ETH zürich

Institute for Dynamic Systems and Control

Alexander Casper Erdin

A Comparison on Robust MPC Methods for Nonlinear Systems

Semester Project

Institute for Dynamic Systems and Control
ETH Zurich

Supervision

Prof. Dr. Melanie Zeilinger
Dr. Johannes Köhler
Antoine Leeman

September 3, 2024

IDSC-MZ-JK-06

IDSC

Abstract

We present a unifying framework that enables automatized comparison of nonlinear robust model predicitve control (RMPC) approaches for nonlinear control- and disturbance-affine systems. We investigate the use of different RMPC approaches to understand how they compare, when to use which method, and why certain methods perform well or not. To achieve this, two opposing RMPC methods were chosen and implemented to represent the spectrum of available RMPC approaches. Namely, we compare system level synthesis-based RMPC (SLS), which operates in discrete time and propagates uncertainties with a linearized error dynamics and control contraction metrics-based RMPC (CCM), which operates in continuous time and uses control contraction metrics to parametrize a disturbance reachable set. Additionally, we introduce an intermediate method prestabilized SLS (SLSK) that aims to reduce conservatism of SLS by introducing a feedback K, which prestabilizes our continuous time dynamics.

Our results, based on two dynamic systems, demonstrate that CCM tends to be more conservative compared to SLS. However, SLS is constrained by its sampling time h and the computational resources required for solving its large joint optimization problem. Our newly introduced approach SLSK showed no performance improvements when being compared to SLS, hinting at the fact that the feedback matrix K that prestabilizes our nonlinear dynamics not necessarily reduces conservatism. In addition, we found the edge case of non-exponentially stabilizing dynamics for CCM which leads to unbounded DRSs, increasing conservatism drastically.

Keywords: Robust MPC, Nonlinear MPC.

Contents

Nomenclature

Acronyms

CCM	Control contraction metrics
DRS	Disturbance reachable set
LTV	Linear time-varying
MPC	Model predicitve control
RK4	Runge-Kutta 4
RKDP	Runge-Kutta Dormand–Prince
RMPC	Robust MPC
SLS	System level synthesis
SLSK	Prestabilized SLS

Symbols

h	Sample time
N	Horizon length
w_{max}	Maximum disturbance

Chapter 1

Introduction

With the growing computational power of micro controllers, model predicitve control (MPC) has become increasingly relevant over the last decades [1]. MPC is an optimal control method known for its ability to satisfy constraints while minimizing a given objective [2]. However, in real-world applications it is essential to also account for model uncertainties and disturbances while still ensuring constraint fulfillment. To address this challenge, robust MPC (RMPC) [3] has been developed with the main idea of assuming a worst case disturbance to provide robust constraint satisfaction. Commonly this is done by keeping track of a nominal trajectory and the computation of a tube quantifying a disturbance reachable set (DRS) in which the error (i.e. the difference of true and nominal state) is contained for all possible disturbance realisations. While this approach is applicable in linear and nonlinear domains, this work is only concerned with nonlinear RMPC.

1.1 Related Work

The design of robust MPC schemes for nonlinear systems is non-trivial and as a result many different RMPC formulations have been developed to approach this problem. For example [4] proposes a method for discrete nonlinear systems propagating uncertainties using Lipschitz constants of the nominal system. This can be implemented easily, but quickly becomes very conservative for long horizons. On the other hand [5] utilizes interval arithmetic for the online computation of DRSs for discrete nonlinear systems. The computational effort for this method is only slightly greater than nominal MPC but suffers from conservatism and the fact that interval-sets are generally not invariant. In [6] min–max differential inequalities are employed for continuous time nonlinear systems together with robust, forward invariant ellipsoidal tubes parametrizing a disturbance reachable set. This approach offers more degrees of freedom and hence a less conservative DRS, but it comes at the cost of increased computational complexity. The authors from [7] make use of boundary layer sliding control to express the tube geometry as a polytope which is jointly optimized with the MPC problem. Recently in [8, 9, 10], control contraction metrics (CCM) are utilized to incrementally stabilize continuous time nonlinear systems. The increased offline design effort to construct these metrics allows an efficient online implementation. [11, 12] employ system level synthesis (SLS) to propagate uncertainties using a linearized error dynamics in discrete time. The resulting joint optimization leads to reduced conservatism at the cost of increased computational effort. All of these methods eventually boil down to optimizing a nominal trajectory using different approaches to parametrize the DRS.

Table 1.1 categorizes the different approaches to RMPC. It distinguishes between offline and online computed DRSs, further dividing into offline and online computed error feedback.

Table 1.1: Different nonlinear RMPC approaches compared

Feedback \ DRS	Offline	Online
Offline	[4, 7, 8, 9, 10]	[5, 13]
Online	-	[6, 11, 12, 14]

1.2 Motivation

While the designs and trade-offs for linear RMPC are relatively well understood, there are still very few comparisons for nonlinear RMPC, despite the variety of methods available. This work aims to provide a basis for understanding of how these different approaches compare, when to use which method, and why certain methods perform well or not. To achieve this, we have chosen two representative methods for comparison in the context of nonlinear RMPC. Specifically, we compare RMPC using SLS as described in [11, 12], which operates in discrete time and propagates uncertainties with a linearized error dynamics, against RMPC using CCM as outlined in [8, 9, 10], which operates in continuous time and uses CCM to parametrize a disturbance reachable set. These two methods were chosen because they both cover extreme cases: we compare a completely offline constructed DRS against a fully online optimized DRS.

A fair comparison of those two methods is not obvious as they rely on different underlying assumptions and operate in different time domains.

1.3 Contribution

The main contribution of this work is a general framework that automatically implements RMPC approaches for any dynamic system which is affine in the control input and it's disturbance. This is accomplished through sub-modules that independently define parameters, dynamics, and controllers (see Chapter 3). As a result, streamlined comparisons on different dynamics can be made without the need for manually tailored controllers. Given this framework, different comparisons have been performed assessing conservatism, intrinsic limitations and ease of implementation by varying different parameters such as the disturbance, MPC horizon length, level of nonlinearity, sample time and the system dynamics.

A second contribution of this work is an intermediate method between SLS and CCM that utilizes a prestabilizing feedback for the continuous time system, aiming to stabilize the Hessian, resulting in a smaller Hessian over-bound. This bridging method is also compared against classical SLS and CCM to allow for a comprehensive understanding.

Chapter 2

Preliminaries

This chapter describes the methods compared and is used to lay out the problem setup and general notation.

2.1 Notation

Matrices / matrix functions are denoted using capital letters, e.g. $A(x)$, whereas vectors / vector functions are denoted using lowercase letters, such as $x(t)$. We distinguish between continuous and discrete time vectors using italic letters, such as $x(t)$, and upright letters, such as $\mathrm{x}[k]$. Discrete time vectors are also abbreviated with x_k. In addition we write stacked vectors compactly using bold symbols like $\boldsymbol{x}$. The set of integers in the interval $[0, m]$ is denoted as $\mathbb{N}_m := \{0, \ldots, m\}$. We use $\langle A \rangle = A + A^\top$ for square matrices $A \in \mathbb{R}^{n,n}$. We abbreviate two stacked vectors using $(a, b) = [a^\top, b^\top]^\top$. The real numbers are denoted by $\mathbb{R}$ and $0_{p,q} \in \mathbb{R}^{p,q}$, $0_m \in \mathbb{R}^m$ denote the zero matrix or the zero vector respectively. Analogously we define the identity matrix $\mathcal{I}_{p,q} \in \mathbb{R}^{p,q}$ and the square identity matrix with $\mathcal{I}_m \in \mathbb{R}^{m,m}$. Let $\mathcal{B}_\infty^m := \{d \in \mathbb{R}^m \mid \|d\|_\infty \leqslant 1\}$ be the unit-norm ball of the ∞-norm and the Minkowski sum is defined as $\mathcal{S}_1 \oplus \mathcal{S}_2 := \{s_1 + s_2 \mid s_1 \in \mathcal{S}_1, s_2 \in \mathcal{S}_2\}$.

2.2 Problem Setup

In this work we consider nonlinear systems that are affine in the control input $u(t) \in \mathbb{R}^{n_u}$ and the disturbance $w(t) \in \mathbb{R}^{n_w}$

$$\dot{x} = f_w(x, u, w) \tag{2.1}$$
$$= f(x) + B(x)u + E(x)w \tag{2.2}$$

where $x(t) \in \mathbb{R}^{n_x}$ is the state vector, and $f(x)$, $B(x)$ and $E(x)$ are vector / matrix functions describing our system dynamics.

Furthermore, we assume compact polytopic constraint sets

$$\mathcal{C} := \{(x, u) \mid a_i^\top (x, u) \leqslant b_i, \forall\, i \in \mathbb{N}_{n_\mathcal{C}}\} \tag{2.3}$$

and bounded disturbances

$$\mathcal{W} := \{w \mid \|w\|_\infty \leqslant w_{max}\}. \tag{2.4}$$

The goal of the two RMPC methods compared can be summarized in the following conceptual robust nonlinear optimal control problem

$$\min_{\pi(\cdot)} \quad \max_{w \in \mathcal{W}} \; J_\infty(x(0), \pi(\cdot), w(\cdot)) \tag{2.5}$$

$$\text{s.t.} \quad \dot{x} = f_w(x, u, w), \quad \forall\, t \geqslant 0, \tag{2.6}$$

$$u = \pi(x(\cdot), t), \tag{2.7}$$

$$(x, u) \in \mathcal{C} \tag{2.8}$$

$$w \in \mathcal{W}. \tag{2.9}$$

with the goal to find a causal policy π minimizing the cost J_∞ given by the stage cost and terminal cost defined by

$$l(z, v) = \|z - x_{ref}\|_Q^2 + \|v - u_{ref}\|_R^2 \tag{2.10}$$

$$l_T(z) \;\; = \|z - x_{ref}\|_P^2 \tag{2.11}$$

and the positive definite matrices $Q \in \mathbb{R}^{n_x, n_x}$, $R \in \mathbb{R}^{n_u, n_u}$ and $P \in \mathbb{R}^{n_x, n_x}$.

2.2.1 Discretization

To discretize our continuous time systems, we utilize one of three methods: Euler discretization, which preserves affinity, or the more accurate methods Runge-Kutta 4 (RK4) and Runge-Kutta Dormand–Prince (RKDP), which do not preserve affinity. RK4 is a four-stage method which is primarily known because of it's ease of implementation while still offering good accuracy, whereas RKDP is a seven-stage method, designed to minimize the error of the fifth-order solution, hence being more accurate than RK4.

2.3 RMPC using System Level Synthesis

This section explains RMPC using SLS according to [11, 12]. As for many RMPC methods SLS divides the optimal control problem into optimizing a nominal trajectory $z[k] \in \mathbb{R}^{n_x}$ and keeping track of a DRS, ensuring that the true state and input is guaranteed to lie in the polytopic constraint set $\mathcal{C}$. What distinguishes SLS from other methods is that it does not rely on the offline design of a stabilizing feedback but directly and jointly optimizes the nominal trajectory, the affine error feedback and the dynamic linearization error over-bound to achieve robust constraint satisfaction.

