Sampled–Data Model Predictive Control Using Adaptive Time–Mesh Refinement Algorithms

Luís Tiago Paiva and Fernando A. C. C. Fontes

Abstract We address sampled–data nonlinear Model Predictive Control (MPC) schemes, in particular we address methods to efficiently and accurately solve the underlying continuous-time optimal control problems (OCP). In nonlinear OCPs, the number of discretization points is a major factor affecting the computational time. Also, the location of these points is a major factor affecting the accuracy of the solutions. We propose the use of an algorithm that iteratively finds the adequate time–mesh to satisfy some pre–defined error estimate on the obtained trajectories. The proposed adaptive time–mesh refinement algorithm provides local mesh resolution considering a time–dependent stopping criterion, enabling an higher accuracy in the initial parts of the receding horizon, which are more relevant to MPC. The results show the advantage of the proposed adaptive mesh strategy, which leads to results obtained approximately as fast as the ones given by a coarse equidistant–spaced mesh and as accurate as the ones given by a fine equidistant–spaced mesh.

Keywords Predictive control · Nonlinear systems · Optimal control · Real–time optimization · Continuous–time systems · Adaptive algorithms · Time–mesh refinement · Sampled-data systems

1 Introduction

This article discusses an adaptive time–mesh refinement algorithm to efficiently and accurately solve optimal control problems (OCP) and proposes its use in Model Predictive Control (MPC) schemes.

In the last decade, most of the MPC literature has been using discrete–time models (e.g. [12, 14, 27]). However, some of the earlier theoretical works on nonlinear MPC used continuous–time models (see [4, 9, 16, 17]). Recently, [13] proposed a multi–

L.T. Paiva (✉) · F.A.C.C. Fontes
Systec–ISR, Faculdade de Engenharia, Universidade do Porto, Porto, Portugal
e-mail: ltpaiva@fe.up.pt

F.A.C.C. Fontes
e-mail: faf@fe.up.pt

P. Garrido et al. (eds.), *CONTROLO 2016*, Lecture Notes
in Electrical Engineering 402, DOI 10.1007/978-3-319-43671-5_13

step MPC scheme, which despite still being a discrete–time scheme, has the optimization and feedback updates done at possibly different time instants. The authors show that such technique can lower the computational load while maintaining stability and quantifiable robust performance estimates. In [21], some of the authors that vastly contributed to spread the use of discrete–time MPC (c.f. [27]) show that a continuous–time method is significantly more efficient than standard discrete–time methods when solving constrained linear quadratic problems.

To implement MPC schemes, some form of discretization, or at least a finite parameterization, is eventually needed to solve the OCP. Nevertheless, there are several advantages in maintaining a continuous–time model until later stages. In addition to being able to obtain more accurate solution to OCPs faster, it might be essencial in nonlinear systems that can rapidly change behavior in some short time intervals or even when there is a discontinuity at some critical instant (e.g. impulsive control [10, 23]). Also, path-following MPC strategies are more complex to implement in discrete-time than in sampled-data schemes (compare [24, 25] with [3, 28]).

We also argue that the time is ripe to start using continuous–time MPC, even in applications [8, 28]. Many theoretical questions of using sampled–data systems are well documented in the literature [5–7, 11, 15, 26] and there are now several ready available software packages to solve nonlinear OCP [18].

In OCP solvers using direct collocation methods, the control and the state are discretized in an appropriately chosen mesh of the time interval. Then, the continuous–time OCP is transcribed into a finite–dimensional nonlinear programming problem (NLP) which can be solved using widely available software packages [18]. Most frequently, in the discretization procedure, regular time meshes having equidistant spacing are used. However, in some cases, these meshes are not the most adequate to deal with nonlinear behaviours. One way to improve the accuracy of the results, while maintaining reasonable computational time and memory requirement, is to construct a mesh having different time steps. The best location for the smaller steps sizes is, in general, not known a priori, so the mesh is refined iteratively. In a mesh–refinement procedure the problem is solved, typically, in an initial coarse uniform mesh in order to capture the basic structure of the solution and of the error. Then, this initial mesh is repeatedly refined according to a chosen strategy until some stopping criterion is attained. Several mesh refinement methods employing direct collocation methods have been described in the recent years [1, 2, 22, 29].

