

Fig. 3 shows the mirror efficiency estimate, $\hat{\alpha}(t)$ as a function of time. This variable starts from the initial estimate provided by the designer and converges to an almost constant value after 2 hours. There are only minor changes of $\hat{\alpha}(t)$ induced by the steps in the reference.

5 Discussion and Conclusions

Adaptive nonlinear receding horizon control of plants described by hyperbolic PDEs has been addressed using a novel approach which combines the Orthogonal Collocation Method, Receding Horizon Control and parameter estimation. The approach has been tested in a detailed model of a distributed collector solar field.

The interest of the work reported is twofold: First, it provides an approach which can be used in a class of plants of technological interest. Furthermore, it presents a case study on adaptive nonlinear receding horizon control, illustrating how recent algorithms may be applied to distributed parameter plants with transport phenomena.

References

[1] Adetola, V. and M. Guay, “Adaptive receding horizon control of nonlinear systems´´, *Proc. 6th IFAC Symp. on Nonlinear Control Systems – NOLCOS 2004*, Stuttgart Germany, 1055-1060, (2004).

[2] Barão M., Lemos J. M. and Silva, R. N., ”Reduced complexity adaptative nonlinear control of a distributed collector solar field”, *J. of Process Control*, **12**, 131-141, (2002).

[3] Camacho, E., M. Berenguel and F. Rubio, *Advanced Control of Solar Plants*, New York: Springer Verlag (1997).

[4] Christofides, P. D., *Nonlinear and Robust Control of PDE Systems*, Birkhauser, (2001).

[5] Dochain, D., J. P. Babary and Tali-Maamar, “Modeling and adaptive control of nonlinear distributed parameter bioreactors via orthogonal collocation´´, *Automatica*, **28**, 873-883, (1992).

[6] Dubljevic, S., P. Mhaskar, N. H. El-Farra and P. D. Christofides, ” Predictive Control of Transport-Reactioon Processes”, *Comp. and Chem. Eng.*, **29**, 2335-2345, (2005).

[7] Mhaskar, P., N. H. El-Farra and Pd. D. Christofides (2005). ”Predictive Control of Switched Nonlinear Systems With Scheduled Mode Transitions”, *IEEE Trans. Autom. Control*, **50**, 1670-1680, (2005).

[8] Primbs, J. A., V. Nevistić and J. Doyle, *A Receding Generalization of Pointwise Min-Norm Controllers*, citeseer.nj.nec.com, (1998).

[9] Shang, H., J. F. Forbes and M. Guay, ”Model Predictive Control for Quasilinear Hyperbolic Distributed Parameter Systems”, *Ind. Eng. Chem. Res.*, **43**, 2140-2149, (2004).

[10] Silva, R. N., J. M. Lemos and L. M. Rato, “Variable sampling adaptive control of a distributed collector solar field”, *IEEE Trans. Control Syst. Tech.*, **11**, 765-772, (2003).

[11] Sontag, E., *Mathematical Control Theory* Springer-Verlag, 2nd Ed., (1998).

A Minimum-Time Optimal Recharging Controller for High Pressure Gas Storage Systems

Kenneth R. Muske, Amanda E. Witmer, and Randy D. Weinstein

Department of Chemical Engineering, Villanova University, Villanova, PA, 19085, USA
kenneth.muske@villanova.edu

Summary. A minimum-time optimal recharging control strategy for high pressure gas storage tank systems is described in this work. The goal of the nonlinear model-based controller is to refill the tank in minimum time with a two-component gas mixture of specified composition subject to hard constraints on the component flow rates, tank temperature, and tank pressure. The nonlinearity in this system arises from the nonideal behavior of the gas at high pressure. The singular minimum-time optimal control law can not be reliably implemented in the target application due to a lack of sensors. Minimum-time optimal control is therefore approximated by a nonlinear model-based constraint controller. In order to account for the uncertainty in the unmeasured state of the storage tank, the state sensitivities to the control and process measurements are propagated along with the state to obtain a state variance estimate. When the variance of the state exceeds a maximum threshold, the constraint control algorithm automatically degrades into a fail-safe operation.

1 Introduction

The gas storage tank recharging system, shown in Figure 1, consists of high pressure sources for each component that supply the gas to the storage tanks. A source pressure sensor and mass flow controller are available for each component. A pressure sensor upstream of the discharge nozzle into the storage tank and an ambient temperature sensor are the only other process measurements. There are no sensors in the storage tank itself because of economic and maintenance reasons. It is less expensive to instrument the supply line from the tank than to replicate and maintain these instruments in each tank.

The controlled variables for this system are the final mass, or total moles, and composition of the gas in the storage tank. The manipulated variables are the setpoints to the component mass flow controllers. The system may be operated by either maintaining the feed gas at the desired composition during the entire refilling process or allowing the feed gas composition to vary with the desired composition being achieved when the tank is refilled. In this work, the first operating philosophy will be adopted. The advantage of the first approach is that the gas in the tank is always at the desired composition. If the refilling process must be terminated for any reason, the storage tank will still have the

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 443–453, 2007.
springerlink.com

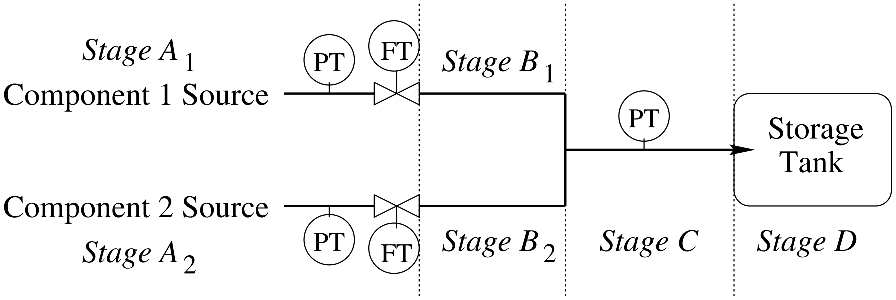


Fig. 1. Gas storage tank recharging system

correct composition. The disadvantage of this approach is that the controller can not use the extra degree of freedom to take advantage of any differences in the component gas properties when refilling the tank.

The objective of the control system is to safely fill the gas storage tank in minimum time with a specified amount of a two-component gas mixture subject to pressure and temperature constraints. Of particular concern in this process is the Joule–Thompson behavior of the gas components comprising the mixture. For systems with a positive Joule–Thompson coefficient, the gas mixture will cool as it expands from source pressure into the storage tank. In this case, the maximum pressure constraint must be lowered to account for the future increase in pressure as the system reaches ambient temperature. For systems with a negative Joule-Thompson coefficient, the gas mixture temperature will rise as the storage tank is filled. In this case, the rate of temperature rise must be controlled to reach the desired final amount without exceeding the maximum temperature limit of the storage tank and delivery system. Over the pressure ranges of interest, however, some gas components can exhibit significant changes in the Joule–Thompson coefficient including sign changes.

2 Thermodynamic Model

The recharging system model is based on the thermodynamic relationships for the transition between each of the four stages shown in Figure 1. The first two stage transitions, A → B and B → C, are modeled as isoenthalpic transitions with no gas accumulation. The result is a series of steady-state algebraic equations relating the temperature, pressure, and density at each stage. The last stage transition, C → D, is modeled using an isoentropic transition through the nozzle and an unsteady-state energy balance over the storage tank. The result is a differential-algebraic system where the algebraic equations arise from the isoentropic transition and the equation of state.

2.1 Equation of State

The following two-coefficient virial equation of state for a binary mixture [1]

$$Z = \frac{P}{RT\rho} = 1 + B_{\text{mix}}\rho + C_{\text{mix}}\rho^2 \tag{1}$$

$$B_{\text{mix}} = b_1(T)x_1^2 + 2b_{12}(T)x_1x_2 + b_2(T)x_2^2 \tag{2}$$

$$C_{\text{mix}} = c_1(T)x_1^3 + 3c_{112}(T)x_1^2x_2 + 3c_{122}(T)x_1x_2^2 + c_2(T)x_2^3 \tag{3}$$

is used in this work where x_1, x_2 are the component mole fractions and B_{mix} and C_{mix} are, in general, functions of temperature. Because the composition of the inlet gas mixture is maintained at the desired target composition, x_1 and x_2 are constant at this composition in stages C and D. The equation of state for the single component streams in stages A_1 and B_1 is obtained by setting $x_1 = 1$ and $x_2 = 0$ in Eqs. 2 and 3. The equation of state for stages A_2 and B_2 is handled in a similar manner by setting $x_1 = 0$ and $x_2 = 1$.

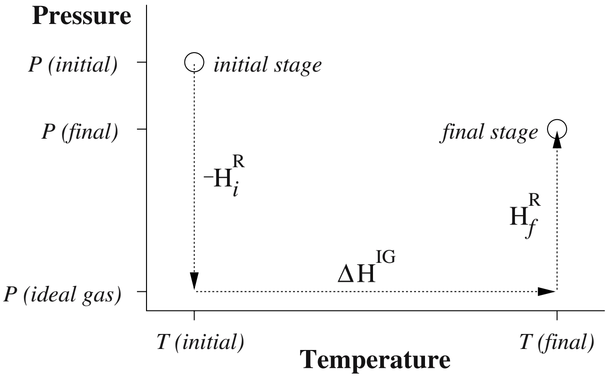


Fig. 2. Isoenthalpic stage transition model

2.2 Isoenthalpic Transition

The first two transitions are described by the path shown in Figure 2 for both the individual pure components and the gas mixture. The resulting equation for the transition from an initial stage i to the next stage f becomes

$$\Delta H_{i\to f} = 0 = -H_i^{\text{R}} + \Delta H^{\text{IG}} + H_f^{\text{R}} \tag{4}$$

where H_i^{R} is the residual enthalpy between the gas at the initial stage and the gas at ideal conditions at the initial temperature, ΔH^{IG} is the change in enthalpy of the gas at ideal conditions between the final and initial temperatures, and H_f^{R} is the residual enthalpy between the gas at the next stage and the gas at ideal conditions at the final temperature. The residual enthalpy is

$$H^{\mathrm{R}} = H - H^{\mathrm{IG}} = -RT^2 \int_0^{\rho_g} \frac{\partial Z(\rho)}{\partial T} \frac{d\rho}{\rho} + RT(Z-1) \tag{5}$$

where H^{IG} is the enthalpy of the gas at ideal conditions (the limit as pressure, or density, goes to zero at the actual temperature), Z is the compressibility, and ρ_g is the gas density [1]. The temperature change and component mixing are done at ideal conditions because the enthalpy of mixing is zero and ideal gas heat capacities, which are only a function of temperature, can be used.

$$\Delta H^{\mathrm{IG}} = \Delta H^{\mathrm{IG}}_{\mathrm{mix}} + \int_{T_i}^{T_f} C_p \, dT \; = \; \int_{T_i}^{T_f} C_p \, dT \tag{6}$$

The ideal gas heat capacity is taken as an empirical function of temperature.

$$C_p = \sum_{j=1}^{2} x_j C_{p_j} = R \sum_{j=1}^{2} x_j \left(\alpha_j + \beta_j T + \gamma_j T^2 + \epsilon_j T^{-2} \right) \tag{7}$$

2.3 Isoentropic Transition

If we assume a perfect nozzle, and therefore isoentropic flow, the transition from stage C to stage D can be described by the transition $\Delta S_{C \to D} = 0$ or by the relationship for isentropic adiabatic flow through a nozzle. The complexity arising from a nonideal gas with sonic flow for some fraction of the time suggests abandoning the flow equation in favor of the entropy relationship

$$\Delta S_{C \to D} = 0 = -S^{\mathrm{R}}_C + \Delta S^{\mathrm{IG}} + S^{\mathrm{R}}_D \tag{8}$$

$$S^{\mathrm{R}} = S - S^{\mathrm{IG}} = R \left[\ln Z - T \int_0^{\rho_g} \left(\frac{\partial Z(\rho)}{\partial T} - \frac{Z(\rho) - 1}{T} \right) \frac{d\rho}{\rho} \right] \tag{9}$$

$$\Delta S^{\mathrm{IG}} = \int_{T_C}^{T_D} C_p \frac{dT}{T} - \int_{P_C}^{P_D} R \frac{dP}{P} \tag{10}$$

where S^{R} is the residual entropy and ΔS^{IG} is the change in enthalpy at ideal conditions [1]. The result is analogous to the enthalpy relationships in Eqs. 4–6 except there is no mixing term because the stage compositions are the same.

2.4 Unsteady-State Energy Balance

The unsteady-state energy balance for the gas mixture in the storage tank is

$$n_D \left[\frac{dH_D}{dt} + \frac{d}{dt} \left(\frac{P_D}{\rho_D} \right) \right] - \dot{n}_D \left[\Delta H_{C \to D} + \frac{P_D}{\rho_D} \right] - \dot{Q}_D = 0 \tag{11}$$

where n_D is the total moles of gas in the storage tank, $\rho_D = n_D / V_D$ is the density of the gas in the tank, V_D is the tank volume, H_D is the enthalpy of the gas mixture in the storage tank, $\Delta H_{C\rightarrow D}$ is the change in enthalpy between the gas mixture upstream of the nozzle and the gas in the storage tank, $\dot{Q}_D$ is the heat transfered from the storage tank to the surroundings, and the rate of change of the moles of gas in the tank is determined by the control $\dot{n}_D = u$. Because the composition of the inlet gas mixture is maintained at the desired target composition, the total molar flow rate is the single control.

The rate of change of the enthalpy of the gas inside the storage tank is

$$\frac{dH_D}{dt} = \frac{\partial H_D}{\partial T}\frac{dT_D}{dt} + \frac{\partial H_D}{\partial \rho}\frac{d\rho_D}{dt} = \left(C_{p_D} + \frac{\partial H_D^{\mathrm{R}}}{\partial T}\right)\frac{dT_D}{dt} + \left(\frac{\rho_D}{n_D}\frac{\partial H_D^{\mathrm{R}}}{\partial \rho}\right)u \quad (12)$$

where the residual enthalpy H_D^{R} is as defined in Eq. 5. The rate of change of P_D/ρ_D can be expressed as a function of the temperature change as follows.

$$\frac{d}{dt}\left(\frac{P_D}{\rho_D}\right) = \frac{d}{dt}(RZ_DT_D) = R\left(Z_D + T_D\frac{\partial Z_D}{\partial T}\right)\frac{dT_D}{dt} + R\left(\frac{T_D\rho_D}{n_D}\frac{\partial Z_D}{\partial \rho}\right)u \quad (13)$$

2.5 Differential-Algebraic System Model

The preceding thermodynamic relationships result in the following differential algebraic modeling equations for the system where we will assume that $\dot{Q} = 0$.

$$H_{B_1}^{\mathrm{R}} + \int_{T_{A_1}}^{T_{B_1}} C_{p_1}\,dT - H_{A_1}^{\mathrm{R}} = 0 \quad (14)$$

$$P_{B_1} - RT_{B_1}\left(\rho_{B_1} + B_{\mathrm{mix}}\rho_{B_1}^2 + C_{\mathrm{mix}}\rho_{B_1}^3\right) = 0 \quad (15)$$

$$H_{B_2}^{\mathrm{R}} + \int_{T_{A_2}}^{T_{B_2}} C_{p_2}\,dT - H_{A_2}^{\mathrm{R}} = 0 \quad (16)$$

$$P_{B_2} - RT_{B_2}\left(\rho_{B_2} + B_{\mathrm{mix}}\rho_{B_2}^2 + C_{\mathrm{mix}}\rho_{B_2}^3\right) = 0 \quad (17)$$

$$H_C^{\mathrm{R}} + x_1\left(\int_{T_{B_1}}^{T_C} C_{p_1}\,dT - H_{B_1}^{\mathrm{R}}\right) + x_2\left(\int_{T_{B_2}}^{T_C} C_{p_2}\,dT - H_{B_2}^{\mathrm{R}}\right) = 0 \quad (18)$$

$$P_C - RT_C\left(\rho_C + B_{\mathrm{mix}}\rho_C^2 + C_{\mathrm{mix}}\rho_C^3\right) = 0 \quad (19)$$

$$S_D^{\mathrm{R}} + \int_{T_C}^{T_D} C_{p_D}\,\frac{dT}{T} - \int_{P_C}^{P_D} R\,\frac{dP}{P} - S_C^{\mathrm{R}} = 0 \quad (20)$$

$$n_D - \rho_D V_D = 0 \quad (21)$$

$$P_D - RT_D\left(\rho_D + B_{\mathrm{mix}}\rho_D^2 + C_{\mathrm{mix}}\rho_D^3\right) = 0 \quad (22)$$

$$\frac{H_D^{\mathrm{R}} + \int_{T_C}^{T_D} C_{p_D}\, dT - H_C^{\mathrm{R}} + \frac{P_D}{\rho_D} - \left(\frac{\partial H_D^{\mathrm{R}}}{\partial \rho} + RT_D \frac{\partial Z_D}{\partial \rho}\right)\rho_D}{n_D\left(C_{p_D} + \frac{\partial H_D^{\mathrm{R}}}{\partial T} + R\left(Z_D + T_D \frac{\partial Z_D}{\partial T}\right)\right)} u = \dot{T}_D \tag{23}$$

$$u = \dot{n}_D \tag{24}$$

There are two differential and nine algebraic equations in Eqs. 14–24 for the thirteen unknowns: P, T, & ρ for stages B_1, B_2, C, & D and n_D. Making the assumption that $P_{B_1} = P_{B_2} = P_C$ reduces the number of unknowns to eleven.

3 Minimum-Time Optimal Control

The optimization problem for the minimum-time optimal controller is

$$\min_{u(t)} \int_{t=0}^{t_f} 1\, dt \quad \text{Subject to:} \quad \begin{array}{l} n_D(t_f) = n_D^{\star} \\ \dot{x} = f(x)u \\ g(x) = 0 \\ h_x(x) \leq 0 \\ h_u(u) \leq 0 \end{array} \tag{25}$$

where x is the system state, u is the control, $n_D^{\star}$ is the desired final moles of gas in the storage tank, $f(x)u$ represents the differential equations in Eqs. 23–24, $g(x)$ represents the algebraic equations in Eqs. 14–22, $h_x(x)$ represents the tank temperature and pressure hard constraints, and $h_u(u)$ represents the component gas flow rate hard constraints. As is common for minimum-time problems, the result is a singular optimal control problem. The optimal control trajectory is either at a constraint, from the minimum principle, or along an optimal singular arc that satisfies the Euler–Lagrange equations [2].

Because the enthalpy of the gas in the tank is a state function, the isoentropic assumption for the stage C to stage D transition neglects losses in the inlet line, and the system is assumed adiabatic, the state of the storage tank determined from the differential-algebraic system model presented in Eqs. 14–24 is path independent. Therefore, all control profiles result in the same final tank state for a given final moles of gas $n_D^{\star}$. If a steady-state analysis determines that a tank constraint is violated at this target, then there is no feasible control profile $u(t)$ that satisfies both the terminal equality constraint $n_D(t_f) = n_D^{\star}$ and the tank state inequality constraints $h_x(x) \leq 0$ for the minimum-time optimal control problem in Eq. 25. In this case, an alternative feasible optimal control approach would be to construct a singular optimal controller that achieves the most limiting tank constraint in minimum time.

An excellent review of solution methods for singular optimal control problems arising from batch processes is presented in [3]. The use of process measurements to improve the robustness of optimal control to model mismatch and unmeasured disturbances is discussed in [4]. The application of these techniques to the controller in this work, however, is restricted by the lack of process measurements.

With only a single pressure measurement to estimate eleven states, state feedback is either impossible (if the integrating state is not detectable) or highly unreliable (because of the variance in the state estimates). Open-loop optimal control approaches, discussed in [5], are inappropriate in this application due to the consequences of a constraint violation. For these reasons, a model-based dynamic constraint controller is proposed.

4 Model-Based Dynamic Constraint Control

We develop a model predictive dynamic constraint controller to approximate the minimum-time optimal recharging controller presented in the previous section. The single process measurement, inlet line pressure, is integrated into the constraint controller by using this measurement to eliminate the isoentropic relationship $\Delta S_{C \rightarrow D} = 0$ (Eq. 20) from the model. The advantage of this integration is that the isoentropic transition assumption is removed from the model which also removes the path independence of the tank state. The disadvantage of this approach is that there is no output feedback correction to the tank state. However, it is unlikely that state estimation based on the single pressure measurement would result in any significant improvement in the model predicted tank state. The uncertainty in the tank state prediction can be monitored by estimating the variance as outlined in the sequel.

The model-based dynamic constraint controller attempts to drive the system to the most limiting constraint in minimum time while relaxing the terminal state equality constraint $n_D(t_f) = n_D^{\star}$ if necessary. The dynamic constraint controller is a model predictive version of the active constraint tracking controller in [6]. This control structure is motivated by the solution to the feasible minimum-time optimal control problem in the previous section which specifies that the system should be operated at an active constraint during the entire refilling process. We note that if the irreversible losses in the system are negligible, then open-loop optimal control, closed-loop model predictive control, and closed-loop dynamic constraint control should all result in this same active constraint tracking input trajectory. If irreversible losses are significant, then constraint control may not be a good approximation to the optimal input trajectory. Preliminary experimental evidence suggests the former case [7].

4.1 Constraint Controller

Constraint prediction is performed by solving the DAE system in Eqs. 14–19, 21–24 from the initial time to the current time using the past control and inlet pressure measurement trajectories. The initial state of the system is available from the ambient temperature measurement and the initial inlet line pressure which is the same as the tank pressure at zero flow. The future state predictions are then obtained by assuming that the control and inlet line pressure remain constant at their current values until the tank is refilled. The target final moles of gas in the tank is determined at each sample period k by

$$n_D^{\mathrm{F}}(k) = \min \left[n_D^{\star}, \, n|_{T_D = T_D^{\max}}(k), \, n|_{T_D = T_D^{\min}}(k), \, n|_{P_D = P_D^{\max}}(k) \right] \tag{26}$$

where $n_D^{\mathrm{F}}(k)$ is the current target final moles of gas at sample period k, $n_D^{\star}$ is the desired final moles of gas, $n|_{T_D = T_D^{\max}}$ is the current predicted moles of gas such that the tank temperature reaches its maximum constraint limit, $n|_{T_D = T_D^{\min}}$ is the current predicted moles of gas such that the tank temperature reaches its minimum constraint limit, $n|_{P_D = P_D^{\max}}$ is the current predicted moles of gas such that the tank pressure reaches its maximum constraint limit, and the *min* operator selects the most limiting model-predicted constraint. The current predicted moles of gas required to reach a tank constraint is determined directly from the predicted future tank state profile. The length of the prediction horizon is always the time required to obtain $n_D^{\star}$ moles of gas in the tank. If a constraint violation is not predicted within this horizon, it is not considered by the *min* operator in Eq. 26. A first-order approximation to the control move required to achieve the most limiting constraint in minimum time is then determined from the current predicted tank state and target by

$$u(k) = \min \left[u^{\max}, \frac{n_D^{\mathrm{F}}(k) - n_D(k)}{\Delta t} \right] \tag{27}$$

where $u(k)$ is the current input, $u^{\max}$ is the maximum flow rate constraint, and $n_D(k)$ is the current prediction of the moles of gas in the storage tank.

This dynamic constraint prediction is computed at every sample period after the initial start-up phase. The start up is carried out at a minimum safe gas flow rate to ensure that the system is operating properly. The constraint prediction is updated by the incorporation of the most recent inlet pressure measurement at the current sample time. We note that on-line optimization is not required to determine the control input because of the assumption that the optimal operation is at an active constraint (motivated by the minimum-time optimal control trajectory). Because the DAE system and the state sensitivities, required for the uncertainty estimate described in the next section, can be computed very quickly, the sample period Δt is not limited by computational issues as is often the case for nonlinear predictive control implementations.

4.2 Fail-Safe Operation

Because there is no direct measurement of the actual tank state, some mechanism to monitor the uncertainty in this state estimate is required for the safe implementation of the proposed controller. Linear approximations to the variance of the tank state can be obtained from the first-order sensitivities [8] between the tank state and the control and inlet line pressure as follows

$$\sigma_{x,P_C}^2 = \left(\frac{\partial x}{\partial P_C} \right)^2 \sigma_{P_C}^2, \quad \sigma_{x,u}^2 = \left(\frac{\partial x}{\partial u} \right)^2 \sigma_u^2, \quad F = \frac{\sigma_{x,P_C}^2}{\lambda + \sigma_{x,u}^2} > F_\alpha \tag{28}$$

where x is the tank temperature or pressure, the partials are the sensitivities, σ^2_{x,P_C} is the estimate using the pressure measurement variance $\sigma^2_{P_C}$, $\sigma^2_{x,u}$ is the estimate using the control variance σ^2_u, F_α is the F statistic at a confidence level α, λ is a tuning parameter to account for measurement noise and normal variation in the inlet line pressure, and $F > F_\alpha$ implies $\sigma^2_{x,P_C} > \sigma^2_{x,u}$ [9]. If the variance estimated from the inlet line pressure measurement exceeds that estimated from the control, the constraint control is terminated to a fail-safe operation. This operation can either shut off the gas flow completely, where the tank pressure could then be determined by the inlet line pressure sensor, or can reduce the gas flow to a predetermined minimum safe value.

5 Example

The control strategy is illustrated using a nitrogen–helium gas mixture. We choose this system because the sign of the Joule-Thompson coefficient is different for each component; negative for helium and positive for nitrogen. There are also significant differences in the intermolecular potentials leading to large deviations from ideal behavior. The coefficients in Eqs. 2–3 are affine functions of temperature taken from [10]. We consider a 1/4 He/N_2 gas blend in a 100 lit storage tank where the component source pressures are both 175 bar, the ambient temperature is 300 K, and the initial tank pressure is 10 bar. Figure 3 presents the predicted tank temperature profiles for a series of flow rates which clearly demonstrate the nonideal behavior and path independence of the tank state. The predicted pressure profiles behave in a similar manner.

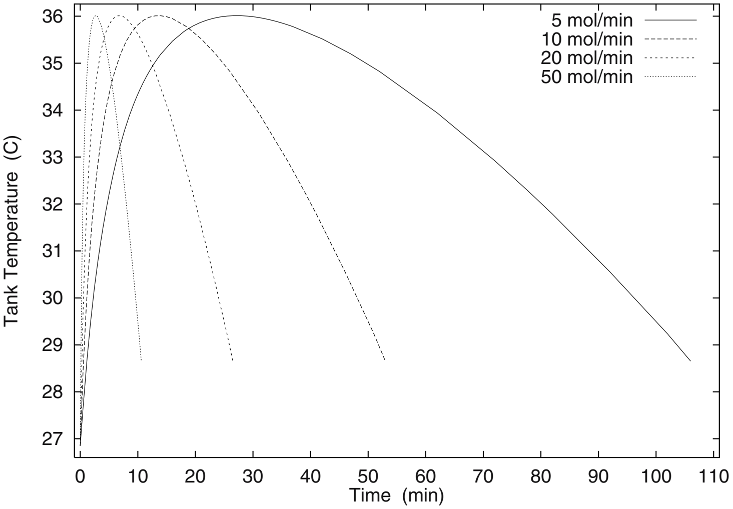


Fig. 3. Predicted tank temperature profiles

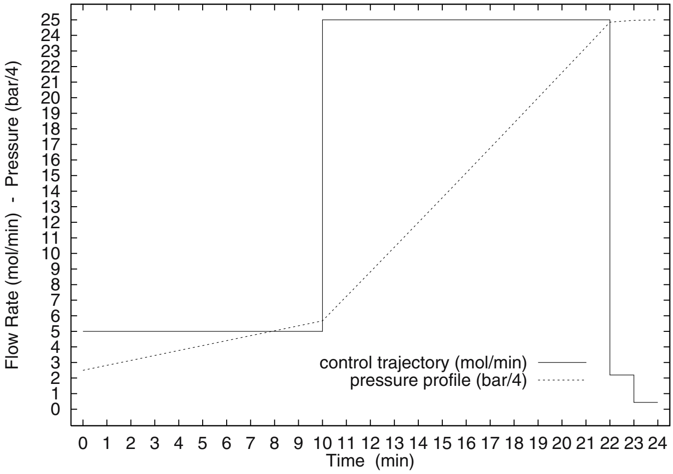


Fig. 4. Control trajectory and simulated tank pressure profile

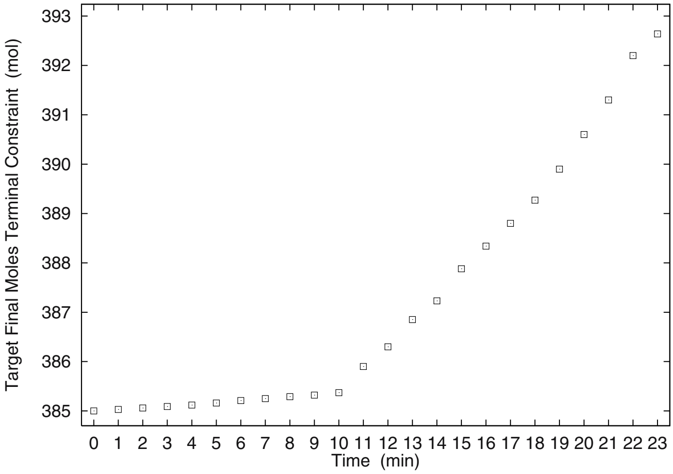


Fig. 5. Target final moles of gas in the tank

We consider a desired value of $n_D^\star = 400$ mol and a maximum operating constraint of $P_D^{\max} = 100$ bar for the tank pressure. At this pressure constraint, only 385 moles of gas can be stored in the tank assuming no thermodynamic losses. Therefore, the maximum tank pressure is the most limiting constraint in this example. The inlet line pressure is simulated using an ideal gas nozzle flow equation with energy loss. The maximum gas flow rate is 25 mol/min. The sample period is one minute. Figure 4 presents the dynamic constraint control trajectory $u(k)$ determined from Eq. 27 and the actual tank pressure. The first ten minutes in this example represents the start-up phase. Figure 5 presents the target final moles of gas, $n_D^{\mathrm{F}}(k)$ in Eq. 26, at each sample time.

The control is initially set to the start-up phase flow rate of 5 mol/min in this example and then is brought to its maximum value when the constraint controller is initiated at 10 min. The control is reduced from its maximum constraint at the end of the recharge when a maximum pressure constraint violation is predicted. The target final moles of gas is determined at each sample period as the amount that results in the predicted tank pressure reaching its maximum constraint. The corrections to this target become larger as the flow rate increases and the tank is filled because the simulated energy losses in the nozzle become larger. We note that measurement noise and initial condition error is not present in this example.

6 Conclusions and Future Work

We have presented a dynamic constraint control approximation to the singular minimum-time optimal control law for recharging high pressure gas storage tanks. This development neglected heat transfer to the storage tank and the surroundings. Although the thermal capacity of the storage tank can reasonably be neglected in the industrial system, heat transfer to the surroundings can become significant with larger changes in both tank temperature and pressure. Future work includes the addition of a thermal model to account for the effect of heat transfer in the thermodynamic model of the system.

Acknowledgments

Support for this work from Air Products and Chemicals Company is gratefully acknowledged.

References

[1] Smith J, Van Ness H, Abbott M (2005) Chemical engineering thermodynamics. 7th ed. McGraw–Hill, New York

[2] Bryson A, Ho Y (1975) Applied optimal control. Taylor & Francis, Levittown

[3] Srinivasan B, Palanki S, Bonvin D (2002) Comput Chem Eng 27(1):1–26

[4] Srinivasan B, Bonvin D, Visser E, Palanki S (2002) Comput Chem Eng 27(1):27–44

[5] Muske K, Badlani M, Dell'Orco P, Brum J (2004) Chem Eng Sci 59(6):1167-1180

[6] Bonvin D, Srinivasan B (2003) ISA Trans 42(1):123–134

[7] Witmer A, Muske K, Weinstein R, Simeone M (2007) Proceedings of the 2007 American Control Conference

[8] Caracotsios M, Stewart W (1985) Comput Chem Eng 9(4):359–365

[9] Christensen R (1996) Analysis of variance, design and regression. Chapman & Hall, London

[10] Zhang W, Schouten J, Hinze H (1992) J Chem Eng Data 37(2):114–119

Robust NMPC for a Benchmark Fed-Batch Reactor with Runaway Conditions

Peter Kühl[1], Moritz Diehl[1], Aleksandra Milewska[2], Eugeniusz Molga[2], and Hans Georg Bock[1]

[1] IWR, University of Heidelberg, Germany
{peter.kuehl,m.diehl,bock}@iwr.uni-heidelberg.de
[2] ICHIP, Warsaw University of Technology, Poland
{milewska,molga}@ichip.pw.edu.pl

Summary. A nonlinear model predictive control (NMPC) formulation is used to prevent an exothermic fed-batch chemical reactor from thermal runaways even in the case of total cooling failure. Detailed modeling of the reaction kinetics and insight into the process dynamics led to the formulation of a suitable optimization problem with safety constraints which is then successively solved within the NMPC scheme. Although NMPC control-loops can exhibit a certain degree of inherent robustness, an explicit consideration of process uncertainties is preferable not only for safety reasons. This is approached by reformulating the open-loop optimization problem as a min-max problem. This corresponds to a worst-case approach and leads to even more cautious control moves of the NMPC in the presence of uncertain process parameters. All results are demonstrated in simulations for the esterification process of 2-butyl.

1 Introduction

Known from extreme accidents like in Seveso, Italy (1976), thermal runaways occur more frequently in smaller fine chemical reactors with high heat release potential. They lead to annoying production losses and equipment damages [2, 18]. To reduce difficulties, potentially dangerous processes are commonly run in fed-batch mode, yet with the most simple feeding strategy of constant dosing rates. The advent of detailed models of batch reactors including complicated reaction schemes can aid the development of more sophisticated feed strategies. A suitable framework for this goal is nonlinear model predictive control (NMPC). NMPC has the appealing attribute that constraints on states and controls are taken into account explicitly. In the fed-batch reactor case, a dosing rate profile delivered by NMPC will steer the process closer to the limits. At the same time, due to the predictive nature of NMPC, the system will be able to avoid runaway conditions.

Plant-model-mismatch in the form of uncertain parameters and initial values require NMPC schemes to be robust. Theoretical considerations have shown that NMPC controllers can inherently possess a certain degree of robustness [5]. For safety-critical processes, an explicit consideration of uncertainty is

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 455–464, 2007.
springerlink.com

desirable. Some robust NMPC schemes have been proposed based on game-theoretic ideas [4], min-max formulations [11], H_∞-control [14], and chance constrained programming [13]. These methods tend to be computationally too complex for practical applications.

In this paper, the optimization problem is reformulated using a min-max approach similar to [15], extending the formulation to path constraints. The resulting semi-infinite control problem is then approximated to obtain a numerically tractable form ([8, 10]). To demonstrate this approach, it is applied to control the exothermic esterification of 2-butanol. While the main task of the state-feedback controller is to quickly finish the batch, safety constraints have to be met at any time of the process.

This paper is organized as follows: In Section 2, the fed-batch process and its mathematical model are described. The open-loop optimization problem needed for the NMPC scheme is developed in Section 3. Particular care is taken of the formulation of a safety constraint that prevents runaways even in the case of a total cooling failure. This section ends with simulation results for the process under nominal NMPC. To take parameter uncertainties directly into account, the approximated min-max formulation is stated in Section 4. It leads to a more robust version of the NMPC as is demonstrated via simulations at the end of the section. The paper concludes with further discussions and an outlook on future research directions.

2 Example: Exothermic Esterification of 2-Butanol

The esterification of 2-butanol (B) with propionic anhydride (A) to 2-butyl propionate (D) and propionic acid (C) is a mildly exothermic reaction that allows one to study runaway situations in a lab. The reaction is catalyzed by two forms of a catalyst, (K_1 and K_2), while the first degrades into the latter in a side reaction:

$$\begin{aligned} \mathrm{A} + \mathrm{B} &\overset{\mathrm{K_1,K_2}}{\longrightarrow} \mathrm{C} + \mathrm{D} \\ \mathrm{K_1} &\longrightarrow \mathrm{K_2} \end{aligned}$$

The reaction is assumed to take place in a fed-batch reactor under isoperibolic conditions (constant jacket temperature). A similar, strongly simplified reaction has also been considered in [17]. The reaction system is modeled by a set of differential equations based on mass and energy balances. Note that the model is only valid for a total amount of A added which is smaller or equal to B. The full model reads as follows:

$$\begin{aligned} \dot{n}_A &= u - r\,V, & \dot{n}_B &= -r\,V, \\ \dot{n}_C &= r\,V, & \dot{n}_{K_1} &= -r_k\,V, \\ (C_{p,I} + C_p)\,\dot{T}_R &= r\,H\,V - q_{\mathrm{dil}} - U\,\Omega\,(T_R - T_J) \\ &\quad -\alpha(T_R - T_a) - u\,c_{p,A}\,(T_R - T_d), \end{aligned} \tag{1}$$

with n_i being the molar amount of component i, V the volume, T_R, T_J, T_a and T_d the reactor, jacket, ambient and dosing temperatures, respectively. r and r_k are reaction rates, H is the reaction enthalpy, q_{dil} the dilution heat, U is a heat transfer coefficient, Ω the heat exchange surface. $C_{p,I}$ denotes the approximated heat capacity of solid inserts (stirrer, baffles), C_p the approximated heat capacity of the entire mixture and $c_{p,A}$ is the specific molar heat capacity of component A. Equations for D and K_2 have been omitted: The number of moles of D equals the number of moles of C (i.e. $n_D(t) = n_C(t) \ \forall \ t$). The amount of ($\text{K}_2$) can be calculated as $n_{\text{K}_2}(t) = n_{\text{K}_1}(0) - n_{\text{K}_1}(t)$. The molarities c_i are calculated as $c_i = n_i/V$.

The dosing rate $u(t)$ of A to the batch reactor serves as the control input and can be assigned between upper and lower bounds.

The defining algebraic equations are:

$$V = 1000\left(\frac{n_A M_A}{\rho_A} + \frac{n_B M_B}{\rho_B} + \frac{n_C M_C}{\rho_C} + \frac{n_C M_D}{\rho_D}\right) \tag{2a}$$

$$\Omega = \Omega_{\min} + \frac{V - V_{\min}}{1000d} \tag{2b}$$

$$U = \left(U_1 + \frac{U_2 - U_1}{V_2 - V_1}\right)(V - V_1) \tag{2c}$$

$$x_A = \frac{n_A(t)}{n_A(t) + n_B(t) + 2n_C(t) + n_{\text{K}_1}(0)} \tag{2d}$$

$$q_{\text{dil}} = \frac{2232.74201}{0.13963}\, e^{\frac{-x_A}{0.13963}}\, \dot{n}_A \tag{2e}$$

$$r = (k_0 + k_2\, c_{K_1})c_A c_B + k_3 c_A\, c_{\text{K}_2} \tag{2f}$$

$$r_k = k_4 10^{-H_R} c_B c_{\text{K}_1}. \tag{2g}$$

In these equations, M_i, ρ_i, c_i denote the molar weight, density and molar concentration of component i respectively. V_1, V_2, U_1, U_2 are geometry-dependent parameters and d is the scaled reactor diameter. The rate of heat loss to the environment is modeled by a constant α of appropriate dimension derived from a constant heat transfer coefficient and an average heat transfer surface area. The constants in equation (2e) have been adjusted properly for the needed dimension (Watt). The reaction rate constants k_i are calculated following the Arrhenius approach as $k_i(t) = A_i e^{\frac{-E_i}{RT_R(t)}}$, and the acidity term H_R is computed as $H_R(t) = -\left(p_1 c_{K1}(t) + p_2 c_C(t)\right)\left(p_3 + \frac{p_4}{T_R(t)}\right)$.

The reactor is initially charged with B and K_1. Then, A is dosed to the reactor until the accumulated number of moles of A is equal to the initial number of moles of B. The batch is complete when nearly all of B is consumed.

3 NMPC Formulation

The solution of an open-loop optimal control problem is a prerequisite for NMPC. For batch processes, the formulation of a suitable optimal control problem tends

Table 1. List of process parameters and initial values

$\Omega_{\min}$	0.011 m^2	p_1	0.200 l/mol	d	0.155 m
$V_{\min}$	0.124 l	p_2	0.032 l/mol	$T_{R,0}$	293.15 K
M_A	0.130 kg/mol	p_3	-21.375	α	0.1 W/K
M_B	0.074 kg/mol	p_4	12706.0 K	C_p	1523.3 J/K
M_C	0.074 kg/mol	E_0	80.479 kJ/mol	V_1	0.8 l
M_D	0.130 kg/mol	E_2	79.160 kJ/mol	V_2	1.6 l
M_{K1}	0.098 kg/mol	E_3	69.975 kJ/mol	$m_{A,0}$	0.00 g
ρ_A	979.381 $\mathrm{kg/m}^3$	E_4	76.617 kJ/mol	$m_{B,0}$	510.98 g
ρ_B	772.288 $\mathrm{kg/m}^3$	R	8.314 J/(mol K)	$m_{\mathrm{K}_1,0}$	5.01 g
ρ_C	955.869 $\mathrm{kg/m}^3$	$c_{p,A}$	238.519 J/(mol K)	U_1	195 $\mathrm{W/(m^2K)}$
ρ_D	830.422 $\mathrm{kg/m}^3$	$C_{p,I}$	89.859 J/K	U_2	155 $\mathrm{W/(m^2K)}$
A_0	5.362e7 l/(mol s)	H	59458 J/mol	m_A	890.00 g
A_2	2.807e10 $\mathrm{l^2/(mol^2\ s)}$	H_{dil}	5070 J/mol	t_f	0.662 l
A_3	3.948e10 l/(mol s)	T_J	293.65 K	T_a	298.85 K
A_4	1.403e8 l/(mol s)	T_d	298.15 K		

to be difficult because of typically appearing end constraints. In this case, end constraints are avoided by observing that simply minimizing the amount of B over time leads to a meaningful solution. The batch is stopped when the amount of B is below a desired threshold. The optimal control problem is formulated as:

$$\min_{u} \int_{0}^{t_f} n_B(\tau)^2 \, d\tau \tag{3}$$

$$\begin{aligned} \text{subject to}\quad & (1),\ (2) \\ & 0 \text{ mol/s} \leq u(t) \leq 0.004 \text{ mol/s} \\ & \int_{t_0=0}^{t_f} u(\tau)\, d\tau = 6.9 \text{ mol} \\ & T_R(t) \leq 333.15 \text{ K} \\ & S(t) \leq 363.15 \text{ K}. \end{aligned}$$

The safety constraint S (defined in the next section) has to ensure that even in the extreme case of a total cooling failure no runaway will occur. The prediction and control horizon of the NMPC controller are specified as 56 times the sample time of $t_s = 50$ s. With this choice, the horizon is slightly larger than the minimum batch time for this process. All states are assumed perfectly measured.

3.1 A Suitable Safety Constraint to Avoid Runaways

In the case of a cooling failure, the heat released during the reaction can no longer be removed from the reactor. This leads to a temperature rise which further accelerates the reactions. If enough reactant had accumulated before, this mechanism results in a thermal runaway with severe safety risks [1]. Once a runaway has started, the best strategy is to immediately stop dosing the reactant. Then, the maximum temperature rise is related to the amount of reactants present in the reactor and can be calculated assuming adiabatic conditions [9].

Such an approach has been formulated more precisely in [17], where for two reactants A, B the safety constraint is $S(t) = T_R(t) + \min(n_A, n_B)\frac{H_A}{\rho c_p V} \leq T_{\max}$. Since the consumption rate for both species A and B is equal and all B is initially present in the reactor, n_A is smaller than n_B and we can set $\min(n_A, n_B) = n_A$ in order to avoid the nondifferentiable min-operator.

Note, that the calculated adiabatic temperature in $S(t)$ is rather conservative and will likely be smaller in reality because of heat losses to the jacket and ambient. Also, the heat capacity of the mixture is assumed to be constant with a value chosen at the upper limit.

3.2 NMPC Simulation Results

The open-loop control problem (3) is successively solved numerically with the direct multiple shooting approach by Bock and Plitt [3]. It is based on a parameterization of the controls and state trajectories. This leads to a large but favorably structured nonlinear program (NLP). The NLP is solved by a generalized Gauss-Newton sequential quadratic programming (SQP) method implemented in the software package MUSCOD-II [12]. Because SQP methods only find a local solution, the initial guess is of importance. For the nominal NMPC, an appropriate constant dosing rate served as an initial guess. For the robust NMPC introduced next, the nominal solution has been used as the initial guess. All integration and differentiation is performed with the DAE solver DAESOL [1], which applies a backward differentiation formula (BDF) method.

The batch is stopped once the remaining amount of 2-butanol falls below a threshold of $n_B \leq 0.01$ mol. Following the ideas of the real-time iteration scheme in [6], the optimization problem (3) is not solved to convergence at each sampling interval. Instead, the control is updated after each iteration step of the NLP solver. Due to a careful initialization from one problem to the next and the favorable contraction properties of the direct multiple shooting method, this procedure allows for close tracking of the optimal solution of the subsequent optimization problems. Note that nominal stability of this real-time iteration NMPC scheme can be shown [7]. Also, the scheme predicts active set changes and is therefore particularly suited for constrained control problems.

The CPU time for one control sample has been in the range of 0.9 to 1.2 seconds and hence is significantly smaller than the sampling time of 50 seconds.

The NMPC has been tested for the nominal set of parameters. Then, to test for robustness, the initial amount of catalyst K_1 has been increased. This

[1] Note, that such a runaway can also occur under normal cooling. However, this is not the focus of this study and, in the optimization results, is automatically suppressed by the upper bound on the process temperature.

process parameter has a rather strong impact on the process behavior. The simulation results for the nominal NMPC in the nominal case ($m_{\mathrm{K}_1}(0) = 5.01$ g) and two more cases with $m_{\mathrm{K}_1}(0) = 5.10$ g and $m_{\mathrm{K}_1}(0) = 5.50$ g are shown in Figure 1. In the nominal case, the dosing rate is at its maximum in the beginning, ensuring a fast ignition of the reaction and quick conversion. During this phase, A accumulates. This poses a potential threat and eventually the adiabatic temperature strongly rises. Once the bound on the adiabatic temperature becomes active, the dosing rate slides along a singular sub-arc until the reactor temperature will reach its upper operation limit. This is when the controller decides to stop the dosing to let the accumulated A be consumed. In the end, the remaining amount of A can safely be added at a maximum dosing rate.

The simulations show that the nominal NMPC scheme based on the open loop problem (3) keeps the batch process within safe operation conditions despite a moderate uncertainty in the initial amount of catalyst (remember that the model always assumes the nominal value of 5.01 g to be valid). Because of the higher temperatures in comparison to the model-based predictions and the feedback mechanism, the singular arcs become steeper and the dosing has to be stopped earlier. In the extreme case of 5.50 g of catalyst in the beginning of the batch, the upper limit of the reactor temperature is slightly violated. The NLP only remained feasible due to a relaxation procedure implemented in the optimization code.

In the following section, the NMPC scheme is modified to take the uncertainty explicitly into account.

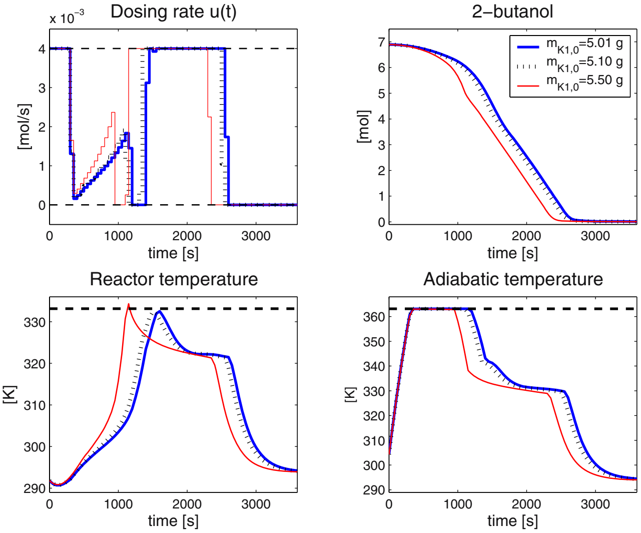


Fig. 1. Simulation results with nominal NMPC controller for the nominal value and two perturbed values of the initial amount of catalyst K_1. Dashed lines denote constraints.

4 A Min-Max NMPC Formulation

This paper has presented an open-loop optimization problem for NMPC with safety constraints for a given process model with exact parameters. In reality, at least some of the parameters will not be known exactly and methods for optimization under uncertainty become important (see [16] for an overview).

It is clear that a changed parameter can deteriorate the performance of a supposedly optimal (open-loop) solution. It seems natural to see parameters as adverse players that try to disturb any control efforts as strongly as possible. One remedy is to minimize the success of parameters to maximally worse the optimization results. In the context of robust batch optimization, such a min-max formulation has been used by Nagy and Braatz [15] who also point out the possibility to extend these ideas to NMPC. Such min-max formulations lead to a semi-infinite programming problem, which is numerically intractable in the real-time context. To overcome this obstacle, Körkel et al. [10] propose an approximated min-max formulation which is also applied in this paper.

The min-max formulation considered here reads as the semi-infinite programming problem

$$\begin{aligned} \min_{u \in \mathcal{U}} \quad & \max_{\|p-\bar{p}\|_{2,\Sigma^{-1}} \leq \gamma} \quad J(x(u,p)) \\ \text{s.t.} \quad & \max_{\|p-\bar{p}\|_{2,\Sigma^{-1}} \leq \gamma} \quad r_i(x(u,p)) \leq \ 0, \quad i = 1, \ldots, n_r. \end{aligned} \tag{4}$$

The parameters p are assumed to lie within a given confidence region with the symmetric covariance matrix Σ and an arbitrary confidence level γ. The state x implicitly depends on the discretized control u and parameters p as a solution to the system equations (1,2). The cost function J is the same as in the nominal problem (3), while $r_i(x(u,p))$ summarizes those constraints in (3) that are considered critical with respect to uncertainty and shall be robustified. All other constraints are treated as in the nominal problem.

First order Taylor expansion of the inner maximization part yields a convex optimization problem that has an analytical solution (cf. [10] for this problem and [8] for a more general problem class). Using this closed form, we finally obtain a minimization problem that can efficiently be solved numerically:

$$\begin{aligned} \min_{u \in \mathcal{U}} \quad & J(x(u,\bar{p})) + \gamma \left\| \frac{\mathrm{d}}{\mathrm{d}p} J(x(u,\bar{p})) \right\|_{2,\Sigma} \\ \text{s.t.} \quad & r_i(x(u,\bar{p})) + \gamma \left\| \frac{\mathrm{d}}{\mathrm{d}p} r_i(x(u,\bar{p})) \right\|_{2,\Sigma} \leq \ 0, \quad i = 1, \ldots, n_r, \end{aligned} \tag{5}$$

where the bar denotes the nominal value of the parameters as assumed for the nominal optimization problem.

In an NMPC scheme, the robust version (5) can replace the nominal control problem (3). In the next section, numerical results for such a robust NMPC are presented and compared to the nominal NMPC.

4.1 Min-Max NMPC Simulation Results

For the numerical solution of (5) the direct multiple shooting approach was used. The NMPC settings are the same as described in Section 3.2. The confidence factor γ was slowly increased from zero to the desired level. For each optimization with a respective γ, the previous result has been used to initialize the states and control. In the case presented here, the initial amount of catalyst K_1 is assumed to be uncertain. The standard deviation of K_1 is 0.17 g. The confidence level has been chosen to be 99.7 %, i.e. we have $\gamma = 3$ to obtain the 3σ-interval. Figure 2 shows the simulation results for the robust version of the NMPC for the nominal amount of catalyst and an increased amount. The solution is compared to the solution of the nominal NMPC for the nominal catalyst amount charged to the reactor. One can see that the robust solution strongly resembles the nominal solution. Only, the dosing is stopped earlier. This ensures that less A is accumulating in the reactor and leaves a safety margin to the adiabatic temperature as can be seen in the lower right graph. When the robust NMPC controller is confronted with a plant-model mismatch it reacts by dosing A more carefully. The singular sub-arc becomes flatter than computed with the nominal NMPC. Eventually, less A is present in the reactor and the temperature peak gets lower. This also leads to a slower consumption of B which, however, is accounted for by a higher reactor temperature at the end of the batch so that the final productivity losses are very small.

The safety margins for the reactor temperature and the adiabatic temperature are the main feature of the robust NMPC scheme. The fact that they are also

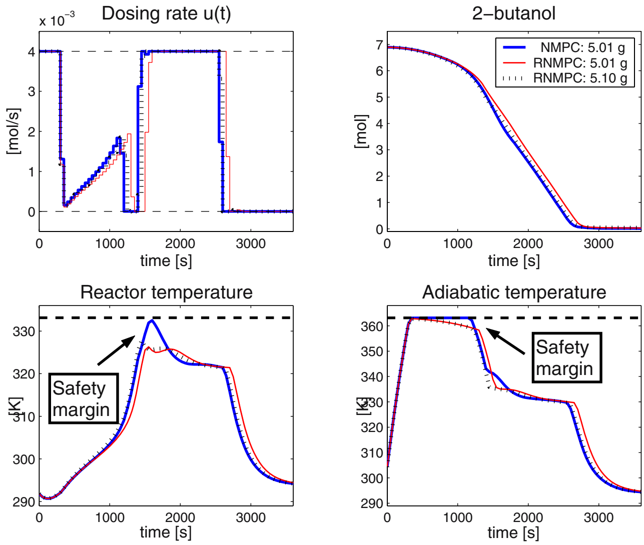


Fig. 2. Simulation results of robust (RNMPC) compared with nominal NMPC. The RNMPC is shown for the nominal and a perturbed initial amount of catalyst K_1. Dashed lines denote constraints. Note the RNMPC safety margins.

present in the nominal catalyst case reflects the conservatism inherent to the min-max approach. For the investigated cases, we have seen that already the nominal controller is robust against the uncertain parameter. So, any further robustification in the presence of uncertainty makes the controller more cautious. Also, the systematic robustness obtained via the min-max formulation is not fully exploited by the closed-loop dynamics. Instead of relying on the old, robustified solution, the optimization problem (5) is newly solved at every iteration with the new, unforeseen temperatures and concentrations. This is why the trajectories of the robust NMPC differ for the nominal and the perturbed realization of the initial amount of catalyst.

5 Conclusions

A detailed model of a fed-batch reactor has been used to demonstrate that a suitable NMPC scheme can avoid runaway situations. This nominal NMPC was robust against small perturbations of the initial amount of catalyst charged to the reactor. A robust formulation of the optimization problem based on an approximated min-max formulation led to additional safety margins with respect to the adiabatic temperature and the reactor temperature. The approximation does not guarantee full robustness for the nonlinear model, but it offers a systematic way to obtain safety margins that explicitly take into account uncertain parameters and their stochastic properties. The simulation also showed the conservatism of the proposed min-max formulation which is due to the open-loop formulation. The optimization method used to solve the nominal and the robust open-loop problems was able to deal with the complex state constraints and delivered control updates fast enough to be applicable to real batch processes. Future studies will treat larger numbers of uncertain parameters and focus on methods to efficiently solve the eventually enlarged optimization problems.

References

[1] Bauer I, Bock HG, Schlöder JP (1999) “DAESOL - a BDF-code for the numerical solution of differential algebraic equations”, Internal report IWR-SFB 359, Universität Heidelberg

[2] Benuzzi A and Zaldivar JM (eds) (1991) “Safety of Chemical Reactors and Storage Tanks”, Kluwer Academic Publishers, Dordrecht, The Netherlands

[3] Bock HG and Plitt KJ (1984) “A multiple shooting algorithm for direct solution of optimal control problems”, Proc. 9th IFAC World Congress Budapest

[4] Chen H, Scherer CW, Allgöwer F (1997) “A game theoretic approach to nonlinear robust receding horizon control of constraint systems”, American Control Conference, Albuquerque, USA: 3073–3077

[5] De Nicolao G, Magni L, Scattolini R (2000) “Stability and robustness of nonlinear receding horizon control”. In: Allgöwer F, Zheng A (eds) Nonlinear Model Predictive Control. Birkhäuser Basel.

[6] Diehl M, Bock HG, Schlöder JP, Findeisen R, Nagy Z, Allgöwer F (2002) “Real-time optimization and Nonlinear Model Predictive Control of Processes governed by differential-algebraic equations”, J. of Proc. Control 12:577–585

[7] Diehl M, Findeisen R, Bock HG, Schlöder JP, Allgöwer F (2005) “Nominal stability of the real-time iteration scheme for nonlinear model predictive control”, IEE Proc. Contr. Theory & Appl. 152(3):296–308

[8] Diehl M, Bock HG, Kostina E (2005) “An approximation technique for robust nonlinear optimization”, Mathematical Programming B (accepted)

[9] Hugo P, Steinbach J, Stoessel F (1988) “Calculation of the maximum temperature in stirred reactors in case of a breakdown of cooling”, Chemical Eng. Science 43:2147–2152

[10] Körkel S, Kostina E, Bock HG, Schlöder JP (2004) “Numerical methods for optimal control problems in design of robust optimal experiments for nonlinear dynamic processes”, Optimization Methods & Software 19:327–338

[11] Lee JH, Yu Z (1997) “Worst-case formulations of model predictive control for systems with bounded parameters”, Automatica 33(5):763–781

[12] Leineweber DB, Bauer I, Bock HG, Schlöder JP (2003) “An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization - part I: theoretical aspects”, Comp. & Chem. Eng. 27:157–166

[13] Li P, Wendt M, Wozny G (2002) ”A probabilistically constrained model predictive controller”, Automatica, 38: 1171–1176

[14] Magni L, Nijmeijer H, van der Schaft AJ (2001) “A receding-horizon approach to the nonlinear H-infinity control problem”, Automatica 37:429–435

[15] Nagy Z, Braatz RD (2004) “Open-loop and closed-loop robust optimal control of batch processes using distributional and worst-case analysis”, Journal of Process Control 14:411–422

[16] Sahinidis NV (2004) “Optimization under uncertainty: state-of-the-art and opportunities”, Comp. & Chem. Eng. 28:971–983

[17] Ubrich O, Srinivasan B, Lerena P, Bonvin D, Stoessel F (1999) “Optimal feed profile for a second order reaction in a semi-batch reactor under safety constraints - experimental study”, J. of Loss Prev. 12:485–493

[18] Westerterp KR, Molga E (2004) “No more Runaways in Fine Chemical Reactors”, Ind. Eng. Chem. Res. 43:4585–4594

Real-Time Implementation of Nonlinear Model Predictive Control of Batch Processes in an Industrial Framework

Zoltan K. Nagy[1], Bernd Mahn[2], Rüdiger Franke[3], and Frank Allgöwer[4]

[1] Loughborough University, Chemical Engineering Department, Loughborough, LE11 3TU, United Kingdom
Z.K.Nagy@lboro.ac.uk
[2] BASF Aktiengesellschaft, Ludwigshafen, Germany
[3] ABB Corporate Research, Ladenburg, Germany
[4] University of Stuttgart, Stuttgart, Germany

Summary. The application of nonlinear model predictive control (NMPC) for the temperature control of an industrial batch polymerization reactor is illustrated. A real-time formulation of the NMPC that takes computational delay into account and uses an efficient multiple shooting algorithm for on-line optimization problem is described. The control relevant model used in the NMPC is derived from the complex first-principles model and is fitted to the experimental data using maximum likelihood estimation. A parameter adaptive extended Kalman filter (PAEKF) is used for state estimation and on-line model adaptation. The performance of the NMPC implementation is assessed via simulation and experimental studies.

1 Introduction

Trends in the process industries toward high value added products have increased the interest in the optimal operation of batch processes, used predominantly for high-tech products. Batch processes are common in the pharmaceutical, microelectronics, food, and fine chemical industries. It is widely recognized at industrial level that advanced control techniques have the potential to improve process performance [QB03]. Since the advent of dynamic matrix control (DMC), model predictive control (MPC) has been the most popular advanced control strategy in chemical industries [ML97]. Linear MPC has been heralded as a major advance in industrial control. However, due to their nonstationary and highly nonlinear nature, linear model based control usually cannot provide satisfactory performance in the case of complex batch processes. Nonlinear model predictive control (NMPC) has been considered as one of the most promising advanced control approaches for batch processes. NMPC reformulates the MPC problem based on nonlinear process models. Different nonlinear models can be used for prediction, from empirical black-box models (e.g. artificial neural networks, Volterra series, etc.) to detailed, first-principles based representations of the system, leading to a wide variety of different NMPC approaches [Hen98], [FA02]. The advantages of using complex nonlinear models in the NMPC are

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 465–472, 2007.
springerlink.com

straightforward. First-principles models are transparent to engineers, give the most insight about the process, and are globally valid, and therefore well suited for optimization that can require extrapolation beyond the range of data used to fit the model. Due to recent developments in computational power and optimization algorithms, NMPC techniques are becoming increasingly accepted in the chemical industries, NMPC being one of the approaches, which inherently can cope with process constraints, nonlinearities, and different objectives derived from economical or environmental considerations. In this paper an efficient real-time NMPC is applied to an industrial pilot batch polymerization reactor. The approach exploits the advantages of an efficient optimization algorithm based on multiple shooting technique [FAww], [Die01] to achieve real-time feasibility of the on-line optimization problem, even in the case of the large control and prediction horizons. The NMPC is used for tight setpoint tracking of the optimal temperature profile. Based on the available measurements the complex model is not observable hence cannot be used directly in the NMPC strategy. To overcome the problem of unobservable states, a grey-box modelling approach is used, where some unobservable parts of the model are described through nonlinear empirical relations, developed from the detailed first-principles model. The resulting control-relevant model is fine tuned using experimental data and maximum likelihood estimation. A parameter adaptive extended Kalman filter (PAEKF) is used for state estimation and on-line parameter adaptation to account for model/plant mismatch.

2 Nonlinear Model Predictive Control

2.1 Algorithm Formulation

Nonlinear model predictive control is an optimization-based multivariable constrained control technique that uses a nonlinear dynamic model for the prediction of the process outputs. At each sampling time the model is updated on the basis of new measurements and state variable estimates. Then the open-loop optimal manipulated variable moves are calculated over a finite prediction horizon with respect to some cost function, and the manipulated variables for the subsequent prediction horizon are implemented. Then the prediction horizon is shifted or shrunk by usually one sampling time into the future and the previous steps are repeated. The optimal control problem to be solved on-line in every sampling time in the NMPC algorithm can be formulated as:

$$\min_{u(t)\,\in\,\mathcal{U}} \{\mathcal{H}(x(t),u(t);\theta)=\mathcal{M}(x(t_F);\,\theta)+\int_{t_k}^{t_F}\mathcal{L}(x(t),u(t);\theta)dt\} \tag{1}$$

$$s.t.\;\; \dot{x}(t)=f(x(t),\;u(t);\;\theta), \qquad x(t_k)=\hat{x}(t_k),\;\; x(t_0)=\hat{x}_0 \tag{2}$$

$$h(x(t),\;u(t);\;\theta)\le 0, \qquad t\in[t_k,\;t_F] \tag{3}$$

where $\mathcal{H}$ is the performance objective, t is the time, t_k is the time at sampling instance k, t_F is the final time at the end of prediction, is the n_x vector of states, $u(t) \in \mathcal{U}$ is the n_u set of input vectors, is the n_y vector of measured variables used to compute the estimated states $\hat{x}(t_k)$, and $\theta \in \Theta \subset \mathcal{R}^{n_\theta}$ is the n_θ vector of possible uncertain parameters, where the set Θ can be either defined by hard bounds or probabilistic, characterized by a multivariate probability density function. The function $f : \mathcal{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathcal{R}^{n_x}$ is the twice continuously differentiable vector function of the dynamic equations of the system, and $h : \mathcal{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathcal{R}^c$ is the vector of functions that describe all linear and nonlinear, time-varying or end-time algebraic constraints for the system, where c denotes the number of these constraints.

We assume that $\mathcal{H} : \mathcal{R}^{n_x} \times \mathcal{U} \times \Theta \rightarrow \mathcal{R}$ is twice continuously differentiable, thus fast optimization algorithms, based on first and second order derivatives may be exploited in the solution of (1). The form of $\mathcal{H}$ is general enough to express a wide range of objectives encountered in NMPC applications. In NMPC the optimization problem (1)-(3) is solved iteratively on-line, in a moving (receding) horizon ($t_F < t_f$) or shrinking horizon ($t_F = t_f$) approach, where t_f is the batch time.

2.2 Solution Strategy and Software Tool

Considering the discrete nature of the on-line control problem, the continuous time optimization problem involved in the NMPC formulation is solved by formulating a discrete approximation to it, that can be handled by conventional nonlinear programming (NLP) solvers [BR91], [Bie00]. The time horizon $t \in [t_0,\ t_f]$ is divided into N equally spaced time intervals Δt (stages), with discrete time steps $t_k = t_0 + k\Delta t$, and $k = 0, 1, ..., N$. Model equations are discretized, $x_{k+1} = f_k(x_k, u_k; \theta)$, and added to the optimization problem as constraints. For the solution of the optimization problem a specially tailored NMPC tool (*OptCon*) was developed that includes a number of desirable features. In particular, the NMPC is based on first-principles or grey box models, and the problem setup can be done in Matlab. The NMPC approach is based on a large-scale NLP solver (HQP) [FAww], which offers an efficient optimization environment, based on multiple shooting algorithm, that devides the optimization horizon into a number of subintervals (stages) with local control parameterizations. The differential equations and cost on these intervals are integrated independently during each optimization iteration, based on the current guess of the control. The continuity/consistency of the final state trajectory at the end of the optimization is enforced by adding consistency constraints to the nonlinear programming problem.

2.3 Real-Time Implementation

In NMPC simulation studies usually immediate feedback is considered, i.e. the optimal feedback control corresponding to the information available up to the moment t_k, is computed, $u^\star(t_k) = [u_{0|t_k}, u_{1|t_k}, \ldots, u_{N|t_k}]$, and the first value ($u_{0|t_k}$)

is introduced into the process considering no delay. However, the solution of the NLP problem requires a certain, usually not negligible, amount of computation time δ_k, while the system will evolve to a different state, where the solution $u^\star(t_k)$ will no longer be optimal [FA02]. Computational delay δ_k has to be taken into consideration in real-time applications. In the approach used here, in moment t_k, first the control input from the second stage of the previous optimization problem $u_{1|t_{k-1}}$ is injected into the process, and then the solution of the current optimization problem is started, with fixed $u_{0|t_k} = u_{1|t_{k-1}}$. After completion, the optimization idles for the remaining period of $t \in (t_k + \delta_k, t_{k+1})$, and then at the beginning of the next stage, at moment $t_{k+1} = t_k + \Delta t$, $u_{1|t_k}$ is introduced into the process, and the algorithm is repeated. This approach requires real-time feasibility for the solution of each open-loop optimization problems ($\delta_k \leq \Delta t$).

2.4 State Estimation

Proper state estimation is crucial for successful practical NMPC applications. Extended Kalman filter (EKF) is widely used in process control applications, however its performance strongly depends on the accuracy of the model. To avoid highly biased model predictions, selected model parameters are estimated together with the states, leading to a parameter adaptive EKF formulation [VG00]. Define $\theta' \subseteq \theta$ as the vector of the estimated parameters from the parameter vector, and $\theta'' \overset{\Delta}{=} \theta \backslash \theta'$ the vector of the remaining parameters. The augmented state vector in this case is given by $\mathcal{X} = [x,\ \theta']^T$, and the augmented model used for estimation is, $\dot{\mathcal{X}} = [f(x,\ \theta', u;\ \theta''),\ \mathbf{0}]^T + [w,\ \ w_{\theta'}]^T$, with w , and $w_{\theta'}$ zero-mean Gaussian white noise variables. The measurement covariance matrix is determined based on the accuracy of the measurements. The appropriate choice of the state covariance matrix, $\mathbf{Q}$, is however often difficult in practical applications. An estimate of $\mathbf{Q}$ can be obtained by assuming that the process noise vector mostly represents the effects of parametric uncertainty [VG00], [NB03]. Based on this assumption the process noise covariance matrix can be computed as $\mathbf{Q}(t) = \mathbf{S}_\theta(t)\mathbf{V}_\theta\mathbf{S}_\theta^T(t)$, with $\mathbf{V}_\theta \in \mathcal{R}^{n_\theta \times n_\theta}$ being the parameter covariance matrix, and $\mathbf{S}_\theta(t) = (\partial f/\partial \theta)_{\hat{x}(t),u(t),\hat{\theta}}$ is the sensitivity jacobian computed using the nominal parameters and estimated states. This approach provides an easily implementable way to estimate the process noise covariance matrix, since the parameter covariance matrix $\mathbf{V}_\theta$ is usually available from parameter estimation, and the sensitivity coefficients in $\mathbf{S}_\theta$ can be computed by finite differences or via sensitivity equations. Note that the above approach leads to a time-varying, full covariance matrix, which has been shown to provide better estimation performance for batch processes than the classically used constant, diagonal $\mathbf{Q}$ [VG00], [NB03].

