ROBUST UPDATED MPC SCHEMES

Von der Fakultät für Mathematik, Physik und Informatik
der Universität Bayreuth

zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigte Abhandlung

vorgelegt von

M.Sc. Vryan Gil Palma

aus Caloocan, Philippinen

1. Gutachter: Prof. Dr. Lars Grüne
2. Gutachter: Prof. Dr. Matthias Gerdts

Tag der Einreichung: 23. Februar 2015
Tag des Kolloquiums: 21. Mai 2015

This research project is supported by the European Union under the 7th Framework Programme FP7-PEOPLE-2010-ITN, Grant agreement number 264735-SADCO.

Abstract

We consider model predictive control (MPC) approaches to approximate the solution of infinite horizon optimal control problems (OCPs) for perturbed nonlinear discrete time systems. MPC provides an algorithmic synthesis of an approximately optimal feedback law by iteratively solving finite horizon OCPs. The optimization problem to be solved at each time step results in a high computational expense and computational latency. As computationally costly MPC controllers may demand implementation on highly powerful computing systems to meet real-time requirements, we address the challenge of developing algorithms that are less computationally demanding without sacrificing the control performance to cater to systems with fast dynamics.

In using the *multistep MPC* strategy, we reduce the number of optimizations to be performed hence considerably lowering the computational load. However, this approach comes with the disadvantage of reduced robustness of the closed-loop solution against perturbations. We introduce the *updated multistep MPC* where an update is performed to the multistep MPC based on re-optimizations on shrinking horizons giving a straightforward approach to provide a coping mechanism to counteract the perturbations. Robust performance improvements due to re-optimization are rigorously quantified. This is achieved by analyzing the *open-loop control strategy* and the *shrinking horizon strategy* on finite horizon OCPs for systems under perturbations where potential performance improvement brought about by the re-optimization is quantified. This analysis of potential benefits extends to the setting where the moving horizon MPC strategy is used for the infinite horizon setting.

Lastly, we consider the *sensitivity-based multistep MPC* which is a particular MPC variant that allows further savings in computational load by using sensitivity analysis. The sensitivities used to update the multistep MPC can be computed efficiently by exploiting the matrix structures resulting from the MPC problem formulation. For this scheme, we show that the sensitivity-based control is a linear approximation of the re-optimization-based control and therefore, the analysis on the performance and stability properties of the updated multistep MPC can be carried over to the sensitivity-based multistep MPC.

We compare the MPC schemes and confirm our theoretical results through numerical examples. We also examine the control performance and computing complexity requirements of the schemes and analyze their potential and suitability to be implemented on embedded systems with limited computing power.

Key words: optimal control, model predictive control, robustness against perturbations, nonlinear programming, sensitivity analysis

Zusammenfassung

Wir untersuchen Modellprädiktive Regelungsalgorithmen (MPC Algorithmen) zur Approximation von Optimalsteuerungsproblemen (OCPs) auf unendlichem Zeithorizont für gestörte nichtlineare diskrete dynamische Systeme. MPC liefert ein approximativ optimales Feedback durch die iterative Lösung von OCPs auf endlichem Zeithorizont. Das in jedem Zeitschritt zu lösende Optimierungsproblem ist sehr rechenaufwändig und führt zu Verzögerungen. Der hohe Rechenaufwand von MPC Algorithmen erfordert große Rechenkapazitäten um Echtzeitanwendungen gerecht zu werden. Diese Arbeit konzentriert sich auf die Entwicklung von Algorithmen mit reduziertem Rechenaufwand, jedoch ohne die Güte der Regelung zu beeinträchtigen, um damit die Anwendbarkeit auf Systeme mit schneller Dynamik zu gewährleisten.

Durch die Anwendung von *Mehrschritt-MPC* reduzieren wir die Anzahl der zu lösenden Optimierungsprobleme und damit den Rechenaufwand signifikant. Allerdings verschlechtert dieser Ansatz die Robustheit des geschlossenen Regelkreises bezüglich Störungen. Wir präsentieren einen *Aktualisierten-Mehrschritt-MPC* Algorithmus, bei dem im Gegensatz zum Mehrschritt-MPC in jedem Schritt das zugrundeliegende Optimierungsproblem auf einem verkürzten Horizont gelöst wird. Dies liefert einen direkten Ansatz, dem Einfluss von Störungen entgegen zu wirken. Die Robustheit und Performanceverbesserung des Verfahrens dank der Reoptimierung wird mathematisch bewiesen. Die Analyse erfolgt durch den Vergleich der *„Open-loop“ Strategie* und der *schrumpfenden Horizont Strategie*, angewandt auf OCPs mit endlichem Zeithorizont und gestörten Systemen. Hierbei wird die mögliche Performancesteigerung durch die Reoptimierung herausgestellt und quantifiziert. Die Analyse und die möglichen Vorteile übertragen sich dann auf MPC Verfahren auf unendlichem Zeithorizont.

Der *Sensitivitätsbasierte-Mehrschritt-MPC* Algorithmus liefert eine weitere Reduzierung des Rechenaufwands. Die Sensitivitäten, die zum Update des Mehrschritt-MPC notwendig sind, können effizient durch Ausnutzung der Matrixstruktur der MPC Formulierung berechnet werden. Durch eine Analyse der Sensitivitäten der zugrundeliegenden Dynamik zeigen wir, dass der sensitivitätsbasierte Regler eine lineare Approximation des reoptimierungsbasierten Reglers darstellt. Wir weisen nach, dass sich daher die Stabilitäts- und Performanceeigenschaften des Aktualisierten-Mehrschritt-MPC auf den Sensitivitätsbasierten-Mehrschritt-MPC übertragen lassen.

Die unterschiedlichen MPC Algorithmen werden anhand von Beispielen miteinander verglichen und die theoretischen Resultate dadurch verifiziert. Zusätzlich untersuchen wir die Performance der Verfahren sowie den notwendigen Rechenaufwand zur Umsetzung der Algorithmen. Zur Analyse des Rechenaufwands wird insbesondere ihr Potenzial für die Anwendung für eingebettete Systeme mit beschränkter Rechenleistung untersucht.

Stichwörter: Optimalsteuerung, Modellprädiktive Regelung, Robustheit gegen Störungen, nichtlineare Optimierung, Sensitivitätsanalyse

Contents

Acronyms

KKT	Karush-Kuhn-Tucker 66
LICQ	linear independence constraint qualification 65
LP	linear programming 63
MPC	model predictive control 1
NLP	nonlinear programming 62
OCP	optimal control problem 1
QP	quadratic programming 63
SBM	sensitivity-based m-step 80
SOSC	Second-order sufficient conditions 66
SQP	sequential quadratic programming 69

Introduction

Model predictive control

The recent decades have seen a rapid development in **model predictive control (MPC)** and its various aspects. It has garnered increased attention as it has proven to be an important tool in the control of nonlinear systems in modern technological applications. MPC is a feedback control design strategy based on the solution, at every sampling instant, of an optimal control problem (OCP) over a chosen horizon. In this optimization-based control technique, an OCP is solved at each time step to determine a sequence of input moves that controls the current and future behavior of a physical system in an optimal manner. Typically for an MPC scheme, after applying the first element of the optimal sequence of inputs, the fixed optimization horizon is shifted by one sampling time into the future and the procedure is repeated.

