We are IntechOpen,
the world's leading publisher of
Open Access books
Built by scientists, for scientists

6,900
Open access books available

185,000
International authors and editors

200M
Downloads

154
Countries delivered to

Our authors are among the

TOP 1%
most cited scientists

12.2%
Contributors from top 500 universities

CLARIVATE ANALYTICS
BOOK CITATION INDEX
INDEXED

WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

1

A Real-Time Gradient Method for Nonlinear Model Predictive Control

Knut Graichen and Bartosz Käpernick
Institute of Measurement, Control and Microtechnology
University of Ulm
Germany

1. Introduction

Model predictive control (MPC) is a modern control scheme that relies on the solution of an optimal control problem (OCP) on a receding horizon. MPC schemes have been developed in various formulations (regarding continuous/discrete-time systems, finite/infinite horizon length, terminal set/equality constraints, etc.). Comprehensive overviews and references on MPC can, for instance, be found in Diehl et al. (2009); Grüne & Pannek (2011); Kothare & Morari (2000); Mayne et al. (2000); Rawlings & Mayne (2009).

Although the methodology of MPC is naturally suited to handle constraints and multiple-input systems, the iterative solution of the underlying OCP is in general computationally expensive. An intuitive approach to reducing the computational load is to solve the OCP approximately, for instance, by using a fixed number of iterations in each sampling step. In the next MPC step, the previous solution can be used for a warm-start of the optimization algorithm in order to successively reduce the suboptimality of the predicted trajectories. This incremental strategy differs from the "optimal" MPC case where the (numerically exact) OCP solution is assumed to be known.

There exist various suboptimal and real-time approaches in the literature with different kinds of terminal constraints and demands on the optimization algorithm (Cannon & Kouvaritakis, 2002; DeHaan & Guay, 2007; Diehl et al., 2005; Graichen & Kugi, 2010; Lee et al., 2002; Michalska & Mayne, 1993; Ohtsuka, 2004; Scokaert et al., 1999). In particular, the approaches of Ohtsuka (2004) and Diehl et al. (2005) are related to the MPC scheme presented in this chapter. In Ohtsuka (2004), an algorithm is developed that traces the solution of the discretized optimality conditions over the single sampling steps. The real-time iteration scheme presented by Diehl et al. (2005) uses a Newton scheme together with terminal constraints in order to compute an approximate solution that is refined in each sampling step.

Suboptimal MPC schemes require special attention regarding their convergence and stability properties. This is particularly important if an MPC formulation without terminal constraints is used in order to minimize the computational complexity and to allow for a real-time implementation for very fast dynamical systems. In this context, a suboptimal MPC approach without terminal constraints was investigated in Graichen & Kugi (2010). Starting from the assumption that an optimization algorithm with a linear rate of convergence exists, it is

www.intechopen.com

shown that exponential stability of the closed-loop system as well as exponential decay of the suboptimality can be guaranteed if the number of iterations per sampling step satisfies a lower bound (Graichen & Kugi, 2010). The decay of the suboptimality also illustrates the incremental improvement of the MPC scheme.

Based on these theoretical considerations (Graichen & Kugi, 2010), this chapter presents a real-time MPC scheme that relies on the gradient method in optimal control (Dunn, 1996; Graichen et al., 2010; Nikol'skii, 2007). This algorithm is particularly suited for a real-time implementation, as it takes full advantage of the MPC formulation without terminal constraints. In addition, the gradient method allows for a memory and time efficient computation of the single iterations, which is of importance in order to employ the MPC scheme for fast dynamical systems.

In this chapter, the gradient-based MPC algorithm is described for continuous-time nonlinear systems subject to control constraints. Starting from the general formulation of the MPC problem, the stability properties in the optimal MPC case are summarized before the suboptimal MPC strategy is discussed. As a starting point for the derivation of the gradient method, the necessary optimality conditions for the underlying OCP formulation without terminal constraints are derived from Pontryagin's Maximum Principle. Based on the optimality conditions, the gradient algorithm is described and its particular implementation within a real-time MPC scheme is detailed. The algorithm as well as its properties and incremental improvement in the MPC scheme are numerically investigated for the double pendulum on a cart, which is a benchmark in nonlinear control. The simulation results as well as the CPU time requirements reveal the efficiency of the gradient-based MPC scheme.

2. MPC formulation

We consider a nonlinear continuous-time system of the form

$$\dot{x}(t) = f(x(t), u(t)) \quad x(t_0) = x_0, \quad t \geq t_0 \tag{1}$$

with the state $x \in \mathbb{R}^n$ and the control $u \in \mathbb{R}^m$ subject to the control constraints

$$u(t) \in [u^-, u^+]. \tag{2}$$

Without loss of generality, we assume that the origin is an equilibrium of the system (1) with $f(0,0) = 0$. Moreover, the system function f is supposed to be continuously differentiable in its arguments. This section summarizes the MPC formulation as well as basic assumptions and basic results for the stability of the MPC scheme in closed-loop.

2.1 Optimal control problem

For stabilizing the origin of the system (1), an MPC scheme based on the following optimal control problem (OCP) is used

$$\min_{\bar{u} \in \mathcal{U}_{[0,T]}} \quad J(x_k, \bar{u}) = V(\bar{x}(T)) + \int_0^T l(\bar{x}(\tau), \bar{u}(\tau)) \, d\tau \tag{3a}$$

$$\text{s.t.} \quad \dot{\bar{x}}(\tau) = f(\bar{x}(\tau), \bar{u}(\tau)), \quad \bar{x}(0) = x_k = x(t_k), \tag{3b}$$

where $\mathcal{U}_{[0,T]}$ is the admissible input space

$$\mathcal{U}_{[0,T]} := \{u(\cdot) \in L_\infty^m[0,T] : u(t) \in [u^-, u^+], \, t \in [0,T]\}. \tag{4}$$

The initial condition $x(t_k) = x_k$ in (3b) denotes the measured (or observed) state of the system (1) at time $t_k = t_0 + k\Delta t$ with the sampling time Δt. The bared variables $\bar{x}(\tau)$, $\bar{u}(\tau)$ represent internal variables of the controller with the MPC prediction time coordinate $\tau \in [0,T]$ and the horizon length $T \geq \Delta t$.

The integral and the terminal cost functions in (3a) are assumed to be continuously differentiable and to satisfy the quadratic bounds

$$m_l(||x||^2 + ||u||^2) \leq l(x,u) \leq M_l(||x||^2 + ||u||^2)$$
$$m_V||x||^2 \leq V(x) \leq M_V||x||^2 \tag{5}$$

for some constants $m_l, M_l > 0$ and $m_V, M_V > 0$. The optimal solution of OCP (3) is denoted by

$$\bar{u}_k^*(\tau) := \bar{u}^*(\tau; x_k), \quad \bar{x}_k^*(\tau) := \bar{x}^*(\tau; x_k, \bar{u}_k^*), \quad \tau \in [0,T], \quad J^*(x_k) := J(x_k, \bar{u}_k^*). \tag{6}$$

To obtain a stabilizing MPC feedback law on the sampling interval $[t_k, t_{k+1})$, the first part of the optimal control $\bar{u}_k^*(\tau)$ is used as control input for the system (1)

$$u(t_k + \tau) = \bar{u}_k^*(\tau) =: \kappa(\bar{x}_k^*(\tau); x_k), \quad \tau \in [0, \Delta t), \tag{7}$$

which can be interpreted as a nonlinear "sampled" control law with $\kappa(0; x_k) = 0$. In the next MPC step at time t_{k+1}, OCP (3) is solved again with the new initial condition x_{k+1}. In the absence of model errors and disturbances, the next point x_{k+1} is given by $x_{k+1} = \bar{x}_k^*(\Delta t)$ and the closed-loop trajectories are

$$x(t) = x(t_k + \tau) = \bar{x}^*(\tau; x_k),$$
$$u(t) = u(t_k + \tau) = \bar{u}^*(\tau; x_k), \quad \tau \in [0, \Delta t), \quad k \in \mathbb{N}_0^+. \tag{8}$$

2.2 Domain of attraction and stability

The following lines summarize important results for the "optimal" MPC case without terminal constraints, i.e. when the optimal solution (6) of OCP (3) is assumed to be known in each sampling step. These results are the basis for the suboptimal MPC case treated in Section 3. Some basic assumptions are necessary to proceed:

Assumption 1. *For every $x_0 \in \mathbb{R}^n$ and $u \in \mathcal{U}_{[0,T]}$, the system (1) has a bounded solution over $[0,T]$.*

Assumption 2. *OCP (3) has an optimal solution (6) for all $x_k \in \mathbb{R}^n$.*

Since u is constrained, Assumption 1 is always satisfied for systems without finite escape time. Moreover, note that the existence of a solution of OCP (3) in Assumption 2 is not very restrictive as no terminal constraints are considered and all functions are assumed to be continuously differentiable.[1].

