TECHNISCHE UNIVERSITÄT WIEN

AUTOMATION & CONTROL INSTITUTE
INSTITUT FÜR AUTOMATISIERUNGS- & REGELUNGSTECHNIK

This document contains a post-print version of the paper

A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results

authored by **A. Steinboeck, M. Guay, and A. Kugi**

and published in *International Journal of Control*.

The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing. Please, scroll down for the article.

Cite this article as:

A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299

BibTex entry:

```
@article{acinpaper,
author = {Steinboeck, A. and Guay, M. and Kugi, A.},
title = {A design technique for fast sampled-data nonlinear model predictive control with convergence and
    stability results},
journal = {International Journal of Control},
volume = {93},
number = {1},
pages = {81--97},
year = {2020},
doi = {10.1080/00207179.2017.1346299},
url = {http://www.tandfonline.com/doi/full/10.1080/00207179.2017.1346299}
}
```

Link to original paper:

http://dx.doi.org/10.1080/00207179.2017.1346299
http://www.tandfonline.com/doi/full/10.1080/00207179.2017.1346299

Read more ACIN papers or get this document:

http://www.acin.tuwien.ac.at/literature

Contact:

Automation and Control Institute (ACIN)
TU Wien
Gusshausstrasse 27-29/E376
1040 Vienna, Austria

Internet: www.acin.tuwien.ac.at
E-mail: office@acin.tuwien.ac.at
Phone: +43 1 58801 37601
Fax: +43 1 58801 37699

Copyright notice:

This is an authors' accepted manuscript of the article A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299 published in *International Journal of Control*, copyright © Taylor & Francis Group, LLC, available online at: http://dx.doi.org/10.1080/00207179.2017.1346299

To appear in the *International Journal of Control*
Vol. 00, No. 00, Month 20XX, 1–25

ARTICLE

A Design Technique for Fast Sampled-Data Nonlinear Model Predictive Control with Convergence and Stability Results

Andreas STEINBOECK[a*], Martin GUAY[b] and Andreas KUGI[a]

[a] *Automation and Control Institute, Vienna University of Technology, 1040 Vienna, Austria*
[b] *Department of Chemical Engineering, Queen's University, Kingston, Ontario, Canada K7L 3N6*

(*Received 00 Month 20XX; accepted 00 Month 20XX*)

In this study, a sampled-data nonlinear model predictive control scheme is developed. The control algorithm uses a prediction horizon with variable length, a terminal constraint set, and a feedback controller defined on this set. Following a suboptimal solution strategy, a defined number of steps of an iterative optimization routine improve the current input trajectory at each sampling point. The value of the objective function monotonically decreases and the state converges to a target set. A discrete-time formulation of the algorithm and a discrete-time design model ensure high computational efficiency and avoid an ad-hoc quasi-continuous implementation. This design technique for a fast sampled-data nonlinear model predictive control algorithm is the main contribution of the paper. Based on a benchmark control problem, the performance of the developed control algorithm is assessed against state-of-the-art nonlinear model predictive control methods available in the literature. This assessment demonstrates that the developed control algorithm stabilizes the system with very low computational effort. Hence, the algorithm is suitable for real-time control of fast dynamical systems.

Keywords: Nonlinear model predictive control, receding horizon control, real-time optimization, sampled-data control, dynamic optimization

1. Introduction

High computational load is still a challenge for real-time nonlinear model predictive control (NMPC) of fast dynamical systems. There are two strategies of overcoming this problem. One can use powerful hardware (which may not yet be available at cost-effective prices). Alternatively, one can improve the controller software to reduce the computational effort. One common strategy to improve the computational requirements of NMPC is to compute approximate solutions of the dynamic optimization problem. Such algorithmic strategies (e.g., Diehl et al., 2011; Gros et al., 2016; Zanelli et al., 2016) are typically referred to as suboptimal NMPC techniques.

Given the complexity of solving nonlinear dynamic optimization problems in real-time, as required in NMPC, suboptimal strategies form the basis of virtually all existing NMPC techniques. In (Michalska and Mayne, 1993), a dual-mode receding horizon controller (RHC) with a terminal constraint set is proposed. As soon as the state trajectory enters this set, a continuous-time linear feedback control law is used and the RHC is switched off. Michalska and Mayne (1993) were among the first to recognize that the use of a terminal constraint set permits a suboptimal solution of the corresponding optimization problem.

A real-time NMPC scheme for discrete-time systems is presented in (Diehl et al., 2005). In this technique, the computational load is minimized by performing just a single Newton-type iteration

*Corresponding author: andreas.steinboeck@tuwien.ac.at

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

at each sampling point. Nominal stability of the scheme applied to discrete-time systems can be proven based on the assumption that the disturbance of the optimization routine due to the shift of the receding horizon is sufficiently small. Then, the objective function exhibits the descend property required by Lyapunov stability theory. To ensure that this disturbance is small for fast systems, one has to rely on short sampling times and hence, a large number of predicted input values are required in the dynamic optimization problem. The similarities and differences of the method and linear MPC as well as details of the implementation like discretization methods are discussed in (Gros et al., 2016).

Zanelli et al. (2016) proposed a discrete-time NMPC scheme that is suboptimal in the sense that the approximation based on a linear time-invariant system representation is (numerically exactly) solved rather than the original nonlinear optimization problem. Nominal local stability of the control scheme is proven and the values of the objective function evaluated with both the optimal and the suboptimal (approximate) solution are compared.

In (Nešić and Grüne, 2006), an unconstrained MPC is used to minimize the mismatch between the solutions of a sampled-data controlled system and a continuous-time closed-loop reference system. The approach requires a stabilizing continuous-time feedback control law to be known for the whole state space.

For a linear time-invariant discrete-time system with polytopic input and state constraints, Zeilinger et al. (2011) developed a suboptimal MPC strategy that combines ideas from explicit MPC and online optimization. A piecewise affine approximation of the optimal solution is computed offline, stored, and then used as warm-start solution of an online executed active set linear programming procedure. Zeilinger et al. (2011) also analyzed the trade-off between the accuracy of the piecewise affine approximation calculated offline and the online computational effort.

Adetola and Guay (2010) proposed a combined real-time optimization and MPC system to stabilize a constrained nonlinear system with unknown parameters. These parameters are found by an adaptive estimation technique. An optimum setpoint in terms of an economic objective function is computed by extremum seeking using barrier functions to incorporate state constraints. To stabilize the system at the (continuously) optimized setpoint, a nonlinear optimization problem is solved in an MPC framework. The method shows advantages in terms of robustness but entails a high computational effort.

A real-time NMPC framework was developed in (DeHaan and Guay, 2007) for continuous-time systems using a terminal constraint set with a stabilizing feedback controller and a suboptimal solution of the optimization problem was considered. With an extra differential equation, the parameters defining the system input are ensured to evolve along a direction that decreases the objective value. The descent direction of the objective function is computed using sensitivity equations.

Feller and Ebenbauer (2013) proposed a continuous-time linear MPC method with convex barrier functions (cf. Wills and Heath, 2004) and ellipsoidal terminal constraint sets. Similar to (DeHaan and Guay, 2007), parameters that define the piecewise constant system input evolve according to an extra differential equation that ensures a continuous decrease of the objective function based on a Newton-type descent direction. Feller and Ebenbauer (2014a) showed that using the same idea with polytopic terminal constraint sets can considerably extend the region of attraction of the controller. The problem of (intermediate) infeasibility of the continuous-time trajectory between two sampling points of the discrete-time system is solved by means of relaxed barrier functions in (Feller and Ebenbauer, 2014b).

Since barrier functions can shift the position of the optimal solution of the underlying optimization problem, gradient-recentered barrier functions are used in (DeHaan and Guay, 2007; Feller and Ebenbauer, 2013, 2014a,b; Wills and Heath, 2004). In contrast, Feller and Ebenbauer (2015) recenter their barrier functions by multiplication with weighting factors that are computed by a tailored quadratic program.

A continuous-time suboptimal NMPC scheme without any terminal constraints was designed by Graichen and Kugi (2010). They computed the minimum number of iterations of the underlying

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

optimization algorithm required to ensure exponential stability. In (Graichen, 2012b), this idea was used in an iterative optimization algorithm for affine-input systems without state constraints. In each iteration, the optimal control is computed based on Pontryagin's maximum principle. The same strategy was used in (Graichen, 2012a) to control a general nonlinear system with input box constraints and without state constraints.

An unconstrained suboptimal discrete-time NMPC scheme was proposed in (Grüne and Pannek, 2010). To ensure stability based on a Lyapunov-type inequality, a customized termination criterion of the optimization algorithm was used. However, the number of necessary iterations is not known a priori. It is worth noting that the studies (Graichen and Kugi, 2010) and (Grüne and Pannek, 2010) are among the first to provide the stability properties of suboptimal NMPC without terminal constraints.

Worthmann et al. (2014) analyzed the influence of the sampling time in a sampled-data NMPC scheme without terminal constraints and terminal costs on stability and performance of the closed-loop system. Based on the NMPC objective function value and the theoretically achievable objective function value, the degree of suboptimality entailed by sampling is analyzed. The results allow the determination of a minimum required sampling rate to reach a defined performance bound, i. e., a trade-off between sampling time and performance can be specified.

In the literature, so-called *fast NMPC* is often confused with *suboptimal NMPC*. The studies cited above focus on the latter. Fast NMPC techniques (like that presented in (Biegler et al., 2015; Zavala et al., 2008)) are tailored optimization algorithms that are applied to the solution of *NMPC-inspired* optimization problems. The techniques proposed in (Biegler et al., 2015; Zavala et al., 2008) use intermittent full solutions of a dynamic optimization problem along with fast sensitivity updates. Biegler et al. (2015) solve interior-point nonlinear programming problems based on predicted system states and perform a fast sensitivity update of the solution based on a local linearization and the estimated actual system state at the respective sampling point. If the optimal solution of the nonlinear programming problem cannot be computed within one sampling interval, multi-step NMPC strategies (Yang and Biegler, 2013) can be used, where sensitivity updates are also performed at intermediate sampling points when the optimization algorithm is still running. Generally, such fast NMPC techniques can be classified as *control-inspired* optimization approaches. It is clear that tailored powerful dynamic optimization code can provide advantages in terms of computational effort if optimality is required.

Given the extensive literature on both continuous-time and discrete-time MPC algorithms, it is interesting that little has been published on suboptimal real-time NMPC in a sampled-data context. We believe that this category of MPC algorithms is highly relevant in practical applications. This motivates our work and this was also the motivation for DeHaan and Guay (2007), who formulated a continuous-time control algorithm. However, any computer implementation requires some sort of time-discretization and allows thus only a *quasi-continuous* realization of the algorithm. In this article, we explore how the idea of DeHaan and Guay (2007) can be algorithmically extended and improved using a systematic discrete-time design. In Section 7, we summarize why this algorithmic extension from continuous-time to discrete-time control design is not obvious and why it is beneficial for control performance and computational demands. The main contributions of this work are as follows:

- The developed design method systematically considers the discrete-time character of the control concept. This concerns a tailored input parametrization, a terminal set feedback controller (sampled-data LQR design), the prediction model, and the NMPC calculations themselves.
- Compared to existing continuous-time approaches, the proposed discrete-time algorithm exhibits benign numerical properties, requires less restrictive assumptions, and entails a significantly reduced computational effort.
- The low computational effort is achieved mainly because of a tailored low-dimensional prob-

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

lem formulation, an efficient gradient calculation, and the use of a suboptimal solution strategy of the corresponding optimization problem.
- The low computational effort, which is verified in a benchmark example problem, renders the method suitable for real-time NMPC of systems with fast dynamics.

The paper is organized as follows: In Section 2, we develop a sampled-data nonlinear RHC algorithm with (variable) finite horizon length and a terminal constraint set. We start with a description of the considered plant and control problem, describe control update functions, and finally formulate a discrete-time problem statement. In Section 3, we describe fundamental update functions, where the time grid defining the system input changes. Incremental update functions, which do not change this time grid and which follow a suboptimal solution strategy, are outlined in Section 4. In Section 5, we describe the design of a feedback control law for extending the prediction horizon into the future. The proposed control algorithm is validated and compared to existing NMPC approaches based on a benchmark example problem in Section 6. Conclusions are drawn in Section 7. In Appendix A, we describe the required assumptions of the controller and discuss the convergence properties of the control scheme. In Appendix B, we propose an alternative update strategy for the time grid defining the system input.

In this paper, the following notations are used: For an arbitrary set $\mathcal{S}$, $\mathring{\mathcal{S}}$ is the open interior of $\mathcal{S}$ and $\partial\mathcal{S}$ is its boundary. Whenever confusion is ruled out, we will omit the argument t denoting the time variable.

2. Receding horizon control considered as a sampled-data scheme

In this section, we develop a general NMPC formulation with variable horizon length and a terminal constraint set.

Similar to Chen and Allgöwer (1998b), we utilize a feedback control law defined on the terminal constraint set as a vehicle for proving closed-loop stability and for extending the prediction horizon into the future.