SLS is a discrete time method for which we consider a discrete dynamics of the following form which is affine in the disturbance

$$\mathrm{x}_{k+1} = f_d(\mathrm{x}_k, \mathrm{u}_k) + E_d\, \mathrm{w}. \tag{2.12}$$

We can obtain a discrete dynamics of this form by discretizing our continuous time dynamics defined in Equation (2.1) using the affinity preserving Euler discretization such that we get

$$\mathrm{x}_{k+1} = \underbrace{\mathrm{x}_k + h(f(\mathrm{x}_k) + B(\mathrm{x}_k)\mathrm{u}_k)}_{f_d(\mathrm{x}_k, \mathrm{u}_k)} + E_d\, \mathrm{w}_k \tag{2.13}$$

with $E_d = \max_{\mathrm{x} \in \mathcal{C}} h \cdot E(\mathrm{x})$. We discuss how to extend this to non-affine discretization methods in Section 3.1.1.

2.3.1 Dynamic Linearization Error Over-Bound

The dynamic linearization error over-bound is needed in order to rewrite the error dynamics as a linear time-varying (LTV) system eventually enabling us to utilize SLS techniques. In order to derive the dynamic linearization error over-bound we first define a nominal trajectory

$$\mathrm{z}_{k+1} = f_d(\mathrm{z}_k, \mathrm{v}_k) \quad \forall\, k \in \mathbb{N}_T, \quad \mathrm{z}_0 = \mathrm{x}_0 \tag{2.14}$$

together with an affine, time-varying error-feedback

$$\pi_k(\mathrm{x}_{0:k}) := \mathrm{v}_k + \sum_{j=0}^{k-1} K^{k-1,j} \Delta \mathrm{x}_{k-j}, \tag{2.15}$$

where $\Delta \mathrm{x}_k = \mathrm{x}_k - \mathrm{z}_k$ denotes the error between the true state and the nominal trajectory, and $\Delta \mathrm{u}_k = \mathrm{u}_k - \mathrm{z}_k$ similarly defines the input error.

Given those definitions we start by deriving the linearization

$$A(\mathrm{z}, \mathrm{v}) := \left.\frac{\partial f_d}{\partial \mathrm{x}}\right|_{\substack{\mathrm{x}=\mathrm{z} \\ u=\mathrm{v}}}, \quad B(\mathrm{z}, \mathrm{v}) := \left.\frac{\partial f_d}{\partial \mathrm{u}}\right|_{\substack{\mathrm{x}=\mathrm{z} \\ \mathrm{u}=\mathrm{v}}} \tag{2.16}$$

such that we can rewrite our dynamics as a first order Taylor expansion around the nominal trajectory plus some linearization error

$$\begin{aligned} \mathrm{x}_{k+1} &= f_d(\mathrm{x}, \mathrm{u}) + E_d \mathrm{w} && (2.17) \\ &= f_d(\mathrm{z}, \mathrm{v}) + A(\mathrm{z}, \mathrm{v})(\mathrm{x} - \mathrm{z}) + B(\mathrm{z}, \mathrm{v})(\mathrm{u} - \mathrm{v}) + r(\mathrm{x}, \mathrm{u}, \mathrm{z}, \mathrm{v}) + E_d \mathrm{w}. && (2.18) \end{aligned}$$

The linearization error $r(\mathrm{x}, \mathrm{u}, \mathrm{z}, \mathrm{v})$ is described exactly by the Lagrange remainder as explained in the appendix A.1 and can be written as

$$r(\mathrm{x}, \mathrm{u}, \mathrm{z}, \mathrm{v}) := \left(\mathrm{e}^\top H_1 \mathrm{e}, \ldots, \mathrm{e}^\top H_{n_x} \mathrm{e}\right) \tag{2.19}$$

with $\mathrm{e} := (\mathrm{x} - \mathrm{z}, \mathrm{u} - \mathrm{z}) \in \mathcal{C}$ being the error of the true and nominal state and input and the symmetric Hessian

$$H_i(\xi) = \left.\begin{bmatrix} \frac{\partial^2 f_{d,i}}{\partial \mathrm{x}^2} & \frac{\partial^2 f_{d,i}}{\partial \mathrm{x} \partial \mathrm{u}} \\ * & \frac{\partial^2 f_{d,i}}{\partial \mathrm{u}^2} \end{bmatrix}\right|_{(\mathrm{x},\mathrm{u})=\xi} \tag{2.20}$$

depending on $f_{d,i}$, the i^{th} component of f_d. The Hessian H_i is evaluated at point $\xi \in \mathbb{R}^{n_x+n_u}$ which is some unknown convex combination of the nominal trajectory (z, v) and the true state and input (x, u) hence lying in the compact polytopic constraint set $\xi \in \mathcal{C}$.

From this one can compute the Lagrange error bound with the worst-case curvature μ defined as

$$\mu := \operatorname{diag}(\mu_1, \ldots, \mu_{n_x}), \quad \mu_i := \frac{1}{2} \max_{\substack{\epsilon \in \mathcal{B}_\infty^{n_x+n_u} \\ \xi \in \mathcal{C}}} \left|\epsilon^\top H_i(\xi) \epsilon\right|. \tag{2.21}$$

such that the Lagrange remainder will always satisfy

$$|r_i(\mathrm{x}, \mathrm{u}, \mathrm{z}, \mathrm{v})| \leqslant \|\mathrm{e}\|_\infty^2 \mu_i \tag{2.22}$$

for any $(\mathrm{x}, \mathrm{u}) \in \mathcal{C}$ and $(\mathrm{z}, \mathrm{v}) \in \mathcal{C}$. The computation of μ is achieved offline and is in fact the only offline computed quantity for SLS.

2.3.2 LTV Error Dynamics

Given the linearization error over-bound we can now derive the LTV error dynamics from the nonlinear error dynamics given by

$$\Delta \mathrm{x}_{k+1} = \mathrm{x}_{k+1} - \mathrm{z}_{k+1} \tag{2.23}$$

by plugging in Equation (2.17) and Equation (2.14) to obtain

$$\Delta \mathrm{x}_{k+1} = A(\mathrm{z}_k, \mathrm{v}_k) \Delta \mathrm{x}_k + B(\mathrm{z}_k, \mathrm{v}_k) \Delta \mathrm{u}_k + \underbrace{r(\mathrm{x}_k, \mathrm{u}_k, \mathrm{z}_k, \mathrm{v}_k) + E_d \mathrm{w}_k}_{=:\mathrm{d}_k}. \tag{2.24}$$

with the lumped disturbance

$$\mathrm{d}_k \in \|\mathrm{e}_k\|_\infty^2 \, \mu \, \mathcal{B}_\infty^{n_x} \oplus E_d \mathcal{W}. \tag{2.25}$$

This formulation is still hard to implement because of the dependence on the realized state $\Delta\mathrm{x}$ and input $\Delta\mathrm{u}$. We mitigate this issue by employing an additional decision variable, namely a filtered disturbance set. Therefore, we define the unit noise $\tilde{\mathbf{w}} \in \mathcal{B}_\infty^{n_x}$ and the filter

$$\mathbf{\Sigma} := \mathrm{blkdiag}(\Sigma_0, \ldots, \Sigma_T), \quad \Sigma_k := \mathrm{diag}(\sigma_{k,1}, \ldots, \sigma_{k,n_x}) \in \mathbb{R}^{n_x \times n_x}. \tag{2.26}$$

with $\sigma_{k,i}$ being the i^{th} diagonal element of the filter at time step k. This filter is constrained such that $\mathbf{\Sigma}\tilde{\mathbf{w}}$ over-bounds the lumped disturbance $\mathbf{d}$.

2.3.3 Parametrization of LTV Dynamics through SLS

The last step involves parameterizing our LTV error dynamics such that it can be used in a nonlinear program. Using established SLS procedures as explained in [15, 16] we can compactly rewrite our closed-loop system using the stacked error vectors $\Delta\mathbf{x}$ and $\Delta\mathbf{u}$ such that

$$\begin{bmatrix}\Delta\mathbf{x}\\ \Delta\mathbf{u}\end{bmatrix} = \begin{bmatrix}(\mathcal{I} - \mathbf{ZA} - \mathbf{ZBK})^{-1}\\ \mathbf{K}(\mathcal{I} - \mathbf{ZA} - \mathbf{ZBK})^{-1}\end{bmatrix} \mathbf{\Sigma}\tilde{\mathbf{w}} = \begin{bmatrix}\mathbf{\Phi}_\mathrm{x}\\ \mathbf{\Phi}_\mathrm{u}\end{bmatrix} \tilde{\mathbf{w}} \tag{2.27}$$

with $\mathbf{\Phi}_\mathrm{x} \in \mathcal{L}^{T, n_x \times n_x}$ and $\mathbf{\Phi}_\mathrm{u} \in \mathcal{L}^{T, n_u \times n_x}$ being the parametrization used in the MPC formulation. The constraint

$$\begin{bmatrix}\mathcal{I} - \mathbf{ZA}, & -\mathbf{ZB}\end{bmatrix} \begin{bmatrix}\mathbf{\Phi}_\mathrm{x}\\ \mathbf{\Phi}_\mathrm{u}\end{bmatrix} = \mathbf{\Sigma} \tag{2.28}$$

guarantees that the reachable set of the nonlinear system in closed-loop with the controller $\mathbf{K} = \mathbf{\Phi}_\mathrm{u}\mathbf{\Phi}_\mathrm{x}^{-1}$ writes according to (2.27).

The complete MPC formulation is as follows

$$\min_{\mathbf{\Phi}_\mathrm{x}, \mathbf{\Phi}_\mathrm{u}, \mathbf{\Sigma}, \mathbf{z}, \mathbf{v}_0, \mathbf{v}, \tau} \; J_T(\mathrm{x}_0, \mathbf{z}, \mathbf{v}, \mathbf{\Phi}), \tag{2.29}$$

$$\text{s.t.} \quad \begin{bmatrix}\mathcal{I} - \mathbf{ZA}(\mathbf{z}, \mathbf{v}), & -\mathbf{ZB}(\mathbf{z}, \mathbf{v})\end{bmatrix} \begin{bmatrix}\mathbf{\Phi}_\mathrm{x}\\ \mathbf{\Phi}_\mathrm{u}\end{bmatrix} = \mathbf{\Sigma}, \tag{2.30}$$

$$\mathrm{z}_0 = \mathrm{x}_0, \tag{2.31}$$

$$\mathrm{z}_{k+1} = f_d(\mathrm{z}_k, \mathrm{v}_k), \qquad \forall\, k \in \mathbb{N}_T \tag{2.32}$$

$$\left\|[E_{d,i} w_{max}, \tau_k^2 \mu_i]\right\|_1 \leqslant \sigma_{k,i}, \qquad \forall\, k \in \mathbb{N}_T,\ \forall\, i \in \mathbb{N}_{n_x} \tag{2.33}$$

$$\left\|a_i^\top \mathbf{\Phi}^{(k-1)}\right\|_1 + a_i^\top(\mathrm{z}_k, \mathrm{v}_k) \leqslant b_i, \qquad \forall\, k \in \mathbb{N}_T,\ \forall\, i \in \mathbb{N}_{n_c} \tag{2.34}$$

$$\left\|\mathbf{\Phi}^{(k-1)}\right\|_\infty \leqslant \tau_k \qquad \forall\, k \in \mathbb{N}_T. \tag{2.35}$$

where $\mathbf{\Phi}^{(k)} = \left(\mathbf{\Phi}_\mathrm{x}^{(k)}, \mathbf{\Phi}_\mathrm{u}^{(k)}\right)$.