---

[1] Theorems on existence and uniqueness of solutions for certain classes of OCPs can, for instance, be found in Berkovitz (1974); Lee & Markus (1967).

An MPC formulation without terminal constraints has been subject of research by several authors, see for instance Graichen & Kugi (2010); Ito & Kunisch (2002); Jadbabaie et al. (2001); Limon et al. (2006); Parisini & Zoppoli (1995). Instead of imposing a terminal constraint, it is often assumed that the terminal cost V represents a (local) Control Lyapunov Function (CLF) on an invariant set S_β containing the origin.

Assumption 3. *There exists a compact non-empty set $S_\beta = \{x \in \mathbb{R}^n : V(x) \leq \beta\}$ and a (local) feedback law $q(x) \in [u^-, u^+]$ such that $\forall x \in S_\beta$*

$$\frac{\partial V}{\partial x} f(x, q(x)) + l(x, q(x)) \leq 0. \tag{9}$$

There exist several approaches in the literature for constructing a CLF as terminal cost, for instance Chen & Allgöwer (1998); Primbs (1999). In particular, $V(x)$ can be designed as a quadratic function $V(x) = x^\mathsf{T} P x$ with the symmetric and positive definite matrix P following from a Lyapunov or Riccati equation provided that the linearization of the system (1) about the origin is stabilizable.

An important requirement for the stability of an MPC scheme without terminal constraints is to ensure that the endpoint of the optimal state trajectory $\bar{x}_k^*(T)$ reaches the CLF region S_β. The following theorem states this property more clearly and relates it to the overall stability of the (optimal) MPC scheme.

Theorem 1 (**Stability of MPC scheme – optimal case**). *Suppose that Assumptions 1-3 are satisfied and consider the compact set*

$$\Gamma_\alpha = \{x \in \mathbb{R}^n : J^*(x) \leq \alpha\}, \quad \alpha := \beta\left(1 + \frac{m_l}{M_V}T\right). \tag{10}$$

Then, for all $x_0 \in \Gamma_\alpha$ the following holds:

1. *For all MPC steps, it holds that $x_k \in \Gamma_\alpha$. Moreover, the endpoint of the optimal state trajectory $\bar{x}_k^*(\tau)$, $\tau \in [0, T]$ reaches the CLF region, i.e. $\bar{x}_k^*(T) \in S_\beta$.*

2. *Γ_α contains the CLF region, i.e. $S_\beta \subseteq \Gamma_\alpha$.*

3. *The optimal cost satisfies*

$$J^*(\bar{x}_k^*(\Delta t)) \leq J^*(x_k) - \int_0^{\Delta t} l(\bar{x}_k^*(\tau), \bar{u}_k^*(\tau)) \, d\tau \quad \forall x_k \in \Gamma_\alpha. \tag{11}$$

4. *The origin of the system (1) under the optimal MPC law (7) is asymptotically stable in the sense that the closed-loop trajectories (8) satisfy $\lim_{t \to \infty} ||x(t)|| = 0$.*

The single statements 1-4 in Theorem 1 are discussed in the following:

1. The sublevel set Γ_α defines the domain of attraction for the MPC scheme without terminal constraints (Graichen & Kugi, 2010; Limon et al., 2006). The proof of this statement is given in Appendix A.

2. Although α in (10) leads to a rather conservative estimate of Γ_α due to the nature of the proof (see Appendix A), it nevertheless reveals that Γ_α can be enlarged by increasing the horizon length T.

3. The decrease condition (11) for the optimal cost at the next point $x_{k+1} = \bar{x}_k^*(\Delta t)$ follows from the CLF property (9) on the set S_β (Jadbabaie et al., 2001). Indeed, consider the trajectories

$$\hat{x}(\tau) = \begin{cases} \bar{x}_k^*(\tau + \Delta t), & \tau \in [0, T - \Delta t) \\ \bar{x}^q(\tau - T + \Delta t), & \tau \in [T - \Delta t, T] \end{cases}, \quad \hat{u}(\tau) = \begin{cases} \bar{u}_k^*(\tau + \Delta t), & \tau \in [0, T - \Delta t) \\ \bar{u}^q(\tau - T + \Delta t), & \tau \in [T - \Delta t, T] \end{cases}$$

where $\bar{x}^q(\tau)$ with $\bar{x}^q(0) = \bar{x}_k^*(T)$ is the state trajectory that results from applying the local CLF law $\bar{u}^q(\tau) = q(\bar{x}^q(\tau))$. Note that $\bar{x}^q(\tau) \in S_\beta$ for all $\tau \geq 0$, i.e. S_β ist positive invariant due to the definition of S_β and the CLF inequality (9) that can be expressed in the form

$$\frac{\mathrm{d}}{\mathrm{d}\tau} V(\bar{x}^q(\tau)) \leq -l(\bar{x}^q(\tau), \bar{u}^q(\tau)). \tag{12}$$

Hence, the following estimates hold

$$\begin{aligned} J^*(x_k^*(\Delta t)) &\leq \int_0^T l(\hat{x}(\tau), \hat{u}(\tau)) \, \mathrm{d}\tau + V(\hat{x}(T)) \\ &= J^*(x_k) - \int_0^{\Delta t} l(\bar{x}_k^*(\tau), \bar{u}_k^*(\tau)) \, \mathrm{d}\tau \\ &\quad + \underbrace{V(\bar{x}^q(\Delta t)) - V(\bar{x}^q(0)) + \int_0^{\Delta t} l(\bar{x}^q(\tau), \bar{u}^q(\tau)) \, \mathrm{d}\tau}_{\leq 0}. \end{aligned} \tag{13}$$

4. Based on (11), Barbalat's Lemma allows one to conclude that the closed-loop trajectories (8) satisfy $\lim_{t \to \infty} ||x(t)|| = 0$, see e.g. Chen & Allgöwer (1998); Fontes (2001). Note that this property is weaker than asymptotic stability in the sense of Lyapunov, which can be proved if the optimal cost $J^*(x_k)$ is continuously differentiable (Findeisen, 2006; Fontes et al., 2007).

3. Suboptimal MPC for real-time feasibility

In practice, the exact solution of the receding horizon optimal control problem is typically approximated by a sufficiently accurate numerical solution of a suitable optimization algorithm. If the sampling time Δt is large enough, this numerical approximation will be sufficiently close to the optimal MPC case considered in the last section. However, for large-scale systems or highly dynamical systems, an accurate near-optimal solution often cannot be determined fast enough. This problem, often encountered in practice, gives rise to suboptimal MPC strategies, where an approximate solution is computed in each sampling step. This section develops the necessary changes and differences to the ideal case due to an incremental solution of the underlying OCP solution for a class of optimization algorithms.

3.1 Suboptimal solution strategy

Several suboptimal MPC strategies were already mentioned in the introduction (Cannon & Kouvaritakis, 2002; DeHaan & Guay, 2007; Diehl et al., 2005; Lee et al., 2002; Michalska & Mayne, 1993; Scokaert et al., 1999). Moreover, a suboptimal MPC scheme without terminal constraints – as considered in this chapter – was investigated in Graichen & Kugi (2010).

Instead of relying on one particular optimization method, it is assumed in Graichen & Kugi (2010) that an optimization algorithm exists that computes a control and state trajectory

$$\bar{u}_k^{(j)}(\tau) := \bar{u}^{(j)}(\tau; x_k), \quad \bar{x}_k^{(j)}(\tau) := \bar{x}^{(j)}(\tau; x_k, \bar{u}_k^{(j)}), \tau \in [0, T], \quad j \in \mathbb{N}_0^+ \tag{14}$$

in each iteration j while satisfying a linear rate of convergence

$$J(x_k, \bar{u}_k^{(j+1)}) - J^*(x_k) \leq p \left(J(x_k, \bar{u}_k^{(j)}) - J^*(x_k) \right), \quad j \in \mathbb{N}_0^+ \tag{15}$$

with a convergence rate $p \in (0, 1)$ and the limit $\lim_{j \to \infty} J(x_k, \bar{u}_k^{(j)}) = J^*(x_k)$.