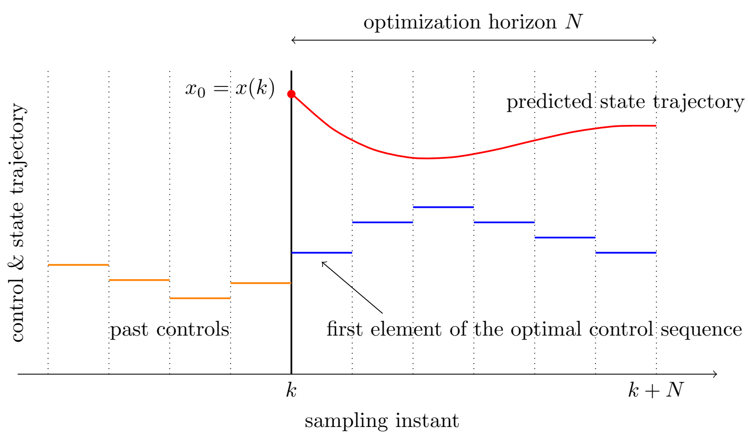

Figure 0.1: The MPC scheme where $x(k)$ is used as the initial value of the OCP solved at instant k. Piecewise constant control is used for the discretized system.

Conditions needed so that the MPC feedback law asymptotically stabilizes the system have been well-understood in the literature. A clever approach in ensuring stability of MPC schemes is based on imposing stabilizing terminal conditions, see, e.g., survey paper Mayne et al. [47] and monographs Rawlings and Mayne [57] and Grüne and Pannek [36, Chapter 5]. Such terminal conditions, however, are not necessary conditions for achieving stability. In this thesis, we consider MPC schemes without terminal conditions. Stability for MPC schemes without terminal conditions are well-studied and developed in several works, e.g., Alamir and Bornard [1], Primbs and Nevistić [54], Jadbabaie and Hauser [39], Grimm et al. [31], Tuna et al. [65], Grüne [32], Grüne et al. [37] and Altmüller et al. [3].

Perturbed systems and robust stability

The explicit model of the system used to predict its future behavior is subject to modeling uncertainty and noise. In addition, the actual system itself is subject to external disturbances. Due to these perturbations, the predicted states and the measured states, obtained once the computed control actions are applied, typically deviate from each other.

Owing to its feedback nature, MPC exhibits certain inherent robustness properties for the perturbed setting, despite performing optimization in each iteration only for the nominal model, i.e., without taking into account perturbations. **Robust stability** refers to the capability of the system to maintain stability and performance specifications for a specified range of uncertainty (see survey paper Bemporad and Morari [7] on robust MPC).

Real-time optimization

When the current system state is measured, a control strategy that minimizes a certain cost must be computed instantaneously, i.e., online, while the system is operating and evolving. Solving an OCP to determine such an optimal control strategy can be very computationally intensive since this usually includes an iterative scheme solved until a convergence criterion is fulfilled. The high computational expense results in **computational latency** or **delay**. As depicted in Figure 0.2, where at time k the state is measured as $x(k)$. Suppose solving the OCP takes δ units of time. This means we obtain the optimal control only at time $k + \delta$. However, in the case of a considerable delay δ, the system has already evolved by time $k + \delta$ where the behavior of the system may have by then changed much.

For this reason, MPC used to be a feasible option only for systems with slow dynamics where there is sufficient time for solving the optimization problem between sampling instants (see history and overview paper Qin and Badgwell [55, 56] on industrial MPC developments).

Hardware implementation

Computationally costly MPC algorithms used to be implemented using highly powerful computing systems (e.g., servers, desktops, industrial PCs) in order to meet real-time requirements.

Along with the development of sophisticated algorithms, digital electronics have advanced during the last years. Nowadays, modern embedded systems feature high numerical computing power (e.g., 1GFlops for each core on an ARM Cortex-A9) with a very low power consumption (<1Watt) and cost (€). This allows the implementation of computationally heavy control schemes for fast dynamical systems at affordable cost (see, e.g., Jerez et al. [40] or Kerrigan [42]). This also allows the feasibility of high performance control techniques to new application

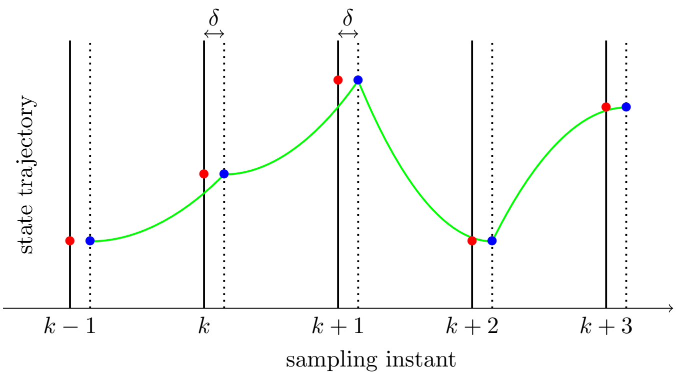

Figure 0.2: The computational delay δ due to the online optimization in implementating MPC.

domains demanding tight real-time requirements.

Yet for a fixed price and size of an embedded hardware which determine its capability and limitation, a researcher-designer still faces a trade-off decision between low computing cost and high performance.

Existing real-time capable MPC schemes in the literature

Many invaluable advances on MPC are geared towards the suitability of MPC for time-critical systems wherein sampling frequencies are higher. The work Binder et al. [12] enumerates primary considerations in designing fast and real-time capable MPC. One may need to determine whether or not the OCP can be solved within a time requirement known a-priori. To this end, one may compute offline or in advance certain quantities not necessarily needed to be computed online to reduce the delay. One may also use suitable approximations for the feedback or exploit similarities in the structures of OCPs being solved.

We briefly enumerate some of the studies from the large body of work on these MPC developments that serve as motivation to the study we present in this thesis.

The work Diehl [18] along with related works, e.g., [22, 21, 20, 19] presents a *real-time iteration technique* based on a direct method (i.e., first discretize, then optimize) within a multiple shooting discretization (see Bock and Plitt [14]) and sequential quadratic programming (see optimization textbooks, e.g., [49, 15, 11]) framework. Initializing the current OCP by the state and control obtained from the previous OCP, taking advantage of the fact that the OCPs are related by a parameter that enters the problems linearly, results in the so-called *initial value embedding* of the OCP into the manifold of perturbed OCPs. This allows approximating the OCP in advance without the knowledge of the actual initial value. In addition, fast solution of the OCP is provided by not iterating the sequential quadratic programming to convergence. Furthermore, the iteration scheme involves a *preparation phase* – a phase where functions and derivatives that do not require the information of the current state are already prepared so the moment the current state is revealed, the remaining computation needed to be undertaken becomes minimal – and a considerably shorter *feedback phase* allowing to reduce the delay.

Bock et al. [13] proposes alternatives to the mentioned preparation phase by using *multilevel updates* to the components of the quadratic programming.

The works Büskens and Maurer [17], Maurer and Pesch [46], Pesch [53], Büskens and Gerdts [16] and Gerdts [25] take advantage of neighboring optimal solutions based on parametric sensitivities. The analysis on the impact of a change in a design parameters allows for updates on the control used in open-loop. Such updates are then used in the context of MPC in Würth et al. [68], Zavala and Biegler [70] and Yang and Biegler [69] to address the demands of real-time optimization. Since the evolution of the system is affected by disturbances and uncertainties, corrective updates of the nominal control are applied assuming the mentioned perturbations are small enough.

The mentioned works [70, 69] also perform computation in the background leaving the remaining tasks to be computed online inexpensive. It exploits the predictive capabilities of the dynamic model to predict the future state of the plant and solve a predicted problem in background between sampling times. Once the current state becomes available at the next sampling time, the controller responds to uncertainties through the online corrective update of the predicted solution. The approach uses simultaneous collocation (see Biegler [10]) and interior point solver (see textbooks on optimization, e.g., [49, 15, 11]).

Another straightforward approach to cut back on computation expense is by using the multistep MPC strategy (refer, e.g., to already mentioned works [32, 37]) the computational load can be lowered considerably by reducing the number of optimizations performed. For time instants which are not multiple of m, the control is immediately available.