This MPC formulation provides robust constraint satisfaction for the discretized nonlinear system in closed-loop through the disturbance feedback controller. Refer to [11, 12] for a rigorous derivation with more details and references.

2.4 RMPC using Control Contraction Metrics

This section discusses RMPC using CCM as proposed by [8]. Again the problem is divided into optimizing a nominal trajectory and a tube parametrizing the DRS. In contrast to the method presented in the previous section, however, this approach works in continuous time using CCMs that parameterize a homothetic tube to dynamically propagate the uncertainties.

We consider the problem setup as described in Section 2.2 together with its differential dynamics given by

$$\dot{\delta}_x = A(x,u,w)\delta_x + B(x)\delta_u + E(x)\delta_w \tag{2.36}$$

with $A(x,u,w) = \frac{\partial f_w}{\partial x}\Big|_{(x,u,w)}$.

2.4.1 Control Contraction Metrics

A stability notion commonly used to prove robust stability for uncertain nonlinear systems is incremental input to state stability. As such we want to show that our system fulfils the following condition

$$\|x(t) - z(t)\| \leqslant \alpha_w(\|w_{max}\|) + \beta(\|x_0 - z_0\|, t), \quad \forall\, t \geqslant 0,\ \alpha_w \in \mathcal{K},\ \beta \in \mathcal{KL} \tag{2.37}$$

which ensures that the true uncertain system remains close to the nominal system when the disturbances are bounded. We can construct a homothetic tube meeting this condition using control contraction metrics [17]. This is achieved by designing a CCM $M(x)$ and a differential feedback $K(x)$ such that they satisfy the following condition $\forall\,(x,u) \in \mathcal{C},\ w \in \mathcal{W}$

$$\dot{M}(x) + \langle M(x) A_{cl}(x,u,w) \rangle \preceq -2\rho M(x) \tag{2.38}$$

$$\underline{M} \preceq M(x) \preceq \overline{M} \tag{2.39}$$

with $\dot{x} = f_w(x,u,w)$ and $A_{cl} = A(x,u,w) + B(x)K(x)$ denoting the stabilized differential dynamics. The matrices $\underline{M}$, $\overline{M} \succ 0$ are bounds on $M(x)$, whereas ρ denotes the contraction rate of the CCM (which is ideally larger than zero, guaranteeing that the DRS is bounded). For $\rho < 0$ we get unbounded tubes which is unavoidable for non-exponentially stabilizable systems. The contraction rate serves as a hyper-parameter whose auto-tuning is discussed in Section 3.2.

The stability of the nonlinear system is directly implied by integrating the differential dynamics stabilized by condition (2.38) and (2.39).

Given the found contraction metrics we integrate along the path γ that connects our true and nominal state to obtain the incremental Lyapunov function $V_\sigma(x,z)$, which parametrizes our DRS

$$V_\sigma(x,z) = \min_\gamma \int_0^1 \|\gamma_s(s)\|_{M(\gamma(s))}\, \mathrm{d}s \tag{2.40}$$

with $\gamma_s = \frac{\partial \gamma}{\partial s}\Big|_s$. Analogously we get our feedback $\kappa(x,z,v)$ by integrating the differential feedback $K(x)$ along the shortest path $\gamma^\star$, connecting true and nominal state

$$\kappa(x,z,v) = v + \int_0^1 K(\gamma^\star(s))\gamma_s^\star(s)\, \mathrm{d}s. \tag{2.41}$$

2.4.2 Tube Dynamics

With our prestabilized system and the incremental Lyapunov function we have all the ingredients needed for the parametrization of the DRS, except for the dynamics needed for the propagation of the tube. The tube is defined as

$$\Omega(z(t), \sigma(t)) := \{x \mid V_\sigma(x, z(t)) \leqslant \sigma(t)\}, \quad \forall\, t \geqslant 0 \tag{2.42}$$

with $\sigma(t)$ being the scalar tube scaling.

The tube dynamics is derived by taking the derivative of the incremental Lyapunov function and can be upper-bounded by $\dot{\sigma}$ as follows

$$\dot{\sigma} = -(\rho - L_{\mathcal{W}})\sigma + \max_{w \in \mathcal{W}} \|E(z)w\|_{M(z)} \tag{2.43}$$

$$= f_\sigma(\sigma, z, v) \tag{2.44}$$

with $L_{\mathcal{W}}$ the continuity constant of the disturbance matrix $E(x)$, which is given by the expression

$$L_{\mathcal{W}} := \max_{\substack{x \in \mathcal{C} \\ w \in \mathcal{W}}} \left\| E_s(x) M(x)^{-\frac{1}{2}} \right\| \tag{2.45}$$

with $E_s(x) = \frac{\partial M(x)^{\frac{1}{2}} E(x) w}{\partial x}\Big|_x$. The dynamic tube scaling reduces conservatism while ensuring that the true and nominal trajectory satisfy Equation (2.46).

2.4.3 Constraint Tightening

For our final MPC formulation we still need to make sure that our true state and input is contained in our constraint set defined in (2.3) given all possible disturbance realisations. We ensure this by tightening our constraints as follows

$$a_i^\top (z, v) + c_i V_\sigma(x, z) \leqslant b_i, \forall\, i \in \mathbb{N}_{n_c}, \tag{2.46}$$

with the constants c_j obtained by

$$c_j := \max_{(z,v) \in \mathbb{C}} \left\| \left(a_i \begin{bmatrix} \mathcal{I}_{n_x} \\ K(z) \end{bmatrix} \right) M(z)^{-\frac{1}{2}} \right\|. \tag{2.47}$$

The complete MPC formulation is given by

$$\begin{aligned}
\min_{\boldsymbol{z}, \boldsymbol{v}, \boldsymbol{\sigma}} \quad & J_T(x_0, \boldsymbol{z}, \boldsymbol{v}), && && (2.48)\\
\text{s.t.} \quad & z_0 = x_0, && && (2.49)\\
& \dot{z} = f_w(\mathrm{z}, v, 0), && \forall\, t \geqslant 0 && (2.50)\\
& \dot{\sigma} = f_\sigma(\sigma, z, v), && \forall\, t \geqslant 0 && (2.51)\\
& a_i^\top (z, v) + c_i\, \sigma \leqslant b_i, && \forall\, t \geqslant 0,\ \forall\, i \in \mathbb{N}_{n_c} && (2.52)
\end{aligned}$$

and provides robust constraint satisfaction, recursive feasibility (given an appropriately chosen terminal set) and certificate tubes around the nominal state in which the error is guaranteed to lie in. In contrast to [8] we do not use a terminal constraint in order to allow a comparison with SLS, which also does not incorporate a terminal constraint. The control contraction metrics, the continuity constant $L_{\mathcal{W}}$ and the tightenings c_j are offline computed constants. The formulas presented in this chapter are a summary of what is proven rigorously in [8]. Refer to the original paper for references and derivations.

Chapter 3

Methods

In this chapter, we explain the methodology used for comparing the two RMPC approaches, including the necessary adaptations to ensure a fair and reproducible evaluation.

Given the fundamental differences between the two RMPC approaches, several considerations must be addressed prior to comparison. The table below highlights the main conceptual differences between the two methods.

Table 3.1: Conceptual differences of SLS and CCM

	Time	**Uncertainty Propagation**	**Feedback Computation**	**Offline Computations**	**Online Complexity**	**Offline Design Effort**
SLS	discrete	linearized	online	μ	high	low
CCM	continuous	CCM	offline	CCM	low	high

The two main distinctions that must be handled carefully are the change from discrete to continuous time and a consistent selection of the hyper-parameter ρ used in the construction of the CCM. These two things are addressed in the following sections.

Furthermore, a framework was developed that allows for a streamlined comparison of the two methods. This was achieved by separating the problem into submodules. This involved a complete refactoring of the code from [8], a routine for saving and loading offline computations and further tools that are needed to generalize the code for nonlinear systems of the form presented in the problem setup (2.1). A flowchart of the final framework used for comparison is depicted to the right.

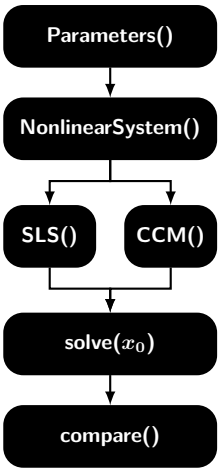

Figure 3.1: General framework for a fair comparison on nonlinear RMPC methods, allowing for different nonlinear dynamics.

3.1 Changes Made to SLS-based RMPC

3.1.1 Non-Affine Discretizations

So far we have seen how to discretize continuous time systems using the affine Euler discretization. Non-affinity preserving discretizations such as RK4 or RKDP can not be utilized in the same way as their discretization nonlinearly depends on w. We can mitigate this issue by using an over-approximation for E_d. Thereby we compute E_d such that it over-bounds the difference of the discretized system with and without noise

$$E_d\, w_{max} = \max_{\substack{(\mathrm{x},\mathrm{u}) \in \mathcal{C},\, w \in \mathcal{W}, \\ f_d(x,u,w) \in \mathcal{C}}} |f_d(x,u,w) - f_d(x,u)| \tag{3.1}$$

with $f_d(x,u,w)$ denoting the discretization of the noisy system and $f_d(x,u,0)$ being the discretized nominal system.

3.1.2 Prestabilizing Feedback Gain

To reduce conservatism, we employ a prestabilizing feedback gain matrix K that directly stabilizes our nonlinear system. This approach results in a smaller Hessian over-bound, assuming that the prestabilizing feedback was designed well, which is less subjective to increasing sample times.

In our implementation, we utilize the feedback gain provided by CCM with a constant parametrization assuming that a feedback that stabilizes our system also leads to a smaller Hessian over-bound. However, it is generally possible to use any method to obtain a suitable feedback gain.

The new prestabilized system writes as

$$\begin{aligned} \dot{x} &= \tilde{f}_w(x,\tilde{u},w) && (3.2)\\ &= \underbrace{f(x) + B(x)Kx}_{\tilde{f}(x)} + B(x)\tilde{u} + E(x)w && (3.3) \end{aligned}$$

with

$$u = Kx + \tilde{u} \tag{3.4}$$

being the true input to our system and $\tilde{u}$ being our virtual input.

Furthermore we shift the constraint set defined in (2.3) to obtain

$$\tilde{\mathcal{C}} := \{(x,\tilde{u}) \mid a_i^\top \tilde{K}\,(x,\tilde{u}) \leqslant b_i, \forall\, i \in \mathbb{N}_{n_c}\} \tag{3.5}$$

with $\tilde{K}$ being the matrix defined by

$$\tilde{K} = \begin{bmatrix} \mathcal{I}_{n_x,n_x} & 0_{n_x,n_u} \\ K & \mathcal{I}_{n_u,n_u} \end{bmatrix}. \tag{3.6}$$

Replacing $a_i^\top$ in our MPC constraint (2.34) with $\tilde{a}_i^\top = a_i^\top \tilde{K}$ hence suffices to account for our prestabilization.