In the spirit of a real-time feasible MPC implementation, the optimization algorithm is stopped after a fixed number of iterations, $j = N$, and the first part of the suboptimal control trajectory $\bar{u}_k^{(N)}(\tau)$ is used as control input

$$u(t_k + \tau) = \bar{u}_k^{(N)}(\tau), \quad \tau \in [0, \Delta t), \quad k \in \mathbb{N}_0^+ \tag{16}$$

to the system (1). In the absence of model errors and disturbances the next point x_{k+1} is given by $x_{k+1} = \bar{x}_k^{(N)}(\Delta t)$ and the closed-loop trajectories are

$$\begin{aligned} x(t) &= x(t_k + \tau) = \bar{x}^{(N)}(\tau; x_k), \\ u(t) &= u(t_k + \tau) = \bar{u}^{(N)}(\tau; x_k), \quad \tau \in [0, \Delta t), \quad k \in \mathbb{N}_0^+. \end{aligned} \tag{17}$$

Compared to the "optimal" MPC case, where the optimal trajectories (6) are computed in each MPC step k, the trajectories (14) are suboptimal, which can be characterized by the *optimization error*

$$\Delta J^{(N)}(x_k) := J(\bar{u}_k^{(N)}, x_k) - J^*(x_k) \geq 0. \tag{18}$$

In the next MPC step, the last control $\bar{u}_k^{(N)}$ (shifted by Δt) is re-used to construct a new initial control

$$\bar{u}_{k+1}^{(0)}(\tau) = \begin{cases} \bar{u}_k^{(N)}(\tau + \Delta t) & \text{if } \tau \in [0, T - \Delta t) \\ q(\bar{x}_k^{(N)}(T)) & \text{if } \tau \in [T - \Delta t, T], \end{cases} \tag{19}$$

where the last part of $\bar{u}_{k+1}^{(0)}$ is determined by the local CLF feedback law. The goal of the suboptimal MPC strategy therefore is to successively reduce the optimization error $\Delta J^{(N)}(x_k)$ in order to improve the MPC scheme in terms of optimality. Figure 1 illustrates this context.

3.2 Stability and incremental improvement

Several further assumptions are necessary to investigate the stability and the evolution of the optimization error for the suboptimal MPC scheme.

Assumption 4. *The optimal control law in* (7) *is locally Lipschitz continuous.*

Assumption 5. *For every $\bar{u} \in \mathcal{U}_{[0,T]}$, the cost $J(x_k, \bar{u})$ is twice continuously differentiable in x_k.*

Assumption 6. *For all $\bar{u} \in \mathcal{U}_{[0,T]}$ and all $x_k \in \Gamma_\alpha$, the cost $J(x_k, \bar{u})$ satisfies the quadratic growth condition $C||\bar{u} - \bar{u}_k^*||_{L_2^m[0,T]}^2 \leq J(x_k, \bar{u}) - J^*(x_k)$ for some constant $C > 0$.*
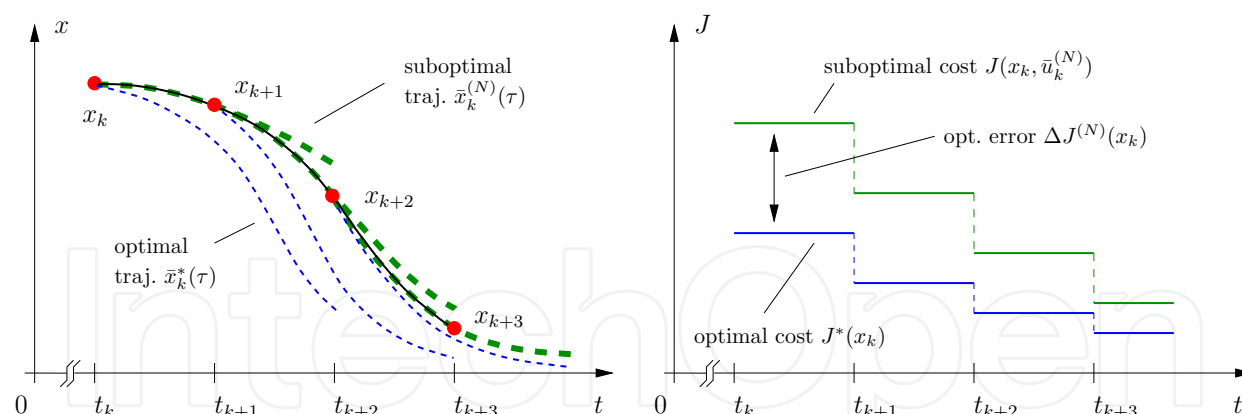
Fig. 1. Illustration of the suboptimal MPC implementation.

Assumption 6 is always satisfied for linear systems with quadratic cost functional as proved in Appendix B. In general, the quadratic growth property in Assumption 6 represents a smoothness assumption which, however, is weaker than assuming strong convexity (it is well known that strong convexity on a compact set implies quadratic growth, see, e.g., Allaire (2007) and Appendix B). [2]

The stability analysis for the suboptimal MPC case is more involved than in the "optimal" MPC case due to the non-vanishing optimization error $\Delta J^{(N)}(x_k)$. An important question in this context is under which conditions the CLF region S_β can be reached by the suboptimal state trajectory $\bar{x}_k^{(N)}(\tau)$. The following theorem addresses this question and also gives sufficient conditions for the stability of the suboptimal MPC scheme.

Theorem 2 (Stability of MPC scheme – suboptimal case). *Suppose that Assumptions 1-6 are satisfied and consider the subset of the domain* (10)

$$\Gamma_{\hat{\alpha}} = \left\{ x \in \mathbb{R}^n : J^*(x) \leq \hat{\alpha} \right\}, \quad \hat{\alpha} = \frac{m_V}{4M_V}\alpha < \alpha. \tag{20}$$

Then, there exists a minimum number of iterations $\hat{N} \geq 1$ and a maximum admissible optimization error $\Delta\hat{J} \geq 0$, such that for all $x_0 \in \Gamma_{\hat{\alpha}}$ and all initial control trajectories $\bar{u}_0^{(0)} \in \mathcal{U}_{[0,T]}$ satisfying $\Delta J^{(0)}(x_0) \leq p^{-N}\Delta\hat{J}$ the following holds:

1. *For all MPC steps, it holds that $x_k \in \Gamma_{\hat{\alpha}}$. Moreover, the endpoint of the (suboptimal) state trajectory $\bar{x}_k^{(N)}(\tau)$, $\tau \in [0, T]$ reaches the CLF region, i.e. $\bar{x}_k^{(N)}(T) \in S_\beta$.*
2. *$\Gamma_{\hat{\alpha}}$ contains the CLF region, i.e. $S_\beta \subseteq \Gamma_{\hat{\alpha}}$, if the horizon length satisfies $T \geq (\frac{4M_V}{m_V} - 1)\frac{M_V}{m_l}$.*
3. *The origin of the system* (1) *under the suboptimal MPC law* (16) *is exponentially stable.*
4. *The optimization error* (18) *decays exponentially.*

The proof of Theorem 2 consists of several intermediate lemmas and steps that are given in details in Graichen & Kugi (2010). The statements 1-4 in Theorem 2 summarize several important points that deserve some comments.

---

[2] A simple example is the function $f(x) = x^2 + 10\sin^2 x$ with the global minimum $f(x^*) = 0$ at $x^* = 0$. Let x be restricted to the interval $x \in [-5, 5]$. Clearly, the quadratic growth property $\frac{1}{2}|x - x^*|^2 \leq f(x) - f(x^*)$ is satisfied for $x \in [-5, 5]$ although $f(x)$ is not convex on this interval.

1. The reduced size of $\Gamma_{\hat{\alpha}}$ compared to Γ_{α} is the necessary "safety" margin to account for the suboptimality of the trajectories (6) characterized by $\Delta J^{(N)}(x_k)$. Thus, the domain of attraction $\Gamma_{\hat{\alpha}}$ together with an admissible upper bound on the optimization error guarantees the reachability of the CLF region S_{β}.

2. An interesting fact is that it can still be guaranteed that $\Gamma_{\hat{\alpha}}$ is at least as large as the CLF region S_{β} provided that the horizon time T satisfies a lower bound that depends on the quadratic estimates (5) of the integral and terminal cost functions. It is apparent from the bound $T \geq (\frac{4M_V}{m_V} - 1)\frac{M_V}{m_l}$ that the more dominant the terminal cost $V(x)$ is with respect to the integral cost function $l(x, u)$, the larger this bound on the horizon length T will be.

3. The minimum number of iterations $\hat{N}$ for which stability can be guaranteed ensures – roughly speaking – that the numerical speed of convergence is faster than the system dynamics. In the proof of the theorem (Graichen & Kugi, 2010), the existence of the lower bound $\hat{N}$ is shown by means of Lipschitz estimates, which usually are too conservative to be used for design purposes. For many practical problems, however, one or two iterations per MPC step are sufficient to ensure stability and a good control performance.