Robust updated MPC schemes

Motivated by the prevailing themes from the discussion above, namely, maintaining robustness and reducing computational load, we propose and analyze in this thesis MPC variants fulfilling these objectives and present rigorous proofs on the robust stability and performance of these schemes.

For a system subject to perturbations, the multistep feedback does not allow the controller to respond, for an extended period of time, against the deviation of the real state to the predicted state. Hence, multistep feedback laws are in general considerably less robust against perturbations as opposed to the standard MPC scheme. To accomplish the goals of robustifying the scheme while keeping the computational cost low, we consider and investigate updating strategies on the multistep scheme.

The first approach is the **updated multistep MPC** which uses re-optimizations on shrinking horizons as a straightforward approach to provide a coping mechanism to counteract the perturbations. Our analysis of this scheme builds upon the study of finite horizon OCPs for systems under perturbations wherein we compare the so-called *nominal control strategy* and the *shrinking horizon strategy*. Potential performance improvement brought about by the re-optimization is quantified using certain *moduli of continuity* of value functions. Switching the attention back to the original problem, i.e., the infinite horizon OCP, we use obtained expressions depending on moduli of continuity to establish improved robust stability and performance of the updated multistep MPC compared to the non-updated one.

Conceptually, the idea of the shrinking horizon strategy on finite horizon OCPs has strong similarities to sensitivity-based techniques for open-loop control used in order to cope with perturbations in the aforementioned works [17, 46, 53, 16, 25]. In the sensitivity-based techniques for open-loop control, instead of a full re-optimization, only an approximate update of the optimal control based on updated state information is performed. This idea can also be used in moving horizon MPC in order to reduce the number of full optimizations to fulfill the requirements of real-time optimization. We call the second approach the **sensitivity-based multistep MPC**. The results on the stability and performance analysis of the updated MPC can be extended to this case owing to the fact that the re-optimizations are replaced by sensitivity-based updates viewing the latter approach as an approximation to the former.

Various other updated MPC schemes exist in the literature aside from those whose updates are derived from sensitivity analysis as in [68, 70, 69]. Our approach has similarities to the abstract updates referred to in Pannek et al. [52] in the sense that updates, in the setting of MPC without stabilizing terminal conditions, are applied in order to cope with the nominal and real model disparity. However, while in [52] the main result states that reasonable updates do not negatively affect stability and performance, our main result in this thesis shows that the shrinking horizon updates of the updated multistep MPC and the particular updates of the sensitivity-based multistep MPC both do indeed allow for improved stability and performance estimates compared to non-updated MPC.

In implementing these proposed MPC variants in real-time, as implemented in the literature mentioned above, one may take advantage of the separation principle among the *online* (quantities computed immediately when the state measurement becomes available), *background* (quantities computed shortly before the state measurement becomes available) and *offline* (quantities computed even before the process starts) computations. This, however, is beyond the scope of the application we present in this thesis and will be left for future direction.

Contribution and overview of the thesis

The thesis deals with the main problem of solving infinite horizon OCPs for perturbed nonlinear systems by MPC. MPC provides an algorithmic synthesis of an approximately optimal feedback law by iteratively solving finite horizon OCPs. This work is organized as follows.

Chapter 1 defines the setting and gives basic tools needed for the MPC analysis. We introduce three MPC algorithms that serve as fundamental algorithms for our analysis, namely, the *standard MPC*, the *multistep MPC* and the *updated multistep MPC*.

We provide in Chapter 2 some existing results on the nominal stability of MPC schemes without terminal conditions. We then introduce the perturbed system setting and present conditions that an MPC variant needs to satisfy to yield robust stability. Since for perturbed systems asymptotic stability is often too strong a property to expect, in this thesis, we develop instead our results using the notion of practical asymptotic stability.

Chapter 3 brings a focus on finite horizon problems. We compare three different settings: the *open-loop controller for the nominal system*, the *nominal open-loop controller applied to the perturbed system* and the *shrinking horizon controller*, i.e., the controller for which at each time step wherein perturbation is experienced, we perform re-optimization. We conduct an analysis on the benefits of re-optimization under perturbations by comparing the three settings and discussing concepts of controllability and stability.

♣ Despite the long existence of these methods, we are not aware of rigorous results which quantify the benefit of the re-optimization in terms of the objective of the optimal control problem in the presence of persisting perturbations. While many papers address feasibility issues, results on the performance of the controller and its potential improvement due to re-optimization are to the best of our knowledge missing up to now. This gap is what we intend to fill in this chapter. A preliminary version of the results we provide in Chapter 3 is published in Grüne and Palma [34].

Chapter 4 transitions back to the infinite horizon problem whose solution is approximated by MPC. We analyze the *nominal multistep MPC*, the *perturbed multistep MPC* and the *updated multistep MPC* and use corresponding properties from the three settings studied in Chapter 3.

♣ One of the key challenges when passing from finite to infinite horizon is that typically asymptotic stability of the approximately optimal solution must be established before we can talk about approximately optimal performance. Rigorously quantifiable robust asymptotic stability and performance estimates are presented in this chapter. As a main result, this chapter shows that the shrinking horizon updates of the updated multistep MPC results in improved stability and performance estimates in comparison to the non-updated MPC. A preliminary version of the results we provide in Chapter 4 is published in Grüne and Palma [33].

Although one can observe that the updated multistep MPC is already computationally less expensive than the standard MPC with re-optimization in full horizon, aiming to further cut down computational cost while maintaining robustness, we propose a scheme approximating the updated multistep scheme. To this end, we first review in Chapter 5 prerequisite results on nonlinear programming and sensitivity analysis.

In Chapter 6, we introduce the *sensitivity-based multistep MPC* which is an MPC variant that provides corrective updates to the multistep MPC computed using the magnitude of the perturbations, i.e., the deviation between the predicted and measured current states, and the sensitivities of the solution of the OCP with respect to current state acting as a perturbed parameter. The idea of this scheme which allows further reduction in terms of computational load is published in Palma and Grüne [50].

♣ Compared to existing MPC approaches that use sensitivity analysis, the sensitivity-based scheme we consider in this thesis uses multistep control with corrective updates yielding the resulting control to be a linear approximation of the control obtained from re-optimization as in the updated MPC strategy. As a consequence, we show that the performance and stability of the updated MPC lends itself to this new variant up to some uncertainty range. Although the updated multistep MPC still gives the best performance, the sensitivity-based multistep MPC, however, has better robustness properties than the nonupdated.

Implementation examples and comparisons of the MPC variants that we tackle in the thesis are presented in Section 4.4 and in Chapter 7 validating our theoretical results.

♣ An implementation example in Chapter 7 shows that the sensitivity-based multistep MPC fulfills both control performance and low computing complexity requirements and investigates its potential for controller design on embedded computing systems. A preliminary version of this study is published in Palma, Suardi and Kerrigan [51].

1 MPC setting and preliminaries

1.1 Setting

We consider the nonlinear discrete time control system

$$x(k+1) = f(x(k), u(k)), \;\; k \in \mathbb{N} \tag{1.1}$$

where x is the state and u is the control value. Let the normed vector spaces X and U be state and control spaces, respectively. For a given state constraint set $\mathbb{X}$ and control constraint sets $\mathbb{U}(x)$, $x \in \mathbb{X}$, we require $x \in \mathbb{X} \subseteq X$ and $u \in \mathbb{U}(x) \subseteq U$. The notation $x_u(\cdot, x_0)$ (or briefly $x_u(\cdot)$) denotes the state trajectory when the initial state x_0 is driven by the control sequence $u(\cdot)$. We refer to (1.1) as the nominal model. In Section 2.2, we extend this model by incorporating perturbations.