2.1 *Continuous-time problem statement*

Consider the nonlinear continuous-time dynamical system

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \qquad \forall\, t \geq 0 \tag{1}$$

with the state

$$\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n, \tag{2a}$$

the input

$$\boldsymbol{u} \in \mathcal{U} \subseteq \mathbb{R}^m, \tag{2b}$$

and the initial state $\boldsymbol{x}(0) = \boldsymbol{x}_0 \in \mathring{\mathcal{X}}$. The states and inputs are constrained to the closed and simply connected sets $\mathcal{X}$ and $\mathcal{U}$, respectively, with nonempty interiors containing the origin, i. e., $\{\boldsymbol{0}\} \subset \mathring{\mathcal{X}}$ and $\{\boldsymbol{0}\} \subset \mathring{\mathcal{U}}$. We assume that $\boldsymbol{f} : \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$ is sufficiently often differentiable for the methods used in this paper and satisfies $\boldsymbol{0} = \boldsymbol{f}(\boldsymbol{0}, \boldsymbol{0})$.

The objective is to stabilize the system to a closed and simply connected *target set* $\mathcal{X}_d \subset \mathring{\mathcal{X}}$. The target set $\mathcal{X}_d$ satisfies $\boldsymbol{0} \in \mathcal{X}_d$ and, depending on the problem, may be $\mathcal{X}_d = \{\boldsymbol{0}\}$. To solve this

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

control problem, we first introduce an additional closed and simply connected set $\mathcal{X}_f$ satisfying $\mathcal{X}_d \subset \mathcal{X}_f \subseteq \mathcal{X}$. $\mathcal{X}_f$ is also referred to as *terminal set* (Alamir, 2006).

We introduce a time grid $T_0, T_1, \ldots, T_N$ for a horizon $[T_0, T_N)$. If not stated otherwise, $T_N < \infty$ and $N < \infty$. This time grid may change over time and may have non-uniform step sizes. Based on this time grid, the continuous-time input $\boldsymbol{u}(t)$ is defined over the horizon $[T_0, T_N)$ as follows

$$\begin{aligned} \boldsymbol{u}(t) = \bar{\boldsymbol{w}}(t, T_{k-1}, T_k, \boldsymbol{\vartheta}_k) \\ \forall\, t \in [T_{k-1}, T_k), k = 1, 2, \ldots, N \end{aligned} \tag{3}$$

with $\bar{\boldsymbol{w}} : \mathbb{R}\times\mathbb{R}\times\mathbb{R}\times\mathbb{R}^{n_\vartheta} \to \mathbb{R}^m$ and parameters assembled in the vectors $\boldsymbol{\vartheta}_k \in \mathbb{R}^{n_\vartheta}$ $(k = 1, 2, \ldots, N)$. We assume that $\bar{\boldsymbol{w}}$ is a vector-valued function that is sufficiently often differentiable for the methods used in this paper.

The main motivation for the parameterization (3) is to have a low-dimensional discrete-time input definition, which is tailored to the respective application and useful for the control problem. The choice of the time grid $T_0, T_1, \ldots, T_N$ directly influences the dimension of the control problem associated with the horizon $[T_0, T_N)$. This choice requires a compromise between the achievable control accuracy and the computational complexity.

We assume that the parameterization of $\boldsymbol{u}(t)$ can always be mapped to a finer time grid without loss of accuracy. That is, if the interval $[T_{k-1}, T_k)$ is split into subintervals $[T_{k-1}, T_{k'})$ and $[T_{k'}, T_k)$ with some $T_{k'}$ that satisfies $T_{k-1} < T_{k'} < T_k$, there always exist new parameter vectors $\boldsymbol{\vartheta}'_k$ and $\boldsymbol{\vartheta}'_{k'}$ so that

$$\begin{aligned} &\bar{\boldsymbol{w}}(t, T_{k-1}, T_k, \boldsymbol{\vartheta}_k) \\ &\quad = \begin{cases} \bar{\boldsymbol{w}}(t, T_{k-1}, T_{k'}, \boldsymbol{\vartheta}'_{k'}) & \text{if } t \in [T_{k-1}, T_{k'}) \\ \bar{\boldsymbol{w}}(t, T_{k'}, T_k, \boldsymbol{\vartheta}'_k) & \text{if } t \in [T_{k'}, T_k) \end{cases}. \end{aligned}$$

For a compact notation, we consider the vectors $\boldsymbol{T} = [T_0, T_1, \ldots, T_N]$ and $\boldsymbol{\theta} = [\boldsymbol{\vartheta}_1^T, \ldots, \boldsymbol{\vartheta}_N^T]^T$, and provide a joint definition of $\bar{\boldsymbol{w}}$ from (3) for the whole horizon $[T_0, T_N)$ in the form

$$\boldsymbol{u}(t) = \boldsymbol{w}(t, \boldsymbol{T}, \boldsymbol{\theta}) \quad \forall\, t \in [T_0, T_N). \tag{4}$$

Given this input parameterization, the search space of the optimal control problem is finite-dimensional. To ensure satisfaction of (2b), we introduce the closed set

$$\mathcal{T}(\boldsymbol{T}) = \{\boldsymbol{\theta} \in \mathbb{R}^{Nn_\vartheta} \,|\, \boldsymbol{w}(t, \boldsymbol{T}, \boldsymbol{\theta}) \in \mathcal{U}\ \forall\, t \in [T_0, T_N)\}.$$

That is, the set of admissible parameterized continuous-time inputs $\boldsymbol{w}(t, \boldsymbol{T}, \boldsymbol{\theta})$ is converted into a set $\mathcal{T}(\boldsymbol{T})$ of admissible input parameters $\boldsymbol{\theta}$ for a time grid defined by $\boldsymbol{T}$. We denote by $\hat{\boldsymbol{x}}_t(\tau)$ the trajectory of (1) in the horizon $\tau \in [t, T_N]$ with initial condition $\hat{\boldsymbol{x}}_t(t) = \boldsymbol{x}(t)$ and the input $\boldsymbol{u}(\tau)$. In many cases, $\hat{\boldsymbol{x}}_t(\tau)$ will be a predicted trajectory. If the input is given by (4), we shall write $\hat{\boldsymbol{x}}_t^{\boldsymbol{T},\boldsymbol{\theta}}(\tau)$.

To solve the stated control problem, we pose the objective function

$$J(\hat{\boldsymbol{x}}_t(\,\cdot\,), \boldsymbol{u}(\,\cdot\,), t, \boldsymbol{T}) = V(\hat{\boldsymbol{x}}_t(T_N)) + \int_t^{T_N} v(\hat{\boldsymbol{x}}_t(\tau), \boldsymbol{u}(\tau))\mathrm{d}\tau \tag{5}$$

defined on some horizon $[t, T_N]$ and subject to (2b). The design of J and the cost functions V and v depends on the primary control objective of reaching the target set $\mathcal{X}_d$. However, it may also reflect some secondary control objectives, like minimum consumption of resources. We assume that V and v are barrier functions, which ensure $\hat{\boldsymbol{x}}_t(T_N) \in \mathring{\mathcal{X}}_f$ and $\hat{\boldsymbol{x}}_t(\tau) \in \mathring{\mathcal{X}}\ \forall\, \tau \in [t, T_N]$, respectively (see

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

also Appendix A.1). These assumptions and terminal feedback control laws that will be introduced in Section 3.1 facilitate a proof of the closed-loop stability in the form of a descent condition of J (Lyapunov-type inequality), which is shown in Appendix A. If $\boldsymbol{T}$ were chosen beforehand, $\boldsymbol{\theta}$ could be found by solving the optimal control problem

$$\min_{\boldsymbol{\theta}\in\mathcal{T}(\boldsymbol{T})} \; J(\hat{\boldsymbol{x}}_t^{\boldsymbol{T},\boldsymbol{\theta}}(\,\cdot\,), \boldsymbol{w}(\,\cdot\,,\boldsymbol{T},\boldsymbol{\theta}), t, \boldsymbol{T}). \tag{6}$$

Let us assume that the functions $V : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$, $v : \mathcal{X}\times\mathcal{U} \to \mathbb{R}_{\geq 0}$, and $\boldsymbol{w} : \mathbb{R}\times\mathbb{R}^{N+1}\times\mathbb{R}^{Nn_\vartheta} \to \mathbb{R}^m$ are sufficiently often differentiable for the method used for solving (6). For instance, they have to belong to the class $\mathcal{C}^1$ for gradient based iterative search methods or to the class $\mathcal{C}^2$ if the Hessian is also used.

2.2 *Control update function*

The input defined by the pair $(\boldsymbol{T},\boldsymbol{\theta})$ can be used in the interval $[T_0, T_N)$. That is, a new set of parameters $(\boldsymbol{T},\boldsymbol{\theta})$ has to be found at least at the point $t = T_N$. However, as usual in RHC, a set of input parameters is typically only used during a small control horizon and the update of $(\boldsymbol{T},\boldsymbol{\theta})$ occurs much earlier, say at the point $t = \tau_i \in [T_0, T_N)$. The update times τ_i define a time grid $\tau_0 = 0$, τ_1, τ_2, ... with the step sizes $\Delta\tau_i = \tau_{i+1} - \tau_i$, which may be non-uniform. The step size is required to satisfy $\Delta\tau_i > \Delta\tau_{\min}$ with some strictly positive constant $\Delta\tau_{\min} > 0$. This requirement implies $\lim_{i\to\infty}\tau_i = \infty$. By $\boldsymbol{T}_{i+1} = [T_{i+1|0}, T_{i+1|1}, \ldots, T_{i+1|N_{i+1}}]$ and $\boldsymbol{\theta}_{i+1} = [\boldsymbol{\vartheta}_{i+1|1}^T, \ldots, \boldsymbol{\vartheta}_{i+1|N_{i+1}}^T]^T$, we denote the parameters generated at τ_i and used in the control horizon $[\tau_i, \tau_{i+1})$. Let an update of the input at τ_i be done by the *control update function*

$$(\boldsymbol{T}_{i+1}, \boldsymbol{\theta}_{i+1}) = \boldsymbol{U}(\tau_i, \boldsymbol{x}(\tau_i), (\boldsymbol{T}_i, \boldsymbol{\theta}_i)).$$

In many cases, the computation of $\boldsymbol{U}$ involves the solution of an optimal control problem, in which case the approach can be classified as NMPC. Note that $\boldsymbol{U}$ may also involve the suboptimal solution of an optimal control problem.

Appendix A.2 describes properties of the recurrently applied function $\boldsymbol{U}$ that are sufficient to achieve the stated control objective and to ensure closed-loop stability. At some update time τ_i, finding an update function $\boldsymbol{U}$ that has these properties is trivial if $T_{i+1|N_{i+1}} = T_{i|N_i}$ or if $T_{i+1|N_{i+1}} < T_{i|N_i}$ and $\hat{\boldsymbol{x}}_{\tau_{i-1}}^{\boldsymbol{T}_i,\boldsymbol{\theta}_i}(T_{i+1|N_{i+1}}) \in \mathring{\mathcal{X}}_f$. In these special cases, the previous solution could be used at the current step. Nevertheless, even in these cases, the available computational resources should be utilized to search for a solution of the optimal control problem that is better than the previous solution. This will improve the performance of the closed-loop control system and will be the focus of Section 4. The essential difficulty in the design of $\boldsymbol{U}$ occurs when $T_{i+1|N_{i+1}} > T_{i|N_i}$, i. e., when the prediction horizon grows. This situation will be treated in detail in Section 3. We will call an update function *incremental* if $\boldsymbol{T}_{i+1} = \boldsymbol{T}_i$ and *fundamental* otherwise. Fundamental and incremental update functions will thus be discussed in Sections 3 and 4, respectively.

In the control update function $\boldsymbol{U}$, a feasible and in many cases very reasonable initial guess for those components of $\boldsymbol{T}_{i+1}$ and $\boldsymbol{\theta}_{i+1}$ that concern the (overlapping) interval $[\tau_i, \min\{T_{i|N_i}, T_{i+1|N_{i+1}}\})$ is found by adopting the previous control $\boldsymbol{w}(t, \boldsymbol{T}_i, \boldsymbol{\theta}_i)$. According to the assumptions concerning the parametrization of $\boldsymbol{u}(t)$ stipulated in Section 2.1, this can always be done exactly, i. e., without any approximation. With this consideration in mind, we will suggest a set of possible update functions in the following sections. At any update time τ_i, they can be combined at will as long as

$$T_{i+1|0} \leq \tau_i < \tau_{i+1} < T_{i+1|N_{i+1}} \tag{7}$$

is satisfied. For instance, an update function $\boldsymbol{U}$ can also be repeated in the form $(\boldsymbol{T}_{i+1}, \boldsymbol{\theta}_{i+1}) =$

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

$\boldsymbol{U}(\tau_i, \boldsymbol{x}(\tau_i), \boldsymbol{U}(\tau_i, \boldsymbol{x}(\tau_i), (\boldsymbol{T}_i, \boldsymbol{\theta}_i)))$ or two update functions $\boldsymbol{U}$ and $\boldsymbol{U}'$ can be concatenated so that $(\boldsymbol{T}_{i+1}, \boldsymbol{\theta}_{i+1}) = \boldsymbol{U}'(\tau_i, \boldsymbol{x}(\tau_i), \boldsymbol{U}(\tau_i, \boldsymbol{x}(\tau_i), (\boldsymbol{T}_i, \boldsymbol{\theta}_i)))$ at the time τ_i.