4. The exponential reduction of the optimization error $\Delta J^{(N)}(x_k)$ follows as part of the proof of stability and reveals the incremental improvement of the suboptimal MPC scheme over the MPC runtime.

4. Gradient projection method

The efficient numerical implementation of the MPC scheme is of importance to guarantee the real-time feasibility for fast dynamical systems. This section describes the well-known gradient projection in optimal control as well as its suboptimal implementation in the context of MPC.

4.1 Optimality conditions and algorithm

The MPC formulation without terminal constraints has particular advantages for the structure of the optimality conditions of the OCP (3). To this end, we define the Hamiltonian

$$H(x, \lambda, u) = l(x, u) + \lambda^\mathsf{T} f(x, u) \tag{21}$$

with the adjoint state $\lambda \in \mathbb{R}^n$. Pontryagin's Maximum Principle [3] states that if $\bar{u}_k^*(\tau), \tau \in [0, T]$ is an optimal control for OCP (3), then there exists an adjoint trajectory $\bar{\lambda}_k^*(\tau), \tau \in [0, T]$ such that $\bar{x}_k^*(\tau)$ und $\bar{\lambda}_k^*(\tau)$ satisfy the canonical boundary value problem (BVP)

$$\dot{\bar{x}}_k^*(\tau) = f(\bar{x}_k^*(\tau), \bar{u}_k^*(\tau)), \qquad \bar{x}_k^*(0) = x_k \tag{22}$$

$$\dot{\bar{\lambda}}_k^*(\tau) = -H_x(\bar{x}_k^*(\tau), \bar{\lambda}_k^*(\tau), \bar{u}_k^*(\tau)), \quad \bar{\lambda}_k^*(T) = V_x(\bar{x}_k^*(T)) \tag{23}$$

and $\bar{u}_k^*(\tau)$ minimizes the Hamiltonian for all times $\tau \in [0, T]$, i.e.

$$H(\bar{x}_k^*(\tau), \bar{\lambda}_k^*(\tau), \bar{u}_k^*(\tau)) \leq H(\bar{x}_k^*(\tau), \bar{\lambda}_k^*(\tau), u), \quad \forall\, u \in [u^-, u^+], \quad \forall\, \tau \in [0, T]. \tag{24}$$

---

[3] The general formulation of Pontryagin's Maximum Principle often uses the Hamiltonian definition $H(x, \lambda, u, \lambda_0) = \lambda_0\, l(x, u) + \lambda^\mathsf{T} f(x, u)$, where λ_0 accounts for "abnormal" problems as, for instance, detailed in Hsu & Meyer (1968). Typically, λ_0 is set to $\lambda_0 = 1$, which corresponds to the definition (21).

The functions H_x and V_x denote the partial derivatives of H and V with respect to x. The minimization condition (24) also allows one to conclude that the partial derivative $H_u = [H_{u,1}, \ldots, H_{u,m}]^\mathsf{T}$ of the Hamiltonian with respect to the control $u = [u_1, \ldots, u_m]^\mathsf{T}$ has to satisfy

$$H_{u,i}(\bar{x}_k^*(\tau), \bar{\lambda}_k^*(\tau), \bar{u}_k^*(\tau)) \begin{cases} > 0 & \text{if } \bar{u}_{k,i}^*(\tau) = u_i^- \\ = 0 & \text{if } \bar{u}_{k,i}^*(\tau) \in (u_i^-, u_i^+), \quad i = 1, \ldots, m, \quad \tau \in [0, T]. \\ < 0 & \text{if } \bar{u}_{k,i}^*(\tau) = u_i^+ \end{cases}$$

The adjoint dynamics in (23) possess n terminal conditions which is due to the free endpoint formulation of OCP (3). This property is taken advantage of by the gradient method, which solves the canonical BVP (22)-(23) iteratively forward and backward in time. Table 1 summarizes the algorithm of the gradient (projection) method.

The search direction $\bar{s}_k^{(j)}(\tau)$, $\tau \in [0, T]$ is the direction of improvement for the current control $\bar{u}_k^{(j)}(\tau)$. The step size $\alpha_k^{(j)}$ is computed in the subsequent line search problem (28) in order to achieve the maximum possible descent of the cost functional (3a). The function

---

1) **Initialization for $j = 0$:**

– Set convergence tolerance ε_J (e.g. $\varepsilon_J = 10^{-6}$)

– Choose initial control trajectory $\bar{u}_k^{(0)} \in \mathcal{U}_{[0,T]}$

– Integrate forward in time

$$\dot{\bar{x}}_k^{(0)}(\tau) = f(\bar{x}_k^{(0)}(\tau), \bar{u}_k^{(0)}(\tau)), \quad \bar{x}_k^{(0)}(0) = x_k \tag{25}$$

2) **Gradient step: While $j \leq N$ Do**

– Integrate backward in time

$$\dot{\bar{\lambda}}_k^{(j)}(\tau) = -H_x(\bar{x}_k^{(j)}(\tau), \bar{\lambda}_k^{(j)}(\tau), \bar{u}_k^{(j)}(\tau)), \quad \bar{\lambda}_k^{(j)}(T) = V_x(\bar{x}_k^{(j)}(T)) \tag{26}$$

– Compute the search direction

$$\bar{s}_k^{(j)}(\tau) = -H_u(\bar{x}_k^{(j)}(\tau), \bar{\lambda}_k^{(j)}(\tau), \bar{u}_k^{(j)}(\tau)), \quad \tau \in [0, T] \tag{27}$$

– Compute the step size $\alpha_k^{(j)}$ by (approximately) solving the line search problem

$$\alpha_k^{(j)} = \arg\min_{\alpha > 0} J\left(x_k, \psi(\bar{u}_k^{(j)} + \alpha \bar{s}_k^{(j)})\right) \tag{28}$$

– Compute the new control trajectory

$$\bar{u}_k^{(j+1)}(\tau) = \psi\left(\bar{u}_k^{(j)}(\tau) + \alpha_k^{(j)} \bar{s}_k^{(j)}(\tau)\right) \tag{29}$$

– Integrate forward in time

$$\dot{\bar{x}}_k^{(j+1)}(\tau) = f(\bar{x}_k^{(j+1)}(\tau), \bar{u}_k^{(j+1)}(\tau)), \quad \bar{x}_k^{(j+1)}(0) = x_k \tag{30}$$

– Quit if $|J(x_k, \bar{u}_k^{(j+1)}) - J(x_k, \bar{u}_k^{(j)})| \leq \varepsilon_J$. Otherwise set $j \leftarrow j + 1$ and return to 2).

Table 1. Gradient projection method for solving OCP (3).

$\psi = [\psi_1, \ldots, \psi_m]^\mathsf{T}$ in (28) represents a projection function of the form

$$\psi_i(u_i) = \begin{cases} u_i^- & \text{if } u_i < u_i^- \\ u_i^+ & \text{if } u_i > u_i^+ , \quad i = 1, \ldots, m \\ u_i & \text{else} \end{cases} \tag{31}$$

which guarantess the adherence of the input constraints $[u^-, u^+]$. For the real-time implementation within a suboptimal MPC scheme, the line search problem (28) can be solved in an approximate manner (see Section 4.2). Finally, the control trajectory $\bar{u}_k^{(j+1)}(\tau)$, $\tau \in [0, T]$ follows from evaluating (29) with $\bar{s}_k^{(j)}(\tau)$ and the step size $\alpha_k^{(j)}$.

The convergence properties of the gradient (projection) method are investigated, for instance, in Dunn (1996); Leese (1977); Nikol'skii (2007). In particular, Dunn (1996) proved under certain convexity and regularity assumptions that the gradient method exhibits a linear rate of convergence of the form (15).

4.2 Adaptive line search

The line search (28) represents a scalar optimization problem that is often solved approximately. The most straightforward way is to use a fixed step size α throughout all gradient iterations. This, however, usually leads to a slow rate of convergence.