3 Practical Implementation of NMPC to an Industrial Pilot Batch Reactor

A schematic representation of the experimental pilot plant is shown on Figure 1. The reactor temperature is controlled using a complex heating-cooling system, which is based on a closed oil circuit, which is recycled through the jacket with a constant flow rate F_j. The heating-cooling medium goes through a multi-tubular heat exchanger where a PI controller is used to keep the temperature difference constant, by adjusting the cooling water flow rate. Heating is performed using an electric heater. The power of the heater is adjusted by a PI controller that regulates the input temperature into the jacket. The setpoint of the PI controller is determined by the higher level NMPC that has the objective to track a predetermined temperature profile in the reactor.

A detailed first-principles model of the process containing material and energy balances as well as detailed kinetic and thermodynamic models was used and identified based on off-line experiments. Since only temperature measurements are available in the plant, many states of the detailed model are not estimable, or not even detectable. The complex model however was used to determine the optimal temperature profile, and for deriving the control-relevant model. Available measurements are: reactor temperature (T_r), and input and output temperatures into and from the jacket, (T_{jin}, T_j). With this set of measurements the following reduced model was used in the NMPC:

$$\dot{n}_M = -Q_r/\Delta H_r \tag{4}$$

$$\dot{T}_{r,k} = \frac{Q_r + U_w A_w (T_{w,k} - T_{r,k}) - (UA)_{loss,r}(T_{r,k} - T_{amb})}{m_M c_{p,M} + m_P c_{p,P} + m_{water} c_{p,water}} \tag{5}$$

$$\dot{T}_{w,k} = (U_j A_j (T_{j,k} - T_{w,k}) - U_w A_w (T_{w,k} - T_{r,k}))/m_w/c_{pw} \tag{6}$$

$$\dot{T}_{j,k} = \frac{\mathcal{N} F_j \rho_j c_{p,j}(T_{j,k-1} - T_{j,k}) - U_j A_j (T_{j,k} - T_{w,k}) - (UA)_{loss,j}(T_{j,k} - T_{amb})}{m_j c_{p,j}} \tag{7}$$

where $k = 1, \ldots, \mathcal{N}$, $T_r = T_{r,\mathcal{N}}$, $T_j = T_{j,\mathcal{N}}$, $T_{j,0} = T_{jin}$, n_M is the number of mol of monomer, ΔH_r is the enthalpy of reaction, T_w is the wall temperature, U and A are heat transfer coefficients and areas from reactor to wall $(\cdot)_w$ or wall to jacket $(\cdot)_j$, $c_{p,M/P/water/w/j}$ and $m_{M/P/water/w/j}$ are the heat capacities and masses of monomer, polymer, water, wall and oil, T_{amb} is the ambient temperature, ρ_j is the density of the oil, $(UA)_{loss,r/j}$ heat loss coefficients in the reactor and jacket, respectively.

To estimate the transport delay, the reactor, wall and jacket were divided in $\mathcal{N} = 4$ elements, leading to a system of 13 differential equations. To achieve proper prediction and maintain the observability of the model, with only temperature measurements available, different approaches have been proposed. Helbig et al. used a time series of the estimated heat generation determined from simulation of a batch [HA96]. The industrial batch MPC product developed by IPCOS determines an empirical nonlinear relation $Q_r = f_Q(n_M, T_r)$, which expresses the heat generation as a function of the conversion and temperature [IP00]. In our case study a similar approach was used. The empirical nonlinear relation was determined from the complex first principle model, simulating the process for different temperature profiles.

Maximum likelihood estimation was used to fit the parameters of the model (4)-(7) to the data obtained from the plant, performing several water batches (when $Q_r = 0$), using $\theta'' = [(UA)_{loss,r},\ (UA)_{loss,j},\ U_jA_j,\ m_w,\ m_j]$ as the parameter vector. This procedure gives the optimal nominal parameter estimates, $\hat{\theta''}^{\star}$, and the corresponding uncertainty description given by the covariance matrix, estimated from the Hessian of the objective used in the maximum likelihood estimation. The good fit between the experimental data and the model is shown on Figure 2.

Model (4)-(7) was used in an adaptive output feedback NMPC approach, where the objective was to provide a tight setpoint tracking, by minimizing online, in every sampling instance k, the following quadratic objective:

$$\min_{u(t)} \int_{t_k}^{t_F} \{(T_r(t) - T_r^{ref}(t))^2 + Q_{\Delta u}(du(t))^2\}dt \tag{8}$$

The optimal setpoint profile T_r^{ref} is generally obtained via off-line optimization using the detailed model. In our implementation however, a suboptimal but typical profile consisting of three piece-wise linear segments was used. The manipulated input of the NMPC, $u(t) = T_{jSP}$, is the setpoint temperature

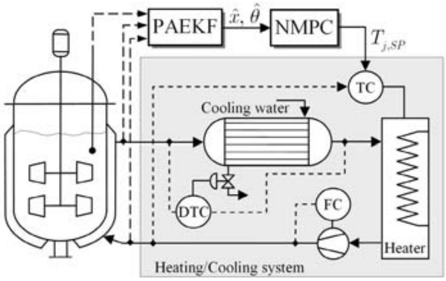


Fig. 1. Schematic representation of the batch reactor system

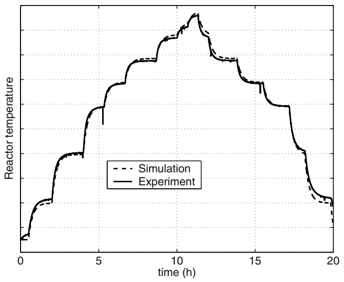


Fig. 2. Validation of the model compared to plant data

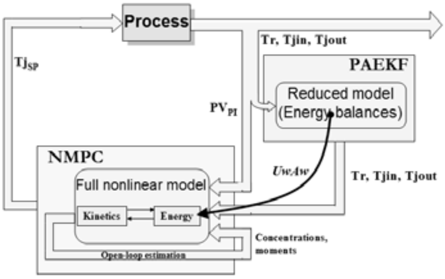


Fig. 3. Implementation structure of the PAEKF and NMPC

to the lower level PI controller, which controls the jacket input temperature. The communication between the real plant and NMPC was performed via the standard OPC interface. The adaptive structure of the implemented NMPC is shown on Figure 3. During the batch the heat transfer properties in the reactor change significantly thus the adaptive algorithm is important. The parameters $\theta' = [Q_r,\ U_w A_w]$ were estimated together with the model states in the PAEKF. Figure 4 indicates a strong variation of $U_w A_w$ during the batch. Figure 5 demonstrates the very good setpoint tracking performance of the NMPC with adapted model. The parameter covariance matrix $\mathbf{V}_\theta$, resulted from the identification was used to compute the state covariance matrix in the estimator [VG00], [NB03]. A weighting coefficient of $Q_{\Delta u} = 0.4$, and prediction and control horizons of 8000s were used, in the optimization, with a sampling time of 20s. The control input was discretized in 400 piecewise constant inputs, leading to a high dimensional optimization problem. The efficient multiple shooting approach guarantees the real-time feasibility of the NMPC implementation. Even with the large control discretization of 400 the computation time was below the sampling time of 20s (approx. 5s). All simulation times are on a Pentium 3, 800 MHz PC running Windows 2000.

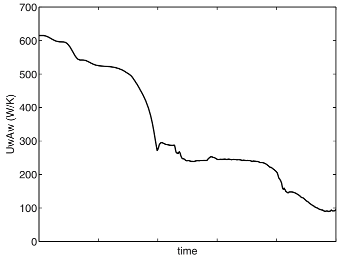


Fig. 4. Estimated Heat transfer coefficient during the batch

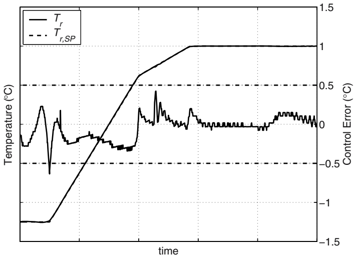


Fig. 5. Experimental results: NMPC of the industrial batch reactor

4 Conclusions

The paper present a computationally efficient NMPC approach that combines output feedback design with efficient optimization technique providing a framework that can be supported in an industrial environment. Detailed first-principles model is used to derive the reduced control-relevant model based on the available measurements, which is tuned using data from the plant, and used then in the NMPC. A PAEKF is combined with the control algorithm for the on-line state estimation and model adaptation to achieve offset free control. Simulation and experimental results demonstrate the efficiency of the NMPC approach in an industrial application.

References

[Die01] Diehl, M.: Real-Time Optimization for Large Scale Nonlinear Processes. PhD Thesis, University of Heidelberg, Heidelberg (2001)

[Bie00] Biegler, L.: Efficient solution of dynamic optimization and NMPC problems. In: F. Allgoewer and A. Zheng, (eds), Nonlinear Predictive Control. Birkhauser, Bagel (2000)

[QB03] Qin S.J., Badgwell, T.: A Survey of Industrial Model Predictive Control Technology. Control Engineering Practice, **11**, 733–764 (2003)

[ML97] Morari, M., Lee, J.H.: Model predictive control: Past, present and future. In Proc. PSE'97-ESCAPE-7 Symposium, Trondheim, (1997)

[Hen98] Henson, A.M.: Nonlinear model predictive control: current status and future directions. Comp. Chem. Eng., **23**, 187–201 (1998)

[FA02] Findeisen, R., Allgoewer, F.: Nonlinear model predictive control: From theory to application. In Int. Symp. on Design, Operation and Control of Chemical Plants (PSE Asia'02), Taipei, Taiwan (2002)

[FAww] Franke, R., Arnold, E. Linke H.: HQP: A solver for nonlinearly constrained large-scale optimization. http://hqp.sourceforge.net

[BR91] Biegler, L.T., Rawlings, J.B.: Optimisation approaches to nonlinear model predictive control. Proceedings of Conf. Chemical Process Control, South Padre Island, Texas, 543–571 (1991)

[VG00] Valappil, J., Georgakis, C.: Systematic estimation of state noise statistics for Extended Kalman Filters. AIChE J., **46**, 292–308 (2000)

[NB03] Nagy, Z.K., Braatz, R.D.: Robust nonlinear model predictive control of batch processes. AIChE J., **49**, 1776–1786 (2003)

[HA96] Helbig, A., Abel, O., M'hamdi, A., Marquardt, W.: Analysis and nonlinear model predictive control of the Chylla-Haase Benchmark problem. Proceedings of UKACC International Conf. on Control 1172–1177 (1996)

[IP00] Van Overschee, P., Van Brempt, W.: polyPROMS IPCOS-ISMC Final Report, Internal Report of the polyPROMS European 5th Framework Research Project GRD1-2000-25555.

Non-linear Model Predictive Control of the Hashimoto Simulated Moving Bed Process

Achim Küpper and Sebastian Engell

Process Control Laboratory, Department of Chemical and Biochemical Engineering,
Universität Dortmund, Emil-Figge-Str. 70, D-44221 Dortmund, Germany
sebastian.engell@bci.uni-dortmund.de,
achim.kuepper@bci.uni-dortmund.de

Summary. In recent years, continuous chromatographic processes have been established as an efficient separation technology in industry, especially when temperature sensitive components or species with similar thermodynamic properties are involved. In SMB processes, a counter-current movement of the liquid and the solid phases is achieved by periodically switching the inlet and the outlet ports in a closed loop of chromatographic columns. The integration of reaction and separation in one single plant is a promising approach to overcome chemical or thermodynamic equilibria and to increase process efficiency. Reactive chromatographic SMB processes in which the columns are packed with catalyst and adsorbent have been proposed and demonstrated successfully. However, a full integration often is not efficient because in the columns in the separating zones, the catalyst is not used or even counterproductive. By placing reactors between the separation columns at specific positions around the feed port, a more efficient process, the Hashimoto SMB process, is established. In this contribution, a non-linear predictive control concept for the Hashimoto SMB process is presented. The controller computes optimal control variables (flow rates and the switching time) to optimize an economic objective over a moving horizon. The purity requirements of the product streams are implemented as constraints and not as controlled variables. The optimization-based controller is combined with a scheme to estimate selected model parameters in order to reduce the influence of the inevitable model errors. Simulative results are presented for the example of the racemization of Tröger's base.

Keywords: Simulated moving bed chromatography (SMB), Hashimoto process, online optimization, parameter estimation.

1 Introduction

Chromatographic separation processes are based on different affinities of the involved components to a solid adsorbent packed in a chromatographic column. Most industrial applications are performed discontinuously involving one single chromatographic column which is charged with pulses of feed solutions. The feed injections are carried through the column by pure eluent. The more retained component travels through the column slower, and hence leaves the column after the less adsorptive component. The separated peaks can be withdrawn with the desired purity at the end of the column. However, batch operation leads to low productivity and high solvent consumption.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 473–483, 2007.
springerlink.com

In recent years, continuous Simulated Moving Bed SMB processes are increasingly applied due to their advantages with respect to the utilization of the adsorbent and the consumption of the solvent. The SMB process consists of several chromatographic columns which are interconnected in series to constitute a closed loop. A counter-current movement of the liquid phase and the solid phase is simulated by periodical and simultaneous switching of the inlet and outlet ports by one column in the direction of the liquid flow.

The Hashimoto SMB [1] is an extension of the Simulated Moving Bed process which integrates reaction into chromatographic separation and is therefore suitable to overcome the limitations of equilibrium reactions. The reactors are fixed in those separation zones of the Hashimoto SMB process where the forward reaction is favourable thus increasing the conversion of the feed.

Since SMB processes are characterized by mixed discrete and continuous dynamics, spatially distributed state variables with steep slopes, and slow and strongly nonlinear responses of the concentrations profiles to changes of the operating parameters, they are difficult to control. An overview of recent achievements in the optimization and control of chromatographic separations can be found in [3]. In [7] and [8], a nonlinear optimizing control scheme was proposed and successfully applied to a three-zone reactive SMB process for glucose isomerization. In each switching period, the operating parameters are optimized to minimize a cost function. The product purities appear as constraints in the optimization problem. In the optimization, a rigorous model of the general rate type is used. Plant/model mismatch is taken into account by error feedback of the predicted and the measured purities. In addition, the model parameters are regularly updated. In [4], the control concept was extended to the more complex processes Varicol and Powerfeed that offer a larger number of degrees of freedom that can be used for the optimization of the process economics while satisfying the required product purities. A slightly different approach to the control of SMB processes was reported by [5] and [6]. Here, the online optimization is based upon a linearized reduced model which is corrected by a Kalman filter that uses the concentration measurements in the product streams. In this work, the switching period is considered as fixed, while in the previously mentioned work it is a parameter in the optimization. In [7] and [8], the prediction is based on the assumption that the columns are uniform (i.e. they all show the same behavior) and that the modelling errors are small. However, the properties of each individual column differ since they have different effective lengths, different packings with adsorbent and catalyst (for the case of reactive chromatography) and the column temperatures can exhibit some variation. In [10], a combined parameter and state estimation scheme for a nonlinear SMB process with individual column properties based on measurements of the concentrations in both product streams and one internal measurement is proposed.

In this paper, a control concept for the Hashimoto SMB process is presented. A non-linear predictive controller for the Hashimoto SMB process is established that computes optimal control variables (flow rates and the switching time) to optimize an economic objective over a moving horizon while the purity requirements of the product streams are considered as constraints. In the optimization, a rigorous model of the general rate type is used. Plant/model mismatch is taken into account by error feedback of the predicted and the measured purities. In addition, the model parameters are regularly updated by a parameter estimation scheme.

The remainder of this paper is structured as follows: in the next section, the model of the Hashimoto SMB process is introduced. Section 3 is devoted to the predictive control concept based upon an online optimization and parameter estimation scheme. Simulation results are presented in section 4. Finally, a summary and an outlook for future research are given.

2 Process Model

In this paper, the racemization of Tröger's base (TB) is considered. Tröger's base consists of the enantiomers TB- and TB+ with TB- as the desired product which is used for the treatment of cardiovascular diseases. Both Tröger's base components form an equimolar equilibrium. Since the product TB- has a higher affinity to the chosen adsorbent, it is withdrawn at the extract port of the Hashimoto process. The TB+ part is withdrawn with the raffinate stream in order to improve the purification of the solvent in zone IV before it is passed on to zone I. Theoretically, this raffinate flow can be converted to the equilibrium by an additional external reactor and added to the feed stream. Alternatively, no raffinate flow can be taken out, thus the whole feed is converted to TB- making the additional unit to convert TB+ redundant. However, in this case a drastic increase of eluent would be required and the process would be less efficient. For the application in this paper, the first case is examined with the Hashimoto configuration depicted in Figure 1. The reactors are placed in zone III where a high concentration of TB+ is present. When the ports of the process are shifted by one column after the period τ has passed, the physical separation columns are switched by one column in the opposite direction to the liquid flow moving through the separation zones. The reactors, however, remain at their positions relative to the ports. The practical realization of the column switching via a process control system is sophisticated and indicated by Figure 2 which shows the

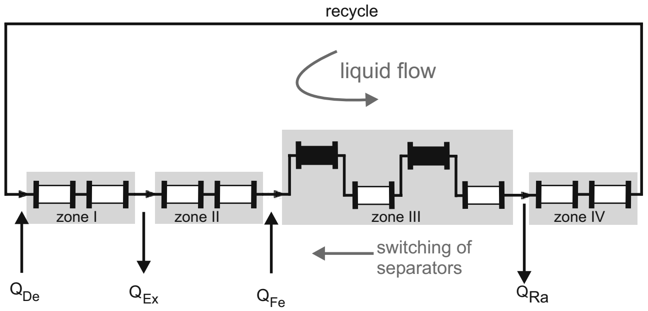


Fig. 1. Hashimoto configuration (reactors: black, separators: white)

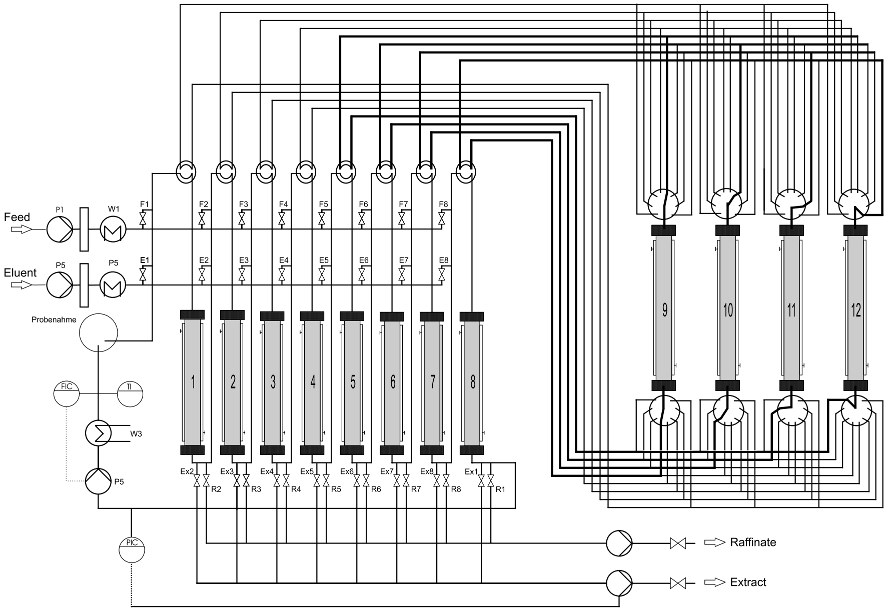


Fig. 2. Flowchart of the Hashimoto process

flow chart of the SMB plant operated at the Universität Dortmund. The ports for the external feed and eluent inlets as well as the extract and raffinate outlets can be connected to each single chromatographic column (1-8). Each reactor (9-12) can be placed in front of each chromatographic column.

Accurate dynamic models of multi-column continuous chromatographic processes consist of dynamic models of the single chromatographic columns and of the tubular reactors and the node balances which describe the connections of the columns and of the switching of the ports. The chromatographic columns are described accurately by the *general rate model* which accounts for all important effects of the column, i.e. mass transfer between the liquid and solid phase, pore diffusion, and axial dispersion. It is assumed that the particles of the solid phase are uniform, spherical, porous (with a constant void fraction ϵ_p), and that the mass transfer between the particle and the surrounding layer of the bulk is in a local equilibrium. The concentration of component i is given by c_i in the liquid phase and by q_i in the solid phase. D_{ax} is the axial dispersion coefficient, u the interstitial velocity, ϵ_b the void fraction of the bulk phase, c_i^{eq} the equilibrium concentration, $k_{l,i}$ the film mass transfer resistance, and $D_{p,i}$ the diffusion coefficient within the particle pores. The concentration within the pores is denoted by $c_{p,i}$. Furthermore, it is assumed that u and c_i are uniformly distributed over the radius. The following set of partial differential equations for the separators and the tubular reactors can be obtained from a mass balance around an infinitely small cross-section of the column (TB- is referred to as A, while TB+ is denoted as B):

Separator

$$\frac{\partial c_{b,i}}{\partial t} + \frac{(1-\epsilon_b)3k_{l,i}}{\epsilon_b R_p}\,(c_{b,i} - c_{p,i}|_{r=R_p}) = D_{ax}\,\frac{\partial^2 c_{b,i}}{\partial z^2} - u\frac{\partial c_{b,i}}{\partial z} \tag{1}$$

$$(1-\epsilon_p)\frac{\partial q_i}{\partial t} + \epsilon_p \frac{\partial c_{p,i}}{\partial t} - \epsilon_p D_{p,i}\left[\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial c_{p,i}}{\partial r}\right)\right] = 0\,, \tag{2}$$

Reactor

$$\frac{\partial c_{b,i}}{\partial t} + r^{\text{liq}}_{\text{kin},i} = D_{ax}\,\frac{\partial^2 c_{b,i}}{\partial z^2} - u\frac{\partial c_{b,i}}{\partial z} \tag{3}$$

with appropriate initial and boundary conditions

$$c_{b,i}|_{t=0} = c_{b,i}(t=0,z), \quad c_{p,i}|_{t=0} = c_{p,i}(t=0,z,r), \tag{4}$$

$$\left.\frac{\partial c_{b,i}}{\partial z}\right|_{z=0} = \frac{u}{D_{ax}}(c_{b,i} - c_i^{in}), \quad \left.\frac{\partial c_{b,i}}{\partial z}\right|_{z=L} = 0, \tag{5}$$

$$\left.\frac{\partial c_{p,i}}{\partial r}\right|_{r=0} = 0, \quad \left.\frac{\partial c_{p,i}}{\partial r}\right|_{r=R_p} = \frac{k_{l,i}}{\epsilon_p D_{p,i}}(c_{b,i} - c_{p,i}|_{r=R_p})\,. \tag{6}$$

The adsorption equilibrium and the reaction kinetics have been determined experimentally in [11]. The adsorptive behaviour can be modelled best by an asymmetric multi-component Langmuir isotherm

$$q_i = \frac{H_i c_i}{1 + \sum\limits_j b_{i,j} c_j} \quad i = A, B, \tag{7}$$

where H_i denotes the Henry coefficient which dominates the adsorption. The racemization of Tröger's base is regarded as homogeneous reaction described by first order kinetics:

$$r^{liq}_{kin,i} = \nu_i k_m (c_{b,i} - c_{b,j}) \quad i,j = A,B \qquad i \neq j. \tag{8}$$

From mass and concentration balances, the relations at the inlet and the outlet nodes result as

$$\text{Desorbent node} \quad Q_{IV} + Q_{De} = Q_I \tag{9}$$

$$c^{out}_{i,IV} Q_{IV} = c^{in}_{i,I} Q_I \qquad i = A,B \tag{10}$$

$$\text{Extract node} \quad Q_I - Q_{Ex} = Q_{II} \tag{11}$$

$$\text{Feed node} \quad Q_{II} + Q_{Fe} = Q_{III} \tag{12}$$

$$c^{out}_{i,II} Q_{II} + c_{i,Fe} Q_{Fe} = c^{in}_{i,III} Q_{III} \qquad i = A,B \tag{13}$$

$$\text{Raffinate node} \quad Q_{III} - Q_{Ra} = Q_{IV}, \tag{14}$$

where $Q_{I,II,III,IV}$ denote the internal flow rates through the corresponding zones I, II, III, IV, Q_{De}, Q_{Ex}, Q_{Fe}, and Q_{Ra} are the external flow rates of the inlet/outlet ports, respectively, and $c_{i,j}^{out}$ and $c_{i,j}^{in}$ denote the concentrations of the component i in the stream leaving or entering the respective zone j.

As eluent, an equimolar mixture of acetic acid and 2-Propanol is utilized. Acetic acid is the catalyst with a high activity at a temperature of 80 $°C$, but negligible activity at room temperature. Thus, the reactors are operated at 80 $°C$ while the separators are operated at room temperature. The separators are packed with the adsorbent Chiaralcel. An efficient numerical solution approach is used as proposed in [9] where a finite element discretization of the bulk phase is combined with orthogonal collocation of the solid phase.

3 Predictive Control

3.1 Online Optimization

The basic idea of the control algorithm is to perform an optimization of the operational degrees of freedom at future switching periods based upon a rigorous model of the plant with respect to an economic cost function (rather than e. g. a cost function involving a tracking error) in which the specifications of the SMB process (purity requirements, limitations of the pumps) as well as the process dynamics are handled as constraints. The inputs located within the control horizon H_C are considered as degrees of freedom of the optimization while the remaining inputs within the larger prediction horizon H_P are set equal to the values in the final control interval. The computed inputs in the first sampling interval are applied to the plant, and the optimization is then repeated for the next time interval with the control and prediction horizon shifted forward by one time interval, using new measurement data and eventually new estimated model parameters. In the application of optimizing control to SMB processes, the sampling time is chosen equal to the length of a cycle (length of a switching period times the number of chromatographic columns) and hence varies during the operation of the process. Due to the slow dynamic response of the concentration profiles of SMB processes to changes in the operating parameters, a modern PC is sufficient to solve the online optimization problems within a process cycle. We here consider a four-zone Hashimoto SMB process with raffinate flow that is described by the nonlinear discrete dynamics (16), (17). The objective of the optimizing controller is to minimize the eluent consumption Q_{De} for a constant feed flow and a given purity requirement of 99% in the presence of a plant/model mismatch. The inevitable mismatch between the model and the behavior of the real plant is taken into account by feedback of the difference of the predicted and the measured product purities. A regularization term is added to the objective function (15) to obtain smooth trajectories of the input variables. The controller has to respect the purity requirement for the extract flow (18) which is averaged over the prediction horizon, the dynamics of the Hashimoto SMB model (16), (17) and the maximal flow rate in zone I (20) due to limited pump capacities

(21). In order to guarantee that at least 70% of the mass of the components fed to the plant averaged over the prediction horizon leaves the plant in the extract product stream, an additional productivity requirement (19) is added. The deviation between the prediction of the model and the plant behavior is considered by the error feedback term (23). The resulting mathematical formulation of the optimization problem is:

$$\min_{\beta_I,\beta_{II},\beta_{III},\beta_{IV}} \quad \sum_{i=1}^{H_P} Q_{De,i} + \Delta\beta R \Delta\beta \tag{15}$$

$$s.t. \quad x^i_{smb} = x^i_{smb,0} + \int\limits_{t=0}^{\tau} f_{smb}(x_{smb}(t), u(t), p)dt \tag{16}$$

$$x^{i+1}_{smb,0} = M x^i_{smb,\tau} \tag{17}$$

$$\frac{\sum\limits_{i=1}^{H_P} Pur_{Ex,i}}{H_P}^{*} \geq \left(Pur^*_{Ex,min} - \Delta Pur_{Ex}\right) \tag{18}$$

$$\frac{\sum\limits_{i=1}^{H_P} m_{Ex,i}}{H_P} \geq 0.7 m_{Fe} - \Delta m_{Ex} \tag{19}$$

$$Q_I \leq Q_{max} \tag{20}$$

$$Q_{De}, Q_{Ex}, Q_{Fe}, Q_{Re} \geq 0, \tag{21}$$

where M is the shifting matrix, τ the period length. The extract purity, the purity error, the mass output, and the mass error are evaluated according to:

$$Pur_{Ex} = \frac{\int\limits_{t=0}^{\tau} c_{Ex,A} dt}{\int\limits_{t=0}^{\tau} (c_{Ex,A} + c_{Ex,B}) dt} \tag{22}$$

$$\Delta Pur_{Ex} = Pur^*_{Ex,plant,i-1} - Pur^*_{Ex,model,i-1} \tag{23}$$

$$m_i = \frac{\int\limits_{0}^{\tau} (c_{i,A} + c_{i,B}) Q_i dt}{\tau} \tag{24}$$

$$\Delta m_{Ex} = m_{Ex,plant,i-1} - m_{Ex,model,i-1}. \tag{25}$$

Since the plant is operated close to 100% extract purity, the purities are scaled $(*)$ according to

$$purity^* = \frac{1}{1 - purity}, \tag{26}$$

that provides a large slope in the scaled purity for very high purities. The numerical tractability is improved by translating the degrees of freedom (period length τ, desorbent flow Q_De, extract flow Q_{Ex}, and recycle flow Q_{Re}) into the so-called beta factors [2] that relate the liquid flow rates Q_i in each separation zone to the simulated solid flow rate Q_s.

$$Q_s = \frac{(1-\epsilon)V_{col}}{\tau} \qquad \frac{1}{\beta_{III}} = \frac{1}{H_A}\left(\frac{Q_{III}}{Q_s} - \frac{1-\epsilon}{\epsilon}\right) \tag{27}$$

$$\beta_I = \frac{1}{H_A}\left(\frac{Q_I}{Q_s} - \frac{1-\epsilon}{\epsilon}\right) \qquad \frac{1}{\beta_{IV}} = \frac{1}{H_B}\left(\frac{Q_{IV}}{Q_s} - \frac{1-\epsilon}{\epsilon}\right). \tag{28}$$

$$\beta_{II} = \frac{1}{H_B}\left(\frac{Q_{II}}{Q_s} - \frac{1-\epsilon}{\epsilon}\right) \tag{29}$$

A feasible path SQP solver is applied to solve the optimization problem. The solver generates a feasible point before it minimizes the objective function. Since the SQP algorithm is a gradient based method, an additional constraint

$$\frac{\sum_{i=1}^{8} Pur_{Ex,i}}{8} \leq 99.9\% \tag{30}$$

is added that enforces the purity to be below 99.9% averaged over a cycle and prevents the purity from reaching 100 % at which the gradient information for the constraint (18) is lost.

3.2 Parameter Estimation

The parameter estimation scheme is based on a measurement device that is fixed behind the physical separation column positioned in front of the recycle line. The recycle measurements are collected over one cycle and simulated by the model to estimate the reaction rate constant k_m and the Henry coefficients H_A and H_B via a least-squares minimization according to:

$$\min_p \sum_{i=A}^{B} \left(\int_0^N (c_{i,meas}(t) - c_{i,Re})^2 dt \right), \tag{31}$$

where N is the number of measurement points.

4 Results

For the simulative run presented here, a column distribution as shown in Figure 1 is assumed (1400 states). The sampling time is set to one cycle (8 periods). The prediction horizon H_P and the control horizon H_C have a length of 6 intervals and 1 interval, respectively. The regularization terms R_i are set to 0.3 for each control variable. Both, controller and estimator, are started at the 72^{nd} period. In the control scenario, an exponential decrease of the catalyst activity is assumed that occurs in the case of a malfunction of the reactor heating. A

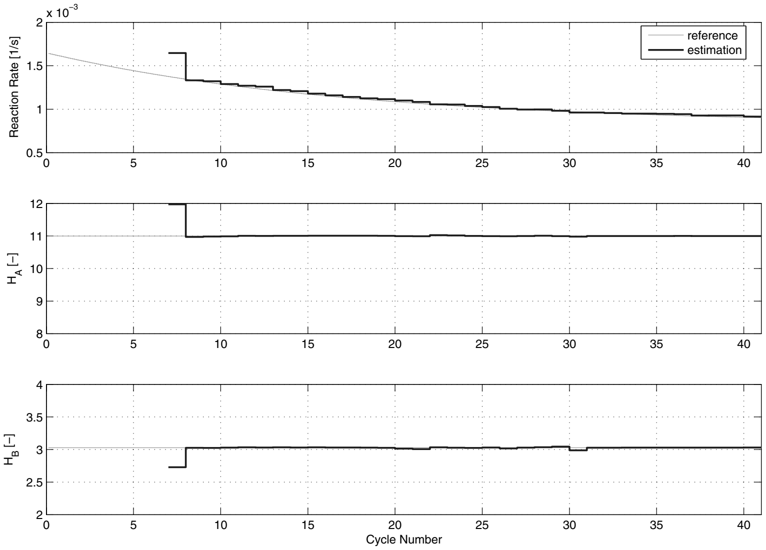


Fig. 3. Estimation of the reaction rate constant and Henry coefficients

separator:

separator length	$L_s = 10cm$	particle diameter	$d_p = 10\mu m$
separator diameter	$D_s = 1cm$	axial diffusion coefficient	D_{ax} acc. to [12]

adsorption coefficients	$H_A = 10.997$	**reactor:**	
	$b_{1,1} = 0.132\frac{l}{g}$	reactor length	$L_r = 100cm$
	$b_{1,2} = 0.543\frac{l}{g}$	reactor diameter	$D_r = 0.53cm$
	$H_B = 3.028$	reaction rate coef.	$k_m = 0.001645\frac{1}{s}$
	$b_{2,1} = 3.413\frac{l}{g}$	stoichiometry	$\nu = [-1; +1]$
	$b_{2,2} = 0.0\frac{l}{g}$	**eluent**:	
film transfer resistance	$k_{l,A} = 0.000302\frac{cm}{s}$	density	$\rho = 0.81867\frac{g}{ml}$
	$k_{l,B} = 0.000302\frac{cm}{s}$	viscosity	$\eta = 0.0271\frac{g}{cms}$
column void fraction	$\epsilon_b = 0.387$	**feed:**	$Q_{Fe} = 0.61\frac{ml}{min}$
particle void fraction	$\epsilon_p = 0.4$		$c_{A,Fe} = 2.5\frac{g}{l}$
overall void fraction	$\epsilon = 0.632$		$c_{B,Fe} = 2.5\frac{g}{l}$
particle diffusion coefficient	$D_p = 0.001\frac{cm^2}{s}$		

further plant/model mismatch is introduced by disturbing the initial Henry coefficients H_A and H_B of the model by +10% and -10%. Figure 3 shows that the parameter estimation scheme estimates the parameters of the plant well. The performance of the controller is illustrated by Figure 4. The controller manages to keep the purity and the productivity above their lower limits, while it improves the economic operation of the plant by reducing the solvent consumption. The

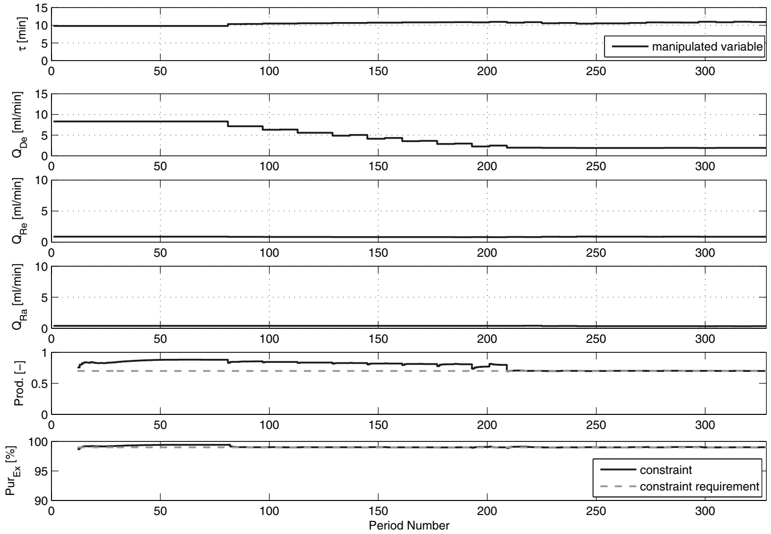


Fig. 4. Simulation of the predictive controller; manipulated variables and constraints

process converges to a stationary operating point. The optimizer converges to the optimum within one sampling time, and hence, can be applied in real-time.

5 Conclusion

An online model-based optimization and parameter estimation scheme for the Hashimoto Simulated Moving Bed process has been presented. The approach has the advantage that the process is automatically operated at its economic optimum while the purity and productivity requirements and plant limitations are fulfilled. In future research, the application to a pilot-plant is planned.

Acknowledgement

We would like to thank Larry Biegler for a fruitful discussion about how to avoid zero gradient information for a SQP solver. This investigation was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant DFG En 152/34. This support is gratefully acknowledged.

References

[1] Hashimoto, K., Adachi, S., Noujima, H. and Ueda, Y., “A New Process Combining Adsorption and Enzyme Reaction for Producing Higher-Fructose Syrup”, *Biotechnology and Bioengineering*, **Vol. 25**, pp. 2371-2393, (1983).

[2] Hashimoto, K., Adachi, S. and Shirai, Y., "Development of a new bioreactors of a simulated moving-bed type", *in* Ganetsos G. & Barker P. (eds.), *Preparative and Production Scale Chromatography*, Marcel Dekker, New-York, pp. 395-417, (1993).

[3] Engell, S. and Toumi, A., "Optimization and Control of Chromatography", *Computers and Chemical Engineering*, **Vol. 29**, pp. 1243-1252, (2005).

[4] Toumi, A., Diehl, M., Engell, S., Bock, H. G. and Schlöder, J., "Finite horizon optimizing of control advanced SMB chromatographic processes", *IFAC World Congress*, Fr-M06-TO/2, (2005).

[5] Erdem, G., Abel, S., Morari, M. and Mazzotti, M., "Automatic control of simulated beds", *Ind. and Eng. Chem. Res.*, **Vol. 43, No. 2**, pp. 405-421, (2004).

[6] Erdem, G., Abel, S., Morari, M. and Mazzotti, M., "Automatic Control of simulated moving beds-II: Nonlinear Iso-therms", *Ind. and Eng. Chem. Res.*, **Vol. 43, No. 14**, pp. 3895-3907, (2004).

[7] Toumi, A. and Engell, S., "Optimal operation and control of a reactive simulated moving bed process", *in* Allgöwer, F. & Gao, F. (eds.), 7^{th} *IFAC International Symposium on Advanced Control of Chemical Processes*, Hongkong, pp. 243-248, (2004).

[8] Toumi, A. and Engell, S., "Optimization-based control of a reactive simulated moving bed process for glucose isomerization", *Chemical Engineering Science*, **Vol. 59**, pp. 3777-3792, (2004).

[9] Gu, T., "Mathematical Modelling and Scale Up of Liquid Chromatography", *Springer Verlag*, New York, (1995).

[10] Küpper, A. and Engell, S., "Parameter and State Estimation in Chromatographic SMB Processes with Individual Columns and Nonlinear Adsorption Isotherms", to be presented at *IFAC ADCHEM*, (2006).

[11] Borren, T., Fricke, J. and Schmidt-Traub, H., "Reactive liquid chromatography", *in* Schmidt-Traub, H. & Górak, A. (eds.), *Process Intensification by Integrated Reaction and Separation Operation*, in preparation, *Springer Verlag*, Berlin (2006).

[12] Chung, S. and Wen, C., "Longitudinal diffusion of liquid flowing through fixed and fluidized beds", *AIChE Journal*, **Vol. 14**, pp. 875-866, (1968).

Receding-Horizon Estimation and Control of Ball Mill Circuits

Renato Lepore[1], Alain Vande Wouwer[1], Marcel Remy[1], and Philippe Bogaerts[2]

[1] Service d'Automatique, Faculté Polytechnique de Mons, Boulevard Dolez 31,
B-7000 Mons, Belgium
{renato.lepore,alain.vandewouwer,marcel.remy}@fpms.ac.be

[2] Chimie Générale et Biosystèmes, Université Libre de Bruxelles, Avenue F.D.
Roosevelt 50, B-1050 Bruxelles, Belgium
philippe.bogaerts@ulb.ac.be

Summary. This paper focuses on the design of a nonlinear model predictive control (NMPC) scheme for a cement grinding circuit, i.e., a ball mill in closed loop with an air classifier. The multivariable controller uses two mass fractions as controlled variables, and the input flow rate and the classifier selectivity as manipulated variables. As the particle size distribution inside the mill is not directly measurable, a receding-horizon observer is designed, using measurements at the mill exit only. The performance of the control scheme in the face of measurement errors and plant-model mismatches is investigated in simulation.

1 Introduction

In cement manufacturing, the grinding process transforms the input material (usually clinker) into a very fine powder (the final product). This process consists of a ball mill in closed loop with an air classifier, where the feed flow rate and the classifier selectivity are used as manipulated variables. The quality indicator used in common practice, which is related to the cement fineness, is the powder specific area or Blaine measurement. Alternative quality indicators can however be defined in terms of the particle size distribution, as further discussed in this study.

Cement grinding circuits can be regulated using standard linear or more advanced nonlinear control schemes [4, 9, 12]. However, most of the control studies reported in the literature consider mass variables only, i.e., mass hold-up of the mill and mass flow rates, whereas control of the product quality requires the consideration of the particle size distribution (or at least, of some related variables). In this connection, a quality control strategy should allow to act on the particle size distribution, as well as to face rapid modification in customer demand (i.e. changes in the cement grade or quality). Hence, efficient setpoint changes have to be achieved despite the process nonlinearities and operating limitations. Such a control strategy appears as an appealing (but challenging) alternative to expensive storage policies.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 485–493, 2007.
springerlink.com

In this study, a control scheme is proposed that takes these objectives and constraints into account. The following ingredients are involved in the control design:

- A nonlinear population model describes the dynamic evolution of the mass fractions in three size intervals.
- The mass fractions at the mill outlet and at the product outlet (i.e. the air classifier outlet) can easily be measured in practice using classical sieving techniques. In addition, they can be used, as an interesting alternative to Blaine measurements, to assess the cement quality and performance of the mill.
- A model-based predictive controller is designed in order to achieve quality control and setpoint changes. This control scheme accounts for actuator saturation (in magnitude and rate of change) and for operating and safety constraints, such as mill plugging and temperature increase.
- To reconstruct on-line the particle size distribution (in three size intervals), a receding-horizon observer is designed, which uses measurements available at the mill exit only. This software sensor takes the measurement errors into account and determines the most-likely initial conditions of the prediction horizon.

In previous works [6, 7], the authors have reported on the design of the multivariable controller and of the receding-horizon observer. Here, the main purpose is to study the performance of the combined scheme (i.e. controller + software sensor) in the face of measurement noise and parametric uncertainties. In addition, a DMC-like correction scheme is proposed, which significantly improves the performance of the control strategy in the case of plant-model mismatches. A simulation case study, corresponding to a typical setpoint change, is used to highlight the advantages and limitations of the proposed strategy.

This paper is organized as follows. Section 2 briefly describes the process and the nonlinear model. In Section 3, the control objectives are discussed, the NMPC strategy is introduced, and the software sensor is presented. The combined scheme is evaluated in Section 4 and conclusions are drawn in Section 5.

2 Process Description and Modelling

A typical cement grinding circuit is represented in Figure 1, which consists of a single-compartment ball mill in closed loop with an air classifier. The raw material flow q_C is fed to the rotating mill where tumbling balls break the material particles by fracture and attrition. At the other end, the mill flow q_M is lifted by a bucket elevator into the classifier where it is separated into two parts: the product flow q_P (fine particles) and the rejected flow q_R (coarse particles). The selectivity of the classifier, i.e. the separation curve, influences the product quality. This selectivity can be modified by positioning registers *Reg* acting on the upward air flow. The material flow q_R is recirculated to the mill inlet and the sum of q_C and q_R is the total flow entering the mill, denoted q_F.

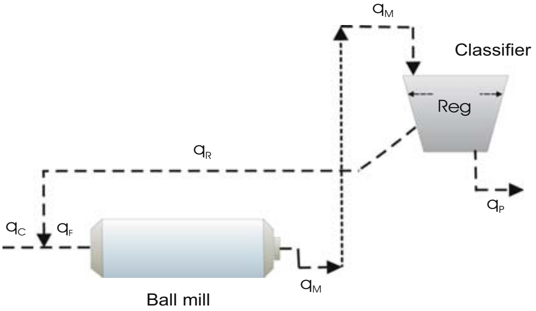


Fig. 1. Closed-loop grinding circuit

A simple population model [5, 6, 7] is used to describe the dynamic evolution of the particle size distribution in three (relatively large) size intervals. This model describes material transportation and breakage inside the ball mill, as well as material separation in the air classifier. This model consists of a set of Partial Differential Equations (PDEs), supplemented by Initial Conditions (ICs) and Boundary Conditions (BCs):

$$\frac{d\mathbf{X}}{dt} = \mathbf{f}_{PDE}\left(\mathbf{X}, \frac{\partial \mathbf{X}}{\partial z}, \frac{\partial^2 \mathbf{X}}{\partial z^2}; \boldsymbol{\theta}_f, \boldsymbol{\theta}_t\right) \tag{1a}$$

$$\mathbf{X}(t_0, z) = \mathbf{X}_0(z), \qquad 0 \le z \le L \tag{1b}$$

$$\mathbf{0} = \mathbf{f}_{BC}\left(\mathbf{X}, \frac{\partial \mathbf{X}}{\partial z}, \mathbf{u}; \boldsymbol{\theta}_t, \boldsymbol{\theta}_{cl}\right), \qquad z = 0, \forall t \tag{1c}$$

$$\mathbf{0} = \frac{\partial \mathbf{X}}{\partial z}, \qquad z = L, \forall t \tag{1d}$$

where the state vector $\mathbf{X}(t, z)$ has 3 components, the k^{th} component being the mass per unit of length (e.g., in tons per meter) in size interval k, at time t and at location z along the mill (L is the mill length). $\mathbf{X}_0(z)$ is the initial-value spatial profile. The input $\mathbf{u}$ has 2 components, q_C and Reg. The parameters $\boldsymbol{\theta}_f$, $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{cl}$ are related to the description of the fragmentation, transportation and classification mechanisms, respectively.

This PDE system is solved numerically using a method of lines [13] strategy. The spatial derivatives are replaced by finite difference approximations, and the resulting system of differential algebraic equations is integrated in time using a readily available solver.

Partitioning the size continuum into three size intervals allows the problem dimension to be reduced as compared to a more detailed description of the particle size distribution (in classical modelling studies 20-30 intervals are considered). Moreover, this specific partition can be directly related to the control objectives, as explained in the next section. In this study, the interval limits (i.e., the upper bounds of the mid-size and small-size intervals) are chosen as 100 and 30 μm.

3 Control Strategy

3.1 Control Objectives and NMPC Scheme

The feed flow rate and the classifier selectivity can be used as manipulated variables. Two mass fractions are used as controlled variables. The first one, denoted w_P^3, corresponds to the fine particles in the product flow. Experimental studies [10] demonstrate that this variable is highly correlated with the compressive strength of the cement, if the upper size of interval 3 is chosen around 30 μm. The second one, denoted w_M^2, corresponds to the mid-size particles in the mill outflow, which can be directly related to the grinding efficiency of the mill (too fine particles correspond to overgrinding whereas too coarse particles correspond to undergrinding).

The use of these several variables is illustrated by the steady-state diagram $w_M^2 = f(q_P, w_P^3)$ of Figure 2, where the curve $\overline{ABC}$ represents all the operating points with $w_P^3 = 0.86$. Clearly, point B corresponds to a maximum product flow rate and, as demonstrated in [3], the arcs $\overline{AB}$ and $\overline{BC}$ correspond to stable and unstable process behaviours, respectively. By setting, for example, $w_M^2 = 0.35$ on arc $\overline{AB}$, a single operating point (point 1) is defined. This corresponds to producing cement of a given fineness ($w_P^3 = 0.86$) at near maximum product flow rate in the stable region. A significant advantage of these controlled variables is that the measurement of mass fractions is simple and inexpensive. A classical sieving technique is used instead of sophisticated (and costly) laser technology.

The design of the NMPC scheme [1, 8] is based on a nonlinear optimization problem, which has to be solved at each sampling time $t_k = kT_s$ (where T_s is the sampling period). More specifically, a cost function measuring the deviation of the controlled variables from the setpoint over the prediction horizon has to be minimized. Denoting $\mathbf{y} = [w_P^3 \; w_M^2]^T$ the controlled variable, the optimization problem is stated as follows:

$$\min_{\{\mathbf{u}_i\}_0^{Nu-1}} \sum_{i=1}^{Np} \{\mathbf{y}^s - \hat{\mathbf{y}}(t_{k+i})\}^T \, \mathbf{Q}_i \, \{\mathbf{y}^s - \hat{\mathbf{y}}(t_{k+i})\} \tag{2}$$

where Nu and Np are the control and prediction horizon lengths, respectively (number of sampling periods with $Nu < Np$). $\{\mathbf{u}_i\}_0^{Nu-1}$ is the sequence of control moves with $\mathbf{u}_i = \mathbf{u}_{Nu-1}$ for $i \geq Nu$ ($\mathbf{u}_i$ is the input applied to the process model from t_{k+i} to t_{k+i+1}). $\hat{\mathbf{y}}(t_{k+i})$ is the output value at time t_{k+i}, as predicted by the model. $\{\mathbf{Q}_i\}_1^{Np}$ are matrices of dimension 2, weighting the coincidence points. $\mathbf{y}^s$ is the reference trajectory (a piecewise constant setpoint in our study).

In addition, the optimization problem is subject to the following constraints:

$$\mathbf{u}^{min} \leq \mathbf{u}_i \leq \mathbf{u}^{max} \tag{3a}$$

$$-\Delta\mathbf{u}^{max} \leq \Delta\mathbf{u}_i \leq +\Delta\mathbf{u}^{max} \tag{3b}$$

$$q_M^{min} \leq q_M \leq q_M^{max} \tag{3c}$$

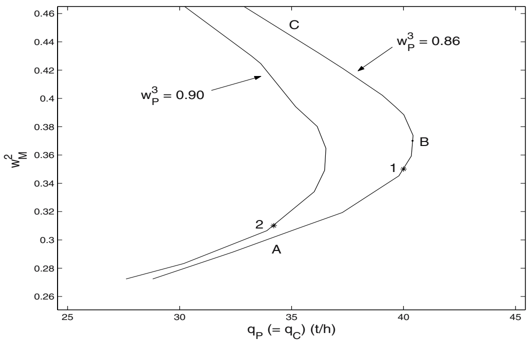


Fig. 2. Steady-state diagram: w_M^2 vs q_P for constant values of w_P^3

On the one hand, equation (3a) represents bound constraints on the manipulated variables, e.g., saturation of the feeding mechanism or in the displacement of the registers, whereas equation (3b) corresponds to limitations of the rate of change of these manipulated variables. On the other hand, equation (3c) expresses constraints on an operating variable, e.g., a lower bound on the mill flow rate to prevent mill emptying and temperature increase, and an upper bound to avoid mill plugging or a drift into the instability region.

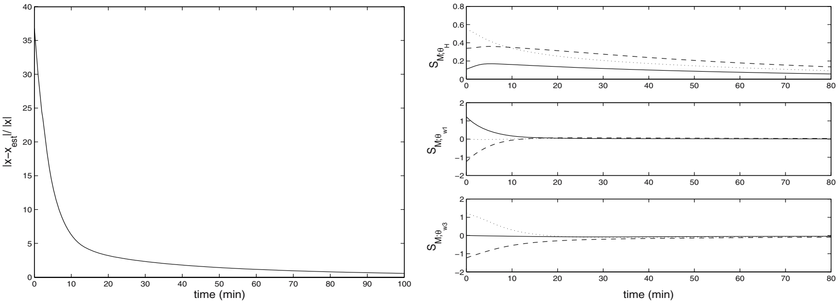
3.2 Software Sensor Design

As the particle size distribution inside the mill is not accessible, a receding-horizon observer [1, 11] is designed, based on the nonlinear process model and a few measurements available at the mill exit. The main advantages of this type of software sensors are that a nonlinear model of arbitrary complexity can be used, and the measurement errors can be taken into account rigorously. An estimate of the most-likely initial conditions, noted $\hat{\mathbf{x}}(0|t_k)$, is obtained by solving the following optimization problem:

$$\min_{\mathbf{x}_0} \sum_{i=k-No}^{k} \left\{\mathbf{y}(t_i) - \mathbf{h}_{obs}(\mathbf{x}(t_i), \mathbf{u}(t_i))\right\}^T \boldsymbol{\Sigma}_i^{-1} \left\{\mathbf{y}(t_i) - \mathbf{h}_{obs}(\mathbf{x}(t_i), \mathbf{u}(t_i))\right\}, \quad (4)$$

where $\mathbf{x}_0$ is the initial-condition, No is the prediction horizon length (number of sampling periods $T_o \neq T_s$), $\mathbf{h}_{obs}(.)$ is the output trajectory, $\mathbf{y}(t_i)$ are the measurements affected by a Gaussian white noise with zero mean and covariance matrix $\boldsymbol{\Sigma}_i$.

However, the finite difference schemes used to solve the model equations lead to a relatively large number of state variables which should be estimated. To circumvent this problem, and to keep the optimization problem tractable, the initial condition profile $\mathbf{x}_0(z)$ is approximated by a combination of simple polynomial laws. An approximate representation of the initial spatial profile is sufficient for state estimation and control purposes, as the effect of the initial conditions on



(a) Norm percentage deviation between the real and simulated state

(b) Parametric sensitivity of the mass fractions at the mill outlet; size interval 1:solid, 2:dashed, 3:dotted

Fig. 3. Receding-horizon estimation: evaluating the impact of the initial conditions

the model solution vanishes rapidly as compared to the process time constants. Figure 3(a) shows the norm percentage deviation between the real and simulated state during a typical run. For an initial deviation of about 40 %, the deviation reduces to 5 % in only 10 min.

To build the polynomial approximation, $\mathbf{x}_0(z)$ is expressed in terms of z_r, a scaled spatial coordinate $\left(\dfrac{z}{L}\right)$, and in terms of the mill material hold-up $H_0(z_r)$ and the mass fractions $w_0^i(z_r)$

$$\mathbf{x}_0(z_r) = H_0(z_r)\left[w_0^1(z_r)\; w_0^2(z_r)\; w_0^3(z_r)\right]^T \tag{5}$$

Simple polynomial laws are then used to represent the several factors of this latter expression. The hold-up is considered uniform $H_0(z_r) = \theta_H$, the mass fraction of coarse particles can be represented by a concave quadratic law $w_0^1(z_r) = 0.3\, z_r(z_r - 2) + \theta_{w1}$, the mass fraction of fine particles can be represented by a convex quadratic law $w_0^3(z_r) = -0.3\, z_r(z_r - 2) + \theta_{w3}$ and $w_0^2(z_r)$ is simply deduced from the two other mass fractions (and is uniform in the present case).

When measurements are available at the mill exit only, it is observed that the relevant information for the determination of θ_{w1} and θ_{w3} vanishes after 10 min. This is apparent in Figure 3(b), which shows the parametric sensitivity of the material mass fractions at the mill outlet. For instance, $S_{M;\theta_H}(t)$ is defined as $\frac{\partial \mathbf{X}}{\partial \theta_H}(L, t)$.

These results justify the use of a simple parameterization of the initial condition profile. In optimization problem (4), a horizon of 20 min with 2 min-sampling intervals is sufficient to ensure convergence and accuracy. On the other hand, the formulation of the state vector using the factorization (5) results in the consideration of simple linear constraints.

The computation time required to solve the optimization problems (2) and (4) would allow the use of shorter sampling intervals, but the measurement procedure (sieving) could be limitative. Here, we have elected to be on the safe side concerning this latter limitation (but shorter sampling intervals would of course improve accuracy and convergence).

4 Numerical Results

In this section, the combined scheme (software sensor + NMPC) is evaluated in simulation, using a typical test run corresponding to a setpoint change. Point 1 in Figure 2 (where $\mathbf{y} = [0.86\ 0.35]^T$) is the initial operating condition, and point 2 (with $\mathbf{y} = [0.90\ 0.31]^T$) represents the target (this point corresponds to a higher product fineness and near maximum product flow rate). The sampling period $T_s = 5$ min and the prediction horizon is 80 min ($Np = 16$). Two manipulated variable moves are used ($Nu = 2$) and the weighting matrix $\mathbf{Q}_i$ is chosen as a constant identity matrix. Amplitude saturations are $q_C^{max} = 60\ \frac{\text{ton}}{\text{hour}}$ and $Reg^{max} = 100$. Limits for the rates of change are $\Delta q_C^{max} = 15\frac{\text{ton}}{\text{hour}}$ and $\Delta Reg^{max} = 80$. Limits on the mill flow rate are $q_M^{min} = 60\ \frac{\text{ton}}{\text{hour}}$ and $q_M^{max} = 90\ \frac{\text{ton}}{\text{hour}}$. The observer parameters are defined in Section 3.2.

It is first assumed that the process model is accurate and that the measurements are noise free. Figure 4 shows the controlled and the manipulated variables (solid lines). The performance is satisfactory, the controlled variables reach the setpoint after about 20 min and the steady state is obtained after 70 min. Moreover, constraints are active in the first 5 min (first sample), as the maximum register displacement and the maximum rate of change of the input flow rate are required.

The performance of the control scheme is then tested when measurements are subject to a noise with a maximum absolute error of 0.02 $\frac{\text{ton}}{m}$ (around a 5% maximum relative error). The software sensor can efficiently take these stochastic disturbances into account, and the performance of the control scheme remains quite satisfactory.

Finally, the influence of parametric uncertainties (plant-model mismatch due to errors at the identification stage) is investigated. A parametric sensitivity analysis is performed, and Figure 5 shows step responses corresponding to either an accurate model or to a −10% error in the fragmentation rate or the transport velocity. Clearly, fragmentation parameters (which represents material hardness

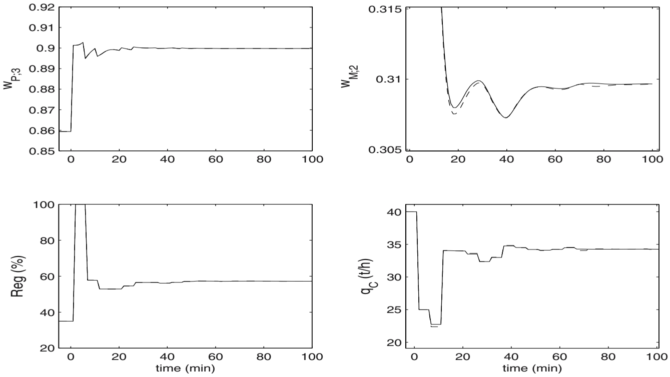


Fig. 4. NMPC and state observer; solid: noise-free measurements, dashed: measurements corrupted by white noise

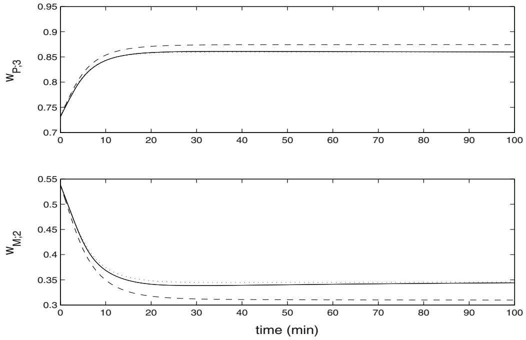


Fig. 5. Parametric sensitivity - Time evolution of the controlled variables; solid: nominal case, dashed: mismatch in the fragmentation rate, dotted: mismatch in the transport velocity

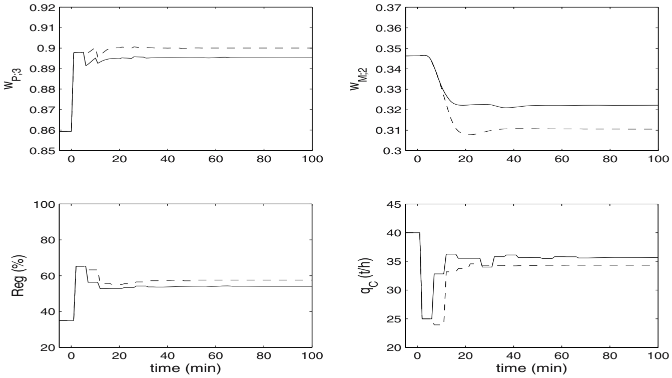


Fig. 6. Plant-model mismatch: solid: without compensation, dotted: with a DMC-like compensation

or grindability) have a larger impact on the model prediction than the material transportation parameters.

The performance of the control scheme is evaluated when a prediction model with erroneous fragmentation rates is used (which is the worst case of plant-model mismatch). Figure 6 shows results corresponding to −5% errors in the fragmentation rates. Clearly, performance deteriorates and a significant steady-state error appears. To alleviate this problem, a DMC-like compensation is proposed, which considers the plant-model mismatch as a constant output disturbance $\hat{\mathbf{d}}_{k+i} = \hat{\mathbf{d}}_k$ over the prediction horizon. An estimate of the disturbance $\hat{\mathbf{d}}_k$ is obtained from the process output $\mathbf{y}_k$ and the observer output, noted $\bar{\mathbf{y}}_k$, as follows:

$$\hat{\mathbf{d}}_k = \mathbf{y}_k - \bar{\mathbf{y}}_k \tag{6}$$

Figure 6 shows that this kind of compensation significantly improves the performance of the control scheme under parametric uncertainties.

5 Conclusion

In this paper, a receding-horizon observer and a receding-horizon controller are designed for a ball mill circuit. The software sensor provides an estimation of the particle size distribution inside the ball mill, based on a nonlinear process model and a few measurements available at the mill exit. The control scheme allows efficient quality control and setpoint changes, even in the face of noisy measurements and significant parametric uncertainties. In addition, a DMC-like compensation of these latter errors improves the performance of the proposed scheme.

References

[1] Allgöwer F., Badgwell T.A., Qin J.S., Rawlings J.B., Wright S.J. (1999) Nonlinear predictive control and moving horizon estimation - an introduction overview. In: Frank P.M. (ed) Advances in Control (Highlights of ECC '99). Springer, London Berlin Heidelberg

[2] Boulvin M. (2000) Contribution à la modélisation dynamique, à la simulation et à la conduite des circuits de broyage à boulets utilisés en cimenterie. PhD Thesis, Faculté Polytechnique de Mons, Mons (Belgium)

[3] Boulvin M., Vande Wouwer A., Lepore R., Renotte C., Remy M. (2003) IEEE transactions on control systems technology 11(5):715–725

[4] Grognard F., Jadot F., Bastin G., Sepulchre R., Wertz V. (2001) IEEE Transactions on Automatic Control 46(4):618–623

[5] Lepore R., Vande Wouwer A., Remy M. (2002) Modeling and predictive control of cement grinding circuits. 15th IFAC World Congress on Automatic Control, Barcelona, Spain

[6] Lepore R., Vande Wouwer A., Remy M. (2003) Nonlinear model predictive control of cement grinding circuits. ADCHEM 2003, Hong-Kong, China

[7] Lepore R., Vande Wouwer A., Remy M., Bogaerts Ph. (2004) State and parameter estimation in cement grinding circuits - practical aspects. DYCOPS 7 2004, Cambridge (MA), U.S.A.

[8] Maciejowski J.M. (2001). Predictive control: with constraints. Prentice Hall

[9] Magni L., Bastin G. and Wertz V. (1999) IEEE transactions on control systems technology 7(4):502–508

[10] Olivier L. (2002) Contribution à la conduite des circuits de broyage à boulets utilisés en cimenterie. MA Thesis. Faculté Polytechnique de Mons, Mons

[11] Rao C.V. (1999) Moving-horizon estimation of constrained and nonlinear systems. PhD Thesis. University of Wisconsin-Madison, Madison

[12] Van Breusegem V., Chen L., Bastin G., Wertz V., Werbrouck V., de Pierpont C. (1996) IEEE Transactions on Industry Applications 32(3):670–677

[13] Vande Wouwer A., Saucez P., Schiesser W.E. (2004) Industrial Engineering and Chemistry Research 43(14):3469–3477

Hybrid NMPC Control of a Sugar House

D. Sarabia[1], C. de Prada, S. Cristea, R. Mazaeda, and W. Colmenares[2]

[1] Dpt. Systems Engineering and Automatic Control. Faculty of Sciences, c/ Real de Burgos, s/n, University of Valladolid, Spain
dsarabia@autom.uva.es

[2] Universidad Simón Bolívar, Dpt. Procesos y Sistemas, Apartado 89000, Caracas 1080, Venezuela
williamc@usb.ve

1 Introduction

Plant-wide control is attracting considerable interest, both as a challenging research field and because of its practical importance. It is a topic [1] characterized by complexity in terms of the number and type of equipments involved, diversity of aims, and lack of adequate models and control policies. In this paper, the MPC control of the final part of a beet sugar factory, the so-called sugar house or sugar end, where sugar crystals are made, is presented. Perhaps the most characteristic aspect of its operation is that batch and continuous units operate jointly, which introduce the need for combining on-line scheduling with continuous control. As such, it is a hybrid process that requires non-conventional control techniques. The paper presents a methodology and a predictive controller that takes into account both, the continuous objectives and manipulated variables, as well as the ones related to the discrete operation and logic of the batch units, and, at the end, simulation results of the controller operation are provided.

Many approaches have appeared in the literature in recent years for hybrid predictive control. A natural approach integrates in a single mathematical formulation the different elements of a hybrid process by using integer variables for representing on/off decisions and integer equations for the logic relations between variables [2] besides the continuous equations. The fact that the internal model of the MPC controller includes continuous and integer variables leads to a mix-integer optimization problem [3], which in many cases is difficult and time consuming to solve.A natural way of approaching complex systems is using a hierarchical point of view, separating the problems that can be solved locally at a lower level from the ones that require a global consideration. This paper focuses on these overall decisions, and describes a controller that takes into account both continuous control of key process variables as well as the scheduling involved in the operation of the crystallizers, which operate in batch mode. The controller follows the MPC paradigm, solving a non-linear model-based optimization problem on-line every sampling time. Moreover, the problem is re-formulated in terms of prescribed patterns of the batch units variables and time of occurrence of the events (real variables), instead of using integer variables, which allows to solve

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 495–502, 2007.
springerlink.com

the optimization problem as a NLP one, saving computation time. The interest of this contribution comes not only from the fact that it is a challenging control problem, but because problems with a similar structure are present in many industrial process factories.

2 The Sugar End and Control Architecture

Sugar factories produce commercial sugar in a set of vacuum pans or "tachas" from an intermediate solution called feed syrup. Each tacha operates in a semi batch mode following a predefined sequence, which main stages are: loading of syrup; heating it with steam; concentration until supersaturation is reached; seeding and growing of the crystals until they reach the desired size and the vacuum pan is full, this stage being known as cooking, and finally unloading the massecuite or cooked mass, which is the mix of crystals and non crystallized syrup (mother liquor). The main source of variability in the operation of each vacuum pan comes from the quality of the feed syrup. The processing time increase if the percentage solid content of the syrup, which is known as brix, decrease, and crystal growth increases with the purity of the syrup, that is, the percentage of pure sacharose in the dissolved solids.