2.3 *Discrete-time problem statement*

The methodology proposed in this paper is a sampled-data scheme in two respects. First, the update function $\boldsymbol{U}$, which can be interpreted as a control law, is executed at discrete times τ_i. Second, in a computer implementation, the calculation of $\hat{\boldsymbol{x}}_t^{\boldsymbol{T},\boldsymbol{\theta}}$, $J(\hat{\boldsymbol{x}}_t^{\boldsymbol{T},\boldsymbol{\theta}}(\,\cdot\,), \boldsymbol{w}(\,\cdot\,, \boldsymbol{T}, \boldsymbol{\theta}), t, \boldsymbol{T})$ from (5), and the derivatives of J inevitably requires a discrete-time formulation. Such calculations may be needed in $\boldsymbol{U}$ to ensure a decrease of J required for closed-loop stability (cf. Appendix A.2). Because these calculations have to be done in real time, i. e., with high computational efficiency, we revisit the problem statement from Section 2.1 for the sampled-data case.

Consider a time grid with grid points t_k, $k = 0, 1, 2, \ldots$. This grid is independent of the generally coarser grid defined by $\boldsymbol{T}$. The discrete-time version of (1) with $\boldsymbol{u}(t)$ from (4) can be written in the form

$$\mathbf{0} = \boldsymbol{F}_k(\boldsymbol{x}_{k+1}, \boldsymbol{x}_k, \boldsymbol{T}, \boldsymbol{\theta}), \tag{8}$$

where $\boldsymbol{x}_k \in \mathbb{R}^n$ is the discrete-time approximation of $\boldsymbol{x}(t_k)$. $\boldsymbol{F}_k : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{N+1} \times \mathbb{R}^{Nn_\vartheta} \to \mathbb{R}^n$ can be chosen from any (explicit or implicit) integration scheme for ordinary differential equations, e. g., $\boldsymbol{F}_k$ can be a Runge-Kutta method (Hairer et al., 1993; Stoer and Bulirsch, 2002). We assume that the discrete-time approximation (8) of (1) is sufficiently accurate in the sense of (Nešić and Teel, 2004; Nešić et al., 1999), i. e., the sampling periods $t_{k+1} - t_k$ are sufficiently small to satisfy the conditions given in (Nešić and Teel, 2004; Nešić et al., 1999). This assumption implies that the closed-loop stability properties obtained for (8) are also valid for the continuous-time plant (1).

Consider that $t = t_{k_0}$ and $T_N = t_{k_1}$, i. e., the considered receding horizon is $[t_{k_0}, t_{k_1}]$. We denote by $(\hat{\boldsymbol{x}}_{k_0,k})$ the discrete-time trajectory of (8) during the receding horizon $[t_{k_0}, t_{k_1}]$ with the initial condition $\hat{\boldsymbol{x}}_{k_0,k_0} = \boldsymbol{x}_{k_0}$. Note that $(\hat{\boldsymbol{x}}_{k_0,k})$ is a series of discrete-time values whereas $\hat{\boldsymbol{x}}_{k_0,k}$ denotes a single element of this series associated with the time t_k. If the input is defined by the pair $\boldsymbol{T}$ and $\boldsymbol{\theta}$, we shall write $(\hat{\boldsymbol{x}}_{k_0,k}^{\boldsymbol{T},\boldsymbol{\theta}})$.

The integral in the objective function (5) is approximated by some numerical quadrature formula (Stoer and Bulirsch, 2002), which will typically use the states $\hat{\boldsymbol{x}}_{k_0,k}$ and the sampled input values $\boldsymbol{w}(t_k, \boldsymbol{T}, \boldsymbol{\theta})$. We can thus approximate (5) in the form

$$\begin{aligned} &J_d((\hat{\boldsymbol{x}}_{k_0,k}), \boldsymbol{T}, \boldsymbol{\theta}, k_0, k_1) \\ &\quad = V(\hat{\boldsymbol{x}}_{k_0,k_1}) + \sum_{k=k_0}^{k_1-1} v_k(\hat{\boldsymbol{x}}_{k_0,k+1}, \hat{\boldsymbol{x}}_{k_0,k}, \boldsymbol{T}, \boldsymbol{\theta}), \end{aligned} \tag{9}$$

with the time-dependent objective function $v_k : \mathbb{R}^n \times \mathbb{R}^n \times \mathbb{R}^{N+1} \times \mathbb{R}^{Nn_\vartheta} \to \mathbb{R}_{\geq 0}$. For instance, if the trapezoidal rule is used, v_k reads as

$$\begin{aligned} v_k(\hat{\boldsymbol{x}}_{k_0,k+1}, \hat{\boldsymbol{x}}_{k_0,k}, \boldsymbol{T}, \boldsymbol{\theta}) = &\frac{t_{k+1} - t_k}{2} \\ &\left(v(\hat{\boldsymbol{x}}_{k_0,k+1}, \boldsymbol{w}(t_{k+1}, \boldsymbol{T}, \boldsymbol{\theta})) + v(\hat{\boldsymbol{x}}_{k_0,k}, \boldsymbol{w}(t_k, \boldsymbol{T}, \boldsymbol{\theta}))\right). \end{aligned}$$

Finally, (6) can be rewritten in the discrete-time form

$$\min_{\boldsymbol{\theta} \in \mathcal{T}(\boldsymbol{T})} \; J_d((\hat{\boldsymbol{x}}_{k_0,k}^{\boldsymbol{T},\boldsymbol{\theta}}), \boldsymbol{T}, \boldsymbol{\theta}, k_0, k_1) \tag{10}$$

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

3. Fundamental update function

We consider fundamental updates $\boldsymbol{U}$ that do not change $\boldsymbol{w}(t, \boldsymbol{T}_i, \boldsymbol{\theta}_i)\ \forall\, t \in [\tau_i, \min\{T_{i|N_i}, T_{i+1|N_{i+1}}\})$. That is, the previous control is preserved in the overlapping time interval.

3.1 *Basic types of fundamental updates*

We suggest three basic types of fundamental updates:

- The update function ***Delete*** shortens the horizon by one elapsed time interval, i. e.,

$$\begin{aligned} N_{i+1} &= N_i - 1 \\ T_{i+1|j} &= T_{i|j+1} \quad \forall\, j = 0, \dots, N_i - 1 \\ \boldsymbol{\vartheta}_{i+1|j} &= \boldsymbol{\vartheta}_{i|j+1} \quad \forall\, j = 1, \dots, N_i - 1. \end{aligned}$$

  Clearly, this update function can only be applied if $\tau_i \geq T_{i|1}$. By full analogy, a similar update function may be devised that shortens the horizon at its end.
- The update function ***Divide*** splits a time interval $(T_{i,k-1}, T_{i,k})$ with $1 \leq k \leq N_i$ into two intervals, i. e.,

$$\begin{aligned} N_{i+1} &= N_i + 1 \\ T_{i+1|j} &= T_{i|j} \quad \forall\, j = 0, \dots, k-1 \\ T_{i+1|j} &= T_{i|j-1} \quad \forall\, j = k+1, \dots, N_i + 1 \\ \boldsymbol{\vartheta}_{i+1|j} &= \boldsymbol{\vartheta}_{i|j} \quad \forall\, j = 1, \dots, k-1 \\ \boldsymbol{\vartheta}_{i+1|j} &= \boldsymbol{\vartheta}_{i|j-1} \quad \forall\, j = k+2, \dots, N_i + 1. \end{aligned}$$

  The new grid point $T_{i+1|k}$ satisfying $T_{i|k-1} < T_{i+1|k} < T_{i|k}$ can be chosen at will. The parameter vectors $\boldsymbol{\vartheta}_{i+1|k}$ and $\boldsymbol{\vartheta}_{i+1,k+1}$ have to be chosen so that

$$\begin{aligned} &\bar{\boldsymbol{w}}(t, T_{i|k-1}, T_{i|k}, \boldsymbol{\vartheta}_{i|k}) \\ &\quad = \begin{cases} \bar{\boldsymbol{w}}(t, T_{i+1|k-1}, T_{i+1|k}, \boldsymbol{\vartheta}_{i+1|k}) \\ \qquad\qquad \text{if } t \in [T_{i+1|k-1}, T_{i+1|k}) \\ \bar{\boldsymbol{w}}(t, T_{i+1|k}, T_{i+1|k+1}, \boldsymbol{\vartheta}_{i+1|k+1}) \\ \qquad\qquad \text{if } t \in [T_{i+1|k}, T_{i+1|k+1}) \end{cases} \end{aligned}$$

  holds exactly. Based on the assumptions stated in Section 2.1, this is always possible.
- The update function ***Append*** enlarges the horizon by appending one time interval in the form

$$\begin{aligned} N_{i+1} &= N_i + 1 \\ T_{i+1|j} &= T_{i|j} \quad \forall\, j = 0, \dots, N_i \\ T_{i+1|N_{i+1}} &= T_{i|N_i} + \kappa_T \\ \boldsymbol{\vartheta}_{i+1|j} &= \boldsymbol{\vartheta}_{i|j} \quad \forall\, j = 1, \dots, N_i \\ \boldsymbol{\vartheta}_{i+1|N_{i+1}} &= \boldsymbol{\kappa}(\hat{\boldsymbol{x}}_{\tau_i}^{\boldsymbol{T}_i, \boldsymbol{\theta}_i}(T_{i|N_i})). \end{aligned}$$

  Here, $\boldsymbol{\kappa} : \mathring{\mathcal{X}}_f \to \mathbb{R}^{n_\vartheta}$ is a state feedback function applied on a uniform time grid with the sampling period $\kappa_T \in \mathbb{R}_{\geq 0}$. This function must decrease J to ensure closed-loop stability (cf.

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

Appendix A.2). Extending the theory and results presented in this paper to cases where κ_T is also a state feedback function, i. e., $\kappa_T : \mathring{\mathcal{X}}_f \to \mathbb{R}_{\geq 0}$ is straightforward.

Fundamental update functions that are more sophisticated than those given here are possible but not needed in this paper. For the update functions *Delete* and *Divide*, the choice of $\boldsymbol{\theta}_{i+1}$ is unique and straightforward and it is easy to show that they decrease J or keep it constant as required for closed-loop stability (cf. Appendix A.2). This is not the case for the update function *Append*, where the choice of κ_T and the synthesis of $\boldsymbol{\kappa}$ are crucial design decisions. They will be addressed in Section 5.

3.2 *Update strategy for the time grid*

The following update strategy for the time grid defined by $\boldsymbol{T}_i$ is a combination of the proposed basic fundamental updates. Consider that fundamental updates are made at $\tau_i = T_{i|1}$ and that

$$\Delta T_{i|j} = T_{i|j+1} - T_{i|j} \qquad \forall\, j = 0, \ldots, N_i - 1.$$

We use a constant value $\Delta T_{i|j} = \Delta T = \kappa_T / N_T$ with some integer $N_T \geq 1$. Moreover, we allow that N_i varies between $N_{\min}$ and $N_{\min} + N_T - 1$. At any update point $\tau_i = T_{i|1}$, we apply the update *Delete*. If additionally $N_i = N_{\min}$, we apply also the updates *Append* and in case of $N_T > 1$ also *Divide*. Figure 1 shows an example of prediction horizons (1, 2, 3, ...) generated using this strategy with $N_T = 2$ and $N_{\min} = 4$. For clarity of presentation, the time grid τ_i is given for fundamental updates only (incremental updates are not considered).

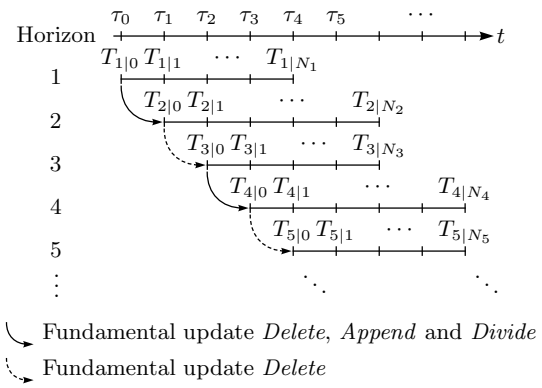

Figure 1. Prediction horizons obtained by the update strategy for the time grid if $N_T = 2$ and $N_{\min} = 4$ (time grid τ_i only shown for fundamental updates).

In order to achieve low-dimensional optimization problems, it may be desirable in RHC concepts to use non-uniform time grids where $\Delta T_{i|j}$ is small at the beginning of the horizon and larger towards its end. An alternative update strategy for the time grid that realizes this idea is presented in Appendix B.