An attractive alternative to a constant step size is to use a polynomial approximation with an underlying interval adaptation. To this end, the cost functional $J(x_k, \psi(\bar{u}_k^{(j)} + \alpha \bar{s}_k^{(j)}))$ in the line search problem (28) is evaluated at three sample points

$$\alpha_1 < \alpha_2 < \alpha_3 \quad \text{with} \quad \alpha_2 = (\alpha_1 + \alpha_3)/2 \tag{32}$$

that are used to construct a quadratic polynomial approximation $g(\alpha)$ of the form

$$J\left(x_k, \psi(\bar{u}_k^{(j)} + \alpha \bar{s}_k^{(j)})\right) \approx g(\alpha) := c_0 + c_1 \alpha + c_2 \alpha^2 . \tag{33}$$

The coefficients c_0, c_1, c_2 are obtained by solving the set of equations

$$J\left(x_k, \psi(\bar{u}_k^{(j)} + \alpha_i \bar{s}_k^{(j)})\right) =: J_i = g(\alpha_i), \quad i = 1, 2, 3 \tag{34}$$

with the explicit solution

$$\begin{aligned} c_0 &= \frac{\alpha_1 (\alpha_1 - \alpha_2) \alpha_2 J_3 + \alpha_2 \alpha_3 (\alpha_2 - \alpha_3) J_1 + \alpha_1 \alpha_3 (\alpha_3 - \alpha_1) J_2}{(\alpha_1 - \alpha_2)(\alpha_1 - \alpha_3)(\alpha_2 - \alpha_3)} \\ c_1 &= \frac{(\alpha_2^2 - \alpha_1^2) J_3 + (\alpha_1^2 - \alpha_3^2) J_2 + (\alpha_3^2 - \alpha_2^2) J_1}{(\alpha_1 - \alpha_2)(\alpha_1 - \alpha_3)(\alpha_2 - \alpha_3)} \\ c_2 &= \frac{(\alpha_1 - \alpha_2) J_3 + (\alpha_2 - \alpha_3) J_1 + (\alpha_3 - \alpha_1) J_2}{(\alpha_1 - \alpha_2)(\alpha_1 - \alpha_3)(\alpha_2 - \alpha_3)} . \end{aligned} \tag{35}$$

If $c_2 > 0$, then the polynomial $g(\alpha)$ has a minimum at the point

$$\hat{\alpha} = -\frac{c_1}{2c_2} . \tag{36}$$

If in addition $\hat{\alpha}$ lies inside the interval $[\alpha_1, \alpha_3]$, then $\hat{\alpha} = \alpha_k^{(j)}$ approximately solves the line search problem (28). Otherwise, $\alpha_k^{(j)}$ is set to one of the interval bounds α_1 or α_3. In this case, the interval $[\alpha_1, \alpha_3]$ can be adapted by a scaling factor to track the minimum point of the line search problem (28) over the single gradient iterations. Table 2 summarizes the overall algorithm for the approximate line search and the interval adaptation.

In general, the gradient method in Table 1 is stopped if the convergence criterion is fulfilled for some tolerance $\varepsilon_J > 0$. In practice this can lead to a large number of iterations that moreover varies from one MPC iteration to the next. In order to ensure a real-time feasible MPC implementation, the gradient algorithm is stopped after N iterations and the re-initialization of the algorithm is done as outlined in Section 3.1.

---

1) **Initialization: Default values and tolerances**
– Set polynomial tolerances ε_c, ε_g (e.g. $\varepsilon_c = 10^{-5}$, $\varepsilon_g = 10^{-6}$)
– Set initial line search interval (32) (e.g. $\alpha_1 = 10^{-2}$, $\alpha_3 = 10^{-1}$)
– Set interval adaptation factors κ^-, κ^+ (e.g. $\kappa^- = \frac{2}{3}$, $\kappa^+ = \frac{3}{2}$)
– Set interval adaptation tolerances ε_α^-, ε_α^+ (e.g. $\varepsilon_\alpha^- = 0.1$, $\varepsilon_\alpha^+ = 0.9$)
– Set interval adaptation limits $\alpha_{\min}$, $\alpha_{\max}$ (e.g. $\alpha_{\min} = 10^{-5}$, $\alpha_{\max} = 1.0$)

2) **Approximate line search**
– Compute the cost values $J_i := J(x_k, \psi(u_k^{(j)} + \alpha_i s_k^{(j)}))$ at the sample points (32)

– Compute the polynomial coefficients (35) and the candidate point (36)

– Compute the approximate step size $\alpha_k^{(j)}$ according to

$$\text{if } c_2 > \varepsilon_c : \qquad \alpha_k^{(j)} = \begin{cases} \alpha_1 & \text{if } \hat{\alpha} < \alpha_1 \\ \alpha_3 & \text{if } \hat{\alpha} > \alpha_3 \\ \hat{\alpha} & \text{else} \end{cases} \tag{37}$$

$$\text{else } (c_2 \le \varepsilon_c) : \quad \alpha_k^{(j)} = \begin{cases} \alpha_1 & \text{if } J_1 + \varepsilon_g \le \min\{J_2, J_3\} \\ \alpha_3 & \text{if } J_3 + \varepsilon_g \le \min\{J_1, J_2\} \\ \alpha_2 & \text{else} \end{cases} \tag{38}$$

– Adapt the line search interval $[\alpha_1, \alpha_3]$ for the next gradient iteration according to

$$[\alpha_1, \alpha_3] \leftarrow \begin{cases} \kappa^+[\alpha_1, \alpha_3] & \text{if } \hat{\alpha} \ge \alpha_1 + \varepsilon_\alpha^+(\alpha_3 - \alpha_1) \text{ and } \alpha_3 \le \alpha_{\max} \\ \kappa^-[\alpha_1, \alpha_3] & \text{if } \hat{\alpha} \le \alpha_1 + \varepsilon_\alpha^-(\alpha_3 - \alpha_1) \text{ and } \alpha_1 \ge \alpha_{\min} , \quad \alpha_2 \leftarrow \frac{\alpha_1 + \alpha_3}{2} \\ [\alpha_1, \alpha_3] & \text{else} \end{cases} \tag{39}$$

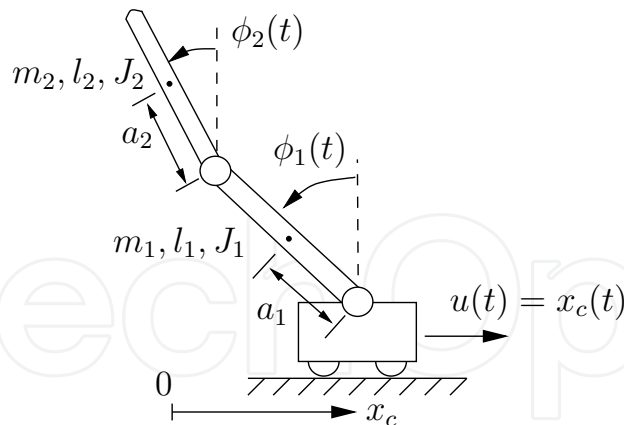Table 2. Adaptive line search for the gradient algorithm in Table 1.

Fig. 2. Inverted double pendulum on a cart.

5. Example – Inverted double pendulum

The inverted double pendulum on a cart is a benchmark problem in control theory due to its highly nonlinear and nonminimum-phase dynamics and its instability in the upward (inverted) position. The double pendulum in Figure 2 consists of two links with the lengths l_i and the angles ϕ_i, $i = 1, 2$ to the vertical direction. The displacement of the cart is given by x_c. The mechanical parameters are listed in Table 3 together with their corresponding values (Graichen et al., 2007). The double pendulum is used in this section as benchmark example for the suboptimal MPC scheme and the gradient algorithm in order to show its performance for a real-time MPC implementation.

5.1 Equations of motion and MPC formulation

Applying the Lagrangian formalism to the double pendulum leads to the equations of motion (Graichen et al., 2007)

$$M(\phi)\ddot{\phi} + c(\phi, \dot{\phi}, \ddot{x}_c) = 0 \tag{40}$$

with the generalized coordinates $\phi = [\phi_1, \phi_2]^\mathsf{T}$ and the functions

$$M(\phi) = \begin{bmatrix} J_1 + a_1^2 m_1 + l_1^2 m_2 & a_2 l_1 m_2 \cos(\phi_1 - \phi_2) \\ a_2 l_1 m_2 \cos(\phi_1 - \phi_2) & J_2 + a_2^2 m_2 \end{bmatrix} \tag{41a}$$

$$c(\phi, \dot{\phi}) = \begin{bmatrix} d_1 \dot{\phi}_1 + d_2(\dot{\phi}_1 - \dot{\phi}_2) + l_1 m_2 a_2 \sin(\phi_1 - \phi_2)\dot{\phi}_2^2 - (a_1 m_1 + l_1 m_2)\left[g\sin(\phi_1) + \cos(\phi_2)\ddot{x}_c\right] \\ d_2(\dot{\phi}_2 - \dot{\phi}_1) - a_2 m_2 \left[g\sin(\phi_2) + l_1 \sin(\phi_1 - \phi_2)\dot{\phi}_1^2 + \cos(\phi_2)\ddot{x}_c\right] \end{bmatrix}. \tag{41b}$$

The acceleration of the cart $\ddot{x}_c$ serves as control input u. Thus, the overall model of the double pendulum can be written as the second-order ordinary differential equations (ODE)

$$\begin{aligned} \ddot{x}_c &= u \\ \ddot{\phi} &= -M^{-1}(\phi)c(\phi, \dot{\phi}, u). \end{aligned} \tag{42}$$

The acceleration of the cart is limited by the constraints

$$u \in [-6, +6] \ \text{m/s}^2 \ . \tag{43}$$

Pendulum link	inner $i = 1$	outer $i = 2$
length l_i [m]	0.323	0.480
distance to center of gravity a_i [m]	0.215	0.223
mass m_i [kg]	0.853	0.510
moment of inertia J_i [N m s^2]	0.013	0.019
friction constant d_i [N m s]	0.005	0.005

Table 3. Mechanical parameters of the double pendulum in Figure 2.