A time-dependent feedback law $\mu : \mathbb{X} \times \mathbb{N} \to \mathbb{U}$ yields the feedback controlled system

$$x(k+1) = f(x(k), \mu(x(\tilde{k}), k)) \tag{1.2}$$

Here, the next state at time instant $k+1$ depends on the current state at time k and the feedback value $\mu(x(\tilde{k}), k)$, which enters the system as a control value. The feedback value, in turn, depends on the system state $x(\tilde{k})$ at a time $\tilde{k} = \tilde{k}(k) \leq k$ which may be strictly smaller than k. We refer to (1.2) as the closed-loop system.

MPC is motivated by the following problem. We aim to find a feedback law μ that approximately solves the infinite horizon OCP

$$\min_{u(\cdot) \in \mathbb{U}^\infty(x_0)} J_\infty\left(x_0, u(\cdot)\right) \tag*{$\mathcal{P}_\infty(x_0)$}$$

where the objective function is given by

$$J_\infty\left(x_0, u(\cdot)\right) := \sum_{k=0}^{\infty} \ell\left(x_u(k, x_0), u(k)\right)$$

which is an infinite sum of nonnegative stage costs $\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_0^+$ along the trajectory with x_0 as the initial value steered by the control sequence $u(\cdot) \in \mathbb{U}^\infty(x_0)$. This type of objective is often related to feedback stabilization problems which will be detailed in Section 2.1. The objective is minimized over all infinite admissible control sequences, i.e., all control sequences $u(\cdot)$ satisfying

$$\mathbb{U}^\infty(x_0) := \left\{ u(\cdot) \in U^\infty \;\middle|\; \begin{array}{c} x_u(k+1, x_0) \in \mathbb{X} \text{ and} \\ u(k) \in \mathbb{U}(x_u(k, x_0)) \text{ for all } k \in \mathbb{N}_0 \end{array} \right\}$$

where U^∞ denotes the set of all infinite admissible control sequences. Its optimal value function is given by

$$V_\infty(x_0) := \inf_{u(\cdot) \in \mathbb{U}^\infty(x_0)} J_\infty(x_0, u)$$

and the infinite horizon closed-loop performance of a given time-dependent feedback μ is given by

$$J_\infty^{\mathrm{cl}}(x_0, \mu) := \sum_{k=0}^{\infty} \ell\left(x_\mu(k, x_0), \mu(x_\mu(\tilde{k}, x_0), k)\right) \tag{1.3}$$

which is the infinite sum of costs along the trajectory driven by the feedback law. Given an initial state, we would like to solve the infinite horizon optimal control problem and obtain an optimal control in feedback form, i.e., to find a feedback μ with $J_\infty^{\mathrm{cl}}(x_0, \mu) = V_\infty(x_0)$. In the general nonlinear setting, however, this problem is often computationally intractable, so we circumvent it by considering the finite horizon minimization problem

$$\min_{u(\cdot) \in \mathbb{U}^N(x_0)} J_N(x_0, u(\cdot)) \tag{$\mathcal{P}_N(x_0)$}$$

for the synthesis of the feedback law μ to be discussed in Section 1.3. The objective function is given by

$$J_N(x_0, u(\cdot)) := \sum_{k=0}^{N-1} \ell(x_u(k, x_0), u(k)) \tag{1.4}$$

representing a cost associated with an initial state x_0, a control sequence $u(\cdot)$ and optimization horizon N. The minimization is performed over all control sequences $u(\cdot) \in \mathbb{U}^N(x_0)$ where

$$\mathbb{U}^N(x_0) := \left\{ u(\cdot) \in U^N \;\middle|\; \begin{array}{c} x_u(k+1, x_0) \in \mathbb{X} \text{ and} \\ u(k) \in \mathbb{U}(x_u(k, x_0)) \text{ for all } k = 0, \ldots, N-1 \end{array} \right\}$$

where U^N denotes the set of all finite admissible control sequences with N elements.

We define the optimal value function associated with the initial state value x_0 by

$$V_N(x_0) := \inf_{u(\cdot) \in \mathbb{U}^N(x_0)} J_N(x_0, u(\cdot))$$

In this work, we assume there exists a (not necesssarily unique) control sequence $u^*(\cdot) \in \mathbb{U}^N(x_0)$ satisfying $V_N(x_0) = J_N(x_0, u^*(\cdot))$, which is called the optimal control sequence. Alternatively, statements could be formulated using ε-optimal control sequences, at the expense of a considerably more technical presentation.

1.2 Basic definitions and theorems

An important concept that we will be using in our analysis is the dynamic programming principle (introduced in Bellman [5], see also, e.g., [9, 8]). It relates the optimal value functions of OCPs of different optimization horizon length for different points along a trajectory.

Theorem 1.2.1. *(Dynamic programming principle) Let x_0 be an initial state value. Let $u^*(0), u^*(1), \dots, u^*(N-1)$ be an optimal control sequence for $\mathcal{P}_N(x_0)$ and $x_{u^*}(0) = x_0, x_{u^*}(1), \dots, x_{u^*}(N)$ denote the corresponding optimal state trajectory. Then for any i, $i = 0, 1, \dots, N-1$, the control sequence $u^*(i), u^*(i+1), \dots, u^*(N-1)$ is an optimal control sequence for $\mathcal{P}_{N-i}(x_{u^*}(i))$.*

Next, we define the following classes of comparison functions.

Definition 1.2.2.

i. A function $\rho : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is a $\mathcal{K}$-function if it is continuous, strictly increasing and $\rho(0) = 0$.

ii. ρ is a $\mathcal{K}_\infty$-function if it is a $\mathcal{K}$-function that is unbounded.

iii. A function $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is a $\mathcal{KL}$-function if it is continuous and if, for each $r \geq 0$, $\beta(r, \cdot)$ is decreasing and satisfies $\lim_{t\to\infty} \beta(r,t) = 0$, and, for each $t \geq 0$, $\beta(\cdot, t) \in \mathcal{K}$.

iv. A function $\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \to \mathbb{R}_0^+$ is a $\mathcal{KL}_0$-function if it is continuous and if, for each $r \geq 0$, $\beta(r, \cdot)$ is decreasing and satisfies $\lim_{t\to\infty} \beta(r,t) = 0$, and, for each $t \geq 0$, either $\beta(\cdot, t) \in \mathcal{K}$ or $\beta(\cdot, t) \equiv 0$.

We also consider the following notion of continuity.

Definition 1.2.3. Consider normed vector spaces Z and Y, a set $A \subset Z$ and an arbitrary set W

i. A function $\phi : Z \to Y$ is said to be uniformly continuous on A if there exists a $\mathcal{K}$-function ω such that for all $z_1, z_2 \in A$

$$\|\phi(z_1) - \phi(z_2)\| \leq \omega\left(\|z_1 - z_2\|\right)$$

ii. A function $\phi : Z \times W \to Y$ is said to be uniformly continuous on A uniformly in $v \in W$ if there exists a function $\omega \in \mathcal{K}$ such that for all $z_1, z_2 \in A$ and all $v \in W$

$$\|\phi(z_1, v) - \phi(z_2, v)\| \leq \omega\left(\|z_1 - z_2\|\right)$$

The function ω is called the modulus of continuity.

Similar to that found in [57, Appendix C], the following theorem gives sufficient conditions for which the optimal value function is a uniformly continuous function without state constraints, i.e., $X = \mathbb{X} = \mathbb{R}^n$.