4. Incremental update function

An update function $\boldsymbol{U}$ applied at some time $\tau_i = t_{k_0}$ is referred to as *incremental* if $\boldsymbol{T}_{i+1} = \boldsymbol{T}_i$, i. e., if the time grid defining the input remains unchanged. The value $\boldsymbol{\theta}_i$ may be improved to $\boldsymbol{\theta}_{i+1}$. The absence of a structural change of the input parameterization allows the interpretation of incremental updates as steps of a primal iterative optimization algorithm solving (6) or its

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

discrete-time companion (10). Examples for such primal iterative solution processes are the steepest descent method, the conjugate gradient method, the quasi-Newton method, the Gauss-Newton method, and the Newton method (Bertsekas, 1999; Griva et al., 2009; Luenberger and Ye, 2008; Nocedal and Wright, 2006). It is clear that the assumptions and conditions that ensure closed-loop stability (cf. Appendix A) do not require the optimal solution of (6) or (10). Just a simple search along a local descent direction of J or J_d suffices to strictly satisfy Lyapunov-like inequalities unless $\boldsymbol{x}(\tau_i) \in \mathcal{X}_d$ (cf. Appendix A.2). In this respect, the proposed NMPC scheme is suboptimal, which facilitates extremely fast computation of $\boldsymbol{U}$ and its use for real-time control of fast dynamical systems.

In this section, an incremental update function is designed based on the discrete-time formulation (10), i. e., a *direct method* is used. The incremental update function $\boldsymbol{U}$ can be a *single* iteration of a dynamic programming algorithm. This iteration typically involves the computation of a *search direction* $\boldsymbol{s}$, which must be a descent direction of J_d, a decision about the *step size* σ along this direction, and a *projection* of the result into the set $\mathcal{T}(\boldsymbol{T}_{i+1})$. If the available computation time suffices, $\boldsymbol{U}$ can involve *several* of these iterations so that the decrease from $J_d((\hat{\boldsymbol{x}}_{k_0,k}^{\boldsymbol{T}_i,\boldsymbol{\theta}_i}), \boldsymbol{T}_i, \boldsymbol{\theta}_i, k_0, k_1)$ to $J_d((\hat{\boldsymbol{x}}_{k_0,k}^{\boldsymbol{T}_{i+1},\boldsymbol{\theta}_{i+1}}), \boldsymbol{T}_{i+1}, \boldsymbol{\theta}_{i+1}, k_0, k_1)$ is larger. In fact, the balance between computational effort and improvement of J_d can be analyzed and monitored (Alamir, 2014) and the optimal number of iterations may be determined as proposed by Alamir (2013). On a theoretical level, such an analysis is beyond the scope of this paper. On a practical level, the influence of the number of iterations executed at each sampling point is studied in an example problem in Section 6. The number of iterations executed at each sampling point is a design parameter of the proposed control approach.

We use a gradient-based dynamic programming algorithm. The discrete-time problem formulation from Section 2.3 facilitates an efficient analytical computation of the gradient $\boldsymbol{g} = \mathrm{d}J_d((\hat{\boldsymbol{x}}_{k_0,k}^{\boldsymbol{T}_i,\boldsymbol{\theta}_i}), \boldsymbol{T}_i, \boldsymbol{\theta}_i, k_0, k_1)/\mathrm{d}\boldsymbol{\theta}_i$. Consider the Lagrangian

$$L = J_d((\hat{\boldsymbol{x}}_{k_0,k}), \boldsymbol{T}_i, \boldsymbol{\theta}_i, k_0, k_1) + \sum_{k=k_0}^{k_1-1} \boldsymbol{\lambda}_k^T \boldsymbol{F}_k(\hat{\boldsymbol{x}}_{k_0,k+1}, \hat{\boldsymbol{x}}_{k_0,k}, \boldsymbol{T}_i, \boldsymbol{\theta}_i)$$

with Lagrangian multipliers (adjoint variables) $\boldsymbol{\lambda}_k \in \mathbb{R}^n$. The gradient follows in the form

$$\boldsymbol{g} = \frac{\partial L}{\partial \boldsymbol{\theta}_i} \tag{11a}$$

given that the equalities

$$\frac{\partial L}{\partial \hat{\boldsymbol{x}}_{k_0,k}} = \boldsymbol{0} \quad \forall\, k = k_0{+}1, \ldots, k_1 \tag{11b}$$

$$\frac{\partial L}{\partial \boldsymbol{\lambda}_k} = \boldsymbol{0} \quad \forall\, k = k_0, \ldots, k_1 - 1 \tag{11c}$$

are satisfied. Eq. (11c) is equivalent to (8). The computation proceeds as follows: a) Compute $\hat{\boldsymbol{x}}_{k_0,k}$ by solving (11c) in forward direction, i. e., for $k = k_0, \ldots, k_1 - 1$. b) Compute $\boldsymbol{\lambda}_k$ by solving the linear Eqs. (11b) in backward direction, i. e., for $k = k_1, \ldots, k_0{+}1$. c) Insert the results $(\hat{\boldsymbol{x}}_{k_0,k})$ and $(\boldsymbol{\lambda}_k)$ into (11a). The matrices occurring in these computations are sparse, which should be utilized in computer implementations. Next, the search direction is computed in the form

$$\boldsymbol{s} = -\boldsymbol{B}^{-1}\boldsymbol{g},$$

where $\boldsymbol{B}$ is the Hessian or an approximation of it.

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

We can now find $\boldsymbol{\theta}_{i+1}$ by searching for a minimum along the direction $\boldsymbol{s}$. To ensure that $\boldsymbol{\theta}_{i+1} \in \mathcal{T}(\boldsymbol{T}_{i+1})$, we use the projection function

$$\boldsymbol{p}(\boldsymbol{T}, \boldsymbol{\theta}) = \arg \min_{\tilde{\boldsymbol{\theta}} \in \mathcal{T}(\boldsymbol{T})} ||\tilde{\boldsymbol{\theta}} - \boldsymbol{\theta}|| \tag{12}$$

with some norm $||\cdot||$. This function returns a point from the set $\mathcal{T}(\boldsymbol{T})$ which is closest to $\boldsymbol{\theta}$.

Remark 1: *The projection function $\boldsymbol{p}$ from* (12) *constitutes an optimization problem. Existence of its solution is guaranteed if $\mathcal{T}(\boldsymbol{T})$ is non-empty. If $\mathcal{T}(\boldsymbol{T})$ is a strictly convex set, the solution of this problem is ensured to be unique. If $\boldsymbol{\theta} \in \mathcal{T}(\boldsymbol{T})$, $\boldsymbol{p}$ simply returns $\boldsymbol{\theta}$. If $\mathcal{T}(\boldsymbol{T})$ is defined by box constraints, a solution of* (12) *is found by elementwise saturation (cf. Bazaraa et al., 2006). Generally, there is no need for an exact (optimal) solution of* (12)*. However, $\boldsymbol{p}(\boldsymbol{T}, \boldsymbol{\theta}) \in \mathcal{T}(\boldsymbol{T})$ must be ensured.*

The step size σ along the direction $\boldsymbol{s}$ is chosen by solving the 1-dimensional optimization problem

$$\sigma = \arg \min_{\tilde{\sigma} \in \mathbb{R}_{\geq 0}} J_d((\hat{\boldsymbol{x}}_{k_0,k}^{\boldsymbol{T}_{i+1},\boldsymbol{\theta}}), \boldsymbol{T}_{i+1}, \boldsymbol{\theta}, k_0, k_1) \tag{13a}$$

$$\text{s. t. } \boldsymbol{\theta} = \boldsymbol{p}(\boldsymbol{T}_{i+1}, \boldsymbol{\theta}_i + \tilde{\sigma}\boldsymbol{s}) \tag{13b}$$

which is often called *line search*. Finally, $\boldsymbol{\theta}_{i+1}$ is computed in the form $\boldsymbol{\theta}_{i+1} = \boldsymbol{p}(\boldsymbol{T}_{i+1}, \boldsymbol{\theta}_i + \sigma\boldsymbol{s})$. Generally, there is no need for an exact (optimal) solution of (13).

Remark 2: *In (DeHaan and Guay, 2007), incremental updates were realized by means of an ordinary differential equation for $\boldsymbol{\theta}$. These updates require extra integration effort and a special Lipschitz projection algorithm arising from nonlinear adaptive control techniques. These updates only allow continuous changes of $\boldsymbol{\theta}(t)$. If fast changes of $\boldsymbol{\theta}(t)$ should be achieved, this may require steep gradients, which can render the closed-loop dynamical system stiff. The method proposed here does not have these shortcomings because it is a true sampled-data formulation and allows arbitrarily large discontinuous changes.*

5. Feedback law for update function *Append*

The choice of κ_T and $\boldsymbol{\kappa}$ used in the update function *Append* (cf. Section 3.1) is by no means unique. This gives freedom in its design. The value of κ_T and the function $\boldsymbol{\kappa}$ depend on each other in the sense that they generally cannot be arbitrarily combined. Based on the assumption that the Jacobian linearization of (1) is stabilizable in $\mathcal{X}_d$, we now present a constructive design method for $\boldsymbol{\kappa}$, which yields appropriate objective functions V and v as a byproduct. However, other nonlinear control design methods (see e. g., Khalil, 2002; Kristić et al., 1995; Sastry, 1999) that satisfy the requirements for closed-loop stability (cf. Appendix A) could equally be used. If the Jacobian linearization of (1) is not stabilizable in $\mathcal{X}_d$, methods like those presented in (Chen and Allgöwer, 1998a; Fontes, 2001) may be applied. This problem occurs for non-holonomic systems, for which tailored NMPC schemes were developed in (Fontes et al., 2007; Worthmann et al., 2016).

We consider a piecewise constant input parameterization of the form

$$\boldsymbol{u}(t) = \bar{\boldsymbol{w}}(t, T_{k-1}, T_k, \boldsymbol{\vartheta}_k) = \boldsymbol{\vartheta}_k \quad \forall\, t \in [T_{k-1}, T_k), k = 1, 2, \ldots, N.$$

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

This means that $\boldsymbol{\kappa}$ can be designed using a zero-order hold. We use the linear feedback law

$$\boldsymbol{\vartheta}_{k+1} = \boldsymbol{\kappa}(\boldsymbol{x}_k) = -\boldsymbol{K}\boldsymbol{x}_k \tag{14}$$

with the constant gain $\boldsymbol{K}$. Here, $\boldsymbol{x}_k = \boldsymbol{x}(T_k)$ and k is the index of the time grid T_k. Inspired by the approach reported in (Chen and Allgöwer, 1998b) for continuous-time NMPC, we suggest a different systematic sampled-data control design based on the locally linearized system. Consider the Jacobian linearization

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}_c\boldsymbol{x}(t) + \boldsymbol{B}_c\boldsymbol{u}(t) \tag{15}$$

of (1) with $\boldsymbol{A}_c = \partial\boldsymbol{f}/\partial\boldsymbol{x}|_{\boldsymbol{x}=\boldsymbol{0},\boldsymbol{u}=\boldsymbol{0}}$, and $\boldsymbol{B}_c = \partial\boldsymbol{f}/\partial\boldsymbol{u}|_{\boldsymbol{x}=\boldsymbol{0},\boldsymbol{u}=\boldsymbol{0}}$. Its discrete-time counterpart found by zero-order hold reads as

$$\boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{\vartheta}_{k+1}$$

with $\boldsymbol{A} = \exp(\kappa_T\boldsymbol{A}_c)$ and $\boldsymbol{B} = \int_0^{\kappa_T}\exp(t\boldsymbol{A}_c)\mathrm{d}t\boldsymbol{B}_c$ and is assumed to be stabilizable in $\mathcal{X}_d$. Using the steady-state LQR design method with the positive definite objective function

$$\tilde{V}(\boldsymbol{x}_k) = \sum_{i=k}^{\infty}\begin{bmatrix}\boldsymbol{x}_k^T & \boldsymbol{\vartheta}_{k+1}^T\end{bmatrix}\underbrace{\begin{bmatrix}\boldsymbol{Q} & \boldsymbol{N}\\ \boldsymbol{N}^T & \boldsymbol{R}\end{bmatrix}}_{=\boldsymbol{\Gamma}}\begin{bmatrix}\boldsymbol{x}_k\\ \boldsymbol{\vartheta}_{k+1}\end{bmatrix} = \boldsymbol{x}_k^T\boldsymbol{P}\boldsymbol{x}_k$$

and the solution $\boldsymbol{P}$ of the corresponding algebraic Riccati equation, we obtain

$$\boldsymbol{K} = (\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{B} + \boldsymbol{R})^{-1}(\boldsymbol{B}^T\boldsymbol{P}\boldsymbol{A} + \boldsymbol{N}^T). \tag{16}$$

Based on the correspondence

$$\begin{bmatrix}-\boldsymbol{\Phi}_c^T & \boldsymbol{\Gamma}_c\\ \boldsymbol{0} & \boldsymbol{\Phi}_c\end{bmatrix} = \frac{1}{\kappa_T}\ln\left(\begin{bmatrix}\boldsymbol{\Phi}^{-T} & \boldsymbol{\Phi}^{-T}\boldsymbol{\Gamma}\\ \boldsymbol{0} & \boldsymbol{\Phi}\end{bmatrix}\right) \tag{17}$$

with

$$\boldsymbol{\Phi}_c = \begin{bmatrix}\boldsymbol{A}_c & \boldsymbol{B}_c\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix},\quad \boldsymbol{\Gamma}_c = \begin{bmatrix}\boldsymbol{Q}_c & \boldsymbol{N}_c\\ \boldsymbol{N}_c^T & \boldsymbol{R}_c\end{bmatrix},\quad \boldsymbol{\Phi} = \begin{bmatrix}\boldsymbol{A} & \boldsymbol{B}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}$$

(Franklin et al., 1997; Van Loan, 1978), the equivalent objective function of the sampled-data LQR design problem reads as

$$\tilde{V}(\boldsymbol{x}(T_k)) = \int_{T_k}^{\infty}\begin{bmatrix}\boldsymbol{x}^T(t) & \boldsymbol{u}^T(t)\end{bmatrix}\boldsymbol{\Gamma}_c\begin{bmatrix}\boldsymbol{x}(t)\\ \boldsymbol{u}(t)\end{bmatrix}\mathrm{d}t.$$

In (17), $\ln(\cdot)$ returns the logarithm of a matrix. This operation is only defined for invertible matrices.