With the state vector $x = [x_c, \dot{x}_c, \phi_1, \dot{\phi}_1, \phi_2, \dot{\phi}_2]^\mathsf{T}$, the second-order ODEs (42) can be written as the general nonlinear system

$$\dot{x} = f(x, u), \quad x(0) = x_0. \tag{44}$$

For the MPC formulation, a quadratic cost functional (3a)

$$J(x_k, \bar{u}) = \Delta\bar{x}^\mathsf{T}(T)P\Delta\bar{x}(T) + \int_0^T \Delta\bar{x}^\mathsf{T}(\tau)Q\Delta\bar{x}(\tau) + \Delta\bar{u}^\mathsf{T}(\tau)R\Delta\bar{u}(\tau)\, d\tau, \tag{45}$$

with $\Delta\bar{x} = \bar{x} - x_{SP}$ and $\Delta\bar{u} = \bar{u} - u_{SP}$ is used, which penalizes the distance to a desired setpoint (x_{SP}, u_{SP}), i.e. $0 = f(x_{SP}, u_{SP})$. The symmetric and positive definite weighting matrices Q, R in the integral part of (45) are chosen as

$$Q = \text{diag}(10, 0.1, 1, 0.1, 1, 0.1), \quad R = 0.001. \tag{46}$$

The CLF condition in Assumption 3 is approximately satisfied by solving the Riccati equation

$$PA + A^\mathsf{T}P - PbR^{-1}b^\mathsf{T}P + Q = 0, \tag{47}$$

where $A = \frac{\partial f}{\partial x}\big|_{x_{SP}, u_{SP}}$ and $b = \frac{\partial f}{\partial u}\big|_{x_{SP}, u_{SP}}$ describe the linearization of the system (44) around the setpoint (x_{SP}, u_{SP}). [4] The sampling time Δt and the prediction horizon T are set to

$$\Delta t = 1 \text{ ms}, \quad T = 0.3 \text{ s} \tag{48}$$

to account for the fast dynamics of the double pendulum and the highly unstable behavior in the inverted position.

5.2 Simulation results

The suboptimal MPC scheme together with the gradient method were implemented as Cmex functions under MATLAB. The functions that are required in the gradient method are

---

[4] For the linearized (stabilizable) system $\Delta\dot{x} = A\Delta x + b\Delta u$, it can be shown that the CLF inequality (9) is exactly fulfilled (in fact, (9) turns into an equality) for the terminal cost $V(x) = \Delta x^\mathsf{T}P\Delta x$ and the linear (unconstrained) feedback law $q(x) = -R^{-1}b^\mathsf{T}P\Delta x$ with P following from the Riccati equation (47).
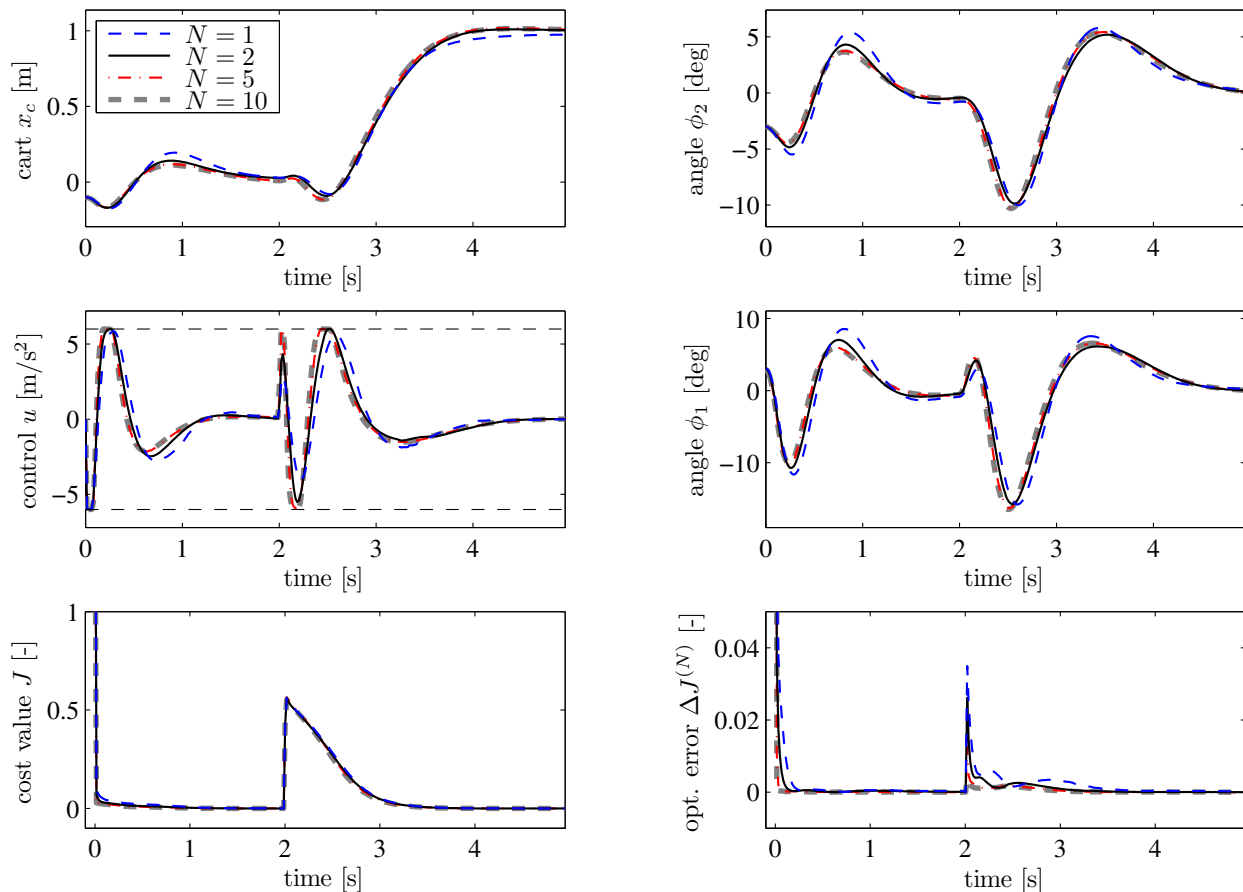
Fig. 3. MPC results for the double pendulum on a cart.

computed under the computer algebra system MATHEMATICA and are exported to MATLAB as optimized C code. The numerical integrations of the canonical equations (25)-(30) are performed by discretizing the time interval $[0, T]$ with a fixed number of 30 equidistant points and using a second order Runge-Kutta method. The nonlinear model (44), respectively (41), is used within the MPC scheme as well as for the simulation of the pendulum.

The considered simulation scenario consists of an initial error around the origin ($x_{SP} = 0$, $u_{SP} = 0$) and a subsequent setpoint step of 1 m in the cart position at time $t = 2$ s ($x_{SP} = [1 \text{ m}, 0, 0, 0, 0, 0]^\mathsf{T}$, $u_{SP} = 0$). Figure 3 shows the simulation results for a two-stage scenario (initial error and setpoint change at $t = 2$ s). Already the case of one gradient iteration per sampling step ($N = 1$) leads to a good control performance and a robust stabilization of the double pendulum. Increasing N results in a more aggressive control behavior and a better exploitation of the control constraints (43).

The lower plots in Figure 3 show the (discrete-time) profiles of the cost value $J(x_k, \bar{u}_k^{(N)})$ and of the optimization error $\Delta J^{(N)}(x_k) = J(x_k, \bar{u}_k^{(N)}) - J^*(x_k)$. In order to determine $\Delta J^{(N)}(x_k)$, the optimal cost $J^*(x_k)$ was computed in each step x_k by solving the OCP (3) for the double pendulum with a collocation-based optimization software. It is apparent from the respective plots in Figure 3 that the cost as well as the optimization error rapidly converge to zero which illustrates the exponential stability of the double pendulum in closed-loop and the incremental improvement of the algorithm. It is also seen in these plots that the performance improvement

MPC iter. N / sampling step	CPU time [ms] / sampling step	mean cost value [–]
1	0.053	0.0709
2	0.095	0.0641
3	0.133	0.0632
5	0.212	0.0610
10	0.405	0.0590

Table 4. CPU time consumption of the real-time MPC scheme for different numbers of gradient iterations N per sampling step.

between $N = 1$ and $N = 10$ iterations per sampling step are comparatively small compared to the increase of numerical load.