What is left to do is to compute the disturbance feedback on our true input. The disturbance feedback of our prestabilized system is given by

$$\begin{aligned} \mathbf{x} &= \mathbf{z} + \mathbf{\Phi}_x \mathbf{w} && (3.7)\\ \tilde{\mathbf{u}} &= \tilde{\mathbf{v}} + \mathbf{\Phi}_{\tilde{u}} \mathbf{w} && (3.8) \end{aligned}$$

as obtained from Equation (2.27). By substituting expression (3.7) and (3.8) into Equation (3.4) we get

$$\mathbf{u} = \underbrace{\tilde{\mathbf{v}} + K\mathbf{z}}_{\mathbf{v}} + \underbrace{\left(\mathbf{\Phi}_{\tilde{u}} + K\mathbf{\Phi}_{\mathrm{x}}\right)}_{\mathbf{\Phi}_{\mathrm{u}}} \mathbf{w} \tag{3.9}$$

with $\mathbf{v} = \tilde{\mathbf{v}} + K\mathbf{z}$ our true nominal input and $\mathbf{\Phi}_{\mathrm{u}} = \mathbf{\Phi}_{\tilde{u}} + K\mathbf{\Phi}_{\mathrm{x}}$ being our true disturbance feedback. This expression is not directly required in the MPC formulation, but the nominal input $\mathbf{v}$ is necessary for closed-loop simulation. Additionally, $\mathbf{\Phi}_{\mathrm{u}}$ characterizes our DRS in relation to the input.

3.2 Changes Made to CCM-based RMPC

3.2.1 Auto-Tuning of ρ

The contraction coefficient ρ is a crucial hyper-parameter for the design of the CCM and hence for the parametrization of the DRS. In the following we propose how to automatically select ρ based on the asymptotic tightening.

Let's recall the constraint tightening defined in Equation (2.46). The original constraints (2.3) hereby get tightened by the coefficients $c_i\,\sigma$. Our tube dynamics is given by Equation (2.43) and can be written as follows assuming that our nominal state converged to our reference state $z_\infty = x_{ref}$

$$\dot{\sigma} = -(\rho - L_{\mathcal{W}})\sigma + \bar{w} \tag{3.10}$$

with $\bar{w} = \max_{w \in \mathcal{W}} \|E(z_\infty)w\|_{M(z_\infty)}$.

Given our stationary $\bar{w}$ we can solve the differential equation in (3.10) such that we get

$$\sigma(t) = \frac{\bar{w}}{\alpha}\left(1 - e^{-\alpha t}\right) \tag{3.11}$$

with $\alpha = (\rho - L_{\mathcal{W}})$.

For $\alpha > 0$ we have a positive effective contraction rate and thus one can derive an asymptotic tube scaling with $t \to \infty$

$$\sigma_\infty = \frac{\bar{w}}{(\rho - L_{\mathcal{W}})}. \tag{3.12}$$

Given the asymptotic tube scaling we can select the optimal contraction rate $\rho^\star$ based on the minimum normalized asymptotic tightening defined by

$$\rho^\star = \arg\min_{\rho} \sum_{i=1}^{n_{\mathcal{C}}} \frac{c_i\,\sigma_\infty}{b_i - a_i^\top(z_\infty, v_\infty)}. \tag{3.13}$$

This performance measure prefers solutions that produce the lowest percental overall tightening. One possible extension to this would be a weighting of the individual tightenings based on a manually selected priority for different states. We can implement this minimization by gridding over ρ effectively computing a CCM for each ρ and than selecting the one that minimizes the above relation. A similar procedure was already hinted at by [10] who swept through a range of different values for ρ manually selecting an optimal $\rho^\star$ that minimizes the DRS.

If we deal with dynamic systems that are not exponentially stabilizable we can encounter situations where we want to choose $\rho < 0$ effectively allowing an unbounded DRS. In this case we need an alternative to quantify σ_∞. One way of doing that is to analyze the tube size over the MPC prediction horizon N which can be done with (3.11). Replacing σ_∞ in (3.13) by σ_T again allows us to quantify the performance of our CCM over our horizon.

Chapter 4

Comparisons

This chapter compares three robust MPC methods, i.e.: SLS (Section 2.3), CCM (Section 2.4) and the intermediate method presented in Section 3.1.2 from here on referred to as SLSK. The methods are compared based on two different dynamics i.e.: the planar quadrotor from [8] and the satellite post-capture stabilization from [12]. The performance measures compared are the tightening with active constraints and the cost. The parameters varied are the maximum disturbance w_{max}, the horizon length $T = h\,N$ and the sample time h.

All online optimizations are done with IPOPT [18] through Casadi [19] in Matlab whereas the offline construction of the CCM is done using Mosek [20] interfaced through YALMIP [21]. We show only feasible solutions in our figures and we mark solutions that are only "Solved To Acceptable Level" with a $\times$. We did not compare the online computation times of the three methods because SLS currently requires a significant amount of time to solve using IPOPT [18]. This issue is addressed with a tailored solver, as proposed by [22], which is however not yet available for nonlinear systems.

The offline computations for SLS, specifically calculating the Hessian over-bound μ, take approximately five minutes with a grid of one million samples. For CCM, the offline computations which involve determining M, K, the tightenings c_j, and the continuity constants L_j over a fixed grid of ten candidate ρ values, takes around five hours with fifty thousand LMI-samples to enforce condition (2.38) and (2.39) and fifty thousand samples for computing the tightenings and continuity constants.

4.1 Planar Quadrotor

This planar quadrotor model has been widely used in previous studies [8, 9, 10] as it offers an intuitively understandable example of an unstable, nonlinear system. It features six generalized coordinates, two inputs, and is subject to a scalar disturbance. We conducted two comparisons using different state constraints applied to this dynamics.

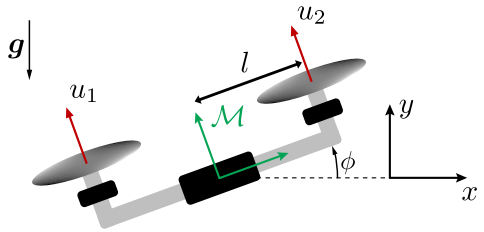

Figure 4.1: The planar quadrotor is defined through the inertial frame with the basis vectors x and y as well as the rotation ϕ. The velocities of the system are expressed in the moving $\mathcal{M}$-frame (green). We further define the mass m, the inertia J, the thrust moment arm l (symmetric), and the inputs u_1 and u_2 the left and right thrust forces respectively.

4.1.1 Dynamics

The set of generalised coordinates that fully describe the state of the system is given by

$$\boldsymbol{x} = \begin{bmatrix} x & y & \phi & \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}^\top \tag{4.1}$$

with the control input being

$$\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\top. \tag{4.2}$$

The dynamics of the planar quadrotor is given as

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \dot{x}\cos(\phi) - \dot{y}\sin(\phi) \\ \dot{x}\sin(\phi) + \dot{y}\cos(\phi) \\ \dot{\phi} \\ \dot{y}\,\dot{\phi} - g\sin(\phi) \\ -\dot{x}\,\dot{\phi} - g\cos(\phi) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{m} & \frac{1}{m} \\ \frac{l}{J} & \frac{-l}{J} \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \cos(\phi) \\ -\sin(\phi) \\ 0 \end{bmatrix} w. \tag{4.3}$$

Further we define the stage cost $Q = \mathcal{I}_{n_x}$ for the state, $R = \mathcal{I}_{n_u}$ for the input, along with a terminal cost $P = 30\,\mathcal{I}_{n_x}$ for our MPCs.

4.1.2 Comparison on Larger Constraint Set

For the first comparison we set the origin as reference for state $\boldsymbol{x}_{ref} = 0_{n_x}$ and input $\boldsymbol{u}_{ref} = 0_{n_u}$. The initial state of the quadrotor is given by $\boldsymbol{x}_0 = [-2 \;\; -15 \;\; 0 \;\; 0 \;\; 0 \;\; 0]^\top$, and the state and input constraints are specified as follows

$$x \in [-20,\ 20], \qquad \dot{x} \in [-2,\ \ 2], \tag{4.4}$$

$$y \in [-20,\ 20], \qquad \dot{y} \in [-1,\ \ 1], \tag{4.5}$$

$$\phi \in \left[-\frac{\pi}{3},\ \frac{\pi}{3}\right], \qquad \dot{\phi} \in [-\pi,\ \ \pi], \tag{4.6}$$

$$u_1 \in [-1,\ 3.5], \qquad u_2 \in [-1,\ 3.5]. \tag{4.7}$$

The discretization method used is RK4.

Figure 4.2 shows a single trajectory obtained using CCM-based RMPC. Each subplot depicts a pair of state or input trajectories, with constraints indicated by thick straight lines. The solid and dashed lines differentiate between the trajectories, while the colored areas surrounding them represent the DRS.

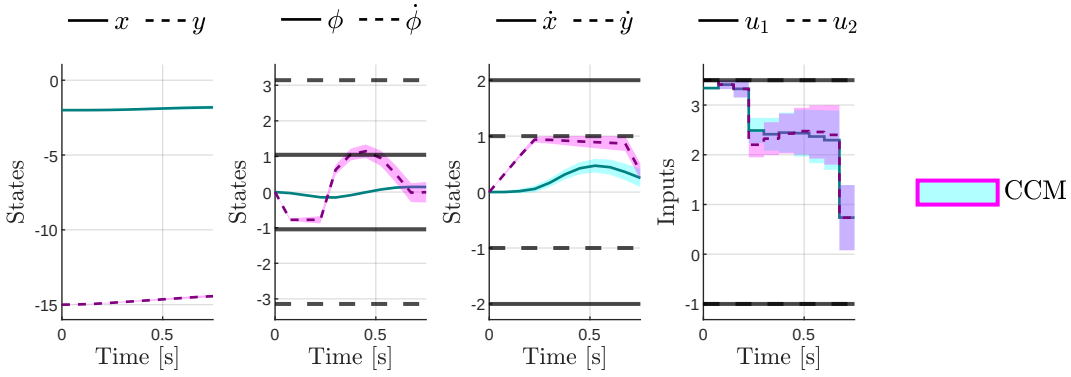

Figure 4.2: Trajectory for CCM with disturbance $w_{max} = 0.1$, sample time $h = 0.075$ and horizon $N = 10$

To use constraint tightenings as a performance measure it is crucial to ensure active constraints as tightenings for states with no active constraints can get arbitrarily big. Hence an appropriate cost, reference state and input as well as initial state must be selected. The tightening is then evaluated at a specific time step and state. The initial- and reference state chosen in (4.1.2) guarantee active constraints for the velocity $\dot{y}$ as we start far from our reference state in the y-direction. Figure 4.2 shows that a reasonable comparison of the tightenings can be made in the $\dot{y}$ direction at the time step $t = h(N-2)$.

Figure 4.3 shows the constraint tightening for the active constraint $\dot{y}$ with varying disturbance w_{max} evaluated at time step $t = h(N-2)$. One can quickly notice that CCM-based RMPC is more conservative than SLS and becomes infeasible earlier. The tightening of SLS grows very rapidly when approaching a disturbance of $w_{max} = 0.22$ and then suddenly stops to increase. It is worth to mention here that the quality of the solution for SLS with $w_{max} > 0.22$ is questionable as it starts to converge to solutions that are only "Solved to Acceptable Level" (marked with a $\times$) which is also reflected in the increasing cost. Furthermore we can see that CCM has a larger cost than SLS for disturbances smaller than $w_{max} < 0.22$ which agrees well with the larger tightenings of CCM.