In this paper, we adapt and apply to an MPC context an adaptive time–mesh refinement algorithm to solve nonlinear OCP [20]. The algorithm computes iteratively an adequate time-mesh that satisfies some pre–defined error estimates on the obtained trajectories. The refinement method used here (a) permits several levels of refinement, obtaining a multi–level time–mesh in a single iteration. (b) it also permits different refinement criteria – the relative error of the primal variables, the relative error of the dual variables or a combination of both; (c) it considers distinct criteria for refining the mesh and for stopping the refinement procedure – the refinement strategy can be driven by the information given by the dual variables and it can be stopped according to the information given by the primal variables. As described in [20], there are advantages in choosing the error of the adjoint multipliers as a refinement criterion. To decrease the overall computational time, the solution computed in the previous iteration is used as a warm start in the next one, which proved to be of major importance to improve computational efficiency. This adaptive strategy leads to results with higher accuracy and yet with lower overall computational time, when compared to results obtained by meshes having equidistant spacing, as is the case when using discrete–time models from the beginning.

In MPC context, the prediction can be interpreted in the sense of planning. When we make plans for the future, we establish planning strategies with detail level depending on the prediction horizon. Combining this idea with the refinement strategy, we obtain an adaptive time–mesh refinement algorithm which generates meshes with higher concentration of node points in the beginning of the prediction horizon and less concentration of node points in the end of the same interval, enforcing the idea of having more nodes point where they are needed and keeping a low overall number of node points. This is an important issue, because we want to increase the accuracy of the solution without compromising CPU times.

2 The Adaptive Mesh Refinement Algorithm for Optimal Control Problems

Let us consider the optimal control problem:

$$\mathcal{P}(t_0, t_f) \, : \, \text{Minimise} \quad \int_{t_0}^{t_f} \mathrm{L}\,(t, \mathbf{x}(t), \mathbf{u}(t))\, \mathrm{d}t + \mathrm{G}\left(\mathbf{x}(t_f)\right) \tag{1}$$

$$\text{subject to} \quad \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)) \qquad \text{a.e. } t \in \left[t_0, t_f\right] , \tag{2}$$

$$\mathbf{x}(t_0) = \mathbf{x}_0 \, , \tag{3}$$

$$\mathbf{x}(t_f) \in \mathbb{X}_1 \subset \mathbb{R}^n \, , \tag{4}$$

$$\mathbf{x}(t) \in \mathbb{X} \subset \mathbb{R}^n \qquad \text{a.e. } t \in \left[t_0, t_f\right] , \tag{5}$$

$$\mathbf{u}(t) \in \mathbb{U} \subset \mathbb{R}^m \qquad \text{a.e. } t \in \left[t_0, t_f\right] , \tag{6}$$

where $\mathbf{x} \, : \, \left[t_0, t_f\right] \to \mathbb{R}^n$ is the state, $\mathbf{u} \, : \, \left[t_0, t_f\right] \to \mathbb{R}^m$ is the control and $t \in \left[t_0, t_f\right]$ is time. The functions involved comprise the running cost $\mathrm{L} \, : \, \left[t_0, t_f\right] \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$, the terminal cost $\mathrm{G} : \mathbb{R}^n \to \mathbb{R}$ and the system dynamics $\mathbf{f} \, : \, \left[t_0, t_f\right] \times \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$.

As stated in [19, 20], the adaptive mesh refinement process starts by discretizing the time interval $[t_0, t_f]$ in a coarse mesh used to solve the NLP problem associated to the OCP in order to catch the main structure of the solution. According to some refinement criteria, the mesh is divided in $K \in \mathbb{N}$ mesh intervals

$$\mathcal{S}_k = [\tau_{k-1}, \tau_k[\, , \quad k = 1, \ldots , K,$$

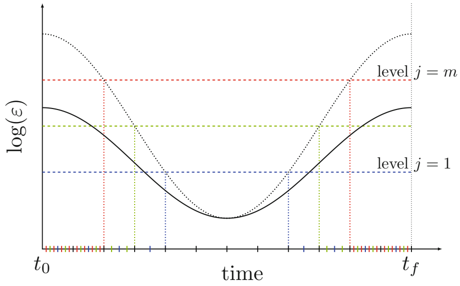

Fig. 1 Illustration of the multi–level adaptive time–mesh refinement strategy

where $(\tau_0, \dots, \tau_K)$ coincide with nodes. These mesh intervals $\mathcal{S}_k$ form a partition of the time interval while the mesh nodes have the property $\tau_0 < \tau_1 < \dots < \tau_K$.