To investigate this point more precisely, Table 4 lists the required CPU time for different MPC settings. The computations were performed on a computer with an Intel i7 CPU (M620, 2.67 GHz) [5], 4 GB of memory, and the operating system MS Windows 7 (64 bit). The overall MPC scheme compiled as Cmex function under MATLAB 2010b (64 bit). All evaluated tests in Table 4 show that the required CPU times are well below the actual sampling time of $\Delta t = 1$ ms. The CPU times are particularly remarkable in view of the high complexity of the nonlinear pendulum model (40)-(42), which illustrates the real-time feasibility of the suboptimal MPC scheme. The last column in Table 4 shows the average cost value that is obtained by integrating the cost profiles in Figure 3 and dividing through the simulation time of 5 s. This index indicates that the MPC scheme increases in terms of control performance for larger numbers of N.

From these numbers and the simulation profiles in Figure 3, the conclusion can be drawn that $N = 2$ gradient iterations per MPC step represents a good compromise between control performance and the low computational demand of approximately 100 μs per MPC step.

6. Conclusions

Suboptimal solution strategies are efficient means to reduce the computational load for a real-time MPC implementation. The suboptimal solution from the previous MPC step is used for a warm-start of the optimization algorithm in the next run with the objective to reduce the suboptimality over the single MPC steps. Section 3 provides theoretical justifications for a suboptimal MPC scheme with a fixed number of iterations per sampling step.

A suitable optimization algorithm is the gradient method in optimal control, which allows for a time and memory efficient calculation of the single MPC iterations and makes the overall MPC scheme suitable for very fast or high dimensional dynamical systems. The control performance and computational efficiency of the gradient method is illustrated in Section 5 for a highly nonlinear and complex model of a double pendulum on a cart. The suboptimal MPC scheme based on a real-time implementation of the gradient method was

---

[5] Only one core of the i7 CPU was used for the computations.

also experimentally validated for a laboratory crane (Graichen et al., 2010) and for a helicopter with three degrees-of-freedom (Graichen et al., 2009), both experiments with sampling times of 1-2 milliseconds. The applicability of the gradient-based MPC scheme to high dimensional systems is demonstrated in (Steinböck et al., 2011) for a reheating furnace in steel industry.

7. Appendix A – Reachability of CLF region (Theorem 1)

This appendix proves the statements 1 and 2 in Theorem 1 concerning the reachability of the CLF region S_β by the MPC formulation (3) without terminal constraints. The discrete-time case was investigated in Limon et al. (2006). The following two lemmas generalize these results to continuous-time systems as considered in this chapter. Lemma 1 represents an intermediate statement that is required to derive the actual result in Lemma 2.

Lemma 1. *Suppose that Assumptions 1-3 are satisfied. If $\bar{x}_k^*(T) \notin S_\beta$ for any $x_k \in \mathbb{R}^n$, then $\bar{x}_k^*(\tau) \notin S_\beta$ for all times $\tau \in [0, T]$.*

Proof. The proof is accomplished by contradiction. Assume that $\bar{x}_k^*(T) \notin S_\beta$ and that there exists a time $\hat{\tau} \in [0, T)$ such that $\bar{x}_k^*(\hat{\tau}) \in S_\beta$. Starting at this point $\bar{x}_k^*(\hat{\tau})$, consider the residual problem

$$\hat{J}^*(\bar{x}_k^*(\hat{\tau})) = \min_{\bar{u} \in \mathcal{U}_{[T-\hat{\tau}]}} \left\{ V\left(\bar{x}(T - \hat{\tau}; \bar{x}_k^*(\hat{\tau}), \bar{u}) \right) + \int_0^{T-\hat{\tau}} l\left(\bar{x}(\tau; \bar{x}_k^*(\hat{\tau}), \bar{u}), \bar{u}(\tau) \right) d\tau \right\}$$

subject to the dynamics (3b), for which the optimal trajectories are $\hat{u}^*(\tau) = \bar{u}_k^*(\hat{\tau} + \tau)$ and $\hat{x}^*(\tau) = \bar{x}_k^*(\hat{\tau} + \tau), \tau \in [0, T - \hat{\tau}]$ by the principle of optimality. Since $\bar{x}_k^*(\hat{\tau}) \in S_\beta$ by assumption, the CLF inequality (9) with $\bar{x}^q(0) = \bar{x}_k^*(\hat{\tau})$ leads to the lower bound

$$
\begin{aligned}
V(\bar{x}_k^*(\hat{\tau})) &\geq V(\bar{x}^q(T - \hat{\tau})) + \int_0^{T-\hat{\tau}} l(\bar{x}^q(\tau), \bar{u}^q(\tau)) d\tau \\
&\geq \hat{J}^*(\bar{x}_k^*(\hat{\tau})) \\
&\geq V(\hat{x}^*(T - \hat{\tau})) = V(\bar{x}_k^*(T)) > \beta.
\end{aligned}
$$

The last line, however, implies that $\bar{x}_k^*(\hat{\tau}) \notin S_\beta$, which contradicts the previous assumption and thus proves the lemma. ☐

Lemma 2. *Suppose that Assumptions 1-3 are satisfied and consider the compact set Γ_α defined by (10). Then, for all $x_k \in \Gamma_\alpha$, the endpoint of the optimal state trajectory satisfies $\bar{x}_k^*(T) \in S_\beta$. Moreover, $S_\beta \subseteq \Gamma_\alpha$.*

Proof. We will again prove the lemma by contradiction. Assume that there exists a $x_k \in \Gamma_\alpha$ such that $\bar{x}_k^*(T) \notin S_\beta$, i.e. $V(\bar{x}_k^*(T)) > \beta$. Then, Lemma 1 states that $\bar{x}_k^*(\tau) \notin S_\beta$ for all $\tau \in [0, T]$, or using (5),

$$||\bar{x}_k^*(\tau)||^2 > \frac{\beta}{M_V} \quad \forall \tau \in [0, T]. \tag{49}$$

This allows one to derive a lower bound on the optimal cost

$$
\begin{aligned}
J^*(x_k) &= V(\bar{x}_k^*(T)) + \int_0^T l(\bar{x}_k^*(\tau), \bar{u}_k^*(t)) \, d\tau \\
&\geq \beta + \int_0^T m_l \frac{\beta}{M_V} d\tau \\
&= \beta \left(1 + \frac{m_l}{M_V} T \right) \\
&= \alpha.
\end{aligned}
\tag{50}
$$

From this last line it can be concluded that $x_k \notin \Gamma_\alpha$ for all $\bar{x}_k^*(T) \notin S_\beta$. This, however, is a contradiction to the previous assumption and implies that $\bar{x}_k^*(T) \in S_\beta$ for all $x_k \in \Gamma_\alpha$. To prove that Γ_α contains the CLF region S_β, consider $x_k \in S_\beta$ and the bound on the optimal cost

$$
J^*(x_k) \leq V(\bar{x}^q(T)) + \int_0^T l(\bar{x}^q(\tau), \bar{u}^q(\tau)) \, dt
\tag{51}
$$

with the CLF trajectories $\bar{x}^q(\tau)$, $\bar{x}^q(0) = x_k$, and $\bar{u}^q(\tau) = q(\bar{x}^q(\tau))$. Similar to the proof of Lemma 1, the CLF inequality (9) implies that

$$
V(\bar{x}^q(T)) \leq V(\bar{x}^q(0)) - \int_0^T l(\bar{x}^q(\tau), \bar{u}^q(\tau)) d\tau.
\tag{52}
$$

Hence, (51)-(52) and definition (10) show that $J^*(x_k) \leq V(x_k) \leq \beta < \alpha$ for all $x_k \in S_\beta$, which proves that Γ_α contains S_β. □