Theorem 1.2.4. *(Uniform continuity of $V_N(\cdot)$) Consider $X = \mathbb{X}$ and $\mathbb{U}(x) \equiv \mathbb{U}$ and suppose that $J_N : \mathbb{X} \times \mathbb{U}^N \to \mathbb{R}_0^+$ is uniformly continuous on a set $S \subset \mathbb{X}$ uniformly in $u(\cdot) \in \mathbb{U}^N$. Then $V_N(\cdot)$ is uniformly continuous on S.*

Proof. From the assumptions, there exists $\omega_{J_N} \in \mathcal{K}$ such that

$$\|J_N(x_1, u(\cdot)) - J_N(x_2, u(\cdot))\| \leq \omega_{J_N}(\|x_1 - x_2\|) \tag{1.5}$$

for all $x_1, x_2 \in S$ and all $u(\cdot) \in \mathbb{U}^N$. Since (1.5) holds for any choice of $u(\cdot) \in \mathbb{U}^N$, let $\varepsilon > 0$ and suppose $u^2_\varepsilon(\cdot)$ is an ε-optimal control for $\mathcal{P}_N(x_2)$. This implies

$$\begin{aligned} V_N(x_1) - V_N(x_2) &\leq J_N(x_1, u^2_\varepsilon(\cdot)) - V_N(x_2) \\ &\leq J_N(x_1, u^2_\varepsilon(\cdot)) - J_N(x_2, u^2_\varepsilon(\cdot)) + \varepsilon \\ &\leq \omega_{J_N}(\|x_1 - x_2\|) + \varepsilon. \end{aligned}$$

Likewise, for an ε-optimal control $u^1_\varepsilon(\cdot)$ we have

$$\begin{aligned} V_N(x_2) - V_N(x_1) &\leq J_N(x_2, u^1_\varepsilon(\cdot)) - V_N(x_1) \\ &\leq J_N(x_2, u^1_\varepsilon(\cdot)) - J_N(x_1, u^1_\varepsilon(\cdot)) + \varepsilon \\ &\leq \omega_{J_N}(\|x_2 - x_1\|) + \varepsilon. \end{aligned}$$

Since $\varepsilon > 0$ is arbitrary,

$$\|V_N(x_1) - V_N(x_2)\| \leq \omega_{J_N}\left(\|x_1 - x_2\|\right)$$

holds for all $x_1, x_2 \in S$ which implies that $V_N(\cdot)$ is uniformly continuous on S.

Particularly, for all $x_1, x_2 \in S$

$$\|V_N(x_1) - V_N(x_2)\| \leq \omega_{V_N}\left(\|x_1 - x_2\|\right)$$

with $\omega_{V_N} \leq \omega_{J_N}$. □

In the presence of state constraints, conditions under which a similar result holds become more technical, see, e.g., [36, Proposition 8.40]. We also note that the modulus of continuity ω_{V_N} represents the sensitivity of the optimal value function to changes in the parameter x of the problem $\mathcal{P}_N(x)$. The proof of the theorem shows that ω_{V_N} is less than or equal to ω_{J_N}, hence we can expect that $\|V_N(x_1) - V_N(x_2)\|$ cannot be that much larger than $\|J_N(x_1, u(\cdot)) - J_N(x_2, u(\cdot))\|$ and will typically be smaller. We will further investigate this relation in Chapter 3.

1.3 MPC algorithms

In this section, we explain how the finite horizon OCP $\mathcal{P}_N(x_0)$ can be used in order to construct an approximately optimal feedback law for the infinite horizon problem $\mathcal{P}_\infty(x_0)$.

The 'usual' or 'standard' MPC algorithm proceeds iteratively as follows.

Algorithm 1.3.1. (Standard MPC)

(1) Measure the state $x(k) \in \mathbb{X}$ of the system at time instant k

(2) Set $x_0 := x(k)$ and solve the finite horizon problem $\mathcal{P}_N(x_0)$. Let $u^*(\cdot) \in \mathbb{U}^N(x_0)$ denote the optimal control sequence and define the MPC feedback law

$$\mu_N(x(k), k) := u^*(0)$$

(3) Apply the control value $\mu_N(x(k), k)$ to the system, set $k := k + 1$ and go to (1)

This iteration, also known as a receding horizon strategy, gives rise to a non-time-dependent feedback μ_N which — under appropriate conditions, see Section 2.1 — approximately solves the infinite horizon problem. It generates a nominal closed-loop trajectory $x_{\mu_N}(k)$ according to the rule

$$x_{\mu_N}(k+1) = f(x_{\mu_N}(k), \mu_N(x_{\mu_N}(k), k)) \tag{1.6}$$

In this work, we consider two other variants of MPC controllers. First, we consider **multistep** or **m-step feedback MPC** (see [32]), $m \in \{2, \ldots, N-1\}$ in which the optimization in Step (2) is performed less often, by applying the first m elements of the optimal control sequence obtained after optimization.

Algorithm 1.3.2. (Multistep or **m-step MPC)**

(1) Measure the state $x(k) \in \mathbb{X}$ of the system at time instant k

(2) Set $x_0 := x(k)$ and solve the finite horizon problem $\mathcal{P}_N(x_0)$. Let $u^*(\cdot) \in \mathbb{U}^N(x_0)$ denote the optimal control sequence and define the time-dependent m-step MPC feedback

$$\mu_{N,m}(x(k), k+j) := u^*(j), \quad j = 0, \ldots, m-1 \tag{1.7}$$

(3) Apply the control values $\mu_{N,m}(x(k), k+j)$, $j = 0, \ldots, m-1$, to the system, set $k := k + m$ and go to (1)

Remark 1.3.3. Observe that through the scheme, the loop is only closed every m-steps, i.e., the system runs in open-loop within m-steps before optimization is performed again to compute a new set of controls.

Here, the value m is called the control horizon. The resulting nominal closed-loop system is given by

$$x_{\mu_{N,m}}(k+1) = f(x_{\mu_{N,m}}(k), \mu_{N,m}(x_{\mu_{N,m}}(\lfloor k \rfloor_m), k)) \tag{1.8}$$

with $\tilde{k}(k) = \lfloor k \rfloor_m$ for the notation introduced in (1.2) where $\lfloor k \rfloor_m$ denotes the largest integer multiple of m less than or equal to k. The motivation behind considering m-step MPC is that the number of optimizations is reduced by the factor $1/m$, thus the computational effort decreases accordingly.

Second, we also consider the **updated multistep feedback MPC** which, similar to the usual MPC, entails performing optimization every time step, but unlike the standard MPC, wherein we perform optimization over full horizon N, we re-optimize over shrinking horizons.

Algorithm 1.3.4. (Updated m-step MPC)

(1) Measure the state $x(k) \in \mathbb{X}$ of the system at time instant k

(2) Set $j := k - \lfloor k \rfloor_m$, $x_j := x(k)$ and solve the finite horizon problem $\mathcal{P}_{N-j}(x_j)$. Let $u^*(\cdot) \in \mathbb{U}^{N-j}(x_0)$ denote the optimal control sequence and define the updated MPC feedback

$$\hat{\mu}_{N,m}(x(k), k) := u^*(0) \tag{1.9}$$

(3) Apply the control value $\hat{\mu}_{N,m}(x(k), k)$ to the system, set $k := k + 1$ and go to (1)

The nominal updated multistep MPC closed loop is then described by

$$x_{\hat{\mu}_{N,m}}(k+1) = f(x_{\hat{\mu}_{N,m}}(k), \hat{\mu}_{N,m}(x_{\hat{\mu}_{N,m}}(k), k)) \tag{1.10}$$

We note that due to the dynamic programming principle in Theorem 1.2.1, in the nominal setting the closed loop generated by the multistep feedback (1.8) and by the updated multistep feedback MPC closed-loop system (1.10) coincide. For this reason, the use of Algorithm 1.3.4 only becomes meaningful in the presence of perturbations. These will be formalized in Section 2.2.

In presence of perturbations, we expect the updated multistep feedback to provide more robustness, in the sense that stability is maintained for larger perturbations and performance degradation is less pronounced as for the non-updated case. This will be rigorously analyzed in Chapter 4. Compared to standard MPC, the optimal control problems on shrinking horizon needed for the updates are faster to solve than the optimal control problems on full horizon. Moreover, for small perturbations the updates may also be replaced by approximative updates in which re-optimizations are approximated by a sensitivity approach, leading to another significant reduction of the computation time. This variant is analyzed in Chapter 6.