The functions

$$\tilde{V}(\boldsymbol{x}) = \boldsymbol{x}^T\boldsymbol{P}\boldsymbol{x},\qquad \tilde{v}(\boldsymbol{x},\boldsymbol{u}) = \begin{bmatrix}\boldsymbol{x}^T & \boldsymbol{u}^T\end{bmatrix}\boldsymbol{\Gamma}_c\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{u}\end{bmatrix} \tag{18}$$

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

are reasonable starting points for choosing the objective functions V and v, respectively, in (5). However, $\tilde{V}$ and $\tilde{v}$ do not satisfy all requirements to ensure closed-loop stability (cf. Appendix A), e. g., because they are neither barrier functions nor do they necessarily vanish for all $(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{X}_d \times \mathcal{U}$. Therefore $\tilde{V}$ and $\tilde{v}$ are supplemented in the form

$$V(\boldsymbol{x}) = C(\boldsymbol{x})(\tilde{V}(\boldsymbol{x}) + \mu B(\boldsymbol{x})) \tag{19a}$$

$$v(\boldsymbol{x}, \boldsymbol{u}) = C(\boldsymbol{x})(\zeta \tilde{v}(\boldsymbol{x}, \boldsymbol{u}) + \eta b(\boldsymbol{x})) \tag{19b}$$

with constant design parameters $\mu > 0$, $\zeta \in (0, 1]$, $\eta > 0$, barrier functions B and b, and an indicator function C. We assume that B, b, and C are sufficiently often differentiable for the methods used in this paper. The barrier functions have to satisfy $\lim_{\boldsymbol{x} \to \partial \mathcal{X}_f} B(\boldsymbol{x}) = \infty$ and $\lim_{\boldsymbol{x} \to \partial \mathcal{X}} b(\boldsymbol{x}) = \infty$ (DeHaan and Guay, 2007; Wills and Heath, 2004). The indicator function has to satisfy $C(\boldsymbol{x}) = 0$ $\forall\, \boldsymbol{x} \in \mathcal{X}_d$ and $\infty > C(\boldsymbol{x}) > 0$ $\forall\, \boldsymbol{x} \in \mathcal{X} \backslash \mathcal{X}_d$ (DeHaan and Guay, 2007). Because of these properties of $C(\boldsymbol{x})$, we do not require the barrier functions $B(\boldsymbol{x})$ and $b(\boldsymbol{x})$ to be centered.

The choice of $B(\boldsymbol{x})$, $b(\boldsymbol{x})$, $C(\boldsymbol{x})$, μ, ζ, and η has to ensure that V and v from (19) satisfy conditions that ensure closed-loop stability (cf. Appendix A). If $\mathcal{X}_d$ and $\mathcal{X}_f$ are chosen to be level sets of $\tilde{V}$, i. e., if there exist constants γ and β so that $\mathcal{X}_d = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \tilde{V}(\boldsymbol{x}) \leq \gamma\}$ and $\mathcal{X}_f = \{\boldsymbol{x} \in \mathbb{R}^n \,|\, \tilde{V}(\boldsymbol{x}) \leq \beta\}$, then

$$C(\boldsymbol{x}) = r\left(\frac{\tilde{V}(\boldsymbol{x}) - \gamma}{\beta - \gamma}\right) \tag{20}$$

with

$$r(\xi) = \begin{cases} 0 & \text{if } \xi \leq 0 \\ 1 & \text{if } \xi \geq 1 \\ 3\xi^2 - 2\xi^3 & \text{otherwise} \end{cases}$$

and

$$B(\boldsymbol{x}) = \frac{1}{\max\{0, \beta - \tilde{V}(\boldsymbol{x})\}} - \frac{1}{\beta} \tag{21}$$

are reasonable formulations. Depending on $\mathcal{X}$, a function analogous to $B(\boldsymbol{x})$ can be used for $b(\boldsymbol{x})$.

Using V and v from (19) and $\boldsymbol{\kappa}$ according to (14) and (16), it is plausible that the update function *Append* can satisfy requirements necessary for closed-loop stability (cf. Appendix A). However, for two reasons, it has to be tested on a case-by-case basis whether V and v from (19) and $\boldsymbol{\kappa}$ from (14) and (16) indeed satisfy these requirements: First, the original objective functions (18) obtained from the LQR design have been supplemented. Second, the controller is designed for the linearized system (15) but used for the original nonlinear system (1). This test can be done, for instance, by exhaustive simulation. The factors μ, η, and ζ are user-defined design parameters which may need to be tuned for the test to be positive. The smaller the mismatch between (1) and (15), the higher ζ can be. Since only the mismatch in the region $\mathcal{X}_f$ (domain of $\boldsymbol{\kappa}$) is relevant, ζ also depends on the size of $\mathcal{X}_f$.

This section showed that the design of κ_T, $\boldsymbol{\kappa}$, and the corresponding Lyapunov function essentially influences the NMPC optimization problem, especially the objective functions V and v. Using the LQR design approach, usually either $\boldsymbol{\Gamma}$ or $\boldsymbol{\Gamma}_c$ is chosen and the other one is computed based on (17). Then $\boldsymbol{P}$, which determines $\tilde{V}$, follows from the corresponding discrete-time algebraic Riccati equation. In some cases, it can be useful to choose $\boldsymbol{P}$ as a diagonal matrix or even the identity matrix. This is possible if other entries of $\boldsymbol{\Gamma}$, e. g., $\boldsymbol{Q}$, are left undetermined and then computed based on the Riccati equation.

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

6. Example problem

To demonstrate the feasibility of the proposed sampled-data NMPC design method and to compare it with existing NMPC schemes, an example problem from (Chen and Allgöwer, 1998b; DeHaan and Guay, 2007) is solved. The system

$$\dot{\boldsymbol{x}}(t) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \boldsymbol{x}(t) + \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 0 & -4 \end{bmatrix} \boldsymbol{x}(t) u(t) + \frac{1}{2} \begin{bmatrix} 1 \\ 1 \end{bmatrix} u(t)$$

with an initial state $\boldsymbol{x}_0 = [-0.683, -0.864]^T$ should be stabilized at the origin ($\mathcal{X}_d = \{\boldsymbol{0}\}$). The scalar input $u(t)$ is constrained to the set $\mathcal{U} = [-2, 2]$. The state $\boldsymbol{x}(t)$ is unconstrained ($\mathcal{X} = \mathbb{R}^2$).

For the proposed sampled-data NMPC, we first design a feedback control law $\kappa(\boldsymbol{x})$ according to Section 5. We use $\boldsymbol{P} = \boldsymbol{I}$, $\boldsymbol{N} = \boldsymbol{0}$, $R = 0.01$, and $\kappa_T = 0.5\,\mathrm{s}$. This yields

$$\boldsymbol{Q}_c = \begin{bmatrix} 1.510 & -0.490 \\ -0.490 & 1.510 \end{bmatrix}, \quad \boldsymbol{N}_c = \begin{bmatrix} -0.165 \\ -0.165 \end{bmatrix},$$

and $R_c = 0.055$. Moreover, we use $\mu = 0.003$, $\zeta = 1$, $\mathcal{X}_d = \{0\}$, and $\mathcal{X}_f = \{\boldsymbol{x} \in \mathbb{R}^n \mid \tilde{V}(\boldsymbol{x}) \leq 0.06\}$. C and B are chosen according to (20) and (21), respectively. As the state $\boldsymbol{x}(t)$ is unconstrained, $b(\boldsymbol{x})$ is not needed. This concludes the definition of the objective functions V and v. Without modifying these functions, a second sampled-data LQR-feedback control law is designed for $\kappa_T = 0.1\,\mathrm{s}$. For both designs of $\kappa(\boldsymbol{x})$, it can be shown by straightforward simulations that the update function *Append* meets the requirements necessary for closed-loop stability (cf. Appendix A). This would not be true if $C(\boldsymbol{x}) = 1$ and $B(\boldsymbol{x}) = 0$, and this shows that the controller in combination with the objective function $\tilde{V}(\hat{\boldsymbol{x}}_t(T_N)) + \int_t^{T_N} \tilde{v}(\hat{\boldsymbol{x}}(\tau), \boldsymbol{u}(\tau))\mathrm{d}\tau$ would not satisfy the overly restrictive requirements stipulated in Assumption A1.5 of (DeHaan and Guay, 2007).

The time grids of the sampled-data NMPC are updated as described in Section 3.2. Incremental updates are implemented on time grids with uniform step sizes $\Delta\tau_i = \Delta\tau$. Fundamental updates are implemented on time grids with uniform step sizes $\Delta T = \kappa_T / N_T$. These are also the time grids where the piecewise constant input $u(t)$ is defined.

For the proposed sampled-data NMPC scheme, we use the objective function J_d from (9) with V and v from (19) and a prediction horizon with a maximum length of 1.5 s. The initial solution during the first horizon $[0, T_{0|N_0})$ with $T_{0|N_0} = 1.5\,\mathrm{s}$ is simply $u(t) = 2$ and ensures that $\hat{\boldsymbol{x}}_0(T_{0|N_0}) \in \mathcal{X}_f$. If such an initial guess were unknown, it could be computed off-line using an optimal control problem with the terminal constraint $\hat{\boldsymbol{x}}_0(T_{0|N_0}) \in \mathcal{X}_f$.

For comparison, the considered example problem is solved with the following controllers:

- Sampled-data 1: sampled-data real-time NMPC scheme as proposed in this paper with single iteration step of the dynamic programming algorithm, implemented in Matlab®
- Sampled-data 2: sampled-data real-time NMPC scheme as proposed in this paper with 2 iteration steps of the dynamic programming algorithm, implemented in Matlab®
- IPOPT: standard NMPC scheme using IPOPT (Wächter and Biegler, 2006) for solving the optimal control problem, implemented in Python/CasADi (Andersson, 2013)
- Suboptimal SQP: real-time NMPC scheme using 2 SQP steps with warm start at each sampling point for solving the optimal control problem, implemented in Python/CasADi (Andersson, 2013)
- RT-3 from (DeHaan and Guay, 2007): continuous-time real-time NMPC scheme as proposed in (DeHaan and Guay, 2007), implemented in Matlab®

The controllers IPOPT and Suboptimal SQP are implemented in Python/CasADi (Andersson, 2013). These controllers use the objective function $\tilde{V}(\hat{\boldsymbol{x}}_t(T_N)) + \int_t^{T_N} \tilde{v}(\hat{\boldsymbol{x}}_t(\tau), \boldsymbol{u}(\tau))\mathrm{d}\tau$ and a pre-

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

diction horizon spanning 1.5 s. In Python/CasADi (Andersson, 2013), equality and inequality constraints can be directly implemented, i. e., without barrier or penalty functions. For the controllers IPOPT and Suboptimal SQP, both the prediction model and the objective function are integrated using the trapezoidal rule and the uniform step size $\Delta\tau$.

The controller RT-3 from (DeHaan and Guay, 2007) does not use the step size $\Delta\tau$. Moreover, it uses the objective function J from (5) with V and v from (19). All other parameters and formulations are equal to those of the proposed sampled-data NMPC scheme.

The above controllers are designed with the step sizes $\Delta\tau$ and ΔT specified in Table 1. The corresponding simulation results are shown in Figs. 2–5. For the simulations in Python/CasADi (Andersson, 2013), the plant model was integrated using an explicit fourth-order Runge-Kutta method with the uniform step size $\Delta\tau/100$. For the simulations in Matlab®, the plant model was integrated using the Dormand-Prince method, which features adaptive step size control (Matlab® command `ode45`).

Table 1. Parameters and performance results of various NMPC designs.