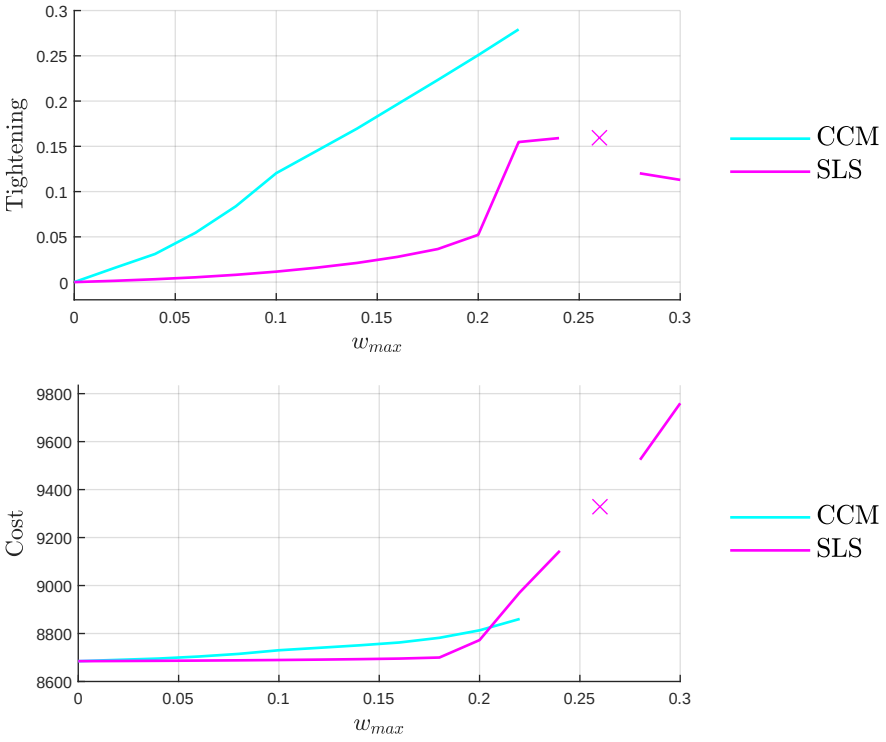

Figure 4.3: Tightening evaluated at time step $t = h(N-2)$ and cost for varying disturbance w_{max} with horizon $N = 10$, and sample time $h = 0.075$

We can analyze the trajectories for CCM for $w_{max} = 0.22$ to understand why it gets infeasible. Looking at Figure 4.4 one can immediately understand that the feasibility issue arises from a significant increase of the DRS of the inputs, which eventually exceeds the input constraints.

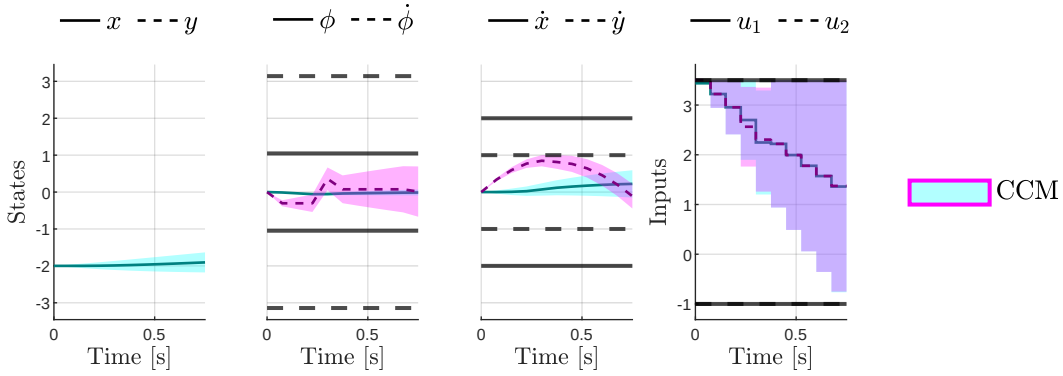

Figure 4.4: Trajectory for CCM with disturbance $w_{max} = 0.22$, sample time $h = 0.075$ and horizon $N = 10$

Shown in Figure 4.5 we can see the tightening for increasing sample times. One can realise that SLS is limited to shorter sample times given the unstable dynamics of the planar quadrotor. CCM on the other hand is less susceptible to increasing sample time as it is a continuous time method only relying on the discretization for enforcing nonlinear constraints. The cost decreases for increasing sample time as we get closer to the reference state with a longer sample time implying a longer time horizon $T = h\,N$. Again SLS exhibits a smaller cost and tightening at the beginning but than starts to grow rapidly as it approaches $h = 0.1$ where it becomes infeasible.

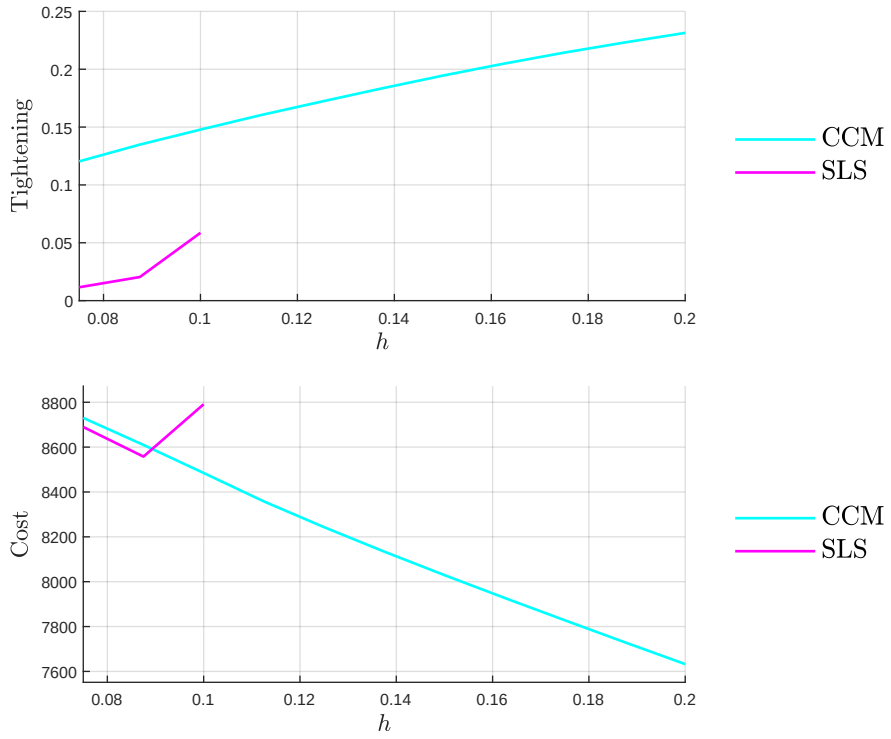

Figure 4.5: Tightening evaluated at time step $t = h(N-2)$ and cost for varying sample time h with horizon $N = 10$, and disturbance $w_{max} = 0.1$

The feasibility issue of SLS can be understood studying the Hessian over-bound μ depicted in Table 4.1 needed to quantify our linearization error. One can see that μ starts to grow quadratically for state 4 and 5 when approaching $h = 0.1$ eventually leading to an infeasible problem. The Hessian over-bound μ primarily influences the velocity constraints, as the constraint set for the x- and y-positions is considerably large.

Table 4.1: μ with varying sample time h shortly before MPC gets infeasible.

h	μ					
0.0870	[0.2825	0.2775	0	2.5439	2.5697	0]
0.1000	[0.3800	0.3729	0	4.3322	4.7782	0]
0.1125	[0.5476	0.5401	0	7.4513	8.6602	0]

The tightening and cost for varying horizon N are depicted in Figure 4.6. We can see the previously observed trends of CCM being more conservative for both tightening and cost. Further one can observe that the tightening of CCM grows according to the tube dynamics (3.11) whereas SLS exhibits a close to quadratic growth. The tightenings are only shown starting from a horizon $N = 6$ as constraints are not active before. The dashed trend-lines intersect the N-axis at $N = 2$ because we evaluate our tightenings at $t = h(N - 2)$. Additionally, the two trend-lines intersect each other at $N = 55$, indicating that SLS becomes more conservative than CCM at this horizon length.

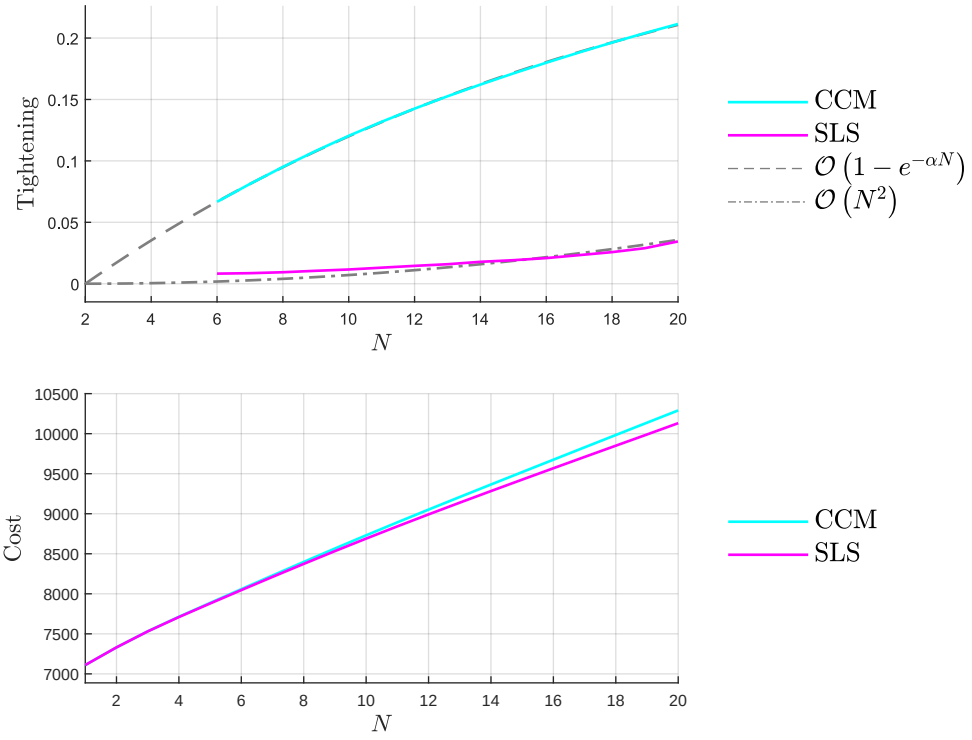

Figure 4.6: Tightening evaluated at time step $t = h(N - 2)$ and cost for varying horizon N, with sample time $h = 0.075$ and disturbance $w_{max} = 0.1$

4.1.3 Results on Smaller Constraint Set

The following results are presented for a constraint set with tightened constraints for ϕ and $\dot{\phi}$ defined by

$$x \in [-20,\ 20], \qquad \dot{x} \in [-2,\ 2], \tag{4.8}$$
$$y \in [-20,\ 20], \qquad \dot{y} \in [-1,\ 1], \tag{4.9}$$
$$\phi \in \left[-\frac{\pi}{20}, \frac{\pi}{20}\right], \qquad \dot{\phi} \in \left[-\frac{\pi}{10}, \frac{\pi}{10}\right], \tag{4.10}$$
$$u_1 \in [-1,\ 3.5], \qquad u_2 \in [-1,\ 3.5] \tag{4.11}$$

to allow a comparison of SLS and CCM against SLSK. The tightened constraints are necessary as SLSK relies on a constant prestabilizing feedback matrix K which can only be found using CCM for smaller constraint sets with reduced nonlinearity.

The integrator used for this comparison is RKDP which provides accurate discretization helping to exclude the discretization method as a parameter influencing the obtained results.