The subintervals $\mathcal{S}_k$ that verify the refinement criteria are refined taking into account different levels of refinement in a single iteration, i.e., they are divided into smaller subintervals according to user–defined levels of refinement $\bar{\varepsilon} =, \left[\varepsilon_1,\ \varepsilon_2,\ \dots,\ \varepsilon_m\right]$. The procedure is repeated until the stopping criterion is achieved. A subinterval $\mathcal{S}_{k,j}$ is at the j^{th} level of refinement if

$$S_{k,j} = \left\{t \in S_k \ : \ \varepsilon(t) \in \left[\varepsilon_j, \varepsilon_{j+1}\right[\right\} \tag{7}$$

for $j = 1, \dots, m$. This procedure adds more node points to the subintervals in higher levels of refinement, corresponding to higher errors, and it adds less node points to those in lower refinement levels (Fig. 1). By defining several levels of refinement, we get a multi–level time–mesh in a single iteration.

3 The Model Predictive Control Framework

Consider a sampling step $\delta > 0$, the prediction horizon T and a sequence of sampling instants $\{t_i\}_{i\geq 0}$ with $t_{i+1} = t_i + \delta$. The sampled-data MPC algorithm follows the receding horizon strategy [9]:

1. Measure state of the plant $\mathbf{x}_{t_i}$;
2. Determine $\bar{\mathbf{u}} : \left[t_i, t_i + T\right] \to \mathbb{R}^m$ solution to the OCP $\mathcal{P}(t_i, t_i + T)$ (1)-(6).
3. Apply the control $\mathbf{u}^*(t) := \bar{\mathbf{u}}(t)$ to the plant in the interval $t \in \left[t_i, t_i + \delta\right]$, disregarding the remaining control $\bar{\mathbf{u}}(t), t > t_i + \delta$;
4. Repeat this procedure for the next sampling time instant $t_i + \delta$.

We extend the adaptive time–mesh refinement algorithm described in [20] in order to allow different refinement levels according to some partition of the time domain. This extension is of relevance in the MPC context, since it is desirable to have a solution with higher accuracy in the initial part of the horizon.

Motivation. The time interval $t \in [t_0, t_f]$, the prediction horizon T, and the sampling step $\delta > 0$, satisfy $\delta << T << t_f - t_0$. When applying the MPC procedure to solve an OCP, at each time instant t_i we compute the solution in $[t_i, t_i + T]$ but we just implement the open–loop control until $t_i + \delta$. Therefore, taking into account the planning strategy discussed above, it would be an improvement if we have an adaptive time–mesh able to cope this feature, i.e., a time–mesh that is highly refined in the lower limit of the time interval $[t_i, t_i + T]$ and it is coarser in the upper limit of the same interval. Then, we would implement the control on the time interval $[t_i, t_i + \delta]$ computed with high accuracy in a refined mesh. For the remaining time interval we have an estimate of the solution.