2 MPC stability and performance

This chapter provides the fundamental theorems that will serve as the basis of the analysis that we will conduct on various MPC schemes. In Section 2.1, we present some established results in the analysis of nominal MPC (see e.g., [36, 32, 37]) consisting of statements on stability guarantees and performance in terms of suboptimality with respect to the infinite horizon problem $\mathcal{P}_\infty(x_0)$. We aim to apply the MPC variants on real systems and for this reason, we introduce in Section 2.2 perturbed systems, as opposed to nominal systems. After having summarized the main steps of the analysis of the nominal MPC without terminal constraints, we adapt the statements to the analysis of feedback laws under perturbations.

2.1 Nominal stability and performance

Suppose x_* is an equilibrium of (1.1). MPC determines $\mu : \mathbb{X} \times \mathbb{N} \to \mathbb{U}$ that approximately solves the infinite horizon OCP such that x_* is asymptotically stable for the feedback-controlled system (1.6) in the following sense.

Definition 2.1.1. An equilibrium $x_* \in \mathbb{X}$ is **asymptotically stable** for the closed-loop system (1.2) if there exists $\beta \in \mathcal{KL}$ such that

$$\|x_\mu(k, x_0)\|_{x_*} \leq \beta(\|x_0\|_{x_*}, k)$$

holds for all $x_0 \in \mathbb{X}$ and all $k \in \mathbb{N}_0$ where $\|x\|_{x_*} := \|x - x_*\|$. In this case, we say that the feedback law μ asymptotically stabilizes x_*.

Conditions ensuring that the MPC feedback law asymptotically stabilizes the system have been well-developed in the literature. On one hand, refer, e.g., to [57], [36, Chapter 5] and references therein, we see that employing stabilizing terminal constraints or adding Lyapunov function terminal costs to the objective function ensure asymptotic stability of the MPC closed loop. On the other hand, see, e.g., [32], [37] and [36, Chapter 6] and references therein, we observe that imposing such terminal constraints and costs are not necessary conditions for achieving stability. In addition, due to the simplicity in design and implementation, MPC without terminal constraints and costs is often preferred in practice and with this motivation, we will be interested in analyzing the properties of MPC without terminal conditions in this thesis.

To achieve asymptotic stability, an appropriate choice of the stage cost ℓ is needed and is typically obtained by penalizing the distance of the state to the desired equilibrium and the control effort. This is enforced by making the following assumption.

Assumption 2.1.2. There exist $\mathcal{K}_\infty$-functions α_1, α_2 such that the inequality

$$\alpha_1(\|x\|_{x_*}) \leq \ell^*(x) \leq \alpha_2(\|x\|_{x_*}) \tag{2.1}$$

holds for all $x \in \mathbb{X}$, where $\ell^*(x) := \inf_{u \in \mathbb{U}} \ell(x,u)$.

The following gives the key statement for the analysis of MPC without terminal constraints or costs.

Proposition 2.1.3. *(i) Consider a time-dependent feedback law $\mu : \mathbb{X} \times \mathbb{N} \to U$, the corresponding solution $x_\mu(k, x_0)$ of (1.2), and a function $V : X \to \mathbb{R}_0^+$ satisfying the* ***relaxed dynamic programming inequality***

$$V(x_0) \geq V(x_\mu(m, x_0)) + \alpha \sum_{k=0}^{m-1} \ell(x_\mu(k, x_0), \mu(x_\mu(\lfloor k \rfloor_m, x_0), k)) \tag{2.2}$$

for some $\alpha \in (0,1]$, some $m \geq 1$ and all $x_0 \in \mathbb{X}$. Then for all $x \in \mathbb{X}$ the estimate

$$V_\infty(x) \leq J_\infty^{cl}(x, \mu) \leq V(x)/\alpha \tag{2.3}$$

holds.

(ii) If, moreover, Assumption 2.1.2 holds and there exist $\alpha_3, \alpha_4 \in \mathcal{K}_\infty$ such that

$$\alpha_3(\|x\|_{x_*}) \leq V(x) \leq \alpha_4(\|x\|_{x_*})$$

for all $x \in \mathbb{X}$, then the equilibrium x_* is asymptotically stable for the closed-loop system.

Proof. (i) The proof follows [32, Proof of Proposition 2.4]. Consider $x_0 \in \mathbb{X}$ and the closed-loop trajectory $x_\mu(k, x_0)$. Then from (2.2) we obtain for all $n \in \mathbb{N}_0$

$$\begin{aligned} \alpha \sum_{k=0}^{m-1} & \ell(x_\mu(nm+k, x_0), \mu(x_\mu(\lfloor nm+k \rfloor_m, x_0), nm+k)) \\ & \leq \; V(x_\mu(nm, x_0)) - V(x_\mu((n+1)m, x_0)) \end{aligned}$$

Performing a summation over n gives

$$\begin{aligned} \alpha \sum_{k=0}^{Km-1} & \ell\left(x_\mu(k, x_0), \mu(x_\mu(\lfloor k \rfloor_m, x_0), k)\right) \\ = \; & \alpha \sum_{n=0}^{K-1} \sum_{k=0}^{m-1} \ell(x_\mu(nm+k, x_0), \mu(x_\mu(\lfloor nm+k \rfloor_m, x_0), nm+k)) \\ \leq \; & V(x_0) - V(x(Km, x_0)) \; \leq \; V(x_0) \end{aligned}$$

The leftmost sum is bounded from above for every $K \in \mathbb{N}$ and is monotonically increasing which implies convergence as $K \to \infty$, therefore

$$V_\infty(x) \leq J_\infty^{\mathrm{cl}}(x, \mu) \leq V(x)/\alpha$$

(ii) Following [32, Proof of Theorem 5.2], by standard construction (see [43, Section 4.4]) we obtain a function $\rho \in \mathcal{KL}$ such that $V(x_\mu(km, x_0)) \le \rho(V(x_0), k)$ holds for all $x_0 \in \mathbb{X}$. Now consider $k \in \mathbb{N}$ which is not an integer multiple of m. By (2.2) with $x_\mu(\lfloor k \rfloor_m, x_0)$ in place of x_0 and the nonnegativity of ℓ, we have

$$\ell\left(x_\mu(k, x_0), \mu(x_\mu(\lfloor k \rfloor_m, x_0), k)\right) \le V(x_\mu(\lfloor k \rfloor_m, x_0))/\alpha$$

Since $V(x) \le \alpha_4 \circ \alpha_1^{-1}(\ell(x, u))$ holds for all u, we obtain

$$\begin{aligned} V(x_\mu(k, x_0)) &\le \alpha_4 \circ \alpha_1^{-1}(V(x_\mu(\lfloor k \rfloor_m, x_0))/\alpha) \\ &\le \alpha_4 \circ \alpha_1^{-1}(\rho(V(x_0), \lfloor k \rfloor_m)/\alpha) \end{aligned}$$

which yields

$$\|x_\mu(k, x_0)\|_{x_*} \le \alpha_3^{-1} \circ \alpha_4 \circ \alpha_1^{-1}(\rho(\alpha_4(\|x_0\|_{x_*}), \lfloor k \rfloor_m)/\alpha)$$

Therefore, $\|x_\mu(k, x_0)\|_{x_*} \le \beta(\|x_0\|_{x_*}, k)$ for all $k \in \mathbb{N}$, i.e., the desired asymptotic stabilty with $\mathcal{KL}$-function

$$\beta(r, k) := \alpha_3^{-1} \circ \alpha_4 \circ \alpha_1^{-1}(\rho(\alpha_4(r), \lfloor k \rfloor_m)/\alpha) + e^{-k}$$

which is easily extended to a $\mathcal{KL}$-function by linear interpolation in its second argument. □

In Proposition 2.1.3, to show asymptotic stability of a closed-loop system driven by $\mu_{N,m}$, we need to show existence of a function V and a value $\alpha \in (0, 1]$ satisfying the relaxed dynamic programming inequality (2.2). The use of the relaxed dynamic programming inequality in the form (2.2) was first introduced for the analysis of MPC schemes in [38]. Other forms, however, were earlier used in [60].