A scheme of the sugar house of the particular case that has been considered can be seen in fig. 1. From the three vacuum pans A, the cooked mass, is unloaded into an agitated heated vessel named "malaxador". From this one, the mix of mother liquor and crystals is separated by means of a set of seven centrifugals. Cooked mass from tachas type A, gives way to commercial white sugar and two kinds of syrup: the so-called lower purity syrup and the higher purity syrup. The later has a small percentage of dissolved crystals and, so, a higher purity, and its is recycled to the feeding tank (melter) of tachas A. On the contrary, the lower purity syrup is sent to another storage vessel (tank B) and processed again in one tacha named B. The proportion between both kinds of syrup can be adjusted using a timer in the local centrifugals control. In tacha B the whole process is repeated, this time with longer operation times due to the lower purity of the syrup, but with an important difference: the sugar produced in the three centrifugal separators B, sugar B, is not commercialised but recycled to a melter due to its color and impurities, while the lower purity syrup is discharged as a by-product called molasses. The overall control objectives of the sugar end section are summarised next:

1. Processing the flow of syrup coming from the previous continuous sections of the factory avoiding bottlenecks in production. This objective implies an adequate scheduling of the vacuum pans operation and a proper use of the shared resources, such as, avoiding the feeding tanks (melter and tank B) and malaxadors A and B from being either empty or overflow. This implies to maintain levels in the previous units between certain lower and upper limits.
2. Maintaining the quality of the crystals in terms of size and size distribution. This is an important objective, but it is solved locally in every vacuum pan, where the operation of the crystallization is managed in order to obtain proper conditions for sugar crystal growth.

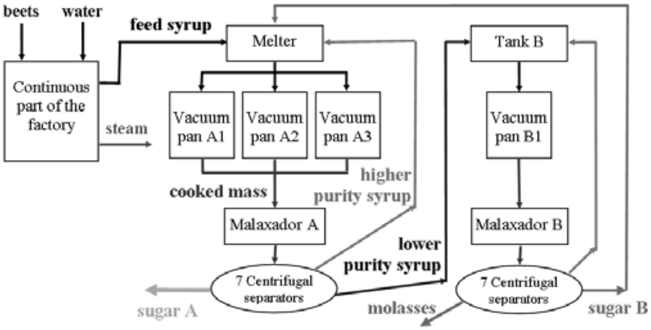


Fig. 1. A simplified scheme of the sugar end section

3. Maintaining brix and purity in melter and in tank B as close as possible to given set points, in order to maximize the amount of sugar A produced, because the processing time and capacity of the tachas depend on both variables. This has also an influence in the first objective.

From control architecture point of view, a common strategy in complex systems, is to decompose the problem in several levels or time scales, so that what can be solved locally, involving a limited set of resources or decision variables is separated from those decisions that involve variables having an effect on the dynamics of the whole system. In our case, this hierarchical decomposition recognisees at least three layers or types of control problems:

Local SISO controllers: Such as the temperature control in the malaxadors, flow controls, etc. These are managed by the Distributed Control Systems (DCS) of the plant and have fast dynamics compared with the ones of the sugar end. They are supposed to operate well using standard controllers.

Sequence control of each batch unit: Such as tachas and centrifugal separators. For every vacuum pan, this control executes the cyclical sequence of stages described at the beginning of this section, necessaries to obtain the final product: sugar (objective number 2 in the previous list). It is implemented also in the DCS as a GRAFCET chart with additional local control of pressure, level, concentration, etc, and operate according to predefined parameters and external orders, such as load and unload each tacha. These local controllers are assumed to do its best in order to complete its tasks each cycle, for instance rejecting disturbances on steam pressure or vacuum. For centrifugal separators, another GRAFCET chart executes the consecutive operations of loaded cooked mass, unloaded low purity syrup, unloaded high purity syrup and unloaded sugar. In this case, the external orders are the frequency of operation of every centrifugal separator and the percentage of lower and higher purity syrup obtained.

Plant wide-control: This layer is responsible for objectives 1 and 3 of the previous list, in spite of changes in the flow and quality of the feed syrup, and it is the objective of this paper. For this purpose, besides the scheduling of the tachas, the controller can manipulate the proportion between lower and higher purity syrup in the centrifugals and its operating frequency, which is equivalent to establishing its total processing flow. These tasks are performed very often manually by the person in charge of the section. From the point of view of this layer, the SISO controllers and sequence control can be considered as included in the process, operating in cascade over them.

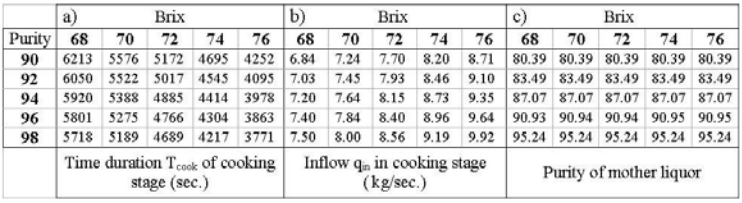
3 Hybrid Control

A natural approach to many decision problems is the one of Model Predictive Control (MPC): A model of the process is used to predict its future behavior as a function of the present and future control actions, which are selected in order to minimize some performance index. The optimal control signals corresponding to the present time are applied to the process and the whole procedure is repeated in the next sampling period.

In MPC of complex systems, it is very important that the internal model that relates controlled and manipulated variables being as simple as possible while still being a good representation of the process. On the other hand, it must correspond to the view and purpose of the plant-wide control. A full first principles model implementing mass and energy balances, as well as crystal growth and local control functions can perform this task, but this approach will lead to a huge model, useless for MPC. Consequently, the model includes only those variables and phenomena relevant to the above mentioned plant-wide control objectives. It combines dynamic mass balances of total mass, solid content and sacharose in the continuous units (feeding tanks and malaxadors) with abstractions and simplifications of the other parts of the process, tachas and centrifugals, because, what is important, is the relationship between these units and the continuous ones are given through the input and output flows and its principal characteristics like purity, brix and percentage of sugar.

A key point is then, the abstract model of the tacha. Notice that, when a tacha is started, the inflow of syrup, the flow and characteristics of the cooked mass unloaded and the time consumed in the operation depends only on the properties of the feed (purity and brix), so, the approach followed has been to use tables like the one in fig. 2 relating these main variables of the vacuum pan with the properties of its feed, purity (P) and brix (B). These tables have been obtained off-line, and for a range of reasonable operating conditions, integrating a full first principles model of the vacuum pan starting from a syrup with different values of purity and brix. For example, fig. 2 a) and b) shows the time duration and inflow of syrup of cooking stage. Also additional tables are needed, see fig. 2 c), such as the ones relating brix and purity in the feeding tank with the total cooked mass obtained, the percentage of crystals in it and brix and purity of mother liquor, and the duration of the rest of stages of the sequence.

This abstract view, makes possible includes the explicit use of the special patterns that input and output flows must follow. Fig. 3 a) and b) show the shape approximation of $q_{in}(P, B)$ and $q_{out}(P, B)$ used in the simplified model of



	a)		Brix			b)		Brix			c)		Brix		
Purity	**68**	**70**	**72**	**74**	**76**	**68**	**70**	**72**	**74**	**76**	**68**	**70**	**72**	**74**	**76**
90	6213	5576	5172	4695	4252	6.84	7.24	7.70	8.20	8.71	80.39	80.39	80.39	80.39	80.39
92	6050	5522	5017	4545	4095	7.03	7.45	7.93	8.46	9.10	83.49	83.49	83.49	83.49	83.49
94	5920	5388	4885	4414	3978	7.20	7.64	8.15	8.73	9.35	87.07	87.07	87.07	87.07	87.07
96	5801	5275	4766	4304	3863	7.40	7.84	8.40	8.96	9.64	90.93	90.94	90.94	90.95	90.95
98	5718	5189	4689	4217	3771	7.50	8.00	8.56	9.19	9.92	95.24	95.24	95.24	95.24	95.24
	Time duration T_{cook} of cooking stage (sec.)					Inflow q_{in} in cooking stage (kg/sec.)					Purity of mother liquor				

Fig. 2. Typical table obtained off-line from the first principles dynamic of a tacha

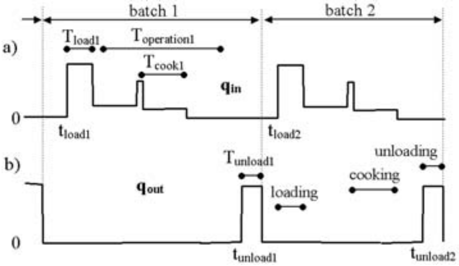


Fig. 3. a) and b) Temporal patterns of input and output flows (q_{in} and q_{out}) of the simplified model of a tacha

the vacuum pan, where two batches are predicted. For simplicity in the graphic, we have named only three stages: loading, cooking and unloading. In fig. 3 a), flow q_{in} is different from zero in several situations: for example, when a loading order arrives at time t_{load1}, and for cooking stage. T_{load1} and T_{cook1} are the duration of the loading stage and cooking stage respectively. The other signal in fig. 3 b) corresponds to the outflow q_{out} which is zero except for the unloading period $T_{unload1}$. The logic of operation implies that the unloading time $t_{unload1}$ must be placed after the operation has finished, which can be translated into a constraint such as $t_{unload1} > t_{load1} + T_{load1} + T_{operation1}$, the latest being the intermediate operation period for the current feeding conditions and it is formed by the sum of the duration of several stages, included T_{cook1}. These periods can be computed as before from interpolation in a table $T_i(P, B)$ (i=every stage) that has also been obtained off-line. In order to complete the vacuum pan model, other constraints must be added reflecting its logic of operation, such as $t_{load2} > t_{load1} + T_{load1} + T_{operation1} + t_{unload1} + T_{unload1}$ that indicates that the next batch 2 must start after the previous batch 1 has been unloaded. These two constraints are necessary for each batch predicted and for each tacha.

In relation with the subjacent time model and the scheduling policy, the classical approach considers the time axis divided in sampling periods, where each sampling time j has an associated integer variable indicating if unit i starts or not its operation in period j. The scheduler solves a MIP problem to determine the optimal start and ending times of the batch units. In this paper we have applied an alternative approach that is coherent with the use of the temporal patterns shown in fig. 3 a) and b). It assumes as unknowns the time of occurrence of the events, t_{load1} and $t_{unload1}$, which are real variables, instead of using integer variables in every sampling period [4]. In this way, all the decision variables of the internal model are continuous. Notice that this approach means that the scheduling problem is not computed separately but it is integrated into the overall predictive control and the need for solving mix integer optimization problem is avoided, being substituted by an NLP one.

3.1 NMPC Controller

Before the non-linear model predictive control problem can be solved, it is necessary to adapt some concepts used in standard continuous MPC to the context of mix continuous-batch processes. The first one is the prediction horizon ($N2$) that will be translated into Np minimum number of full bathes performed for all batch unit. The concept of control horizon (Nu) is split into batch control horizon (Nb_i) and continuous control horizon (Nc). The first refers to the number of batches performed of each batch unit i ($i = A1, A2, A3, B1$) in which the decision variables t_{load} and t_{unload} will be computed. From Nb_i until the end of the prediction horizon (Np), these values will be equal to the ones of the last batch. Notice that this implies the assumption that a stable cyclic pattern will be reached at the end of the prediction horizon, in a similar way to how the future control signal is treated in continuous MPC. Each Nb_i will fix the number of unknown time instants t_{load} and t_{unload}, two per batch performed and per unit. Finally the Nc horizon has the classical meaning for the classical continuous manipulated variables. The control decisions are computed solving an NLP optimization problem where the aim is to minimize a quadratic cost function J, subject to the decision variables u_j:

$$J = \int_{0}^{Tstop} \sum_{i} \alpha_i \left(y_i(t) - y_i^{ref} \right)^2 dt \tag{1}$$

with the usual constraints $y_i^{min} \leqslant y_i(t) \leqslant y_i^{max}$ and $u_j^{min} \leqslant u_j(t) \leqslant u_j^{max}$, where the y_i's extend to purities and brixes in the feeding tanks (P_A, B_A, P_B, B_B) and the levels in these tanks (L_A, L_B) and in the two malaxadors (L_{MA}, L_{MB}). $Tstop$ is the total time of prediction fixed by Np, prediction ends when at least Np full cycles are performed for all tachas. Respect to the future manipulated variables, u_j are times of load and unload every vacuum pan plus total flow and proportion of higher and lower purity syrup in the centrifugal separators of section A and B. In total the decision variables are $2 \times Nb_{A1} + 2 \times Nb_{A2} + 2 \times Nb_{A3} + 2 \times Nb_{A3} + 4 \times Nc$. α_i and β_j are given values of weights. The optimization is subjected to the internal model of the process and additional constraints imposed by the range and operation of the vacuum pans and other units.

4 Simulation Results and Conclusions

The control strategy described in the previous sections was tested in simulation using the state-of-the-art EcosimPro environment. The process was represented

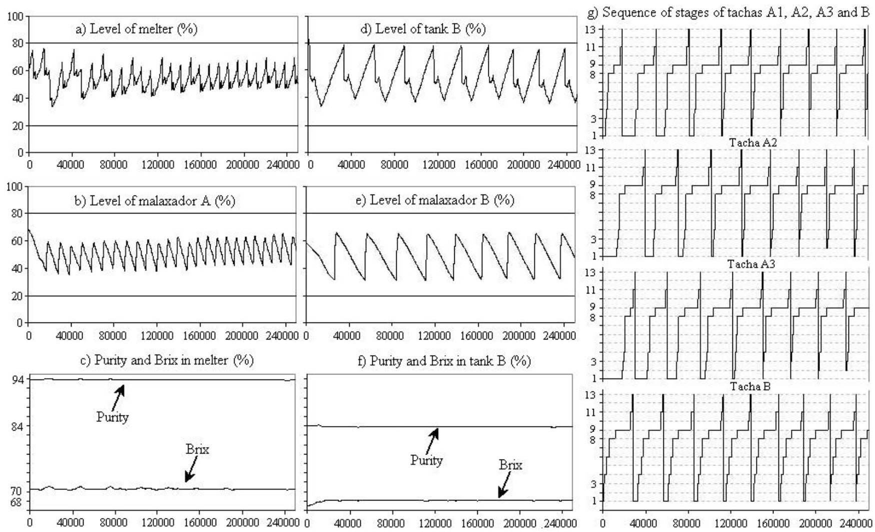


Fig. 4. a), b) and c) Controlled variables for section A. d), e) and f) Same variables but for section B. g) Sequencing of each tacha.

by a detailed simulation built using validated models of the Sugar Processes Library [5] including sequential and local controls of all units. This model involves 14719 variables and 5652 differential-algebraical equations (DAES). The controller was programmed in C++ and contains the SQP algorithm which is able to call another EcosimPro simulation with the MPC internal model (only 1823 variables and 130 DAES) for computing the cost function J each time it was needed. The sample time was chosen as 15 min. We present an experiment of 69.4 hours (250000 sec.), with an inflow of feed syrup of 6 kg/sec. with 94.4 of purity and 72 of brix. All batch control horizons ($Nb_i, i = A1, A2, A3, B1$) were fixed in 2, and continuous control horizon (Nc) was fixed in 4, so, the number of decision variables is 32. On the other hand prediction horizon (Np) was fixed in 3, that is to say, 25 hours of predictions. Control objectives (references and maximum/minimum values permitted for controlled variables) and weights in cost function (1) are:

	$L_A(\%)$	$P_A(\%)$	$B_A(\%)$	$L_{MA}(\%)$	$L_B(\%)$	$P_B(\%)$	$B_B(\%)$	$L_{MB}(\%)$
maximum	80	-	-	80	80	-	-	80
minimum	20	-	-	20	20	-	-	20
reference	50	94	70.5	50	50	84	68	50
weight	0.1	1	1	0.1	0.1	1	1	1

Fig. 4 a) and b) shows the levels of melter and malaxador A, and its minimum and maximum values allowed, fig. 4 c) shows purity and brix in the melter. Fig. 4 d) e) and f) shows the same variables but for section B. The sequence of stages of vacuum pans A1, A2, A3 and B can be seen in fig. 4 g). Time of stage 1 is the manipulated variable to load syrup and time of stage 9 is the manipulated variable to unload cooked mass. Cooking, load and unload stages correspond with numbers 8, 3 and 11. The hybrid controller is able to operate well the process: performing an adequate scheduling of tachas and maintaining purities and brixes close of its set points and levels within permitted range.

In this paper a plant-wide control strategy for the crystallization section of a beet sugar factory has been presented. It is based in a hierarchical view of the problem and, in the use of MPC with a simplified model that combines material balances of the continuous units and an abstract model of the batch ones. This is described in terms of tables computed off-line and prescribed patterns of the batch units variables and time of occurrence of the events, instead of using integer variables, allowing to use NLP algorithms instead of MIP ones. The strategy has proved to perform well in a simulated environment and opens the door to practical implementations at industrial scale.

Acknowledgements

The authors wish to express their gratitude to the Spanish Ministry of Education and Science (former MCYT) for its support through project DPI2003-0013, as well as the European Commission through VI FP NoE HYCON.

References

[1] Erickson K T, Hedrick J L (1999) Plant-Wide Process Control. Wiley Publishers

[2] Floudas C A (1995) Non-linear and Mix-Integer Optimization. Oxford Univ. Press

[3] Bemporad A, Morari M (1999) Control of systems integrating logic, dynamics, and constraints. Automatica 35: 407-427

[4] Prada C de, Cristea S, Sarabia D, Colmenares W (2004) Hybrid control of a mixed continuous-batch process. ESCAPE14 739-744 ISBN: 0-444-51694-8

[5] Prada C de Merino A, Pelayo S, Acebes F, Alves R (2003) A simulator of sugar factories for operators training. AFoT 2003 97-100 ISBN: 84-95999-46-3

Application of the NEPSAC Nonlinear Predictive Control Strategy to a Semiconductor Reactor

Robin De Keyser[1] and James Donald III[2]

[1] Ghent University, EeSA-department of Electrical energy, Systems and Automation,
Technologiepark 913, 9052 GENT, Belgium
rdk@autoctrl.UGent.be

[2] ASMA, Phoenix, Arizona, USA

1 Introduction

Increased requirements of flexible production have led to the development of *single-wafer* processing equipment for integrated circuit fabrication. For commercially feasible throughput, it is substantial to minimize the process cycle time by heating only the wafer surface, in an extremely short time period. This is only possible using radiation heating, leading to RTP systems - Rapid Thermal Processing. Under such circumstances the system is no longer isothermal and *temperature uniformity* control becomes an issue of considerable concern and technical difficulty. Commercial RTCVD reactors (Rapid Thermal Chemical Vapor Deposition) have been in use for more than a decade, but the technology still suffers from some limitations [6]. One of these is the inability to achieve with commercial control equipment an adequate temperature uniformity across the wafer surface during the rapid heating phases (e.g. from room temperature up to 1100^oC in the order of 1 minute). Deposition of silicon should be performed in a manner which minimizes crystalline growth defects, such as lattice slip. Such defects are induced by thermal gradients in the wafer during high temperature processing. For example, while gradients of about 100^oC across a wafer may be tolerable at a process temperature of 800^oC, respective gradients of only $2 - 3^o$C are allowable at process temperatures of 1100^oC. Due to the radiant type of heating, these semiconductor reactors represent a highly nonlinear interactive multi-input multi-output system.

The problem of RTP-control has been extensively dealt with in many research projects during the 1990's [6]. The current paper presents (partial) results of an extensive research project. This project ran during the 2^{nd} half of the 90's between ASM America Inc. (manufacturer of RTP equipment) and Ghent University (developer of EPSAC Model based Predictive Control technology). The EPSAC (Extended Prediction Self-Adaptive Control) strategy is further referred in [2, 3, 4, 5]. The paper is organized as follows: section 2 introduces the underlying control problem; section 3 reviews briefly the EPSAC/NEPSAC strategy;

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 503–512, 2007.
springerlink.com

section 4 is the main outcome of this paper as it presents some of the interesting experimental results on a real-life semiconductor reactor.

2 The RTCVD Reactor

A schematic representation of the RTCVD reactor is given in Fig. 1. The low profile, horizontal quartz chamber (3) provides the process environment. At the beginning of a process cycle, a manipulator (2) places the substrate (wafer) into the reaction chamber onto a susceptor. A reactant gas flows through the reaction chamber to deposit materials on the substrate. This is a temperature-controlled process: during a process cycle, a specified sequence of thermal process steps proceeds in concert with the reactive gas processing (see further Sect. 4). The system operates at temperatures ranging up to 1200^{o}C. Uniform heating of the wafer is of paramount importance. The radiant heating system used for rapid wafer heating consists of halogen lamps, which are grouped into 4 independently-controllable heating zones: **C**(enter), **F**(ront), **S**(ide), **R**(ear). The configuration, shown in Fig. 1, consists of 17 high-power (6 KW/lamp) halogen lamps located above and below the quartz-glass reaction chamber. The temperature measurement is done at 4 positions, indicated as C, F, S, R in Fig. 1. The considerations given above render temperature control essentially a multi-input multi-output (MIMO) problem with strong interaction between zones C-F-S-R. Moreover, the dynamic relationship between process inputs (manipulated variables = lamp powers $\mathbf{u}$) and process outputs (controlled variables = wafer surface temperatures $\mathbf{y}$) is *nonlinear*. This is clear from a well-known law of physics, stating that the radiant heat exchange depends on the 4th power of the involved temperatures T. During a typical recipe the reactor response can thus be quite different during a transition from 800^{o}C to 1200^{o}C.

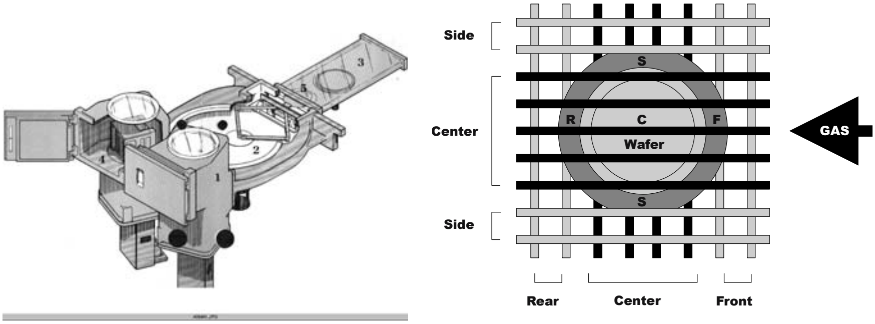


Fig. 1. Left: schematic overview of a single-wafer polysilicon deposition reactor (3=reaction chamber with wafer). Right: wafer and lamp-bank configuration (top view of reactor chamber); notice the symmetry of the system taking into account the direction of the gas flow.

3 EPSAC Model Based Predictive Control Strategy

Among the diversity of control engineering principles available today, MPC has clearly some useful characteristics to tackle above mentioned challenges: the latest MPC-versions can deal with nonlinear models, it is a multivariable control strategy and it takes into account the system constraints *a priori* by using constrained optimization methods. In this application we used our in-house EPSAC predictive control method, which has been originally described in [2, 3] and has been continuously improved over time [5]. The latest version, NEPSAC, is a nonlinear predictive controller which is essentially characterized by its simplicity since it consists of repetitive application of the basic linear EPSAC algorithm during the controller sampling interval. It leads in an iterative way, after convergence, to the optimal solution for the underlying nonlinear problem.

Many powerful NMPC (Nonlinear Model based Predictive Control) strategies exists today; they have been widely published in the control literature, e.g. [1]. The advantages of NEPSAC compared to other NMPC methods are mainly from a practical point-of-view: the approach provides a NMPC algorithm which is quite suitable for real-life applications as it does not require significant modification of the basic EPSAC software and as it is computationally simple and fast compared to other NMPC strategies [7]. The shortcomings are mainly from a theoretical point-of-view: convergence of the iterative strategy and closed-loop stability could not (yet) be proven in a formal theoretical way, although numerous simulation studies and several real-life applications have resulted in very satisfying performance.

3.1 Process Model

The basic control structure is illustrated in Fig. 2. For use in the MPC strategy, the process is modelled as

$$\boxed{y(t) = x(t) + n(t)} \tag{1}$$

with *y(t)* = process output (TC temperature measurement); *u(t)* = process input (voltage to SCR power pack); *x(t)* = model output; *n(t)* = process/model disturbance. The model (1) is presented for a SISO-process (Single Input Single Output), i.e. only 1 TC sensor and 1 SCR control input. This is done for clarity only. The (straightforward) extension to MIMO-systems, in this case a 4x4 system, is presented in detail in [5].

The process disturbance *n(t)* includes all effects in the measured output *y(t)* which do not come from the model output *x(t)*. This is a *fictitious* (non-measurable) signal. It includes effects of deposition, gas flow, measurement noise, model errors, ... These disturbances have a stochastic nature with non-zero average value. They can be modelled by a colored noise process:

$$\boxed{n(t) = C(q^{-1})/D(q^{-1}) \;\; e(t)} \tag{2}$$

$$\begin{cases} e(t) = \text{white noise (uncorrelated noise with zero mean value)} \\ C(q^{-1}) = 1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c} \\ D(q^{-1}) = 1 + d_1 q^{-1} + \ldots + d_{n_d} q^{-n_d} \end{cases}$$

The filter $C(q^{-1})/D(q^{-1})$ is the *disturbance model.* It is specified as a *design* filter, mainly affecting the robustness of the control loop against non-measurable disturbances and against modelling errors [4].

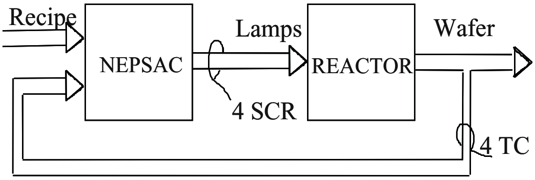


Fig. 2. Basic control structure

The model output $x(t)$ represents the effect (as described by a process model) of the process input $u(t)$ on the process output $y(t)$ and is also a non-measurable signal. The relationship between $u(t)$ and $x(t)$ is given by the generic dynamic model:

$$x(t) = f[x(t-1), x(t-2), ...; u(t-1), u(t-2)...] \tag{3}$$

where $f[.]$ represents a *known* function (process model), which can be a linear or a nonlinear function of $x(.), u(.)$.

The nonlinear multivariable model used in the RTCVD predictive controller consisted of 4 sub-models (one for each output $i = 1..4$):

$$\begin{aligned} x_i(t) = c^i x_i(t-1)^4 + a_1^i x_i(t-1) \ldots + a_3^i x_i(t-3) + b_1^{i_1} u_1(t-1) + \ldots \\ + b_3^{i_1} u_1(t-3) + \ldots + b_1^{i_4} u_4(t-1) + \ldots + b_3^{i_4} u_4(t-3) + d^i \end{aligned} \tag{4}$$

From a theoretical standpoint, the model (4) should include other nonlinear terms; indeed, the radiative heat flux is based on the temperature difference between heat source and wafer surface raised to the 4^{th} power. However, from an engineering standpoint, a simplified yet accurate model was obtained during the identification phase. This simplified model (4) still contains 68 parameters. The parameters have been identified successfully from a multiple of real-life experiments on a pilot-plant RTCVD-reactor. Some model responses can be observed in Fig. 3, clearly illustrating the interactive multivariable and nonlinear character of the process.

3.2 EPSAC

The fundamental step in MPC methodology consists in prediction of the process output $y(t+k)$ at time instant t, indicated by $y(t+k|t), k = 1..N_2$ over the *prediction horizon* N_2 and based on:

⇒ all measurements at time t: $\{y(t), y(t-1), ..., u(t-1), u(t-2), ...\}$
⇒ future values of the input: $\{u(t|t), u(t+1|t), ...\}$ (*postulated* at time t)
Using the generic model (1), the predicted values of the output are:

$$y(t+k|t) = x(t+k|t) + n(t+k|t) \tag{5}$$

Prediction of $x(t+k|t)$ and of $n(t+k|t)$ can be done respectively by *a*) recursion of the process model (3) and by *b*) using filtering techniques on the noise model (2). A detailed description is available in [5].

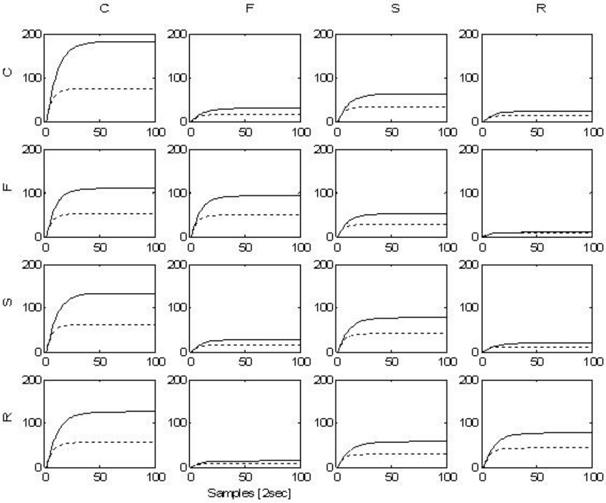


Fig. 3. Matrix of unit step responses at 860^{o}C (solid) and at 1080^{o}C (dotted) (columns correspond to SCR-inputs, rows correspond to TC-outputs)

In EPSAC, the future response is then considered as being the cumulative result of two effects:

$$y(t+k|t) = y_{\text{base}}(t+k|t) + y_{\text{optimize}}(t+k|t) \tag{6}$$

The two contributions have the following origins:

$\boxed{y_{\text{base}}(t+k|t):}$

- effect of past control $u(t-1)$, $u(t-2)$, ... (initial conditions at time t);
- effect of a base future control scenario, called $u_{\text{base}}(t+k|t),\ k \geq 0$, which is defined a priori; ref. Sect. 3.3 for some ideas on how to choose it;
- effect of future (predicted) disturbances $n(t+k|t)$.

The component $y_{\text{base}}(t+k|t)$ can be easily and correctly obtained using (2)(3)(5) – for linear as well as for nonlinear process models $f[.]$ – by taking $u_{\text{base}}(t+k|t)$ as the model input in (3): $u(t+k|t) = u_{base}(t+k|t)$.

$\boxed{y_{\text{optimize}}(t+k|t):}$

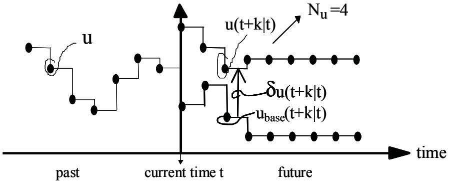


Fig. 4. The EPSAC concept of *base/optimizing* controls

- effect of the optimizing future control actions $\{\delta u(t|t),\ \delta u(t+1|t),\ \ldots \delta u(t+N_u-1|t)\}$ with $\delta u(t+k|t) = u(t+k|t) - u_{\text{base}}(t+k|t)$.

Figure 4 illustrates the concept. The *design* parameter N_u is called the *control horizon*, a standard concept from the MPC literature. From Fig. 4 it is obvious that the component $y_{\text{optimize}}(t+k|t)$ is the cumulative effect of a series of *impulse* inputs and a *step* input:

$$\begin{aligned} y_{\text{optimize}}(t+k|t) &= h_k \delta u(t|t) + h_{k-1}\delta u(t+1|t) + ... \\ &\quad + g_{k-N_u+1}\delta u(t+N_u-1|t) \end{aligned} \tag{7}$$

The parameters $g_1, g_2, ...g_k, ...$ are the coefficients of the *unit step response* of the system, i.e. the response of the system output for a stepwise change of the system input (scaled to amplitude 1). The parameters $h_1, h_2, ...h_k, ...$ are the coefficients of the *unit impulse response* of the system ($h_k = g_k - g_{k-1}$). For a linear system, the unit step response does not depend on the operating point, and its coefficients can be calculated once off-line, using the process model. Notice that this will not be the case in the NEPSAC strategy (ref. Sect. 3.3). Indeed, in the case of a nonlinear system, the step response is different for each operating point. The coefficients have to be obtained at *each* sampling instant by *explicitly* entering a step of suitable size in the process model (4), which is initialized at the current process state. The value of the step-size should have the order of magnitude of the normal process input variations. However, its exact value is not critical, since the effect of the parameters g_k, h_k will gradually disappear in NEPSAC (ref. Sect. 3.3).

Using (6) and (7), the **key EPSAC-MPC equation:**

$$\boxed{\mathbf{Y} = \bar{\mathbf{Y}} + \mathbf{G}\,\mathbf{U}} \tag{8}$$

is obtained, after introducing the following matrix notation:

$$\left.\begin{aligned} \mathbf{Y} &= [y(t+N_1|t)\ \cdots y(t+N_2|t)]^T \\ \bar{\mathbf{Y}} &= [y_{\text{base}}(t+N_1|t)\ \cdots y_{\text{base}}(t+N_2|t)]^T \\ \mathbf{U} &= [\delta u(t|t)\ \cdots \delta u(t+N_u-1|t)]^T \end{aligned}\right| \mathbf{G} = \begin{bmatrix} h_{N_1} & \cdots & g_{N_1-N_u+1} \\ h_{N_1+1} & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ h_{N_2} & \cdots & g_{N_2-N_u+1} \end{bmatrix} \tag{9}$$

The controller output is then the result of minimizing the cost function:

$$V(\mathbf{U}) = \sum_{k=N_1}^{N_2} [r(t+k|t) - y(t+k|t)]^2 \tag{10}$$

with $r(t+k|t)$ the desired reference trajectory (called *recipe* in semiconductor terminology) and the *horizons* N_1, N_2 being design parameters. It is now straightforward to derive the (unconstrained) EPSAC solution:

$$\mathbf{U}^* = [\mathbf{G^T G}]^{\mathbf{-1}}[\mathbf{G^T}(\mathbf{R} - \overline{\mathbf{Y}})] \tag{11}$$

Only the first element $\delta u^*(t|t)$ in $\mathbf{U}^*$ is required in order to compute the actual control action applied to the process. At the next sampling instant $t+1$, the whole procedure is repeated, taking into account the new measurement information $y(t+1)$; this is called the principle of *receding horizon control*. As well-known in current MPC-practice, the cost index (10) can be extended with constraints, leading to a *quadratic programming* problem. This has been the approach to tackle input saturation constraints in the RTCVD application.

3.3 NEPSAC

The calculation of the predicted output with (6) involves the superposition principle. When a nonlinear system model $f[.]$ is used in (3), above strategy is only valid - from a practical point of view - if the term $y_{optimize}(t+k|t)$ in (6) is small enough compared to the term $y_{base}(t+k|t)$. When this term would be zero, the superposition principle would no longer be involved. The term $y_{optimize}(t+k|t)$ will be *small* if $\delta u(t+k|t)$ is small, see (7). Referring to Fig. 4, $\delta u(t+k|t)$ will be small if $u_{base}(t+k|t)$ is *close* to the optimal $u^*(t+k|t)$.

This can be realized iteratively, by executing the following steps at each controller sampling instant:

1. Initialize $u_{base}(t+k|t)$ as: $u^1_{base}(t+k|t) = u^*(t+k|t-1)$, i.e. the optimal control sequence as computed during the previous sampling instant; in other words: $u^*(t+k|t-1)$ is used as a 1^{st} *estimate* for $u^*(t+k|t)$
2. Calculate $\delta u^1(t+k|t)$ using the linear EPSAC algorithm
3. Calculate the corresponding $y^1_{optimize}(t+k|t)$ with (7) and compare it to $y^1_{base}(t+k|t)$, which is the result of $u^1_{base}(t+k|t)$
4. - In case $y^1_{optimize}(t+k|t)$ is NOT small enough compared to $y^1_{base}(t+k|t)$: re-define $u_{base}(t+k|t)$ as $u^2_{base}(t+k|t) = u^1_{base}(t+k|t) + \delta u^1(t+k|t)$ and go to 2. The underlying idea is that $u^1_{base}(t+k|t) + \delta u^1(t+k|t)$ - which is the optimal $u^*(t+k|t)$ for a linear system - can act as a 2^{nd} *estimate* for the optimal $u^*(t+k|t)$ in case of a nonlinear system
   - In case $y^i_{optimize}(t+k|t)$ is small enough compared to $y^i_{base}(t+k|t)$: use $u(t) = u^i_{base}(t|t) + \delta u^i(t|t)$ as the resulting control action of the current sampling instant (notice that $i = 1, 2, \ldots$, according to the number of iterations).

This algorithm results after convergence to the optimal solution for the underlying nonlinear predictive control problem. A convergence proof is not available; however, simulation results and practical experience both look very promising. The number of required iterations depends on how far the optimal $u^*(t+k|t)$ is away from the optimal $u^*(t+k|t-1)$. In quasi-steady-state situations, the number of iterations is low $(1 \ldots 2)$. On the other hand, during transients the number of iterations might raise to 10. As the NEPSAC algorithm consists of a repetitive use of the basic linear EPSAC algorithm and as EPSAC requires a low computational effort, this is acceptable in practice. In the RTCVD application, 10 iterations require about 100 ms, which is a small fraction of the controller sampling period (2 s).

4 Experimental Results

Hundreds of test runs on different types of real-life RTCVD-reactors have shown the excellent performance of the MPC strategy compared to the traditional (commercial) PID approach. During the comparison, the PID controllers were configured and tuned by experienced staff of the company, according to their expert skill of many years. Figure 5 presents typical PID results during a recipe with ramp rates of about $6^o\mathbf{C/s}$. This is the maximum ramp rate that is feasible on this kind of equipment under PID control (note that there is a trend towards higher ramp rates in order to reduce the process cycle time; however the higher the ramp rate, the more difficult it is to keep control of the temperature-uniformity over the wafer surface). Figure 5 presents also typical NEPSAC-MPC results during a recipe with ramp rates of about $15^o\mathbf{C/s}$. Although the ramp rate is much higher, it is clear that tight control of the temperature uniformity is still possible.

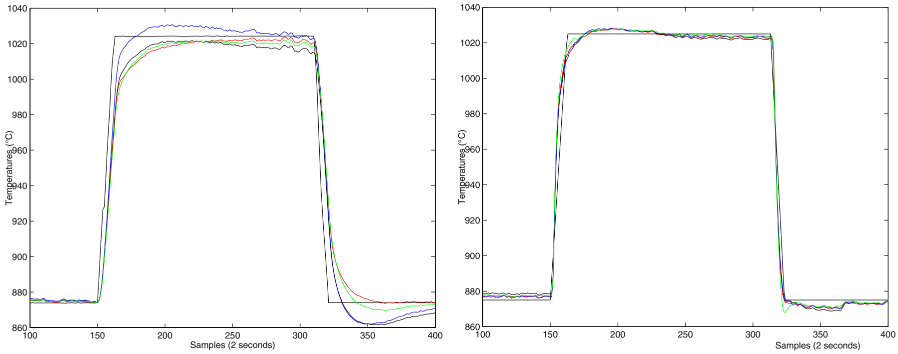


Fig. 5. PID with ramp rate $6^o\mathbf{C/s}$ (center/front/side/rear wafer surface temperatures) (left) and NEPSAC with ramp rate $15^o\mathbf{C/s}$ (center/front/side/rear wafer surface temperatures) (right)

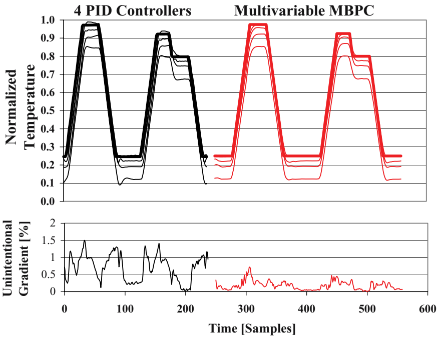


Fig. 6. Controller performance

To indicate the *practically* useful improvement that can be obtained with the multivariable NEPSAC controller, its performance versus 4 independent commercial PID controllers (=current practice) was compared for a blanket epitaxial deposition process. Figure 6 indicates the desired temperature profile during this process. The upper part represents a typical recipe; temperature offsets are intentional and required due to the position of the thermocouples. The lower part represents the un-intentional gradient in the wafer. In Table 1 are presented the key controller performance parameters during and after ramping-up. The superior performance of the MPC directly translates in a process time reduction. The achievable ramp rate with the 4 PID controllers is limited due to the thermal gradients, which introduce crystallographic dislocations into the wafer. With a twofold increased ramp rate and a reduced settling time, the nonlinear multivariable MPC results in a robust process without introducing dislocations into the wafer.

Table 1. Controller Performance

	4 PID	NEPSAC
Overshoot (^{o}C)	7	0
Stabilization time (seconds)	25	7
Average Unintentional Gradient (^{o}C)	4.1	1.3

5 Conclusions

The objective of this paper was to describe a real-life application of NMPC in the field of semiconductor processing. The process configuration and operation of a RTCVD-reactor (Rapid Thermal Chemical Vapor Deposition) has served as an example to illustrate the possibilities of advanced control. This application is a real challenge for control engineering, in that it is a highly interactive multi-input multi-output nonlinear process with very stringent performance specifications for the control system. An extensive series of experiments has been done, comparing the performance of the commercial PID-type control system to that of the NEPSAC predictive control strategy. The results indicate that the performance of the predictive control strategy surpasses by far the performance of the controllers which have traditionally been used in this kind of application.

References

[1] Allgöwer, F., Zheng, A. (Eds.), "Nonlinear Predictive Control", Birkhauser Verlag Basel, (2000).

[2] De Keyser, R., Van Cauwenberghe, A., "A Self-Tuning Multistep Predictor Application", *Automatica*, **17**, 167-174,(1981).

[3] De Keyser, R., Van Cauwenberghe, A., "Extended Prediction Self-Adaptive Control", *IFAC Symposium on Identification, Pergamon Press Oxford*, 1255-1260, (1985).

[4] De Keyser, R., Ionescu, C., "The disturbance model in model based predictive control", *IEEE Conf. on Control Application*, paper 001472, (2003).

[5] De Keyser, R., "Model Based Predictive Control for Linear Systems", *UNESCO Encyclopaedia of Life Support Systems, Article contribution 6.43.16.1, Eolss Publishers Co Ltd, Oxford, ISBN 0 9542 989 18-26-34 (www.eolss.net)*, 30p, (2003).

[6] Kailath, T., Schaper, C., Cho, Y., Gyugyi, P. , Norman, S., Park,P., Boyd, S., Franklin, G., Saraswat, K., Moslehi, M., Davis, C., "Control for Advanced Semiconductor Device Manufacturing: A Case History", *In: William S. Levine (Ed.), "The Control Handbook"* , CRC-press, IEEE-press, (1996).

[7] Rueda, A., Cristea, S., De Prada, C., De Keyser, R., "Nonlinear Predictive Control of a Distillation Column", *IEEE CDC-ECC Joint Int. Conf.*, 5156-5161, (2005).

Integrating Fault Diagnosis with Nonlinear Model Predictive Control

Anjali Deshpande[1], Sachin C. Patwardhan[2], and Shankar Narasimhan[3]

[1] Systems and Control Engineering, Indian Institute of Technology, Bombay,
Mumbai, 400076 India
apdesh@iitb.ac.in

[2] Department of Chemical Engineering, Indian Institute of Technology, Bombay,
Mumbai, 400076 India
sachinp@iitb.ac.in

[3] Department of Chemical Engineering, Indian Institute of Technology, Madras,
Chennai, 600036 India
naras@che.iitm.ac.in

1 Introduction

The abundance of batch processes and continuous processes with wide operating ranges has motivated the development of nonlinear MPC (NMPC) techniques, which employ nonlinear models for prediction. The prediction model is typically developed once in the beginning of implementation of an NMPC scheme. However, as time progresses, slow drifts in unmeasured disturbances and changes in process parameters can lead to significant mismatch in plant and model behavior. Also, NMPC schemes are typically developed under the assumption that sensors and actuators are free from faults. However, *soft faults*, such as biases in sensors or actuators, are frequently encountered in the process industry. In addition to this, some actuator(s) may fail during operation, which results in loss of degrees of freedom for control. Occurrences of such faults and failures can lead to a significant degradation in the closed loop performance of the NMPC.

The conventional approach to deal with the plant model mismatch in the NMPC formulations is through the introduction of additional artificial states in the state observer. The main limitation of this approach is that number of extra states introduced cannot exceed the number of measurements. This implies that it is necessary to have a priori knowledge of which subset of faults are most likely to occur or which parameters are most likely to drift. In such a formulation, the state estimates can become biased when un-anticipated faults occur. Moreover, the permanent state augmentation approach cannot systematically deal with the difficulties arising out of sensor biases or actuator failures.

Attempts to develop fault-tolerant MPC schemes have mainly focused on dealing sensor or actuator failures [1]. Recently, Prakash et al. [2] have proposed an active fault tolerant linear MPC (FTMPC) scheme, which can systematically

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 513–521, 2007.
springerlink.com

deal with soft faults in a unified framework. The main limitation of this approach arises from the use of linear perturbation model for performing control and diagnosis tasks. The use of linear models not only restricts its applicability to a narrow operating range but also limits the diagnostic abilities of fault detection and identification (FDI) components to only linear additive type faults. As a consequence, many faults that nonlinearly affect the system dynamics, such as abrupt changes in model parameters or unmeasured disturbances, have to be approximated as linear additive faults. Moreover, the FTMPC scheme doesn't deal with failures of sensors or actuators.

In the present work, we propose a fault tolerant NMPC (FTNMPC) formulation with an intelligent nonlinear state estimator, Extended Kalman Filter (EKF), which can diagnose the root cause of model plant mismatch and correct itself. The whiteness of innovation sequence generated by the state estimator is taken as an indicator of good health of the model. A significant and sustained departure from this behavior is assumed to result from model plant mismatch and a nonlinear version of generalized likelihood ratio (GLR) based FDI scheme is used to analyze the root cause of model plant mismatch. The proposed FDI method also generates an estimate of the magnitude of the fault, which is used to compute an online bias correction to the model at the location isolated by the FDI scheme. The model correction strategy overcomes the limitation on the number of extra states that can be added to the state space model in NMPC for offset removal and allows bias compensation for more variables than the number of measured outputs. The proposed FTNMPC eliminates offset between the true values and set points of controlled variables in presence of variety of faults while conventional NMPC does not. Also, the true values of state variables, manipulated inputs and measured variables are maintained within their imposed bounds in FTNMPC while in conventional NMPC these may be violated when soft faults occur. When an actuator fails, the proposed FTNMPC formulation is able to make modifications in the controller objective function and constraint set to account for the loss of a degree of freedom. These advantages of the proposed scheme are demonstrated using simulation studies on a benchmark continuous stirred tank reactor (CSTR) control problem, which exhibits strongly nonlinear dynamics.

2 Fault Diagnosis Using Nonlinear GLR Method

In this section, we first describe the FDI method as applied once when a fault is detected for the first time. Consider a continuous time nonlinear stochastic system described by the following set of equations

$$\mathbf{x}(k+1) = \mathbf{x}(k) + \int_{kT}^{(k+1)T} \mathbf{F}\left[\mathbf{x}(t), \mathbf{u}(k), \mathbf{p}, \mathbf{d}(k)\right] dt \tag{1}$$

$$\mathbf{y}(k) = \mathbf{H}\left[\mathbf{x}(k)\right] + \mathbf{v}(k) \quad ; \quad \mathbf{d}(k) = \overline{\mathbf{d}} + \mathbf{w}(k) \tag{2}$$

where $\mathbf{x} \in R^n, \mathbf{y} \in R^r$ and $\mathbf{u} \in R^m$ represent the state variables, measured outputs and manipulated inputs, respectively. The variables $\mathbf{p} \in R^p$ and $\mathbf{d} \in R^d$ represent the vector of parameters and unmeasured disturbance variables, respectively, which are likely to undergo deterministic changes. The unmeasured disturbances are also assumed to undergo random fluctuations. For mathematical tractability, these are simulated as piecewise constant between each sampling period and changing randomly from their nominal value at each sampling instant. Here, $\mathbf{v}(k)$ and $\mathbf{w}(k)$ are zero mean Gaussian white noise sequences with known covariance matrices. Equations 1 and 2 represent the normal or fault free behavior of the process and are used to develop the standard linearized EKF [3]. In remainder of the text, we refer to this EKF as *normal EKF*.

In order to isolate faults and estimate their magnitudes, it is necessary to develop a model for each hypothesized fault that describes its effect on the evolution of the process variables. The models that are used to describe some of the faults are as follows:

- **Bias in j^{th} sensor :** Subsequent to occurrence of bias in the sensor at instant t, the behavior of measured outputs is modeled as

  $$\mathbf{y}_{y_j}(k) = \mathbf{H}\left[\mathbf{x}(k)\right] + b_{y_j}\mathbf{e}_{y_j}\sigma(k-t) + \mathbf{v}(k)$$

  Here, b_{y_j} represents sensor bias magnitude, $\mathbf{e}_{y_j}$ represents sensor fault vector with j^{th} element equal to unity and all other elements equal to zero and $\sigma(k-t)$ represents a unit step function defined as

  $$\sigma(k-t) = 0 \text{ if } k \prec t \quad ; \quad \sigma(k-t) = 1 \text{ if } k \succeq t$$

- **Abrupt change in j^{th} unmeasured disturbance variable:**

  $$\mathbf{d}_{d_j}(k) = \overline{\mathbf{d}} + \mathbf{w}(k) + b_{d_j}\mathbf{e}_{d_j}\sigma(k-t)$$

- **Failure of j^{th}Actuator / Sensor:**

  $$\mathbf{u}_{m_j}(k) = \mathbf{m}(k) + \left[b_{m_j} - \mathbf{e}_{m_j}^T\mathbf{m}(k)\right]\mathbf{e}_{m_j}\sigma(k-t) \tag{3}$$

  $$\mathbf{y}_{s_j}(k) = \mathbf{H}\left[\mathbf{x}(k)\right] + \left[b_{s_j} - \mathbf{e}_{s_j}^T\mathbf{H}\left[\mathbf{x}(k)\right]\right]\mathbf{e}_{s_j}\sigma(k-t) + \mathbf{v}(k) \tag{4}$$

  where b_{m_j}/b_{s_j} represents constant value at which the j^{th}actuator/sensor is stuck. Note that we differentiate the controller output $\mathbf{m}$ and manipulated input $\mathbf{u}$ entering the process. The controller output equals the manipulated input under the fault free conditions. Similar fault models can be formulated for other faults.

To detect occurrence of a fault, it is assumed that the sequence of innovations $\gamma(k)$ generated by the *normal EKF* is a zero mean Gaussian white noise process with covariance matrix $\mathbf{V}(k)$. A sustained departure from this behavior is assumed to result from a fault. Simple statistical tests, namely, fault detection test (FDT) and fault confirmation test (FCT) as given in [4] are modified based on innovations obtained from EKF and used for estimating time of occurrence of fault. Taking motivation from *nonlinear GLR* method proposed for gross error detection under steady-state conditions [5], we propose a version of *nonlinear GLR* method under dynamic operating conditions. By this approach, once the FCT confirms the occurrence of a fault at instant t, we formulate a separate EKF over a time window $[t, t+N]$ for each hypothesized fault. For example, assuming that actuator j has failed at instant t, the process behavior over window $[t, t+N]$ can be described as follows

$$\mathbf{x}_{m_j}(i+1) = \mathbf{x}_{m_j}(i) + \int_{iT}^{(i+1)T} \mathbf{F}\left[\mathbf{x}_{m_j}(t),\mathbf{u}_{m_j}(i),\overline{\mathbf{p}},\overline{\mathbf{d}}\right] dt \tag{5}$$

$$\mathbf{y}_{m_j}(i) = \mathbf{H}\left[\mathbf{x}_{m_j}(i)\right] + \mathbf{v}(k) \tag{6}$$

where $\mathbf{u}_{m_j}(i)$ is given by equation 3. The magnitude estimation problem can now be formulated as a nonlinear optimization problem as follows

$$\min_{b_{m_j}} (\varPsi_{m_j}) = \sum_{i=t}^{t+N} \gamma_{m_j}^T(i)\mathbf{V}_{m_j}(i)^{-1}\gamma_{m_j}(i) \tag{7}$$

where $\gamma_{m_j}(i)$ and $\mathbf{V}_{m_j}(i)$ are the innovations and the innovations covariance matrices, respectively, generated by the EKF constructed using equations 5 and 6 with initial state $\widehat{\mathbf{x}}(t|t)$. The estimates of fault magnitude can be generated for each hypothesized fault in this manner. The fault isolation is viewed as a problem of finding the observer that best explains the output behavior observed over the window. Thus, the fault that corresponds to minimum value of the objective function, $\varPsi_{f_j}$, with respect to f_j, where $f \in (p, d, y, u, m, s)$ represents the fault type, is taken as the fault that has occurred at instant t. Since the above method is computationally expensive, we use a simplified version of nonlinear GLR proposed by Vijaybaskar, [6] for fault isolation. This method makes use of the recurrence relationships for *signature matrices* derived under linear GLR framework [4], which capture the effect of faults on state estimation error and innovation sequence. If a fault of magnitude b_{f_j} occurs at time t, the expected values of the innovations generated by the *normal EKF* at any subsequent time are approximated as

$$E\left[\gamma(i)\right] = b_{f_j}\mathbf{G}_f(i;t)\mathbf{e}_{f_j} + \mathbf{g}_{f_i} \quad \forall i \succeq t \tag{8}$$

Here, $\mathbf{G}_f(i;t)$ and $\mathbf{g}_{f_j}(i;t)$ represent fault signature matrix and fault signature vector, respectively, which depend on type, location and time of occurrence of a fault. For example, if j^{th} actuator fails, then the corresponding signature matrices and the signature vectors can be computed using the following recurrence relations for $i \in [t, t+N]$:

$$\mathbf{G}_m(i;t) = \mathbf{C}(i)\mathbf{\Gamma}_u(i) - \mathbf{C}(i)\mathbf{\Phi}(i)\mathbf{J}_m(i-1;t) \tag{9}$$

$$\mathbf{g}_{m_j}(i;t) = \mathbf{C}(i)\mathbf{\Gamma}_u(i)\left[\mathbf{e}_{m_j}^T\mathbf{m}(i)\right]\mathbf{e}_{m_j} - \mathbf{C}(i)\mathbf{\Phi}(i)\mathbf{j}_{m_j}(i-1;t) \tag{10}$$

$$\mathbf{J}_m(i;t) = \mathbf{\Phi}(i)\mathbf{J}_m(i-1;t) + \mathbf{L}(i)\mathbf{G}_m(i-1;t) - \mathbf{\Gamma}_u(i) \tag{11}$$

$$\mathbf{j}_{m_j}(i;t) = \mathbf{\Phi}(i)\mathbf{j}_{m_j}(i-1;t) + \mathbf{L}(i)\mathbf{g}_{m_j}(i-1;t) - \mathbf{\Gamma}_u(i)\left[\mathbf{e}_{m_j}^T\mathbf{m}(i)\right]\mathbf{e}_{m_j} \tag{12}$$

Here,

$$\mathbf{\Gamma}_u(i) = \int_0^T \exp\left(\mathbf{A}(i)q\right)\mathbf{B}_u(i)dq \quad ; \quad \mathbf{B}_u(i) = \left[\frac{\partial \mathbf{F}(\mathbf{x},\mathbf{m},\mathbf{p},\mathbf{d})}{\partial \mathbf{m}}\right]_{(\widehat{\mathbf{x}}(i|i),\mathbf{m}(i),\overline{\mathbf{p}},\overline{\mathbf{d}})}$$

$$\mathbf{\Phi}(i) = \exp\left[\mathbf{A}(i)T\right] \;\; ; \;\; \mathbf{A}(i) = \left[\frac{\partial \mathbf{F}}{\partial \mathbf{x}}\right]_{(\widehat{\mathbf{x}}(i|i),\mathbf{m}(i),\overline{p},\overline{\mathbf{d}})} \quad ; \quad \mathbf{C}(i) = \left[\frac{\partial \mathbf{H}(\mathbf{x})}{\partial \mathbf{x}}\right]_{(\widehat{\mathbf{x}}(i|i)}$$

are the linearized discrete time varying system matrices and $\mathbf{L}(i)$ is the Kalman gain computed using the *normal EKF*. Similar recurrence relations can be constructed for other types of faults. For each hypothesized fault, the log likelihood ratio, T_{f_j}, is computed as follows

$$T_{f_j} = \left[d_{f_j}^2/c_{f_j}\right] + \sum_{i=t}^{t+N} \mathbf{g}_{f_j}^T(i;t)\mathbf{V}(i)^{-1}\left[2\gamma(i) - \mathbf{g}_{f_j}(i;t)\right] \tag{13}$$

$$d_{f_j} = \mathbf{e}_{f_j}\sum_{i=t}^{t+N} \mathbf{G}_f^T(i;t)\mathbf{V}(i)^{-1}\left[\gamma(i) - \mathbf{g}_{f_j}(i;t)\right] \tag{14}$$

$$c_{f_j} = \mathbf{e}_{f_j}^T\sum_{i=t}^{t+L} \mathbf{G}_f^T(i;t)\mathbf{V}(i)^{-1}\mathbf{G}_f(i;t)\mathbf{e}_{f_j} \tag{15}$$

where $\gamma(i)$ and $\mathbf{V}(i)$ are obtained using *normal EKF*. The fault location can be obtained from the maximum value of the test statistic T_{f_j}. An estimate of the bias magnitude is generated as $b_{f_j}^{(0)} = d_{f_j}/c_{f_j}$. Once a fault f_j is isolated, a refined estimate of the fault magnitude is generated by formulating a nonlinear optimization problem as described above, starting from the initial guess of $b_{f_j}^{(0)}$.

3 Fault Tolerant NMPC (FTNMPC) Formulation

To begin with, let us consider conventional NMPC formulation. Let us assume that at any instant k, we are given p future manipulated input moves

$$\{\mathbf{m}(k|k), \mathbf{m}(k+1|k)........\mathbf{m}(k+p-1|k)\}$$

The future (predicted) estimates of the state variables and outputs, which have been compensated for plant model mismatch, are given as follows

$$\widetilde{\mathbf{x}}(k+j+1|k) = \widehat{\mathbf{x}}(k+j|k) + \int_{(k+1)T}^{(k+l+1)T} \mathbf{F}\left[\widehat{\mathbf{x}}(\tau), \mathbf{m}(k+j|k), \overline{\mathbf{p}}, \overline{\mathbf{d}}\right] d\tau \tag{16}$$

$$\widehat{\mathbf{x}}(k+j+1|k) = \widetilde{\mathbf{x}}(k+j+1|k) + \mathbf{L}(k)\gamma(k); \;\; \varepsilon(k) = \mathbf{y}(k) - \widehat{\mathbf{y}}(k|k) \tag{17}$$

$$\widehat{\mathbf{y}}(k+j|k) = \mathbf{G}\left[\widehat{\mathbf{x}}(k+j|k)\right] + \varepsilon(k) \;;\; j \in [0,\; p] \tag{18}$$

At any sampling instant k, the nonlinear model predictive control problem is defined as a constrained optimization problem whereby the future manipulated input moves are determined by minimizing an objective function

$$\underset{\mathbf{m}(k|k),\mathbf{m}(k+1|k)........\mathbf{m}(k+q-1|k)}{\min} \begin{Bmatrix} \sum_{j=1}^{p} \mathbf{e}_f(k+j|k)^T W_E \mathbf{e}_f(k+j|k)+ \\ \sum_{j=0}^{q-1} \Delta\mathbf{m}(k+j|k)^T W_u \Delta\mathbf{m}(k+j|k) \end{Bmatrix}$$

subject to following constraints

$$\begin{aligned} \mathbf{m}(k+q|k) &= \mathbf{m}(k+q+1|k) = ...\mathbf{m}(k+p-1|k) = \mathbf{m}(k+q-1|k) \\ \mathbf{m}^L &\leq \mathbf{m}(k+j|k) \leq \mathbf{m}^U \text{ (for } j = 0..q-1) \\ \Delta\mathbf{m}^L &\leq \Delta\mathbf{m}(k+j|k) \leq \Delta\mathbf{m}^U \text{ (for } j = 0..q-1) \\ \mathbf{e}_f(k+j|k) &= \mathbf{y}_r(k+j|k) - \widehat{\mathbf{y}}(k+j|k) \\ \Delta\mathbf{m}(k+j|k) &= \mathbf{m}(k+j|k) - \mathbf{m}(k+j-1|k) \end{aligned}$$

Here, $\mathbf{y}_r(k+j|k)$ represents the future setpoint trajectory.

We now present the modifications necessary in the NMPC formulation when a fault is detected for the first time by FDI component. Consider a situation where FDT has been rejected at time instant t and subsequently FCT has been rejected at time $t+N$ for the first time. Further assume that at instant $t+N$ we have isolated a fault f using modified GLR method and estimated the fault magnitude using data collected in the interval $[t, t+N]$. During the interval $[t, t+N]$, the NMPC formulation is based on the prediction model given by equations 16 to 18. However after the identification of the fault at instant $t+N$, we modify the model for $k \geq t+N$ as follows:

- **Sensor faults:** If sensor bias is isolated, the measured output is compensated as $\mathbf{y}_c(k) = \mathbf{y}(k) - \widehat{b}_{y_j}\mathbf{e}_{y_j}$ and used in FDI as well as MPC formulation for computing innovation sequence. If a sensor failure is diagnosed, the measurements coming from a failed sensor are replaced by corresponding estimates in the FTNMPC formulation.
- **Step jump in unmeasured disturbance:** The prediction equation in the state estimator and future predictions in NMPC are modified as follows

$$\widehat{\mathbf{x}}(k+1|k) = \widehat{\mathbf{x}}(k|k) + \int_{kT}^{(k+1)T} \mathbf{F}\left[\widehat{\mathbf{x}}(t), \mathbf{m}(k), \overline{\mathbf{p}}, \overline{\mathbf{d}} + \widehat{b}_{d_j}\mathbf{e}_{d_j}\right] dt$$

$$\begin{aligned} \widehat{\mathbf{x}}(k+l+1|k) &= \widehat{\mathbf{x}}(k+l|k) \\ &+ \int_{(k+l)T}^{(k+l+1)T} \mathbf{F}\left[\widehat{\mathbf{x}}(\tau), \mathbf{m}(k+l|k), \overline{\mathbf{p}}, \overline{\mathbf{d}} + \widehat{b}_{d_j}\mathbf{e}_{d_j}\right] d\tau + \mathbf{L}(k)\gamma(k) \end{aligned}$$

- **Failed actuator:** In state estimation the failed actuator is treated as constant $\mathbf{m}_j(k) = \widehat{b}_{m_j}$, where $\widehat{b}_{m_j}$ is the estimate of stuck actuator signal for j^{th} actuator. Also, in the NMPC formulation, we introduce additional constraints as $\mathbf{m}_j(k+l|k) = \widehat{b}_{m_j}$ for $l = 0....q-1$. If number of setpoints specified in the NMPC formulation equals the number of manipulated inputs, then we modify the NMPC objective function by relaxing setpoint on one of the controlled outputs.

The main concern with the above approach is that the magnitude and the position of the fault may not be accurately estimated. Thus, there is a need to introduce integral action in such a way that the errors in estimation of fault magnitude or position can be corrected in the course of time. Furthermore, other faults may occur at subsequent time instants. Thus, in the on-line implementation of FTNMPC, we resume application of FDI method starting at $t + N + 1$. The FDI method may identify a fault in the previously identified location or a new fault may be identified. In either case, we modify the above equations with cumulative estimate of the bias as described in Prakash et al. [2]. These are computed as $\widetilde{b}_{f_j} = \sum_{l=1}^{n_{f_j}} \widehat{b}_{f_j}(l)$ with initial value $\widehat{b}_{f_j}(0) = 0$, where n_{f_j} represents the number of times a fault of type f was isolated in the j^{th} position. The use of cumulative bias estimates can be looked upon as a method of introducing integral action to account for plant model mismatch, in which some of the states (cumulative bias estimates) are integrated at much slower rate and at regular sampling intervals.

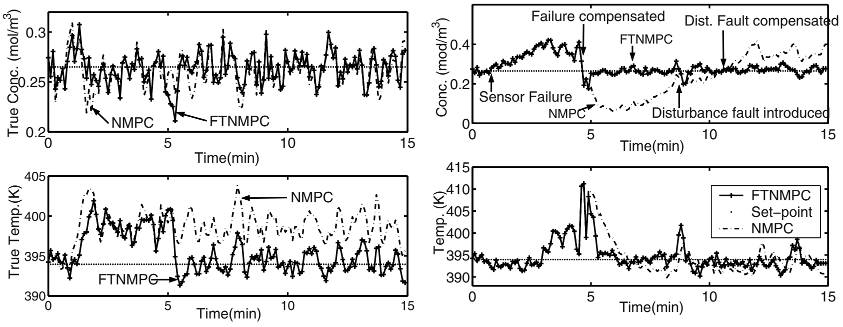
4 Simulation Case Study

Simulation studies are carried out to evaluate the proposed FTNMPC scheme on non-isothermal CSTR system. The reactor system has two state variables, the reactor concentration (C_A) and the reactor temperature(T), both of which are measured and controlled. The coolant flow rate F$_c$ and feed flow rate F are the manipulated inputs while the feed concentration(C_{AO}) is treated as a disturbance variable. Model equations are given in Marlin. ([7]) and nominal parameters, simulation conditions and tuning parameters used for controller tuning are described in Prakash et al. [4] and [2] . The bounds imposed on the inputs are as follows

$$0 \leq F_c \leq 30m^3/\min$$
$$0 \leq F \leq 2m^3/\min$$

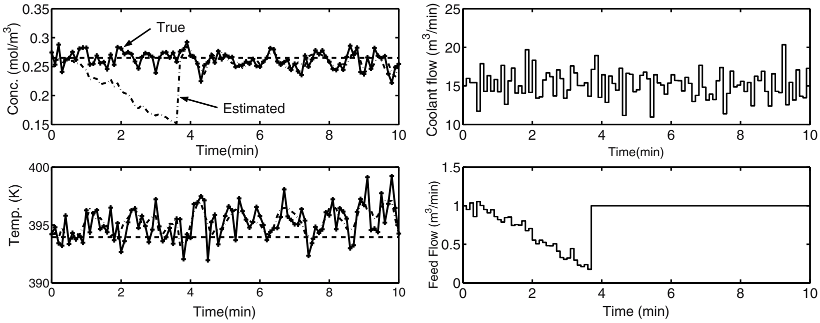
Ten different faults consisting of biases in two measurements, biases in two actuators, failures of the two actuators, failures of two sensors, step change in inlet concentration and change in the frequency factor were hypothesized for this process.

In the conventional NMPC and FTNMPC the control objective is to maintain the temperature close to 393.95^oK, while ensuring that the temperature does not exceed the set-point by more than 1.5^oK, i.e. $T \leq 395.45^oK$ A comparison of performances of conventional NMPC and FTNMPC, when a bias of magnitude -5^oK occurs in the measured temperature at sampling instant $k = 11$, is given in Figure 1(a). In case of NMPC, the true temperature exceeds the constraint limit when the bias occurs. Thus, the conventional NMPC leads to an offset between the true temperature and the set-point as well as violation of constraint. The FTNMPC scheme on the other hand, correctly isolates the fault, compensates for the bias in temperature measurement (estimated magnitude -4.75^oK)



(a) Sensor Bias (b) Inferential control

Fig. 1. Comparison of Closed Loop Responses of NMPC and FTNMPC



(a) True and estimated outputs (b) Manipulated inputs

Fig. 2. FTNMPC behavior for Actuator Failure

and thereby maintains the true temperature within the constraint limit. Thus FTNMPC also eliminates the offset between true temperature and the set-point, as illustrated in Figure 1(a).

It may be expected that the advantages of FTNMPC will become visible in case of inferential control where estimated states are used for control. In order to verify this we simulate failure of sensor for concentration. After the failure is detected and diagnosed, FTNMPC switches over to inferential control using concentration estimates generated using temperature measurements. The comparison of performances of conventional NMPC and FTNMPC, when sensor 1 fails at k =6 and a step jump of 0.5 kmol/m^3 is introduced at k =86 in the inlet concentration (estimated magnitude 0.5233 kmol/m^3) are shown in Figure 1(b). It can be seen that the conventional NMPC results in offset between true concentration and the setpoint. The FTNMPC formulation on the other hand is able to maintain it at the desired setpoint even after the sensor failure and the step change in the disturbance. This can be attributed to the fact that unbiased state estimates are obtained once the faults are correctly identified by the FDI component and model is corrected subsequently to accommodate the faults.