Time–Mesh Refinement Algorithm. In this extension, we also consider a time–dependent stopping criterion for the mesh refinement algorithm with different levels $\bar{\varepsilon}_{\mathbf{x}}(t)$. Instead of having a fixed stopping criterion $\varepsilon_{\mathbf{x}}^{\max}$, now we have a time–dependent $\bar{\varepsilon}_{\mathbf{x}}(t)$ stopping criterion which sets different levels of accuracy for the solution, along the time domain. For example, the time–dependent levels of refinement can be defined as

$$\bar{\varepsilon}_{\mathbf{x}}(t) = \begin{cases} \varepsilon_{\mathbf{x}}^{\max}, & t \in [t_i, t_i + \beta_1 T] \\ \alpha_1 \varepsilon_{\mathbf{x}}^{\max}, & t \in]t_i + \beta_1 T, t_i + \beta_2 T] \\ \cdots & \\ \alpha_j \varepsilon_{\mathbf{x}}^{\max}, & t \in]t_i + \beta_j T, t_i + T] \end{cases}$$

where $1 < \alpha_1 < \cdots < \alpha_j \leq \varepsilon_{\mathbf{x}}^{\max}$ and $0 < \beta_1 < \beta_2 < \cdots < \beta_j < 1$ are user–defined scalars.

This procedure adds more node points to the subintervals that are in lower levels of the stopping criterion for the refinement procedure, corresponding to time instants close to the initial time as illustrated in Fig. 2.

Refinement and Stopping Criteria. In order to proceed with the mesh refinement strategy, we have to define some refinement criteria and a stopping criterion. We consider as refinement criteria: the estimate of the relative error of the adjoint

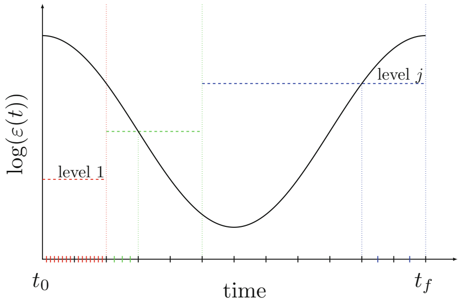

Fig. 2 Illustration of the extended (time–dependent) time–mesh refinement strategy

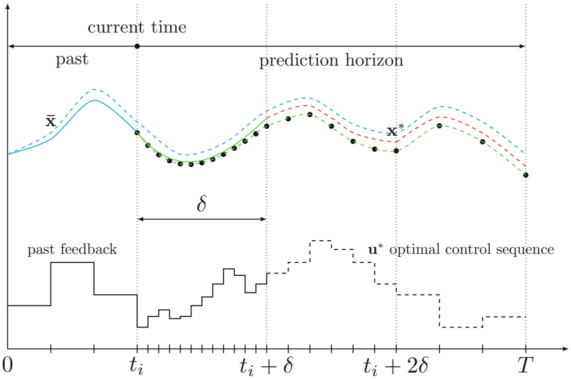

Fig. 3 Time–mesh refinement algorithm for MPC

multipliers (dual variables). For the stopping criterion, we consider a threshold for the relative error of the trajectory $\|\varepsilon_{\mathbf{x}}\|_\infty$.
Warm Start. Since we are solving a sequence of open–loop OCPs, to decrease the CPU time, when going from the problem in $[t_i, t_i + T]$ to the one in $[t_i + \delta, t_i + T$ $+\delta]$, the solution of the previous is used as a warm start for the problem. To create this warm start, the solution obtained in $[t_i, t_i + T]$ is projected in the new mesh in $[t_i + \delta, t_i + T + \delta]$ using the cubic Hermite interpolation.
Model Predictive Control coupled with the Extended Algorithm. We start the MPC procedure in the typical way but considering an adaptive mesh refinement strategy. We descritise the time interval $[t_0, t_f]$ and we solve our OCP in open–loop. Then, we implement the control in the first sampling interval. When starting the next MPC step, we apply the time–mesh refinement strategy in order to find the best mesh suited to the solve the OCP in the second sampling interval (Fig. 3). In the MPC algorithm, step 2 is modified as follows:

2. (a) Select the intervals $S_{k,j}$ to be refined according to the time–dependent levels of refinement $\bar{\varepsilon}_{\mathbf{x}}(t)$ and generate a new time grid;
   (b) Determine $\bar{\mathbf{u}} : [t_i, t_i + T] \to \mathbb{R}^m$ solution to the OCP $\mathcal{P}(t_i, t_i + T)$ (1)–(6), in the new time-grid;