Proposition 2.1.3 implies that aside from providing the estimate (2.3) (on which a so-called suboptimality estimate, discussed towards the end of the section, will be based), showing the existence of a positive α also ensures asymptotic stability for the closed-loop system. In the sequel, we examine the feedback law $\mu_{N,m}$ and consider $V := V_N$. We present in the following an approach of computing α. One way to obtain α is by requiring the following assumption.

Assumption 2.1.4. There exists $B_k \in \mathcal{K}_\infty$ such that the optimal value functions of $\mathcal{P}_k(x_0)$ satisfy

$$V_k(x) \le B_k(\ell^*(x)) \quad \text{for all} \quad x \in \mathbb{X} \text{ and all } \quad k = 2, \dots, N$$

Remark 2.1.5. The existence of the functions B_k can be concluded, for instance, by assuming certain controllability assumptions. See, e.g., [36, Assumption 6.4] or [66, Assumption 3.2 and Lemma 3.5] wherein the system is assumed to be **asymptotically controllable with respect to** ℓ, i.e. if there exists $\beta \in \mathcal{KL}_0$ such that for every $x \in \mathbb{X}$ and every $N \in \mathbb{N}$, there exists an admissible control sequence $u_x \in \mathbb{U}^N(x)$ satisfying

$$\ell(x_{u_x}(k, x), u_x(k)) \le \beta(\ell^*(x), k)$$

for all $k \in \{0, \dots, N-1\}$.

Example 2.1.6. Suppose there exist constants $C > 0$ and $\sigma \in (0, 1)$ such that for every $x \in \mathbb{X}$ and every $N \in \mathbb{N}$, there is $u_x \in \mathbb{U}^N(x)$ such that

$$\ell(x_{u_x}(k,x), u_x(k)) \le C\sigma^k \ell^*(x)$$

for all $k \in \{0,\dots,N-1\}$. Then we take $\beta(r,k) = C\sigma^k r \in \mathcal{KL}_0$ giving $B_N(r) = \sum_{k=0}^{N-1} \beta(r,k) = C\sum_{k=0}^{N-1} \sigma^k r$ that fulfills Assumption 2.1.4. In this case, the system is said to be exponentially controllable with respect to ℓ.

The following proposition considers arbitrary values $\lambda_n,\ n = 0,\dots,N-1$, and ν and gives necessary conditions which hold if these values coincide with optimal stage costs $\ell(x_{u^*}(n,x_0), u^*(n))$ and optimal values $V_N(x_{u^*}(m,x_0))$, respectively.

Proposition 2.1.7. *Let Assumption 2.1.4 hold and consider $N \ge 1, m \in \{1,\dots,N-1\}$, a sequence $\lambda_n \ge 0,\ n = 0,\dots,N-1$, a value $\nu \ge 0$. Consider $x_0 \in X$ and assume that there exists an optimal control function $u^*(\cdot) \in \mathbb{U}$ for the finite horizon problem $\mathcal{P}_N(x_0)$ with horizon length N, such that*

$$\lambda_n = \ell(x_{u^*}(n,x_0), u^*(n)), \quad n = 0,\dots,N-1$$

holds. Then

$$\sum_{n=k}^{N-1} \lambda_n \le B_{N-k}(\lambda_k), \quad k = 0,\dots,N-2 \tag{2.4}$$

holds. If, furthermore,

$$\nu = V_N(x_{u^*}(m,x_0))$$

holds, then

$$\nu \le \sum_{n=0}^{j-1} \lambda_{n+m} + B_{N-j}(\lambda_{j+m}), \quad j = 0,\dots,N-m-1 \tag{2.5}$$

holds.

Proof. Observe that for $k = 0,\dots,N-2$,

$$\begin{aligned} V_N(x_0) &= J_k(x_0,u^*(\cdot)) + J_{N-k}(x_{u^*}(k,x_0), u^*(k+\cdot)) && (2.6)\\ &= J_k(x_0,u^*(\cdot)) + V_{N-k}(x_{u^*}(k,x_0)) && (2.7)\end{aligned}$$

by (2.7) and Assumption 2.1.4, we have

$$V_N(x_0) \le J_k(x_0,u^*(\cdot)) + B_{N-k}(\ell(x_{u^*}(k,x_0))) \tag{2.8}$$

Subtracting (2.6) from (2.8) gives

$$J_{N-k}(x_{u^*}(k,x_0), u^*(k+\cdot)) \le B_{N-k}(\ell^*(x_{u^*}(k,x_0)))$$

yielding (2.4). Next we define the control function

$$\tilde{u}(n) = \begin{cases} u^*(m+n), & n \le j-1 \\ u^{**}(n), & n \le j \end{cases}$$

where $u^{**}(\cdot)$ is the optimal control for $\mathcal{P}_{N-j}(x_{u^*}(m+j))$. Then we obtain

$$\begin{aligned}
V_N(x_{u^*}(m,x_0)) &= J_N(x_{u^*}(m),\tilde{u}(\cdot)) \\
&= J_j(x_{u^*}(m,x_0),u^*(m+\cdot)) + J_{N-j}(x_{u^*}(m+j,x_0),u^{**}(\cdot)) \\
&= J_j(x_{u^*}(m,x_0),u^*(m+\cdot)) + V_{N-j}(x_{u^*}(m+j,x_0)) \\
&\leq J_j(x_{u^*}(m,x_0),u^*(m+\cdot)) + B_{N-j}(\ell^*(x_{u^*}(m+j,x_0)))
\end{aligned}$$

yielding (2.5). □

By using the proposition, we arrive at the following theorem giving sufficient conditions for suboptimality and stability of the m-step MPC feedback law $\mu_{N,m}$ and an approach to compute the suboptimality index α.

Theorem 2.1.8. *Let Assumption 2.1.4 hold and assume that the optimization problem*

$$\begin{aligned}
&\alpha := \inf_{\lambda_0,\ldots,\lambda_{N-1},\nu} \frac{\sum_{n=0}^{N-1}\lambda_n - \nu}{\sum_{n=0}^{m-1}\lambda_n} \\
&\textit{subject to the constraints } (2.4) \textit{ and } (2.5) \\
&\textit{and } \ \sum_{n=0}^{m-1}\lambda_n > 0,\ \lambda_0,\ldots,\lambda_{N-1},\nu \geq 0
\end{aligned} \qquad \mathcal{P}_\alpha$$

has an optimal value $\alpha \in (0,1]$. Then, the optimal value function V_N of $\mathcal{P}_N(x)$ and the m-step MPC feedback law $\mu_{N,m}$ satisfy the assumptions of Proposition 2.1.3(i) and, in particular, the inequality

$$V_\infty(x) \leq J_\infty^{cl}(x,\mu_{N,m}) \leq V_N(x)/\alpha \leq V_\infty(x)/\alpha \tag{2.9}$$

holds for all $x \in \mathbb{X}$. If, moreover, Assumption 2.1.2 holds then the closed loop is asymptotically stable.