Figure 2(a) shows response of FTNMPC when the actuator for coolant flow is stuck at 1.04 $\mathrm{m}^3/\min$ subsequent to $k = 6$. The corresponding manipulated input variation is shown in Figure 2(b). As evident from Figure 2(a), the state estimation deteriorates subsequent to the failure of the actuator. There is an offset in the true values and the setpoints during the time window used for fault isolation. However, the FDI component correctly isolates the actuator failure and estimates the constant value as 1.037 $\mathrm{m}^3/\min$. Subsequent to on-line correction of the model, the state estimate improves and concentration is again controlled close to the setpoint using the remaining degree of freedom.

5 Conclusions

In this work, a fault tolerant NMPC scheme, equipped with an intelligent state estimator has been proposed. In FTNMPC formulation, to account for plant model mismatch, the corrections to the model are made as and when necessary and at the qualified locations identified by the nonlinear FDI component. The proposed fault accommodation strategy overcomes the limitation on the number of extra states that can be augmented to the state space model in NMPC and FDI formulations and allows bias compensation for more variables than the number of measured outputs. The proposed FTNMPC has significant advantages over the conventional NMPC while dealing with soft faults such as actuator and sensor biases and step jumps in unmeasured disturbances or model parameters. When sensor or actuator failure is isolated, the proposed FTNMPC formulation redefines the controller objectives to accommodate the fault.

References

[1] Yu, Z. H., Li, W., Lee, J. H. and Morari, M. State Estimation Based Model Predictive Control Applied to Shell Control Problem: A Case Study. Chem. Eng. Sci., 14-22, (1994).

[2] Prakash, J., Narasimhan, S., Patwardhan, S.C., “Integrating Model Based Fault Diagnosis with Model Predictive Control”, *Ind. Eng. Chem. Res.*,44,4344-4360,(2005)

[3] Muske, K. R., Edgar, T. F., ?Nonlinear State Estimation?, in *Nonlinear Process Control*, Henson M. A. and D. E. Seborg (Eds.), Prentice Hall, pp 332-340, (1997).

[4] Prakash, J., Patwardhan S. C. and Narasimhan, S., “A Supervisory Approach to Fault Tolerant Control of Linear Multivariable Control systems”, *Ind. Eng. Chem. Res.*, 41, 2270-2281, (2002).

[5] Renganathan, Narasimhan, S., “A Strategy for Detection of Gross Errors in Nonlinear Processes”, *Ind. Eng. Chem. Res.*, 38, 2391-2399, (1999).

[6] Vijaybaskar, R., “Fault Diagnosis and Fault tolerant Control of Nonlinear Systems”, M. S. Dissertation, IIT Madras, (2004).

[7] Marlin, T. E., “Process Control”, Chemical Engineering Series, McGraw-Hill International Editions: New York, (1995).

A Low Dimensional Contractive NMPC Scheme for Nonlinear Systems Stabilization: Theoretical Framework and Numerical Investigation on Relatively Fast Systems

Mazen Alamir

Laboratoire d'Automatique de Grenoble, CNRS-INPG-UJF, BP 46, Domaine
Universitaire, 38400 Saint Martin d'Hères, France
mazen.alamir@inpg.fr

Summary. In this paper, a new contractive receding horizon scheme is proposed for the stabilization of constrained nonlinear systems. The proposed formulation uses a free finite prediction horizon without explicit use of a contraction stability constraint. Another appealing feature is the fact that the resulting receding horizon control is in pure feedback form unlike existing contractive schemes where open-loop phases or a memorized threshold are used to ensure the contraction property in closed loop. The control scheme is validating on the swing-up and stabilization problem of a simple and a double inverted pendulums.

1 Introduction

Since the first rigorous proof of the stability of nonlinear receding horizon control schemes [2], it appeared clearly that the closed loop stability is related to some terminal conditions. The early versions of this terminal constraint took the form of an infinite prediction horizon [2] or an equality constraint on the state [2, 3, 4]. These two forms show evident drawbacks since infinite horizon formulations are impossible to compute for general nonlinear systems while the equality constraints on the state makes the underlying optimization problem hardly tractable numerically. These drawbacks gave rise to formulations where the final state is forced to belong to some TERMINAL REGION of appropriate properties. By doing so, the final equality constrained is replaced by an inequality constraint [5, 6, 9]. It goes without saying that an exhaustive survey of all existing formulations that lead to closed loop stability is beyond the scope of the present paper. An excellent survey can be found in [7].

In this paper, interest is focused on contractive receding horizon schemes [8]. These schemes are based on the assumption according to which there exists a contraction factor $\gamma \in [0,1[$ such that for any initial state x_0 there is a control profile $u(\cdot)$ such that the solution $x_u(\cdot)$ satisfies the contraction constraint $\|x_u(T(x_0))\|_S \leq \gamma\|x_0\|_S$ for some time $T(x_0)$ and some weighting positive definite matrix S. Therefore, given the present state x, the associated open loop optimal control problem is given by [7, 8] :

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 523–535, 2007.
springerlink.com

$$\min_{u(\cdot),T} V(x_u(\cdot,x),T) \quad \text{under} \quad u(\cdot)\in\mathbb{U} \quad \text{and} \quad \|x_u(T,x)\|_S \le \gamma\|x\|_S. \tag{1}$$

Once optimal solutions $\hat{u}(\cdot,x)$ and $\hat{T}(x)$ are obtained, two possible implementations are classically proposed [8]:

X Either the optimal control $\hat{u}(\cdot,x(t))$ is applied in an open-loop way during the interval $[t,t+\hat{T}(x(t))]$. This means that no feedback is applied during $\hat{T}(x(t))$ time units that may be too long.

X Or the state $x(t)$ is memorized together with the duration $\hat{T}(x(t))$ and during the interval $[t,\hat{T}(x(t))]$, a sampling period $\tau>0$ is used such that $N\tau=\hat{T}(x(t))$ and a fixed final time receding horizon scheme is used on $[t,t+\hat{T}(x(t))]$ based on the following optimization problem

$$\min_{u(\cdot)} V(x_u(\cdot,x(t+j\tau))) \qquad \text{under} \quad u(\cdot)\in\mathbb{U}$$
$$\text{and} \quad \|x_u(t+\hat{T}(x(t)),x(t+j\tau))\|_S \le \gamma\|x(t)\|_S, \tag{2}$$

which makes the behavior heavily dependent on the past information $x(t)$ and $\hat{T}(x(t))$ that might become irrelevant due to external disturbances that may even make (2) unfeasible. The aim of the present paper is to propose a contractive scheme that leads to a pure state feedback form without memory effect. This is done using the supremum norm and without an explicit contractive constraint in the problem formulation. Furthermore, the open loop control parametrization is explicitly handled by introducing the notion of translatable parametrization. The paper is organized as follows : Section 2 states the problem and gives some notations and definitions. The proposed contractive formulation is presented in section 3 with the related stability results. Finally section 4 shows some illustrative examples.

2 Definitions, Notations and Problem Statement

Consider the class of nonlinear systems given by

$$\dot{x}=f(x,u) \quad ; \quad x\in\mathbb{R}^n \quad ; \quad u\in\mathbb{R}^m, \tag{3}$$

where x and u stand for the state and the control vectors respectively. $F(t,x_0,\mathbf{u})$ denotes the solution of (3) with initial state x_0 under the control profile $\mathbf{u}$ defined on $[0,t]$. The aim of this paper is to define a sampled state feedback of the form :

$$u(t)=K(x(k\tau_s)) \quad ; \quad \forall t\in[k\tau_s \ (k+1)\tau_s[, \tag{4}$$

that asymptotically stabilizes the equilibrium state $x=0$. The following assumption is needed to establish the main result of this paper:

Assumption 2.1. *For all finite horizon $T > 0$, the following asymptotic property holds :*

$$\lim_{\|x_0\|\to\infty}\left[\min_{\mathbf{u}\in\mathbb{W}^{[0,T]}}\quad\min_{t\in[0,T]}\|F(t,x_0,\mathbf{u})\|\right]=\infty \tag{5}$$

for all compact subset $\mathbb{W} \subset \mathbb{R}^m$. (In other words, infinitely fast state excursions need infinite control) $\flat$

Note that assumption 2.1 is rather technical since it only excludes systems with finite inverse escape time.

2.1 Piece-Wise Constant Control Parametrization

Let some sampling period $\tau_s > 0$ be given. One way to define a low dimensional parametrization of piece-wise constant control profiles over the time interval $[0, N\tau_s]$ that belongs to a closed subset $\mathbb{U} \subset \mathbb{R}^m$ is to follow the following two step procedure :

1. First, define a map

   $$C:\ \mathbb{P}\to\mathbb{R}^m\times\cdots\times\mathbb{R}^m\quad p\rightsquigarrow C(p)=\big(u^1(p),\dots,u^N(p)\big)\quad;\quad u^i(p)\in\mathbb{R}^m.$$

2. Project $C(p)$ on the admissible subset $\mathbb{U}^N$ using the projection map $P_{\mathbb{U}}$, namely :

   $$P_{\mathbb{U}^N}\circ C\colon\ \mathbb{P}\to\mathbb{U}\times\cdots\times\mathbb{U}\quad p\rightsquigarrow P_{\mathbb{U}^N}\circ C(p)=\big(P_{\mathbb{U}}(u^1(p)),\dots,P_{\mathbb{U}}(u^N(p))\big)/$$

3. For all $t \in [(k-1)\tau_s, k\tau_s]$, the control is given by $\mathbf{u}(t) = P_{\mathbb{U}}(u^k(p)) =: \mathcal{U}_{pwc}(t,p)$.

Definition 1. *The map C defined above is called the parametrization map while for given C and $\mathbb{U}$, the family $\big\{\mathcal{U}_{pwc}(\cdot,p)\big\}_{p\in\mathbb{P}}$ is called a $\mathbb{P}$-admissible parametrization of control profiles.* $\flat$

Definition 2. *A $\mathbb{P}$-admissible parametrization is said to be translatable if and only if for each $p \in \mathbb{P}$, there exists some $p^+ \in \mathbb{P}$ such that $u^i(p^+) = u^{i+1}(p)$ for all $i \in \{1,\dots,N-1\}$* $\flat$

Definition 3. *A $\mathbb{P}$-admissible parametrization $\{\mathcal{U}_{pwc}(\cdot,p)\}_{p\in\mathbb{P}}$ is called proper if and only if for all $t_1 < t_2$, one has $\lim_{p\to\infty}\int_{t_1}^{t_2}\|\mathcal{U}_{pwc}(\tau,p)\|^2\,d\tau=\infty$ whenever $\mathbb{P}$ is radially unbounded.* $\flat$

In what follows, the short notation $F(\cdot,x,p)$ is used instead of $F(\cdot,x,\mathcal{U}_{pwc}(\cdot,p))$.

2.2 The Contraction Property

Let some sampling period $\tau_s > 0$ be given together with an associated $\mathbb{P}$-admissible control parametrization $\{\mathcal{U}_{pwc}(\cdot,p)\}_{p\in\mathbb{P}}$.

Definition 4. *The system (3) and the control parametrization $\{\mathcal{U}_{pwc}(\cdot,p)\}_{p\in\mathbb{P}}$ satisfy the contraction property if and only if there exists $\gamma\in]0,1[$ s.t. for all x, there exists $p^c(x)\in\mathbb{P}$ such that :*

$$\min_{q\in\{1,\dots,N\}} \|F(q\tau_s,x,p^c(x))\|^2 \leq \gamma\|x\|^2, \tag{6}$$

where $p_c(\cdot)$ is bounded over bounded sets of initial conditions. If moreover, there exists a continuous function $\varphi:\mathbb{R}^n\rightarrow\mathbb{R}_+$ s.t. for all x :

$$\|F_N(\cdot,x,p^c(x))\|_\infty^2 \leq \varphi(x)\cdot\|x\|^2 \quad \text{where} \quad \|F_q(\cdot,x,p)\|_\infty^2 = \max_{i\in\{1,\dots,q\}} \|F(i\tau_s,x,p)\|^2,$$

then the contraction property is said to be strong. ♭

2.3 Further Notations

For any bounded subset $\mathcal{S}$ of an euclidian space, $\rho(\mathcal{S})$ denotes the radius of $\mathcal{S}$. For all integer $k\in\mathbb{N}$, the notation $k^+ := k+1$ is used. $B(0,r)$ denotes the open ball centered at 0 and of radius r in some euclidian space that is identified from the context. Finally, the projection step is systematically implicitly assumed by writing $u^i(p)$ to denote $P_{\mathbb{U}}(u^i(p))$.

3 A Contractive Receding-Horizon Scheme

In all meaningful and realistic applications, there always exists a set of admissible initial conditions, say $\mathbb{X}\subset\mathbb{R}^n$ that corresponds to realistic initial configurations of the controlled system. Therefore, let such subset $\mathbb{X}\subset\mathbb{R}^n$ be fixed once and for all. Assume that a $\mathbb{P}$-admissible control parametrization is defined and that the strong contraction assumption holds (see definition 4). Associated to the set $\mathbb{X}$ of initial conditions, a subset of admissible control parameters, denoted hereafter by $\mathbb{P}_{\mathbb{X}}$ is defined as follows :

$$\mathbb{P}_{\mathbb{X}} := \mathbb{P}\cap B\Big(0, \sup_{x\in\bar{B}(0,\rho(\mathbb{X}))} \|p^c(x)\| + \varepsilon\Big) \subseteq \mathbb{P} \subseteq \mathbb{R}^{n_p}. \tag{7}$$

Namely, a subset of the ball in $\mathbb{R}^{n_p}$ that contains, among others, all the vectors of parameters :

$$\Big\{p^c(x)\Big\}_{x\in\bar{B}(0,\rho(\mathbb{X}))}$$

invoked in the strong contraction assumption. It goes without saying that since $p^c(\cdot)$ is assumed to exist but is not explicitly known, the exact computation of the radius of the ball defining $\mathbb{P}_{\mathbb{X}}$ cannot be easily done. Therefore, in the forthcoming developments, when $\mathbb{P}_{\mathbb{X}}$ is referred to, it is an superset of it that is to be understood. This superset is obtained by taking a sufficiently high radius for a ball in $\mathbb{R}^{n_p}$ centered at the origin. Consider the following open-loop optimal control problem defined for some $\alpha > 0$ and $\varepsilon > 0$:

$$\begin{aligned} P_\alpha^{\varepsilon,*}(x) \quad : \quad \min_{(q,p)\in\{1,\dots,N\}\times\mathbb{P}_{\mathbb{X}}} \quad & J^*(x,q,p) = \\ & \|F(q\tau_s,x,p)\|^2 + \alpha\frac{q}{N}\cdot\min\bigl\{\varepsilon^2, \|F_q(\cdot,x,p)\|_\infty^2\bigr\}. \end{aligned} \tag{8}$$

Note that if all the functions involved in the definition of the problem (the system's map f and the control parametrization) are continuous then the cost function is continuous in p. This together with the compactness of the set $\mathbb{P}_{\mathbb{X}}$ guarantee that the problem $P_\alpha^{\varepsilon,*}(x)$ admits a solution for all $x \in \mathbb{X}$ and hence is well posed. Therefore, let us denote the solution of (8) for some $x \in \mathbb{X}$ by $\hat{q}(x) \in \{1,\dots,N\}$ and $\hat{p}(x) \in \mathbb{P}_{\mathbb{X}}$. These solutions are then used to define the receding horizon state feedback given by :

$$u(k\tau_s+\tau) = u^1(\hat{p}(x(k\tau_s))) \quad \forall \tau \in [0,\tau_s[. \tag{9}$$

The stability result associated to the resulting feedback strategy is stated in the following proposition :

Proposition 1. If *the following conditions hold :*

1. *The function f in (3) and the parametrization map are continuous and satisfy the strong contraction property (see definition 4). Moreover, the system (3) satisfies assumption 2.1.*
2. *For all $x \in \mathbb{X}$ and all admissible $\mathbf{u} = \mathcal{U}_{pwc}(\cdot,p)$, the solution of (3) is defined for all $t \in [0,N\tau_s]$ and all $p \in \mathbb{P}_{\mathbb{X}}$. (No explosion in finite time shorter than $N\tau_s$).*
3. *The control parametrization is translatable on $\mathbb{P}_{\mathbb{X}}$ in the sense of definition 2.*

Then, *there exist sufficiently small $\varepsilon > 0$ and $\alpha > 0$ such that the receding horizon state feedback (9) associated to the open-loop optimal control problem (8) is well defined and makes the origin $x = 0$ asymptotically stable for the resulting closed loop dynamics with a region of attraction that contains $\mathbb{X}$.* ♭

Proof. The fact that the feedback law is well defined directly results from the continuity of the functions being involved together with the compactness of $\mathbb{P}_{\mathbb{X}}$. Let us denote by $x_{cl}(\cdot)$ the closed loop trajectory under the receding horizon state feedback law. Let us denote by $V(x)$ the optimal value of the cost function, namely : $V(x) = J^*(x,\hat{q}(x),\hat{p}(x))$.

♭ V is continuous $V(x)$ can clearly be written as follows

$$V(x) = \inf\Bigl\{V_1(x),\dots,V_N(x)\Bigr\} \quad ; \quad V_q(x) := \min_{p\in\mathbb{P}_{\mathbb{X}}} J^*(x,q,p). \tag{10}$$

But for given q, $J^*(x,q,p)$ is continuous in (x,p), therefore $V_q(\cdot)$ is a continuous function of x. Since V is the sum of N continuous functions $(V_j)_{j=1,\dots,N}$, it is continuous itself.

- $\flat$ V is radially unbounded
  Since the control parametrization is supposed to be continuous, the set of controls given by :

$$\mathbb{U} := \Big\{\mathcal{U}_{pwc}(t,p)\Big\}_{(t,p)\in[0,N\tau_s]\times\mathbb{P}_\mathbb{X}},$$

  is necessarily bounded. using assumption 2.1 with $\mathbb{W} = \mathbb{U}$ gives the results.
- $\flat$ Finally it is clear that $V(0) = 0$ since zero is an autonomous equilibrium state.

Decreasing properties of V

Two situations have to be distinguished :

Case where $\hat{\mathbf{q}}(\mathbf{k}) > 1$. In this case, let us investigate candidate solutions for the optimization problem $P_\alpha^{\varepsilon,*}(x_{cl}(k^+))$ where $x_{cl}(k^+)$ is the next state on the closed loop trajectory, namely :

$$x_{cl}(k^+) = F\Big(\tau_s, x_{cl}(k), u^1(\hat{p}(x_{cl}(k)))\Big).$$

A natural candidate solution to the optimal control problem $P_\alpha^{\varepsilon,*}(x_{cl}(k^+))$ is the one associated to the translatable character of the control parametrization, namely

$$p_{cand}(k^+) := \hat{p}^+(x_{cl}(k)) \quad ; \quad q_{cand}(k^+) := \hat{q}(x_{cl}(k)) - 1 \geq 1. \tag{11}$$

In the following sequel, the following short notations are used

$$\hat{p}(k) = \hat{p}(x_{cl}(k\tau_s)) \quad ; \quad \hat{q}(k) = \hat{q}(x_{cl}(k\tau_s)) \quad ; \quad V(k) = V(x_{cl}(k)).$$

By the very definition of p^+, it comes that :

$$\begin{aligned} \|F(q_{cand}(k^+)\tau_s, x_{cl}(k^+), p_{cand}(k^+))\|^2 &= \|F(\hat{q}(k)\tau_s, x_{cl}(k), \hat{p}(k))\|^2 \\ = V(x_{cl}(k)) - \alpha\frac{\hat{q}(k)}{N}&\min\{\varepsilon, \|F_{\hat{q}(k)}(\cdot, x_{cl}(k), \hat{p}(k))\|_\infty^2\}, \end{aligned} \tag{12}$$

and since $V(x_{cl}(k^+))$ satisfies by definition, one has :

$$\begin{aligned} V(x_{cl}(k^+)) \leq\ & \|F(q_{cand}(k^+)\tau_s, x_{cl}(k^+), p_{cand}(k^+))\|^2 + \\ & + \alpha\frac{\hat{q}(k)-1}{N}\min\{\varepsilon, \|F_{\hat{q}(k)-1}(\cdot, x_{cl}(k^+), p_{cand}(k^+))\|_\infty^2\}, \end{aligned}$$

This with (12) gives :

$$\begin{aligned} V(x_{cl}(k^+)) \leq V(x_{cl}(k)) - \alpha\frac{\hat{q}(k)}{N}&\min\{\varepsilon, \|F_{\hat{q}(k)}(\cdot, x_{cl}(k), \hat{p}(k))\|_\infty^2\} + \\ + \alpha\frac{\hat{q}(k)-1}{N}&\min\{\varepsilon, \|F_{\hat{q}(k)-1}(\cdot, x_{cl}(k^+), p_{cand}(k^+))\|_\infty^2\}. \end{aligned} \tag{13}$$

But one clearly has by definition of $p_{cand}(k^+)$:

$$\|F_{\hat{q}(k)-1}(\cdot, x_{cl}(k^+), p_{cand}(k^+))\|_\infty^2 \leq \|F_{\hat{q}(k)}(\cdot, x_{cl}(k), \hat{p}(k))\|_\infty^2.$$

Using the last equation in (13) gives

$$V(x_{cl}(k^+)) \leq V(x_{cl}(k)) - \frac{\alpha}{N}\min\{\varepsilon, \|F_{\hat{q}(k)}(\cdot, x_{cl}(k), \hat{p}(k))\|_\infty^2\}. \tag{14}$$

Case where $\hat{\mathbf{q}}(\mathbf{k}) = 1$
We shall first prove that each time this situation occurs, one necessarily has :

$$x_{cl}(k^+) \in \bar{B}(0, \rho(\mathbb{X})). \tag{15}$$

Proof of (15) Consider a sequence of instant $0 = t_0 < t_1 < \cdots < t_N < \dots$ where for all $i \geq 1$, $t_i = k_i\tau_s$ such that $\hat{q}(k_i) = 1$ for all $i \geq 1$ we shall prove the two following facts :

1. $x_{cl}(k_1^+) \in \bar{B}(0, \rho(\mathbb{X}))$
2. If $x_{cl}(k_i^+) \in \bar{B}(0, \rho(\mathbb{X}))$ then $x_{cl}(k_{i+1}^+) \in \bar{B}(0, \rho(\mathbb{X}))$

If these two facts are proved then by induction, it comes that :

$$\{\hat{q}(k) = 1\} \quad \Rightarrow \quad \{x_{cl}(k^+) \in \bar{B}(0, \rho(\mathbb{X}))\}. \tag{16}$$

To prove 1., note that at $k = 0$, $x_{cl}(0) \in \mathbb{X}$ and therefore, the contraction property can be applied to consider $p^c(x_{cl}(0))$ as a candidate value for the initial optimal control problem $P_\alpha^{\varepsilon,*}(x_{cl}(0))$. Therefore,

$$V(x_{cl}(0)) \leq \gamma\|x_{cl}(0)\|^2 + \alpha \cdot \varepsilon. \tag{17}$$

Now during the next steps until k_1 occurs, the result (14) can be used to infer that the function V decreases on the closed loop trajectory. Therefore, one has at instant $k_1\tau_s$:

$$V(x_{cl}(k_1)) \leq \gamma\|x_{cl}(0)\|^2 + \alpha \cdot \varepsilon \quad ; \quad \hat{q}(k_1) = 1. \tag{18}$$

But when $\hat{q}(k_1) = 1$, one has also :

$$\|x_{cl}(k_1^+)\|^2 \leq V(x_{cl}(k_1)) \leq \gamma\|x_{cl}(0)\|^2 + \alpha \cdot \varepsilon, \tag{19}$$

and for sufficiently small α and $\varepsilon > 0$, this leads to $x_{cl}(k_1^+) \in \bar{B}(0, \rho(\mathbb{X}))$ which ends the proof of point 1.

The proof of point 2. follows exactly the same argumentation than the one used above starting from the fact that since $x_{cl}(k_i^+)$ is in $\bar{B}(0, \rho(\mathbb{X}))$, one can rewrite the above demonstration with $x_{cl}(k_i^+)$ playing the role of $x_{cl}(0)$ and $x_{cl}(k_{i+1})$ playing that of $x_{cl}(k_1)$. This clearly gives (16). Consequently, by definition of $\mathbb{P}_\mathbb{X}$, there exists some $p^c(x_{cl}(k^+))$ such that

$$V(x_{cl}(k^+)) \leq \gamma\|x_{cl}(k^+)\|^2 + \frac{\alpha}{N}\min\{\varepsilon, \|F_N(\cdot, x_{cl}(k^+), p^c(x_{cl}(k^+)))\|_\infty^2)\}. \tag{20}$$

But according to the strong contraction assumption, one has :

$$\|F_N(\cdot, x_{cl}(k^+), p^c(x_{cl}(k^+)))\|_\infty^2 \leq \varphi(x_{cl}(k^+)) \cdot \|x_{cl}(k^+)\|^2.$$

therefore (20) becomes ($\lambda := \sup_{\xi \in \bar{B}(0,\rho(\mathbb{X}))} \Big[\varphi(\xi)\Big]$) :

$$\begin{aligned} V(x_{cl}(k^+)) &\leq \gamma \|x_{cl}(k^+)\|^2 + \frac{\alpha}{N} \min\{\varepsilon, \varphi(x_{cl}(k^+)) \cdot \|x_{cl}(k^+)\|^2\}, \\ &\leq \gamma \|x_{cl}(k^+)\|^2 + \frac{\alpha}{N} \min\{\varepsilon, \lambda \cdot \|x_{cl}(k^+)\|^2\}. \end{aligned} \tag{21}$$

On the other hand, since $\hat{q}(k) = 1$ by assumption, one clearly has :

$$\|x_{cl}(k^+)\|^2 \leq V(x_{cl}(k)) - \frac{\alpha}{N} \min\{\varepsilon, \|x_{cl}(k^+)\|^2\} \leq V(x_{cl}(k)). \tag{22}$$

Therefore, using (22) in (21) gives $V(x_{cl}(k^+)) \leq \gamma V(x_{cl}(k)) + \alpha \min\{\varepsilon, \lambda \cdot V(x_{cl}(k))\}$ and one can write $V(x_{cl}(k^+)) \leq (\gamma + \alpha\lambda) V(x_{cl}(k))$ which, for sufficiently small α gives $V(x_{cl}(k^+)) \leq \theta \cdot V(x_{cl}(k))$ for $\theta < 1$. To summarize, it has been shown that the optimal cost function $V(x)$ satisfies the following decreasing properties :

$$V(x_{cl}(k^+)) \leq \begin{cases} V(x_{cl}(k)) - \frac{\alpha}{N} \min\{\varepsilon, \|F_{\hat{q}(k)}(\cdot, x_{cl}(k), \hat{p}(k))\|_\infty^2\} & \text{if } \hat{q}(x_{cl}(k)) > 1 \\ \theta \cdot V(x_{cl}(k)) \quad ; \quad \theta < 1 & \text{if } \hat{q}(x_{cl}(k)) = 1 \end{cases} \tag{23}$$

This clearly shows that the closed loop trajectory converges to the largest invariant set contained in

$$\Big\{x \in \mathbb{R}^n \quad | \quad \|F_{\hat{q}(x)}(\cdot, x, \hat{p}(x))\|_\infty = 0\Big\},$$

which clearly shows that $\lim_{k \to \infty} x_{cl}(k) = 0$ by the very definition of $F_q(\cdot, x, p)$. ♢

Note that proposition 1 shows that the contractive receding horizon feedback may be used alone to asymptotically stabilizes the system. However, in many situations, improved behavior around the desired position may be obtained by using the proposed feedback as a steering controller to bring the state to a neighborhood of the desired target and then to switch to some locally stabilizing controller based (for instance) on linearized model. This is commonly referred to as a dual mode control scheme. In the following section, both ways of using the proposed receding horizon feedback are illustrated on two different systems.

Finally, it is worth noting that all the above discussion remains valid if $\|x\|^2$ [resp. $\|F(t, x, p)\|^2$] are replaced by $h(x)$ [resp. $h(F(t, x, p))$] where $h(\cdot)$ is some positive definite function of the state. In this case, the optimization problem (8) writes :

$$P_\alpha^\varepsilon(x) \ : \ \min_{(q,p) \in \{1,\dots,N\} \times \mathbb{P}_\mathbb{X}} \ J(x, q, p) = h(q\tau_s, x, p) + \alpha \frac{q}{N} \cdot \min\big\{\varepsilon^2, h_q^\infty(\cdot, x, p)\big\}, \tag{24}$$

where $h(q\tau_s, x, p) = h(F(q\tau_s, x, p))$ and $h_q^\infty(\cdot, x, p) := \max_{i \in \{1,\dots,N\}} \ h(i\tau_s, x, p)$.

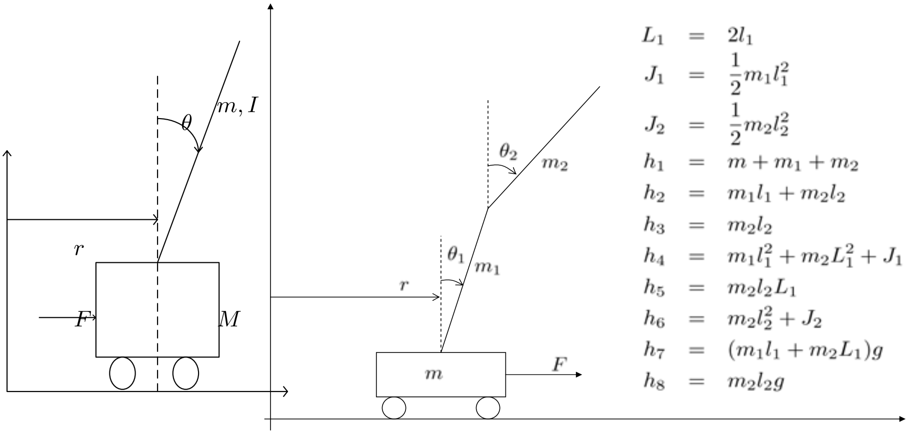


Fig. 1. Description of the simple and the double inverted pendulums

4 Illustrative Examples

4.1 Swing-Up and Stabilization of a Simple Inverted Pendulum on a Cart: A Stand-Alone RHC Scheme

The inverted pendulum on a cart is probably the most famous system in the nonlinear control literature (see figure 1). The dynamics of the inverted pendulum can be described by the following equations :

$$\begin{pmatrix} mL^2 + I & mL\cos\theta \\ mL\cos\theta & m + M \end{pmatrix} \begin{pmatrix} \ddot{\theta} \\ \ddot{r} \end{pmatrix} = \begin{pmatrix} mLg\sin\theta - k_\theta\dot{\theta} \\ F + mL\dot{\theta}^2\sin\theta - k_x\dot{r} \end{pmatrix}. \tag{25}$$

Choosing the state vector $x := \begin{pmatrix}\theta & r & \dot{\theta} & \dot{r}\end{pmatrix}^T \in \mathbb{R}^4$ and applying the following pre-compensation (change in the control variable) :

$$F = -K_{pre}\begin{pmatrix} r \\ \dot{r} \end{pmatrix} + u, \tag{26}$$

where K_{pre} is chosen such that the dynamics $\ddot{r} = -K_{pre}\begin{pmatrix} r \\ \dot{r} \end{pmatrix}$ is asymptotically stable leads to a system of the form (3). Consider the scalar exponential control parametrization (that is clearly translatable with $p^+ = p \cdot e^{-\tau_s/t_r}$) :

$$\mathbb{P} = [p_{min}, p_{max}] \subset \mathbb{R} \quad ; \quad u^i(p) = p \cdot e^{t_i/t_r} \quad ; \quad t_i = \frac{(i-1)\tau_s}{N}, \tag{27}$$

where $\tau_s > 0$ is the control sampling period, $N \cdot \tau_s$ the prediction horizon length while t_r is the characteristic time of the exponential control parametrization.

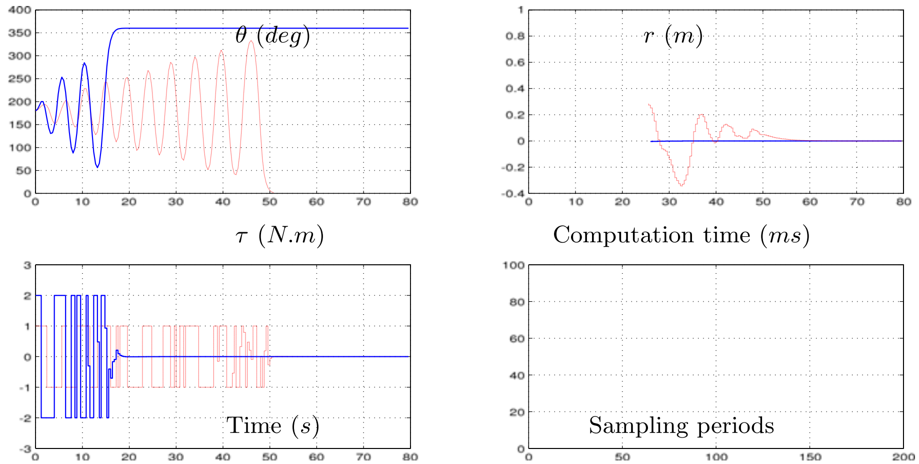


Fig. 2. Stabilization of the inverted pendulum for two different saturation levels: $F_{max} = 1.0\ N$ (dotted thin line) / $F_{max} = 2.0\ N$ (continuous thick line). Initial condition: Downward equilibrium $x = (\pi, 0, 0, 0)^T$.

Define the weighting function $h(x)$ by :

$$h(x) = \frac{1}{2}\left[\dot{\theta}^2 + \beta r^2 + \dot{r}^2\right] + [1 - \cos(\theta)]^2 = \frac{1}{2}\left[x_3^2 + \beta x_2^2 + x_4^2\right] + [1 - \cos(x_1)]^2 \tag{28}$$

In order to explicitly handle the saturation constraint on the force, the constraint has to be expressed in term of the new control variable u, namely :

$$|-K_{pre}\begin{pmatrix} x_2 \\ x_4 \end{pmatrix} + u| \leq F_{max}. \tag{29}$$

Using the expression of the control parametrization (27) this yields the following state dependent definition of the parameter bounds p_{min} and p_{max} :

$$p_{min}(x) = -F_{max} + K_{pre_1}x_2 + K_{pre_2}x_4 \tag{30}$$

$$p_{max}(x) = +F_{max} + K_{pre_1}x_2 + K_{pre_2}x_4 \tag{31}$$

These bounds are used in the definition of the optimization problem $P_\alpha^\varepsilon(x)$:

$$P_\alpha^\varepsilon(x) \quad : \quad \min_{(q,p)\in\{1,\dots,N\}\times[p_{min}(x)\ ,\ p_{max}(x)]} J(x,q,p) = h(q\tau_s, x, p) + \frac{\alpha}{N}\cdot\min\{\varepsilon, h_q^\infty(\cdot, x, p)\}. \tag{32}$$

Let $\hat{p}(x)$ and $\hat{q}(x)$ be optimal solutions of $P_\alpha^\varepsilon(x)$. This defines the feedback $K_{RH}(x) = u^1(\hat{p}(x))$ according to the receding horizon principle. The values of the system's parameters used in the forthcoming simulations are given by :

$$(m, M, L, k_x, k_\theta, I) = (0.3, 5.0, 0.3, 0.001, 0.001, 0.009)$$

while the values of the parameters used in the controller definition are the following :

$$(\tau_s, N, t_r, \alpha, \beta) = (0.4, 8, 0.2, 0.01, 10)\ ;\ K_{pre} = (2.5, 10)\ ;\ F_{max} \in \{1, 2\}$$

The behavior of the closed loop systems under the contractive receding horizon control is depicted on figure 2. Two scenarios are presented for different values of the input saturation levels $F_{max} = 1$ and $F_{max} = 2$. The computation times are also given vs the sampling period (the computations have been performed on a 1.3 GHz PC-Pentium III). Note that these computation times never exceeded 0.1 s. This has to be compared to the sampling period $\tau_s = 0.4$ s. This suggests that the proposed receding horizon feedback can be implementable in real time context.

4.2 Swing Up and Stabilization of a Double Inverted Pendulum on Cart: A Hybrid Scheme

The system is depicted on figure 1 together with the definition of some auxiliary variables. The numerical values are given by :

$$(m_1, m_2, m, l_1, l_2, J_1, J_2) = (0.3, 0.2, 5.0, 0.3, 0.2, 1.3 \times 10^{-2}, 4 \times 10^{-3}).$$

The system equations are given by [1] :

$$\begin{aligned}
h_1\ddot{r} + h_2\ddot{\theta}_1\cos\theta_1 + h_3\ddot{\theta}_2\cos\theta_2 &= h_2\dot{\theta}_1^2\sin\theta_1 + h_3\dot{\theta}_2^2\sin\theta_2 + F\\
h_2\ddot{r}\cos\theta_1 + h_4\ddot{\theta}_1 + h_5\ddot{\theta}_2\cos(\theta_1 - \theta_2) &= h_7\sin\theta_1 - h_5\dot{\theta}_2^2\sin(\theta_1 - \theta_2)\\
h_3\ddot{r}\cos\theta_2 + h_5\ddot{\theta}_1\cos(\theta_1 - \theta_2) + h_6\ddot{\theta}_2 &= h_5\dot{\theta}_1^2\sin(\theta_1 - \theta_2) + h_8\sin\theta_2
\end{aligned}$$

Again, a pre-compensation is done using the change in control variable given by :

$$F = -K_{pre} \cdot \begin{pmatrix} r \\ \dot{r} \end{pmatrix} + u, \tag{33}$$

while a two-dimensional control parametrization is needed this time :

$$\mathbb{P} = [p_{min}, p_{max}]^2 \subset \mathbb{R}^2 \quad ; \quad u^i(p) = p_1 \cdot e^{\lambda_1 t_i} + p_2 e^{-\lambda_2 t_i} \quad ; \quad t_i = \frac{(i-1)\tau_s}{N} \tag{34}$$

The weighting function $h(\cdot)$ invoked in the general formulation (24) is here taken as follows

$$h(x) = \frac{h_4}{2}\dot{\theta}_1^2 + \frac{h_6}{2}\dot{\theta}_2^2 + h_5\dot{\theta}_1\dot{\theta}_2\cos(\theta_1 - \theta_2) + h_7\big[1 - \cos(\theta_1)\big] + h_8\big[1 - \cos(\theta_2)\big] + \\ + h_1\big[r^2 + \dot{r}^2\big].$$

This is inspired by the expression of the total energy given in [1]. The constrained open-loop optimal control problem is then given by (24) in which the admissible domain of the parameter vector is $[p_{min}(x), p_{max}(x)]^2$ where :

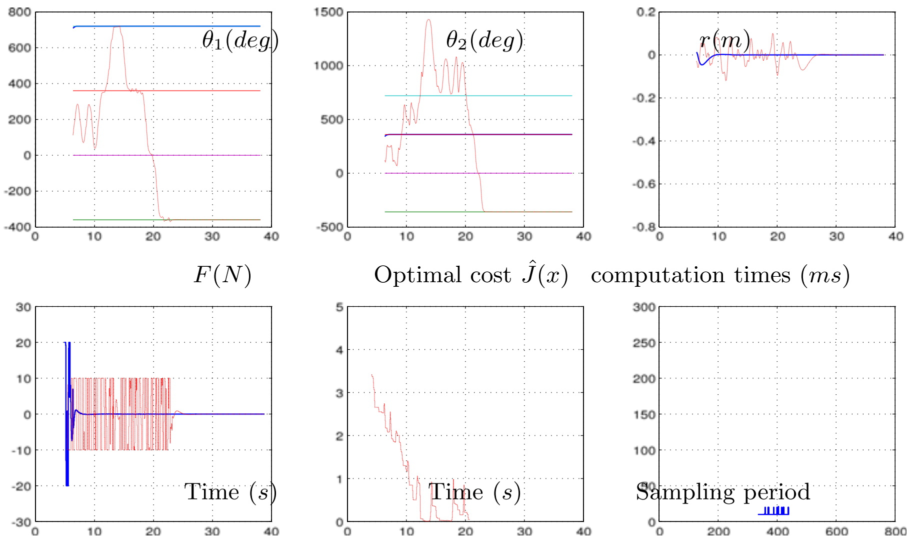


Fig. 3. Closed loop behavior of the double inverted pendulum system under the hybrid controller given by (35) with the design parameters values given by $(\tau_s, N, R, \lambda_1, \lambda_2, \eta) = (0.3, 10, 100, 100, 20, 1)$, $L = (360, 30)$, and $Q = diag(1, 1, 10^4, 1, 1, 1)$ for two different force saturation levels: $F_{max} = 20\ N$ (continuous thick line) / $F_{max} = 10\ N$ (dotted thin line). The maximum number of function evaluations parameter in the optimization code has been set to 20 in order to make the solution real-time implementable. This may explain the behavior of the optimal cost for the lower values that is not monotonically decreasing. Initial condition: downward equilibrium.

$$p_{min}(x) := \frac{1}{2}\Big[-F_{max} + K_{pre}\begin{pmatrix} r \\ \dot{r} \end{pmatrix}\Big] \quad ; \quad p_{max}(x) := \frac{1}{2}\Big[+F_{max} + K_{pre}\begin{pmatrix} r \\ \dot{r} \end{pmatrix}\Big].$$

that clearly enables to meet the requirement $|F(t)| \le F_{max}$ given the parametrization (34) being used. Again, denoting by $(\hat{q}(x), \hat{p}(x))$ the optimal solutions, the nonlinear receding-horizon control is given by :

$$u(k\tau_s + t) = K_{RH}(x(k\tau_s)) := u^1(\hat{p}(x(k\tau_s))) \quad ; \quad t \in [0, \tau_s[.$$

Since a hybrid scheme is used here, the local controller has to be defined. This is done by using an LQR-based method that enables a feedback gain L to be computed. Hence, the local controller is given by $K_L(x) = -L \cdot \big(x_1^m\ x_2^m\ x_3\ \ldots\ x_6\big)^T$ where x_1^m and x_2^m are the minimum norm angles that are equal (modulo 2π) to θ_1 and θ_2 respectively while the gain matrix $L \in \mathbb{R}^{1\times 6}$ satisfies the following Riccati equation for some positive definite matrices S and Q :

$$A_d^T S A_d - S - (A_d^T S B_d)(R + B_d^T S B_d^T)(B_d^T S A_d) + Q = 0.$$

where (A_d, B_d) are the matrices of the discrete linearized system around the upward position. To summarize, the hybrid controller is given by

$$u(k\tau_s + \tau) = \begin{cases} K_{RH}(x(k\tau_s)) & \text{if} \quad \|x(k\tau_s)\|_S^2 > \eta \\ K_L(x(k\tau_s)) & \text{otherwise} \end{cases} \tag{35}$$

The positive real $\eta > 0$ is a threshold that must be sufficiently small for the ball

$$B_\eta := \Big\{ x \in \mathbb{R}^6 \quad \text{s.t} \quad \|x\|_S^2 \leq \eta \Big\}$$

to be both entirely included in the region of attraction and invariant under the linear control law $K_L(\cdot)$. Such $\eta > 0$ clearly exists.

The behavior of the closed loop system under the hybrid controller is shown on figure 3 for two different saturation levels $F_{max} = 10\ N$ and $F_{max} = 20\ N$. Note that the maximum number of function evaluations during the on-line optimization has been set to 20. This led to computation times that never exceeded the sampling period $\tau_s = 0.3\ s$.

References

[1] Zhong, W. and Rock H., “Energy and passivity based control of the double inverted pendulum on a cart”, *Proceedings of the 2001 IEEE Conference on Decision and Control, Mexico.*, 896-901, (2001).

[2] Keerthi, S. S. and E. G. Gilbert, “Optimal infinite horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving horizon approximations”, *Journal of Optimization Theory and Applications*, **volume 57**, 265-293, (1988).

[3] Mayne, D. Q. and H. Michalska, “Receding Horizon Control of Nonlinear Systems”, *IEEE Transactions on Automatic Control*, **volume 35**, 814-824, (1990).

[4] Meadows, E. and M. Henson and J. Eaton and J. Rawlings, “Receding Horizon Control and Discontinuous State Feedback Stabilization”, *Int. Journal of Control*, **volume 62** 1217-1229, (1995).

[5] Michalska, H. and D. Q. Mayne, “Robust Receding Horizon Control of Constrained Nonlinear Systems”, *IEEE Transactions on Automatic Control*, **volume 38** 1623-1632, (1993).

[6] De Nicolao, G. and L. Magnani and L. Magni and R. Scattolini, “On Stabilizing Receding Horizon Control for Nonlinear Discrete Time Systems”, *Proceedings of the 38th IEEE Conference on Decision and Control* (1995).

[7] Mayne, D. Q. and J. B. Rawlings and C. V. Rao and P. O. M. Scokaert, “Constrained Model Predictive Control: Stability and Optimality”, *Automatica*, **volume 36** 789-814, (2000).

[8] Kothare, S. L. de Oliveira and Morari, M., “Contractive Model Predictive Control for Constrained Nonlinear Systems”, *IEEE Transcations on Automatic Control*, **volume 45** 1053-1071, (2000).

[9] Chen, H. and Allgöwer, “A quasi-infinite horizon nonlinear controller model predictive control scheme with guaranteed stability”, *Automatica*, **volume 34** 1205-1218, (1998).

A New Real-Time Method for Nonlinear Model Predictive Control

Darryl DeHaan and Martin Guay

Department of Chemical Engineering, Queen's University, Kingston, Ontario,
Canada, K7L 3N6
{dehaan,guaym}@chee.queensu.ca

Summary. A formulation of continuous-time nonlinear MPC is proposed in which input trajectories are described by general time-varying parameterizations. The approach entails a limiting case of suboptimal single-shooting, in which the dynamics of the associated NLP are allowed to evolve within the same timescale as the process dynamics, resulting in a unique type of continuous-time dynamic state feedback which is proven to preserve stability and feasibility.

1 Introduction

In this note we study the continuous-time evolution of nonlinear model predictive control in cases where the optimization must necessarily evolve in the same timescale as the process dynamics. This is particularly relevant for applications involving "fast" dynamics such as those found in aerospace, automotive, or robotics applications in which the computational lag associated with iterative optimization algorithms significantly limits the application of predictive control approaches.

In an attempt to reduce computational lag, interest has been focussed on the use of suboptimal solutions arrived at by early termination of the nonlinear program being solved online. Real-time computational algorithms such as [1] push this concept to evaluating only a single NLP iteration per discrete sampling interval. A similar concept of incremental improvement underlies realtime works such as [2, 3], where the input parameters are treated as evolving according to continuous-time differential equations driven by descent-based vector fields. In particular, [3] illustrates how this approach is effectively a type of adaptive feedback.

In this work, we present a form of real-time MPC which, in the spirit of [2] and [3], treats the evolving optimization as an adaptive control action. However, our results are more general in that we do not require global asymptotic stability of the unforced dynamics (unlike [3]), and our approach preserves stability without requiring "sufficiently many" parameters in the description of the input (unlike [2]). One important aspect of our approach is that the open-loop parameterization of the input is defined relative to a time partition that can potentially be

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 537–550, 2007.
springerlink.com

adapted online to make optimal use of the finite number of parameters used to describe the input. While the manner in which the input is parameterized has similarities to sampled-data approaches such as [4], a key difference is that our approach involves continuous measurement and control implementation throughout the intervals of the time partition, and as a result there are no intervals of open-loop behaviour introduced into the feedback path.

This paper is organized as follows. The basic problem is described in Section 2, with finite input parameterizations and local stabilizing controllers discussed in Sections 3 and 4, respectively. Section 5 discusses the realtime design approach, with an example in Section 6. Proofs are in the Appendix. In the following, we will use the notation $\mathring{\mathbb{S}}$ to denote the open interior of a closed set $\mathbb{S}$, and $\partial\mathbb{S}$ for the boundary $\mathbb{S} \setminus \mathring{\mathbb{S}}$. Furthermore, we denote by $\|z\|_{\mathbb{S}}$ the orthogonal distance of a point z to the set $\mathbb{S}$; i.e. $\|z\|_{\mathbb{S}} = \inf_{s\in\mathbb{S}} \|z - s\|$. A continuous function $\gamma : [0, \infty) \to \mathbb{R}_{\geq 0}$ is defined as class $\mathcal{K}$ if it is strictly increasing from $\gamma(0) = 0$, and class $\mathcal{K}_\infty$ if it is furthermore radially unbounded. Finally, a function will be described as C^{m+} if it is C^m, with all derivatives of order m yielding locally Lipschitz functions.

2 Problem Setup

Our control objective is the regulation of the dynamics

$$\dot{x} = f(x, u) \tag{1}$$

to the compact target set $\Sigma_{\mathbb{X}} \subset \mathbb{R}^n$, which is assumed to be weakly invariant for controls in some compact set $u \in \Sigma_{\mathbb{U}}(x) \subset \mathbb{R}^m$; i.e. there exists a static feedback rendering the set $\Sigma \triangleq \{(x, u) \in \Sigma_{\mathbb{X}} \times \mathbb{R}^m | u \in \Sigma_{\mathbb{U}}(x)\}$ forward invariant. Set stabilization allows for more general control problems than simple stabilization to a point, and in particular encompasses the notion of "practical-stabilization". We are interested in continuous-time model predictive control problems of the form

$$\min_{u(\cdot)} \left\{ \int_t^{t+T} L(x^p, u)\, d\tau + W(x^p(t+T)) \right\} \tag{2a}$$

$$s.t. \quad \dot{x}^p = f(x^p, u), \qquad x^p(t) = x \tag{2b}$$

$$(x^p, u) \in \mathbb{X} \times \mathbb{U}, \quad \forall \tau \in [t,\, t+T] \tag{2c}$$

$$x^p(t+T) \in \mathcal{X}_f. \tag{2d}$$

Since the motivating problem of interest is assumed to involve an infinite horizon, the horizon length in (2a) is interpreted as designer-specifiable. The sets $\mathbb{X} \subset \mathbb{R}^n$ and $\mathbb{U} \subset \mathbb{R}^m$ represent pointwise-in-time constraints, and are assumed to be compact, connected, of non-zero measure (i.e. $\mathring{\mathbb{X}}, \mathring{\mathbb{U}} \neq \emptyset$), and to satisfy the containment $\Sigma \subset \mathring{\mathbb{X}} \times \mathring{\mathbb{U}}$. The compact, connected terminal set $\mathcal{X}_f$ is typically designer-specified, and is assumed to strictly satisfy $\Sigma_{\mathbb{X}} \subset \mathcal{X}_f \subset \mathring{\mathbb{X}}$. The mapping

$L : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_{\geq 0}$ is assumed to satisfy $\gamma_L(\|x, u\|_\Sigma) \leq L(x, u) \leq \gamma_U(\|x, u\|_\Sigma)$ for some $\gamma_L, \gamma_H \in \mathcal{K}_\infty$, although this could be relaxed to an appropriate detectability condition. The mapping $W : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$ is assumed to be positive semi-definite, and identically zero on the set $\Sigma_\mathbb{X} \subset \mathcal{X}_f$. For the purposes of this paper, the functions $L(\cdot, \cdot)$, $W(\cdot)$ and $f(\cdot, \cdot)$ are all assumed to be C^{1+} on their respective domains of definition, although this could be relaxed to locally Lipschitz with relative ease.

3 Finite-Dimensional Input Parameterizations

Increasing horizon length has definite benefits in terms of optimality and stability of the closed loop process. However, while a longer horizon obviously increases the computation time for model predictions, of significantly greater computational concern are the additional degrees of freedom introduced into the minimization in (2a). This implies that instead of enforcing a constant horizon length, it may be more beneficial to instead maintain a constant number of input parameters whose distribution across the prediction interval can be varied according to how "active" or "tame" the dynamics may be in different regions.

Towards this end, it is assumed that the prediction horizon is partitioned into N intervals of the form $[t^\theta_{i-1},\ t^\theta_i]$, $i = 1 \dots N$, with $t \in [t^\theta_0,\ t^\theta_1]$. The input trajectory $u : [t^\theta_0,\ t^\theta_N] \to \mathbb{R}^m$ is then defined in the following piecewise manner

$$u(\tau) = u_\phi(\tau, t^\theta, \theta, \phi) \triangleq \begin{cases} \phi(\tau - t^\theta_0,\ \theta_1) & \tau \in [t^\theta_0,\ t^\theta_1] \\ \phi(\tau - t^\theta_{i-1},\ \theta_i) & \tau \in (t^\theta_{i-1},\ t^\theta_i],\ i \in \{2 \dots N\} \end{cases} \tag{3}$$

with individual parameter vectors $\theta_i \in \Theta \subset \mathbb{R}^{n_\theta}$, $n_\theta \geq m$, for each interval, and $\theta = \{\theta_i \,|\, i \in \{1, \dots N\}\} \in \Theta^N$. The function $\phi : \mathbb{R}_{\geq 0} \times \Theta \to \mathbb{R}^m$ may consist of any smoothly parameterized (vector-valued) basis in time, including such choices as constants, polynomials, exponentials, radial bases, etc. In the remainder, a (*control-* or *input-*) *parameterization* shall refer to a triple $\mathcal{P} \triangleq (\phi,\ \mathbb{R}^{N+1},\ \Theta^N)$ with specified N, although this definition may be abused at times to refer to the family of input trajectories spanned by this triple (i.e. the set-valued range of $\phi(\mathbb{R}^{N+1}, \Theta^N)$).

Assumption 1. *The C^{1+} mapping $\phi : \mathbb{R}_{\geq 0} \times \Theta \to \mathbb{R}^m$ and the set Θ are such that 1) Θ is compact and convex, and 2) the image of Θ under ϕ satisfies $\mathbb{U} \subseteq \phi(0, \Theta)$.*

Let $(t_0,\ x_0) \in \mathbb{R} \times \overset{\circ}{\mathbb{X}}$ represent an arbitrary initial condition for system (1), and let $(t^\theta,\ \theta)$ be an arbitrary choice of parameters corresponding to some parameterization $\mathcal{P}$. We denote the resulting solution to the prediction model in (2b), defined on some maximal subinterval of $[t_0,\ t^\theta_N]$, by $x^p(\cdot, t_0, x_0, t^\theta, \theta, \phi)$. At times we will condense this notation, and that of (3), to $x^p(\tau)$, $u_\phi(\tau)$.

A particular choice of control parameters (t^θ, θ) corresponding to some parameterization $\mathcal{P}$ will be called *feasible* with respect to $(t_0,\ x_0)$ if, for every

$\tau \in [t_0,\, t_N^\theta]$, the solution $x^p(\tau, t_0, x_0, t^\theta, \theta, \phi)$ exists and satisfies $x^p(\tau) \in \mathring{\mathbb{X}}$, $u_\phi(\tau) \in \mathring{\mathbb{U}}$, and $x^p(t_N^\theta) \in \mathring{\mathcal{X}}_f$. We let $\Phi(t_0, x_0, \mathcal{P}) \subseteq \mathbb{R}^{N+1} \times \Theta^N$ denote the set of all such feasible parameter values for a given (t_0, x_0) and parameterization $\mathcal{P}$. This leads to the following result, which is a straightforward extension of a similar result in [5].

Lemma 1. *Let $\mathbb{X}^0 \subseteq \mathring{\mathbb{X}}$ denote the set of initial states x_0 for which there exists open-loop pairs $(x(\cdot),\, u(\cdot))$ solving* (1)*, defined on some interval $t \in [t_0,\, t_f]$ (on which $u(\cdot)$ has a finite number of discontinuities), and satisfying the constraints $x(t_f) \in \mathring{\mathcal{X}}_f$, and $(x,\, u)(t) \in \mathring{\mathbb{X}} \times \mathring{\mathbb{U}}$, $\forall t \in [t_0,\, t_f]$. Then, for every $(t_0,\, x_0) \in \mathbb{R} \times \mathbb{X}^0$ and every $(\phi,\, \Theta)$ satisfying Assumption 1, there exists $N^* \equiv N^*(x_0, \phi, \Theta)$ such that $\Phi(t_0, x_0, \mathcal{P})$ has positive Lebesgue measure in $\mathbb{R}^{N+1} \times \Theta^N$ for all $N \geq N^*$.*

4 Requirements for a Local Stabilizing Control Law

Sufficient conditions for stability of NMPC presented in [6] require that $\mathcal{X}_f$ be a control-invariant set, and that the function $W(\cdot)$ be a control Lyapunov function on the domain $\mathcal{X}_f$. The following assumption represents a slight strengthening of those conditions - presented in integral rather than differential form - as applicable to the input parameterizations from the preceding section. In particular, a pair of feedbacks satisfying the assumption are required to be explicitly known, and the strict decrease in (4) is added to enable the use of interior-point methods for constraint handling.

Assumption 2. *The penalty $W : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$, the sets $\mathcal{X}_f$ and Σ, the mapping ϕ, and a pair of known feedbacks $\delta : \mathcal{X}_f \to \mathbb{R}_{>0}$ and $\kappa : \mathcal{X}_f \to \Theta$ are all chosen s.t.*

1. *$\Sigma_{\mathbb{X}} \subset \mathring{\mathcal{X}}_f$, $\mathcal{X}_f \subset \mathring{\mathbb{X}}$, both $\mathcal{X}_f$ and Σ compact.*
2. *there exists a compact set $\mathbb{U}^0 \subset \mathring{\mathbb{U}}$ s.t. $\forall x \in \mathcal{X}_f$, $\sup\limits_{\tau \in [0,\, \delta(x)]} \|\phi(\tau, \kappa(x))\|_{\mathbb{U}^0} = 0$.*
3. *Σ and $\mathcal{X}_f$ are both rendered positive invariant in the following sense:*
   - *there exists a constant $\varepsilon_\delta > 0$ such that $\delta(x_0) \geq \varepsilon_\delta$ for all $x_0 \in \mathcal{X}_f$.*
   - *for every $x_0 \in \Sigma_{\mathbb{X}}$, the (open-loop) solution to $\dot{x}_\kappa = f(x_\kappa, \phi(\tau_\kappa, \kappa(x_0)))$, $x_\kappa(0) = x_0$ exists and satisfies $(x_\kappa(\tau_\kappa), \phi(\tau_\kappa, \kappa(x_0))) \in \Sigma$ for $\tau_\kappa \in [0,\, \delta(x_0)]$.*
   - *$\exists \varepsilon^* > 0$ and a family of sets $\mathcal{X}_f^\varepsilon = \{\, x \in \mathcal{X}_f \ :\ \inf_{s \in \partial \mathcal{X}_f} \|s - x\| \geq \varepsilon \,\}$, $\varepsilon \in [0,\, \varepsilon^*]$, such that $x_0 \in \mathcal{X}_f^\varepsilon \implies x_\kappa(t) \in \mathcal{X}_f^\varepsilon, \forall t \in [0,\, \delta(x_0)]$, $\forall \varepsilon \in [0,\, \varepsilon^*]$*
4. *there exists $\gamma \in \mathcal{K}$ such that for all $x_0 \in \mathcal{X}_f$, (with $x_f \triangleq x_\kappa(\delta(x_0))$),*

$$W(x_f) - W(x_0) + \int_0^{\delta(x_0)} L(x_\kappa, \phi(\tau, \kappa(x_0)))\, d\tau \leq -\int_0^{\delta(x_0)} \gamma(\|x_\kappa\|_{\Sigma_{\mathbb{X}}}) d\tau \qquad (4)$$

4.1 Design Considerations

For the purposes of this work, any locally stabilizing pair $(\kappa,\, \delta)$ satisfying Assumption 2 can be used. For the case where ϕ is a piecewise-constant parameterization, several different approaches exist in the literature for the design of such feedbacks (see [7, 8] and references therein). Below we present one possible extension of these approaches for finding κ and δ in the case of more general parameterizations.

1. Assume that a known feedback $u = k_f(x)$ and associated CLF $W(x)$ satisfy

$$\frac{\partial W}{\partial x} f(x, k_f(x)) + L(x, k_f(x)) \le -\gamma_k(\|x\|_{\Sigma_{\mathbb{X}}}) \qquad \forall x \in \mathcal{X}_f \tag{5}$$

   for some $\gamma_k \in \mathcal{K}$, with $\mathring{\Sigma} \neq \emptyset$ (if necessary, take Σ as a small neighbourhood of the true target). Let Σ^ε denote a family of nested inner approximations of Σ. For some $\varepsilon^* > 0$, the sets $\mathcal{X}_f^\varepsilon$ and Σ^ε are assumed forward-invariant with respect to $\dot{x} = f(x, k_f(x))$, and $k_f(x) \in \mathbb{U}^0$ for all $x \in \mathcal{X}_f$, $\varepsilon \in [0, \varepsilon^*]$.
2. Without loss of generality, assume a number $r \in \{0, 1, \ldots, \text{floor}(n_\theta/m) - 1\}$ is known such that $k_f \in C^{r+}$, and

$$\text{span}_{\theta_i \in \Theta} \begin{bmatrix} \phi(0, \theta_i) \\ \vdots \\ \frac{\partial^r \phi}{\partial \tau^r}(0, \theta_i) \end{bmatrix} = \mathbb{U} \oplus \mathbb{R}^{rm}. \tag{6}$$

   Select any C^{1+} mapping $\kappa(x) : \mathcal{X}_f \to \{ \varpi \in \Theta \ : \ \varpi \text{ satisfies (7) for } x \}$, whose range is nonempty by (6) and Assumption 1. (i.e. invert the function $\phi(0, \cdot)$)

$$\begin{bmatrix} k_f(x) \\ \frac{\partial k_f}{\partial x} f(x, k_f(x)) \\ \vdots \\ L_f^r k_f \end{bmatrix} = \begin{bmatrix} \phi(0, \varpi) \\ \frac{\partial \phi}{\partial \tau}(0, \varpi) \\ \vdots \\ \frac{\partial^r \phi}{\partial \tau^r}(0, \varpi) \end{bmatrix} \tag{7}$$

3. Specify $\gamma = \frac{1}{2}\gamma_k$, and simulate the dynamics forward from $x_\kappa(0) = x$ under control $u = \phi(\tau_\kappa, \omega)$ until one of the conditions in Assumption 2 fails, at a time $\tau_\kappa = \delta^*$. Set $\delta(x) = c_\delta \delta^*$, for any $c_\delta \in (0, 1)$.

This approach effectively assigns $\kappa(x)$ by fitting a series approximation of order r to the input trajectory generated by $u = k_f(x)$. By the invariance (and compactness) of the inner approximations $\mathcal{X}_f^\varepsilon$ and Σ^ε for some $\varepsilon^* > 0$, a lower bound $\varepsilon_\delta \equiv \varepsilon_\delta(\varepsilon^*) > 0$ exists such that $\delta(x) \ge c_\delta \varepsilon_\delta$, $\forall x \in \mathcal{X}_f$. In contrast, a similar problem of initializing input trajectories is solved in [4] by using forward simulation of the dynamics $\dot{x} = f(x, k_f(x))$ to generate $u(t)$. Within our framework, however, it could be difficult to ensure that these generated trajectories lie within the span of $\mathcal{P}$.

5 Real-Time Design Approach

5.1 Constraint Handling

While both active-set and interior-point approaches have been successfully used to handle constraints in NMPC problems, one limitation of using active sets within the context of our realtime framework is that constraint violation can only be tested at discrete, pre-defined points in time along the prediction interval. In contrast, interior point approaches such as [9] preserve constraint feasibility all points along the prediction trajectory, which is advantageous when the time support t^θ is nonuniform and potentially involves large intervals. A second benefit of using interior-point methods is that nominal robustness in the presence of state constraints is guaranteed automatically, whereas it is shown in [10] that active set approaches must be modified to use interior approximations of the constraint in order to guarantee nominal robustness. To this end, the constraints are incorporated defining

$$L^a(x,u) = L(x,u) + \mu\left(B_x(x) + B_u(u)\right), \qquad W^a(x_f) = W(x_f) + \mu\, B_{x_f}(x_f) \tag{8}$$

where $\mu > 0$ is a design constant, and B_x, B_u, B_{x_f} are barrier functions on the respective domains $\mathbb{X}$, $\mathbb{U}$ and $\mathcal{X}_f$. For the purposes of this work, it is assumed that the barrier functions are selected a-priori to satisfy the following minimum criteria, where the pair $(s,\mathbb{S})$ is understood to represent $\{(x,\mathbb{X})$, $(u,\mathbb{U})$, $(x_f,\mathcal{X}_f)\}$.[-1mm]

Criterion 1. *The individual barrier functions each satisfy*

1. $B_s : \mathbb{S} \to \mathbb{R}_{\geq 0} \cup \{\infty\}$*, and* B_s *is* C^{1+} *on the open set* $\mathring{\mathbb{S}}$.
2. $s \to \partial\mathbb{S}$ *(from within) implies* $B_s(s) \to \infty$.
3. $B_s \equiv 0$ *on* $s \in \Sigma_{\mathbb{S}}$*, and* $B_s \geq 0$ *on* $s \in \mathbb{S} \setminus \Sigma_{\mathbb{S}}$.

The assumed differentiability of B_s is for convenience, and could be relaxed to locally Lipschitz. We note that additional properties such as convexity of $\mathbb{S}$ and B_s or self-concordance of B_s (see [9, 11]) are not technically required, although in practice they are highly advantageous. The third criterion implies that the B_s is "centered" around the target set Σ. For basic regulation problems ($\Sigma = \{(0,0)\}$) with convex constraints a self concordance-preserving recentering technique is given in [9], which could be extended to more general Σ, but likely at the expense of self-concordance. For nonconvex constraints, a barrier function satisfying Criterion 1 must be designed directly. In addition to the above criteria, it must be ensured that substituting (8) does not compromise the stability condition (4). Thus we require:

Criterion 2. *For a given local stabilizer satisfying Assumption 2, the barrier functions* B_x*,* B_u*,* B_{x_f} *and multiplier* μ *are chosen to satisfy, for all* $x \in \mathcal{X}_f$*,*

$$\sup_{(\tau,\,x_0)\in\mathcal{I}(x)} \left\{\nabla B_{x_f}(x)^T f(x,\phi(\tau,\kappa(x_0))) + B_x(x) + B_u(\phi(\tau,\kappa(x_0)))\right\} \leq \frac{1}{\mu}\gamma(\|x\|_{\Sigma_{\mathbb{X}}})$$

$$\mathcal{I}(x) \triangleq \{\ (\tau,\ x_0) \in [0,\ \delta(x_0)] \times \mathcal{X}_f\ :\ \dot{x}_\kappa = f(x_\kappa,\phi(t,x_0)),\ x_\kappa(0) = x_0 \text{ and } x_\kappa(\tau) = x\ \} \tag{9}$$

In general, Criterion 2 can be readily satisfied if 1) level curves of $B_{\mathcal{X}_f}$ are invariant; i.e. they align with level curves of W, 2) μ is chosen sufficiently small, and 3) the growth rates of B_x and $\phi \circ B_u$ are less than that of γ in an open neighbourhood of Σ. When using the design approach for κ and δ in Section 4.1, one can treat Criterion 2 as a constraint on the interval length $\delta(x)$ by designing the barriers to satisfy

$$\nabla B_{x_f}(x)^T f(x, k_f(x)) + B_x(x) + B_u(k_f(x)) < \frac{1}{\mu}\gamma(\|x\|_{\Sigma_{\mathbb{X}}}) \ . \tag{10}$$

5.2 Description of Closed-Loop Behaviour

Before detailing our MPC controller, it will be useful to denote $z \triangleq [x^T,\, t^{\theta^T},\, \theta^T]^T$ as the vector of closed-loop states. The cost function is then defined as

$$\begin{aligned} J(t,z) \quad &= \textstyle\int_t^{t_N^\theta} L^a(x^p(\tau), u_\phi(\tau))\, d\tau + W^a(x^p(t_N^\theta)) && \text{(11a)} \\ s.t. \quad & \tfrac{dx^p}{d\tau} = f(x^p, u_\phi(\tau, z, \phi)), \qquad x^p|_{\tau=t} = x \ . && \text{(11b)} \end{aligned}$$

Step 1: Initialization of t^θ and θ
Let $(t_0,\, x_0) \in \mathbb{R} \times \mathbb{X}^0$ denote an arbitrary feasible initial condition for (1). The first step is to initialize the control parameters to any value in the feasible set $\Phi(t_0, x_0, \mathcal{P})$, which is guaranteed by Lemma 1 to be tractable. In the simple case where $\mathbb{X}^0 \subseteq \mathcal{X}_f$, then feasible parameter values can be obtained from forward simulation of the dynamics under the feedbacks $\kappa(\cdot)$ and $\delta(\cdot)$; otherwise a dual programming program could be solved to identify feasible initial parameter values.