4 Application: Parking Manoeuvres

In order to apply our MPC strategy, let us consider the car–like system problem with $t \in [0, 20]$, in seconds, $\mathbf{x}(t) = (x(t), y(t), \psi(t))$ and $\mathbf{u}(t) = (u(t), c(t))$. Aiming minimum energy, this problem ($\mathrm{P_{CP}}$) can be stated as:

$$\text{Minimise} \int_0^{20} u^2(t)\mathrm{d}t \tag{8}$$

$$\begin{aligned}
\text{subject to}\quad & \dot{x}(t) = u(t)\cos(\psi(t)) && \text{a.e. } t \in [0,20] \\
& \dot{y}(t) = u(t)\sin(\psi(t)) && \text{a.e. } t \in [0,20] \qquad (9)\\
& \dot{\psi}(t) = u(t)\,c(t) && \text{a.e. } t \in [0,20] \\
& \mathbf{x}(0) = \mathbf{x}_0\,, && \qquad (10)\\
& \mathbf{x}(20) \in \mathbb{X}_1, && \qquad (11)\\
& \mathbf{x}(t) \in \mathbb{X}\,, && \forall t \in \left[t_0, t_f\right] \qquad (12)\\
& -1 \le u(t) \le 1 && \text{a.e. } t \in [0,20] \qquad (13)\\
& -0.7 \le c(t) \le 0.7 && \text{a.e. } t \in [0,20] \qquad (14)
\end{aligned}$$

where $u(t)$ is the speed and $c(t)$ is the curvature. The end–point constraints are specified as

$$\mathbf{x}_0 = (x_0,\, y_0,\, \psi_0) = (1.5, 3.5, {}^{\pi}\!/_{2}) \tag{15}$$

$$\mathbb{X}_1 = \left\{ (x, y, \psi)\,:\, \left(x - x_f\right)^2 + \left(y - y_f\right)^2 + \left(\psi - \psi_f\right)^2 \le r^2 \right\} \tag{16}$$

where $r^2 = 0.1$, and $\mathbf{x}_f = (x_f,\, y_f,\, \psi_f) = (4, 0, 0)$ is a user–defined target point. Moreover, we define a pathwise state constraint (see Fig. 4) set $\mathbb{X}$ is the set of points $(x,\, y,\, \psi)$ satisfying

$$\begin{cases} -M \le y \le M & \text{if } x \in [x_0, x^*] \\ -b\,(\mathbf{x}) \le y \le b\,(\mathbf{x}) & \text{if } x \in [x^*, x^\star] \\ -m \le y \le m & \text{if } x \in [x^\star, x_f] \end{cases} \tag{17}$$

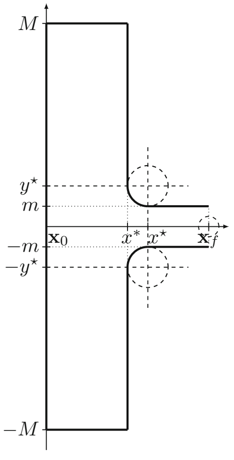

Fig. 4 Pathwise state constraints (17) for ($\mathrm{P_{CP}}$)

where $x^* = 2, x^\star = 3, y^\star = -1.5$, $M = 4$ and

$$b(\mathbf{x}) = y^\star - \sqrt{\rho^2 - (x - x^\star)^2}\,, \quad \rho = |x^\star - x^*|\,.$$

In order to test the MPC algorithm, we start by introducing some perturbations on the system dynamics test–plant:

$$\begin{cases} \dot{x}(t) = u(t)\left(1 + \delta_u\right)\cos(\psi(t)) \\ \dot{y}(t) = u(t)\left(1 + \delta_u\right)\sin(\psi(t)) \\ \dot{\psi}(t) = u(t)\left(1 + \delta_u\right)\, c(t)\left(1 + \delta_c\right) \end{cases} . \tag{18}$$