Figure 4.7 depicts the tightening for varying disturbance w_{max} evaluated at time step $t = h(N-2)$ and the active constraint $\dot{\phi}$. When comparing this to Figure 4.3 one can observe that the tightenings obtained with CCM are almost unaffected by the tightened constraint set whereas SLS is able to exploit the reduced nonlinearity resulting in a strongly reduced tightening. Regarding SLSK we see a slightly worse tightening than with SLS and a cost which is substantially larger than the one from SLS. Apparently the prestabilizing feedback K is not helping to reduce the Hessian over-bound effectively. The same can be observed for the other comparisons which can be found in the Appendix A.2.

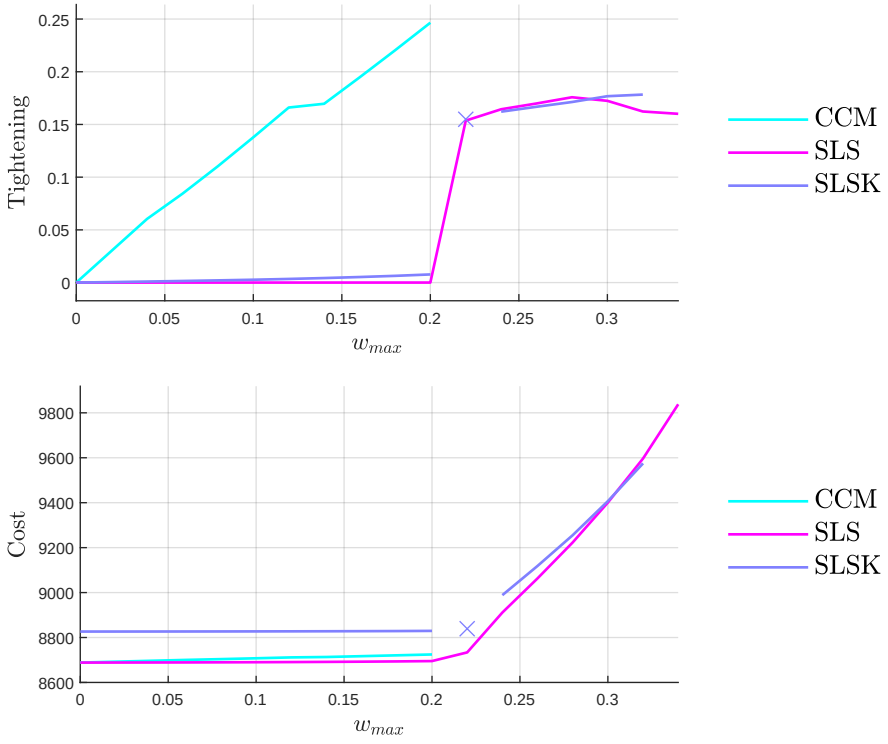

Figure 4.7: Tightening evaluated at time step $t = h(N-2)$ and cost for varying disturbance w_{max} with horizon $N = 10$, and sample time $h = 0.075$

This hints at the fact that the prestabilization of SLSK either does not work due to an intrinsic limitation or because of badly designed feedback K. According to Table 4.2, the feedback matrix that stabilizes our nonlinear system does not consistently reduce the Hessian over-bound. Specifically, certain entries of μ decrease for some sample times but not for others (e.g., entry no. 4). Further analysis is needed to show if the Hessian over-bound can be reduced consistently by a suitable feedback matrix.

Table 4.2: μ with varying sample time h shortly before MPC gets infeasible.

h		μ					
0.1050	SLS	[0.2831	0.2419	0.0000	3.2452	3.3533	0.0000]
	SLSK	[0.3879	0.3385	0.0167	2.6927	2.1102	0.6620]
0.1125	SLS	[0.3385	0.3017	0.0000	3.9024	4.0442	0.0000]
	SLSK	[0.4987	0.4551	0.0213	3.2847	2.5637	0.8094]
0.1200	SLS	[0.4187	0.4032	0.0000	4.6561	4.8484	0.0000]
	SLSK	[0.6750	0.6278	0.0270	4.6632	3.2011	0.9968]

4.2 Satellite Post-Capture Stabilization

The satellite post-capture stabilization as used in [12] provides and example of a non-exponentially stabilizable system which results in an unbounded DRS for CCM. It features similarly to the planar quadrotor six states and two inputs and is subject to a three-dimensional disturbance.

4.2.1 Dynamics

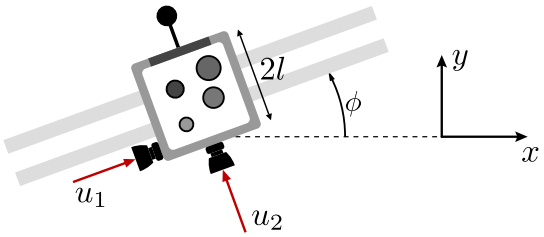

Figure 4.8: The satellite post-capture stabilization is defined through the x- and y-position as well as the rotation ϕ. We further define the mass m, the inertia J, the thrust moment arm l, and the inputs u_1 and u_2 the left and lower thrust forces respectively.

The set of generalised coordinates that fully describe the state of the system is given by

$$\boldsymbol{x} = \begin{bmatrix} x & y & \phi & \dot{x} & \dot{y} & \dot{\phi} \end{bmatrix}^\top \tag{4.12}$$

with the control input being

$$\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^\top = \begin{bmatrix} v_x & v_y \end{bmatrix}^\top. \tag{4.13}$$

The dynamics of the post-capture stabilization is given as

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\phi} \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{\cos(\phi)}{m} & -\frac{\sin(\phi)}{m} \\ \frac{\sin(\phi)}{m} & \frac{\cos(\phi)}{m} \\ \frac{l}{J} & 0 \end{bmatrix} \boldsymbol{u} + \begin{bmatrix} 0_{3,3} \\ \mathcal{I}_3 \end{bmatrix} w. \tag{4.14}$$

Further we define the stage cost $Q = \mathcal{I}_{n_x}$ for the state, $R = \mathcal{I}_{n_u}$ for the input, along with a terminal cost $P = 30\mathcal{I}_{n_x}$ for our MPCs.

4.2.2 Results

The comparison was done with the reference state $\boldsymbol{x}_{ref} = [0 \quad 15 \quad 0 \quad 0 \quad 0 \quad 0]^\top$, and input $\boldsymbol{u}_{ref} = 0_{n_u}$. The initial state is given by $\boldsymbol{x}_0 = [0.7 \quad 0.7 \quad 0.5 \quad 0.5 \quad 0.5 \quad 0.5]^\top$, and the state and input constraints are defined as $-1 \leqslant \boldsymbol{x}_i \leqslant 1 \; \forall \, i \in \mathbb{N}_{n_x}$ and $-1 \leqslant \boldsymbol{u}_i \leqslant 1 \; \forall \, i \in \mathbb{N}_{n_u}$ respectively. The discretization method used is RK4.

Figure 4.9 shows the constraint tightening for the active constraint y with varying disturbance w_{max} evaluated at time step $t = hN$. The trends that we observe here are very different from what we previously saw in Figure 4.3 and 4.7. With the non-exponentially stabilizable dynamics and CCMs unbounded DRS we can observe a significant difference in the tightenings. SLS shows a minimal tightening, with only a slight increase as disturbances grow, whereas CCM exhibits rapid growth in tightening. The same can be observed when analyzing the cost. Note that the gap visible in the curve for SLS is due to an infeasible solution. Further plots showing these trends can be found in Appendix A.3.

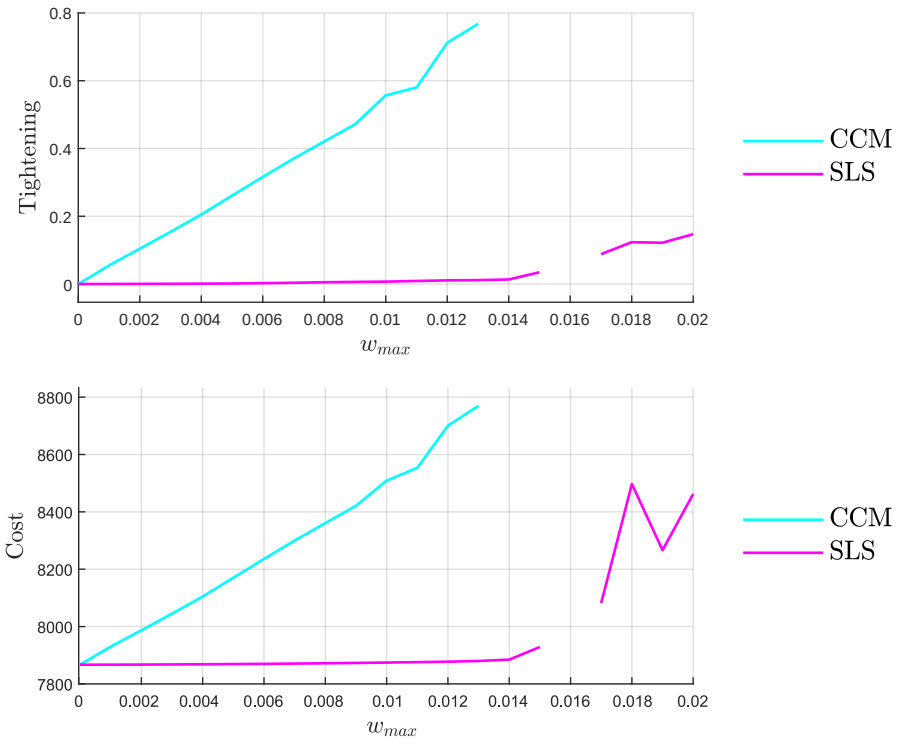

Figure 4.9: Tightening evaluated at time step $t = hN$ and cost for varying disturbance w_{max} with horizon $N = 10$, and sample time $h = 0.5$

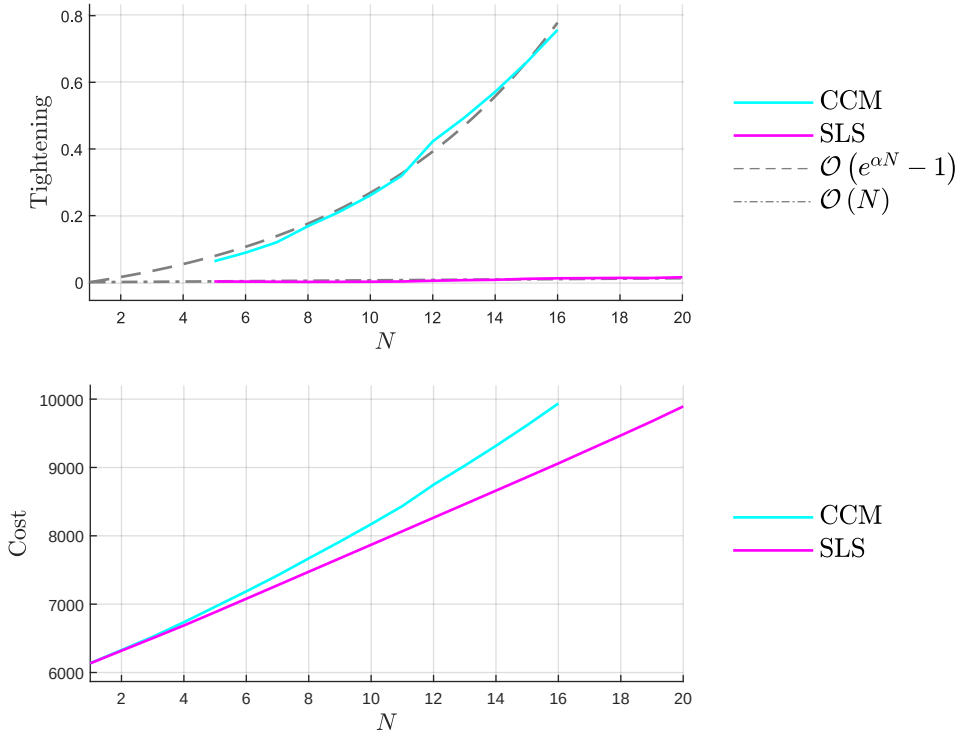

Figure 4.10: Tightening evaluated at time step $t = hN$ and cost for varying horizon N, with sample time $h = 0.5$ and disturbance $w_{max} = 0.005$

When analyzing the tightening and the cost as shown in Figure 4.10 we can observe the exponential growth of CCMs DRS as expected from the tube dynamics (3.11) for non-exponentially stabilizable systems. Here again the tightenings of SLS remain small over the entire horizon.