Step 2: Continuous flow under dynamic feedback
At any instant $t \in [t_0,\, t_1^\theta]$ we assume that the model prediction $x^p(\tau, t, z, \phi)$ is 'instantaneously' available. This prediction information is used to update the control states in real time, so the closed-loop dynamics evolve under dynamic feedback as:

$$\dot{z} = \begin{bmatrix} \dot{x} \\ \dot{t}^\theta \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} f(x,\, \phi(t - t_0^\theta, \theta_1)) \\ \mathrm{Proj}\left\{-k_t\, \alpha(t,z)\, \Gamma_t\, \nabla_{t^\theta} J^T,\ \Xi(t)\right\} \\ \mathrm{Proj}\left\{-k_\theta \Gamma_\theta\, \nabla_\theta J^T,\ \Theta^N\right\} \end{bmatrix} \qquad \text{while } t \le t_1^\theta \tag{12a}$$

$$\alpha(t,z) \triangleq \begin{bmatrix} 1 & 0 & 0 \\ 0 & \mathrm{sat}\left(\frac{t_1^\theta - t}{\epsilon},\ [0,\, 1]\right) & 0 \\ 0 & 0 & I \end{bmatrix}_{n_\theta \times n_\theta} \tag{12b}$$

$$\Xi(t) = \left\{ t^\theta \in \mathbb{R}^{N+1} \;\middle|\; (\pi_i(t, t^\theta) \ge 0,\ i = 1, \dots N) \text{ and } \left(\textstyle\sum_{i=1,\dots N} \pi_i \le T\right) \right\} \tag{12c}$$

where $\epsilon > 0$ is a small constant, and π represents the coordinate transformation

$$\pi_i(t, t^\theta) = \left\{ \begin{matrix} t - t_0^\theta & i = 0 \\ t_i^\theta - t_{i-1}^\theta & i = 1, \dots N \end{matrix} \right\} \in \mathbb{R}_{\ge 0}^{N+1} \tag{13}$$

The function α serves to restrict the adaptation of t_1^θ such that the intersection $t = t_1^\theta$ is transversal, resulting in deterministic closed-loop behaviour. Although (12a) appears nonautonomous, all time-dependence disappears under transformation (13).

The gradient terms $\nabla_\theta J$ and $\nabla_{t^\theta} J$ in (12a) represent sensitivities of (11a), for which differential sensitivity expressions must be solved. Fortunately, several efficient algorithms (for example [12]) exist for simultaneous solution of ODE's with their parametric sensitivity equations, which can additionally be efficiently decomposed by the intervals of t^θ. The matrices $\Gamma_{t^\theta} > 0$ and $\Gamma_\theta > 0$ define the type of descent-based optimization used. While constant matrices generating (scaled-) steepest-descent trajectories are the simplest choice, higher order definitions such as Gauss-Newton or full order Newton (appropriately convexified) could be used.

The operator in (12a) of the form $\dot{s} = \text{Proj}(\nu, \mathbb{S})$ denotes a (Lipschitz) parameter projection like those defined in [13], where the component of ν orthogonal to $\partial\mathbb{S}$ is removed as s approaches $\partial\mathbb{S}$. This results in the properties 1) $s(t_0) \in \mathbb{S} \implies s \in \mathbb{S}$ for all $t \geq t_0$, and 2) $\nabla_s J \cdot \text{Proj}(-k\Gamma \nabla_s J^T, \mathbb{S}) \leq 0$. For brevity, the reader is referred to [13] and reference therein for details on the design of such an operator. We note that applying this operator to θ serves simply to ensure that $\theta(t) \in \Theta$, not to enforce $u(t) \in \mathbb{U}$. Enforcing $u(t) \in \mathbb{U}$ by selection of Θ (rather than using B_u) is possible in special cases when $\mathbb{U}$ is a convex set, and ϕ is convex in both arguments.

Lemma 2. *Over any interval of existence $t \in [t_0, t_1]$ of the solution to* (12a) *starting from $(t^\theta, \theta)(t_0) \in \Phi(t_0, x(t_0), \mathcal{P})$, the closed-loop flows satisfy 1) $\frac{dJ}{dt} = \nabla_t J + \nabla_z J\, \dot{z} < 0$ when $x \notin \Sigma_{\mathbb{X}}$, and 2) $(t^\theta, \theta)(t) \in \Phi(t, x(t), \mathcal{P})$.*

Step 3: Parameter re-initialization

When the equality $t = t_1^\theta$ occurs, the n_θ parameters assigned to the first interval are no longer useful as degrees of freedom for minimizing (11a); instead, it is more beneficial to reassign these degrees of freedom to a new interval at the tail of the prediction horizon. This takes the form of the discrete jump mapping

$$z^+ = \left\{ \begin{array}{rcl} x^+ & = & x \\ (t_i^\theta)^+ & = & \begin{cases} t_{i+1}^\theta & i = 0 \ldots (N-1) \\ t_N^\theta + \delta(x^p(t_N^\theta)) & i = N \end{cases} \\ (\theta_i)^+ & = & \begin{cases} \theta_{i+1} & i = 1 \ldots (N-1) \\ \kappa(x^p(t_N^\theta)) & i = N \end{cases} \end{array} \right\} \quad \text{if } t \geq t_1^\theta \qquad (14)$$

where the feedbacks $\kappa(\cdot)$ and $\delta(\cdot)$ are used to initialize the parameters for the new interval. Following execution of (14), the algorithm repeats back to Step 2.

Lemma 3. *The jump mapping in* (14) *is such that 1) $J(t, z^+) - J(t, z) \leq 0$, and 2) $(t^\theta, \theta)^+ \in \Phi(t, x, \mathcal{P})$*

Remark 1. *The manner in which the horizon t_N^θ recedes (i.e. by* (14)*) differs from many other realtime approaches, in which the horizons recede continuously.*

While it may seem more natural to enforce a continuous recede $\dot{t}^\theta$, this generally violates the dynamic programming principle, in which case stability can only be claimed if one assumes either 1) $N = 1$ and (1) *is globally prestabilized [3], 2) ϕ contains a very large number of bases, or 3) $\left|t_i^\theta - t_{i-1}^\theta\right|$ is very small [2]. In contrast, we require none of these assumptions. (While Lemma 1 implies "sufficiently large N", the requirements for feasible initialization are* significantly *less conservative than for preservation of stability as in [2]).*

5.3 Hybrid Trajectories and Stability

The closed-loop behaviour resulting from the algorithm in Section 5.2 is that of a dynamic control law whose controller states exhibit discontinuous jumps. As such, neither classical notions of a "solution" nor those from the sampled-data literature apply to the closed-loop dynamics. Instead, a notion of solution developed for hybrid systems in [14] (and other recent work by the same authors) can be applied, in which trajectories are described as evolving over the "hybrid time" domain - i.e. a subset of $[0, \infty) \times \mathbb{N}_0$ given as a union of intervals of the form $[t_j, t_{j+1}] \times \{j\}$. In this context, the continuous dynamics (12a) have the form $\dot{z}_\pi = F(z_\pi)$ on the *flow domain*

$$S_F \triangleq \{\, z_\pi \ : \ \pi_0 \le \pi_1 \text{ and } (t^\theta,\, \theta) \in \Phi(t, x, \mathcal{P}) \,\}, \quad t^\theta \equiv t^\theta(t, \pi), \quad t \text{ arbitrary} \tag{15}$$

where z_π denotes a coordinate change of z with t^θ transformed by (13). Likewise, (14) has the form $z_\pi^+ = H(z_\pi)$ on the *jump domain*

$$S_H \triangleq \{\, z_\pi \ : \ \pi_0 \ge \pi_1 \text{ and } (t^\theta,\, \theta) \in \Phi(t, x, \mathcal{P}) \,\}, \quad t^\theta \equiv t^\theta(t, \pi), \quad t \text{ arbitrary} \tag{16}$$

Lemmas 2 and 3 guarantee the invariance of $S_F \cup S_H$, the domain on which either a flow or jump is always defined. Although S_F and S_H intersect, uniqueness of solutions results from the fact that $F(z_\pi)$ points out of S_F on $S_F \cap S_H$ [15, Thm III.1]. In the language of [15], the resulting closed-loop system is a nonblocking, deterministic hybrid automaton which accepts a unique, infinite execution. Using this notion of solution, the behaviour can be summarized as follows:

Theorem 1. *Let an input parameterization $\mathcal{P}$ be selected to satisfy Assumption 1, and assume that a corresponding local stabilizer $\kappa(x)$, $\delta(x)$ and penalty function $W(x)$ are found which satisfy Assumption 2 on the $\mathcal{X}_f$. Furthermore, let the constraints in* (2c) *be enforced by barrier functions satisfying Criteria 1 and 2. Then, using the dynamic feedback algorithm detailed in Section 5.2, the target set Σ is feasibly, asymptotically stabilized with domain of attraction $\mathbb{X}_{doa}(N)$ containing $\mathcal{X}_f$. Furthermore, $\exists N^* \ge 1$ such that $\mathbb{X}_{doa}(N) \equiv \mathbb{X}^0$ for $N \ge N^*$.*

6 Simulation Example

To illustrate implementation of our approach, we consider regulation of the stirred tank reactor from [16], with exothermic reaction $A \longrightarrow B$ resulting in dynamics

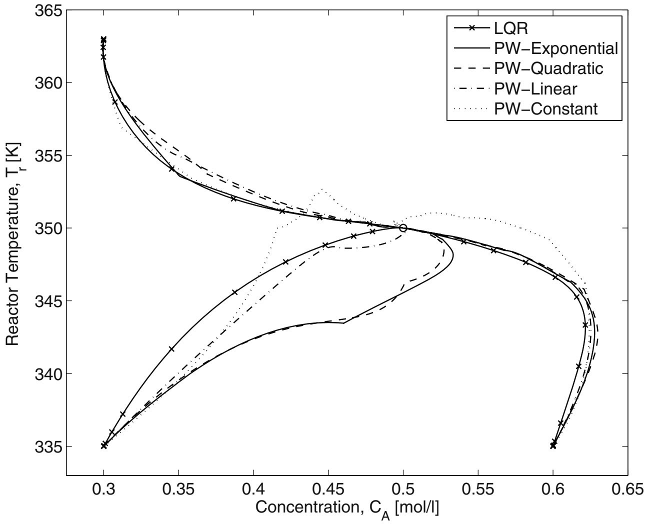


Fig. 1. Closed-loop state profiles from three different x_0, using various ϕ

$$\dot{C}_A = \frac{v}{V}\left(C_{Ain} - C_A\right) - k_0 \exp\left(\frac{-E}{R\,T_r}\right) C_A$$

$$\dot{T}_r = \frac{v}{V}\left(T_{in} - T_r\right) - \frac{\Delta H}{\rho\, c_p}\, k_0 \exp\left(\frac{-E}{R\,T_r}\right) C_A + \frac{UA}{\rho\, c_p\, V}\left(T_c - T_r\right)$$

Constants are taken from [16]: v=100 ℓ/min, V=100 ℓ, $\rho\, c_p$=239 J/ℓ K, E/R = 8750 K, k_0=7.2×10^{10} min^{-1}, UA=5×10^4 J/min·K, ΔH=−5×10^4 J/mol, C_{Ain}=1 mol/ℓ, T_{in}=350 K. The target is to regulate the unstable equilibrium C_A^{eq}=0.5 mol/ℓ, T_r^{eq}=350 K, T_c^{eq}=300 K, using the coolant temperature T_c as the input, subject to the constraints $0 \leq C_A \leq 1$, $280 \leq T_r \leq 370$ and $280 \leq T_c \leq 370$.

Using the cost function $L(x,u) = x'Qx + u'Qu$, with $x = [C_A{-}C_A^{eq},\ T_r - T_r^{eq}]'$, $u = (T_c - T_c^{eq})$, $Q = diag(2,\ 1/350)$, $R = 1/300$, the linearized local controller $k_f(x) = [109.1,\ 3.3242]\, x$ and cost $W(x) = x'Px$, $P = [17.53,\ 0.3475;\ 0.3475,\ 0.0106]$, were chosen. Four different choices of the basis $\phi(\tau, \theta_i)$ were tested,

$$\phi_C = \theta_{i1} \quad \phi_L = \theta_{i1} + \theta_{i2}\tau \quad \phi_Q = \theta_{i1} + \theta_{i2}\tau + \theta_{i3}\tau^2 \quad \phi_E = \theta_{i1}\exp\left(-\theta_{i2}\tau\right)$$

with N chosen (intentionally small) such that the total size of θ remained similar (N_C=8, N_L=N_E=4, N_Q=3). In each case, the gains k_θ=0.1 and k_t=0.5 were used in the update laws, with $\Gamma_t \equiv I$ and Γ_θ chosen as a diagonally scaled identity matrix (i.e. scaled steepest-descent updates). The feedbacks $\kappa(x)$ were derived by analytically solving (7), while $\delta(x)$ was chosen using forward simulation as described in Section 4.1. In all cases, initial conditions for t^θ and θ were chosen to approximate the trajectory $T_c(t)$, $t \in [0, 1.5]$, resulting under LQR feedback $u = k_f(x)$.

Three different initial conditions were tested, and the closed-loop state profile for each parameterization are shown in Figures 1 and 2, with corresponding

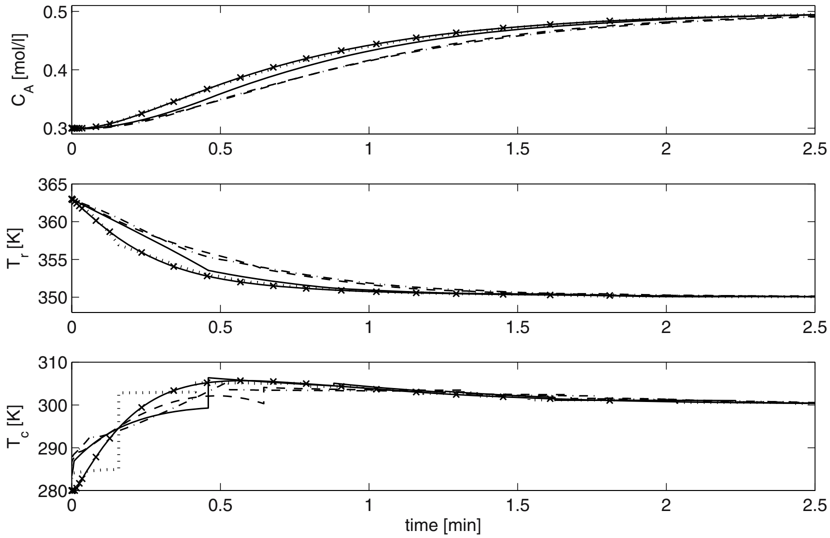


Fig. 2. Closed-loop trajectories from $(C_A, T) = (0.3, 363)$. Symbols same as Fig. 1.

Table 1. Actual closed-loop cost to practically (at t=10min) stabilize to setpoint

$(C_A, T_r)_0$	LQR	ϕ_C	ϕ_L	ϕ_Q	ϕ_E
(0.3, 363)	0.285	0.310	0.281	0.278	0.279
(0.3, 335)	1.74	1.80	1.55	1.42	1.41
(0.6, 335)	0.596	0.723	0.570	0.567	0.558

closed-loop costs reported in Table 1. Using higher-order parameterizations such as ϕ_E and ϕ_Q over coarse time-intervals resulted in lower cost than ϕ_C (which used smaller intervals), thus making better use of approximately the same number of optimization parameters. Although the equilibrium is open-loop unstable, large interval-lengths are not problematic since (12a) provides a continuous-time state-feedback for $k_\theta > 0$.

7 Conclusions

In this work, a framework has been proposed for continuous-time NMPC in which the dynamics associated with the nonlinear program are allowed to evolve in the same timescale as the process dynamics, without compromising closed-loop stability. The unique manner in which the prediction horizon recedes accommodates the use of efficient basis functions capable of parameterizing the input trajectory over large intervals using relatively few parameters. Adapting the time support of the parameterization, if desired, helps to maximize the efficiency of the parameterization. By allowing for stabilization to a general target set, a broad class of control problems can be addressed within the given framework.

References

[1] Diehl, M., Bock, H., and Schloder, J., "A real-time iteration scheme for nonlinear optimization in optimal feedback control", *SIAM Journal of Control and Optimization*, **43(5)**, 1714–1736 (2005).

[2] Ohtsuka, T., "A continuation/GMRES method for fast computation of nonlinear receding horizon control", *Automatica*, **4(40)**, 563–574 (2004).

[3] Cannon, M. and Kouvaritakis, B., "Continuous-time predictive control of constrained non-linear systems", in: F. Allgower and A. Zheng (eds.), *Nonlinear Model Predictive Control: Assessment and Future Directions for Research*, Birkhauser-Verlag (2000).

[4] Findeisen, R. and Allgöwer, F., "Stabilization using sampled-data open-loop feedback – a nonlinear model predictive control perspective", in: *Proc. IFAC Symposium on Nonlinear Control Systems*, 735–740 (2004).

[5] DeHaan, D. and Guay, M., "A real-time framework for model predictive control of continuous-time nonlinear systems", in: *Proc. IEEE Conf. on Decision and Control* (2005), To Appear.

[6] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. M., "Constrained model predictive control: Stability and optimality", *Automatica*, **36**, 789–814 (2000).

[7] Clarke, F., Ledyaev, Y., Sontag, E., and Subbotin, A., "Asymptotic controllability implies feedback stabilization", *IEEE Trans. Automat. Contr.*, **42(10)**, 1394–1407 (1997).

[8] Nešić, D. and Teel, A., "A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models", *IEEE Trans. Automat. Contr.*, **49(7)**, 1103–1122 (2004).

[9] Wills, A. and Heath, W., "Barrier function based model predictive control", *Automatica*, **40**, 1415–1422 (2004).

[10] Grimm, G., Messina, M., Tuna, S., and Teel, A., "Examples when model predictive control is non-robust", *Automatica*, **40(10)**, 1729–1738 (2004).

[11] Nesterov, Y. and Nemirovskii, A., *Interior-Point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia (1994).

[12] Leis, J. and Kramer, M., "The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations", *ACM Transactions on Mathematical Sofware*, **14(1)**, 45–60 (1988).

[13] Krstic, M., Kanellakopoulos, I., and Kokotovic, P., *Nonlinear and Adaptive Control Design*, Wiley and Sons, New York (1995).

[14] Goebel, R., Hespanha, J., Teel, A., Cia, C., and Sanfelice, R., "Hybrid systems: generalized solutions and robust stability", in: *Proc. IFAC Symposium on Nonlinear Control Systems*, 1–12 (2004).

[15] Lygeros, J., Johansson, K., Simić, S., Zhang, J., and Sastry, S., "Dynamical properties of hybrid automata", *IEEE Trans. Automat. Contr.*, **48(1)**, 2–17 (2003).

[16] Magni, L., De Nicolao, G., Magnani, L., and Scattolini, R., "A stabilizing model-based predictive control for nonlinear systems", *Automatica*, **37(9)**, 1351–1362 (2001).

A Proof of Lemma 2

It can be shown from (11a) that $\nabla_t J = -L^a(x^p, u_\phi) - \langle \nabla_x J,\, f(x^p, u_\phi)\rangle$. From (12a),

$$\begin{aligned}\frac{dJ}{dt} &= \nabla_t J + \nabla_z J\,\dot{z} \\ &= -L^a(x^p, u_\phi) - \big\langle \nabla_{t^\theta} J,\ \mathrm{Proj}\{k_t\,\alpha\,\Gamma_t\,\nabla_{t^\theta} J^T,\ \Xi\}\big\rangle - \big\langle \nabla_\theta J,\ \mathrm{Proj}\{k_\theta \Gamma_\theta\,\nabla_\theta J^T,\ \Theta^N\}\big\rangle \\ &\le -\gamma_L(\|x^p, u_\phi\|_\Sigma)\end{aligned}$$

The conditions of the lemma guarantee that $J(t_0, z_0)$ is bounded (although not uniformly), and the above ensures that $J(t, z) \le J(t_0, z_0)$, for all $t \in [t_0, t_1]$. Since all dynamics in (12a) are locally Lipschitz on the set $\mathcal{Z} = \{\ z\ :\ (t^\theta,\ \theta) \in \Phi(t, x(t), \mathcal{P})\ \}$, continuity of the solution implies that the states can only exit $\mathcal{Z}$ by either 1) reaching the boundary $\mathcal{A} = cl\{\mathcal{Z}\} \setminus \mathcal{Z}$ (i.e. the set where $x \in \partial\mathbb{X}$, $u_\phi \in \partial\mathbb{U}$, or $x^p(t_N^\theta) \in \partial\mathcal{X}_f$), or 2) passing through the boundary $\mathcal{B} = \mathcal{Z} \setminus \mathring{\mathcal{Z}}$. The first case is impossible given the decreasing nature of J and $\lim_{z \to \mathcal{A}} J(t, z) = \infty$, while the second case is prevented by the parameter projection in (12a).

B Proof of Lemma 3

The first claim follows from

$$\begin{aligned}J(t, z^+) - J(t, z) &= \int_{t_N^\theta}^{t_N^{\theta+}} L^a(x^p(\tau, t, z^+, \phi),\ u_\phi(\tau, z^+, \phi))\,d\tau + W^a(x_f^{p+}) - W^a(x_f^p) \\ &= \int_0^{\delta(x_f^p)} L(x_\kappa(\tau),\ \phi(\tau, \kappa(x_f^p))) + \mu\Big(B_x(x_\kappa(\tau)) + B_u(\phi(\tau, \kappa(x_f^p)))\Big)\,d\tau \\ &\qquad + W(x_f^{p+}) - W(x_f^p) + \mu\Big(B_{x_f}(x_f^{p+}) - B_{x_f}(x_f^p)\Big) \\ &\le 0 \qquad \text{(by (4) and (9))}\end{aligned}$$

where $x_f^p \triangleq x^p(t_N^\theta, t, z, \phi)$, $x_f^{p+} \triangleq x^p(t_N^{\theta+}, t, z^+, \phi)$, and $x_\kappa(\cdot)$ is the solution to $\dot{x}_\kappa = f(x_\kappa, \phi(t, \kappa(x_f^p)))$, $x_\kappa(0) = x_f^p$. The second claim follows by the properties of $\kappa(x)$ guaranteed by Assumption 2, since the portion of the $x^p(\tau)$ and $u_\phi(\tau)$ trajectories defined on $\tau \in (t,\, t_N^\theta]$ are unaffected by (14).

C Proof of Theorem 1

Using the cost $J(z_\pi)$ as an energy function (where $J(z_\pi) \equiv J(s, x, t^\theta - s, \theta)$ from (11a), with s arbitrary), the result follows from the Invariance principle in [15, Thm IV.1]. The conditions of [15, Thm IV.1] are guaranteed by Lemmas 2, 3, and the boundedness of the sets $\mathbb{X}$, $\mathbb{U}$, Θ and Ξ (which ensures that trajectories remain in a compact subset of $S_F \cup S_H$). Thus, z_π asymptotically converge to M, the largest invariant subset of $\{z_\pi \mid \dot{J} = 0$ under (12a)$\} \cup \{\ z_\pi\ :\ J^+ - J = 0$ under (14) $\}$. Since H maps into the interior of S_F (strictly away from S_H), zeno solutions are not possible. This implies $M \subset \{z_\pi \mid \dot{J} = 0\}$, and thus from the proof of Lemma 2 it follows that $M = \{\ z_\pi\ :\ (x, u_\phi) \in \Sigma\ \}$. Feasibility holds from Lemmas 2 and 3, while the last claim follows from Lemma 1 and the compactness of $\mathbb{X}^0$.

A Two-Time-Scale Control Scheme for Fast Unconstrained Systems

Sebastien Gros, Davide Buccieri, Philippe Mullhaupt, and Dominique Bonvin

Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland

Summary. Model predictive control (MPC) is a very effective approach to control nonlinear systems, especially when the systems are high dimensional and/or constrained. MPC formulates the problem of input trajectory generation as an optimization problem. However, due to model mismatch and disturbances, frequent re-calculation of the trajectories is typically called for. This paper proposes a two-time-scale control scheme that uses less frequent repeated trajectory generation in a slow loop and time-varying linear feedback in a faster loop. Since the fast loop reduces considerably the effect of uncertainty, trajectory generation can be done much less frequently. The problem of trajectory generation can be treated using either optimization-based MPC or flatness-based system inversion. As proposed, the method cannot handle hard constraints. Both MPC and the two-time-scale control scheme are tested via the simulation of a flying robotic structure. It is seen that the MPC scheme is too slow to be considered for real-time implementation on a fast system. In contrast, the two-time-scale control scheme is fast, effective and robust.

1 Introduction

Model predictive control (MPC) is an effective approach for tackling problems with nonlinear dynamics and constraints, especially when analytical computation of the control law is difficult [4, 14, 17]. MPC involves re-calculating at every sampling instant the inputs that minimize a criterion defined over a horizon window in the future, taking into account the current state of the system.

A crucial point in MPC is the extensive use of the dynamic model. Since the model is not always accurate, the predicted state evolution may differ from the actual plant evolution, which requires frequent re-calculation of the inputs. Solutions to this problem are proposed in the literature. One possibility is to cast the problem into a robust framework, where optimization is performed by taking the uncertainty into account explicitly. Robust predictive control computes input trajectories that represent a compromise solution for the range of uncertainty considered [2, 11, 12]. Such a methodology is widely used in the process industry, where system dynamics are sufficiently slow to permit its implementation. However, due to the complexity of the calculations involved in robust predictive control, its applicability to fast dynamics is rather limited. Several promising

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 551–563, 2007.
springerlink.com

approaches have been proposed to decrease the computational time required by nonlinear MPC schemes. These approaches rely mostly on cleverly chosen coarse parametrizations [6, 13]. Though very effective, these approaches require that the input trajectories be sufficiently simple to tolerate low-dimensional representations.

Another solution consists in tracking the system trajectories with a fast feedback loop. If the local dynamics are nearly time invariant, linear control theory provides effective tools to design this feedback loop. However, for systems having strongly-varying local dynamics, there is no systematic way of designing such a feedback law [1, 15, 16]. This trajectory-tracking problem can be tackled by the neighboring-extremal theory whenever the inputs and states are not constrained. For small deviations from the optimal solution, a linear approximation of the system and a quadratic approximation of the cost are quite reasonable. In such a case, a neighboring-extremal (NE) controller provides a closed-form solution to the optimization problem. Hence, the optimal inputs can be approximated using state feedback, i.e. without explicit numerical re-optimization.

This paper presents two approaches to control a simulated robotic flying structure known as VTOL (Vertical Take-Off and Landing). The structure has 4 inputs and 16 states. It is a fast and strongly nonlinear system. The control schemes are computed based on a simplified model of the system, while the simulations use the original model. The simplified VTOL model is flat [7, 8].

The first control approach is based on MPC. The use of repeated optimization of a cost function describing the control problem provides a control sequence that supposedly rejects uncertainties in the system.

The second control approach combines a flatness-based feedforward trajectory generation in a slow loop and a linear time-varying NE-controller in a faster loop. The slow loop generates the reference input and state trajectories, while the fast loop ensures good tracking of the state trajectories. This control scheme is sufficiently effective to make re-generation of the reference trajectories unnecessary.

The paper is organized as follows. Section 2 briefly revisits optimization-based MPC, system inversion for flat systems and NE-control. The proposed two-time-scale control scheme is detailed in Section 3. Section 4 presents the simulated operation of a VTOL structure. Finally, conclusions are provided in Section 5.

2 Preliminaries

2.1 Nonlinear MPC

Consider the nonlinear dynamic process:

$$\dot{x} = F(x, u), \qquad x(0) = x_0 \tag{1}$$

where the state x and the input u are vectors of dimension n and m, respectively. x_0 represents the initial conditions, and F the process dynamics.

Predictive control of (1) is based on repeatedly solving the following optimization problem:

$$\min_{u[t_k,t_k+T_c)} J = \Phi(x(t_k+T_p)) + \int_{t_k}^{t_k+T_p} L(x(\tau),u(\tau))d\tau \tag{2}$$

$$s.t. \quad \dot{x} = F(x,u), \qquad x(t_k) = x_m(t_k) \tag{3}$$

$$\mathcal{S}(x,u) \le 0 \quad \mathcal{T}(x(t+T_p)) = 0 \tag{4}$$

where Φ is an arbitrary scalar function of the states and L an arbitrary scalar function of the states and inputs. $x_m(t)$ represents the measured or estimated value of $x(t)$. $\mathcal{S}$ is a vector function of the states and inputs that represents inequality constraints and $\mathcal{T}$ is a vector function of the final states that represents equality constraints. The prediction horizon is noted T_p and the control horizon is noted T_c. In the following, $T_c = T_p = T$ will be used. The solution to problem (2)-(4) will be noted (x^*, u^*). A lower bound for the re-optimization interval $\delta = t_{k+1} - t_k$ is determined by the performance of the available optimization tools.

2.2 System Inversion for Flat Systems

If the system (1) is flat in the sense of [7, 8], with $y = h(x)$ being named the flat output, then, for a given sufficiently smooth trajectory $y(t)$ and a finite number σ of its derivatives, it is possible to compute the corresponding inputs and states:

$$u = u(y, \dot{y}, ..., y^{(\sigma)}) \qquad x = x(y, \dot{y}, ..., y^{(\sigma)}). \tag{5}$$

$u(y, \dot{y}, ..., y^{(\sigma)})$ is a nonlinear function that inverts the system.

2.3 Neighboring-Extremal Control

Upon including the dynamic constraints of the optimization problem in the cost function, the augmented cost function, $\bar{J}$, reads:

$$\bar{J} = \Phi(x(t_k+T)) + \int_{t_k}^{t_k+T} \left(H - \lambda^T \dot{x}\right) dt \tag{6}$$

where $H = L + \lambda^T F(x,u)$, and $\lambda(t)$ is the n-dimensional vector of adjoint states or Lagrange multipliers for the system equations. The first-order variation of $\bar{J}$ is zero at the optimum. For a variation $\Delta x(t) = x(t) - x^\star(t)$ of the states, minimizing the second-order variation of $\bar{J}$, $\Delta^2 \bar{J}$, with respect to $\Delta u(t) = u(t) - u^*(t)$ represents a time-varying Linear Quadratic Regulator (LQR) problem, for which a closed-form solution is available [3]:

$$\Delta u(t) = -K(t)\Delta x(t) \tag{7}$$

$$K = H_{uu}^{-1}\left(H_{ux} + F_u^T S\right) \tag{8}$$

$$\dot{S} = -H_{xx} + S(F_u H_{uu}^{-1} H_{ux} - F_x) \tag{9}$$
$$+(H_{xu} H_{uu}^{-1} F_u^T - F_x^T)S + S F_u H_{uu}^{-1} F_u^T S + H_{xu} H_{uu}^{-1} H_{ux}$$
$$S(t_k + T) = \Phi(x(t_k + T)) \tag{10}$$

The S matrix is computed backward in time. This computation can be numerically demanding. Controller (7)-(10) is termed the NE-controller. Note that it does not take constraints into account.

3 Two-Time-Scale Control Scheme

The repeated solution of (2)-(4) provides feedback to the system. Yet, since the time necessary to perform the optimization can be rather large compared to the system dynamics, the feedback provided by the re-optimization tends to be too slow to guarantee performance and robustness. Hence, it is proposed to add a fast feedback loop in the form of a NE-controller. The resulting control scheme is displayed in Figure 1. The NE-controller operates in the fast loop at a sampling frequency appropriate for the system, while the reference trajectories are generated in the slow loop at a frequency permitting their computation. Note that, if the time-scale separation between the two loops is sufficient, $u_{ref}(t)$ can be considered as a feedforward term for the fast loop.

3.1 Trajectory Generation

The generation of the reference trajectories $u_{ref}(t)$ and $x_{ref}(t)$ can be computed via optimization (e.g. nonlinear MPC) or direct system inversion (as is possible for example for flat systems [10]).

3.2 Tracking NE-Controller

The NE-controller (7)-(10) can be numerically difficult to compute. Its computation is simplified if the optimization problem considers trajectory tracking. Indeed, for tracking the trajectories $u_{ref}(t)$ and $x_{ref}(t)$, Φ and L can be chosen as:

$$\Phi = \frac{1}{2}(x - x_{ref})^T P (x - x_{ref}) \tag{11}$$
$$L = \frac{1}{2}(x - x_{ref})^T Q (x - x_{ref}) + \frac{1}{2}(u - u_{ref})^T R (u - u_{ref}) \tag{12}$$

for which the solution to problem (2)-(3) is $u^*(t) = u_{ref}(t)$ and $x^*(t) = x_{ref}(t)$. Furthermore, the adjoints read:

$$\dot{\lambda} = -H_x^T = -F_x^T \lambda - Q(x - x_{ref}) \tag{13}$$
$$\lambda(t_k + T) = \Phi_x(t_k + T) = 0 \tag{14}$$

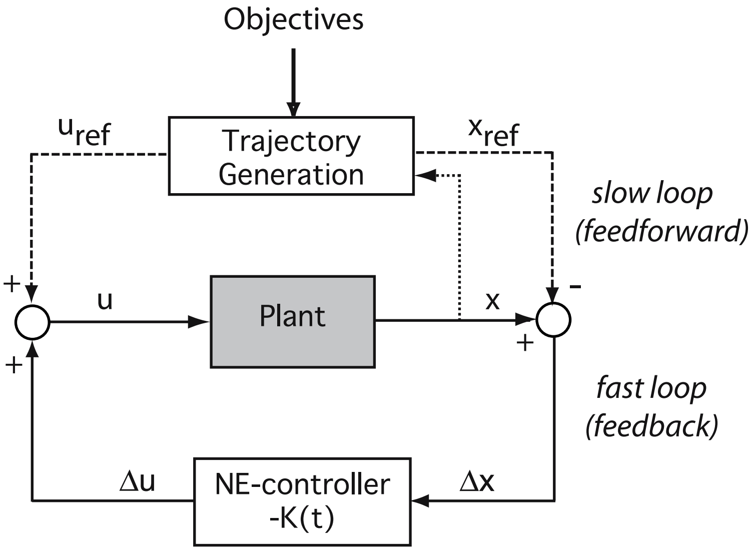


Fig. 1. Scheme combining trajectory generation (via optimization or system inversion) and NE-control

i.e. they are zero along the whole trajectory. Hence, the NE-controller reduces to:

$$\Delta u(t) = -K(t)\Delta x(t) \tag{15}$$

$$K = R^{-1}F_u^T S \tag{16}$$

$$\dot{S} = -Q - SF_x - F_x^T S + SF_u R^{-1} F_u^T S \tag{17}$$

$$S(t_k + T) = P \tag{18}$$

which can be viewed as a time-varying LQR. Note that, if the local system dynamics are nearly constant, the NE-controller is well approximated by a LQR with a constant gain matrix K. In contrast, if the system is strongly time-varying, it is necessary to compute the time-varying NE-controller (15)-(18).

4 Application to a VTOL Structure

4.1 System Dynamics

The simulated example is a VTOL structure.

The structure is made of four propellers mounted on the four ends of an orthogonal cross. Each propeller is motorized independently. The propeller rotational velocities are opposed as follows (when top viewed, counted counterclockwise): propellers 1 and 3 rotate counterclockwise, while propellers 2 and

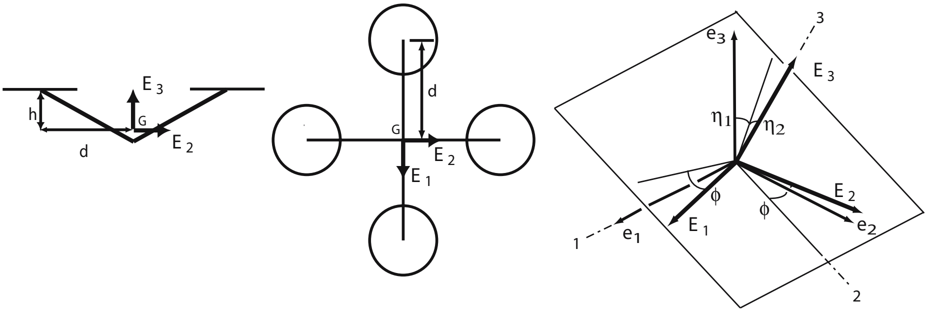


Fig. 2. (E_1, E_2, E_3) is a frame attached to the VTOL at its center of mass G. h is the vertical distance between center of mass and propeller center. d is the horizontal distance between center of mass and propeller axis. The transformation $(e_1, e_2, e_3) \rightarrow (E_1, E_2, E_3)$ is defined by (i) rotation of angle η_1 around axis 1, (ii) rotation of angle η_2 around axis 2, (iii) rotation of angle ϕ around axis 3.

4 rotate clockwise. The angle of attack (AoA) of the blades and the positions of the propellers are fixed relative to the structure. The VTOL is controlled by means of the four motor torques. The states of the system are:

$$X = \begin{bmatrix} x \; y \; z \; \eta_1 \; \eta_2 \; \phi \; \dot{x} \; \dot{y} \; \dot{z} \; \dot{\eta}_1 \; \dot{\eta}_2 \; \dot{\phi} \; \dot{\rho}_1 \; \dot{\rho}_2 \; \dot{\rho}_3 \; \dot{\rho}_4 \end{bmatrix} \tag{19}$$

Variables $\begin{bmatrix} x \; y \; z \end{bmatrix}$ give the position of the center of gravity G in $[m]$ within the laboratory referential (e_1, e_2, e_3). Variables $\begin{bmatrix} \eta_1 \; \eta_2 \; \phi \end{bmatrix}$ give the angular attitude of the structure in $[rad]$, with the transformation from the laboratory referential to the VTOL referential, $(e_1, e_2, e_3) \rightarrow (E_1, E_2, E_3)$, being described by the matrix $\Phi(\eta_1, \eta_2, \phi) = R_{e_3}(\phi) R_{e_2}(\eta_2) R_{e_1}(\eta_1)$, where $R_e(\alpha)$ is a rotation of angle α around the basis vector e.

Variable $\dot{\rho}_k$ is the speed of the propeller k in $[rad/s]$. The model of the VTOL can be computed by means of analytical mechanics. The aerodynamical forces and torques generated by the propellers are modeled using the standard squared velocity law. The resulting model is rather complicated and will not be explicited here. The reader is referred to [9] for details. The model is nonlinear, and its local dynamics are strongly time varying.

A simplified model can be computed by removing certain non-linearities, which is well justified in practice. Introducing the notations

$$v_1 = \sum_{k=1}^{4} \dot{\rho}_k^2 \qquad v_2 = \sum_{k=1}^{4} (-1)^k \dot{\rho}_k^2 \qquad v_3 = \dot{\rho}_1^2 - \dot{\rho}_3^2$$

$$v_4 = \dot{\rho}_2^2 - \dot{\rho}_4^2 \qquad v_5 = \sum_{k=1}^{4} \dot{\rho}_k$$

the simplified model can be written as:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = C_{xyz} \begin{bmatrix} sin(\eta_2) \\ -cos(\eta_2)sin(\eta_1) \\ cos(\eta_2)cos(\eta_1) \end{bmatrix} v_1 - \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{20}$$

$$\ddot{\phi} = C_\phi v_2 \tag{21}$$

$$\begin{aligned} \ddot{\eta}_1 = & \frac{1}{C_{\eta_1}^1 cos(\eta_2)^2 + C_{\eta_1}^2}(C_d sin(\eta_2)v_2 + C_s\ d\ cos(\eta_2)(sin(\phi)v_3 + cos(\phi)v_4) \\ & - I_A^M cos(\eta_2)\dot{\eta}_2 v_5) \qquad (22) \\ \ddot{\eta}_2 = & C_{\eta_2}(-cos(\phi)v_3 + sin(\phi)v_4 + I_M^A cos(\eta_2)\dot{\eta}_1 v_5) \qquad (23) \\ \dot{v}_k = & u_k \qquad k = 1, ..., 4 \qquad (24) \end{aligned}$$

with the constants

$$\begin{aligned} C_{xyz} &= \frac{C_s}{M_5 + 4m} \qquad C_\phi = \frac{C_d}{4I_M^A + 4md^2 + I_M^5} \\ C_{\eta_1}^1 &= -I_M^5 + I_S^5 - 4I_M^A + 4I_S^A - 2md^2 + 4mh^2 \qquad C_{\eta_1}^2 = I_5^M + 4I_M^A + 4md^2 \\ C_{\eta_2} &= \frac{C_s d}{4I_S^A + 2md^2 + 4mh^2 + I_S^5} \end{aligned}$$

The inputs u_k, $k = 1...4$, do not represent the physical inputs (the motor torques M_k), but are related to them by invertible algebraic relationships. The numerical values of the parameters used in the simulations are given in Table 1. Parameters I_M^5, I_S^5, I_M^A, I_S^A are the inertias of the main body and the propellers, respectively. Parameters C_s and C_d are the aerodynamical parameters of the propellers. Parameters M_5, m are the masses of the main body and propellers, respectively.

The simplified model is flat. The flat outputs are: $Y = \begin{bmatrix} x\ y\ z\ \phi \end{bmatrix}$.

Table 1. Model parameters

C_s	3.64×10^{-6}	Ns^2	d	0.3	m
C_d	1.26×10^{-6}	Nms^2	I_M^5	181×10^{-4}	Nms^2
M_5	0.5	kg	I_S^5	96×10^{-4}	Nms^2
m	2.5×10^{-2}	kg	I_M^A	6.26×10^{-6}	Nms^2
h	0.03	m	I_S^A	1.25×10^{-6}	Nms^2

4.2 Control Problem

The control problem is of the tracking type: the VTOL structure must be driven smoothly from some initial configuration to another predefined configuration. A translation from the position ($x = 0\,[m], y = 0\,[m], z = 0\,[m], \phi = 0\,[rad]$) to the position ($x = 1\,[m], y = 1\,[m], z = 1\,[m], \phi = 2\pi\,[rad]$) will be considered. The speeds and accelerations are zero initially. The control problem is unconstrained. The controllers are computed using the simplified model, while the simulations are done with the original model. Perturbations are introduced in the aerodynamical parameters C_s and C_d to represent the uncertainty resulting from self-induced turbulences and surface effects: C_s^1, C_d^1, C_s^2, C_d^2 are perturbed +50 percent and C_s^3, C_d^3, C_s^4, C_d^4 −50 percent, C_d^k and C_s^k being the aerodynamical coefficients of propeller k.

4.3 MPC

The choice of cost function for the MPC scheme is:

$$J = \frac{1}{2} \int_{t_k}^{t_k+T} [(\bar{x} - \bar{x}_{sp})^T Q (\bar{x} - \bar{x}_{sp}) + \bar{u}^T R \bar{u}] dt$$

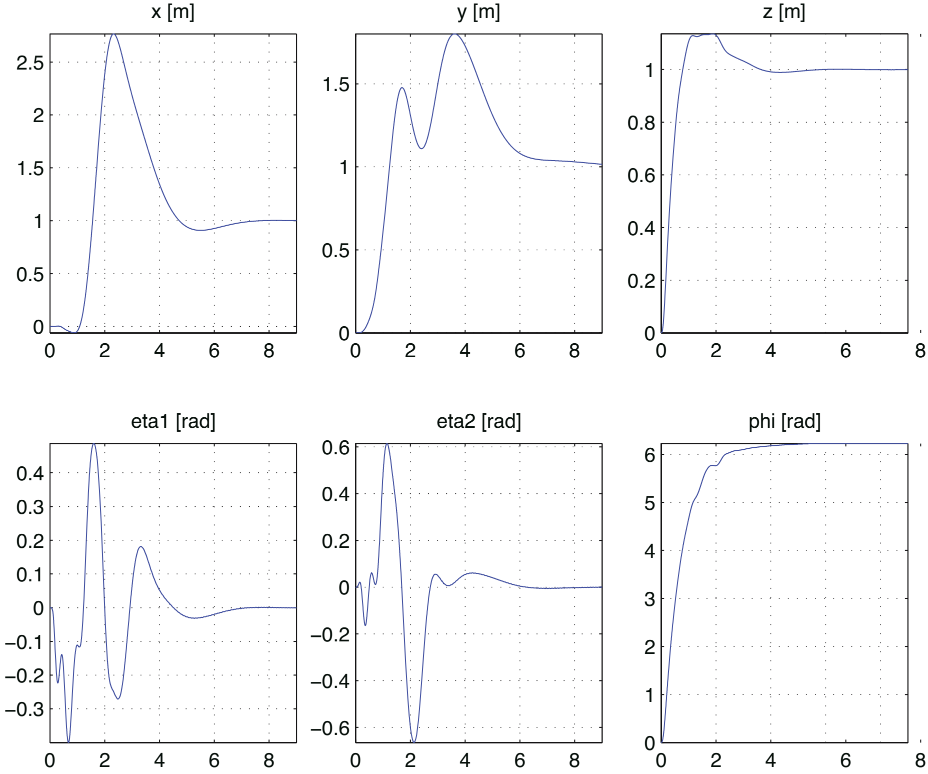


Fig. 3. Simulation results for the MPC scheme

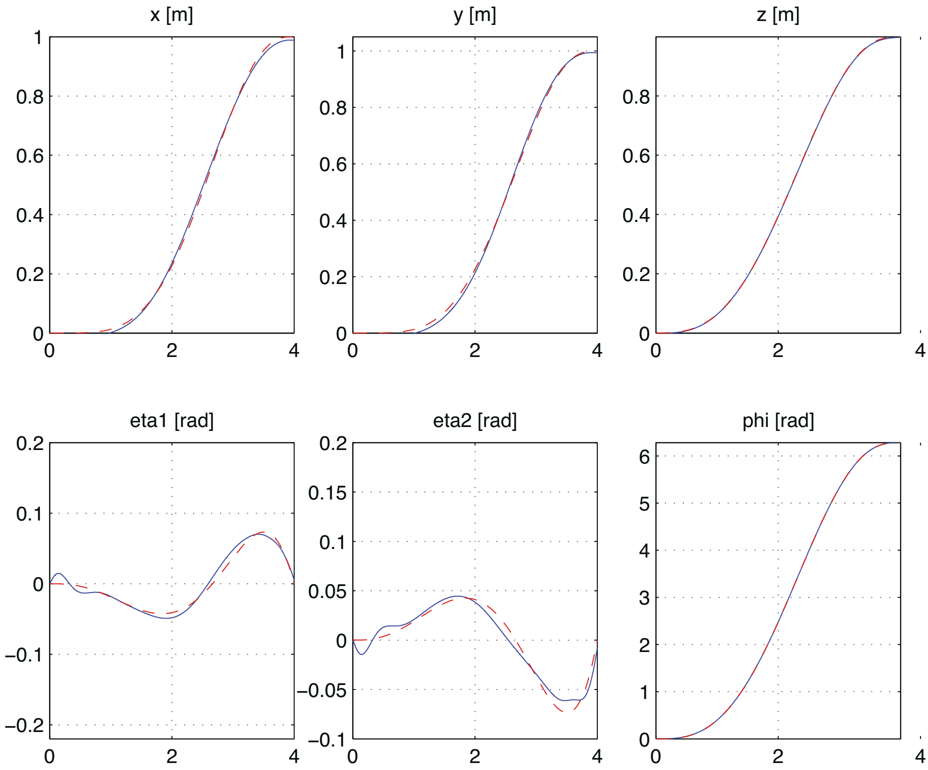


Fig. 4. Simulation results for the Two-time-scale scheme. x_{ref} and x are given by the dashed and solid lines, respectively.

with $\bar{x}_{sp}$ the state setpoints, $\bar{x} = \left[\, X(1:12)\, v_1\ v_2\ \bar{v}_3\ \bar{v}_4 \,\right]^T$,and $\bar{u} = \left[\, u_1\ u_2\ \bar{u}_3\ \bar{u}_4 \,\right]^T$, where

$$\bar{v}_3 = sin(\phi)v_3 + cos(\phi)v_4 \quad \dot{\bar{v}}_3 = \bar{u}_3 \tag{25}$$

$$\bar{v}_4 = -cos(\phi)v_3 + sin(\phi)v_4 \quad \dot{\bar{v}}_4 = \bar{u}_4 \tag{26}$$

The setpoints correspond to the final states to which the system must be driven. This change of variables removes the coupling between the states. The corresponding reference torques and states trajectories for the original model can be computed algebraically from $\bar{x}$ and $\bar{u}$. The weighting matrices R and Q are chosen so as to obtain the desired dynamics. Here, they are chosen diagonal, with the diagonal terms R_D and Q_D given as:

$$R_D = 10^{-9} \cdot \left[1.8 \cdot 10^{-3}\ 0.1058\ 28.125\ 27.38\right]$$

$$Q_D(1:8) = 10^3 \cdot \left[1\ 1.5\ 1\ 0.5\ 0.1\ 1\ 0.5\ 0.45\right]$$

$$Q_D(9:16) = 10^3 \cdot \left[0.5\ 0.5\ 0.5\ 0.5\ 0\ 0\ 0\ 0\right]$$

The choice of the prediction horizon T for the nonlinear MPC scheme is not obvious. Too short a prediction horizon tends to lead to stability problems, while too long an horizon is not desirable from a computational point of view [4]. The prediction horizon chosen for the MPC is $T = 4[s]$, with the sampling time $\delta = 0.1\,[s]$.

With no parametric uncertainty in C_s and C_d, the MPC scheme is able to move the VTOL nicely to the desired setpoints despite the fact that the inputs are computed based on the simplified model. However, MPC struggles when the parametric uncertainty on the aerodynamical coefficient exceeds 10 percent. Figure 3 shows the control performance for a 10 percent perturbation, the perturbation being applied to the four propellers as indicated in Subsection 4.2. The control is slow, each optimization takes minutes (the exact computation time depends on the algorithm used) and exhibits a large overshoot.

4.4 Two-Time-Scale Control

The flatness property of the simplified model allows generating the reference input and state trajectories algebraically, which reduces the computation time significantly. The NE-controller in the fast loop ensures good tracking of the state references. The cost function is :

$$J = \frac{1}{2}\int_{t_k}^{T_f} [(\bar{x} - \bar{x}_{ref})^T Q(\bar{x} - \bar{x}_{ref}) + (\bar{u} - \bar{u}_{ref})^T R(\bar{u} - \bar{u}_{ref})]dt$$

where the matrices Q and R are the same as for MPC, and $(\bar{x}_{ref}, \bar{u}_{ref})$ are the reference trajectories generated by the system inversion loop. The choice of final time for trajectory generation is $T_f = 4[s]$, and the outputs trajectories are chosen such that the reference velocities and accelerations are zero at T_f.

This control scheme exhibits a nice behavior as shown in Figure 4. The reference input and state trajectories are parametrized using polynomials. They are generated once, and no re-calculation is needed. The computation time for the flatness-based trajectory and feedback generation is fairly low ($1.5[s]$ for the trajectory considered). Figure 5 displays the gains of the NE-controller. Since the gains are strongly time varying, the NE-controller cannot be approximated by a LQR.

4.5 Stability

The stability analysis of two-time-scale systems is usually treated within the singular perturbation framework, such as in [5]. However, considering that the NE-controller approximates the optimality objective of the MPC, it is reasonable to seek a stability proof that shows that the slow and the fast loops work toward the same goal, which does not require a time-scale separation. This work is part of ongoing research.

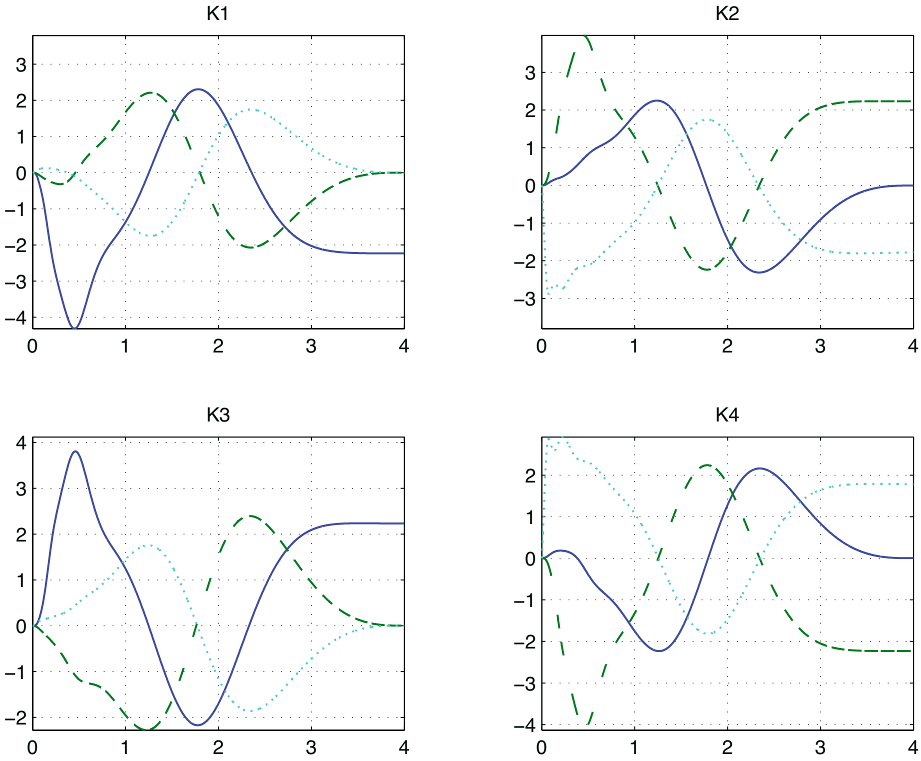


Fig. 5. Selected elements of the (4 × 16)-dimensional matrix of NE-controller gains to track the flatness-based trajectories; the gains shown correspond to the states η_1 (solid), η_2 (dashed) and ϕ (dotted)

4.6 Handling Constraints

MPC is well suited to handle input and state constraints. As far as the system-inversion method is concerned, the flat output trajectories are parameterizations of the input and state trajectories. Hence, using system inversion, it is possible to generate flat output reference trajectories that take input and state constraints into account. Though the problem of adjusting the flat output trajectories in order to respect inputs and states constraints can be difficult [4], designing flat output trajectories that respect output constraints is straightforward. Furthermore, since the flat outputs are independent and freely assignable, no dynamical constraint limits their tracking performance. Though hard constraints cannot be taken into account by the NE-controller, using a reasonable back-off from the flat output constraints and a sufficiently aggressive NE-controller, it is arguably possible to respect the flat output constraints (provided that states and inputs are unconstrained). Yet, enforcing input and state constraints using the proposed approach is still far more challenging.

5 Conclusion

This paper has proposed a two-time-scale control scheme that uses repeated trajectory generation in a slow loop and time-varying linear feedback based on the neighboring-extremal approach in a faster loop. The slow loop provides reasonable reference trajectories, while the fast loop ensures robustness. Feedforward trajectory generation is obtained using flatness-based system inversion.

The two-time-scale approach as well as MPC have been used in simulation to control a VTOL flying structure. Though the simplified model of the structure is flat, control based on feedback linearization is not appropriate because it lacks robustness with respect to the model uncertainties typically encountered in VTOL structures. MPC requires a high re-optimization frequency and, in addition, cannot accommodate large model uncertainties. In contrast, the proposed two-time-scale control scheme is sufficiently robust that it does not require recalculation of the reference trajectories. The flatness-based approach is very fast, and will be used for experimental implementation on a laboratory-scale VTOL structure, for which re-generation of the reference trajectories may be necessary.

References

[1] Bemporad, A. (1998). Reducing conservatism in predictive control of constrained systems with disturbances. In: *37th IEEE Control and Decision Conference. Tampa, FL. pp. 1384–89.*

[2] Bemporad, A. and M. Morari (1999). Robust Model Predictive Control: A Survey. *Springer Verlag.*

[3] Bryson, A. E. (1999). Dynamic Optimization. *Addison-Wesley, Menlo Park, California*

[4] Camacho, E. and Bordons C. (2005). Nonlinear Model Predictive Control: an Introductory Survey *NMPC05, Freudenstadt, Germany*

[5] Christofides, P.D. and Daoutidis P. (1996). Feedback Control of Two-Time-Scale Nonlinear Systems *International Journal of Control.* **63**, 965–994

[6] DeHaan D. and Guay M. (2005). A New Real-Time Method for Nonlinear Model Predictive Control *NMPC05, Freudenstadt, Germany*

[7] Fliess, M., J. Lévine, Ph. Martin and P. Rouchon (1995) Flatness and defect of nonlinear systems: Introductory theory and examples *International Journal of Control* **61**(6), 1327–1361.

[8] Fliess, M., J. Lévine, Ph. Martin and P. Rouchon (1999). A Lie-Bäcklund approach to equivalence and flatness of nonlinear systems *IEEE Trans. Automat. Contr.* **38**, 700–716.

[9] Gros, S. (2005). Modeling and control of a VTOL structure. *Internal Report. Laboratoire d'Automatique, Ecole Polytechnique Fédérale de Lausanne*

[10] Hagenmeyer V. and Delaleau E. (2003). Exact feedforward linearization based on differential flatness *International Journal of Control* **76**, 573–556.

[11] Kouvaritakis, B., J. A. Rossiter and J. Schuurmans (2000). Efficient robust predictive control. *IEEE Trans. Automat. Contr.* **45**(8), 1545–49.

[12] Lee, J. H. and Z. Yu (1997). Worst-case formulations of model-predictive control for systems with bounded parameters. *Automatica* **33**(5), 763–781.

[13] Alamir M. (2005). A low dimensional contractive NMPC scheme for nonlinear systems stabilization : Theoretical framework and numerical investigation on relatively fast systems. *NMPC05, Freudenstadt, Germany.*

[14] Mayne, D. Q., J. B. Rawlings, C. V. Rao and P. O. M. Scokaert (2000). Constrained model predictive control: Stability and optimality. *Automatica* **36**(6), 789–814.

[15] Morari, M. and J. H. Lee (1999). Model predictive control: Past, present, and future. *Comp. Chem. Eng.* **23**, 667–682.

[16] Ronco, E., B. Srinivasan, J. Y. Favez and D. Bonvin (2001). Predictive control with added feedback for fast nonlinear systems. In: *European Control Conference*. Porto, Portugal. pp. 3167–3172.

[17] Scokaert, P. O. and D. Q. Mayne (1998). Min-max feedback model predictive control for constrained linear systems. *IEEE Trans. Automat. Contr.* **43**, 1136–1142.

Receding Horizon Control for Free-Flight Path Optimization

Xiao-Bing Hu[1] and Wen-Hua Chen[2]

[1] Department of Informatics, University of Sussex, UK
xiaobing.Hu@Sussex.ac.uk
[2] Department of Aero. and Auto. Engineering, Loughborough University, UK
W.Chen@lboro.ac.uk

Summary. This paper presents a Receding Horizon Control (RHC) algorithm to the problem of on-line flight path optimization for aircraft in a dynamic Free-Flight (FF) environment. The motivation to introduce the concept of RHC is to improve the robust performance of solutions in a dynamic and uncertain environment, and also to satisfy the restrictive time limit in the real-time optimization of this complicated air traffic control problem. Compared with existing algorithms, the new algorithm proves more efficient and promising for practical applications.

1 Introduction

"Free-Flight"(FF) is one of the most promising strategies for future air traffic control (ATC) systems [1, 2]. Within the FF framework, each individual aircraft has the first responsibility to plan its flight in terms of safety, efficiency and flexibility. One of the key enabling techniques is real-time path planning using onboard flight management systems. Reference [3] proposes an effective Genetic Algorithm (GA) for searching optimal flight paths in an FF environment, where no pre-defined flight routes network exists. However, two questions arise for the GA in [3]: how to cope with unreliable information in a dynamic environment, and how to improve real-time properties.

This paper introduces the concept of Receding Horizon Control (RHC) to the GA in [3] and then develops a more efficient algorithm for online optimizing flight paths in a dynamical FF environment. As an N-step-ahead online optimization strategy, firstly, RHC provides a promising way to deal with unreliable information for far future, and therefore increase the robustness/adoptation of the algorithms against environmental uncertainties/changes; secondly, the introduction of RHC can significantly reduce the heavy computational burden of the GA in [3] to an acceptable level. These achievements mainly rely on carefully choosing horizon length and properly designing terminal penalty in the newly proposed algorithm.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 565–572, 2007.
springerlink.com

2 Online Flight Path Optimization in FF Environment

2.1 Optional Free Flight Paths

In contrast to conventional pre-defined flight routes networks, there are numerous optional free flight paths in an ideal FF environment, as illustrated in Fig.1. Following [4], this paper uses the concept of "time-slice" and a set of discrete optional headings to transform the non-conflict-airspace into a dynamic flight routes network, and the optimization problem can be reasonably simplified.

Time-slice and discrete optional headings set are two system parameters which determine the complexity of the flight routes network. As discussed in [3], longer time-slice and less optional headings lead to a less flexible network; in the opposite extreme, the network becomes unnecessarily complicated. Referring to some papers on air conflict detection and resolution [4], where 5-min-long time interval and 10° discrete angular change for optimizing only local manoeuvres is adopted, this paper, to optimize global flight paths, uses a 10-min time-slice and a discrete set

$$\Omega = [0^\circ, 10^\circ, 20^\circ, \cdots, 350^\circ, \theta_{dire}] \tag{1}$$

where θ_{dire} is the direct-heading, which is defined as the direction of the destination airport with reference to the waypoint where the aircraft arrives at the end of the current time-slice. The ground ATC systems are supposed to periodically broadcast environmental information, particularly data of unavailable-regions, to each individual aircraft. Each individual aircraft uses the latest information to optimize the remained flight path starting from the next time-slice. An optional flight path is composed of a series of sub-trajectories associated with time-slices. The sub-trajectory for the current time-slice is determined by the previous run of optimization.

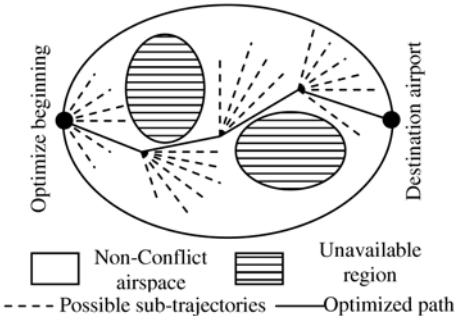


Fig. 1. Optimized path in an FF environment

2.2 Performance Index for Flight Path Optimization

In this paper, for the sake of simplification, only flight time cost is chosen as the index for flight path optimization. Flight time cost can be easily transformed into

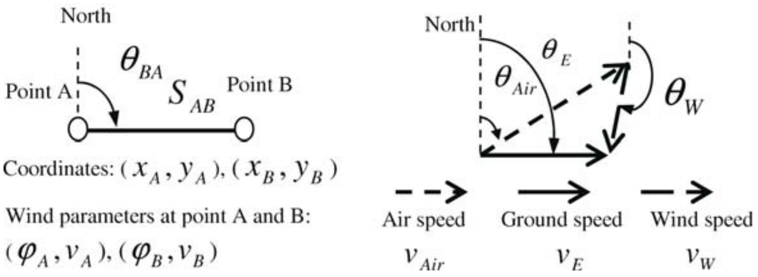


Fig. 2. Variables and parameters of a sub-trajectory and related speeds

other useful indexes for flight path optimization, such as fuel cost [5]. According to the discussion in Section 2.1, an optional flight path is determined by a number of waypoints. When the heading and the beginning waypoint of a sub-trajectory are given, since the flight time along a sub-trajectory is a time-slice (i.e., 10 minutes), the coordinates of the end waypoint of this sub-trajectory can be calculated by referring to Fig.2

$$x_B = x_A + S_{AB} \cos\theta_{BA}, \qquad y_B = y_A + S_{AB} \sin\theta_{BA} \tag{2}$$

where S_{AB} is the distance between two waypoints and

$$S_{AB} = v_E T_{ts}, \theta_{BA} = \theta_E \tag{3}$$

$$v_E = \sqrt{v_W^2 + v_{Air}^2 + 2v_E V_{Air} \cos(\theta_W - \theta_{Air})} \tag{4}$$

$$\theta_E = \theta_{Air} + \sin^{-1}(v_W \sin(\theta_W - \theta_{Air})/V_E) \tag{5}$$

$$v_{Air} = f_{M2v}(M_{opti}, h_C) \qquad \theta_W = \varphi_A \qquad v_W = v_A \tag{6}$$

M_{opti} and h_c are cruise Mach and cruise altitude respectively, $f_{M2V}(\cdot)$ is a function calculating air speed with M_{opti} and h_c as inputs, and T_{ts} is 10 minutes. Since a sub-trajectory is very short as the result of the 10-min-long time-slice, it is reasonable to assume that the average wind parameters along the sub-trajectory are the same as those at the beginning waypoint, as described in Eq. 6. (x_B, y_B) are then used as the beginning waypoint of new sub-trajectory, and the wind parameter (φ_B, v_B) can then be calculated by an interpolation method proposed in [6] based on (x_B, y_B) and atmospheric conditions broadcasted by ATC agencies. The coordinates of the end waypoint of the new sub-trajectory can be calculated in the same way. The computation of sub-trajectories keeps going on until the destination airport is reached.

For the last sub-trajectory in an optional flight path, the end waypoint is the destination airport, and the actual flight time along the last sub-trajectory needs to be calculated using a similar method as Eq.(2-6). Suppose the flight time along the last sub-trajectory is t_{last}, and, excluding the last sub-trajectory, there are sub-trajectories in an optional flight path. Then the corresponding flight time cost is

$$J_1 = \bar{N}T_{ts} + t_{last} \tag{7}$$

3 RHC Algorithms

Similar to most other existing methods (e.g., see [7]), to online optimize FF paths, the GA in [3] optimizes, in each time-slice, the rest flight path from the end of current sub-trajectory to the destination airport. As a consequence, it suffers heavy computational burden, although it was proved to be effective in searching optimal paths in an FF environment. Also, the robustness of the algorithm in [3] against unreliable information in a dynamic FF environment has not been addressed.

3.1 The Idea of RHC

The proposed algorithm takes advantage of the concept of RHC to overcome the above problems in [3]. RHC is a widely accepted scheme in the area of control engineering, and has many advantages against other control strategies. Recently, attention has been paid to applications of RHC in those areas such as management and operations research [8]. Simply speaking, RHC is an N-step-ahead online optimization strategy. At each step, i.e., time-slice, the proposed RHC algorithm optimizes the flight path for the next N time-slices into the near future. Therefore, no matter how long the flight distance is, the online computational time for each optimization is covered by an upper bound, which mainly depends on N, the horizon length. Also, a properly chosen receding horizon can work like a filter to remove unreliable information for the far future.

The online optimization problem in the proposed RHC algorithm is quite different from that in conventional dynamic optimization based methods, such as the GA in [3], where J_1 given in (7) is chosen as the performance index to be minimized in online optimization. The performance index adopted by the proposed RHC algorithm is given as

$$J_2(k) = N(k)T_{ts} + W_{term}(k) \tag{8}$$

where $W_{term}(k)$ is a terminal penalty to assess the flight time from the last waypoint to the destination airport. The discussion about $W_{term}(k)$ will be given later and more detailed discussion can be found in [9]. The proposed RHC algorithm for optimizing flight paths in a dynamic FF environment can be described as following:

S1: When an aircraft takes off from the source airport, fly the departure program, let $k = 0$, and set $P(0)$ as the allocated departure fix of the departure program.