We consider $\delta = 2$ s which means that we will solve a sequence of 10 open–loop OCPs and we define $\delta_u = \delta_c = 0.1$. We also set $\varepsilon_{\mathbf{x}}^{\max} = 5 \times 10^{-5}$ and

$$\bar{\varepsilon}_{\mathbf{x}}(t) = \begin{cases} \varepsilon_{\mathbf{x}}^{\max}, & t \in \left[t_i, t_i + 0.1T\right] \\ 10 \times \varepsilon_{\mathbf{x}}^{\max}, & t \in \left]t_i + 0.1T, t_i + 0.3T\right] \\ 10^3 \times \varepsilon_{\mathbf{x}}^{\max}, & t \in \left]t_i + 0.3T, t_i + T\right] \end{cases}$$

This problem is solved considering three meshes:

(a) π_{ML}: the multi–level time–mesh refinement strategy with MPC;
(b) π_{F}: equidistant–spaced with the smallest time step of π_{ML};
(c) π_{C}: equidistant–spaced with the largest time step of π_{ML}.

As it can be seen in Fig. 5a, considering the mesh π_{ML}, the car–like system successfully stops when the terminal condition (16) is satisfied without violating any constraint. The sequence of solutions given by each sampling step on MPC is shown in Fig. 5b. The predictions are plotted with a dashed line, while the implemented controls are plotted with a solid line. Each segment is drawn with a different color representing different MPC sampling times.

The numerical results concerning the three meshes are shown in Table 1, which shows information about the number of nodes, the smallest time step, the number of iterations needed to solve the NLP problem, the maximum absolute local errors of the trajectory, and the CPU times for solving the OCP problem and for computing the local error as well.

According to Table 1, the mesh π_{ML} has only 11.4 % of the nodes of π_{F}, nevertheless both meshes have maximum absolute local error of the same order of magnitude. Computing the solution using π_{ML} takes less than 20 % of the time needed to get a solution using π_{F}, resulting in significant savings in memory and computational cost.

The mesh π_{C} is the initial coarse mesh considering equidistant spacing. Without applying our refinement strategy, the MPC produces a solution with lower accuracy, 1.261E^{-3}, when compared against the solution obtained via refinement procedure, 4.169E^{-5}. Moreover, the CPU time spent to compute solution using π_{ML} is, as expected, 50 % higher than the one spent to obtain a solution using π_{C}, however

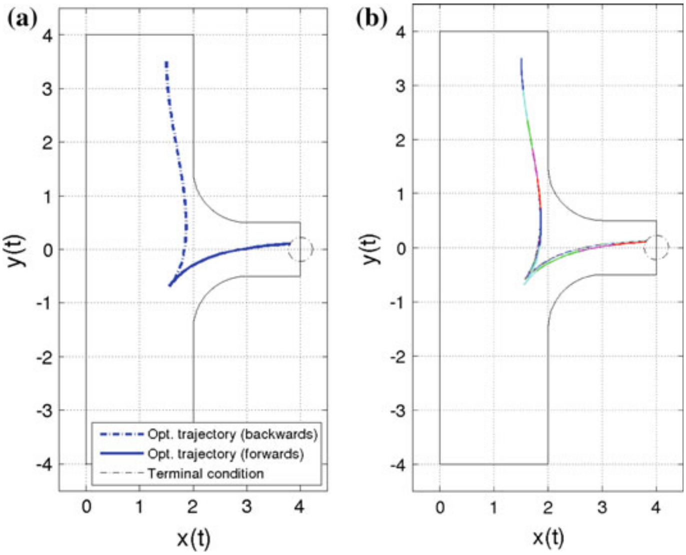

Fig. 5 Optimal trajectory for (P_{CP}). **a** MPC trajectory, **b** Sequence of optimal trajectories

Table 1 Comparing MPC results for the problem (P_{CP})

π_j	N_j	Δt_j	I_j	$\left\lVert \varepsilon_{\mathbf{x}}^{(j)} \right\rVert_\infty$	CPU time solver	(s) $\varepsilon_{\mathbf{x}}$
π_{ML}	365	$^1/_{3200}$	304|13|13|13|13|10|16|5|5|5	4.169E^{-5}	11.448	5.231
π_{F}	3201	$^1/_{3200}$	371|34|22|20|18|9|8|7|7|7	3.730E^{-5}	53.493	31.239
π_{C}	201	$^1/_{200}$	233|81|13|11|6|6|6|5|5|5	1.261E^{-3}	8.667	1.960

it is a good trade–off since the accuracy of the solution increases by two orders of magnitude. In all tests, the procedure gives the optimal solution which is computed spending a few seconds overall to solve 10 MPC steps.