Controller	$\Delta\tau$	ΔT	κ_T	J_5	CPU time
Sampled-data 1	0.1 s	0.5 s	0.5 s	1.840	0.09 %
	0.01 s	0.5 s	0.5 s	1.841	0.54 %
	0.01 s	0.1 s	0.1 s	1.833	0.58 %
	0.01 s	0.02 s	0.1 s	1.833	0.62 %
Sampled-data 2	0.1 s	0.5 s	0.5 s	1.839	0.10 %
	0.01 s	0.5 s	0.5 s	1.841	0.77 %
	0.01 s	0.1 s	0.1 s	1.833	0.80 %
	0.01 s	0.02 s	0.1 s	1.833	0.83 %
IPOPT	0.1 s	0.5 s	-	1.842	0.22 %
	0.01 s	0.5 s	-	1.842	3.6 %
	0.01 s	0.1 s	-	1.833	23 %
	0.01 s	0.02 s	-	1.833	234 %
Suboptimal SQP	0.1 s	0.5 s	-	1.869	0.07 %
	0.01 s	0.5 s	-	1.866	0.38 %
	0.01 s	0.1 s	-	1.834	1.4 %
	0.01 s	0.02 s	-	1.833	5.5 %
RT-3 from (DeHaan and Guay, 2007)	-	0.5 s	0.5 s	1.845	100 %

For performance assessment, the accumulated objective value $J_5 = \tilde{V}(\boldsymbol{x}(5)) + \int_0^5 \tilde{v}(\boldsymbol{x}(t), u(t))\mathrm{d}t$ and the CPU time normalized with respect to the value of the controller RT-3 from (DeHaan and Guay, 2007) are also given in Table 1. The values J_5 exhibit only moderate differences because most of the cost is incurred during the period where $u(t)$ attains its maximum value, i. e., where a constraint is active. The results show that the proposed sampled-data NMPC scheme is effective at stabilizing the system. The proposed control scheme requires significantly less CPU time than the continuous-time controller RT-3 defined in (DeHaan and Guay, 2007) and generally also less than the standard NMPC approaches based on IPOPT or SQP optimization routines. The CPU time required by these existing NMPC approaches grows significantly if the sampling period ΔT is reduced. The CPU time required by the proposed sampled-data NMPC scheme is less sensitive to changes of ΔT. In terms of the objective value J_5, the proposed sampled-data NMPC scheme also outperforms all other implemented controllers.

A comparison between the controllers Sampled-data 1 and Sampled-data 2 reveals that there is neither a significant difference in terms of control performance nor in terms of the obtained trajectories nor in terms of the computational effort. Hence, using just a single or two iteration steps of the dynamic programming algorithm described in Section 4 at every sampling point does not entail significant differences. Further simulations (results not given here) showed that the control performance and the trajectories also remain effectively the same if five iteration steps are

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

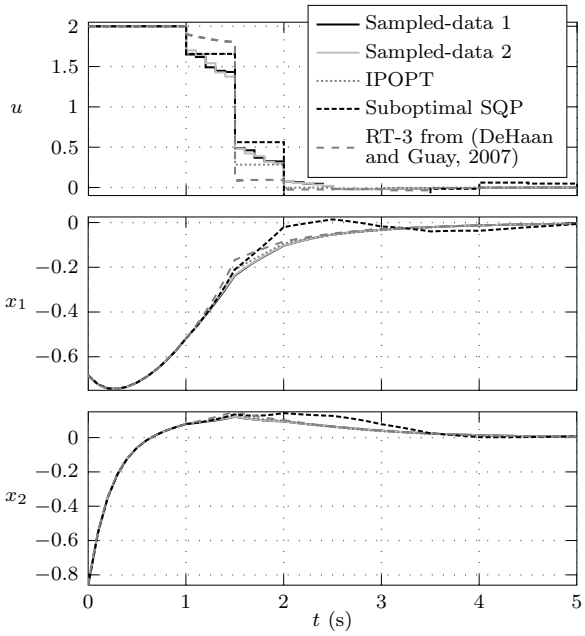

Figure 2. Input and state trajectories of closed-loop NMPC systems for $\Delta\tau = 0.1\,\mathrm{s}$ and $\Delta T = 0.5\,\mathrm{s}$.

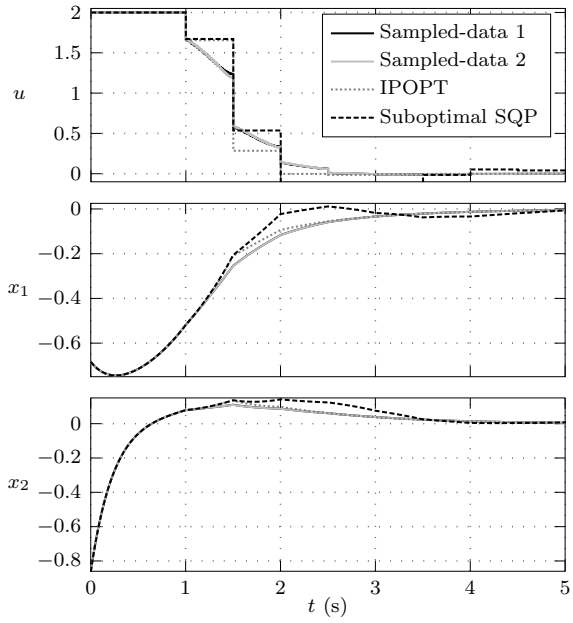

Figure 3. Input and state trajectories of closed-loop NMPC systems for $\Delta\tau = 0.01\,\mathrm{s}$ and $\Delta T = 0.5\,\mathrm{s}$.

executed whereas the computational effort does increase with the number of iteration steps.

Compared to standard NMPC formulations, the proposed control scheme clearly benefits from the incremental updates, which facilitate a modification of the control input on a time grid with the typically small step sizes $\Delta\tau_i$. This capability is independent from the time-discretization of the control input with the typically larger step size $\Delta T_{i|j}$.

We emphasize that these conclusions are based on the results from the considered example problem and an implementation of the proposed controller in Matlab®. Other problem scenarios

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

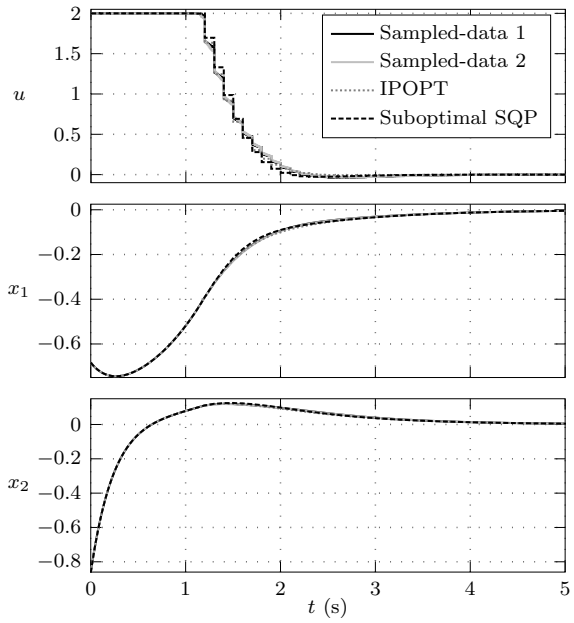

Figure 4. Input and state trajectories of closed-loop NMPC systems for $\Delta\tau = 0.01\,\mathrm{s}$ and $\Delta T = 0.1\,\mathrm{s}$.

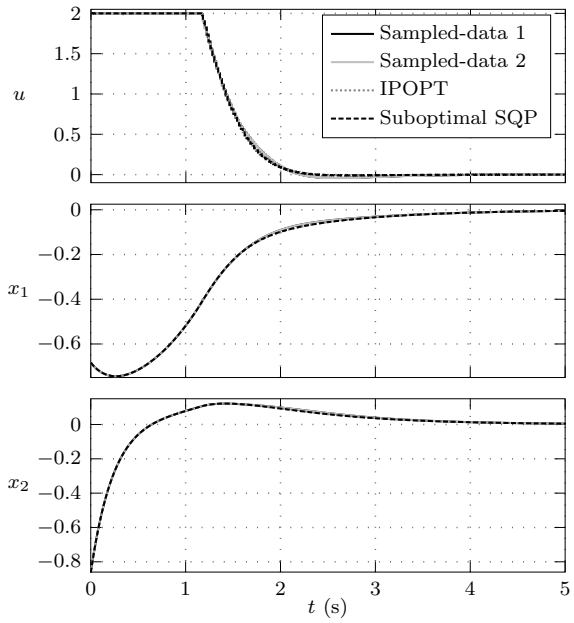

Figure 5. Input and state trajectories of closed-loop NMPC systems for $\Delta\tau = 0.01\,\mathrm{s}$ and $\Delta T = 0.02\,\mathrm{s}$.

or software implementations could lead to different conclusions.

7. Conclusions

In this paper, we developed a sampled-data scheme for real-time NMPC with finite horizon and a terminal constraint set. Rather than using an ad-hoc emulation design, which works best for sampling times being as small as possible, we systematically consider the discrete-time character of

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

the design system in two respects: First, the input parameterization and the terminal set feedback controller (sampled-data LQR design) are defined on discrete time grids. Second, the prediction model and the NMPC calculations use a discrete time grid.

Compared to the earliest available NMPC schemes (e. g., Chen and Allgöwer, 1998b) which were usually formulated as continuous-time control systems, the proposed method follows a sampled-data strategy. This is advantageous in terms of the computational load. Compared to (Diehl et al., 2005), the proposed method has the advantage that it does not require assumptions concerning the disturbance caused by the shift of the receding horizon. Some ideas of the developed real-time NMPC method were adopted from DeHaan and Guay (2007). Compared to their approach, the developed method has the advantage that the gradient $\partial J/\partial \boldsymbol{u}$, which is a function of time, is not required and that the line search facilitates arbitrarily large immediate changes of the control input. In the suboptimal NMPC solutions proposed in (Graichen, 2012b; Graichen and Kugi, 2010; Grüne and Pannek, 2010), the minimum number of required iterations of the underlying optimization problem is a key parameter. This number is a priori unknown for the method described in (Grüne and Pannek, 2010). However, this method does not require terminal costs. In contrast, a single iteration suffices in the method proposed here. In fact, this guarantees a limited execution time of the proposed algorithm, which renders the proposed scheme suitable for real-time control.

The following additional conclusions can be drawn based on the theory and the example problem presented in this work:

- Compared to existing NMPC approaches, the closed-loop performance of the controller proposed in this paper is only moderately better. However, it generally requires significantly less CPU time, especially if the control input is discretized with small time steps. The comparison with existing NMPC approaches shows that the systematic discrete-time design proposed in this paper significantly reduces the computational effort.
- The proposed NMPC method rests on a tailored design of the objective functions, which have to satisfy certain monotonicity relations. Lyapunov-type inequalities are sufficient to prove convergence of the proposed NMPC approach.
- The ongoing reduction of the predicted objective value is due to a shrinking horizon length and a change of the input following a descent direction. These two effects are decoupled and we essentially focused on the second effect by a gradient-based descent method. This is because the contribution of the shrinking horizon is not valuable in terms of control performance.
- Thanks to its high computational efficiency, the proposed scheme can be used for systems with fast dynamics.
- Controlling the system just with the state feedback functions κ_T and $\boldsymbol{\kappa}$ ensures local, practical, asymptotic stability of the set $\mathcal{X}_d$ with a region of attraction containing $\mathring{\mathcal{X}}_f$. The proposed NMPC approach enlarges this region of attraction so that it contains $\mathring{\mathcal{X}}$. In many cases $\mathring{\mathcal{X}}$ is significantly larger than $\mathring{\mathcal{X}}_f$.

References

Adetola, V. and Guay, M. (2010). Integration of real-time optimization and model predictive control. *Journal of Process Control*, 20(2):125–133.

Alamir, M. (2006). *Stabilization of Nonlinear Systems Using Receding-horizon Control Schemes: A Parametrized Approach for Fast Systems*. Springer, London.

Alamir, M. (2013). Monitoring control updating period in fast gradient based NMPC. In *2013 European Control Conference (ECC)*, pages 3621–3626.

Alamir, M. (2014). Fast NMPC: A reality-steered paradigm: Key properties of fast NMPC algorithms. In *2014 European Control Conference (ECC)*, pages 2472–2477.

Andersson, J. (2013). *A General-Purpose Software Framework for Dynamic Optimization*. PhD thesis, Arenberg Doctoral School, KU Leuven, Heverlee, Belgium.

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299

The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

Bazaraa, M.S., Sherali, H.D., and Shetty, C.M. (2006). *Nonlinear Programming: Theory and Algorithms*. John Wiley & Sons, New Jersey, 3rd edition.

Bertsekas, D.P. (1999). *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts, 2nd edition.

Biegler, L.T., Yang, X., and Fischer, G.A.G. (2015). Advances in sensitivity-based nonlinear model predictive control and dynamic real-time optimization. *Journal of Process Control*, 30:104–116.

Cagienard, R., Grieder, P., Kerrigan, E.C., and Morari, M. (2007). Move blocking strategies in receding horizon control. *Journal of Process Control*, 17(6):563–570.

Chen, H. and Allgöwer, F. (1998a). *Nonlinear Model Based Process Control*, chapter Nonlinear Model Predictive Control Schemes with Guaranteed Stability, pages 465–494. Springer, Dordrecht.

Chen, H. and Allgöwer, F. (1998b). A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. *Automatica*, 34(10):1205–1217.

DeHaan, D. and Guay, M. (2007). A real-time framework for model-predictive control of continuous-time nonlinear systems. *IEEE Transactions on Automatic Control*, 52(11):2047–2057.

Diehl, M., Amrit, R., and Rawlings, J.B. (2011). A Lyapunov function for economic optimizing model predictive control. *IEEE Transactions on Automatic Control*, 56(3):703–707.

Diehl, M., Findeisen, R., Allgöwer, F., Bock, H.G., and Schlöder, J.P. (2005). Nominal stability of real-time iteration scheme for nonlinear model predictive control. *IEE Proceedings Control Theory and Applications*, 152(3):296–308.