S2: Receive updated environment data from ATC agencies, set P(k) as the initial point to start flight path optimization, and then solve the following minimization problem

$$\min_{P(k+1|k),P(k+2|k),\ldots,P(k+N|k)} J_2(k) \tag{9}$$

subject to available headings in Ω and unavailable regions, where $P(k+i|k), i=1,\ldots,N$, is the end waypoint of ith sub-trajectory in an original potential flight path at kth step. Denote the optimal solution as $[\hat{P}(k+1|k), \hat{P}(k+2|k), \ldots, \hat{P}(k+N)|k]$, and the associated shortcut-taken flight path as $[\hat{P}(k+1|k), \hat{P}(k+2|k), \ldots, \hat{P}(k+ceil(M(k))|k]$, where $M(k)$ is the number of time slices in the shortcut-taken flight path, and *ceil* rounds $M(k)$ to the nearest integer towards infinity.

S3: When the aircraft arrives at $P(k)$, set $P(k+1) = \hat{P}_f(k+1|k)$ and then fly along the sub-trajectory determined by $[P(k), P(k+1)]$.

S4: If $P(k+1)$ is not the destination airport, let $k = k+1$, and go to Step 2; otherwise, the algorithm stops.

3.2 The Length of Receding Horizon and Terminal Penalty

The choice of N, the horizon length, is important to design the proposed algorithm. The online computational time for each optimization is covered by an upper bound, which mainly depends on N and can be estimated through simulations. As long as the time-slice is larger than the upper bound, no matter how long the entire flight distance is, the real-time properties of the proposed algorithm are always guaranteed. Also, a properly chosen receding horizon can work like a filter to remove unreliable information for the far future. A larger N results in heavier online computational burden, but if N is too small, the RHC algorithm becomes "shortsighted", and the performance significantly degrades. A properly chosen N should be a good trade-off on these factors which depend on the dynamics of the systems and the quality of the information.

However, the nature of the receding horizon concept makes the proposed algorithm only taking into account the cost within the receding horizon, which implies shortsightedness in some sense. The introduction of terminal penalty $W_{term}(k)$ in $J_2(k)$ can compensate for this shortsightedness. When applying RHC in online FF path optimization, if no terminal penalty is used, very poor performance even instability (in the sense that the aircraft fails to arrive at the destination airport) is observed in [9]. Several choices of the terminal penalty have been proposed and investigated in [9]. Due to space limit, only one terminal penalty is presented in this paper, which is defined as

$$W_{term}(k) = (\beta|\theta_3|/\theta_4 + 1) dis(P_{last}(k), P_{DA})/v_E \tag{10}$$

where θ_3, θ_4 and β are illustrated in Fig.3. $P_{SA}, P_{DA}, P_{prev}(k)$ and $P_{last}(k)$ are the source airport, the destination airport, the second last waypoint in an optional FF path, and the last waypoint in an optional FF path, respectively,

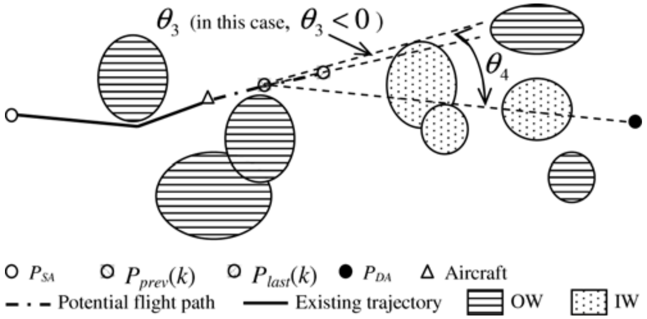


Fig. 3. Definition of a terminal penalty

and IW/OW stands for unavailable airspace regions located on/outside the way directly from $P_{last}(k)$ to P_{DA}. From Fig.3, one can see that: $\theta_3 > 0$ means that the heading of the last sub-trajectory in a potential flight path is over-turning, i.e., aircraft will turn unnecessarily far away from P_{DA}; $\theta_3 < 0$ means under-turning, i.e., aircraft will fly into IW unavailable airspace. $|\theta_3|/\theta_4$ is used to assess how much the over-turning or under-turning is when compared with θ_4. A larger value of $|\theta_3|/\theta_4$ means more excessive turning of the aircraft (either over-turning or under-turning), and will therefore lead to a heavier terminal penalty. $\beta \geq 0$ is a tuning coefficient, and $\beta = 0$ when there is no IW unavailable airspace.

In order to evaluate the proposed RHC algorithm, the simulation system reported in [3] is adopted to set up different FF environments, and the conventional dynamic optimization based GA in [3], denoted as CDO, is also used for the comparative purpose. The proposed RHC algorithm, denoted as RHC, modifies the online optimizer in [3] by taking into account the concept of RHC, as discussed before. More details of the GA optimizer can be found in [3]. In the simulation, unless it is specifically pointed out, the horizon length is $N = 6$, and the terminal penalty $W_{term}(k)$ defined in (10) is adopted for RHC. Six simulation cases are defined in Tab. 1 with different degrees of complexity of the FF environment, where DD stands for the Direct Distance from the source airport to the destination one, and UR for Unavailable Region. In Cases 1 to 3, the UR's are static, while the UR's vary in Cases 4 to 6; in other words, they may move, change in size, and/or disappear randomly. The comparative simulation focuses on online computational times (OCT's) and performances, i.e., actual flight times (AFT's) from the source airport to the destination one. Numerical results are given in Tables 2 to 4, where 10 simulation runs are conducted under either RHC or CDO for each static case, while 200 simulation runs are carried out for each dynamic case. Firstly, RHC is compared with CDO in static cases, and simulation results are given in Table 2. One can see that CDO achieves the best performance, i.e., the least AFT's, in all 3 cases. This is understandable because conventional dynamic optimization strategy, by its nature, should be the best in terms of

Table 1. Six simulation cases

	Static environment			Dynamic environment		
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
DD (nm)	500	1000	2000	500	1000	2000
No. of UR's	1	6	14	1	6	14

Table 2. Simulation results in static cases

	CDO			RHC		
	Case 1	Case 2	Case 3	Case 1	Case 2	Case 3
Ave. OCT(s)	1.2687	8.3675	77.536	2.5675	4.8498	7.3047
Ave. AFT (s)	3965.6	7407.3	14868	3966.2	7421.5	14905
Max.OCT(s)	5.3970	37.479	364.92	5.7970	7.408	15.551
Max. AFT(s)	3966.9	7435.7	14913	3968.7	7480.4	15052

Table 3. Simulation results in dynamic cases

	CDO			RHC		
	Case 4	Case 5	Case 6	Case 4	Case 5	Case 6
Ave. OCT(s)	0.9623	9.448	68.9219	2.4930	3.8419	7.8754
Ave. AFT (s)	4222.0	7475	16192	4221.6	7454.3	15932
Max.OCT(s)	5.317	38.96	347.915	5.8990	6.3190	17.694
Max. AFT(s)	4223.9	8492	16638	4223.1	7995.8	16118

a given performance index when no uncertainties are present. Table 2 shows that the performance of RHC is very close to that of CDO, which implies that RHC works very well in static cases. As for OCT's, RHC is clearly much more efficient than CDO. Since one time-slice is 10-minutes-long, one can see that there is no problem for RHC to run in real-time, while CDO does struggle to complete online computation in some cases. Dynamic cases are our main concern, and some corresponding simulation results are given in Table 3. As for performance, in relatively simple cases like Case 4 and Case 5, CDO and RHC have similar AFT's, while in complicated cases like Case 6, the performance of RHC is better than that of CDO. Again, RHC provides reliable and promising real-time properties against CDO.Tab. 4 highlights that the horizon length N should be properly chosen. If N is too small, the performance is very poor, as is the case of N=1 and N=3 in Tab. 4. However, if N is too large, OCT's increase, but the performance is not necessarily improved further. Instead, the performance could degrade in dynamic cases, as shown for $N = 9$.

Table 4. Influence of N on RHC

		Static environment			Dynamic environment		
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
N=1	OCT(s)	0.8340	0.9365	1.336	0.7337	0.8465	1.2590
	AFT(s)	4006.5	8054.9	17891	4225.1	7976.8	16922
N=3	OCT(s)	1.3003	1.9507	2.539	1.2907	1.4612	2.2652
	AFT(s)	3965.0	7811.0	15674	4226.5	7482.6	16207
N=6	OCT(s)	2.5675	4.8498	7.305	2.4930	3.8419	7.8754
	AFT(s)	3966.2	7421.5	14905	4221.6	7454.3	15932
N=9	OCT(s)	4.6264	10.6017	18.25	4.0966	8.5754	17.737
	AFT(s)	3965.9	7407.6	14894	4221.9	7462.4	16074

4 Conclusions

This paper introduces the concept of RHC to the online optimization of flight paths in a dynamical FF environment. Attention is particularly paid to the horizon length and terminal penalty to guarantee the success of the proposed algorithm. Simulation results show that, regarding performance, the proposed RHC algorithm is as good as the existing algorithm in the absence of uncertainties, and achieves better solutions in a dynamic environment. The main advantage of the RHC algorithm is its high efficiency regarding the online computational time.

References

[1] C.D. Wickens, et al (1998). Airspace System Integration-The Concept of Free Flight. National Academy Press.

[2] S. Kahne (2000). Annual reviews in Control, 24:21–29.

[3] XB Hu, et al (2004). Eng. Applications of Artificial Intelligence,17:897–907.

[4] N. Durand N, et al (1995). Air Traffic Control Quarterly, 3(3).

[5] S.F. Wu (1990). Studies on the flight performance management, Ph.D. Thesis, Nanjing University of Aero. and Astro., China.

[6] J.A. McDonald and Y. Zhao (2000). Time benefits of free-flight for a commercial aircraft. In *The Proc. of the AIAA GNC Conference*, Denver,CO,USA, 2000.

[7] G. Guastalla, et al (2000). Transportation Sci., 34:394–401.

[8] S. Chand et al (2002). Manufacturing and Service Operations Manag., 4:25-43.

[9] X.B. Hu (2005). *New Results and New Developments in Model Predictive Control.* Ph.D. Thesis, Dept. of Aero. and Auto. Eng., Loughborough University.

An Experimental Study of Stabilizing Receding Horizon Control of Visual Feedback System with Planar Manipulators

Masayuki Fujita, Toshiyuki Murao, Yasunori Kawai, and Yujiro Nakaso

Department of Mechanical and Control Engineering, Tokyo Institute of Technology,
2-12-1 S5-26 O-okayama Meguro-ku, Tokyo 152-8552, Japan
fujita@ctrl.titech.ac.jp

Summary. This paper investigates vision based robot control based on a receding horizon control strategy. The stability of the receding horizon control scheme is guaranteed by using the terminal cost derived from an energy function of the visual feedback system. By applying the proposed control scheme to a two-link direct drive manipulator with a CCD camera, it is shown that the stabilizing receding horizon control nicely works for a planar visual feedback system. Furthermore, actual nonlinear experimental results are assessed with respect to the stability and the performance.

1 Introduction

Robotics and intelligent machines need sensory information to behave autonomously in dynamical environments. Visual information is particularly suited to recognize unknown surroundings. In this sense, vision is one of the highest sensing modalities that currently exist. Vision based control of robotic systems involves the fusion of robot kinematics, dynamics, and computer vision to control the motion of the robot in an efficient manner. The combination of mechanical control with visual information, so-called visual feedback control or visual servoing, is important when we consider a mechanical system working in dynamical environments [1].

In previous works, Kelly [2] considered the set-point problem with a static target for a dynamic visual feedback system that includes the manipulator dynamics which is not be negligible for high speed tasks. The authors discussed passivity based control of the eye-in-hand system [3, 4]. However, the control law proposed in [3] is not based on optimization, the desired control performance cannot be guaranteed explicitly.

Receding horizon control, also recognized as model predictive control is a well-known control strategy in which the current control action is computed by solving, a finite horizon optimal control problem on-line [5]. A large number of industrial applications using model predictive control can be found in chemical industries where the processes have relatively slow dynamics. On the contrary, for nonlinear and relatively fast systems such as in robotics, few implementations of the receding horizon control have been reported. For the receding horizon control, many researchers have tackled the problem of stability guarantees. An

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 573–580, 2007.
springerlink.com

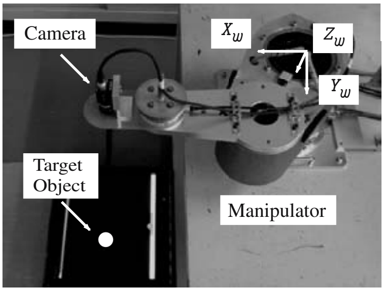


Fig. 1. Planar visual feedback system

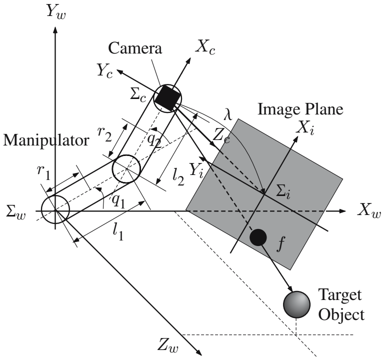


Fig. 2. Schematic diagram

approach proposed by Parisini *et al.* [6] is based on using a quadratic endpoint penalty of the form $ax^T(t+T)Px(t+T)$ for some $a > 0$, some positive definite matrix P and a terminal state $x(t+T)$. Jadbabaie *et al.* [7] showed that closed-loop stability is ensured through the use of a terminal cost consisting of a control Lyapunov function. Moreover, these results were applied to the Caltech Ducted Fan to perform aggressive maneuvers [8, 9]. Visual feedback, however, is not considered here. Predictive control could be of significant benefit when used in conjunction with visual servoing. With the incorporation of visual information, the system could anticipate the target's future position and be waiting there to intercept it [10].

In this paper, stabilizing receding horizon control is applied to the planar visual feedback system in [3], a highly nonlinear and relatively fast system. This represents a first step towards high performance visual servoing targeting more aggressive maneuvers. The main idea is the use of the terminal cost derived from an energy function of the visual feedback system. By applying the proposed control scheme to a two-link direct drive manipulator with a CCD camera, it is shown that the stabilizing receding horizon control nicely works for the planar visual feedback system. Furthermore, the experimental results are assessed with respect to performance.

First, passivity-based control of a planar visual feedback system is reviewed. Next, a stabilizing receding horizon control for a planar visual feedback system using a control Lyapunov function is proposed. Then, the control performance of the stabilizing receding horizon control scheme is evaluated through experiments with a two-link direct drive manipulator with a camera as shown in Fig. 1.

2 Visual Feedback System with Planar Manipulator

The dynamics of n-link rigid robot manipulators can be written as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) = \tau \tag{1}$$

where q, $\dot{q}$ and $\ddot{q}$ are the joint angle, velocity and acceleration, respectively, τ is the vector of the input torque [11]. We assume that the robot evolves in a plane of $n = 2$, referring to Figs. 1 and 2.

The objective of visual feedback control is to bring the camera which is mounted on the end-effector of the manipulator to the position of the target object, i.e., to bring a image feature parameter vector $f = [f_x \ f_y]^T$ to the origin. The image feature parameter vector f is obtained from a perspective transformation. Although details are omitted for lack of space, the planar visual feedback system is given as follows [3, 12].

$$\begin{bmatrix} \dot{\xi} \\ \dot{f} \end{bmatrix} = \begin{bmatrix} -M(q)^{-1}C(q,\dot{q})\xi + w_f M(q)^{-1} J_p^T R_{wc} f \\ -\frac{s\lambda}{z_{wo}} R_{wc}^T J_p \xi - R_{wc}^T \dot{R}_{wc} f \end{bmatrix} + \begin{bmatrix} M(q)^{-1} & 0 \\ 0 & -\frac{s\lambda}{z_{wo}} R_{wc}^T J_p \end{bmatrix} u \quad (2)$$

where $u := [u_\xi^T \ u_d^T]^T$ is the control input, $\xi := \dot{q} - u_d$ is the error vector with respect to the joint velocity, the scalar $w_f > 0$ is a weight for the input torque, R_{wc} is a rotation matrix and J_p is the manipulator Jacobian. A scalar $s > 0$ is the scale factor in pixel/m, λ is the focal length of the camera and z_{wo} is a constant depth parameter. We define the state of the visual feedback system as $x := [\xi^T \ f^T]^T$. The purpose of this paper is to control this planar visual feedback system (2) by using stabilizing receding horizon control.

In previous work [3], the passivity of the visual feedback system (2) is derived by using the following energy function $V(x)$

$$V(x) = \frac{1}{2}\xi^T M(q)\xi + \frac{w_f z_{wo}}{2s\lambda} f^T f. \quad (3)$$

Here, we consider the following control input

$$u = -K\nu := u_k, \quad K := \begin{bmatrix} K_\xi & 0 \\ 0 & K_d \end{bmatrix}, \quad \nu := Nx := \begin{bmatrix} I & 0 \\ 0 & -w_f J_p^T R_{wc} \end{bmatrix} x, \quad (4)$$

where $K_\xi := \mathrm{diag}\{k_{\xi 1}, k_{\xi 2}\} \in \mathcal{R}^{2\times 2}$ and $K_d := \mathrm{diag}\{k_{d1}, k_{d2}\} \in \mathcal{R}^{2\times 2}$ are positive gain matrices. Differentiating $V(x)$ along the trajectory of the system and using the control input u_k, the next equation is derived.

$$\dot{V} = \nu^T u = -x^T N^T K N x. \quad (5)$$

Therefore, the equilibrium point $x = 0$ for the closed-loop system (2) and (4) is asymptotic stable, i.e., u_k is a stabilizing control law for the system.

3 Stabilizing Receding Horizon Control

In this section, the finite horizon optimal control problem for the visual feedback system (2) is considered. Receding horizon schemes are often based on the following cost function.

$$J(u,t) = \int_t^{t+T} l(x(\tau), u(\tau))d\tau + F(x(t+T)), \quad F(x(t+T)) \geq 0 \tag{6}$$

$$l(x(t), u(t)) = x^T(t)Q(t)x(t) + u^T(t)R(t)u(t), \quad Q(t) \geq 0,\ R(t) > 0. \tag{7}$$

The resulting open loop optimal control input u^* is implemented until a new state update occurs, usually at pre-specified sampling intervals. Repeating these calculations yields a feedback control law.

The following lemma concerning a control Lyapunov function is important to prove a stabilizing receding horizon control. The definition for a control Lyapunov function $M(x)$ is given by

$$\inf_u \left[\dot{M}(x) + l(x,u)\right] \leq 0, \tag{8}$$

where $l(x,u)$ is a positive definite function [7].

Lemma 1. *Suppose that the following matrix P is positive semi definite.*

$$P := \rho N^T K N - Q - N^T K^T R K N, \quad \rho > 0. \tag{9}$$

Then, the energy function $\rho V(x)$ of the visual feedback system (2) can be regarded as a control Lyapunov function.

The proof is straightforward using a positive definite function $l(x(t), u(t))$ (7) and the stabilizing control law u_k (4) for the system. Suppose that the terminal cost is the control Lyapunov function $\rho V(x)$, the following theorem concerning the stability of the receding horizon control holds.

Theorem 1. *Consider the following cost function for the visual feedback system (2).*

$$J(u,t) = \int_t^{t+T} l(x(\tau), u(\tau))d\tau + F(x(t+T)) \tag{10}$$

$$l(x(t), u(t)) = x^T(t)Q(t)x(t) + u^T(t)R(t)u(t), \quad Q(t) \geq 0,\ R(t) > 0 \tag{11}$$

$$F(x) = \rho V(x), \quad \rho > 0. \tag{12}$$

Suppose that P (9) is positive semi definite, then the receding horizon control for the visual feedback system is asymptotically stabilizing.

This theorem is proven by using a similar method as in [7], details are omitted due to lack of space. Theorem 1 guarantees the stability of the receding horizon control using a control Lyapunov function for the planar visual feedback system (2) which is a highly nonlinear and relatively fast system. Since the stabilizing receding horizon control design is based on optimal control theory, the control performance should be improved compared to the simple passivity-based control [3], under the condition of adequate gain assignment in the cost function. In this paper, as a first step, we propose unconstrained stabilizing receding horizon control schemes. In the near future, we will consider constraints which represent one of the advantages of receding horizon control, and develop it using level set, see [7].

Moreover, focused on the inverse optimality approach [12], the following corollary is derived.

Corollary 1. *Consider the following weights of the cost function (10)-(12).*

$$Q(t) = qN^T(t)KN(t), \quad q\geq 0 \tag{13}$$

$$R(t) = rK^{-1}, \quad r > 0 \tag{14}$$

$$\rho = 2\sqrt{qr}. \tag{15}$$

Then, the receding horizon control for the visual feedback system is asymptotically stabilizing, the receding horizon control law is

$$u^* = -\sqrt{\frac{q}{r}}KNx \tag{16}$$

and the cost-to-go is given by

$$J^* = \rho V(x). \tag{17}$$

If the weights of the terminal cost function are set to (13)-(15), then the controller that satisfies $\inf_u[\dot{M}(x) + l(x,u)] = 0$ is analytically derived.

In the next section, the stabilizing receding horizon control is applied to a planar visual feedback system. It is expected that the control performance is improved using the receding horizon control.

4 Experimental Results

In this section, the proposed stabilizing receding horizon control is tested on an actual planar visual feedback system which is an image based direct visual servo system. The manipulator used in the experiments (see Fig. 1), is controlled by a digital signal processor (DSP) from dSPACE Inc., which utilizes a powerPC 750 running at 480 MHz. Control programs are written in MATLAB and SIMULINK, and implemented on the DSP using the Real-Time Workshop and dSPACE Software which includes ControlDesk and Real-Time Interface. A XC-HR57 camera is attached to the tip of the manipulator. The video signals are acquired by a frame graver board PicPort-Stereo-H4D and the image processing software HALCON. The sampling time of the controller and the frame rate provided by the camera are 16.7 [ms] and 60 [fps], respectively. To solve the real time optimization problem, the software C/GMRES [13] is utilized. The target object is projected on the liquid crystal monitor. The control objective is to bring the image feature parameter vector f to the origin. The experiment is carried out with the initial condition $q_1(0) = \pi/6$ [rad], $q_2(0) = -\pi/6$ [rad],

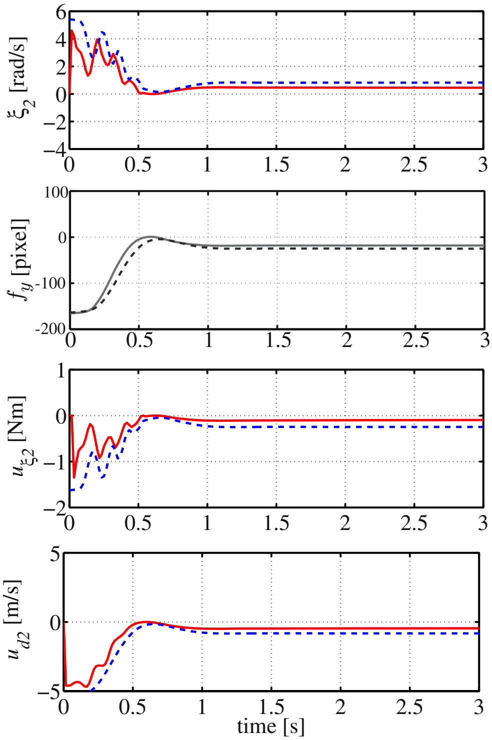


Fig. 3. Experimental comparison with different control schemes (solid: receding horizon control ($T = 0.02$ [s], $\rho = 1$), dashed: passivity based control)

$\dot{q}_1(0) = \dot{q}_2(0) = 0$ [rad/s], $w_f = 0.0001$, $z_{wo} = 0.9$ [m], $s\lambda = 1230$ [pixel], $f(0) = [-120 \ -160]^T$ [pixel] (1 [pixel] = 0.74 [mm]).

In this experiment, we compare the performance of the receding horizon control law proposed in Theorem 1 and the passivity based control law u_k (4). The weights of the cost function (10) were selected as $Q = \mathrm{diag}\{65, 1.5, 10, 100\} \times 10^{-9}$, $R = \mathrm{diag}\{0.04, 1.7, 0.005, 0.00045\}$ and $\rho = 1$ satisfy $P{\geq}0$. The controller parameters for the passivity based control law u_k (4) were empirically selected as $K_\xi = \mathrm{diag}\{6.5, 0.15\}$ and $K_d = \mathrm{diag}\{50, 550\}$. The control input with the receding horizon control is updated within every 16.7 [ms]. It must be calculated by the receding horizon controller within that period. The horizon was selected as $T = 0.02$ [s].

The experimental results are presented in Fig. 3, showing the velocity error ξ_2, the image feature parameter f_y and the control inputs $u_{\xi 2}$ and u_{d2}, respectively. The rise time applying the receding horizon control is shorter than that for the passivity based control. The controller predicts the movement of the target object using the visual information, as a result the manipulator moves more aggressively. This validates one of the expected advantages of the

Table 1. Values of the Integral Cost

Control Scheme	cost
Passivity based Control	106.1
Receding Horizon Control ($T = 0.02$ [s], $\rho = 1$)	61.8
Receding Horizon Control ($T = 0.02$ [s], $\rho = 1.05$)	108.9
Receding Horizon Control ($T = 0.02$ [s], $\rho = 1.1$)	209.2
Receding Horizon Control ($T = 0.05$ [s], $\rho = 1$)	56.3
Receding Horizon Control ($T = 0.1$ [s], $\rho = 1$)	55.1

stabilizing receding horizon control for the visual feedback system. From Fig. 3, the asymptotic stability can be also confirmed experimentally. The steady state performance is also better than for the passivity based control. Still, a non-vanishing steady state error is observed most probably due to the influence of the unmodeled manipulator dynamics(e.g. friction). This problem will be investigated in the near future. We assume that an integrator in the control will improve the steady state performance [14].

The performance for other parameter values T and ρ is compared in terms of the integral cost in Table 1. Since the cost of the stabilizing receding horizon method is smaller than the passivity based control method under conditions of the adequate cost function, it can be easily verified that the control performance is improved. With increasing weight of the terminal cost from $\rho = 1$ to $\rho = 1.1$ the cost increases, too. With higher terminal cost the state value is reduced more strictly, using a large control input. In this experiment, since the weights of the control input are larger than those of the state, the cost increased consequently. As the horizon length increases from $T = 0.02$ to $T = 0.1$, the cost is reduced. In the case of $T = 0.5$, the calculation can not be completed within one sampling interval, due to limited computing power.

5 Conclusions

This paper proposes a stabilizing receding horizon control for a planar visual feedback system, which is a highly nonlinear and relatively fast system. It is shown that the stability of the receding horizon control scheme is guaranteed by using the terminal cost derived from an energy function of the visual feedback system. Furthermore, it is verified that the stabilizing receding horizon control nicely works for the planar visual feedback system through experiments with a nonlinear experimental system. In the experimental results, the control performance of the stabilizing receding horizon control is improved compared to that of the simple passivity based control. In this paper, the stabilizing receding controller was implemented for a low level inner loop, in the near future, we would like to tackle the implementation on a high level outer loop.

Acknowledgement

The authors would like to thank Mr. S. Mimoto, Mr. H. Matsuda and Mr. T. Yamada, Tokyo Institute of Technology for their time and invaluable help.

References

[1] S. Hutchinson, G.D. Hager and P.I. Corke (1996). A tutorial on visual servo control. IEEE Trans. Robotics and Automation. 12(5):651–670.

[2] R. Kelly (1996). Robust asymptotically stable visual servoing of planar robots. IEEE Trans. Robotics and Automation. 12(5):759–766.

[3] A. Maruyama and M. Fujita (1998). Robust control for planar manipulators with image feature parameter potential. Advanced Robotics. 12(1):67–80.

[4] H. Kawai and M. Fujita (2004). Passivity-based dynamic visual feedback control for three dimensional target tracking: stability and L_2-gain performance analysis. Proc. 2004 American Control Conference. 1522–1527.

[5] D.Q. Mayne, J.B. Rawlings, C.V. Rao and P.O.M. Scokaert (2000). Constrained model predictive control: stability and optimality. Automatica. 36(6):789–814.

[6] T. Parisini and R. Zoppoli (1995). A receding-horizon regulator for nonlinear systems and a neural approximation. Automatica. 31(10):1443–1451.

[7] A. Jadbabaie, J. Yu and J. Hauser (2001). Unconstrained receding-horizon control of nonlinear systems. IEEE Trans. Automatic Control. 46(5):776–783.

[8] J. Yu, A. Jadbabaie, J. Primbs and Y. Huang (2001). Comparison of nonlinear control design techniques on a model of the caltech ducted fan. Automatica. 37(12):1971–1978.

[9] A. Jadbabaie and J. Hauser (2002). Control of a thrust-vectored flying wing: a receding horizon – LPV approach. International Journal of Robust and Nonlinear Control. 12(9):869–896.

[10] A.E. Hunt and A.C. Sanderson (1982). Vision-based predictive robotic tracking of a moving target. Technical Report. Carnegie Mellon University.

[11] M.W. Spong, S. Hutchinson and M. Vidyasagar (2006). Robot modeling and control. John Wiley & Sons.

[12] M. Fujita, A. Maruyama, M. Watanabe and H. Kawai (2000). Inverse optimal H_∞ disturbance attenuation for planar manipulators with the eye-in-hand system. Proc. 39th IEEE Conference on Decision and Control. 3945–3950.

[13] T. Ohtsuka (2004). A continuation/GMRES method for fast computation of nonlinear receding horizon control. Automatica. 40(4):563–574.

[14] J.B. Rawlings (2000). Tutorial overview of model predictive control. IEEE Control Systems Magazine. 20(3):38–52.

Coordination of Networked Dynamical Systems

Alessandro Casavola[1], Domenico Famularo[2], and Giuseppe Franzè[1]

[1] DEIS, Università della Calabria, Rende (CS), 87036 Italy
{casavola,franze}@deis.unical.it

[2] DIMET, Università Mediterranea di Reggio Calabria, Reggio Calabria, 89100 Italy
domenico.famularo@unirc.it

Summary. In this paper we present a nonlinear predictive control strategy for the supervision of networked control systems subject to coordination constraints. Such a system paradigm, referred hereafter to as constrained dynamic network, is characterized by a set of spatially distributed dynamic systems, connected via communication channels, with possible dynamical coupling and constraints amongst them which need to be controlled and coordinated in order to accomplish their overall objective. The significance of the method is that it is capable of ensuring no constraints violation and loss of stability regardless of any, possibly unbounded, time-delay occurrence. An application to the coordination of two autonomous vehicles under input-saturation and formation accuracy constraints is presented.

1 Introduction

The advent of wireless communication networks allows the conceivability of new challenging control applications, as those of accomplishing coordinated dynamic tasks amongst a network of remotely located dynamic systems connected via the Internet or other communication networks as depicted in Fig. 1. There, the master station is in charge of supervising and coordinating the slave systems. In particular, r_i, w_i, x_i, y_i and c_i represent respectively: the nominal references, the feasible references, the states, the performance-related outputs and the coordination-related outputs of the slave systems. In such a context, the supervision task can be expressed as the requirement of satisfying some tracking performance, viz. $y_i \approx r_i$, whereas the coordination task consists of enforcing some constraints $c_i \in \mathcal{C}_i$ and/or $f(c_1, c_2, ..., c_N) \in \mathcal{C}$ on each slave system and/or on the overall network. To this end, the supervisor is in charge of modifying the nominal references into the feasible ones, when the tracking of the nominal path would produce constraints violation.

Examples of constrained spatial networks which would require advanced coordination ability include unmanned flight formations [5] and satellite constellations [8]; fault tolerant control systems for intelligent vehicle highway [10], electric power grids [1] and telerobotics [6]. See also [7, 9] and references therein for a comprehensive and up-to-date discussion on the theoretical and applicative challenges on the topic.

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 581–589, 2007.
springerlink.com

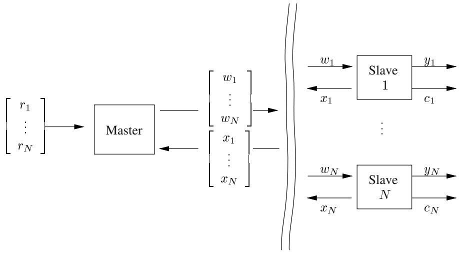


Fig. 1. Dynamic network

The effectiveness of the proposed method will be demonstrated by considering the coordination of the planar motion of two autonomous vehicles, simply modelled as two masses, subject to a dynamic coupling due to elastic and viscous forces between them. See also [8] for a similar but more realistic application. Moreover, actuator saturation is present and has to be taken into account and the coordination task will consist of satisfying, along a given path, a set of formation accuracy constraints with respect to the rigid motion (the two masses rigidly connected). It will be shown that if the round-trip delay of the communication network can be assumed bounded, only data transfer between the master and each slave is necessary. On the contrary, in the case of possibly unbounded time-delay (as over the Internet), also direct data transfer with guaranteed time-to-delivering properties amongst the slaves is required. Teleoperation applications of this strategy have also been undertaken and have been reported in [3].

The overall scheme is based on a extension of the Command Governor approach, introduced in [2] in more standard control contexts, to a distributed master/slaves structure in the presence of communication time-delay. Then, in the next two sessions we briefly introduce the basic CG approach and indicate how such a generalization can be accomplished. Finally, a simulative example will be presented and conclusive remarks end the paper.

2 The Distributed Master/Slaves CG Approach

In this section we will illustrate how to extend the basic CG approach of [2], a suitable variant of standard NMPC schemes, to master/slaves distributed control structures in the presence of no-negligible communication time-delay. For simplicity, the presentation will be limited to a single slave. The extension to the more general case of many-slaves is, *mutatis mutandis*, direct.

The typical system structure we will consider for each remote side is depicted in Fig. 2-(Left) where τ indicates a generic time-delay. It consists of a primal compensated plant, described by the following state-space representation

$$\begin{cases} x(t+1) = \Phi x(t) + G w(t) \\ y(t) = H_y x(t) \\ c(t) = H_c x(t) + L w(t) \end{cases} \tag{1}$$

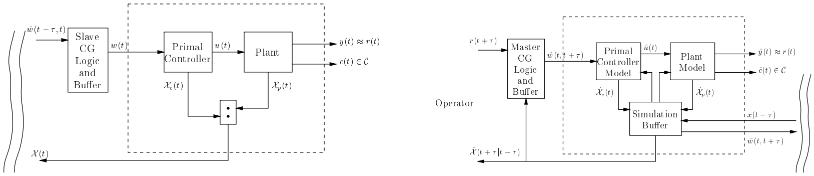


Fig. 2. (Left) Slave side. (Right) Master side.

and of the slave part of the command governor (CG) logic and buffering devices. It is assumed that:

A1 (**Stability**) The system (1) is asymptotically stable;
A2 (**Offset-free**) $H_y(I_n - \Phi)^{-1}G = I_p$, viz. the system features zero tracking error in steady-state for constant inputs;

In (1), $x(t) \in \mathbb{R}^n$ is an enlarged state which may collect plant and compensator states; $\hat{w}(t-\tau,t) \in \mathbb{R}^m$ the command received from the master site. It is to be understood as generated at time $t-\tau$ for being applied exactly at time t or never; $w(t) \in \mathbb{R}^m$ the command that the slave CG logic actually applies to the plant at time t. It would typically be $w(t) = \hat{w}(t-\tau,t)$ if a constant time-delay τ were present. However, when the latter would not be available, the slave CG logic is typically instructed to apply the previous applied command $w(t-1)$; $r(t) \in \mathbb{R}^m$ is a reference sequence which the output $y(t) \in \mathbb{R}^m$ is required to track. Typically $\hat{w}(t-\tau,t) = r(t)$ if no constraints were present; and finally $c(t) \in \mathbb{R}^{n_c}$ the prescribed constraint output vector, viz. $c(t) \in \mathcal{C}$, $\forall t \in \mathbb{Z}_+$, with $\mathcal{C}$ a specified convex and compact set.

At the master side, we consider the system structure of Fig. 2-(Right). In this case, we have the models of the remote systems and their primal controllers along with a buffering data structure which allows predictions and real data fusion. In particular, all future state predictions are updated each time a new piece of information is received from the remote sites. All "hatted" variables in the figure have the same meaning of their slave side counterparts.

The basic idea here is that the master CG logic device acts as if the time-delay would not be present by modifying, whenever necessary, the reference $r(t+\tau)$ into $\hat{w}(t,t+\tau)$ so as to avoid possible constraint violations. Because of random, possibly unbounded, time-delay there exists a certain amount of uncertainty at the master side on the actual sequence of commands which the slave CG unit will apply from t onward and, in turn, on the remote state. Therefore, an effective choice of $\hat{w}(t,t+\tau)$ cannot be based only on the state prediction $\hat{x}(t+\tau)$, which would correspond to the timely application of all subsequent commands generated by the master unit as it would result from a constant time-delay τ. On the contrary, it should be based on a discrete set of state predictions $\hat{\mathcal{X}}(t+\tau|t)$, consisting of all state predictions $\hat{x}(t+\tau)$ based on the information available at time t and corresponding to all possible command sequences $\{w(t), w(t+1), ..., w(t+\tau-1)\}$ potentially generated by the slave CG selection strategy as a consequence of all possible combinations of missing data in the given time interval. It is worth pointing out here that, because of constraints, $\hat{w}(t, t+\tau)$ is guaranteed to be admissible only if applied at time $t+\tau$ when, supposedly, the state of the remote plant will be one of the vectors contained in $\hat{\mathcal{X}}(t+\tau|t)$.

In order to make our discussion more precise, let $\tau_f(t)$ and $\tau_b(t)$ be the forward and, respectively, backward time-delays at each time instant t, viz. $\tau_f(t)$ is the delay from the master to the slave unit whereas $\tau_b(t)$ is the delay in the opposite direction. We assume further that the following upper-bounds

$$\tau_f(t) \leq \bar{\tau}_f \text{ and } \tau_b(t) \leq \bar{\tau}_b, \quad \forall t \in \mathbb{Z}_+ \tag{2}$$

are either known due to the nature of the communication channel or prescribed as operative limits within to ensure some level of tracking performance. In the latter case, we can distinguish two different operative modes

$$\text{Normal case:} \quad (2) \text{ holds true;} \tag{3}$$
$$\text{Abnormal case:} \quad \exists t \text{ s.t. } \tau_f(t) > \bar{\tau}_f. \tag{4}$$

Note that the abnormal mode depends only on the forward time-delay. The previous assumptions on time-delay and the two different operative modes allow one to cover in a unified fashion most of communication channels of interest, e.g. different delays between the forward and backward directions, constant or random delay, bounded or possibly unbounded delay, etc.

At each time instant t, let $t_b \leq t$ and $t_f \leq t$ denote respectively the most recent time instants in which the master has received a piece of information from the slave and vice versa. In the normal case (3) it results that $t \geq t_b \geq t - \bar{\tau}_b$ and $t \geq t_f \geq t - \bar{\tau}_f$. On the contrary, $t - t_f$ and $t - t_b$ can be arbitrarily large in the abnormal case (4). Then, we consider the following family of master/slave CG strategies

$$\text{Master CG -}\hat{w}(t, t+\bar{\tau}_f) := F_1\left(r(t+\bar{\tau}_f), \hat{\mathcal{X}}(t+\bar{\tau}_f|t_b), \hat{w}(t-1, t+\bar{\tau}_f-1)\right) \tag{5}$$
$$\text{Slave CG -} \quad w(t) := F_2\left(w(t-1), r(t), \hat{w}(t-\bar{\tau}_f, t)\right) \tag{6}$$

where F_1 and F_2 are memoryless functions which implement the master/slave CG logic. In particular, $\hat{w}(t, t+\bar{\tau}_f)$ in (5) is the command computed at time t for being applied at time $t+\bar{\tau}_f$ and $\hat{w}(t-\bar{\tau}_f, t)$ in (6) is the command generated in the past to be applied at time instant t. Note that such a command may possibly not be available at time t at the slave side. In designing F_1 and F_2 we want to build up a distributed mechanism which consists in selecting, at each time t, commands $\hat{w}(t, t+\bar{\tau}_f)$ and $w(t)$ in such a way that $w(t)$ is the best approximation of $r(t)$ at time t, under the constraint $c(t) \in \mathcal{C}$, $\forall t \in \mathbb{Z}_+$, and irrespective of all possible time-delays such as (3)-(4). Moreover, in the normal case (3) it is further required that: 1) $w(t) \to w_r$ when $r(t) \to r$, w_r being the best feasible approximation of r; 2) the overall master/slave CG logic has a finite time response, viz. $w(t) = \hat{r}$ in finite time, whenever $r(t) \equiv r$. It is worth commenting that in the abnormal case (4) the latter tracking performance cannot be satisfied and only stability and constraint satisfaction can be ensured.

A suitable generalization of the standard CG selection logic is given, for the master part, by

$$\hat{w}(t,t+\bar{\tau}_f) = \begin{cases} \min\limits_{w \in \mathcal{V}(\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b))} \|w - r(t+\bar{\tau}_f)\|^2_{\Psi} \text{ if } \mathcal{V}(\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b)) \text{ is non-empty} \\ \hat{w}(t-1, t+\bar{\tau}_f - 1), \text{ otherwise} \end{cases} \tag{7}$$

where $\Psi = \Psi' > 0_p$ and $\|w\|^2_{\Psi} := w'\Psi w$, and $\mathcal{V}(\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b))$ collects all step virtual commands, whose corresponding constraints vector predicted evolutions starting at time $t+\bar{\tau}_f$ from any $x \in \hat{\mathcal{X}}(t+\bar{\tau}_f|t_b)$ are satisfied for all future time instants. The rationale underlying the above strategy hinges upon the property of virtual command sequences ensuring that if w is an admissible command at time t from the state x, it will be as such in all future time instants if constantly applied. Note that $\mathcal{V}(\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b))$ may be empty, this means that the actual uncertainty on $x(t+\bar{\tau}_f)$ is so large that we cannot use (7) to compute an admissible virtual command. However, the previously computed virtual command $\hat{w}$ is still admissible and we are authorized to send it to the slave CG unit for being applied at time t.

The slave part of the CG logic is far simpler and reduces to

$$w(t) = \begin{cases} \hat{w}(\cdot,t), \text{ if available and } \|\hat{w}(\cdot,t) - r(t)\|^2_{\Psi_w} < \|w(t-1) - r(t)\|^2_{\Psi_w} \\ w(t-1), \text{ otherwise} \end{cases} \tag{8}$$

The basic properties of the above master/slave strategy have been described in details in [4] and are here briefly condensed.

Theorem 1. Let the assumptions A1-A2 hold true. Consider the system (1) along with the CG master/slave (7)-(8) selection strategy. Then:
Abnormal case (possibly unbounded time-delay):

1. The cardinality of $\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b)$ may become unbounded;
2. The set $\mathcal{V}(\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b))$ is finitely determined;
3. $\mathcal{V}(\bar{\tau}_f|0)) \neq \emptyset$, need not imply that $\mathcal{V}(\mathcal{X}(t+\bar{\tau}_f|t_b))$ will be non-empty in some future time-instant;
4. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$;
5. The overall system remains asymptotically stable but tracking performance may be lost, viz. $w(t) \not\to w_r$ as $r(t) \equiv r$;

Normal case (bounded time-delay):

1. The cardinality of $\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b)$ is bounded;
2. If the set $\mathcal{V}(\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b))$ is empty at a certain time instant t, it remains empty for a finite number of steps only;
3. If the set $\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b)$ consists of a single vector, then $\mathcal{V}(\hat{\mathcal{X}}(\bar{\tau}_f|0))$ non-empty implies $\mathcal{V}(\hat{\mathcal{X}}(t+\bar{\tau}_f|t-\bar{\tau}_b))$ non-empty for all $t \in \mathbb{Z}_+$;

4. $c(t) \in \mathcal{C}$ for all $t \in \mathbb{Z}_+$;
5. The overall system remains asymptotically stable and tracking performance are never lost. In particular, $w(t) \to w_r$ as $r(t) \equiv r$. □

The previous CG master/slave strategy has a lot of customizing possibilities that can be exploited in order to trade-off between tracking performance and robustness with respect to time-delay, ultimately depending on the choice of the predictions to be contained in $\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b)$. Hereafter we will consider the extreme cases with respect to the cardinality of $\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b)$, denoted as Lowest-Data-Redundancy (LDR) and Highest-Data-Redundancy (HDR) schemes.

LDR: It contains only one prediction, viz. $\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b) = \{\hat{x}(t+\bar{\tau}_f|t_b)\}$ with $\hat{x}(t+\bar{\tau}_f|t_b) := \Phi^{t+\bar{\tau}_f-t_b}x(t_b) + \sum_{k=0}^{t+\bar{\tau}_f-t_b-1} \Phi^k G\hat{w}(t-1-k, t+\bar{\tau}_f-1-k)$. It is based on the optimistic assumption that data are never lost. It works well during normal phases but it may degrade remarkably during abnormal phases;

HDR: It is based on the pessimistic assumption that data are always lost. Then, $\hat{\mathcal{X}}(t+\bar{\tau}_f|t_b)$ has to contain all predictions corresponding to all possible combinations of admissible commands application. Observe that at each time instant two possibilities arise: to apply the scheduled command (if available) or keep to apply the most recent applied command. For this strategy, the tracking performance is quite independent from the occurrences of normal or abnormal phases.

Remark 1. It is worth pointing out that the LDR strategy requires a data resynchronization procedure each time the slave CG unit does not receive a new command from the master. This procedure has been described in [3].

3 Example: Two Dynamically Coupled Autonomous Vehicles

In this example we want to coordinate the motion of two autonomous vehicles by using a communication channel subject to possibly unbounded time-delay.

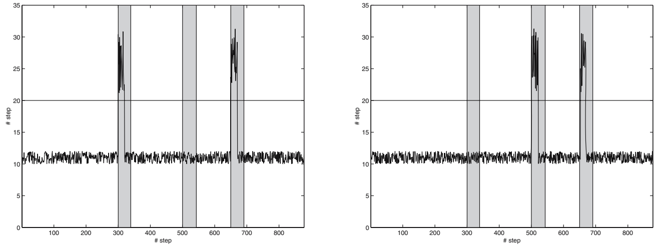


Fig. 3. (Left) Communication time-delay between Master and Slave 1. (Middle) Communication time-delay between Master and Slave 2.

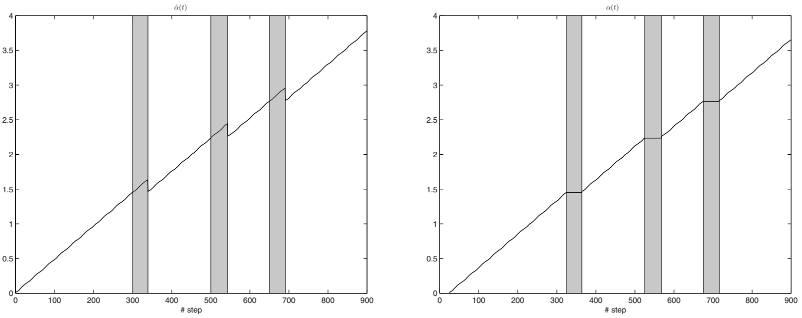


Fig. 4. (Left) (Middle) $\hat{\alpha}(t)$ computed, $r(\hat{\alpha}(t))$ is the corresponding reference. (Right) $\alpha(t)$ applied, $r(\alpha(t))$ is the corresponding reference.

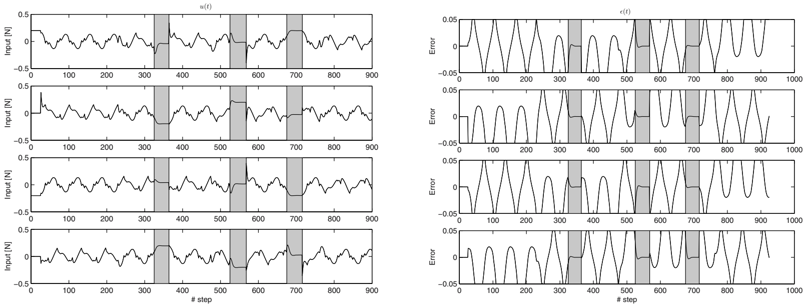


Fig. 5. (Left) Forces $F_{i,j}^{x,y}$ within the saturation limits. (Right) Tracking errors within the prescribed limits.

The LDR strategy will be used and a direct link between the two vehicles is assumed.

The state-space representation of the overall system is given by the following equations

$$\begin{aligned} m_1\ddot{x}_1 &= -k(x_1 - x_2) - \beta(\dot{x}_1 - \dot{x}_2) + F_1^x,\ m_1\ddot{y}_1 = -k(y_1 - y_2) - \beta(\dot{y}_1 - \dot{y}_2) + F_1^y \\ m_2\ddot{x}_2 &= -k(x_2 - x_1) - \beta(\dot{x}_2 - \dot{x}_1) + F_2^x,\ m_2\ddot{y}_2 = -k(y_2 - y_1) - \beta(\dot{y}_2 - \dot{y}_1) + F_2^y \end{aligned} \tag{9}$$

where m_1 and m_2 are the two masses, k the spring constant, β the viscous coefficient of the damper and F_i^x and F_i^y, $i = 1, 2$, the forces acting as inputs. Each subsystem is locally pre-compensated by a suitable controller ensuring offset-free tracking error to constant set-points on positions.

The constraint set $\mathcal{C}$ is described by $\left|F_{i,j}^{x,y}(t)\right| < 1\ [N],\ \ |y_i(t) - r_i(\alpha(t))| < 0.05\ [m]$, where $\alpha(t)$ is a time-depending real parameter in $[0,\ 1]$ which will be used to parameterize the nominal path $r(\alpha(t))$. It is possible to show that the previous CG scheme can be formulated in terms of $\hat{\alpha}(t)$ (in place of $\hat{w}(t)$), which allows the achievement of the feasible paths by selecting the largest admissible

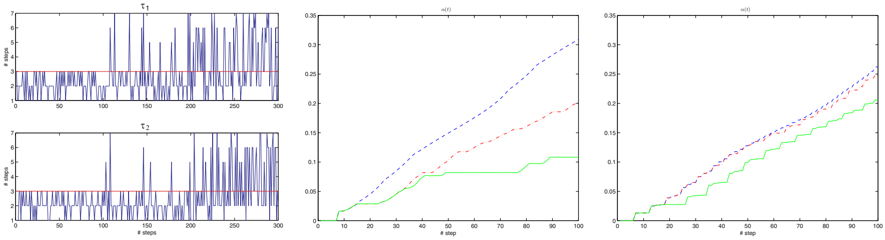


Fig. 6. (Left) Communication time-delays: Master/Slave 1 (upper), Master/Slave 2 (lower) - 100% Normal cases (steps 1-100), 90%/10% Normal/Abnormal cases (steps 100-200), 60%/40% Normal/Abnormal cases (steps 200-300) (Middle) LDR - path-following profiles: 100% Normal cases (dash), 90%/10% Normal/Abnormal cases (dash-dot), 60%/40% Normal/Abnormal cases (continuous). (Right) HDR - path-following profiles: 100% Normal cases (dash), 90%/10% Normal/Abnormal cases (dash-dot), 60%/40% Normal/Abnormal cases (continuous).

increment $\delta\hat{\alpha}(t) := \hat{\alpha}(t+1) - \hat{\alpha}(t)$ along the nominal path. In the simulation we have used an upper-bound of $\bar{\tau}_f = 25$ sampling steps on the time-delay. The LDR scheme is only considered in Figs 3-5. The grey zones here denote the re-synchronization procedure. In particular, in Fig. 3 are reported the communication time-delays, in Fig. 4 the corresponding $\hat{\alpha}(t)$ and $\alpha(t)$ sequences and in Fig. 5 the constrained signals.

A further experiment was carried out in Fig. 6 in order to compare the LDR and HDR strategies. On the left, the used time-delay occurrences were reported. They consist of 100% of Normal cases between steps $1-100$, a 90%/10% percentage of Normal/Abnormal cases between steps $100-200$ and 60%/40% between steps $200-300$. A way to evaluate how the proposed LDR and HDR strategies perform, in terms of tracking performance under Normal and Abnormal phases, is that of considering the corresponding $\alpha(t)$ plots. They are reported in Figs. 6-(Middle) for LDR and (Left) and HDR. One can see that LDR is the best strategy under normal conditions but it is very sensible to the presence of abnormal phases whereas HDR performance is essentially invariant for small percentages of abnormal cases and degrades only slightly for larger percentages. Anyway, for both strategies stability and constraints fulfillments are always ensured.

4 Conclusions

In this paper we have presented a predictive control strategy for the constrained supervision and coordination of dynamic systems in spatial networks. All relevant properties have been summarized and the effectiveness demonstrated by a simulative experiment.

Acknowledgments

This work has been partially supported by MIUR (Italian Ministry for Education, University and Research) under the FIRB Project 2001 *"Constrained control strategies for long-distance robotic surgery"* and the PRIN Project 2004 *"Fault detection and diagnosis, supervision and control and set-points reconfiguration"*.

References

[1] Amin M., (1998), "Complexity and the deregulation of the eletric power industry", Proceedings of *3rd Embracing Complexity (EC3)*, pp. 101-106, Cambridge, MA, USA.

[2] Bemporad A., Casavola A., Mosca E., (1997), "Nonlinear control of constrained linear systems via predictive reference management", *IEEE Trans. Automatic Control*, 42:340–349.

[3] Casavola A., Mosca E., Papini M. (2006), "Predictive teleoperation of constrained dynamical systems via internet-like channels", *IEEE Transaction on Control Systems Technology*, 14:4-681-694

[4] Casavola A., Papini M., Franzè G. (2006), "Supervision of networked dynamical systems under coordination constraints", *IEEE Transaction on Automatic Control*, in press.

[5] Chicka D.F., Speyer J.L. and Park C.G., (1999), "Peak-seeking control with application to formation flight", Proc. of *IEEE CDC 1999*, 2463-2470, Phoenix, Arizona, USA.

[6] Conway L., Volz R.A., Walker M.E., (1990), "Teleautonomous systems: projecting and coordinating intelligent action at distance", *IEEE Trans. Robotics and Automation*, 6:146–158.

[7] Girard A.R., de Sousa J.B., Hedrick K., (2001), "An overview of emerging results in networked multi-vehicle systems", Proc. of *IEEE CDC 2001*, Orlando, FL, USA.

[8] Nakasuka S. and Motohashi S., (1999), "On-orbit dynamics and control of large scaled membrane with satellite at its corners", Proc. of the *14th IFAC Symposium on Automatic Control in Aerospace*, 146-151, Seoul, Korea.

[9] Speyer J.L., (2000), "Control of dynamic systems in spatial networks: applications, results and challenges", *Annual Reviews in Control*, 24:95–104.

[10] Varaiya P., (1993), "Smart carts on smart roads: problems of control", *IEEE Trans. Automatic Control*, 38:195-207.

Distributed Model Predictive Control of Large-Scale Systems

Aswin N. Venkat[1], James B. Rawlings[2], and Stephen J. Wright[3]

[1] Shell Global Solutions (US) Inc., Westhollow Technology Center, P.O. Box 4327, Hooston, Tx 77210, USA
[2] Department of Chemical and Biological Engineering, University of Wisconsin, Madison, WI 53706, USA
venkat@bevo.che.wisc.edu, jbraw@bevo.che.wisc.edu
[3] Computer Sciences Department University of Wisconsin, Madison, WI 53706, USA
swright@cs.wisc.edu

Summary. Completely centralized control of large, networked systems is impractical. Completely decentralized control of such systems, on the other hand, frequently results in unacceptable control performance. In this article, a distributed MPC framework with guaranteed feasibility and nominal stability properties is described. All iterates generated by the proposed distributed MPC algorithm are feasible and the distributed controller, defined by terminating the algorithm at any intermediate iterate, stabilizes the closed-loop system. The above two features allow the practitioner to terminate the distributed MPC algorithm at the end of the sampling interval, even if convergence is not attained. Further, the distributed MPC framework achieves optimal systemwide performance (centralized control) at convergence. Feasibility, stability and optimality properties for the described distributed MPC framework are established. Several examples are presented to demonstrate the efficacy of the proposed approach.

1 Introduction

With ever increasing demands on productivity and efficiency of operation, the chemical industry today places significant importance on plantwide automation. Improvements in practical control technology can cut costs and raise profits. Designing the best decentralized control configuration for a given large-scale plant is an area of active research. A recent review article on this topic is available [8]. A number of articles have focused on improved plantwide decentralized control. A survey of decentralized control methods for large-scale systems can be found in [16]. Performance limitations arising from the decentralized control framework are described in [3]. Most decentralized controller design approaches approximate or ignore the interactions between the various subsystems [10, 17].

The broad industrial impact of model predictive control (MPC), especially in the chemical industry, is evident from recent reviews [13, 20]. However, MPC subsystems may interact significantly, causing a deterioration in systemwide control performance. A suboptimal strategy for centralized MPC of interconnected systems was proposed in [1]. In [21], a plantwide control strategy based on the integration of linear and nonlinear MPC coupled with a plant decomposition

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 591–605, 2007.
springerlink.com

procedure was described. The opportunity presented for cross-integration within the MPC framework and potential requirements and benefits of such technology has been discussed in [4, 7, 9].

2 Motivation

Consider the distillation column described in [12, p. 813]. Tray temperatures act as inferential variables for composition control. The outputs y_{21}, y_7 are the temperatures of trays 21 and 7 respectively and the inputs L, V denote the reflux flowrate and the vapor boilup flowrate to the distillation column. Two SISO PID controllers with anti-reset windup are used to control the temperatures of the two trays. The implications of the relative gain array (RGA) elements on controller design has been studied in [18]. While the RGA for this system suggests pairing L with y_{21} and V with y_7, we intentionally choose a bad control variable–manipulated variable pairing. PID-1 manipulates V to control y_{21}. PID-2 controls y_7 by manipulating L. It will be shown later that unlike cooperation-based formulations, decentralized or communication-based strategies cannot repair this kind of bad design choice. The PID controllers are tuned employing the rules described in [12, p. 531]. The system exhibits unstable closed-loop behavior due to the poor manipulated variable–control variable pairing. Expectedly, decentralized MPC does not fare any better. For such a small control problem, the fix is well known and obvious; switching the manipulated variable–control variable pairing gives much improved performance (though suboptimal) with the two decentralized PID and MPC controllers.

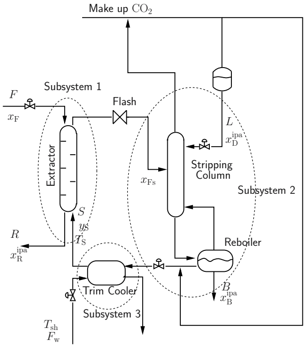


Fig. 1. Supercritical fluid extraction (SFE) process. Supercritical CO_2 is used to extract iso-propyl alcohol (ipa) from a dilute aqueous solution.

In a chemical plant with a network of interconnected units, the choice of which manipulated variable(s) to pair with which control variable(s) is a complicated one. Often, the allocation of the set of control variables and corresponding manipulated variables is dictated by physical and/or operational constraints. In many situations, the simple interconnection of these unit-based control configurations achieves unacceptable closed-loop performance. To illustrate this point, consider the supercritical fluid extraction process (SFE) described in [14, 15]. The SFE process consists of four main units–the extractor, stripper, reboiler and trim cooler. The SFE process is characterized by significant coupling between the different units and is known to represent a challenging plantwide control problem. A fixed decentralized control configuration, based on a physical decomposition of the plant, is assumed. A schematic of the SFE plant with the control divisions is shown in Figure 1. In the decentralized control configuration there are three MPCs, one for each control subsystem. The first MPC manipulates the solvent flowrate (S) to the extractor to control the raffinate composition ($x_{\mathrm{R}}^{\mathrm{ipa}}$) at the outlet of the extractor. The second MPC manipulates the reflux and boilup flow rates (L, V) to control the top and bottom compositions in the stripper ($x_{\mathrm{D}}^{\mathrm{ipa}}, x_{\mathrm{B}}^{\mathrm{ipa}}$). The third MPC manipulates the shell tube temperature (T_{sh}) in the trim cooler to control the temperature of the solvent entering the extractor (T_{S}). All inputs are constrained to lie between an upper and lower limit. In the centralized MPC framework, a single MPC controls the whole plant.

In the decentralized MPC framework, the interconnections between the units are ignored. Consequently, the setpoint tracking performance of outputs $x_{\mathrm{R}}^{\mathrm{ipa}}$ and $x_{\mathrm{D}}^{\mathrm{ipa}}$ exhibits large tracking errors. Also, the upper bound constraint on the solvent flow rate S to the extractor is active at steady state (see Figure 3). Therefore, the setpoint is unreachable under decentralized MPC. Centralized MPC, on the other hand, tracks the new setpoint with much smaller tracking errors. None of the input constraints are active at steady state. Quantitatively, centralized MPC outperforms decentralized MPC by a factor of 350 (Table 1) based on measured closed-loop performance.

In most cases, however, centralized control is not a viable plantwide control framework. To the best of our knowledge, no large-scale centralized models are available today in any field. Operators of large, interconnected systems (an entire chemical plant, for instance) view centralized control as monolithic and inflexible. With many plants already functional with some form of decentralized MPCs, practitioners do not wish to engage in complete control system re-design as would be necessary to implement centralized MPC. While a decentralized philosophy creates tractable modeling and control problems, choosing to ignore the interconnections between subsystems may result in poor systemwide control performance.

A recent article [5] points out that there has been a strong tendency in the control community to look for centralized control solutions; however, the exponential growth of the centralized control law with system size makes its implementation unrealistic for large, networked systems. The only recourse in such situations is to establish a "divide and conquer" strategy that allows one to break the large-scale control problem into several smaller subproblems. The underlying challenge in any "divide and conquer" control strategy is to establish a protocol for integrating different components of the interconnected system to achieve good overall performance.

In this work, a cooperation-based control strategy that facilitates the integration of the various subsystem-based MPCs is described. The interactions among the subsystems are assumed to be stable; system re-design is recommended otherwise. The proposed cooperation-based distributed MPC algorithm is iterative in nature. At convergence, the distributed MPC algorithm achieves optimal (centralized) control performance. In addition, the control algorithm can be terminated at any intermediate iterate without compromising feasibility or closed-loop stability of the resulting distributed controller. The proposed method also serves to equip the practitioner with a low-risk strategy to explore the benefits achievable with centralized control by implementing cooperating MPC controllers instead. In many situations, the structure of the system and nature of the interconnections establishes a natural, distributed hierarchy for modeling and control. A distributed control framework also fosters implementation of a cooperation-based strategy for several interacting processes that are not owned by the same organization.

3 Modeling for Integrating MPCs

Consider a plant comprised of M interconnected subsystems. The notation $\{1, M\}$ is used to represent the sequence of integers $1, 2, \ldots M$.

Decentralized models. Let the decentralized (local) model for each subsystem be represented by a discrete, linear time invariant (LTI) model of the form

$$x_{ii}(k+1) = A_{ii}x_{ii}(k) + B_{ii}u_i(k), \tag{1a}$$

$$y_{ii}(k) = C_{ii}x_{ii}(k), \ \forall\ i \in \{1, M\}, \tag{1b}$$

in which k is discrete time, and we assume (A_{ii}, B_{ii}, C_{ii}) is a minimal realization for each (u_i, y_i) input-output pair.

In the decentralized modeling framework, it is assumed that the subsystem-subsystem interactions have a negligible effect on system variables. Frequently, components of the networked system are tightly coupled due to material/energy and/or information flow between them. In such cases, the "decentralized" assumption leads to a loss in achievable control performance.

Interaction models (IM). Consider any subsystem $i \in \{1, M\}$. The effect of any interacting subsystem $j \neq i$ on subsystem i is represented through a discrete LTI model of the form

$$x_{ij}(k+1) = A_{ij}x_{ij}(k) + B_{ij}u_j(k) \tag{2a}$$

$$y_{ij}(k) = C_{ij}x_{ij}(k), \ \forall\ i, j \in \{1, M\}, j \neq i \tag{2b}$$

in which (A_{ij}, B_{ij}, C_{ij}) denotes a minimal realization for each $(u_{j \neq i}, y_i)$ interacting input-local output pair. The subsystem output is given by $y_i(k) = \sum_{j=1}^{M} y_{ij}(k)$.

Composite models (CM). For each subsystem, the combination of the decentralized model and all the interaction models is termed the composite model (CM). The decentralized state vector x_{ii} is augmented with states arising due to the influence of external subsystems.

Let $x_i = \left[x_{i1}^{'}, \ldots, x_{ii}^{'}, \ldots, x_{iM}^{'}\right]^{'}$ denote the CM states for subsystem i. For notational convenience, the CM for subsystem i is written as

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j \neq i} W_{ij} u_j(k) \tag{3a}$$

$$y_i(k) = C_i x_i(k) \tag{3b}$$

in which $C_i = [C_{i1} \ldots C_{ii} \ldots C_{iM}]$ and

$$A_i = \begin{bmatrix} A_{i1} & & & & \\ & \ddots & & & \\ & & A_{ii} & & \\ & & & \ddots & \\ & & & & A_{iM} \end{bmatrix}, \quad B_i = \begin{bmatrix} 0 \\ \vdots \\ B_{ii} \\ 0 \\ \vdots \end{bmatrix}, \quad W_{ij} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ B_{ij} \\ \vdots \end{pmatrix}.$$

The CM for the entire plant can be written as

$$\left[\begin{array}{c} x_{11} \\ \vdots \\ x_{1M} \\ \hline \vdots \\ \hline x_{M1} \\ \vdots \\ x_{MM} \end{array}\right](k+1) = \left[\begin{array}{ccc|c|ccc} A_{11} & & & & & & \\ & \ddots & & & & & \\ & & A_{1M} & & & & \\ \hline & & & \ddots & & & \\ \hline & & & & A_{M1} & & \\ & & & & & \ddots & \\ & & & & & & A_{MM} \end{array}\right] \left[\begin{array}{c} x_{11} \\ \vdots \\ x_{1M} \\ \hline \vdots \\ \hline x_{M1} \\ \vdots \\ x_{MM} \end{array}\right](k) + \left[\begin{array}{ccc} B_{11} & & \\ & \ddots & \\ & & B_{1M} \\ \hline & \vdots & \\ \hline B_{M1} & & \\ & \ddots & \\ & & B_{MM} \end{array}\right] \begin{bmatrix} u_1 \\ \vdots \\ u_M \end{bmatrix}(k),$$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix}(k) = \left[\begin{array}{ccc|c|ccc} C_{11} & \cdots & C_{1M} & & & & \\ & & & \ddots & & & \\ & & & & C_{M1} & \cdots & C_{MM} \end{array}\right] \left[\begin{array}{c} x_{11} \\ \vdots \\ x_{1M} \\ \hline \vdots \\ \hline x_{M1} \\ \vdots \\ x_{MM} \end{array}\right](k).$$

For large, networked systems after identification of the significant interactions from closed-loop operating data, we expect many of the interaction terms to be zero. In the decentralized modeling framework, all of the interaction terms are assumed zero.

Centralized model. The full plant (centralized) model can be thought of as a minimal realization of the CM for the entire plant. The centralized model is

$$x(k+1) = Ax(k) + Bu(k), \tag{5a}$$

$$y(k) = Cx(k). \tag{5b}$$

4 Communication-Based MPC [1]

In the communication-based MPC (comm-MPC) formulation, each subsystem's MPC exchanges predicted state and input trajectory information with MPCs of interconnected subsystems until all trajectories converge. Convergence of the communicated trajectories is implicitly assumed and is, consequently, a limitation of this control framework. The communication-based controller utilizes the interaction models to quantify the influence of the interacting subsystem inputs on the local subsystem. The effect of the interconnections is considered in the computation of the optimal control law. The objective function is the one for the local subsystem only.