The use of adaptive mesh refinement algorithm in real time optimization problems has additional benefits since it is possible to quickly obtain a solution even if the refinement procedure is interrupted at an early stage.

5 Conclusions

We develop an extended adaptive time–mesh refinement algorithm providing local mesh resolution refining only where it is required. In this extension, we consider a time–dependent stopping criterion for the mesh refinement algorithm with different levels $\bar{\varepsilon}(t)$. In the end, the OCPs are solved within MPC with an adapted mesh with local mesh resolution which has less nodes in the overall procedure, yet having maximum absolute local error of the same order of magnitude when compared against a refined mesh with equidistant–spacing.

Due to the fast response of the algorithm, it can be used to solve real–time optimization problems. The application demonstrates the advantage of the proposed adaptive mesh strategy, which leads to results obtained approximately as fast as the ones given by a coarse equidistant–spacing mesh and as accurate as the ones given by a fine equidistant–spacing mesh.

With this framework we can use continuous–time plant models directly. The discretization procedure can be automated and there is no need to select *a priori* the adequate time step.

Even if the optimization procedure is forced to stop in an early stage, as might be the case in real-time, we can still obtain a meaningful solution, although it might be a less accurate one.

Acknowledgments Research carried out while the 2nd author was a visiting scholar at Texas A&M University, College Station, USA. The support from Texas A&M and FEDER/COMPETE2020-POCI/FCT funds through grants POCI-01-0145-FEDER-006933 - SYSTEC and PTDC/EEI-AUT/2933/2014 is acknowledged.

References

1. Betts, J.T.: Practical methods for optimal control using nonlinear programming. SIAM (2001)
2. Betts, J.T., Huffman, W.P.: Mesh refinement in direct transcription methods for optimal control. Optim. Control Appl. Methods **19**(1), 1–21 (1998)
3. Caldeira, A., Fontes, F.: Model predictive control for path-following of nonholonomic systems. In: IFAC (ed.) Proceedings of the 10th Portuguese Conference in Automatic Control. pp. 374–379 (2010)
4. Chen, H., Allgöwer, F.: A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. Automatica **34**(10), 1205–1217 (1998)
5. Findeisen, R., Imsland, L., Allgöwer, F., Foss, B.: Towards a sampled-data theory for nonlinear model predictive control. In: Kang, W., Xiao, M., Borges, C. (eds.) New Trends in Nonlinear Dynamics and Control, and their applications. Lecture Notes in Control and Information Sciences, vol. 295, pp. 295–311. Springer Verlag, Berlin (2003)
6. Fontes, F.A.C.C., Magni, L.: Min-max model predictive control of nonlinear systems using discontinuous feedbacks. IEEE Trans. Autom. Control **48**, 1750–1755 (2003)
7. Fontes, F.A.C.C., Magni, L., Gyurkovics, E.: Sampled-data model predictive control for nonlinear time-varying systems: Stability and robustness. In: Allgöwer, F., Findeisen, R., Biegler, L. (eds.) Assessment and Future Directions of Nonlinear Model Predictive Control, Lecture Notes in Control and Information Systems, vol. 358, pp. 115–129. Springer Verlag (2007)
8. Fontes, F., Fontes, D., Caldeira, A.: Model Predictive Control of Vehicle Formations, Lecture Notes in Control and Information Sciences, vol. 381. Springer-Verlag (2009)
9. Fontes, F.A.C.C.: A general framework to design stabilizing nonlinear model predictive controllers. Syst. Control Lett. **42**(2), 127–143 (2001)
10. Fontes, F.A., Pereira, F.L.: Model predictive control of impulsive dynamical systems. In: Proceedings of the 4th IFAC conference on nonlinear model predictive control, NMPC'12. IFAC Proceedings Volumes (IFAC-PapersOnline), vol. 4, pp. 305–310. Noordwijkerhout; Netherlands (2012)