Figure 4.11 explains why SLS can effectively minimize both the tightening and the cost across all horizon lengths. The method demonstrates the ability to squeeze its tube at the end of the horizon, particularly evident in the state y, which is used to evaluate the tightenings. Consequently, the tightenings shown in Figure 4.10 are consistently small.

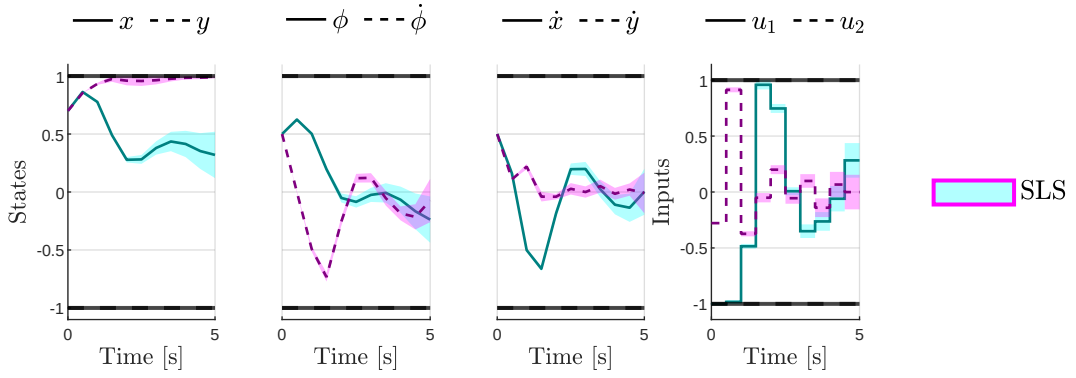

Figure 4.11: Trajectory for SLS with disturbance $w_{max} = 0.014$, sample time $h = 0.5$ and horizon $N = 10$

Chapter 5

Conclusion

In this work, a general framework was developed for comparing nonlinear RMPC methods on any dynamics of the form 2.1. The framework was implemented for SLS- and CCM-based RMPC and tested on the planar-quadrotor dynamics (Section 4.1) as well as the satellite post-capture stabilization dynamics (Section 4.2). In addition, a third intermediate method SLSK was introduced (Section 3.1.2), implemented and tested based on the planar quadrotor dynamics (Subsection 4.1.3).

Various parameters have been varied, such as the disturbance w_{max}, the horizon length N, the sample time h and the level of nonlinearity, in order to study the conservatism and intrinsic limitations of the RMPC approaches compared.

Based on all our comparisons, we found that CCM is generally more conservative than SLS. We were able to show that SLS can exploit reduced nonlinearity effectively, e.g. given through (tight) constraints, while CCM is still rather conservative. In addition to that, we found non-exponentially stabilizing dynamics such as the satellite post-capture stabilization as an edge case for CCM. A limiting factor that was discovered for SLS is the sample time h, especially for unstable systems. Further limiting is the computational power, as it requires a significant amount of time to solve the large joint optimization problems. Regarding SLSK we had to realize that the feedback matrix K that stabilizes our nonlinear dynamics does not consistently reduce our Hessian over-bound μ. Accordingly SLSK did not perform better than SLS.

As a rule of thumb we suggest to resort to CCM-based RMPC for systems with reasonably large constraint sets and a fast dynamics. We see the application for SLS on the other hand for systems with a slow dynamics and tight constraints, such as e.g. building control or offline (path-) planning.

Further work could include an in-depth analysis for SLSK to conclude if the Hessian over-bound can be reduced consistently by a suitable feedback matrix K. Additionally, comparing the two RMPC approaches on other dynamics, such as e.g. from [23], would be reasonable, as the developed framework now makes this process straightforward and further insights could be gained. Last but not least it would be interesting to see other RMPC approaches compared on the framework developed. The methods described in [5, 13], which fix a prestabilizing error feedback offline while optimizing the DRS online, could be a valuable enhancement, as they pose intermediate methods to the approaches compared in this thesis.

Appendix A

Appendix

A.1 Lagrange Error Bound

The Lagrange error bound gives us a way to qualify the error of the n$^{\text{th}}$-Taylor approximation. The Taylor approximation with it's reminder is given by the expression below.

$$f(x) = f(x_0)+f^{(1)}(x_0)(x-x_0)+\frac{f^{(2)}(x_0)}{2}(x-x_0)^2+\cdots+\frac{f^{(n)}(x_0)}{n!}(x-x_0)^n+\underbrace{\int_{x_0}^{x}\cdots\int_{x_0}^{x} f^{(n+1)}(x)dx\cdots dx}_{n+1 \text{ integrals}}$$

This expression is obtained by integrating f^{n+1} (i.e. the $(n+1)^{\text{th}}$ derivative) $n+1$ times. This can e.g. be done by integration by parts. Let's rewrite this expression into

$$f(x) = T_n(x) + R_n(x)$$

with $T_n(x)$ being the Taylor approximation and $R_n(x)$ being the remainder. The Lagrange error bound is obtained when upper bounding the remainder which gives us

$$R_n(x) = \int_{x_0}^{x}\cdots\int_{x_0}^{x} f^{(n+1)}(x)dx\cdots dx \leqslant M\frac{|x-x_0|^{n+1}}{(n+1)!}$$

with $M = \max_{x_0\leqslant z\leqslant x}|f^{(n+1)}(z)|$.

Hence we found an upper bound for the Taylor approximation

$$R_n(x) = f(x) - T_n(x) \leqslant M\frac{|x-x_0|^{n+1}}{(n+1)!}$$

A.2 Planar Quadrotor Plots

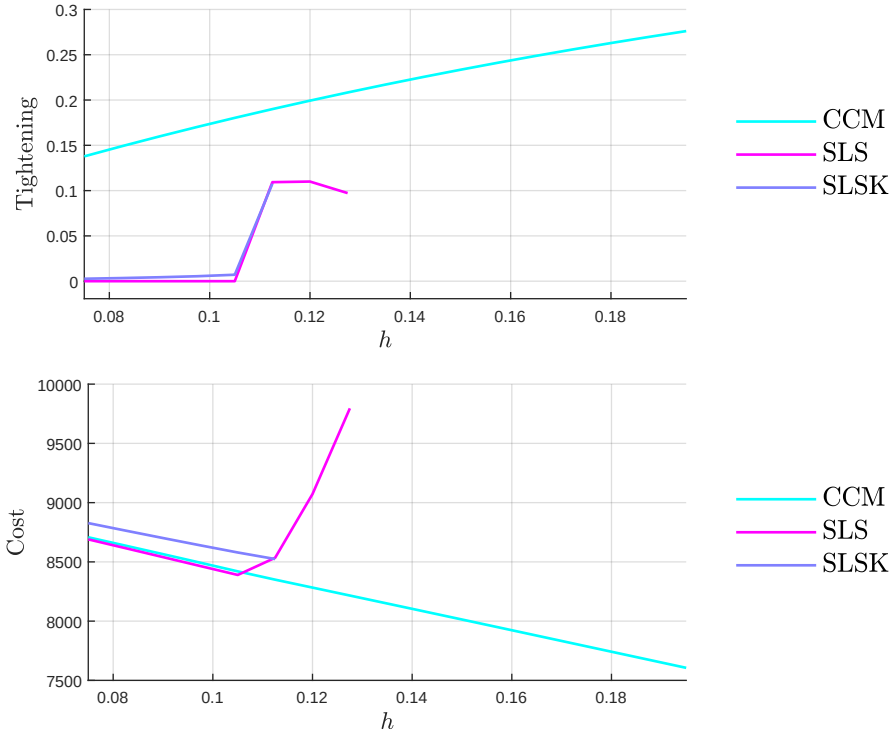

Figure A.1: Tightening evaluated at time step $t = h(N-2)$ and cost for varying sample time h with horizon $N = 10$, and disturbance $w_{max} = 0.1$

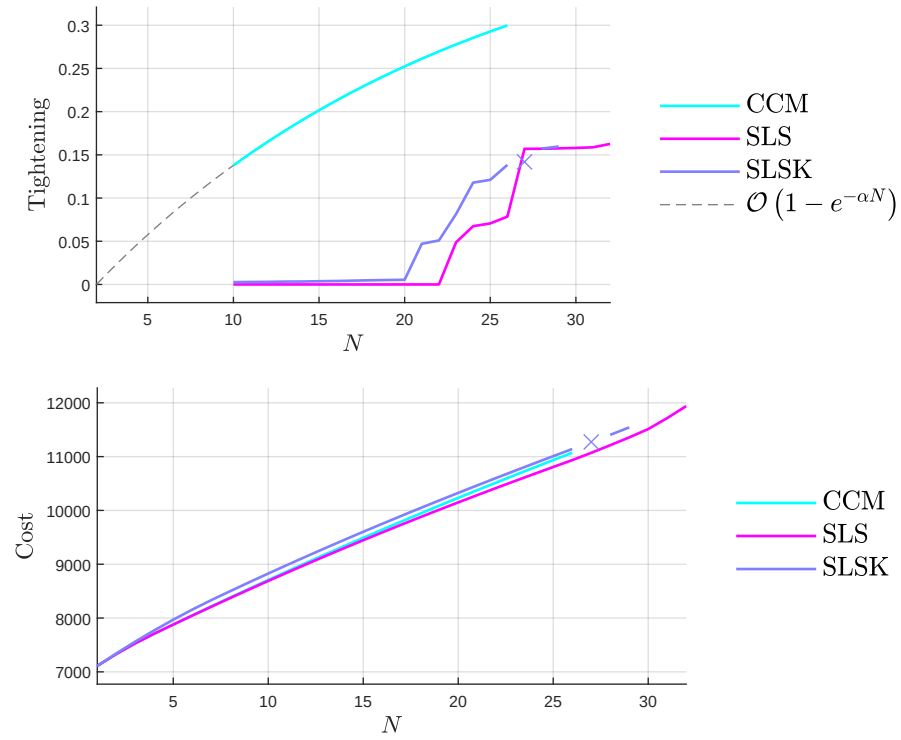

Figure A.2: Tightening evaluated at time step $t = h(N-2)$ and cost for varying horizon N, with sample time $h = 0.075$ and disturbance $w_{max} = 0.1$