Unstable closed-loop behavior with comm-MPC. It has been shown in [19] that the lack of well defined properties for comm-MPC makes it an unreliable strategy for plantwide control. To illustrate the undesirable consequences of employing the comm-MPC formulation, we revisit the distillation column example described in Section 2. Figure 2 illustrates the performance of the different MPC based control formulations. While centralized MPC (cent-MPC) drives the system to the new setpoint, the inputs V, L in the comm-MPC formulation saturate and subsequently the system is closed-loop unstable.

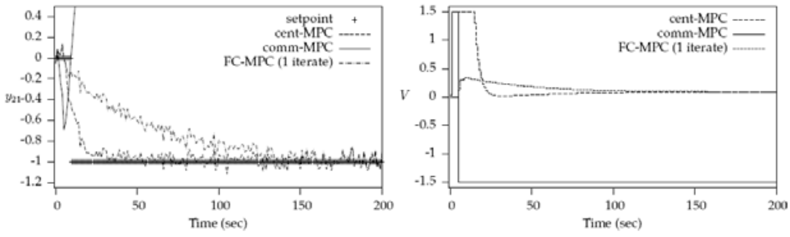


Fig. 2. Distillation column ([12]). Performance of different MPC frameworks.

5 Feasible Cooperation-Based MPC (FC-MPC)

In devising an MPC-based plantwide control strategy, one might expect that modeling the interconnections between the subsystems is sufficient to improve control performance. We know from Section 4 that this idea is incorrect and can potentially cause unstable closed-loop behavior. To provide a means for

[1] Similar Ideas Proposed by [2, 6].

cooperative behavior among the different subsystem-based MPCs, we replace each local objective $\phi_i, i \in \{1, M\}$ by one that represents the systemwide impact of local control actions. One simple choice is to employ a strong convex combination of the local subsystems' objectives as the objective function in each FC-MPC *i.e.*, $\phi = \sum w_i \phi_i,\ w_i > 0,\ \sum_i w_i = 1$.

For each subsystem $i \in \{1, M\}$, the set of admissible controls Ω_i is assumed to be a compact, convex set containing the origin in its interior. We assume that the CM $(A_i, B_i, \{W_{ij}\}_{j \neq i}, C_i)$ is available for each subsystem $i \in \{1, M\}$.

Cost function. The local objective for each subsystem-based MPC is

$$\phi_i\left(\boldsymbol{x}_i^p(k), \boldsymbol{u}_i^p(k); x_i(k)\right) = \sum_{t=k}^{\infty} \frac{1}{2}\left[y_i^p(t|k)' Q_{y_i} y_i^p(t|k) + u_i^p(t|k)' R_i u_i^p(t|k)\right], \tag{6}$$

in which $\boldsymbol{x}_i^p(k) = [x_i^p(k+1|k)', x_i^p(k+2|k)', \ldots]'$, $\boldsymbol{u}_i^p(k) = [u_i^p(k|k)',\ u_i^p(k+1|k)', \ldots]'$ and $Q_i = C_i' Q_{y_i} C_i \geq 0$, $R_i > 0$ are symmetric weighting matrices with $(Q_i^{1/2}, A_i)$ detectable. The notation p is used to indicate the iteration number. During each MPC optimization, the state and input trajectories $(\boldsymbol{x}_{j \neq i}(k), \boldsymbol{u}_{j \neq i}(k))$ of the interacting subsystems' MPCs are not updated; they remain at $(\boldsymbol{x}_{j \neq i}^{p-1}(k), \boldsymbol{u}_{j \neq i}^{p-1}(k))$. For notational convenience, we write $\boldsymbol{x}_i \equiv \boldsymbol{x}_i(k)$ and $\boldsymbol{u}_i \equiv \boldsymbol{u}_i(k)$.

In some cases, the process sampling time may be shorter than the time required for convergence of the iterative algorithm. To allow for intermediate termination, we require that all iterates generated by the cooperation-based MPC algorithm are feasible and that the resulting distributed control law guarantees stability of the nominal closed-loop system. To ensure strict systemwide feasibility of the intermediate iterates, the CM states $x_i(t|k),\ t \geq k$ are eliminated from (6) using (3).

For subsystem i and iterate p, the optimal input sequence $\boldsymbol{u}_i^{p(*)}$ is obtained as

$$\overline{\boldsymbol{u}}_i^{p(*)} \in \arg\min_{\overline{\boldsymbol{u}}_i} \quad \frac{1}{2} \overline{\boldsymbol{u}}_i' \mathcal{R}_i \overline{\boldsymbol{u}}_i + \left(r_i + \sum_{j \neq i}^{M} H_{ij} \overline{\boldsymbol{u}}_j^{p-1}\right)' \overline{\boldsymbol{u}}_i + \text{constant} \tag{7a}$$

$$\text{subject to } u_i(t|k) \in \Omega_i, k \leq t \leq k+N-1, \tag{7b}$$

$$\overline{\boldsymbol{u}}_i' = \left[u_i(k|k)', u_i(k+1|k)', \ldots, u_i(k+N-1|k)'\right], \quad H_{ij} = \sum_{l=1}^{M} w_l E_{li}'(\cdot) Q_l E_{lj},$$

$$\mathcal{R}_i = w_i(\cdot) R_i + w_i E_{ii}^T(\cdot) Q_i E_{ii} + \sum_{j \neq i}^{M} w_j E_{ji}'(\cdot) Q_j E_{ji},$$

$$r_i(k) = w_i E_{ii}'(\cdot) Q_i f_i \widehat{x}_i(k) + \sum_{j \neq i}^{M} w_j E_{ji}'(\cdot) Q_j f_i \widehat{x}_j(k),$$

$$E_{ij} = \begin{bmatrix} B_{ij} & 0 & \ldots\ldots & & 0 \\ A_i B_{ij} & B_{ij} & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ A_i^{N-1} B_{ij} & \ldots & \ldots\ldots & & B_{ij} \end{bmatrix}, \quad f_i = \begin{bmatrix} A_i \\ A_i^2 \\ \vdots \\ \vdots \\ A_i^N \end{bmatrix}, \quad \begin{array}{l} (\cdot)Q_i = \text{diag}\left(Q_i, \ldots, Q_i, \overline{Q}_i\right), \\ (\cdot)R_i = \text{diag}\left(R_i, R_i, \ldots, R_i\right), \end{array}$$

and $\overline{Q}_i$ is an appropriately chosen terminal penalty, as described in the sequel. The symbol $\widehat{x}_i(k)$ denotes the estimate of the CM state vector for subsystem i at discrete time k. The corresponding infinite horizon input trajectory is $\boldsymbol{u}_i^{p(*)\prime} = \left[\overline{\boldsymbol{u}}_i^{p(*)\prime}, 0, 0, \ldots\right]$.

Let the state sequence generated by the input sequence $\boldsymbol{u}_i$ and initial state x be represented as $\boldsymbol{x}_i^{(\boldsymbol{u}_i;x)}$. The notation $\widehat{x}$ is used to denote $[\widehat{x}_1{}', \widehat{x}_2{}', \ldots, \widehat{x}_M{}']$. An algorithm for FC-MPC is described below.

Algorithm 1. *Given* $\left(\boldsymbol{u}_i^0, x_i(k)\right)$ $(\cdot)Q_i \geq 0, (\cdot)R_i \geq 0, i \in \{1, M\}$, $p_{max}(k) \geq 0$ *and* $\epsilon > 0$

```
$p \leftarrow 1, e_i \leftarrow \Gamma\epsilon,\ \Gamma \gg 1$
while $e_i > \epsilon$ for some $i \in \{1, M\}$ and $p \leq p_{max}(k)$
  do $\forall\ i \in \{1, M\}$
    $\boldsymbol{u}_i^{*(p)} \in$ arg min FC-MPC$_i$, (see (7))
    $\boldsymbol{u}_i^p = w_i \boldsymbol{u}_i^{*(p)} + (1 - w_i)\, \boldsymbol{u}_i^{p-1}$
    $e_i = \|\boldsymbol{u}_i^p - \boldsymbol{u}_i^{p-1}\|$
  end (do)
  for each $i \in \{1, M\}$
    Transmit generated input trajectories $(\boldsymbol{u}_i^p)$ to interconnected subsystems
  end (for)
  $\boldsymbol{x}_i^p \leftarrow \boldsymbol{x}_i^{\left(\boldsymbol{u}_1^p, \boldsymbol{u}_2^p, \ldots, \boldsymbol{u}_M^p; \widehat{x}(k)\right)},\ \forall\ i \in \{1, M\}$
  $p \leftarrow p + 1$
end (while)
```

The following properties can be established for cooperation-based MPC using Algorithm 1.

- The cooperation-based cost function is a nonincreasing function of the iteration number p. Since the cost function is also bounded below, it is convergent.
- All limit points generated by Algorithm 1 are optimal *i.e.*, the solution obtained at convergence of the FC-MPC algorithm (Algorithm 1) is within a pre-specified tolerance of the centralized MPC solution.

5.1 State Feedback

Distributed MPC control law. At time k, let the FC-MPC algorithm be terminated after $p(k) = s$ iterates. For the state feedback case, $\widehat{x}_i = x_i$, the actual subsystem state. Let

$$\boldsymbol{u}_i^s(x(k)) = \left[u_i^s(x(k), k)', u_i^s(x(k), k+1)', \ldots\right]',\ \forall\ i\ \in\ \{1, M\} \tag{8}$$

represent the solution to Algorithm 1 after s iterates. Under the state feedback distributed MPC control law, the input applied to subsystem i is $u_i(k) = u_i^s(k|k) \equiv u_i^s(x(k), k)$.

Closed-loop stability (Stable decentralized modes). If $Q_i > 0, R_i > 0$ and the terminal penalty $\overline{Q}_i$ is the solution to the Lyapunov equation $A_i{}'\overline{Q}_i A_i - \overline{Q}_i = -Q_i, \ \forall \ i \in \{1, M\}$ then the closed-loop system $x(k+1) = Ax(k) + Bu(x(k))$, in which $u(x(k)) = [u_1^{p(k)}(x(k), k)', u_2^{p(k)}(x(k), k)', \ldots, \ldots, u_M^{p(k)}(x(k), k)']'$, is exponentially stable under the distributed MPC control law defined by (8) for $x(k) \in \mathcal{X}$, the constrained stabilizable set for the system, and all $p(k) > 0$.

Controller performance index. For the examples presented in this paper, the controller performance index for each plantwide control configuration is calculated as

$$\Lambda_{\text{cost}}(k) = \frac{1}{k} \sum_{j=0}^{k} \sum_{i=1}^{M} \frac{1}{2} \left[x_i(j)' Q_i x_i(j) + u_i(j)' R_i u_i(j) \right]. \tag{9}$$

Table 1. SFE process. Closed-loop control costs associated with different MPC based plantwide control strategies.$\Delta\Lambda_{\text{cost}}(\text{config})\% = \dfrac{\Lambda_{\text{cost}}(\text{config}) - \Lambda_{\text{cost}}(\text{cent})}{\Lambda_{\text{cost}}(\text{cent})} \times 100$.

	Λ_{cost}	$\Delta\Lambda_{\text{cost}}\%$
Centralized MPC	0.306	
Decentralized MPC	107.5	3.5×10^4
FC-MPC (1 iterate)	0.316	0.33
FC-MPC (10 iterates)	0.306	0

SFE process. Consider the SFE process described in Section 2. The control costs associated with the different MPC based plantwide control frameworks are given in Table 1. The presence of an active steady-state input constraint makes the setpoint unreachable under decentralized MPC. The closed-loop performance of the distributed controller derived by terminating the FC-MPC algorithm after 1 iterate is within 0.35% of cent-MPC performance. After 10 iterates, the closed-loop performance of FC-MPC is indistinguishable from that of cent-MPC. The performance of cent-MPC, decent-MPC and FC-MPC for control of the extractor is shown in Figure 3.

Closed-loop stability (Unstable decentralized modes). The real Schur decomposition for each $A_i, i \in \{1, M\}$ gives

$$A_i = \begin{bmatrix} U_{s_i} & U_{u_i} \end{bmatrix} \begin{bmatrix} A_{s_i} & (\cdot)A_i \\ & A_{u_i} \end{bmatrix} \begin{bmatrix} U_{s_i}{}' \\ U_{u_i}{}' \end{bmatrix},$$

in which A_{s_i} and A_{u_i} represent the stable and unstable eigenvalue blocks respectively. Since the decentralized model is minimal, (A_{ii}, B_{ii}) is controllable. The control horizon $N \geq r_i$, the number of unstable modes, $\forall \ i \in \{1, M\}$. An

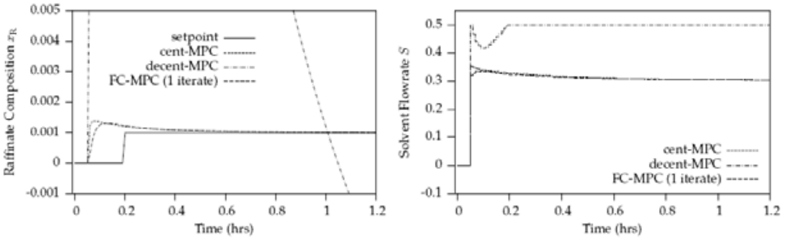


Fig. 3. SFE process (Extractor). Closed-loop performance of different plantwide control formulations.

additional terminal constraint $U_{u_i}{}' x_i(k+N|k) = 0$, that forces the unstable decentralized modes to the origin at the end of the control horizon, is enforced in the FC-MPC optimization problem (7). It is assumed that $x(k) \in \mathcal{X}_{\mathrm{N}}$, the N-step constrained stabilizable set for the system. For initialization, a feasible input trajectory for each subsystem $i \in \{1, M\}$ can be computed by solving a linear program. If $Q_i > 0, R_i > 0$ and the terminal penalty is chosen to be $\overline{Q}_i = U_{s_i} t_{\mathrm{lower}} Q_i U_{s_i}{}'$, in which $t_{\mathrm{lower}} Q_i$ is the solution to the Lyapunov equation $A_{s_i}{}' t_{\mathrm{lower}} Q_i A_{s_i} - t_{\mathrm{lower}} Q_i = -U_{s_i}{}' Q_i U_{s_i}$, $\forall\, i \in \{1, M\}$, then the closed-loop system $x(k+1) = Ax(k) + Bu(x(k))$ is exponentially stable under the distributed control law defined by (8) for $x(k) \in \mathcal{X}_{\mathrm{N}}$ and all $p(k) > 0$.

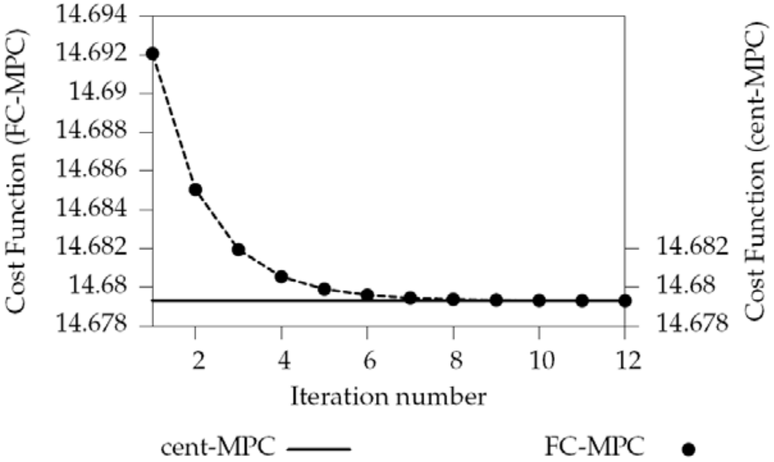


Fig. 4. Behavior of the cooperation-based cost function with iteration number at $k = 6$

Example. We consider a 5 input (m), 5 output (y), 30 state (n) plant with 3 subsystems. Each subsystem has 1 unstable decentralized mode. Subsystem 1 is represented by a CM consisting of 2 inputs, 2 outputs and 10 states. The CMs for subsystems 2, 3 have $m = 2, y = 2, n = 14$ and $m = 1, y = 1, n = 6$.

respectively. The performance of FC-MPC terminated after 1 iterate is within 5% of centralized MPC performance. Figure 4 shows the behavior of the cooperation-based cost function with iteration number at $k = 6$. The FC-MPC algorithm converges to the centralized MPC solution after 9 cooperation-based iterates.

5.2 Output Feedback

It is assumed that a steady-state Kalman filter, utilizing the following model, is employed to estimate the subsystem model states from local measurements.

$$x_i(k+1) = A_i x_i(k) + B_i u_i(k) + \sum_{j \neq i} W_{ij} u_j + w_i(k), \tag{10a}$$

$$y_i = C_i x_i(k) + \nu_i(k), \tag{10b}$$

where $w_i(k)$ and $\nu_i(k)$ are zero-mean, normally distributed disturbances with covariances Q_{x_i} and R_{v_i} respectively. In the presence of nonzero mean disturbances, suitable disturbance models may be used to remove steady-state offset. Disturbance models that realize off-set free performance under decentralized MPC can be employed in the FC-MPC framework to achieve zero off-set steady-state behavior.

Distributed MPC control law. At time k, define the open-loop trajectory obtained after s cooperation-based iterates as

$$\boldsymbol{u}_i^s(\widehat{x}(k)) = \left[u_i^s(\widehat{x}(k), k)', u_i^s(\widehat{x}(k), k+1)', \ldots\right]' \quad, \forall\ i\ \in\ \{1, M\}. \tag{11}$$

The distributed MPC control law under output feedback is defined as $u_i(k) = u_i^s(k|k) \equiv u_i^s(\widehat{x}(k), k), \forall\ i \in \{1, M\}$.

Closed-loop stability (Stable decentralized modes). We assume that a stable, steady-state Kalman filter exists for each subsystem[2]. If $Q_i > 0, R_i > 0$, $\forall\ i\ \in\ \{1, M\}$, the closed-loop system $x(k+1) = Ax(k) + Bu(\widehat{x}(k))$, in which $u(\widehat{x}(k)) = [u_1^{p(k)}(\widehat{x}(k), k)', u_2^{p(k)}(\widehat{x}(k), k)', \ldots, u_M^{p(k)}(\widehat{x}(k), k)']'$ is exponentially stable under the distributed control law defined by (11) for $\widehat{x}(k) \in \mathcal{X}$ and all $p(k) > 0$.

Distillation column ([12]). The performance of the FC-MPC formulation terminated after 1 cooperation-based iterate is shown in Figure 2. Unlike comm-MPC, which destabilizes the system, the controller derived by terminating the FC-MPC algorithm after just 1 iterate stabilizes the closed-loop system.

Closed-loop stability (Unstable decentralized modes). We know (A_{ii}, B_{ii}) is controllable and $N \geq r_i$ for each $i \in \{1, M\}$. With slight abuse of notation, define $\mathcal{X}_{\mathrm{e,N}}^i(k) \equiv \mathcal{X}_{\mathrm{e,N}}^i(\widehat{x}_i(k))$ to be the set of subsystem state estimate errors $e_i(k)$ for which there exists a perturbed input sequence

[2] Detectability of (A_i, C_i) guarantees the existence of a stable, steady-state Kalman filter.

$\{u_i(k+j|k+1) = u_i^{p(k)}(k+j|k) + v_i(k+j|k+1) \in \Omega_i\}_{j=1}^{N-1}$ at time $k+1$ such that $U_{u_i}{}'\widehat{x}_i(k+N|k+1) = 0$. As in the state feedback case, a terminal constraint $U_{u_i}{}'\widehat{x}_i(k+N|k) = 0$ is enforced in the FC-MPC optimization problem (7). We define $e(k) = [e_1(k)', e_2(k)', \ldots, e_M(k)']'$ and use the notation $\mathcal{X}_{\mathrm{e,N}} \equiv \mathcal{X}_{\mathrm{e,N}}(\widehat{x})$ to denote $\mathcal{X}_{\mathrm{e,N}}^1 \times \mathcal{X}_{\mathrm{e,N}}^2 \times \ldots \mathcal{X}_{\mathrm{e,N}}^M$. The domain of attraction for the closed-loop system is the set $\Gamma_N \triangleq \{(\widehat{x}(k), e(k)) \mid \widehat{x}(s) \in \mathcal{X}_N$ and $e(s) \in \mathcal{X}_{\mathrm{e,N}} \ \forall\ s \geq k\}$. If $Q_i > 0, R_i > 0, \forall\ i \in \{1, M\}$, the closed-loop system $x(k+1) = Ax(k) + Bu(\widehat{x}(k))$, in which $u(\widehat{x}(k)) = [u_1^{p(k)}(\widehat{x}(k),k)', u_2^{p(k)}(\widehat{x}(k),k)', \ldots, u_M^{p(k)}(\widehat{x}(k),k)']'$ is exponentially stable under the distributed control law defined by (11) for all $(\widehat{x}(k), e(k)) \in \Gamma_N$ and all $p(k) > 0$.

6 Partial Feasible Cooperation-Based MPC (pFC-MPC)

In the the FC-MPC framework, the objective of each local MPC is known to all interconnected subsystem MPCs. This global sharing of objectives may not be desirable in some situations. As a simple example, consider the system depicted in Figure 5. Assume that the y_2 setpoint is unreachable and that u_2 is at its bound constraint. From a practitioner's standpoint, it is desirable to manipulate input u_1, to the largest extent possible, to achieve all future y_1 setpoint changes. Conversely, it is desirable to manipulate u_2 to track setpoint changes in y_2. By definition, a decentralized control structure is geared to realize this operational objective. However, the resulting closed-loop performance may be quite poor. Centralized control, on the other hand, utilizes an optimal combination of the inputs u_1, u_2 to achieve the new setpoint. The centralized MPC framework, though optimal, may manipulate both u_1 and u_2 significantly.

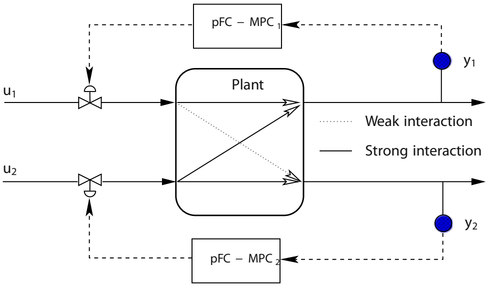


Fig. 5. 2×2 interacting system. Effect of input u_1 on output y_2 is small compared to $u_1 - y_1, u_2 - y_1$ and $u_2 - y_2$ interactions.

To track the setpoint of y_1 exclusively with input u_1 and setpoint of y_2 primarily with u_2, the concept of partial cooperation is employed. This approach of designing controllers to explicitly handle operational objectives is similar in philosophy to the modular multivariable controller (MMC) approach of [11]. The

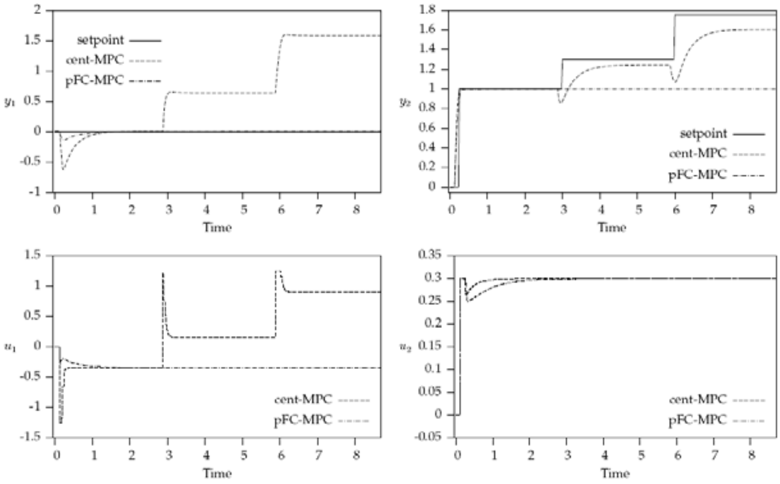


Fig. 6. Closed-loop performance of pFC-MPC and cent-MPC for the system in Figure 5

principal goal is to meet operational objectives, even if the resulting controller performance is not optimal. The partial cooperation-based MPC for subsystem 1 (pFC− MPC_1) manipulates u_1 but has access only to the local objective ϕ_1(see (6)) that quantifies the cost of control action u_1 on y_1. The partial cooperation-based MPC for subsystem 2 ($\text{pFC} - \text{MPC}_2$) manipulates u_2 and retains access to both subsystem objectives ϕ_1 and ϕ_2. Therefore, $\text{pFC} - \text{MPC}_2$ evaluates the cost of control action u_2 on a global level *i.e.*, its effect on both system outputs y_1 and y_2.

We consider an example in which an initial (reachable) setpoint change is made to y_2. Tracking errors for output y_2 are weighted 50 times more than tracking errors for output y_1. At times 3 and 6, unreachable y_2 setpoint changes are made. For each of the new y_2 setpoints, the input u_2 is at its upper bound at steady state. The pFC-MPC algorithm is terminated after 1 iterate. The closed-loop performance of cent-MPC and pFC-MPC are shown in Figure 6. Cent-MPC, in violation of the desired mode of operation, manipulates input u_1 (in addition to u_2) to track the y_2 target optimally. Since the y_1 setpoint is unchanged and $\text{pFC} - \text{MPC}_1$ has access only to objective ϕ_1, u_1 remains unaltered. To control y_2, $\text{pFC} - \text{MPC}_2$ can only manipulate u_2. However, u_2 is already at its bound constraint and consequently, y_2 is unchanged. Thus, the $\text{pFC} - \text{MPC}$ formulation, though suboptimal, achieves the desired operational objectives.

7 Conclusions

An iterative, cooperation-based MPC algorithm for integrating the different subsystem MPCs was described. The proposed algorithm achieves optimal systemwide control performance (centralized control) at convergence. Further, all intermediate iterates are feasible and the resulting distributed control law guarantees closed-loop stability under state and output feedback. Several examples were presented to illustrate feasibility, nominal stability and performance performance properties of the proposed distributed MPC framework.

Acknowledgment

The authors gratefully acknowledge the financial support of the industrial members of the Texas-Wisconsin Modeling and Control Consortium, and NSF through grant #CTS-0456694.

References

[1] J.G.V. Antwerp and R.D. Braatz. Model predictive control of large scale processes. *J. Proc. Control*, 10:1–8, 2000.

[2] Eduardo Camponogara, Dong Jia, Bruce H. Krogh, and Sarosh Talukdar. Distributed model predictive control. *IEEE Ctl. Sys. Mag.*, pages 44–52, February 2002.

[3] H. Cui and E.W. Jacobsen. Performance limitations in decentralized control. *J. Proc. Control*, 12:485–494, 2002.

[4] V. Havlena and J. Lu. A distributed automation framework for plant-wide control, optimisation, scheduling and planning. In *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, July 2005.

[5] Yu-Chi Ho. On Centralized Optimal Control. *IEEE Trans. Auto. Cont.*, 50(4):537–538, 2005.

[6] Dong Jia and Bruce H. Krogh. Distributed model predictive control. In *Proceedings of the American Control Conference*, Arlington, Virginia, June 2001.

[7] Rudolf Kulhavý, Joseph Lu, and Tariq Samad. Emerging technologies for enterprise optimization in the process industries. In James B. Rawlings, Babatunde A. Ogunnaike, and John W. Eaton, editors, *Chemical Process Control–VI: Sixth International Conference on Chemical Process Control*, pages 352–363, Tucson, Arizona, January 2001. AIChE Symposium Series, Volume 98, Number 326.

[8] Truls Larsson and Sigurd Skogestad. Plantwide control-a review and a new design procedure. *Mod. Ident. Control*, 21(4):209–240, 2000.

[9] J.Z. Lu. Challenging control problems and emerging technologies in enterprise optimization. *Control Eng. Prac.*, 11(8):847–858, August 2003.

[10] J. Lunze. *Feedback Control of Large Scale Systems.* Prentice-Hall, London, U.K, 1992.

[11] T.A. Meadowcroft, G. Stephanopoulos, and C. Brosilow. The Modular Multivariable Controller: 1: Steady-state properties. *AIChE J.*, 38(8):1254–1278, 1992.

[12] Babatunde A. Ogunnaike and W. Harmon Ray. *Process Dynamics, Modeling, and Control.* Oxford University Press, New York, 1994.

[13] S. Joe Qin and Thomas A. Badgwell. A survey of industrial model predictive control technology. *Control Eng. Prac.*, 11(7):733–764, 2003.

[14] B. Ramachandran, J.B. Riggs, H.R. Heichelheim, A.F. Seibert, and J.R. Fair. Dynamic simulation of a supercritical fluid extraction process. *Ind. Eng. Chem. Res.*, 31:281–290, 1992.

[15] Yudi Samyudia, Peter L. Lee, Ian T. Cameron, and Michael Green. Control strategies for a supercritical fluid extraction process. *Chem. Eng. Sci.*, 51(5):769–787, 1996.

[16] Nils R. Sandell-Jr., Pravin Varaiya, Michael Athans, and Michael Safonov. Survey of decentralized control methods for larger scale systems. *IEEE Trans. Auto. Cont.*, 23(2):108–128, 1978.

[17] D.D. Siljak. *Decentralized Control of Complex Systems*. Academic Press, London, 1991.

[18] S. Skogestad and M. Morari. Implications of large RGA elements on control performance. *Ind. Eng. Chem. Res.*, 26:2323–2330, 1987.

[19] Aswin N. Venkat, James B. Rawlings, and Stephen J. Wright. Stability and optimality of distributed model predictive control. In *Proceedings of the Joint 44th IEEE Conference on Decision and Control and European Control Conference*, Seville, Spain, December 2005.

[20] Robert E. Young, R. Donald Bartusiak, and Robert W. Fontaine. Evolution of an industrial nonlinear model predictive controller. In James B. Rawlings, Babatunde A. Ogunnaike, and John W. Eaton, editors, *Chemical Process Control–VI: Sixth International Conference on Chemical Process Control*, pages 342–351, Tucson, Arizona, January 2001. AIChE Symposium Series, Volume 98, Number 326.

[21] G. Zhu and M.A. Henson. Model predictive control of interconnected linear and nonlinear processes. *Ind. Eng. Chem. Res.*, 41:801–816, 2002.

Distributed MPC for Dynamic Supply Chain Management

William B. Dunbar[1] and S. Desa[2]

[1] Computer Engineering, Baskin School of Engineering, University of California, Santa Cruz, 95064, USA
dunbar@soe.ucsc.edu

[2] Technology and Information Management, Baskin School of Engineering, University of California, Santa Cruz, 95064, USA

Summary. The purpose of this paper is to demonstrate the application of a recently developed theory for distributed nonlinear model predictive control (NMPC) to a promising domain for NMPC: dynamic management of supply chain networks. Recent work by the first author provides a distributed implementation of NMPC for application in large scale systems comprised of cooperative dynamic subsystems. By the implementation, each subsystem optimizes locally for its own policy, and communicates the most recent policy to those subsystems to which it is coupled. Stabilization and feasibility are guaranteed for arbitrary interconnection topologies, provided each subsystem not deviate too far from the previous policy, consistent with traditional MPC move suppression penalties. In this paper, we demonstrate the scalability and performance of the distributed implementation in a supply chain simulation example, where stages in the chain update in parallel and in the presence of cycles in the interconnection network topology. Using anticipative action, the implementation shows improved performance when compared to a nominal management policy that is derived in the supply chain literature and verified by real supply chain data.

1 Introduction

A supply chain can be defined as the interconnection and evolution of a demand network. Example subsystems, referred to as stages, include raw materials, distributors of the raw materials, manufacturers, distributors of the manufactured products, retailers, and customers. Between interconnected stages, there are two types of process flows: 1) information flows, such as an order requesting goods, and 2) material flows, i.e., the actual shipment of goods. Key elements to an efficient supply chain are accurate pinpointing of process flows and timing of supply needs at each stage, both of which enable stages to request items as they are needed, thereby reducing safety stock levels to free space and capital [3]. Recently, Braun *et al.* [2] demonstrated the effectiveness of model predictive control (MPC) in realizing these elements for management of a dynamic semiconductor chain, citing benefits over traditional approaches and robustness to model and demand forecast uncertainties. In this context, the chain is isolated from competition, and so a cooperative approach is appropriate. Limitations of their approach are that it requires acyclic interconnection network topologies,

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 607–615, 2007.
springerlink.com

and sequential updates from downstream to upstream stages. Realistic supply chains contain cycles in the interconnection network, and generally do not operate sequentially, i.e., stages typically update their policies in parallel, often asynchronously. To be effective in the general case, a distributed MPC approach should demonstrate scalability (stages are locally managed), stability, permit parallel updates, and allow for cycles in the interconnection network topology. The purpose of this paper is to demonstrate the application of a recently developed distributed implementation of nonlinear MPC (NMPC) [4, 5] to the problem of dynamic management of supply chain networks. By this implementation, each subsystem optimizes locally for its own policy, and communicates the most recent policy to those subsystems to which it is coupled. Stabilization is guaranteed for arbitrary interconnection topologies (permitting cycles), provided each subsystem not deviate too far from the previous policy. A contribution of this paper is to demonstrate the relevance and efficacy of the distributed NMPC approach in the venue of supply chain management.

2 Problem Description

A supply chain consists of all the stages involved in fulfilling a customer request [3]. A three stage supply chain network consisting of a supplier S, a manufacturer M, and a retailer R is shown in Figure 1, and will be the focus of this paper. Dell employs a "build-to-order" management strategy that is based on a version of the chain in Figure 1, where R is the customer, M is Dell, S is a chip supplier [3]. Each variable shown has a superscript denoting the corresponding stage it is associated with. The classic MIT "Beer Game" [7] is used as an example three stage supply chain. In the beer game, the supplier S may be thought of as the supplier of bottles to the manufacturer M, who brews and "bottles" the beer, and then ships it to the retailer R for sale to customers. The supply chain is therefore driven by customer demand (number of cases sold per day), which then triggers a series of information flows and material flows. The *information flows* are assumed to have negligible time delays, and are represented by the three left pointing arrows in Figure 1. The *material flows* are assumed to have shipment delays, and are represented by the arrows that pass through blocks

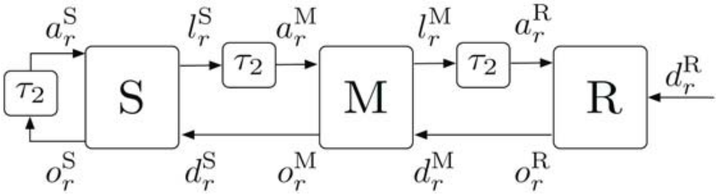


Fig. 1. Block diagram of a three stage supply chain comprised of a supplier S, a manufacturer M, and a retailer R

labeled τ_2, where τ_2 is a constant representing the amount of delay in days to move the goods. In the case of the supplier, the outgoing information flow (o_r^{S}) is converted through fabrication into materials, and this conversion process is modeled as a simple delay. Since material flows downstream, we say that R is *downstream* from M (likewise, M is downstream from S), while M is *upstream* from R (likewise, S is upstream from M). The customer can be thought of as a stage downstream (not shown) from R in our model.

Each stage $x \in \{\text{S,M,R}\}$ in Figure 1 is characterized by 3 state variables, defined as follows. The stock level s^x is the number of items currently available in stage x for shipment to the downstream stage. The unfulfilled order of stock o_u^x is the number of items that stage x has yet to receive from the upstream stage. The backlog of stock b^x is the number of committed items that stage x has yet to ship to the downstream stage. The exogenous *inputs* (assumed measureable) are the demand rate d_r^x, defined as the number of items per day ordered by the downstream stage, and the acquisition rate a_r^x, defined as the number of items per day acquired from the upstream stage. The *outputs* are the order rate o_r^x, defined as the number of items per day ordered from the upstream stage, and the shipment rate l_r^x, defined as the number of items per day shipped to the downstream stage. The order rate is the *decision variable* (control). By our notation, all rate variables are denoted by an r subscript. The model, state and control constraints for any stage $x \in \{\text{S,M,R}\}$ are

$$\left.\begin{aligned} \dot{s}^x(t) &= a_r^x(t) - l_r^x(t) \\ \dot{o}_u^x(t) &= o_r^x(t) - a_r^x(t) \\ \dot{b}^x(t) &= d_r^x(t) - l_r^x(t) \end{aligned}\right\}, \quad t \geq 0, \tag{1}$$

$$\text{subject to} \quad \left.\begin{aligned} 0 \leq (s^x(t), o_u^x(t), b^x(t)) &\leq \mathrm{s}_{\max} \\ 0 \leq o_r^x(t) \qquad\qquad\quad &\leq o_{r,\max} \end{aligned}\right\}, \quad t \geq 0, \tag{2}$$

where $l_r^x(t) = d_r^x(t - \tau_1) + b^x(t)/t_b$. The dynamics of the supply chain in the present work arise either from rates of accumulation, or from one of two types of material flow delay (see [7], Chapter 11). Equation (1) describes the first-order dynamics for stock, unfulfilled orders, and backlog, each arising from rates of accumulation. The constraints on the state and control in (2) reflect that stock, unfulfilled order and backlog are independently bounded from below by zero and from above by a common constant $\mathrm{s}_{\max}$, and that the control (order rate) is non-negative and bounded by the positive constant $o_{r,\max}$. The objective of supply chain management is to minimize total costs, which includes avoiding backlog (keep near zero) and keeping unfulfilled orders and stock near desired (typically low) levels [7]. Specifically, the control objective for each stage is $(s^x(t), o_u^x(t)) \rightarrow (s_d, o_{ud}^x(t))$, where s_d is a constant desired stock (common to every stage) and $o_{ud}^x(t) = t_l l_r^x(t)$ is the desired unfulfilled order. The flow constant t_l represents the lead time from the downstream stage. Note that if the demand rate converges to a steady value $d_r^x(t) \rightarrow d_r$, then backlog will converge to zero, the shipment rate converges $l_r^x(t) \rightarrow d_r$, and the desired unfulfilled order becomes the constant $o_{ud}^x = t_l d_r$. For each stage $x \in \{\text{S,M,R}\}$, the acquisition rate $a_r^x(t)$ and the demand rate $d_r^x(t)$ are defined as follows: $\underline{\text{S:}}$ $a_r^{\mathrm{S}}(t) = o_r^{\mathrm{S}}(t-\tau_2)$, $d_r^{\mathrm{S}}(t) = o_r^{\mathrm{M}}(t)$; $\underline{\text{M:}}$ $a_r^{\mathrm{M}}(t) = l_r^{\mathrm{S}}(t-\tau_2)$, $d_r^{\mathrm{M}}(t) = o_r^{\mathrm{R}}(t)$; and $\underline{\text{R:}}$ $a_r^{\mathrm{R}}(t) = l_r^{\mathrm{M}}(t-\tau_2)$. The demand rate at the retailer $d_r^{\mathrm{R}}(t)$ is an input defined as the current/projected customer demand. After substitutions, we have the following models for each of the three stages. For the supplier stage,

$$\left.\begin{aligned} \dot{s}^{\mathrm{S}}(t) &= o_r^{\mathrm{S}}(t-\tau_2) - o_r^{\mathrm{M}}(t-\tau_1) - b^{\mathrm{S}}(t)/t_b \\ \dot{o}_u^{\mathrm{S}}(t) &= o_r^{\mathrm{S}}(t) - o_r^{\mathrm{S}}(t-\tau_2) \\ \dot{b}^{\mathrm{S}}(t) &= o_r^{\mathrm{M}}(t) - o_r^{\mathrm{M}}(t-\tau_1) - b^{\mathrm{S}}(t)/t_b \end{aligned}\right\}. \tag{3}$$

For the manufacturer stage,

$$\left.\begin{aligned} \dot{s}^{\mathrm{M}}(t) &= o_r^{\mathrm{M}}(t-\tau_1-\tau_2) + b^{\mathrm{S}}(t-\tau_2)/t_b - o_r^{\mathrm{R}}(t-\tau_1) - b^{\mathrm{M}}(t)/t_b \\ \dot{o}_u^{\mathrm{M}}(t) &= o_r^{\mathrm{M}}(t) - o_r^{\mathrm{M}}(t-\tau_1-\tau_2) - b^{\mathrm{S}}(t-\tau_2) \\ \dot{b}^{\mathrm{M}}(t) &= o_r^{\mathrm{R}}(t) - o_r^{\mathrm{R}}(t-\tau_1) - b^{\mathrm{M}}(t)/t_b \end{aligned}\right\}. \tag{4}$$

For the retailer stage,

$$\left.\begin{aligned} \dot{s}^{\mathrm{R}}(t) &= o_r^{\mathrm{R}}(t-\tau_1-\tau_2) + b^{\mathrm{M}}(t-\tau_2)/t_b - d_r^{\mathrm{R}}(t-\tau_1) - b^{\mathrm{R}}(t)/t_b \\ \dot{o}_u^{\mathrm{R}}(t) &= o_r^{\mathrm{R}}(t) - o_r^{\mathrm{R}}(t-\tau_1-\tau_2) - b^{\mathrm{M}}(t-\tau_2)/t_b \\ \dot{b}^{\mathrm{R}}(t) &= d_r^{\mathrm{R}}(t) - d_r^{\mathrm{R}}(t-\tau_1) - b^{\mathrm{R}}(t)/t_b \end{aligned}\right\}. \tag{5}$$

We say that two stages have bidirectional coupling if the differential equation models of both stages depend upon the state and/or input of the other stage. Equations (3)–(5) demonstrate the dynamic bidirectional coupling between stages S and M, and stages M and R. Due to the bidirectional coupling, there are two cycles of information dependence present in this chain. Cycle one: the model (3) for S requires the order rate o_r^{M} from M, and the model (4) for M requires the backlog b^{S} from S. Cycle two: the model (4) for M requires the order rate o_r^{R} from R, and the model (5) for R requires the backlog b^{M} from M. Cycles complicate decentralized/distributed MPC implementations, since at any MPC update, coupled stages in each cycle must *assume* predictions for the states/inputs of one another. Such predictions are different in general than the *actual* locally computed predictions for those states/inputs. When cycles are not present, life is easier, as the stages can update sequentially, i.e., stages update in order from downstream to upstream, and the actual predictions from downstream stages can be transmitted to upstream stages at each update. In accordance with the MPC approach, the first portion of these actual predictions is implemented by each stage. Thus, the absence of cycles implies that stages can transmit policies that will be implemented. The sequential update approach is taken by Braun *et al.* [2], whose supply chain example contains no cycles. When cycles are present, on the other hand, actual predictions are not mutually available. Thus, some predictions must be assumed, incurring an unavoidable discrepancy between what a stage will do and what coupled stages assume it will do. One way to address this issue is to assume that the other stages react worst case, i.e., as bounded contracting disturbances, as done first by Jia and

Krogh [6]. The implementation employed here address the cycle issue in another way [4, 5]. Coupled stages receive the *previously* computed predictions from one another prior to each update, and rely on the remainder of these predictions as the assumed prediction at each update. To bound the unavoidable discrepancy between assumed and actual predictions, each stage includes a local move suppression penalty on the deviation between the current (actual) prediction and the remainder of the previous prediction.

3 Control Approaches

The nominal feedback policy, derived in [7], is given by

$$o_r^x(t) = l_r^x(t) + k_1[s_d - s^x(t)] + k_2[o_{ud}^x(t) - o_u^x(t)], \quad k_1, k_2 \in (0, \infty).$$

In the simulations in Section 4, the state and control constraints (2) are enforced by using saturation functions. The nominal control is decentralized in that the feedback for each stage depends only on the states of that stage. Simulation-based analysis and comparisons with real data from actual supply chains is presented as a justification for this choice of control in [7].

For the distributed MPC approach, the continuous time models are first discretized, using the discrete time samples $t_k = k * \delta$, with $\delta = 0.2$ days as the sample period, and $k \in \mathbb{N} = \{0, 1, 2, ...\}$. The prediction horizon is $T_p = P * \delta$ days, with $P = 75$, and the control horizon is $T_m = M * \delta$ days, with $M = 10$. For all three stages, the stock s^x and unfulfilled order o_u^x models are included in the MPC optimization problem. The backlog b^x, on the other hand, is not included in the optimization problem, as it is uncontrollable. Instead, the backlog is computed locally at each stage using the discretized model, the appropriate exogenous inputs that the model requires, and the saturation constraint in (2). For update time t_k, the *actual* locally predicted stock defined at times $\{t_k, ..., t_{k+P}\}$ is denoted $\{s^x(t_k; t_k), ..., s^x(t_{k+P}; t_k)\}$, using likewise notation for all other variables. The *true* stock at any time t_k is simply denoted $s^x(t_k)$, and so $s^x(t_k) = s^x(t_k; t_k)$, again using likewise notation for all other variables. In line with the notational framework in the MATLAB MPC toolbox manual [1], the set of measurable inputs are termed measured disturbances (MDs). By our distributed MPC algorithm, the MDs are *assumed* predictions. The set of MDs for each stage $x \in \{\text{S,M,R}\}$ is denoted $\mathcal{D}^x(t_k)$, associated with any update time t_k. The MDs for the three stages are $\mathcal{D}^{\text{S}}(t_k) = \{\mathbf{b}_{\text{as}}^{\text{S}}(k), \mathbf{o}_{r,\text{as}}^{\text{M}}(k)\}$, $\mathcal{D}^{\text{M}} = \{\mathbf{b}_{\text{as}}^{\text{M}}(k), \mathbf{b}_{\text{as}}^{\text{S}}(k), \mathbf{o}_{r,\text{as}}^{\text{R}}(k)\}$ and $\mathcal{D}^{\text{R}} = \{\mathbf{b}_{\text{as}}^{\text{R}}(k), \mathbf{b}_{\text{as}}^{\text{M}}(k), d_r^{\text{R}}\}$, where $\mathbf{o}_{r,\text{as}}^x(k) = \{o_{r,\text{as}}^x(t_k; t_k), ..., o_{r,\text{as}}^x(t_{k+P}; t_k)\}$ and $\mathbf{b}_{r,\text{as}}^x(k)$ is defined similarly using the assumed predicted backlog. The $(\cdot)_{\text{as}}$ subscript notation refers to the fact that, except for the demand rate at the retailer d_r^{R}, all of the MDs contain *assumed* predictions for each of the associated variables. It is presumed at the outset that a customer demand $d_r^{\text{R}}(\cdot) : [0, \infty) \to \mathbb{R}$ is known well into the future and without error. As this is a strong assumption, we are considering stochastic demand rates in our more recent work. Although it is locally computed, each stage's backlog is treated as an MD since it relies on the assumed demand rate prediction from the downstream stage. Note that the initial backlog is always the true backlog, i.e., $b^x_{r,\mathrm{as}}(t_k;t_k) = b^x(t_k)$ for each stage x and at any update time t_k. Let the set $\mathcal{X}^x(t_k) = \{s_d, o^x_{ud}(t_k;t_k), ..., o^x_{ud}(t_{k+P};t_k)\}$ denote the desired states associated with stage x and update time t_k. Using the equations from the previous section, the desired unfulfilled order prediction $o^x_{ud}(\cdot;t_k)$ in $\mathcal{X}^x(t_k)$ can be computed locally for each stage x given the MDs $\mathcal{D}^x(t_k)$. By our distributed MPC implementation, stages update their control in parallel at each update time t_k. The optimal control problem and distributed MPC algorithm for any stage are defined as follows.

Problem 1. For any stage $x \in \{\text{S,M,R}\}$, and at any update time t_k, $k \in \mathbb{N}$:
Given: the current state $(s^x(t_k), o^x_u(t_k))$, the MDs $\mathcal{D}^x(t_k)$, the desired states $\mathcal{X}^x(t_k)$, the non-negative weighting constants $(W_s, W_{o_u}, W_u, W_{\delta u})$, and a non-negative target order rate o^{targ}_r,

Find: the optimal control $\mathbf{o}^x_{r,*}(k) \triangleq \{o^x_{r,*}(t_k;t_k), o^x_{r,*}(t_{k+1};t_k), ..., o^x_{r,*}(t_{k+M-1};t_k)\}$ satisfying

$$\mathbf{o}^x_{r,*}(k) = \arg\min \Bigg\{ \sum_{i=1}^{P} W_s \left[s^x(t_{k+i};t_k) - s_d\right]^2 + W_{o_u} \left[o^x_u(t_{k+i};t_k) - o^x_{ud}(t_{k+i};t_k)\right]^2$$
$$+ \sum_{j=0}^{M-1} W_u \left[o^x_r(t_{k+j};t_k) - o^{\mathrm{targ}}_r\right]^2 + W_{\delta u} \left[o^x_r(t_{k+j};t_k) - o^x_r(t_{k+j-1};t_k)\right]^2 \Bigg\},$$

where $o^x_r(t_{k-1};t_k) \triangleq o^x_{r,*}(t_{k-1};t_{k-1})$, subject to the discrete-time version of the appropriate model (equation (3), (4) or (5)), and the constraints in equation (2). ■

Algorithm 1. The distributed MPC law for any stage $x \in \{\text{S,M,R}\}$ is as follows:

Data: Current state: $(s^x(t_0), o^x_u(t_0), b^x(t_0))$. Parameters: δ, M, P, $(W_s, W_{o_u}, W_u, W_{\delta u})$, and o^{targ}_r.
Initialization: At initial time $t_0 = 0$, generate $\mathcal{D}^x(t_0)$ as follows: (a) Choose a nominal constant order rate $o^{x,\mathrm{nom}}_r$, set $o^x_{r,\mathrm{as}}(t_i;t_0) = o^{x,\mathrm{nom}}_r$, for $i = 0, ..., P$, and if $x =$ R or M, transmit $\mathbf{o}^x_{r,\mathrm{as}}(0)$ to M or S, respectively; (b) Compute $\mathbf{b}^x_{r,\mathrm{as}}(0)$, and if $x =$ S or M, transmit to M or R, respectively. Compute $\mathcal{X}^x(t_0)$ and solve Problem 1 for $\mathbf{o}^x_{r,*}(0)$.
Controller:

1. Between updates t_k and t_{k+1}, implement the current control action $o^x_{r,*}(t_k;t_k)$.
2. At update time t_{k+1}:
   a) Obtain $(s^x(t_{k+1}), o^x_u(t_{k+1}), b^x(t_{k+1}))$.
   b) Generate $\mathcal{D}^x(t_{k+1})$ as follows:
      i. Set $o^x_{r,\mathrm{as}}(t_{j+k+1};t_{k+1}) = o^x_{r,*}(t_{j+k+1};t_k)$, for $j = 0, ..., M-2$ and $o^x_{r,\mathrm{as}}(t_{j+k+1};t_{k+1}) = o^x_{r,*}(t_{k+M-1};t_k)$ for $i = M-1, ..., P$. If $x =$ R or M, transmit $\mathbf{o}^x_{r,\mathrm{as}}(k+1)$ to M or S, respectively.
      ii. Compute $\mathbf{b}^x_{r,\mathrm{as}}(k+1)$, and if $x =$ S or M, transmit to M or R, respectively.
   c) Compute $\mathcal{X}^x(t_{k+1})$ and solve Problem 1 for $\mathbf{o}^x_{r,*}(k+1)$.
3. Set $k = k+1$ and return to step 1. ■

By this algorithm, each stage initially computes an optimal order rate policy assuming neighboring stages employ a nominal constant order rate. For every subsequent update, each stage computes an optimal order rate policy, assuming that the MDs are based on the remainder of the previously computed policies computed of neighboring stages.

4 Numerical Experiments

The simulations were carried out in MATLAB 7.0, using Simulink 6.2 and the Model Predictive Control Toolbox 2.2. The nominal and distributed MPC approaches are compared on the full three stage problem, given a step increase and decrease in the customer demand rate at the retailer. For simulation purposes, we choose $d_r^{\mathrm{R}}(t) = 200$ cases/day for $t \in [0, \infty) \setminus [5, 15)$ and $d_r^{\mathrm{R}}(t) = 300$ for $t \in [5, 15)$. The response for the three stages under the nominal control policy ($k_1 = 1/15$, $k_2 = 1/30$) is shown in Figure 2. To implement the distributed MPC

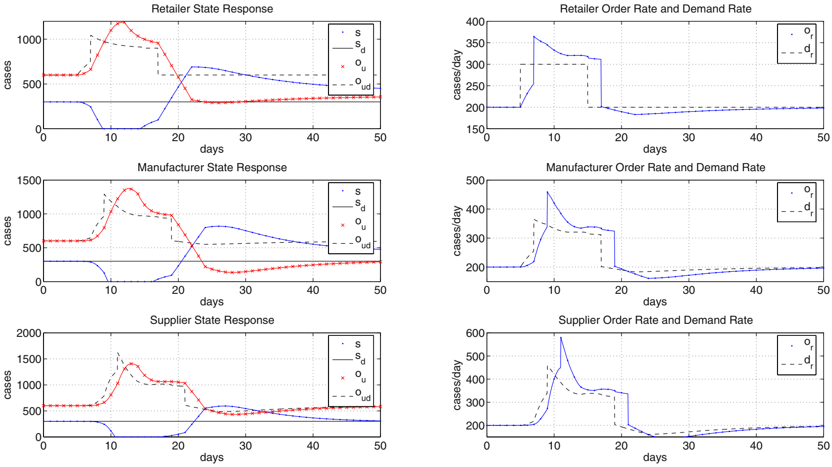


Fig. 2. Nominal response to step increase at 5 days and decrease at 15 days in retailer customer demand rate d_r^{R}

Algorithm 1, the anticipative action of the MPC Toolbox is employed so that each entire assumed prediction can be used. Recall that the assumed predictions are not the actual predictions, although the move suppression terms ($W_{\delta u}$ weighted) in the cost are used to ensure that these predictions are not too far apart. The forecasted demand rate at the retailer is also used with the anticipation option turned on. A more "apples-to-apples" comparison would be to incorporate internal models with the nominal approach that use the forecasted customer demand rate. The response for the three stages under the distributed MPC policy with anticipation is shown in Figure 3. The weights used in MPC for each stage are $(W_u, W_{\delta u}, W_s, W_{o_u}) = (1, 5, 5, 1)$. The stock and unfulfilled

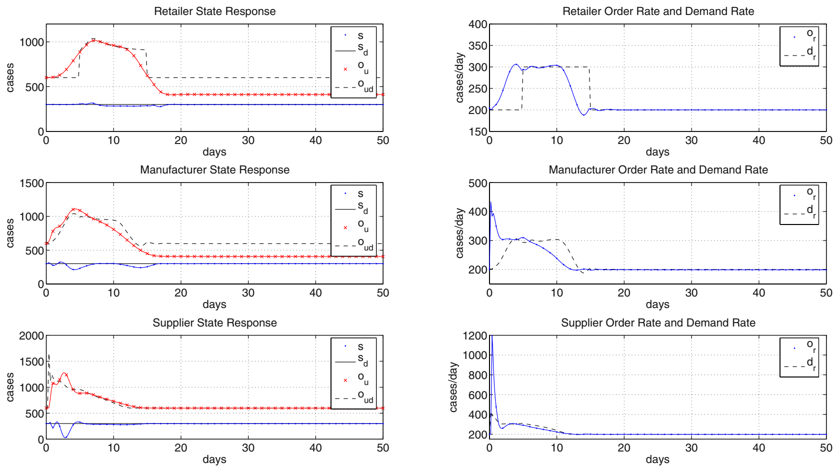


Fig. 3. Distributed MPC response to the same demand rate d_r^{R}. By using anticipation, the state responses are improved, and the order rates are smoother.

order state responses are an improvement over the nominal approach, both in terms of steady-state error and settling time. The nonzero steady-state error in the unfulfilled order and stock of stages M and R can be predicted by using system-type analysis. The well known "bullwhip effect" [3, 7] encountered in the coordination of a multi-stage supply chain is also seen in both figures, indicated by the increase in the maximum order rate excursion as one moves upstream from retailer to supplier.

5 Conclusions and Extensions

In this paper, a supply chain management problem was defined using the classic MIT "Beer Game" [7]. A nominal feedback policy, derived and experimentally validated in the supply chain literature, was then compared to a distributed MPC algorithm. The numerical experiments showed that the algorithm yielded improved performance over the nominal policy when the customer demand rate can be reliably forecasted. While one might redefine the nominal approach to include internal models that leverage forecasts, it is clear that MPC trivializes making use of forecasted inputs via anticipation, while respecting state and control constraints. As part of our on going work, we will consider a multi-echelon supply chain problem [3], in which at least two players operate within each stage. The decision problem becomes more complicated in these chains, since the update rates of different players in a stage are different in general, requiring an extension of the distributed MPC theory to asynchronous timing conditions. Additionally, we will consider stochastic (brownian) demand rate forecasts, and more realistic production models in the manufacturing stage.

References

[1] Bemporad, A., Morari, M. and Ricker, N. L., "Model Predictive Control Toolbox – for use with MATLAB," User's Guide, MathWorks, Inc. (2005).

[2] Braun, M. W., Rivera, D. E., Carlyle, W. M. and Kempf, K. G., "Application of Model Predictive Control to Robust Management of Multiechelon Demand Networks in Semiconductor Manufacturing," *Simulation*, **79** (3), pp. 139–156 (2003).

[3] Chopra, S. and Meindl, P., *Supply Chain Management, Strategy, Planning, and Operations*, Second Edition, Pearson Prentice-Hall, New Jersey (2004).

[4] Dunbar, W. B., "Distributed Receding Horizon Control of Dynamically Coupled Nonlinear Systems," Submitted to *IEEE Transaction on Automatic Control*, UCSC Baskin School of Engineering Technical Report ucsc-crl-06-03 (2006).

[5] Dunbar, W. B. and Murray, R. M., "Distributed Receding Horizon Control for Multi-Vehicle Formation Stabilization," *Automatica*, Vol. 2, No. 4 (2006).

[6] Jia, D. and Krogh, B. H., "Min-Max Feedback Model Predictive Control for Distributed Control With Communication," In *Proceedings of the IEEE American Control Conference*, Anchorage, AK (2002).

[7] Sterman, J. D., *Business Dynamics - Systems Thinking and Modelling in a Complex World*, McGraw-Hill, New York (2000).

Robust Model Predictive Control for Obstacle Avoidance: Discrete Time Case

Saša V. Raković and David Q. Mayne

Imperial College London, London SW7 2BT, United Kingdom
sasa.rakovic@imperial.ac.uk, d.mayne@imperial.ac.uk

1 Introduction

The importance of the obstacle avoidance problem is stressed in [4]. Computation of reachability sets for the obstacle avoidance problem is addressed, for continuous-time systems in [4, 5] and for discrete-time systems in [12]; further results appear in, for instance [2, 17, 18]. The obstacle avoidance problem is inherently non–convex. Most existing results are developed for the deterministic case when external disturbances are not present. The main purpose of this paper is to demonstrate that the obstacle avoidance problem in the discrete time setup has considerable structure even when disturbances are present. We extend the robust model predictive schemes using tubes *(sequences of sets of states)* [9, 11, 14] to address the robust obstacle avoidance problem and provide a mixed integer programming algorithm for robust control of constrained linear systems that are required to avoid specified obstacles. The resultant robust optimal control problem that is solved on–line has marginally increased complexity compared with that required for model predictive control for obstacle avoidance in the deterministic case.

This paper is organized as follows. Section 2 discusses the general idea of *robust control invariant tubes* for the obstacle avoidance problem. Section 3 considers in more detail the case when system being controlled is linear. Section 4 presents a simple tube controller, establishes its properties and provides an illustrative example. Finally, Section 5 gives conclusions and indicates further extensions.

Notation: Let $\mathbf{N} \triangleq \{0, 1, 2, \ldots\}$ and $\mathbf{N}_q \triangleq \{0, 1, \ldots, q\}$ for $q \in \mathbf{N}$. A *polyhedron* is the (convex) intersection of a finite number of open and/or closed half-spaces, a *polytope* is the closed and bounded polyhedron and a closed (open) *polygon* is the union of a finite number of polytopes (polyhedra). Given two sets $\mathcal{U} \subset \mathbf{R}^n$ and $\mathcal{V} \subset \mathbf{R}^n$, the Minkowski set addition is defined by $\mathcal{U} \oplus \mathcal{V} \triangleq \{u + v \mid u \in \mathcal{U},\ v \in \mathcal{V}\}$, the Minkowski/Pontryagin set difference is: $\mathcal{U} \ominus \mathcal{V} \triangleq \{x \mid x \oplus \mathcal{V} \subseteq \mathcal{U}\}$. The distance of a point z from a set X is denoted by $d(z, X) \triangleq \inf\{|z - x| \mid x \in X\}$.

2 Problem Formulation

We consider the following discrete-time, time-invariant system:

$$x^+ = f(x, u, w) \tag{1}$$

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 617–627, 2007.
springerlink.com

where $x \in \mathbf{R}^n$ is the current state, $u \in \mathbf{R}^n$ is the current control input and x^+ is the successor state and $f : \mathbf{R}^n \times \mathbf{R}^m \times \mathbf{R}^p \to \mathbf{R}^n$; the bounded disturbance w is known only to that extent that it belongs to the compact set $\mathbf{W} \subset \mathbf{R}^p$ that contains the origin in its interior. The system is subject to the following set of hard constraints:

$$(x, u, w) \in \mathbf{X} \times \mathbf{U} \times \mathbf{W} \tag{2}$$

where $\mathbf{X}$ and $\mathbf{U}$ are compact (closed and bounded) sets respectively, each containing the origin in its interior. Additionally it is required that the state trajectories avoid a predefined *open* set $\mathbf{O}$, generally specified as the union of a finite number of open sets, introducing an additional state constraint

$$x \notin \mathbf{O},\ \mathbf{O} \triangleq \bigcup_{j \in \mathbf{N}_q} \mathbf{O}_j, \tag{3}$$

The hard state constraints (2) and (3) can be converted into a single non–convex state constraint:

$$x \in \mathbf{X_O} \triangleq \mathbf{X} \setminus \mathbf{O} \tag{4}$$

Let $\mathbf{W} \triangleq \bar{\mathbf{W}}^N$ denote the class of admissible disturbance sequences $\mathbf{w} \triangleq \{w(i) \mid i \in \mathbf{N}_{N-1}\}$. Let $\phi(i; x, \pi, \mathbf{w})$ denote the solution at time i of (2) when the control policy is π, the disturbance sequence is $\mathbf{w}$ and the initial state is x at time 0; a policy π is a sequence of control laws, i.e. $\pi \triangleq \{\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$ where $\mu_i(\cdot)$ is the control law (mapping state to control) at time i.

Given a set $X \subset \mathbf{R}^n$ and a control law $\mu : X \to U$ where $U \subset \mathbf{R}^m$ we define:

$$X^+ \triangleq \mathcal{F}(X, \mu, \mathbf{W}),\ \mathcal{F}(X, \mu, \mathbf{W}) \triangleq \{f(x, \mu(x), w) \mid (x, w) \in X \times \mathbf{W}\} \tag{5}$$

$$U(X, \mu) \triangleq \{\mu(x) \mid x \in X\} \tag{6}$$

Robust model predictive control is defined, as usual, by specifying a finite-horizon robust optimal control problem that is solved on-line. In this paper, the robust optimal control problem is the determination of *an appropriate tube*, defined as a sequence $\mathbf{X} \triangleq \{X_0, X_1, \ldots, X_N\}$ of *sets of states*, and an associated control policy $\pi = \{\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$ that minimize an appropriately chosen cost function and satisfy the following set of constraints, for a given initial condition $x \in \mathbf{X_O}$, that generalize corresponding constraints in [6]:

$$x \in X_0, \tag{7}$$

$$X_i \subseteq \mathbf{X_O},\ \forall i \in \mathbf{N}_{N-1} \tag{8}$$

$$X_N \subseteq \mathbf{X}_f \subseteq \mathbf{T} \subseteq \mathbf{X_O}, \tag{9}$$

$$U(X_i, \mu_i) \subseteq \mathbf{U},\ \forall i \in \mathbf{N}_{N-1} \tag{10}$$

$$\mathcal{F}(X_i, \mu_i, \mathbf{W}) \subseteq X_{i+1},\ \forall i \in \mathbf{N}_{N-1} \tag{11}$$

where $\mathcal{F}(X_i, \mu_i, \mathbf{W})$ and $U(X_i, \mu_i)$ are defined, respectively, by (5) and (6), $\mathbf{T} \subseteq \mathbf{X_O}$ and $\mathbf{X}_f \subseteq \mathbf{T}$ are a target set and its appropriate subset. *It is assumed that* $\mathbf{X_O} \neq \emptyset$ *and moreover that the set* $\mathbf{T}$ *is assumed to be compact (i.e. closed and bounded) and convex set containing the origin in its interior*. The relevance of the constraints (7)– (11) is shown by the following result [6, 11]:

Proposition 1 (Robust Constraint Satisfaction). *Suppose that the tube* $\mathbf{X}$ *and the associated policy* π *satisfy the constraints* (7)–(11)*. Then the state of the controlled system satisfies* $\phi(i; x, \pi, \mathbf{w}) \in X_i \subseteq \mathbf{X_O}$ *for all* $i \in \mathbf{N}_{N-1}$*, the control satisfies* $\mu_i(\phi(i; x, \pi, \mathbf{w})) \in U(X_i, \mu_i) \subseteq \mathbf{U}$ *for all* $i \in \mathbf{N}_{N-1}$*, and the terminal state satisfies* $\phi(N; x, \pi, \mathbf{w}) \in \mathbf{X}_f \subseteq \mathbf{T} \subseteq \mathbf{X_O}$ *for every initial state* $x \in X_0$ *and every admissible disturbance sequence* $\mathbf{w} \in \bar{\mathbf{W}}$*.*

Let $\theta \triangleq \{\mathbf{X}, \pi\}$ and let, for a given state $x \in \mathbf{X_O}$, $\Theta(x)$ (the set of admissible θ) be defined by:

$$\begin{aligned}\Theta(x) \triangleq \{&\theta \mid x \in X_0,\ X_i \subseteq \mathbf{X_O},\ U(X_i, \mu_i) \subseteq \mathbf{U},\\ &\mathcal{F}(X_i, \mu_i, \mathbf{W}) \subseteq X_{i+1},\quad \forall i \in \mathbf{N}_{N-1},\ X_N \subseteq \mathbf{X}_f \subseteq \mathbf{T}\}\end{aligned} \tag{12}$$

Consider the cost function defined by:

$$V_N(x, \theta) \triangleq \sum_{i=0}^{N-1} \ell(X_i, \mu_i(\cdot)) + V_f(X_N) \tag{13}$$

where $\ell(\cdot)$ is path cost and $V_f(\cdot)$ is terminal cost and consider the following, finite horizon, robust optimal control problem $\mathbf{P}_N(\cdot)$:

$$\mathbf{P}_N(x): \qquad V_N^0(x) = \arg\inf_{\theta}\{V_N(x, \theta) \mid \theta \in \Theta(x)\} \tag{14}$$

$$\theta^0(x) \in \arg\inf_{\theta}\{V_N(x, \theta) \mid \theta \in \Theta(x)\} \tag{15}$$

The set of states for which there exists an admissible tube–control policy pair θ is clearly given by:

$$\mathcal{X}_N \triangleq \{x \mid \Theta(x) \neq \emptyset\} \tag{16}$$

The robust optimal control problem $\mathbf{P}_N(x)$ is highly complex in general case, since it requires optimization over control policies and sets. *We focus attention on the case when the system being controlled is linear and constraints specified by* (2) *are polytopic while obstacle avoidance constraint* (3) *are polygonic so that the overall state constraints* (4) *are polygonic.*

3 Linear – Polygonic Case

Here we consider the linear discrete-time, time invariant, system:

$$x^+ = f(x, u, w) \triangleq Ax + Bu + w \tag{17}$$

where, as before, $x \in \mathbf{R}^n$ is the current state, $u \in \mathbf{R}^m$ is the current control action, x^+ is the successor state, $w \in \mathbf{R}^n$ is an unknown disturbance and $(A, B) \in \mathbb{R}^{n \times n} \times \mathbb{R}^{n \times m}$. The disturbance w is persistent, but contained in a convex and compact set $\mathbf{W} \subset \mathbf{R}^n$ that contains the origin. *We make the standing* *assumption that the couple* (A, B) *is controllable.* With system (17) we associate the corresponding nominal system:

$$z^+ = Az + Bv \tag{18}$$

where $z \in \mathbf{R}^n$ is the current state, $v \in \mathbf{R}^m$ is the current control action and z^+ is the successor state of the nominal system. Let $\phi(i; x, \pi, \mathbf{w})$ denote the solution at time i of (17) when the control policy is $\pi \triangleq \{\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$, the disturbance sequence is $\mathbf{w}$ and the initial state is x at time 0. If the initial state of nominal model is z at time 0 then $\bar{\phi}(k; z, \mathbf{v})$ denotes the solution to (18) at time instant k, given the control sequence $\mathbf{v} \triangleq \{v_0, v_1 \ldots v_{N-1}\}$.