Feller, C. and Ebenbauer, C. (2013). A barrier function based continuous-time algorithm for linear model predictive control. In *2013 European Control Conference (ECC)*, pages 19–26.

Feller, C. and Ebenbauer, C. (2014a). Barrier function based linear model predictive control with polytopic terminal sets. In *53rd IEEE Conference on Decision and Control*, pages 6683–6688.

Feller, C. and Ebenbauer, C. (2014b). Continuous-time linear MPC algorithms based on relaxed logarithmic barrier functions. *IFAC Proceedings Volumes*, 47(3):2481–2488. 19th IFAC World Congress.

Feller, C. and Ebenbauer, C. (2015). Weight recentered barrier functions and smooth polytopic terminal set formulations for linear model predictive control. In *2015 American Control Conference (ACC)*, pages 1647–1652.

Fontes, F.A.C.C. (2001). A general framework to design stabilizing nonlinear model predictive controllers. *Systems & Control Letters*, 42(2):127–143.

Fontes, F.A.C.C., Magni, L., and Gyurkovics, E. (2007). *Assessment and Future Directions of Nonlinear Model Predictive Control*, chapter Sampled-Data Model Predictive Control for Nonlinear Time-Varying Systems: Stability and Robustness, pages 115–129. Springer, Berlin, Heidelberg.

Franklin, G.F., Powell, J.D., and Workman, M. (1997). *Digital Control of Dynamic Systems*. Prentice Hall, Upper Saddle River, NJ, 3rd edition.

Graichen, K. und Käpernick, B. (2012a). *Frontiers of Model Predictive Control*, chapter A real-time gradient method for nonlinear model predictive control, pages 9–28. InTech.

Graichen, K. (2012b). A fixed-point iteration scheme for real-time model predictive control. *Automatica*, 48(7):1300–1305.

Graichen, K. and Kugi, A. (2010). Stability and incremental improvement of suboptimal MPC without terminal constraints. *IEEE Transactions on Automatic Control*, 55(11).

Griva, I., Nash, S.G., and Sofer, A. (2009). *Linear and Nonlinear Optimization*. Society for Industrial and Applied Mathematics, 2nd edition.

Gros, S., Zanon, M., Quirynen, R., Bemporad, A., and Diehl, M. (2016). From linear to nonlinear MPC: bridging the gap via the real-time iteration. *Accepted for publication in International Journal of Control*, pages 1–19.

Grüne, L. and Pannek, J. (2010). Analysis of unconstrained NMPC schemes with incomplete optimization. *Proceedings of the 8th IFAC Symposium on Nonlinear Control Systems, Bologna, Italy*, pages 238–243.

Hairer, E., Nørsett, S.P., and Wanner, G. (1993). *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer, Berlin, New York, 2nd edition.

Khalil, H.K. (2002). *Nonlinear Systems*. Prentice Hall, New Jersey, 3rd edition.

Kristić, M., Kanellakopoulos, I., and Kokotović, P. (1995). *Nonlinear and Adaptive Control Design*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley & Sons, New York, Chichester.

Luenberger, D.G. and Ye, Y. (2008). *Linear and Nonlinear Programming*. International Series in Operations Research and Management Science. Springer, New York, 3rd edition.

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299

The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

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

Nešić, D. and Grüne, L. (2006). A receding horizon control approach to sampled-data implementation of continuous-time controllers. *Systems & Control Letters*, 55(8):660–672.

Nešić, D. and Teel, A.R. (2004). A framework for stabilization of nonlinear sampled-data systems based on their approximate discrete-time models. *IEEE Transactions on Automatic Control*, 49(7):1103–1122.

Nešić, D., Teel, A.R., and Kokotović, P.V. (1999). Sufficient conditions for stabilization of sampled-data nonlinear systems via discrete-time approximations. *Systems & Control Letters*, 38(4-5):259–270.

Nocedal, J. and Wright, S.J. (2006). *Numerical Optimization*. Springer Series in Operations Research. Springer, New York, 2nd edition.

Rawlings, J.B. and Mayne, D.Q. (2009). *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, Wisconsin.

Sastry, S. (1999). *Nonlinear Systems: Analysis, Stability, and Crontrol*. Springer, New York.

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

Stoer, J. and Bulirsch, R. (2002). *Introduction to Numerical Analysis*. Number 12 in Texts in Applied Mathematics. Springer, New York, Berlin, 3rd edition.

Streif, S., Kögel, M., Bäthge, T., and Findeisen, R. (2014). Robust nonlinear model predictive control with constraint satisfaction: A relaxation-based approach. *IFAC Proceedings Volumes*, 47(3):11073–11079.

Van Loan, C.F. (1978). Computing integrals involving the matrix exponential. *IEEE Transactions on Automatic Control*, 23(3):395–404.

Wächter, A. and Biegler, L.T. (2006). On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57.

Wills, A.G. and Heath, W.P. (2004). Barrier function based model predictive control. *Automatica*, 40(8):1415–1422.

Worthmann, K., Mehrez, M. W., Zanon, M., Mann, G. K. I., Gosine, R. G., and Diehl, M. (2016). Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs. *IEEE Transactions on Control Systems Technology*, 24(4):1394–1406.

Worthmann, K., Reble, M., Grüne, L., and Allgöwer, F. (2014). The role of sampling for stability and performance in unconstrained nonlinear model predictive control. *SIAM Journal on Control and Optimization*, 52(1):581–605.

Yang, X. and Biegler, L.T. (2013). Advanced-multi-step nonlinear model predictive control. *Journal of Process Control*, 23(8):1116–1128.

Yu, M. and Biegler, L.T. (2016). A stable and robust NMPC strategy with reduced models and nonuniform grids. *IFAC Proceedings Volumes*, 49(7):31–36.

Yu, S., Chen, H., Böhm, C., and Allgöwer, F. (2009). *Nonlinear Model Predictive Control: Towards New Challenging Applications*, chapter Enlarging the Terminal Region of NMPC with Parameter-Dependent Terminal Control Law, pages 69–78. Springer, Berlin, Heidelberg.

Zanelli, A., Quirynen, R., and Diehl, M. (2016). An efficient inexact NMPC scheme with stability and feasibility guarantees. In *Proceedings of 10th IFAC Symposium on Nonlinear Control Systems*, Monterey, CA, USA.

Zavala, V.M., Laird, C.D., and Biegler, L.T. (2008). Fast implementations and rigorous models: Can both be accommodated in NMPC? *International Journal of Robust and Nonlinear Control*, 18(8):800–815.

Zeilinger, M.N., Jones, C.N., and Morari, M. (2011). Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization. *IEEE Transactions on Automatic Control*, 56(7):1524–1534.

Appendix A. Stability analysis

This appendix contains the proof of convergence and asymptotic stability of the developed control scheme.

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299

The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

A.1 *Assumptions*

We state three key assumptions that influence the choice of V and v. Let $\mathcal{V}_f(\alpha)$ with arbitrary $\alpha \in [0, \infty)$ be a level set of $V(\boldsymbol{x})$ in the form $\mathcal{V}_f(\alpha) = \{\boldsymbol{x} \in \mathcal{X}_f \,|\, V(\boldsymbol{x}) \leq \alpha\}$. Similarly, let $\mathcal{V}(\alpha)$ with arbitrary $\alpha \in [0, \infty)$ be a level set of $v(\boldsymbol{x}, \boldsymbol{u})$ in the form $\mathcal{V}(\alpha) = \{(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{X} \times \mathcal{U} \,|\, v(\boldsymbol{x}, \boldsymbol{u}) \leq \alpha\}$.

Assumption 1: *For arbitrary scalar values $0 < \alpha_1 < \alpha_2 < \infty$, the objective functions $V : \mathcal{X}_f \to \mathbb{R}_{\geq 0}$ and $v : \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$ satisfy the strict set inclusions*

$$\begin{aligned} \mathcal{X}_d = \mathcal{V}_f(0) \subset \mathcal{V}_f(\alpha_1) \subset \mathcal{V}_f(\alpha_2) \subset \lim_{\alpha \to \infty} \mathcal{V}_f(\alpha) = \mathcal{X}_f \\ \mathcal{X}_d \times \mathcal{U} = \mathcal{V}(0) \subset \mathcal{V}(\alpha_1) \subset \mathcal{V}(\alpha_2) \subset \lim_{\alpha \to \infty} \mathcal{V}(\alpha) = \mathcal{X} \times \mathcal{U}. \end{aligned}$$

Remark 3: *Because the set inclusions in Assumption 1 are strict, V and v are barrier functions, which ensure satisfaction of $\hat{\boldsymbol{x}}_t(T_N) \in \mathring{\mathcal{X}}_f$ and $\hat{\boldsymbol{x}}_t(\tau) \in \mathring{\mathcal{X}}$ $\forall\, \tau \in [t, T_N]$, respectively.*

Assumption 2: *For any $\boldsymbol{x}_0 \in \mathring{\mathcal{X}}$, there exists an input parameterization defined by the pair $\boldsymbol{T}$ and $\boldsymbol{\theta} \in \mathcal{T}(\boldsymbol{T})$ with $T_0 = 0$ and sufficiently large values $T_N < \infty$ and $N < \infty$ so that $\hat{\boldsymbol{x}}_0^{\boldsymbol{T},\boldsymbol{\theta}}$ is bounded and satisfies*

$$\begin{aligned} &\hat{\boldsymbol{x}}_0^{\boldsymbol{T},\boldsymbol{\theta}}(t) \in \mathring{\mathcal{X}}, \qquad \forall\, t \in [0, T_N) \\ &\hat{\boldsymbol{x}}_0^{\boldsymbol{T},\boldsymbol{\theta}}(T_N) \in \mathring{\mathcal{X}}_f. \end{aligned}$$

Assumption 2 is also used in (Alamir, 2006; Michalska and Mayne, 1993; Rawlings and Mayne, 2009; Scokaert et al., 1999; Streif et al., 2014; Yu et al., 2009). It guarantees the existence of a feasible solution of (6) at any time. Clearly, in practice, finding this solution can be a laborious task, which may, for instance, involve nonlinear programming (Bertsekas, 1999; Griva et al., 2009; Luenberger and Ye, 2008; Nocedal and Wright, 2006).

Assumption 3: *The solutions of* (6) *constitute a simply connected set. Outside this set, J does not have any local minima or stationary points.*

The existence of a feasible initial solution of (6) is guaranteed by Assumption 2. If such an initial solution is known, Assumption 3 ensures that a global optimal solution of (6) within the set $\mathcal{T}(\boldsymbol{T})$ can always be found by a simple search along the local descent direction of J. Because of Assumption 3, minimizers are generally nonstrict and constitute a simply connected set. Assumption 3 avoids the problem of being caught in a local optimum that is not a global optimum. This assumption simplifies the analysis in this paper and may be relaxed in practical applications.

Assumption 4: *For any $\boldsymbol{x}(t) \in \mathring{\mathcal{X}}_f$, there exists an input parameterization defined by $\boldsymbol{T}$ and $\boldsymbol{\theta} \in \mathcal{T}(\boldsymbol{T})$ with $T_0 = t$, $T_N \to \infty$ and $N \to \infty$ so that $\hat{\boldsymbol{x}}_t^{\boldsymbol{T},\boldsymbol{\theta}}$ is bounded and satisfies*

$$\begin{aligned} &\hat{\boldsymbol{x}}_t^{\boldsymbol{T},\boldsymbol{\theta}}(\tau) \in \mathring{\mathcal{X}} \qquad \forall\, \tau \geq t \\ &\hat{\boldsymbol{x}}_t^{\boldsymbol{T},\boldsymbol{\theta}}(T_j) \in \mathring{\mathcal{X}}_f \qquad \forall\, j = 0, 1, 2, \ldots \\ &\lim_{\tau \to \infty} \hat{\boldsymbol{x}}_t^{\boldsymbol{T},\boldsymbol{\theta}}(\tau) \in \mathcal{X}_d. \end{aligned}$$

To actually compute the entries of $\boldsymbol{T}$ and $\boldsymbol{\theta}$ from Assumption 4, we may use the following approach. We set $T_0 = t$, consider the initial value $\boldsymbol{x}(t)$, use the sampling period $\kappa_T \in \mathbb{R}_{\geq 0}$, and recursively apply the state feedback function $\boldsymbol{\kappa} : \mathring{\mathcal{X}}_f \to \mathbb{R}^{n_\vartheta}$ so that

$$T_{k+1} = T_k + \kappa_T \qquad \forall\, k = 0, 1, 2, \ldots \tag{A1a}$$

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

$$\boldsymbol{\vartheta}_{k+1} = \boldsymbol{\kappa}(\hat{\boldsymbol{x}}_t^{\boldsymbol{T},\boldsymbol{\theta}}(T_k)) \qquad \forall\, k = 0, 1, 2, \ldots \tag{A1b}$$

and

$$\bar{\boldsymbol{w}}(t, T_k, T_{k+1}, \boldsymbol{\vartheta}_{k+1}) \in \mathcal{U} \qquad \forall\, t \in [T_k, T_{k+1}),\ k = 0, 1, 2, \ldots$$

It is assumed that the resulting values $\boldsymbol{T}$ and $\boldsymbol{\theta}$ satisfy the requirements of Assumption 4. Clearly, $\mathring{\mathcal{X}}_f$ is a positive control invariant set of the system (1) with sampled-data control defined by the feedback function $\boldsymbol{\kappa}$. The existence of such a feedback function is sufficient though not necessary for Assumption 4.