11. Grüne, L., Nesic, D., Pannek, J.: Model predictive control for nonlinear sampled-data systems. In: F. Allgöwer, L. Biegler, R.F.e. (eds.) Assessment and Future Directions of Nonlinear Model Predictive Control (NMPC05), Lecture Notes in Control and Information Sciences, vol. 358, pp. 105–113. Springer Verlag, Heidelberg (2007)
12. Grüne, L., Pannek, J.: Nonlinear model predictive control. Springer (2011)
13. Grüne, L., Palma, V.G.: Robustness of performance and stability for multistep and updated multistep MPC schemes. Discret. Contin. Dyn. Syst. **35**(9), 4385–4414 (2015)
14. Lazar, M., Allgower, F., Van den Hof, P., Cott, B. (eds.): 4th IFAC Conference on Nonlinear Model Predictive Control, NMPC'12. IFAC Proceedings Volumes (IFAC-PapersOnline), IFAC, Noordwijkerhout; Netherlands (2012)
15. Magni, L., Scattolini, R.: Model predictive control of continuous-time nonlinear systems with piecewise constant control. IEEE Trans. Autom. Control **49**, 900–906 (2004)
16. Mayne, D.Q., Michalska, H.: Receding horizon control of nonlinear systems. IEEE Trans. Autom. Control **35**, 814–824 (1990)
17. Michalska, H., Mayne, D.Q.: Robust receding horizon control of constrained nonlinear systems. IEEE Trans. Autom. Control **38**, 1623–1633 (1993)
18. Paiva, L.T.: Optimal control in constrained and hybrid nonlinear system: Solvers and interfaces. Faculdade de Engenharia, Universidade do Porto, Technical Report (2013)
19. Paiva, L.T.: Numerical Methods in Optimal Control and Model Predictive Control. Ph.D., Universidade do Porto, Porto, Portugal (Dec 2014)
20. Paiva, L.T., Fontes, F.A.: Adaptive time-mesh refinement in optimal control problems with state constraints. Discret. Contin. Dyn. Syst. **35**(9), 4553–4572 (2015)
21. Pannocchia, G., Rawlings, J., Mayne, D., Mancuso, G.: Whither discrete time model predictive control? IEEE Trans. Autom. Control **60**(1), 246–252 (2015)
22. Patterson, M.A., Hager, W.W., Rao, A.V.: A ph mesh refinement method for optimal control. Optimal Control Applications and Methods (Feb 2014)
23. Pereira, F.L., Fontes, F.A., Aguiar, A.P., de Sousa, J.B.: An optimization-based framework for impulsive control systems. In: Developments in Model-Based Optimization and Control, pp. 277–300. Springer (2015)
24. Prodan, I., Olaru, S., Fontes, F.A., Pereira, F.L., de Sousa, J.B., Maniu, C.S., Niculescu, S.I.: Predictive control for path-following. from trajectory generation to the parametrization of the discrete tracking sequences. In: Developments in Model-Based Optimization and Control, pp. 161–181. Springer (2015)
25. Prodan, I., Olaru, S., Fontes, F.A., Stoica, C., Niculescu, S.I.: A predictive control-based algorithm for path following of autonomous aerial vehicles. In: 2013 IEEE International Conference on Control Applications (CCA), pp. 1042–1047. IEEE (2013)
26. Raković, S.V., Kouramas, K.: Invariant approximations of the minimal robust positively invariant set via finite time Aumann integrals. In: 2007 46th IEEE Conference on Decision and Control, pp. 194–199. IEEE (2007)
27. Rawlings, J.B., Mayne, D.Q.: Model Predictive Control: Theory and Design. Nob Hill Pub. (Aug 2009)
28. Rucco, A., Aguiar, A.P., Fontes, F.A., Pereira, F.L., de Sousa, J.B.: A model predictive control-based architecture for cooperative path-following of multiple unmanned aerial vehicles. In: Developments in Model-Based Optimization and Control, pp. 141–160. Springer (2015)
29. Zhao, Y., Tsiotras, P.: Density functions for mesh refinement in numerical optimal control. J. Guid. Control Dyn. **34**(1), 271–277 (2011)