Definition 1. *[1] A set $\Omega \subset \mathbf{R}^n$ is a* robust positively invariant *(RPI) set for system $x^+ = f(x, w)$ and constraint set $(\mathbf{X}, \mathbf{W})$ if $\Omega \subseteq \mathbf{X}$ and $f(x, w) \in \Omega,\ \forall w \in \mathbf{W}, \forall x \in \Omega$.*

A set $\Omega \subset \mathbf{R}^n$ is a positively invariant *(PI) set for system $x^+ = f(x)$ and constraint set $\mathbf{X}$ if $\Omega \subseteq \mathbf{X}$ and $f(x) \in \Omega,\ \forall x \in \Omega$.*

A set $\Omega \subset \mathbf{R}^n$ is a robust control invariant *(RCI) set for system $x^+ = f(x, u, w)$ and constraint set $(\mathbf{X}, \mathbf{U}, \mathbf{W})$ if $\Omega \subseteq \mathbf{X}$ and for every $x \in \Omega$ there exists a $u \in \mathbf{U}$ such that $f(x, u, w) \in \Omega,\ \forall w \in \mathbf{W}$.*

A set $\Omega \subset \mathbf{R}^n$ is a control invariant *(CI) set for system $x^+ = f(x, u)$ and constraint set $(\mathbf{X}, \mathbf{U})$ if $\Omega \subseteq \mathbf{X}$ and for every $x \in \Omega$ there exists a $u \in \mathbf{U}$ such that $f(x, u) \in \Omega$.*

If the set Ω is a RCI set for system $x^+ = f(x, u, w)$ and constraint set $(\mathbf{X}, \mathbf{U}, \mathbf{W})$, then there exists a control law $\nu : \Omega \to \mathbf{U}$ such that the set Ω is a RPI set for system $x^+ = f(x, \nu(x), w)$ and constraint set $(\mathbf{X}_\nu, \mathbf{W})$ with $\mathbf{X}_\nu \triangleq \{x \in \mathbf{X} \mid \nu(x) \in \mathbf{U}\}$. The control law $\nu(\cdot)$ is any control law satisfying:

$$\nu(x) \in \mathcal{U}(x),\ \mathcal{U}(x) \triangleq \{u \in \mathbf{U} \mid f(x, u, w) \in \Omega,\ \forall w \in \mathbf{W}\},\ x \in \Omega \tag{19}$$

An interesting observation [7, 14] is recalled next:

Proposition 2. *Let Ω be a RPI set for system $x^+ = Ax + B\nu(x) + w$ and constraint set $(\mathbf{X}_\nu, \mathbf{W})$, where $\mathbf{X}_\nu \triangleq \{x \in \mathbf{X} \mid \nu(x) \in \mathbf{U}\}$. Let also $x \in z \oplus \Omega$ and $u = v + \nu(x - z)$. Then for all $v \in \mathbf{R}^m$, $x^+ \in z^+ \oplus \Omega$ where $x^+ \triangleq Ax + Bu + w,\ w \in \mathbf{W}$ and $z^+ \triangleq Az + Bv$.*

Proposition 2 allows us to exploit a simple parameterization of the tube-policy pair $(\mathbf{X}, \pi)$ as follows. The state tube $\mathbf{X} = \{X_0, X_1, \ldots, X_N\}$ is parametrized by $\{z_i\}$ and $\mathcal{R}$ as follows:

$$X_i \triangleq z_i \oplus \mathcal{R} \tag{20}$$

where z_i is the *tube cross–section center* at time i and $\mathcal{R}$ is a set representing the *tube cross–section*. The control laws $\mu_i(\cdot)$ defining the control policy $\pi = \{\mu_0(\cdot), \mu_1(\cdot), \ldots, \mu_{N-1}(\cdot)\}$ are parametrized by $\{z_i\}$ and $\{v_i\}$ as follows:

$$\mu_i(y) \triangleq v_i + \nu(y - z_i),\ y \in X_i, \tag{21}$$

for all $i \in \mathbf{N}_{N-1}$, where v_i is the feedforward component of the control law and $\nu(\cdot)$ is feedback component of the control law $\mu_i(\cdot)$. With appropriate constraints on $v_i,\ i \in \mathbf{N}_{N-1}$ and *tube cross–section* $\mathcal{R}$, the tube $\mathbf{X}$ and associated policy π satisfy (7)– (11) and, hence, Proposition 1. For a fixed *tube cross–section* $\mathcal{R}$ of appropriate properties, discussed next, the sequence $\{z_i\}$ is *the sequence of tube centers* and is required to satisfy (18), subject to tighter constraints than those in (4), as discussed in the sequel.

3.1 Construction of Simple Robust Control Invariant Tube

Appropriate Tube Cross–Section $\mathcal{R}$

A suitable choice for the *tube cross–section* $\mathcal{R}$ is any RCI set for $x^+ = Ax+Bu+w$ and constraint set $(\mathbf{T}, \mathbf{U}, \mathbf{W})$. Since $\mathbf{T} \subseteq \mathbf{X_O}$ by assumption it is clear that any RCI set $\mathcal{R}$ for $x^+ = Ax+Bu+w$ and constraint set $(\mathbf{T}, \mathbf{U}, \mathbf{W})$ is also a RCI set for $x^+ = Ax+Bu+w$ and constraint set $(\mathbf{X_O}, \mathbf{U}, \mathbf{W})$. Consequently, we assume that:

Assumption 3.1 *The set* $\mathcal{R}$ *is a compact RCI set for system* (17) *and constraint set* $(\alpha\mathbf{T}, \beta\mathbf{U}, \mathbf{W})$ *where* $(\alpha, \beta) \in [0,1) \times [0,1)$.

A practical consequence of Assumption 3.1 is the fact that there exists a control law $\nu\ :\ \mathcal{R} \to \mathbf{U}$ such that $\mathcal{R}$ is RPI set for system $x^+ = Ax + B\nu(x) + w$ and constraint set $(\mathbf{X}_\nu, \mathbf{W})$, where $\mathbf{X}_\nu \triangleq \{x \in \alpha\mathbf{T} \mid \nu(x) \in \beta\mathbf{U}\}$ with $(\alpha, \beta) \in [0,1) \times [0,1)$. *Note that* the set $\mathcal{R}$ need not be a convex (for instance polytopic, ellipsoidal) set*; we merely require that it satisfies Assumption 3.1.* Clearly, it is desirable to reduce conservatism introduced by the simple tube-control policy parametrization. Several methods can be employed to construct the RCI set $\mathcal{R}$ and corresponding control policy $\nu(\cdot)$, see for instance [1, 3, 10] for a set of the standard methods. Additionally a set of recent and improved methods, that allow one to minimize an appropriate norm of the set $\mathcal{R}$ and to compute this set by solving an appropriately specified optimization problem can be found in [15, 16].

Given a set $\mathcal{R}$ satisfying Assumption 3.1 and corresponding control law $\nu(\cdot)$ such that the set $\mathcal{R}$ is RPI set for system $Ax + B\nu(x) + w$ and constraint set $(\mathbf{X}_\nu, \mathbf{W})$, where $\mathbf{X}_\nu \triangleq \{x \in \alpha\mathbf{T} \mid \nu(x) \in \beta\mathbf{U}\}$ with $(\alpha, \beta) \in [0,1) \times [0,1)$, let the sets $U_\nu,\ \mathbf{Z_O},\ \mathbf{T}_f,\ \mathbf{V}$ be defined as follows:

$$U_\nu \triangleq \{\nu(x) \mid x \in \mathcal{R}\},\ \mathbf{Z_O} \triangleq \mathbf{X_O} \ominus \mathcal{R},\ \mathbf{T}_f \triangleq \mathbf{T} \ominus \mathcal{R},\ \mathbf{V} \triangleq \mathbf{U} \ominus U_\nu. \tag{22}$$

Since $\mathbf{T} \subseteq \mathbf{X_O}$, Assumption 3.1 implies the sets $\mathbf{Z_O},\ \mathbf{T}_f,\ \mathbf{V}$ are non–empty sets. If the sets $\mathcal{R},\ U_\nu$ and $\mathbf{T}$ are polytopes, the set $\mathbf{Z_O}$ is polygonic and the sets $\mathbf{V}$ and $\mathbf{T}_f$ are polytopic and all contain the origin [12, 16]. Note that $z \oplus \mathcal{R} \subseteq \mathbf{X_O}$ for any $z \in \mathbf{Z_O}$ by definition of the Minkowski/Pontryagin set difference.

Appropriate Tube Terminal Set $\mathbf{X}_f \subseteq \mathbf{T}$

The parametrization for the state tube $\mathbf{X}$ motivates the introduction of a *set of sets* of the form $\Phi \triangleq \{z \oplus \mathcal{R} \mid z \in Z_f\}$ (Φ is a *set of sets*, each of the form $z \oplus \mathcal{R}$ where $\mathcal{R}$ is a set) that is *set robust control invariant* [13, 16]:

Definition 2. *A set of sets $\varPhi$ is* set robust control invariant *(SRCI) for system $x^+ = f(x,u,w)$ and constraint set $(\mathbf{T}, \mathbf{U}, \mathbf{W})$ if, for any set $X \in \varPhi$, (i) $X \subseteq \mathbf{T}$ and, (ii) there exists a policy $\theta_X : X \to \mathbf{U}$ such that $X^+ = \mathcal{F}(X, \theta_X, \mathbf{W}) \triangleq \{f(x, \theta_X(x), w) \mid (x,w) \in X \times \mathbf{W}\} \subseteq Y$ for some set $Y \in \varPhi$.*

In order to characterize a simple *set robust control invariant set $\varPhi$*, we additionally assume that:

Assumption 3.2 *The set $\mathbf{Z}_f$ is a control invariant set for the system* (18) *and constraint set* $(\mathbf{T}_f, \mathbf{V})$*.*

We now recall the following result recently established in [13, 16]:

Theorem 1. *Suppose that assumptions 3.1 and 3.2 are satisfied. Then $\varPhi \triangleq \{z \oplus \mathcal{R} \mid z \in \mathbf{Z}_f\}$ is a* set robust control invariant *for system $x^+ = Ax + Bu + w$ and constraint set $(\mathbf{T}, \mathbf{U}, \mathbf{W})$.*

Given an $X \triangleq z \oplus \mathcal{R} \in \varPhi$ the corresponding policy $\theta_X : X \to \mathbf{U}$ can be defined by:

$$\theta_X(x) = \varphi(z) + \nu(x - z),\ x \in X,\ X = z \oplus \mathcal{R} \in \varPhi \tag{23}$$

where $\varphi(\cdot)$ is a control law such that $\mathbf{Z}_f$ is PI set for $z^+ = Az + B\varphi(z)$ and $\mathbf{Z}_\varphi \triangleq \mathbf{T}_f \cap \{z \mid \varphi(z) \in \mathbf{V}\}$ and $\nu(\cdot)$ is a control law such that $\mathcal{R}$ is RPI set for $x^+ = Ax + B\nu(x) + w$ and $(\mathbf{X}_\nu, \mathbf{W})$. As discussed in [13, 14], an appropriate tube *terminal set* $\mathbf{X}_f$ such that $\mathbf{X}_f \subseteq \mathbf{T} \subseteq \mathbf{X_O}$ is defined by:

$$\mathbf{X}_f \triangleq \mathbf{Z}_f \oplus \mathcal{R} \tag{24}$$

where the sets $\mathcal{R}$ and $\mathbf{Z}_f$ satisfy assumptions 3.1 and 3.2 respectively. With this choice for the *terminal set* the domain of attraction is enlarged (compared to the case when $\mathbf{X}_f = \mathcal{R}$) [13, 14]. *In the sequel, we assume that Assumptions 3.1– 3.2 hold and additionally that the terminal set $\mathbf{Z}_f$, $\mathbf{V}$ and $\mathcal{R}$ are polytopes.*

4 Robust Model Predictive Controller

4.1 Simple Tube Controller

We are now in position to propose a relatively simple optimal control problem that approximates the robust optimal control problem $\mathbf{P}_N(x)$ and whose solution, if it exists, yields the tube–policy pairs satisfying all the conditions specified by (7)– (11). We require that the state trajectory of the nominal model (*the sequence of tube centers*) $\mathbf{z} \triangleq \{z_0, z_1, \ldots, z_N\}$ and corresponding control sequence $\mathbf{v} = \{v_0, v_1, \ldots, v_{N-1}\}$ satisfy the tighter constraints, defined in (22). Let the set $\mathcal{V}_N(x)$ of admissible control–state pairs for nominal system at state x be defined as follows:

$$\begin{aligned}\mathcal{V}_N(x) \triangleq \{(\mathbf{v}, z) \mid (\bar{\phi}(k; z, \mathbf{v}), v_k) \in \mathbf{Z_O} \times \mathbf{V},\ \forall k \in \mathbf{N}_{N-1},\\ \bar{\phi}(N; z, \mathbf{v}) \in \mathbf{Z}_f,\ x \in z \oplus \mathcal{R}\}\end{aligned} \tag{25}$$

Since the set $\mathbf{Z_O}$ is polygon and the sets $\mathbf{Z}_f$, $\mathbf{V}$, $\mathcal{R}$ are polytopes it follows that the set $\mathcal{V}_N(x)$ is a polygonic set; it is, in principle, possible to obtain more detailed characterization of the set $\mathcal{V}_N(x)$ [12, 16]. An appropriate cost function can be defined as follows:

$$V_N(\mathbf{v}, z) \triangleq \sum_{i=0}^{N-1} \ell(z_i, v_i) + V_f(z_N),\ \ell(x, u) \triangleq |x|_Q^2 + |u|_R^2,\ V_f(x) \triangleq |x|_P^2 \quad (26)$$

where for all i, $z_i \triangleq \bar{\phi}(i; z, \mathbf{v})$ and $\ell(\cdot)$ is the stage cost and $V_f(\cdot)$ is the terminal cost, and where P, Q and R are positive definite matrices of appopriate dimensions. We also assume, as is standard [8], that:

Assumption 4.1 *The terminal cost satisfies $V_f(Az + B\varphi(z)) + \ell(z, \varphi(z)) \leq V_f(z)$ for all $z \in Z_f$.*

We consider the resultant, simplified, optimal control problem defined by :

$$\mathbf{P}_N^S(x): \quad V_N^0(x) \triangleq \inf_{\mathbf{v}, z}\{V_N(\mathbf{v}, z)\ |(\mathbf{v}, z) \in \mathcal{V}_N(x)\} \quad (27)$$

$$(\mathbf{v}^0(x), z^0(x)) \in \arg\inf_{\mathbf{v}, z}\{V_N(\mathbf{v}, z)\ |(\mathbf{v}, z) \in \mathcal{V}_N(x)\} \quad (28)$$

The domain of the value function $V_N^0(\cdot)$, the controllability set, is:

$$\mathcal{X}_N \triangleq \{x \mid \mathcal{V}_N(x) \neq \emptyset\} \quad (29)$$

Note that *the simplified optimal control problem $\mathbf{P}_N^S(x)$ can be posed as a mixed integer quadratic programming problem*, since $\mathcal{V}_N(x)$ is polygonic and $V_N(\cdot)$ is quadratic, so that its global minimizer (or a set of global minimizers, in which case an appropriate selection can be made) can be found for any $x \in \mathcal{X}_N$. For each i let $\mathcal{V}_i(x)$ and $\mathcal{X}_i$ be defined, respectively, by (25) and (29) with i replacing N. The sequence $\{\mathcal{X}_i\}$ is a monotonically non-decreasing set sequence, i.e. $\mathcal{X}_i \subseteq \mathcal{X}_{i+1}$ for all $i \in \mathbf{N}$. The sets $\mathcal{X}_i$ are in general polygons [12] and are not necessarily connected due to nonconvexity of state constraints. Given any $x \in \mathcal{X}_N$ the solution to $\mathbb{P}_N^S(x)$ defines the corresponding optimal simple tube:

$$\mathbf{X}^0(x) = \{X_i^0(x)\},\ X_i^0(x) = z_i^0(x) \oplus \mathcal{R}, \quad (30)$$

for $i \in \mathbf{N}_N$, and the corresponding control policy $\pi^0(x) = \{\mu_i^0(\cdot) \mid i \in \mathbf{N}_{N-1}\}$ with

$$\mu_i^0(y; x) = v_i^0(x) + \nu(y - z_i^0(x)),\ y \in X_i^0(x) \quad (31)$$

where, for each i, $z_i^0(x) = \bar{\phi}(i; z^0(x), \mathbf{v}^0(x))$. We now establish that the simple RCI tube $\mathbf{X}^0(x)$ and corresponding policy $\pi^0(x)$ satisfy all the constraints specified by (7)– (11) for any $x \in \mathcal{X}_N$. Let:

$$U(X_i^0(\cdot), \mu_i^0(\cdot)) \triangleq v_i^0(x) \oplus U_\nu \quad (32)$$

By construction $X_i^0(x) \subseteq \mathbf{X_O}$ for all $i \in \mathbf{N}_N$ because $z_i^0(x) \in \mathbf{Z_O}$ yields that $z_i^0(x) \oplus \mathcal{R} \subseteq \mathbf{Z_O} \oplus \mathcal{R} \subseteq \mathbf{X_O}$ for all $i \in \mathbf{N}_N$. Also, $z_N^0(x) \in \mathbf{Z}_f$ yields that

$z^0_N(x) \oplus \mathcal{R} \subseteq \mathbf{Z}_f \oplus \mathcal{R} = \mathbf{X}_f \subseteq \mathbf{T}$. Similarly, $U(X^0_i(\cdot), \mu^0_i(\cdot)) \subseteq \mathbf{U}$ for all $i \in \mathbf{N}_{N-1}$ because $v^0_i(x) \in \mathbf{V}$ yields that $v^0_i(x) \oplus U_\nu \subseteq \mathbf{V} \oplus U_\nu \subseteq \mathbf{U}$ for all $i \in \mathbf{N}_{N-1}$. Finally, by Proposition 2 it follows that $\{Ay + B\mu^0_i(y;x) + w \mid (y,w) \in X^0_i(x) \times \mathbf{W}\} \subseteq X^0_{i+1}(x)$ for all $i \in \mathbf{N}_{N-1}$. Clearly, an analogous observation holds for any arbitrary couple $(\mathbf{v}, z) \in \mathcal{V}_N(x)$ given any arbitrary $x \in \mathcal{X}_N$.

We consider the following implicit robust model predictive control law $\kappa^0_N(\cdot)$ yielded by the solution of $\mathbb{P}^S_N(x)$:

$$\kappa^0_N(x) \triangleq v^0_0(x) + \nu(x - z^0(x)) \tag{33}$$

We establish some relevant properties of the proposed controller $\kappa^0_N(\cdot)$ by exploiting the results reported in [9].

Proposition 3. *(i) For all $x \in \mathcal{R}$, $V^0_N(x) = 0$, $z^0(x) = 0$, $\mathbf{v}^0(x) = \{0, 0, \ldots, 0\}$ and $\kappa^0_N(x) = \nu(x)$. (ii) Let $x \in \mathcal{X}_N$ and let $(\mathbf{v}^0(x), z^0(x))$ be defined by (28), then for all $x^+ \in Ax + B\kappa^0_N(x) \oplus \mathbf{W}$ there exists $(\mathbf{v}(x^+), z(x^+)) \in \mathcal{V}_N(x^+)$ and*

$$V^0_N(x^+) \leq V^0_N(x) - \ell(z^0(x), v^0_0(x)). \tag{34}$$

The main stability result follows from Theorem 1 in [9] (definition of robust exponential stability of a set can be found in [9, 14]):

Theorem 2. *The set $\mathcal{R}$ is robustly exponentially stable for controlled uncertain system $x^+ = Ax + B\kappa^0_N(x) + w,\ w \in \mathbf{W}$. The region of attraction is $\mathcal{X}_N$.*

The proposed controller $\kappa^0_N(\cdot)$ results in a set sequence $\{X^0_0(x(i))\}$, where:

$$X^0_0(x(i)) = z^0(x(i)) \oplus \mathcal{R},\ i \in \mathbf{N} \tag{35}$$

and $z^0(x(i)) \to 0$ exponentially as $i \to \infty$. The actual trajectory $x(\cdot) \triangleq \{x(i)\}$, where $x(i)$ is the solution of $x^+ = Ax + B\kappa^0_N(x) + w$ at time $i \in \mathbf{N}$, corresponding to a particular realization of an infinite admissible disturbance sequence $w(\cdot) \triangleq \{w_i\}$, satisfies $x(i) \in X^0_0(x(i)), \forall i \in \mathbf{N}$. Proposition 3 implies that $X^0_0(x(i)) \subseteq \mathcal{X}_N,\ \forall i \in \mathbf{N}$ and Theorem 2 implies that $X^0_0(x(i)) \to \mathcal{R}$ (where $\mathcal{R} \subseteq \mathbf{T}$) as $i \to \infty$ exponentially in the Hausdorff metric.

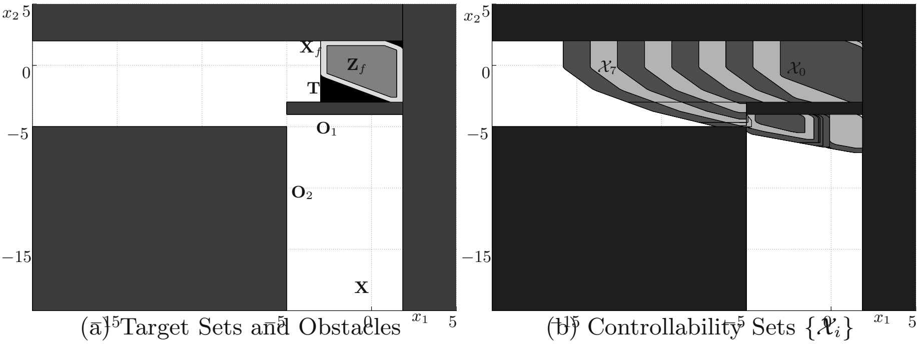
4.2 Illustrative Example

Our illustrative example is a double integrator:

$$x^+ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u + w \tag{36}$$

with $w \in \mathbf{W} \triangleq \{\ w \in \mathbb{R}^2\ :\ |w|_\infty \leq 0.2\ \}$, $x \in \mathbf{X} \triangleq \{x \in \mathbb{R}^2 \mid |x|_\infty \leq 20,\ x^1 \leq 1.85,\ x^2 \leq 2\}$, $u \in \mathbf{U} \triangleq \{u \mid |u| \leq 2\}$ and $\mathbf{T} \triangleq \{\ x \in \mathbb{R}^2\ :\ |x|_\infty \leq 3\ \} \cap \mathbf{X}$, where x^i is the i^{th} coordinate of a vector x. The cost function is defined by (26) with $Q = 100I$, $R = 100$; the terminal cost $V_f(x)$ is the value function $(1/2)x'P_f x$ for the optimal unconstrained problem for the nominal system. The horizon is $N = 8$. The *tube cross-section* $\mathcal{R}$ is constructed by using methods of [15, 16].

The sequence of the sets $\{\mathcal{X}_i\}$, $i = 0, 1, \ldots, 8$, where $\mathcal{X}_i$ is the domain of $V_i^0(\cdot)$ and the terminal set $\mathbf{X}_f = \mathbf{Z}_f \oplus \mathcal{R}$ where $\mathbf{Z}_f$ satisfies Assumption 3.2 and is the maximal positively invariant set [1] for system $z^+ = (A+BK)z$ under the tighter constraints $\mathbf{T}_f = \mathbf{T} \ominus \mathcal{R}$ and $\mathbf{V} = \mathbf{U} \ominus U_\nu$ where K is unconstrained DLQR controller for (A, B, Q, R), is shown in Figure 1 together with obstacles $\mathbf{O}_1$ and $\mathbf{O}_2$.



(a) Target Sets and Obstacles

(b) Controllability Sets $\{\mathcal{X}_i\}$

Fig. 1. Sets $\{\mathcal{X}_i\}_{i=0}^{8}$, $\mathbf{T}$, $\mathbf{Z}_f$, $\mathbf{X}_f$ and Obstacles $\mathbf{O}_1$ and $\mathbf{O}_2$

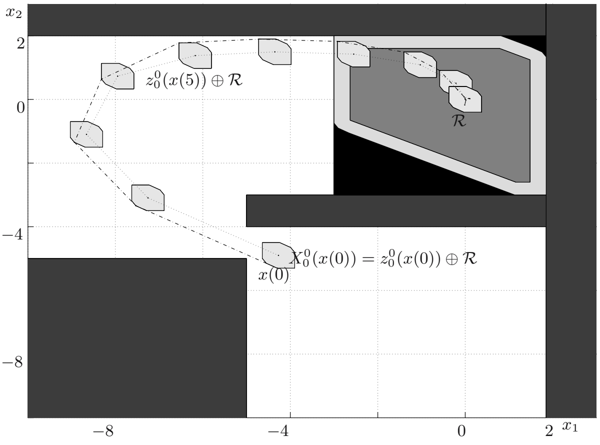


Fig. 2. RMPC Tube Trajectory

A RMPC tube $\{z_0^0(x(i)) \oplus \mathcal{R}\}$ for initial state $x_0 = (-4.3, -5.3)'$ is shown in Figure 2 for a sequence of random admissible disturbances. The dash-dot line is the actual trajectory $\{x(i)\}$ due to the disturbance realization while the dotted line is the sequence $\{z_0^0(x(i))\}$ of optimal initial states for corresponding nominal system.

5 Conclusions and Extensions

This note has introduced a relatively simple tube controller for the obstacle avoidance problem for uncertain linear discrete time systems. The robust model predictive controller ensures robust exponential stability of $\mathcal{R}$, a RCI set – the 'origin' for the controlled uncertain system. *The complexity of the corresponding robust optimal control problem is marginally increased compared with that for conventional model predictive control.* The proposed robust model predictive scheme guarantees robust obstacle avoidance at discrete moments; thus the resultant controller, if applied to a continuous time system, will not necessarily ensure satisfaction of constraints between sampling instants. However, this problem can be dealt with as will be shown in a future paper. It is possible to consider the cases when control objective is merely reaching the target set $\mathbf{T}$ rather than stabilizing an "equilibrium point" $\mathcal{R} \subseteq \mathbf{T}$; it is also, in principle, possible to treat the "multi-system" – "multi-target" case. These modifications are relatively straight–forward, but they require a set of appropriate changes. Finally, combining the results reported in [11], an extension of the proposed robust model predictive scheme to the class of piecewise affine discrete time systems is possible.

References

[1] F. Blanchini. Set invariance in control. *Automatica*, 35:1747–1767, 1999. survey paper.

[2] H.L. Hagenaars, J. Imura, and H. Nijmeijer. Approximate continuous-time optimal control in obstacle avoidance by time/space discretization of non-convex constraints. In *IEEE Conference on Control Applications*, pages 878–883, September 2004.

[3] I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariance sets for discrete-time linear systems. *Mathematical Problems in Engineering: Theory, Methods and Applications*, 4:317–367, 1998.

[4] A. B. Kurzhanski. Dynamic optimization for nonlinear target control synthesis. In *Proceedings of the 6th IFAC Symposium – NOLCOS2004*, pages 2–34, Stuttgart, Germany, September 2004.

[5] A. B. Kurzhanski, I. M. Mitchell, and P. Varaiya. Control synthesis for state constrained systems and obstacle problems. In *Proc. 6th IFAC Symposium – NOLCOS2004*, pages 813–818, Stuttgart, Germany, September 2004.

[6] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40:125–133, 2004.

[7] D. Q. Mayne and W. Langson. Robustifying model predictive control of constrained linear systems. *Electronics Letters*, 37:1422–1423, 2001.

[8] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36:789–814, 2000. Survey paper.

[9] D. Q. Mayne, M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41:219–224, 2005.

[10] S. V. Raković, E.C. Kerrigan, K.I. Kouramas, and D. Q. Mayne. Invariant approximations of the minimal robustly positively invariant sets. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005.

[11] S. V. Raković and D. Q. Mayne. Robust model predictive control of constrained piecewise affine discrete time systems. In *Proceedings of the 6th IFAC Symposium – NOLCOS2004*, pages 741–746, Stuttgart, Germany, September 2004.

[12] S. V. Raković and D. Q. Mayne. Robust time optimal obstacle avoidance problem for constrained discrete time systems. In *44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005.

[13] S. V. Raković and D. Q. Mayne. Set robust control invariance for linear discrete time systems. In *Proceedings of the 44th IEEE Conference on Decision and Control*, Seville, Spain, December 2005.

[14] S. V. Raković and D. Q. Mayne. A simple tube controller for efficient robust model predictive control of constrained linear discrete time systems subject to bounded disturbances. In *Proceedings of the 16th IFAC World Congress IFAC 2005*, Praha, Czech Republic, July 2005. Invited Session.

[15] S. V. Raković, D. Q. Mayne, E. C. Kerrigan, and K. I. Kouramas. Optimized robust control invariant sets for constrained linear discrete – time systems. In *Proceedings of the 16th IFAC World Congress IFAC 2005*, Praha, Czech Republic, July 2005.

[16] Saša V. Raković. *Robust Control of Constrained Discrete Time Systems: Characterization and Implementation.* PhD thesis, Imperial College London, London, United Kingdom, 2005.

[17] A. Richards and J.P. How. Aircraft trajectory planning with collision avoidance using mixed integer linear programming. In *Proc. American Control Conference*, pages 1936 – 1941, 2002.

[18] S. Sundar and Z. Shiller. Optimal obstacle avoidance based on Hamilton–Jacobi–Bellman equation. *IEEE transactions on robotics and automation*, 13(2):305 – 310, 1997.

Trajectory Control of Multiple Aircraft: An NMPC Approach

Juan J. Arrieta–Camacho[1], Lorenz T. Biegler[1],
and Dharmashankar Subramanian[2]

[1] Chemical Engineering Department, Carnegie Mellon University,
Pittsburgh, PA 15213
{jarrieta,lb01}@andrew.cmu.edu
[2] IBM Watson Research Center, Yorktown Heights, NY 10598
dharmash@us.ibm.com

Summary. A multi-stage nonlinear model predictive controller is derived for the real-time coordination of multiple aircraft. In order to couple the versatility of hybrid systems theory with the power of NMPC, a finite state machine is coupled to a real time optimal control formulation. This methodology aims to integrate real-time optimal control with higher level logic rules, in order to assist mission design for flight operations like collision avoidance, conflict resolution, and reacting to changes in the environment. Specifically, the controller is able to consider new information as it becomes available. Stability properties for nonlinear model predictive control are described briefly along the lines of a dual-mode controller. Finally, a small case study is presented that considers the coordination of two aircraft, where the aircraft are able to avoid obstacles and each other, reach their targets and minimize a cost function over time.

1 Introduction

Coordination of aircraft that share common air space is an important problem in both civil and military domains. Ensuring safe separation among aircraft, and avoidance of obstacles and no-fly zones are key concerns along with optimization of fuel consumption, mission duration and other criteria. In previous work [10] we developed an optimal control formulation for this problem with path constraints to define the avoidance requirements and flyability constraints. There we considered a direct transcription, nonlinear programming strategy solved with the IPOPT solver [11]. Results for conflict resolution, using detailed flight models and with up to eight aircraft, were obtained quickly, and motivated the implementation of such strategy in real time.

The level of information for these problems, including recognition of obstacles and the presence of other aircraft, evolves over time and can be incomplete at a given instant. This motivates the design of an on-line strategy able to consider new information as it becomes available. For this purpose we propose a nonlinear model predictive control (NMPC) approach. This approach integrates real-time optimal control with higher level logic rules, in order to assist mission design for flight operations like collision avoidance and conflict resolution. In this work,

R. Findeisen et al. (Eds.): Assessment and Future Directions, LNCIS 358, pp. 629–639, 2007.
springerlink.com

such integration is achieved by coupling a Finite State Machine (FSM) with an NMPC regulator. The FSM receives the current state of the environment and outputs a collection of sets, which is used to alter a nominal optimal control problem (OCP) in the NMPC regulator. For instance, the detection of a new obstacle leads the FSM to add a new element to the relevant set. The update then alters a nominal OCP by adding the constraints pertinent to the obstacle just detected, thus leading to an optimal avoidance maneuver.

In the next section we derive the multi-stage NMPC problem formulation. Within this framework the NMPC regulator incorporates a 3 degree-of-freedom nonlinear dynamic model of each aircraft, and considers a path constrained OCP that minimizes a performance index over a moving time horizon. In addition, we describe characteristics of the NMPC formulation that allow the aircraft to meet their targets. Stability properties for NMPC are discussed and adapted to the particular characteristics of this application in Section 3. In Section 4, our overall approach is applied to a small case study which demonstrates collision avoidance as well as implementation of the NMPC controller within the FSM framework. Finally, Section 5 concludes the paper and presents directions for future work.

2 Optimization Background and Formulation

We begin with a discussion of the dynamic optimization strategy used to develop our NMPC controller.

Optimization of a system of Differential Algebraic Equations (DAEs) aims to find a control action $u \in \mathbb{U} \subseteq \mathbb{R}^{n_u}$ such that a cost functional is minimized. The minimization is subject to operational constraints and leads to the following Optimal Control Problem (OCP):

$$\begin{array}{lll} \min\limits_{u} & J[z_d(t_F), t_F] & \\ \text{subject to: } \dot{z_d} & = f_d[z_d, z_a, u], & t \in T_H \\ 0 & = z_d(t_I) - z_{d,I} & \\ 0 & = f_a[z_d, z_a, u], & t \in T_H \\ 0 & \leq g[z_d, z_a, u, t], u(t) \in \mathbb{U}, & t \in T_H \end{array} \tag{1}$$

where $z_d \in \mathbb{R}^{n_d}$ and $z_a \in \mathbb{R}^{n_a}$ are the vectors of differential and algebraic variables, respectively. Given $u(t)$, a time horizon of interest $T_H := [t_I, t_F]$ and appropriate initial conditions $z_d(t_I) = z_{d,I}$, the dynamic behavior of the aircraft can be simulated by solving the system of DAEs: $\dot{z_d} = f_d[z_d, z_a, u]$, $f_a[z_d, z_a, u] = 0$, with this DAE assumed to be index 1. Notice that some constraints are enforced over the *entire* time interval T_H. In this study, we solve this problem with a direct transcription method [4, 5], which applies a simultaneous solution and optimization strategy. Direct transcription methods reduce the original problem to a finite dimension by applying a certain level of *discretization.* The discretized version of the OCP, a sparse nonlinear programming (NLP) problem, can be solved with well known NLP algorithms [5] like sequential quadratic programming or interior point methods. The size of the NLP resulting from the discretization procedure can be very large, so the NLP algorithm used for the solution must be suitable for large scale problems.

In this work, the OCP is transcribed into an NLP via collocation on finite elements. As described in [5], the interval T_H is divided into n_E *finite elements*. Within element i, the location of collocation point j occurs at the scaled root of an orthogonal polynomial. In this work, roots of Radau polynomials are used, as they allow to stabilize the system [3] when high index constraints are present. State profiles are approximated in each element by polynomials; differential states are represented by monomial basis polynomials while algebraic states and controls are represented by Lagrange basis polynomials. These polynomials are substituted into the DAE model and the DAE is enforced, over time, at Radau collocation points over finite elements. Continuity across element boundaries is also enforced for the differential state profiles. With this approximation, the optimal control problem (1) can be written as:

$$\begin{aligned} \min \;\; & \phi(w) \\ \text{subject to:} \;\; & c(w) = 0 \\ & w_L \leq w \leq w_U . \end{aligned} \tag{2}$$

Here, the equality constraint vector $c(w)$ contains the discretized differential equations and constraints of (1). Notice that inequality constraints are enforced as equalities via slack variables. In a similar manner, the vector w consists of the polynomial coefficients for the state, control, algebraic and (possibly) slack variables.

The NLP (2) is solved using a primal-dual interior point method. Specifically, we use the Interior Point OPTimizer– IPOPT [11]. This solver follows a barrier approach, in which the bounds on the variables of the NLP problem (2) are replaced by a logarithmic barrier term added to the objective function, and a sequence of these barrier problems is solved for decreasing values of the penalty parameter. In essence, IPOPT approaches the solution of (2) from the interior of the feasible region defined by the bounds. A detailed description of IPOPT, both from the theoretical and algorithmic standpoints, can be found in [11]. In this study, IPOPT is used through its interface with AMPL [7], a modeling language that eases the problem declaration and provides the solver with exact first and second derivatives via automatic differentiation.

2.1 Nonlinear Model Predictive Control

If a perfect model is available for dynamic behavior of the aircraft, as well as full information regarding the surrounding environment, an *a priori* computation of the optimal control actions would be possible. However, neither of these occur in practice; the dynamic models merely approximate the behavior of the aircraft, and the system operates in a partially unknown airspace. An alternative to handle the modeling inaccuracies and relative lack of information, is to compute the optimal controls (maneuvers) in real time.

Nonlinear Model Predictive Control (NMPC) is a closed loop control strategy in which a nonlinear model of the system is used to compute an optimal control via the solution of an optimal control problem. This computation is performed in real time, at every sampling interval [1]. Among the main advantages of NMPC, is the ability to compute the control using higher fidelity nonlinear models (as opposed to linear-model approximation of the dynamics) and impose constraints explicitly. For a thorough overview of both the generalities and formal treatment of NMPC and related on-line control strategies, please refer to [1, 6, 8].

In the context of the methodology presented above, the NMPC controller requires us to formally represent the DAE model in (2) as the discrete time, nonlinear, autonomous system

$$z(k+1) = \bar{f}[z(k), u(k)], \tag{3}$$

where $z(k) \in \mathbb{R}^n$ and $u(k) \in \mathbb{R}^m$ are, respectively, the (differential) state and control variables, evaluated at time points t_k with integers $k > 0$. (Note that since the DAE system in (2) is index one, the algebraic variables can be represented as implicit functions of $z(k)$.) The nonlinear function $\bar{f} : \mathbb{R}^{n\times m} \mapsto \mathbb{R}^n$ is assumed to be twice continuously differentiable with respect to its arguments, and the evolution in (3) results from the solution of the DAE in (1). The goal is to find a control law such that a performance index is minimized, and both states and controls belong to a given set: $z(k) \in \mathbb{Z}$ and $u(k) \in \mathbb{U}, \forall k$.

It is important to distinguish between the *actual* states and controls, and the *predicted* or *computed* states and controls. For this reason, we introduce the following notation: $z(k)$ is the *actual* state of the physical system at time step k, which is reached by the *actual* implementation of the control action $u(k-1)$. On the other hand, $\bar{z}(l)$ is the *predicted* state from time step k, l steps into the future, by the simulation of the system with the *computed* control action $\bar{u}(l-1)$.

At time step k, we define the performance index

$$J[z(k), \bar{u}, N] = \sum_{l=0}^{N-1} \psi[\bar{z}(l), \bar{u}(l)] + F[\bar{z}(N)], \tag{4}$$

which is a function of the initial condition $z(k)$, the vector of control actions $\bar{u}$ used to simulate the system, and the length of the prediction horizon N. In the interest of finding the best performance index, an optimization problem is formulated:

$$\begin{aligned} \min_{\bar{u}} \; J[z(k), \bar{u}, N] &= \sum_{l=0}^{N-1} \psi[\bar{z}(l), \bar{u}(l)] + F[\bar{z}(N)] \\ \text{subject to:}& \\ \bar{z}(l+1) &= \bar{f}[\bar{z}(l), \bar{u}(l)] \\ \bar{z}(0) &= z(k) \\ \bar{g}[\bar{z}(l), \bar{u}(l)] &\leq 0 \\ \bar{u} &\in \mathbb{U}. \end{aligned} \tag{5}$$

where the inequality constraints $\bar{g}[\cdot] \leq 0$ correspond to inequality constraints from (2).

This problem can be considered within the framework of (1) and is solved using the direct transcription strategy outlined in the previous section. The solution to (5) is given by $\bar{u}_k^* = [\bar{u}^*(0), \bar{u}^*(1), \ldots, \bar{u}^*(N-1)]$.

In NMPC, the first element of $\bar{u}^*$ is implemented on the actual system, defining the control law $u(k) = \bar{\kappa}[z(k)] := \bar{u}^*(0)$ that leads to the closed loop system $z(k+1) = \bar{f}[z(k), \bar{\kappa}[z(k)]] = \bar{f}[z(k), u(k)]$. At the next sampling interval $k+1$, a new control action, $u(k+1)$, is found in a similar manner.

2.2 Multistage Controller

In the application at hand, some information about the environment is not known a priori. For instance, the presence of an obstacle could be unknown until such obstacle is within radar distance of the aircraft. For this reason, it is not possible to include all the pertinent constraints in the optimization problem a priori. Also, a new way-point might be assigned to an aircraft at any given time. These difficulties can be overcome by using a multi-stage controller. Specifically, we couple a finite state machine (FSM) with the NMPC controller.

An FSM is an event-driven system, that makes a transition from one state to another when the condition defining the transition is true. In our application, to each state of the FSM corresponds a set $\mathcal{S}$ relevant to a nominal OCP. The OCP is formed by constraints and variables that are indexed by $\mathcal{S}$. The FSM is also able to alter parameters relevant to the OCP, for instance, the position and radius of a recently detected obstacle. The new information is passed to the nominal OCP by altering the set $\mathcal{S}$, and irrelevant information is removed in a similar manner.

The states in the FSM correspond to the modes of operation: **provide_mission**: which assigns missions to to the corresponding aircraft and issues an appropriate trigger, **wait**: which forces aircraft to wait until a mission is assigned, **cruise**: where control actions are computed and implemented for each aircraft to reach the setpoint defined by the current mission and detect obstacles, **avoid**: which obtains geography (e.g. position and radius) of detected obstacles and formulates appropriate constraints for the **cruise** mode, **assess_outcome**: which verifies whether the targets have been reached and triggers new missions, and **lock** mode, described below. Additional information related to the FSM can be found in [2]. The NMPC controller, formed by a nominal OCP whose constraints and variables are indexed by the set $\mathcal{S}$, is embedded into the **cruise** and **lock** modes.

The NMPC block solves an OCP that includes the following constraints: **DAE system** describing the dynamic response of the aircraft and flyability constraints (like stall speed, maximum dynamic pressure, and others); **conflict resolution** enforcing a minimum radial separation among aircraft; and **obstacle avoidance** enforcing a minimum separation between aircraft and obstacles. A schematic view of the coupling between the FSM and the NMPC block is presented in Figure 1.

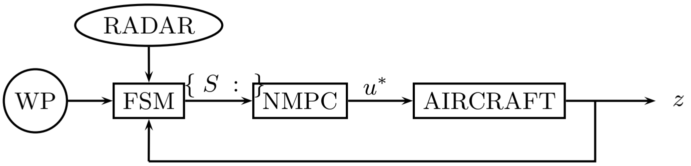


Fig. 1. Schematic view of the coupling between the FSM and the NMPC block. The current state together with the set-point and radar readings cause the FSM to update the set S which, in return, alters the structure of the nominal OCP within the NMPC block. The optimal control action u^* obtained in the NMPC block is implemented in the system.

In order for each aircraft to reach a given target in an efficient manner, we define the following objective functional for each prediction horizon k:

$$J[z(k), \bar{u}, N] = \sum_i P_i \left[\frac{1}{2} \int_{t_I^k}^{t_F^k} \left(\bar{u}_{1,i}^2 + \bar{u}_{2,i}^2 \right) dt + \eta_i \Phi(\bar{\mathbf{z}}_i, \mathbf{z}_i^{sp})|_{t_F^k} \right] \tag{6}$$

where $u_{1,i}$ and $u_{2,i}$ are the forward and vertical load factors for aircraft i, respectively. In this application it suffices to consider the load factor as the acceleration experienced by the aircraft. We choose to minimize the load factor terms because there is a direct relation between the acceleration of an aircraft and fuel consumption (higher forward or upward accelerations require more fuel) and pilot safety and comfort.

In (6), the contributions of each aircraft are added up, weighted by a factor $P_i \geq 0$ representing the *priority* of each aircraft. Each contribution includes an integral term, that measures the control effort, and an exact penalty term $\Phi(\bar{\mathbf{z}}_i, \mathbf{z}_i^{sp})|_{t_F^k} = \|\bar{\mathbf{z}}_i(t_F^k) - \mathbf{z}_i^{sp}\|_1$, weighted by a factor $\eta_i \gg 0$, that enforces the target.

The target is imposed with an exact penalty term and not with a hard constraint, because it is not possible to know *a priori* when the aircraft will reach the target. If the aircraft are far from their targets, the exact penalty formulation encourages a closer distance to the target, without necessarily reaching it. On the other hand, *if the target can be reached within the time horizon of the NMPC controller, the exact penalty is equivalent to a hard constraint*, provided that the weighting factor η_i is sufficiently large (see [9]; in this work we use $\eta_i = 10^5$). If the target can be reached within the time horizon k, the FSM transitions to the **lock** mode, which reduces the interval t_F by one unit at $k+1$, until the target is reached. The penalty term has important implications on the stability properties, as discussed in the next section. The objective functional (6) together with the above constraints and appropriate initial conditions specify the OCP given to the NMPC block.

The FSM was implemented in MATLAB as a collection of `switch` statements. The optimization step of the NMPC block is implemented in AMPL using

IPOPT as the NLP solver. Communication between the NMPC and the FSM was carried out through files updated in every time horizon. In this work, we use a prediction horizon of $N = 10$, $t_F - t_I \leq 60$ seconds, and a sampling time of 6 seconds. We acknowledge that the length of the prediction and implementation horizons are critical tuning parameters in NMPC. The main trade-off is that longer horizons provide solutions closer to the off-line, full-length optimization, but require longer CPU times.

3 Stability Properties

In this section we consider the nominal stability for a particular stage of our controller which solves (5). Ensuring stability of $z(k+1) = \bar{f}[z(k), h(z(k)]$ is a central problem in NMPC, and can be achieved by several methods [1]. The aim is to find a control law such that the origin for the closed-loop system (without loss of generality assumed to be the setpoint) is asymptotically stable, with a large region of attraction. All techniques require some modification to the OCP (5) solved on-line, but have the following in common:

- a positive definite, scalar cost function or performance index $J(\cdot)$, with a final penalty term $F(\cdot)$,
- a nonlinear model $\bar{f}(\cdot)$ describing the dynamic response of the system, from an initial condition $z(0)$, N steps into the future.
- control constraints $\mathbb{U}$ and state constraints $\mathbb{Z}$, and
- a terminal constraint $z \in \mathbb{Z}_s$.

For instance, setting $N = \infty$ in (5) leads to an Infinite Horizon nonlinear control (IH), which can be proved to provide a stabilizing control law. However, its implementation requires the approximation of an infinite summation, leading to a difficult optimization problem that normally cannot be solved in a reasonable time frame. The difficulties associated with the implementation of the IH, motivated the development of control strategies based on finite-horizon (FH) optimization. In particular Nonlinear Receding Horizon (NRH) control, is a group of methodologies (of which NMPC is a member) that *specifically* aims to solve problem (5).

Important cases of NRH include the *zero-state* (ZS) terminal constraint for which the terminal cost $F(\cdot) \equiv 0$ and $\mathbb{Z}_s = \{0\}$, meaning that the end point constraint is enforced as a hard constraint. ZS can guarantee stability if there exists a nonempty neighborhood of the origin $Z^C(N)$ for which it is possible to find a control sequence $u(k),\ k = \{0, \ldots, N-1\}$ capable of driving $z(k+1) = \bar{f}[z(k), u(k)]$ to the origin in N steps (i.e. $z(N) = 0$), and the initial condition $z(0)$ is within that neighborhood. An important drawback of the ZS methodology is that it can require prohibitively long time horizons for $Z^C(N)$ to exist and, even if $Z^C(N)$ exists for a short horizon, this might result in excessive control effort. In addition, satisfying the equality constraint can be computationally demanding.

The idea of replacing the equality constraint by an inequality, which is much easier to satisfy, motivates the *Dual Mode* (DM) controller, for which the $F(\cdot)$ is chosen as an upper bound on the cost of some stabilizing controller that regulates the system, whenever it is within the neighborhood of the origin defined by $\mathbb{Z}_s$. In the implementation of DM, FH control is applied until $z(t) \in \mathbb{Z}_s$, at which point the controller switches to a stabilizing state feedback controller $u(t) = \kappa(z(t))$. Stability of the DM controller can be paraphrased by the following theorem [1, 6, 8]:

Theorem 1 (Nominal Stability of NMPC). *Consider the system described by (3), then with advancing $k > 0$, the NMPC controller leads to a monotonic decrease of $J[z(k)]$ and it is asymptotically stable within a region at least twice the size of $\mathbb{Z}_s$, if we assume:*

- $F(z) > 0, \forall z \in \mathbb{Z}_s \backslash \{0\}$,
- there exists a local control law $u = \kappa(z)$ defined on $\mathbb{Z}_s$, such that $\bar{f}(z, \kappa(z)) \in \mathbb{Z}_s, \forall z \in \mathbb{Z}_s$, and
- $F(\bar{f}[z, \kappa(z)]) - F(z) \leq -\psi[z, \kappa(z)], \forall z \in \mathbb{Z}_s$.

We can apply this result directly for a particular assigned set of way-points if we assume that the cost of some stabilizing controller (including manual control of the aircraft) can be overestimated by the exact penalty term in (6) over the entire test field, i.e., $F(z(t_F^k)) = \eta \Phi(\mathbf{z}_i, \mathbf{z}_i^{sp})|_{t_F^k}$ and $\mathbb{Z}_s = \mathbb{Z}$. A practical realization of this assumption occurs for η suitably large. Because of this assumption and the implementation of the exact penalty term, the stability result applies to (5) for aircraft only in the **cruise** and **lock** modes, and the performance index decreases monotonically within these modes. However, we caution that this result does not imply monotonic decrease over the *entire* set of missions. As new missions are assigned or as different constraints are added in the **avoid** mode, the performance index may indeed increase. The analysis of overall stability (the global case) is left for future work.

4 Two Aircraft Case Study

We now consider the case of two aircraft that accomplish separate missions (defined by way-points (wp)) in a constrained airspace. The trajectory through which a given aircraft reaches the target must be obstacle free and, at every point in time, the different aircraft must maintain a safe distance from each other. The airspace is known to have obstacles, for some of which the position and size are known *a priori*. The aircraft are also equipped with radar, which can detect a previously unknown obstacle. It is assumed that the radar is able to determine both shape and location of a given obstacle within its scope.

Aircraft dynamics can be described by the state variables, $z_d =^{\mathrm{T}} [x\ y\ h\ v\ \chi\ \gamma]$, corresponding to east-range, north-range, altitude, air speed, heading angle and flight path angle, respectively. The control variables are given by $u =^{\mathrm{T}} [u_1\ u_2\ u_3]$ and correspond to forward load factor, vertical load factor, and bank angle,

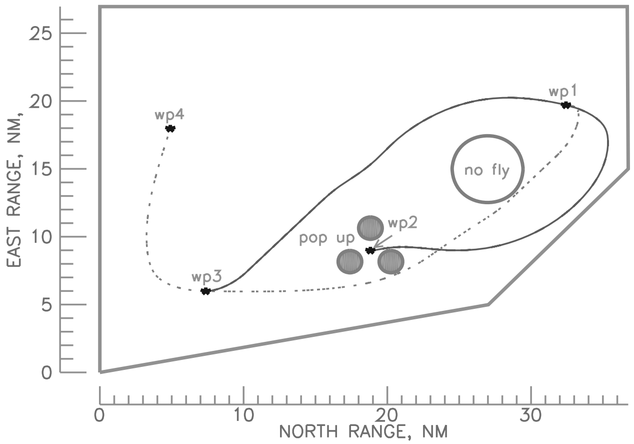


Fig. 2. Test field for the case studies. Aircraft 1 (dotted) and 2 (solid).

respectively. After some simplifying assumptions, the equations of motion are stated for each aircraft as:

$$\begin{aligned} \dot{x} &= v\cos\gamma\cos\chi, & \dot{v} &= g(u_1 - \sin\gamma), \\ \dot{y} &= v\cos\gamma\sin\chi, & \dot{\chi} &= -\tfrac{g}{v}\left(\tfrac{u_2\sin u_3}{\cos\gamma}\right), \\ \dot{h} &= v\sin\gamma, & \dot{\gamma} &= -\tfrac{g}{v}\left(u_2\cos u_3 + \cos\gamma\right), \end{aligned} \tag{7}$$

where g is the standard acceleration. In order to produce flyable maneuvers, constraints defining the flight envelope and other restrictions modeling the performance capabilities of the aircraft are added to the formulation. Using SI units, we have the air density, $\rho = 1.222\exp(-h/9144.0)$ and bounds on velocity, $v \geq v_{\mathrm{S}}\sqrt{9144.0/\rho}$, $v^2 \leq 2q_{\max}/\rho$ and control variables $u_j \in [u_{j\ \min}, u_{j\ \max}]$, $j = 1, \ldots 3$. Here v_{S} is the stall speed and $q_{\max}$ is the maximum dynamic pressure.

We now consider two aircraft flying in the test field presented in Figure 2, where the three small cylinders are pop-up obstacles; their presence is not known *a priori*. Two missions are assigned to each aircraft: **wp1→wp3→wp4** for aircraft 1, and **wp3→wp1→wp2** for aircraft 2. Using the proposed multi-stage NMPC approach, both aircraft are able to reach the assigned way-points, while avoiding obstacles and (locally) minimizing the load factor terms. In Figure 3, notice that aircraft 1 reached the second way-point in 560 seconds, while aircraft 2 reached the second way-point in 660 seconds. The optimization problem solved at each NMPC horizon varies in size, since different information is added and subtracted as the flight evolves. The largest NLP solved consists of 1406 variables and 1342 constraints. The average CPU time required to solve the NMPC problem was 0.2236 seconds, and the maximum CPU time required was of 0.8729 seconds.[1]

[1] SUN Java Workstation: dual AMD64-250 processors @ 2.4GHz with 16GB RAM, running Linux operating system.

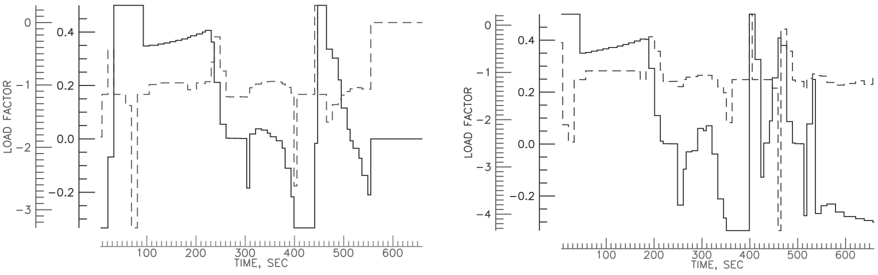


Fig. 3. Optimal control actions for aircraft 1 (left) and 2 (right). Forward load factor (solid, inner axis) and vertical load factor (dashed, outer axis).

5 Conclusions and Future Work

We present a multi-stage, NMPC-based control strategy for the real time coordination of multiple aircraft. The controller couples a finite state machine with a nonlinear model predictive controller. With the proposed methodology, it is possible to coordinate several aircraft, such that they can perform several missions in partially unknown environments. The main advantage of this controller is its ability to consider new information as it becomes available and its ability to define several modes of operation.

A case study with two aircraft was presented. It is noticed that the CPU times required to compute the control action are small compared to the physical time of the implementation (3.7%, on average). Stability of the controller is achieved based on properties of the dual mode NMPC controller and robustness can be promoted by tuning certain parameters within the NMPC regulator. Although good results can be obtained with the methodology presented, it is desirable to investigate more general conditions under which the controller is stable for the entire FSM, in the presence of disturbances, and also with known robustness margins.

The NMPC controller could also be used to assist in the decision-making process involved in the unmanned control of aerospace vehicles. We believe that the concept of combining the versatility of hybrid systems theory with the power of large-scale optimal control can prove very useful in the design of advanced control strategies for the efficient coordination of multiple aircraft.

References

[1] Allgöwer, F., T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright, *Nonlinear predictive control and moving horizon estimation: An introductory overview*, in Advances in Control, Highlights of ECC99, P. M. Frank, ed., Springer, 1999, pp. 391.

[2] Arrieta-Camacho, J. J., L. T. Biegler and D. Subramanian, *NMPC-Based Real-Time Coordination of Multiple Aircraft*, submitted for publication (2005).

[3] Asher, U. M. and Petzold, L., *Computer Methods for Ordinary Differential Equations and Differential Algebraic Equations*, SIAM, 1998.

[4] Betts, J. T., *Practical Methods for Optimal Control Using Nonlinear Programming*, SIAM, Philadelphia, PA, 2001.

[5] Biegler, L. T., *Efficient Solution of Dynamic Optimization and NMPC problems," Nonlinear Model Predictive Control*, edited by F. Allgöwer and E. Zheng, Birkhäuser, 2000, pp. 219.

[6] de Nicolao, G., L. Magni, and R. Scattolini, *Stability and robustness of nonlinear receding horizon control*, in Nonlinear Predictive Control, F. Allgöwer and A. Zheng, eds., vol. 26 of Progress in Systems Theory, Basel, 2000, Birkhauser, pp. 3, 23.

[7] Fourer, R., Gay, D., and Kernighan, B. W., *A Modeling Language for Mathematical Programming*, Management Science, Vol. 36, 1990, pp. 519-554.

[8] Mayne, D., *Nonlinear model predictive control: Challenges and opportunities*, in Nonlinear Predictive Control, F. Allgöwer and A. Zheng, eds., vol. 26 of Progress in Systems Theory, Basel, 2000, Birkhäuser, pp. 23,44.

[9] Nocedal, J. N., and S. J. Wright, *Numerical Optimization*, Springer, Berlin (1998).

[10] Raghunathan, A., V. Gopal, D. Subramanian, L. T. Biegler and T. Samad, *Dynamic Optimization Strategies for 3D Conflict Resolution of Multiple Aircrafts*, AIAA J. of Guidance, Control and Dynamics, 27 (4), pp. 586-594 (2004).

[11] Wächter, A., and L. T. Biegler, *On the Implementation of an Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming*, Mathematical Programming, 106, 1, pp. 25-57 (2006)

Author Index

Lecture Notes in Control and Information Sciences

Edited by M. Thoma, M. Morari

Further volumes of this series can be found on our homepage:
springer.com

Vol. 358: Findeisen R.; Allgöwer F.; Biegler L.T. (Eds.):
Assessment and Future Directions of Nonlinear Model Predictive Control
642 p. 2007 [978-3-540-72698-2]

Vol. 357: Queinnec I.; Tarbouriech S.; Garcia G.; Niculescu S.-I. (Eds.):
Biology and Control Theory: Current Challenges
589 p. 2007 [978-3-540-71987-8]

Vol. 356: Karatkevich A.:
Dynamic Analysis of Petri Net-Based Discrete Systems
166 p. 2007 [978-3-540-71464-4]

Vol. 355: Zhang H.; Xie L.:
Control and Estimation of Systems with
Input/Output Delays
213 p. 2007 [978-3-540-71118-6]

Vol. 354: Witczak M.:
Modelling and Estimation Strategies for Fault
Diagnosis of Non-Linear Systems
215 p. 2007 [978-3-540-71114-8]

Vol. 353: Bonivento C.; Isidori A.; Marconi L.; Rossi C. (Eds.)
Advances in Control Theory and Applications
305 p. 2007 [978-3-540-70700-4]

Vol. 352: Chiasson, J.; Loiseau, J.J. (Eds.)
Applications of Time Delay Systems
358 p. 2007 [978-3-540-49555-0]

Vol. 351: Lin, C.; Wang, Q.-G.; Lee, T.H., He, Y.
LMI Approach to Analysis and Control of
Takagi-Sugeno Fuzzy Systems with Time Delay
204 p. 2007 [978-3-540-49552-9]

Vol. 350: Bandyopadhyay, B.; Manjunath, T.C.; Umapathy, M.
Modeling, Control and Implementation of Smart Structures 250 p. 2007 [978-3-540-48393-9]

Vol. 349: Rogers, E.T.A.; Galkowski, K.;
Owens, D.H.
Control Systems Theory
and Applications for Linear
Repetitive Processes 482 p. 2007 [978-3-540-42663-9]

Vol. 347: Assawinchaichote, W.; Nguang, K.S.; Shi P.
Fuzzy Control and Filter Design
for Uncertain Fuzzy Systems
188 p. 2006 [978-3-540-37011-6]

Vol. 346: Tarbouriech, S.; Garcia, G.; Glattfelder, A.H. (Eds.)
Advanced Strategies in Control Systems
with Input and Output Constraints
480 p. 2006 [978-3-540-37009-3]

Vol. 345: Huang, D.-S.; Li, K.; Irwin, G.W. (Eds.)
Intelligent Computing in Signal Processing
and Pattern Recognition
1179 p. 2006 [978-3-540-37257-8]

Vol. 344: Huang, D.-S.; Li, K.; Irwin, G.W. (Eds.)
Intelligent Control and Automation
1121 p. 2006 [978-3-540-37255-4]

Vol. 341: Commault, C.; Marchand, N. (Eds.)
Positive Systems
448 p. 2006 [978-3-540-34771-2]
Vol. 340: Diehl, M.; Mombaur, K. (Eds.)
Fast Motions in Biomechanics and Robotics
500 p. 2006 [978-3-540-36118-3]

Vol. 339: Alamir, M.
Stabilization of Nonlinear Systems Using
Receding-horizon Control Schemes
325 p. 2006 [978-1-84628-470-0]

Vol. 338: Tokarzewski, J.
Finite Zeros in Discrete Time Control Systems
325 p. 2006 [978-3-540-33464-4]

Vol. 337: Blom, H.; Lygeros, J. (Eds.)
Stochastic Hybrid Systems
395 p. 2006 [978-3-540-33466-8]

Vol. 336: Pettersen, K.Y.; Gravdahl, J.T.;
Nijmeijer, H. (Eds.)
Group Coordination and Cooperative Control
310 p. 2006 [978-3-540-33468-2]

Vol. 335: Kozłowski, K. (Ed.)
Robot Motion and Control
424 p. 2006 [978-1-84628-404-5]

Vol. 334: Edwards, C.; Fossas Colet, E.;
Fridman, L. (Eds.)
Advances in Variable Structure and Sliding Mode Control
504 p. 2006 [978-3-540-32800-1]

Vol. 333: Banavar, R.N.; Sankaranarayanan, V.
Switched Finite Time Control of a Class of
Underactuated Systems
99 p. 2006 [978-3-540-32799-8]

Vol. 332: Xu, S.; Lam, J.
Robust Control and Filtering of Singular Systems
234 p. 2006 [978-3-540-32797-4]

Vol. 331: Antsaklis, P.J.; Tabuada, P. (Eds.)
Networked Embedded Sensing and Control
367 p. 2006 [978-3-540-32794-3]

Vol. 330: Koumoutsakos, P.; Mezic, I. (Eds.)
Control of Fluid Flow
200 p. 2006 [978-3-540-25140-8]

Vol. 329: Francis, B.A.; Smith, M.C.; Willems, J.C. (Eds.)
Control of Uncertain Systems: Modelling, Approximation, and Design
429 p. 2006 [978-3-540-31754-8]

Vol. 328: Loría, A.; Lamnabhi-Lagarrigue, F.; Panteley, E. (Eds.)
Advanced Topics in Control Systems Theory
305 p. 2006 [978-1-84628-313-0]

Vol. 327: Fournier, J.-D.; Grimm, J.; Leblond, J.; Partington, J.R. (Eds.)
Harmonic Analysis and Rational Approximation
301 p. 2006 [978-3-540-30922-2]

Vol. 326: Wang, H.-S.; Yung, C.-F.; Chang, F.-R.
H_∞ Control for Nonlinear Descriptor Systems
164 p. 2006 [978-1-84628-289-8]

Vol. 325: Amato, F.
Robust Control of Linear Systems Subject to Uncertain
Time-Varying Parameters
180 p. 2006 [978-3-540-23950-5]

Vol. 324: Christofides, P.; El-Farra, N.
Control of Nonlinear and Hybrid Process Systems
446 p. 2005 [978-3-540-28456-7]

Vol. 323: Bandyopadhyay, B.; Janardhanan, S.
Discrete-time Sliding Mode Control
147 p. 2005 [978-3-540-28140-5]

Vol. 322: Meurer, T.; Graichen, K.; Gilles, E.D. (Eds.)
Control and Observer Design for Nonlinear Finite and Infinite Dimensional Systems
422 p. 2005 [978-3-540-27938-9]

Vol. 321: Dayawansa, W.P.; Lindquist, A.;
Zhou, Y. (Eds.)
New Directions and Applications in Control Theory
400 p. 2005 [978-3-540-23953-6]

Vol. 320: Steffen, T.
Control Reconfiguration of Dynamical Systems
290 p. 2005 [978-3-540-25730-1]

Vol. 319: Hofbaur, M.W.
Hybrid Estimation of Complex Systems
148 p. 2005 [978-3-540-25727-1]

Vol. 318: Gershon, E.; Shaked, U.; Yaesh, I.
H_∞ Control and Estimation of State-multiplicative Linear Systems
256 p. 2005 [978-1-85233-997-5]

Vol. 317: Ma, C.; Wonham, M.
Nonblocking Supervisory Control of State Tree Structures
208 p. 2005 [978-3-540-25069-2]

Vol. 316: Patel, R.V.; Shadpey, F.
Control of Redundant Robot Manipulators
224 p. 2005 [978-3-540-25071-5]

Vol. 315: Herbordt, W.
Sound Capture for Human/Machine Interfaces: Practical Aspects of Microphone Array Signal Processing
286 p. 2005 [978-3-540-23954-3]

Vol. 314: Gil', M.I.
Explicit Stability Conditions for Continuous Systems
193 p. 2005 [978-3-540-23984-0]

Vol. 313: Li, Z.; Soh, Y.; Wen, C.
Switched and Impulsive Systems
277 p. 2005 [978-3-540-23952-9]

Vol. 312: Henrion, D.; Garulli, A. (Eds.)
Positive Polynomials in Control
313 p. 2005 [978-3-540-23948-2]

Vol. 311: Lamnabhi-Lagarrigue, F.; Loría, A.; Panteley, E. (Eds.)
Advanced Topics in Control Systems Theory
294 p. 2005 [978-1-85233-923-4]

Vol. 310: Janczak, A.
Identification of Nonlinear Systems Using Neural Networks and Polynomial Models
197 p. 2005 [978-3-540-23185-1]

Vol. 309: Kumar, V.; Leonard, N.; Morse, A.S. (Eds.)
Cooperative Control
301 p. 2005 [978-3-540-22861-5]

Vol. 308: Tarbouriech, S.; Abdallah, C.T.; Chiasson, J. (Eds.)
Advances in Communication Control Networks
358 p. 2005 [978-3-540-22819-6]

Vol. 307: Kwon, S.J.; Chung, W.K.
Perturbation Compensator based Robust Tracking Control and State Estimation of Mechanical Systems
158 p. 2004 [978-3-540-22077-0]

Vol. 306: Bien, Z.Z.; Stefanov, D. (Eds.)
Advances in Rehabilitation
472 p. 2004 [978-3-540-21986-6]

Vol. 305: Nebylov, A.
Ensuring Control Accuracy
256 p. 2004 [978-3-540-21876-0]

Vol. 304: Margaris, N.I.
Theory of the Non-linear Analog Phase Locked Loop
303 p. 2004 [978-3-540-21339-0]