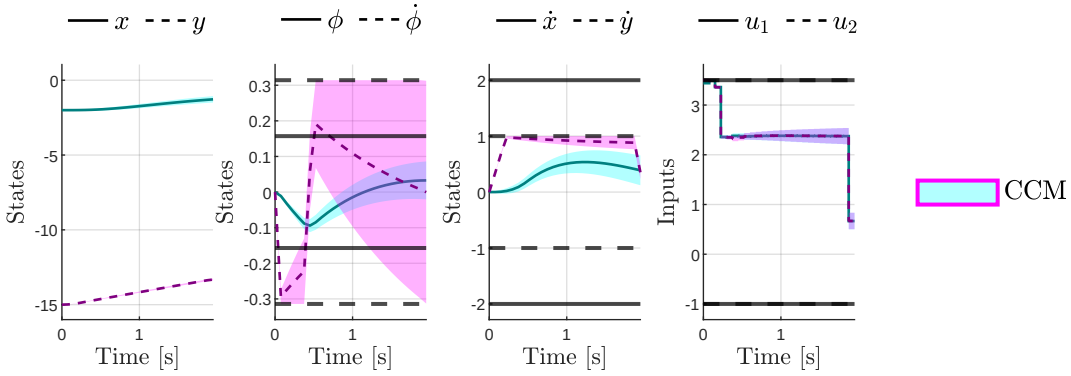

Figure A.3: Trajectory for CCM with disturbance $w_{max} = 0.1$, sample time $h = 0.075$ and horizon $N = 26$

A.3 Satellite Post-Capture Stabilization Plots

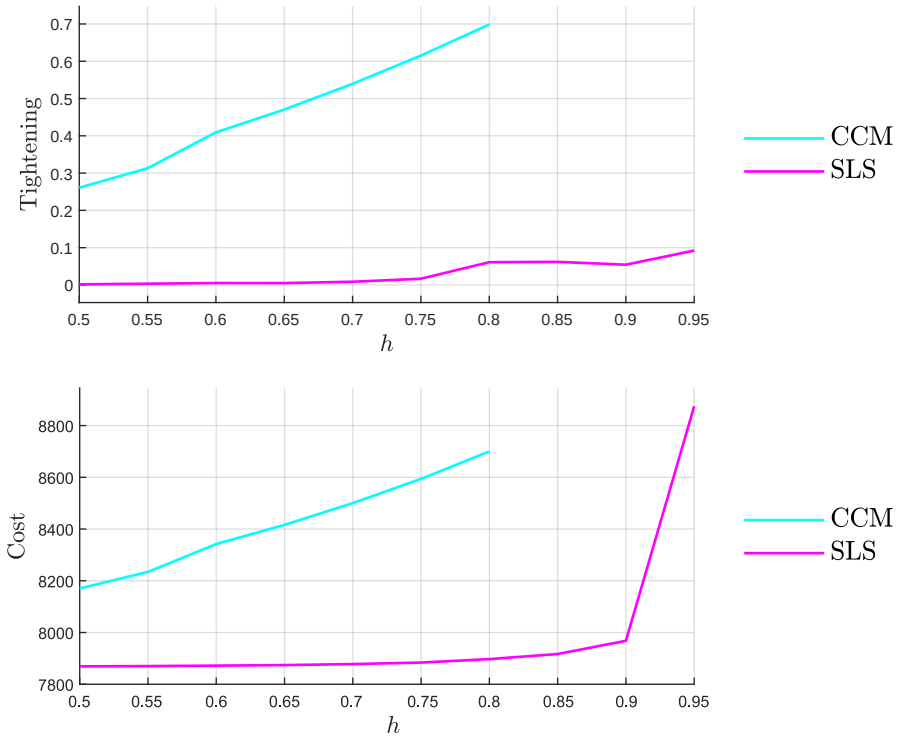

Figure A.4: Tightening evaluated at time step $t = hN$ and cost for varying sample time h with horizon $N = 10$, and disturbance $w_{max} = 0.005$

Bibliography

[1] S. J. Qin and T. A. Badgwell, "A survey of industrial model predictive control technology," *Control Engineering Practice*, vol. 11, no. 7, pp. 733–764, Jul. 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0967066102001867

[2] J. B. Rawlings, D. Q. Mayne, and M. Diehl, *Model Predictive Control: Theory, Computation, and Design*, 2nd ed. Santa Barbara, California: Nob Hill Publishing, LLC, 2017, vol. 1, 2020942771. [Online]. Available: http://www.nobhillpublishing.com/mpc-paperback/index-mpc.html

[3] B. Kouvaritakis and M. Cannon, *Model Predictive Control*, ser. Advanced Textbooks in Control and Signal Processing. Cham: Springer International Publishing, 2016. [Online]. Available: http://link.springer.com/10.1007/978-3-319-24853-0

[4] D. Marruedo, T. Alamo, and E. Camacho, "Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties," in *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.*, vol. 4. Las Vegas, NV, USA: IEEE, 2002, pp. 4619–4624. [Online]. Available: http://ieeexplore.ieee.org/document/1185106/

[5] D. Limon, J. Bravo, T. Alamo, and E. Camacho, "Robust MPC of Constrained Nonlinear Systems Based on Interval Arithmetic," *Control Theory and Applications, IEE Proceedings* -, vol. 152, pp. 325–332, Jun. 2005. [Online]. Available: https://www.researchgate.net/publication/3352729_Robust_MPC_of_constrained_nonlinear_systems_based_on_interval_arithmetic

[6] M. E. Villanueva, R. Quirynen, M. Diehl, B. Chachuat, and B. Houska, "Robust MPC via min–max differential inequalities," *Automatica*, vol. 77, pp. 311–321, Mar. 2017. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0005109816304575

[7] B. T. Lopez, J.-J. E. Slotine, and J. P. How, "Dynamic Tube MPC for Nonlinear Systems," Jul. 2019, arXiv:1907.06553 [cs, eess]. [Online]. Available: http://arxiv.org/abs/1907.06553

[8] A. Sasfi, M. N. Zeilinger, and J. Köhler, "Robust adaptive MPC using control contraction metrics," *Automatica*, vol. 155, p. 111169, Sep. 2023, accepted: 2023-07-18T14:12:14Z Publisher: Elsevier. [Online]. Available: https://www.research-collection.ethz.ch/handle/20.500.11850/622228

[9] S. Singh, B. Landry, A. Majumdar, J.-J. Slotine, and M. Pavone, "Robust Feedback Motion Planning via Contraction Theory," *The International Journal of Robotics Research*, vol. 42, no. 9, pp. 655–688, Aug. 2023, publisher: SAGE Publications Ltd STM. [Online]. Available: https://doi.org/10.1177/02783649231186165

[10] P. Zhao, A. Lakshmanan, K. Ackerman, A. Gahlawat, M. Pavone, and N. Hovakimyan, "Tube-Certified Trajectory Tracking for Nonlinear Systems With Robust Control Contraction Metrics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5528–5535, Apr. 2022, arXiv:2109.04453 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2109.04453

[11] A. P. Leeman, J. Köhler, A. Zanelli, S. Bennani, and M. N. Zeilinger, "Robust Nonlinear Optimal Control via System Level Synthesis," Feb. 2024, arXiv:2301.04943 [cs, eess, math]. [Online]. Available: http://arxiv.org/abs/2301.04943

[12] A. P. Leeman, J. Sieber, S. Bennani, and M. N. Zeilinger, "Robust Optimal Control for Nonlinear Systems with Parametric Uncertainties via System Level Synthesis," Sep. 2023, arXiv:2304.00752 [cs, eess, math]. [Online]. Available: http://arxiv.org/abs/2304.00752

[13] A. Zanelli, J. Frey, F. Messerer, and M. Diehl, "Zero-Order Robust Nonlinear Model Predictive Control with Ellipsoidal Uncertainty Sets," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 50–57, 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2405896321012982

[14] F. Messerer and M. Diehl, "An Efficient Algorithm for Tube-based Robust Nonlinear Optimal Control with Optimal Linear Feedback," in *2021 60th IEEE Conference on Decision and Control (CDC)*. Austin, TX, USA: IEEE, Dec. 2021, pp. 6714–6721. [Online]. Available: https://ieeexplore.ieee.org/document/9683712/

[15] S. Chen, V. M. Preciado, M. Morari, and N. Matni, "Robust Model Predictive Control with Polytopic Model Uncertainty through System Level Synthesis," Sep. 2023, arXiv:2203.11375 [cs, eess, math]. [Online]. Available: http://arxiv.org/abs/2203.11375

[16] J. Anderson, J. C. Doyle, S. Low, and N. Matni, "System Level Synthesis," Apr. 2019, arXiv:1904.01634 [cs, math]. [Online]. Available: http://arxiv.org/abs/1904.01634

[17] I. R. Manchester and J.-J. E. Slotine, "Control Contraction Metrics: Convex and Intrinsic Criteria for Nonlinear Feedback Design," Feb. 2017, arXiv:1503.03144 [cs, math]. [Online]. Available: http://arxiv.org/abs/1503.03144

[18] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Mar. 2006. [Online]. Available: http://link.springer.com/10.1007/s10107-004-0559-y

[19] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi: a software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, Mar. 2019. [Online]. Available: http://link.springer.com/10.1007/s12532-018-0139-4

[20] E. D. Andersen and K. D. Andersen, "The Mosek Interior Point Optimizer for Linear Programming: An Implementation of the Homogeneous Algorithm," in *High Performance Optimization*, P. M. Pardalos, D. Hearn, H. Frenk, K. Roos, T. Terlaky, and S. Zhang, Eds. Boston, MA: Springer US, 2000, vol. 33, pp. 197–232, series Title: Applied Optimization. [Online]. Available: http://link.springer.com/10.1007/978-1-4757-3216-0_8

[21] J. Lofberg, "YALMIP : a toolbox for modeling and optimization in MATLAB," in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*. Taipei, Taiwan: IEEE, 2004, pp. 284–289. [Online]. Available: http://ieeexplore.ieee.org/document/1393890/

[22] A. P. Leeman, J. Köhler, F. Messerer, A. Lahr, M. Diehl, and M. N. Zeilinger, "Fast System Level Synthesis: Robust Model Predictive Control using Riccati Recursions," Jan. 2024, arXiv:2401.13762 [cs, eess, math]. [Online]. Available: http://arxiv.org/abs/2401.13762

[23] Z. Yuan, A. W. Hall, S. Zhou, L. Brunke, M. Greeff, J. Panerati, and A. P. Schoellig, "safe-control-gym: a Unified Benchmark Suite for Safe Learning-based Control and Reinforcement Learning in Robotics," Jul. 2022, arXiv:2109.06325 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2109.06325

ETH zürich

Institute for Dynamic Systems and Control

Prof. Dr. R. D'Andrea, Prof. Dr. E. Frazzoli, Prof. Dr. C. Onder, Prof. Dr. M. Zeilinger

Title of work:

A Comparison on Robust MPC Methods for Nonlinear Systems

Thesis type and date:

Semester Project, September 3, 2024

Supervision:

Prof. Dr. Melanie Zeilinger
Dr. Johannes Köhler
Antoine Leeman

Student:

Name: Alexander Casper Erdin
E-mail: aerdin@student.ethz.ch
Legi-Nr.: 17-671-413
Semester: FS 24

Statement regarding plagiarism:

By signing this statement, I affirm that I have read and signed the Declaration of Originality, independently produced this paper, and adhered to the general practice of source citation in this subject-area.

Declaration of Originality:

https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/declaration-originality.pdf

Zurich, 3.9.2024: _____