For the discrete-time implementation of the control algorithm, we assume that the discrete-time optimal control problem (10) satisfies the discrete-time companions of all assumptions made in this Section.

A.2 *Convergence of receding horizon control*

The following theorem describes properties of the recurrently applied function $\boldsymbol{U}$ that are sufficient to achieve the stated control objective.

Theorem 1: *Given that Assumptions 1–4 hold, that an initially feasible solution* $(\boldsymbol{T}_0, \boldsymbol{\theta}_0)$ *satisfying* $\boldsymbol{\theta}_0 \in \mathcal{T}(\boldsymbol{T}_0)$ *is known, that the update function* $\boldsymbol{U}$ *is applied at discrete times* τ_i $(i = 0, 1, 2, \ldots)$*, and that* $\boldsymbol{U}$ *always ensures* (7)*,* $\boldsymbol{\theta}_{i+1} \in \mathcal{T}(\boldsymbol{T}_{i+1})$*, and*

$$J(\hat{\boldsymbol{x}}_{\tau_i}^{\boldsymbol{T}_{i+1},\boldsymbol{\theta}_{i+1}}(\,\cdot\,), \boldsymbol{w}(\,\cdot\,, \boldsymbol{T}_{i+1}, \boldsymbol{\theta}_{i+1}), \tau_i, \boldsymbol{T}_{i+1}) \begin{cases} = 0 & \text{if } \boldsymbol{x}(\tau_i) \in \mathcal{X}_d \\ \leq J(\hat{\boldsymbol{x}}_{\tau_i}^{\boldsymbol{T}_i,\boldsymbol{\theta}_i}(\,\cdot\,), \boldsymbol{w}(\,\cdot\,, \boldsymbol{T}_i, \boldsymbol{\theta}_i), \tau_i, \boldsymbol{T}_i) & \text{otherwise,} \end{cases} \tag{A2}$$

then, in the absence of disturbances and for any initial state $\boldsymbol{x}_0 \in \mathring{\mathcal{X}}$*, the state* $\boldsymbol{x}$ *of the closed-loop system*

- *converges to the set* $\mathcal{V}_f(\alpha)$ *for any constant value* $\alpha \in (0, \infty)$ *within finite time,*
- *asymptotically converges to the set* $\mathcal{X}_d$*, and*
- *satisfies the constraints* (2a)*.*

Proof:

Assumptions 2 and 4 ensure the existence of a pair $(\boldsymbol{T}, \boldsymbol{\theta})$ with $T_N \to \infty$ that asymptotically brings $\boldsymbol{x}$ from any $\boldsymbol{x}_0 \in \mathring{\mathcal{X}}$ to the set $\mathcal{X}_d$. Feasibility of the known initial solution $(\boldsymbol{T}_0, \boldsymbol{\theta}_0)$ with $T_{0|N_0} < \infty$ implies $J(\hat{\boldsymbol{x}}_0^{\boldsymbol{T}_0,\boldsymbol{\theta}_0}(\,\cdot\,), \boldsymbol{w}(\,\cdot\,, \boldsymbol{T}_0, \boldsymbol{\theta}_0), \tau_0, \boldsymbol{T}_0) < \infty$. Hence, (A2) ensures that the constraints (2a) are satisfied. By Assumption 1, it follows that $J = 0 \Leftrightarrow \boldsymbol{x}(\tau) \in \mathcal{X}_d\, \forall\, \tau \in [\tau_i, T_N]$. We distinguish between the two cases used in (A2):

- If the initial state $\boldsymbol{x}(\tau_i)$ satisfies $\boldsymbol{x}(\tau_i) \in \mathcal{X}_d$, $J(\hat{\boldsymbol{x}}_{\tau_i}^{\boldsymbol{T}_{i+1},\boldsymbol{\theta}_{i+1}}(\,\cdot\,), \boldsymbol{w}(\,\cdot\,, \boldsymbol{T}_{i+1}, \boldsymbol{\theta}_{i+1}), \tau_i, \boldsymbol{T}_{i+1}) = 0$ ensures that the state will remain in $\mathcal{X}_d$ during the new horizon $[\tau_i, \tau_{i+1})$. Hence, it will remain there forever.
- If $\boldsymbol{x}(\tau_i) \notin \mathcal{X}_d$, initial feasibility and the inequality in (A2) ensure that $\boldsymbol{x}(\tau_i) \in \mathcal{X}$. Because $\boldsymbol{x}(\tau_i) \notin \mathcal{X}_d$ and $\Delta\tau_i > \Delta\tau_{\min} > 0$, the following difference, i. e., the change of J over the

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

interval (τ_i, τ_{i+1}), is strictly negative and does not converge to 0 unless $\boldsymbol{x}(\tau_i) \in \mathcal{X}_d$:

$$\begin{aligned} &J(\hat{\boldsymbol{x}}_{\tau_{i+1}}^{\boldsymbol{T}_{i+1},\boldsymbol{\theta}_{i+1}}(\,\cdot\,), \boldsymbol{w}(\,\cdot\,, \boldsymbol{T}_{i+1}, \boldsymbol{\theta}_{i+1}), \tau_{i+1}, \boldsymbol{T}_{i+1}) \\ &\quad - J(\hat{\boldsymbol{x}}_{\tau_i}^{\boldsymbol{T}_{i+1},\boldsymbol{\theta}_{i+1}}(\,\cdot\,), \boldsymbol{w}(\,\cdot\,, \boldsymbol{T}_{i+1}, \boldsymbol{\theta}_{i+1}), \tau_i, \boldsymbol{T}_{i+1}) \\ &= -\int_{\tau_i}^{\tau_{i+1}} v(\hat{\boldsymbol{x}}_{\tau_i}^{\boldsymbol{T}_{i+1},\boldsymbol{\theta}_{i+1}}(t), \boldsymbol{w}(t, \boldsymbol{T}_{i+1}, \boldsymbol{\theta}_{i+1}))\mathrm{d}t \\ &\leq -\Delta J(\boldsymbol{x}(\tau_i), \Delta\tau_i) \leq -\delta(\alpha) \leq 0 \end{aligned} \tag{A3}$$

Here, $\Delta J(\boldsymbol{x}(\tau_i), \Delta\tau_i)$ is the result of the dynamic optimization problem

$$\begin{aligned} \Delta J(\boldsymbol{x}(\tau_i), \Delta\tau_i) = \min_{\boldsymbol{u}(\,\cdot\,)\in\mathcal{U}} &\int_0^{\Delta\tau_i} v(\hat{\boldsymbol{x}}(t), \boldsymbol{u}(t))\mathrm{d}t \\ \text{s. t. } &\dot{\hat{\boldsymbol{x}}} = \boldsymbol{f}(\hat{\boldsymbol{x}}, \boldsymbol{u}) \\ &\hat{\boldsymbol{x}}(0) = \boldsymbol{x}(\tau_i) \\ &\hat{\boldsymbol{x}} \in \mathcal{X}. \end{aligned}$$

This shows that $\boldsymbol{x}(\tau_i) \notin \mathcal{X}_d \Rightarrow \Delta J(\boldsymbol{x}(\tau_i), \Delta\tau_i) > 0$, which implies together with (A2) that $\lim_{t\to\infty} \boldsymbol{x}(t) \in \mathcal{X}_d$, i. e., the state $\boldsymbol{x}$ of the closed-loop system asymptocially converges to the set $\mathcal{X}_d$ (Khalil, 2002). Moreover, $\delta(\alpha)$ is the result of the minimization problem

$$\delta(\alpha) = \inf_{\boldsymbol{x}\in\mathcal{X}\setminus\mathcal{V}_f(\alpha)} \Delta J(\boldsymbol{x}, \min_i \Delta\tau_i).$$

This shows that during any interval (τ_i, τ_{i+1}) the cost function J decreases at least by $\delta(\alpha)$ for any $\alpha \in (0, \infty)$ as long as $\boldsymbol{x}(\tau_i) \notin \mathcal{V}_f(\alpha)$. Therefore, if (A2) is satisfied and $\boldsymbol{x}(\tau_i) \in \mathcal{V}_f(\alpha_0)$ with some $\alpha_0 \in (0, \infty)$, the state $\boldsymbol{x}$ of the closed-loop system converges to the set $\mathcal{V}_f(\alpha)$ for any constant value $\alpha \in (0, \alpha_0]$ at least within the finite time $\tau_{i+\lceil(\alpha_0-\alpha)/\delta(\alpha)\rceil} - \tau_i$, where $\lceil\,\cdot\,\rceil$ represents the ceiling function.

■

Remark 4: *Based on Lyapunov-type inequalities, Theorem 1 guarantees convergence of the nominal, undisturbed system. For this system, the change of J during the interval (τ_i, τ_{i+1}) is strictly negative (cf.* (A3)*), which is a source of robustness against model errors and disturbances. As demonstrated by Michalska and Mayne (1993), additional robustness can also be achieved by conservative design of the constraint sets. An analysis of robustness against model mismatch can be done also for the design proposed in this paper by full analogy to (Michalska and Mayne, 1993).*

Appendix B. Alternative update strategy for the time grid

In Section 3.2, we proposed an update strategy for the time grid with a uniform step size $\Delta T_{i|j}$. In order to achieve low-dimensional optimization problems, it may however be desirable to define control inputs $\boldsymbol{u}$ on non-uniform time grids where the step sizes $\Delta T_{i|j}$ are as large as possible. Generally, $\Delta T_{i|j}$ should be small at the beginning of the horizon and may be larger towards its end because the discretization error of an RHC input $\boldsymbol{u}$ should be small at the beginning of the horizon and may be larger towards its end. An update strategy that resembles this idea is known as move-blocking strategy (Cagienard et al., 2007; Yu and Biegler, 2016). We present a similar strategy as an alternative to the approach defined in Section 3.2.

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.

We require $\Delta T_{i|j+1}$ to be an integer multiple of $\Delta T_{i|j}$. The strategy is uniquely defined in the following way: For each horizon i (i. e., at the update point τ_{i-1}), ensure that $T_{i|N_i}$ and N_i are minimum and that

$$\begin{aligned}
&\Delta T_{i|0} = \frac{\kappa_T}{(N_T)^{N_{\min}-1}} \quad \wedge \quad \Delta T_{i|N_i-1} = \kappa_T \quad \wedge \\
&(\Delta T_{i|j} = \Delta T_{i|j-1} \ \vee \ \Delta T_{i|j} = N_T \Delta T_{i|j-1} \\
&\qquad\qquad \forall\, j = 1, \ldots, N_i - 1) \quad \wedge \\
&(T_{i-1|j} < T_{i|0} \ \vee \ \exists\, \bar{j} : T_{i|\bar{j}} = T_{i-1|j} \\
&\qquad\qquad \forall\, j = 1, \ldots, N_{i-1}).
\end{aligned}$$

Therefore, N_i varies between $N_{\min}$ and $N_T(N_{\min} - 1) + 1$. The above conditions ensure that the step size of the grid defined by $\boldsymbol{T}_i$ grows from $\kappa_T/(N_T)^{N_{\min}-1}$ at the beginning of the horizon to κ_T at the end of the horizon. Step sizes of neighboring sampling intervals are either equal or grow by the factor N_T. Moreover, these conditions ensure that sampling points defined in $\boldsymbol{T}_{i-1}$ are reused if they are inside the interval $[T_{i|0}, T_{i|N_i}]$. Figure B1 shows an example of prediction horizons generated using this strategy with $N_T = 2$ and $N_{\min} = 3$. For clarity of presentation, the time grid τ_i is given for fundamental updates only (incremental updates are not considered).

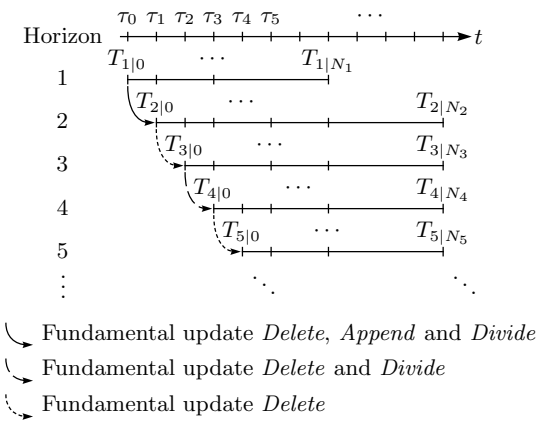

Figure B1. Prediction horizons of the alternative update strategy for $N_T = 2$ and $N_{\min} = 3$ (time grid τ_i only for fundamental updates).

Post-print version of the article: A. Steinboeck, M. Guay, and A. Kugi, "A design technique for fast sampled-data nonlinear model predictive control with convergence and stability results", *International Journal of Control*, vol. 93, no. 1, pp. 81–97, 2020. DOI: 10.1080/00207179.2017.1346299
The content of this post-print version is identical to the published paper but without the publisher's final layout or copy editing.