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

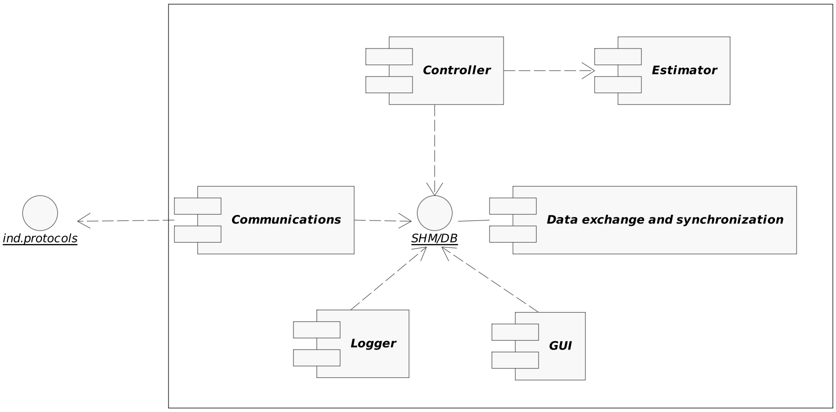

Fig. 1. NEWCON Component diagram

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]:

$$\min_{X,U,\epsilon} \Upsilon_i\big(Y, U, \epsilon\big) = \Psi_i\big(Y, U\big) + P_i(\epsilon) \tag{3}$$

$$\text{s.t. } u_{i+k} = u_{i+m-1}, \quad k = m, \cdots, p-1 \tag{4}$$

$$\bar{x}_{i+k} - \phi(\bar{x}_{i+k-1}, \bar{u}_{i+k-1}) = 0, \quad k = 1, \ldots, p \tag{5}$$

$$y_{\mathrm{sp},i+p} - y_{i+p} = 0 \tag{6}$$

$$X_\mathrm{L} - \epsilon_\mathrm{x} \leqslant X \leqslant X_\mathrm{U} + \epsilon_\mathrm{x} \tag{7}$$

$$Y_\mathrm{L} - \epsilon_\mathrm{y} \leqslant Y \leqslant Y_\mathrm{U} + \epsilon_\mathrm{y} \tag{8}$$

$$U_\mathrm{L} - \epsilon_\mathrm{u} \leqslant U \leqslant U_\mathrm{U} + \epsilon_\mathrm{u} \tag{9}$$

$$\Delta U_\mathrm{min} - \epsilon_{\Delta\mathrm{u}} \leqslant \Delta U \leqslant \Delta U_\mathrm{max} + \epsilon_{\Delta\mathrm{u}} \tag{10}$$

$$\epsilon \geqslant 0 \tag{11}$$

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

$$\Psi_i(Y,U) = \sum_{k=1}^{p} e_{i+k}^{\mathrm{T}} Q_{\mathrm{y}k} e_{i+k} + \sum_{k=1}^{m} \left(u_{i+k-1} - u_{\mathrm{r},i+k-1}\right)^{\mathrm{T}} Q_{\mathrm{u}k} \left(u_{i+k-1} - u_{\mathrm{r},i+k-1}\right)$$

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]

$$z_{i+k} = z_{i+k-1} + K_{\mathrm{I}}\left(y_{i+k} - y_{\mathrm{sp},i+k}\right), \quad k = 1, \ldots, p \tag{12}$$

with $z_i = z_0$, where $z_i \in \mathbb{R}^{n_o}$, $K_{\mathrm{I}} \in \mathbb{R}^{n_o \times n_o}$, 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 $\rightarrow$ 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

$$\frac{\mathrm{d}C_{\mathrm{A}}}{\mathrm{d}t} = \frac{F_0}{V}\,(C_{\mathrm{A0}} - C_{\mathrm{A}}) - k_0\,\mathrm{e}^{-E_{\mathrm{a}}/(R\,T_{\mathrm{r}})}\,. \tag{14}$$

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	$\rho\,,\ \rho_j$	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 $Q_{uk} = \mathrm{diag}(10^{-1}, 10^{-3})$, $k = 1, \cdots, p$.

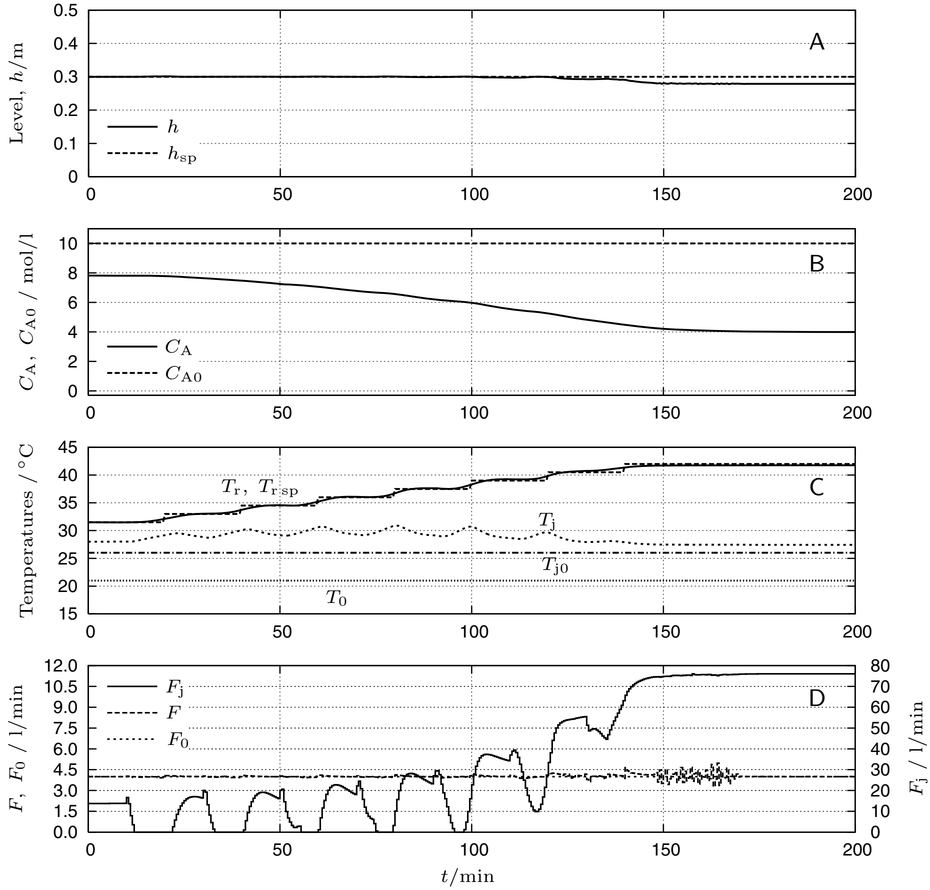

Fig. 2. Reactor closed loop response to a sequence of step changes in the reactor temperature set-point

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.