8. Appendix B – Verification of Assumption 6 for linear-quadratic OCPs

The following lines show that Assumption 6 is fulfilled for OCPs of the form

$$
\min_{u \in \mathcal{U}_{[0,T]}} \quad J(u) = x^\mathsf{T}(T) P x(T) + \int_0^T x^\mathsf{T}(t) Q x(t) + u^\mathsf{T}(t) R u(t) \, dt,
\tag{53}
$$

$$
\text{subj. to} \qquad \dot{x} = Ax + Bu \quad x(0) = x_0, \quad x \in \mathbb{R}^n, \quad u \in \mathbb{R}^m
\tag{54}
$$

with the quadratic cost functional (53), the linear dynamics (54) and some initial state $x_0 \in \mathbb{R}^n$. The admissible input set $\mathcal{U}_{[0,T]}$ is assumed to be convex and the weighting matrices P, Q, R are symmetric and positiv definite. A useful property of the linear-quadratic problem is the strong convexity property (Allaire, 2007)

$$
C||u - v||_{L_2^m[0,T]}^2 \leq J(u) + J(v) - 2J(\tfrac{1}{2}u + \tfrac{1}{2}v)
\tag{55}
$$

for some constant $C > 0$ and all control functions $u, v \in \mathcal{U}_{[0,T]}$. To show this, first consider the control term of the cost functional (53) and the right-hand side of (55), which can be written in the form

$$
\int_0^T u^\mathsf{T} R u + v^\mathsf{T} R v - \frac{1}{2}(u + v)^\mathsf{T} R (u + v) \, dt = \frac{1}{2} \int_0^T (u - v)^\mathsf{T} R (u - v) \, dt.
$$

The same simplifications can be used for the state-dependent terms in (53) since the linear dynamics (54) ensures that the superposition of two input signals $w(t) = \frac{1}{2}u(t) + \frac{1}{2}v(t)$ yield a corresponding superposed state response $x_w(t) = \frac{1}{2}x_u(t) + \frac{1}{2}x_v(t)$ with $x_w(0) = x_0$. Hence, the right-hand side of (55) can be written as

$$
\begin{aligned}
J(u) + J(v) - 2J(\tfrac{1}{2}u + \tfrac{1}{2}v) &= \frac{1}{2}\Delta x^\mathsf{T}(T)P\Delta x(T) + \frac{1}{2}\int_0^\mathsf{T} \Delta x^\mathsf{T}(t)Q\Delta x(t)\,\mathrm{d}t \\
&\quad + \frac{1}{2}\int_0^\mathsf{T}\Big(u(t) - v(t)\Big)^\mathsf{T} R\Big(u(t) - v(t)\Big)\,\mathrm{d}t \\
&\geq C||u - v||^2_{L_2^m[0,T]}
\end{aligned}
$$

with $\Delta x(t) = x_u(t) - x_v(t)$ and the constant $C = \lambda_{\min}(R)/2$. Since $J(u)$ is strongly (and therefore also strictly) convex on the convex set $\mathcal{U}_{[0,T]}$, it follows from standard arguments (Allaire, 2007) that there exists a global and unique minimum point $u^* \in \mathcal{U}_{[0,T]}$. Moreover, since $\mathcal{U}_{[0,T]}$ is convex, $\frac{1}{2}(u + u^*) \in \mathcal{U}_{[0,T]}$ for all $u \in \mathcal{U}_{[0,T]}$ such that $J(\frac{1}{2}u + \frac{1}{2}u^*) \geq J(u^*)$. Hence, the strong convexity inequality (55) can be turned into the quadratic growth property

$$
C||u - u^*||^2_{L_2^m[0,T]} \leq J(u) + J(u^*) - 2J(\tfrac{1}{2}u + \tfrac{1}{2}u^*) \leq J(u) - J(u^*) \quad \forall\, u \in \mathcal{U}_{[0,T]}.
$$

This shows that Assumption 6 is indeed satisfied for linear-quadratic OCPs of the form (53).

9. Acknowledgements

10. References

Allaire, G. (2007). *Numerical Analysis and Optimization*, Oxford University Press, New York.

Berkovitz, L. (1974). *Optimal Control Theory*, Springer, New York.

Cannon, M. & Kouvaritakis, B. (2002). Efficient constrained model predictive control with asymptotic optimality, *SIAM Journal on Control and Optimization* 41(1): 60–82.

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

DeHaan, D. & 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., Ferreau, H. & Haverbeke, N. (2009). Efficient numerical methods for nonlinear MPC and moving horizon estimation, *in* L. Magni, D. Raimondo & F. Allgöwer (eds), *Nonlinear Model Predictive Control – Towards New Challenging Applications*, pp. 391–417.

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

Dunn, J. (1996). On l^2 conditions amd the gradient projection method for optimal control problems, *SIAM Journal on Control and Optimization* 34(4): 1270–1290.

Findeisen, R. (2006). *Nonlinear Model Predictive Control: A Sampled-Data Feedback Perspective*, Vol. 1087, Fortschritt-Berichte VDI Reihe 8.

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

Fontes, F., Magni, L. & Gyurkovics, E. (2007). Sampled-data model predictive control for nonlinear time-varying systems: Stability and robustness, *in* R. Findeisen, F. Allgöwer & L. Biegler (eds), *Assessment and Future Directions of Nonlinear Model Predictive Control*, LNCIS 358, Springer, Berlin, pp. 115–129.

Graichen, K., Egretzberger, M. & Kugi, A. (2010). Suboptimal model predictive control of a laboratory crane, *8th IFAC Symposium on Nonlinear Control Systems (NOLCOS 2010)*, Budapest (Hungary).

Graichen, K., Kiefer, T. & Kugi, A. (2009). Real-time trajectory optimization under input constraints for a flatness-controlled laboratory helicopter, *European Control Conference (ECC) 2009*, Budapest (Hungary), pp. 2061–2066.

Graichen, K. & Kugi, A. (2010). Stability and incremental improvement of suboptimal MPC without terminal constraints, *IEEE Transactions on Automatic Control* 55(11): 2576–2580.

Graichen, K., Treuer, M. & Zeitz, M. (2007). Swing-up of the double pendulum on a cart by feedforward and feedback control with experimental validation, *Automatica* 43(1): 63–71.

Grüne, L. & Pannek, J. (2011). *Nonlinear Model Predictive Control: Theory and Algorithms*, Springer, London.

Hsu, J. & Meyer, A. (1968). *Modern Control Principles and Applications*, McGraw-Hill, New York.

Ito, K. & Kunisch, K. (2002). Receding horizon optimal control for infinite dimensional systems, *ESAIM: Control, Optimisation and Calculus of Variations* 8: 741–760.

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

Kothare, S. D. O. & Morari, M. (2000). Contractive model predictive control for constrained nonlinear systems, *IEEE Transactions on Automatic Control* 45(6): 1053–1071.

Lee, E. & Markus, L. (1967). *Foundations of Optimal Control*, Wiley, New York.

Lee, Y., Kouvaritakis, B. & Cannon, M. (2002). Constrained receding horizon predictive control for nonlinear systems, *Automatica* 38(12): 2093–2102.

Leese, S. (1977). Convergence of gradient methods for optimal control problems, *Journal of Optimization Theory and Applications* 21(3): 329–337.

Limon, D., Alamo, T., Salas, F. & Camacho, E. (2006). On the stability of constrained MPC without terminal constraint, *IEEE Transactions on Automatic Control* 51(5): 832–836.

Mayne, D., Rawlings, J., Rao, C. & Scokaert, P. (2000). Constrained model predictive control: stability and optimality, *Automatica* 36(6): 789–814.

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

Nikol'skii, M. (2007). Convergence of the gradient projection method in optimal control problems, *Computational Mathematics and Modeling* 18(2): 148–156.

Ohtsuka, T. (2004). A continuation/GMRES method for fast computation of nonlinear receding horizon control, *Automatica* 40(4): 563–574.

Parisini, T. & Zoppoli, R. (1995). A receding-horizon regulator for nonlinear systems and a neural approximation, *Automatica* 31(10): 1443–1451.

Primbs, J. (1999). *Nonlinear Optimal Control: A Receding Horizon Approach*, PhD thesis, California Institute of Technology, Pasadena, CA.

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

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

Steinböck, A., Graichen, K. & Kugi, A. (2011). Dynamic optimization of a slab reheating furnace with consistent approximation of control variables, *IEEE Transactions on Control Systems Technology* 16(6): 1444–1456.

Frontiers of Model Predictive Control

Edited by Prof. Tao Zheng

Model Predictive Control (MPC) usually refers to a class of control algorithms in which a dynamic process model is used to predict and optimize process performance, but it is can also be seen as a term denoting a natural control strategy that matches the human thought form most closely. Half a century after its birth, it has been widely accepted in many engineering fields and has brought much benefit to us. The purpose of the book is to show the recent advancements of MPC to the readers, both in theory and in engineering. The idea was to offer guidance to researchers and engineers who are interested in the frontiers of MPC. The examples provided in the first part of this exciting collection will help you comprehend some typical boundaries in theoretical research of MPC. In the second part of the book, some excellent applications of MPC in modern engineering field are presented. With the rapid development of modeling and computational technology, we believe that MPC will remain as energetic in the future.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

INTECH
open science | open minds