Proof. From the solution $u^*(\cdot)$ of $\mathcal{P}_N(x_0)$ for $x_0 \in \mathbb{X}$, we construct the m-step feedback $\mu_{N,m}$ giving the equalities

$$\begin{aligned}
\mu_{N,m}(x_0,k) &= u^*(k), & k &= 0,\ldots,m-1 \\
x_{\mu_{N,m}}(k,x_0) &= x_{u^*}(k,x_0), & k &= 0,\ldots,m \\
\ell(x_{\mu_{N,m}}(k,x_0),\mu_{N,m}(x_0,k)) &= \ell(x_{u^*}(k,x_0),u^*(k)), & k &= 0,\ldots,m-1
\end{aligned}$$

which implies

$$\begin{aligned}
V_N(x_{\mu_{N,m}}(m,x_0)) &+ \alpha\sum_{k=0}^{m-1}\ell(x_{\mu_{N,m}}(k,x_0),\mu_{N,m}(x_{\mu_{N,m}}(k,x_0),k)) \\
&= V_N(x_{u^*}(m,x_0)) + \alpha\sum_{k=0}^{m-1}\ell(x_{u^*}(k,x_0),u^*(k))
\end{aligned} \tag{2.10}$$

for any $\alpha \in \mathbb{R}$. Since $\mathcal{P}_\alpha$ has a solution, the values $\lambda_k = \ell(x_{u^*}(k,x_0),u^*(k))$ and $\nu = V_N(x_{u^*}(m,x_0)$ satisfy (2.4), (2.5) and

$$\sum_{k=0}^{N-1}\lambda_k - \nu \geq \alpha\sum_{k=0}^{m-1}\lambda_k$$

Hence, we obtain

$$\begin{aligned} V_N(x_{u^*}(m,x_0)) + \alpha \sum_{k=0}^{m-1} \ell(x_{u^*}(k,x_0),u^*(k)) &= \nu + \alpha \sum_{k=0}^{m-1} \lambda_k \leq \sum_{k=0}^{N-1} \lambda_k \\ &= \sum_{k=0}^{N-1} \ell(x_{u^*}(k,x_0),u^*(k)) \\ &= V_N(x_0) \end{aligned}$$

Together with (2.10), this yields (2.2) and thus the assertion. The second assertion follows from Proposition 2.1.3(ii) setting $\alpha_4 := B_N$. $\square$

Because of (2.9), we refer to α as an **index of suboptimality** which provides a performance bound indicating how well the feedback law $\mu_{N,m}$ approximates the solution of the infinite horizon problem $\mathcal{P}_\infty(x_0)$. If $\alpha = 1$, then the feedback law is infinite horizon optimal. This implies that the closer to 1 the positive index α is, the closer the feedback law approximates the solution of $\mathcal{P}_\infty(x_0)$ while the smaller α is, the larger the suboptimality gap becomes.

Remark 2.1.9. The proof of Theorem 2.1.8 particularly shows the relaxed dynamic programming inequality (2.2) for $V = V_N$ and $\mu = \mu_{N,m}$, i.e.,

$$V_N(x_{\mu_{N,m}}(m,x_0)) \leq V_N(x_0) - \alpha \sum_{k=0}^{m-1} \ell(x_{\mu_{N,m}}(k,x_0), \mu_{N,m}(x_{\mu_{N,m}}(k,x_0),k)) \tag{2.11}$$

for all $x_0 \in \mathbb{X}$. This inequality can be seen as a Lyapunov inequality and shows that V_N is an m-step Lyapunov function indicating the descent property of the value function along the closed-loop trajectory at every m time instants. Refer, e.g., to [36, Section 2.3], [57, Appendix B] or [43, Chapter 4] for discussions on Lyapunov stability theory.

The optimization problem $\mathcal{P}_\alpha$ may be nonlinear depending on the nature of $B_k(r)$ from Assumption 2.1.2. However, $\mathcal{P}_\alpha$ becomes a linear program in r if $B_k(r)$ is linear. An explicit formula for α can be derived in this case.

Theorem 2.1.10. *Let B_K, $K = 2,\ldots,N$, be linear functions and define $\gamma_K := B_K(r)/r$. Then the optimal value α of problem $\mathcal{P}_\alpha$ for given optimization horizon N, control horizon m satisfies satisfies $\alpha = 1$ if and only if $\gamma_{m+1} \leq 1$ and*

$$\alpha \geq 1 - \frac{(\gamma_{m+1}-1)\prod\limits_{i=m+2}^{N}(\gamma_i - 1)\prod\limits_{i=N-m+1}^{N}(\gamma_i-1)}{\left[\prod\limits_{i=m+1}^{N}\gamma_i - (\gamma_{m+1}-1)\prod\limits_{i=m+2}^{N}(\gamma_i-1)\right]\left[\prod\limits_{i=N-m+1}^{N}\gamma_i - \prod\limits_{i=N-m+1}^{N}(\gamma_i-1)\right]} \tag{2.12}$$

otherwise. If, moreover, the B_K are of the form $B_K(r) := \sum_{k=0}^{K-1}\beta(r,k)$ for some $\beta \in \mathcal{KL}_0$ satisfying $\beta(r,n+m) \leq \beta(\beta(r,n),m)$ for all $r \geq 0, n,m \in \mathbb{N}_0$, then equality holds in (2.12).

Proof. See Theorem 5.4 and Remark 5.5 of [37]. $\square$

The analysis on [37] assesses the impact of the optimization horizon on stability and performance of the closed loop. By closely examining (2.12), one can find that $\alpha \to 1$ as $N \to \infty$ if there exists $\bar{\gamma} \in \mathbb{R}$ with $\gamma_k \leq \bar{\gamma}$ for all $k \in \mathbb{N}$ [37, Corollary 6.1]. Therefore, under this condition, stability and performance arbitrarily close to the infinite horizon optimal performance can always be achieved by choosing N sufficiently large. In addition, the right-hand side value of (2.12) for $m = 1$ is always less than or equal to the value for $m \geq 2$ [37, Proposition 7.3]. This means that if Theorem 2.1.10 guarantees asymptotic stability (i.e., $\alpha > 0$) of standard MPC $m = 1$ (Algorithm 1.3.1), then it also guarantees stability of m-step MPC for arbitrary $m = 2, \ldots, N-1$ (Algorithms 1.3.2 and 1.3.4).

Before we proceed to analyze the properties of the feedback law under perturbation, we conclude the section by summarizing the presented course of reasoning. Assumption 2.1.4 allows for the formulation of the optimization problem $\mathcal{P}_\alpha$. If $\mathcal{P}_\alpha$ has a solution $\alpha > 0$, then this implies the Lyapunov inequality (2.11) fulfilling the assumptions of Proposition 2.1.3 from which asymptotic stability and performance estimates can be obtained. In case B_K in Assumption 2.1.4 is linear, an explicit formula for the solution of $\mathcal{P}_\alpha$ is provided by (2.12). We extend this setting and approach to analyze perturbed systems in the remainder of the thesis.

2.2 Perturbed systems, robust stability and performance

The results discussed in the previous section are based on a nominal setting wherein no perturbations are acting on the system dynamics. In this section, we generalize Proposition 2.1.3 to the perturbed situation. A counterpart of Theorem 2.1.8 for $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ in the perturbed setting will be obtained in Chapter 4.

Typically, a real world system is represented by a mathematical model that may fail to take into account disturbance and other various sources of uncertainties. Mathematical models are approximations of real systems where there is usually a mismatch between the predicted states and those that are measured from the real plant. This mismatch can be viewed as perturbations and can be taken into account via the **perturbed closed-loop model**[1]

$$\tilde{x}(k+1) = f\left(\tilde{x}(k), \mu(\tilde{x}(\tilde{k}), k)\right) + d(k) \tag{2.13}$$

Here, $d(k) \in X$ represents external perturbation and modeling errors.

Remark 2.2.1. For simplicity and brevity of exposition, we focus on the analysis of the closed-loop model (2.13) instead of the more general model

$$\tilde{x}(k+1) = f\left(\tilde{x}(k), \mu(\tilde{x}(\tilde{k}) + e(k), k)\right) + d(k)$$

where $e(k) \in X$ represents state measurement errors. Stability and performance statements for this model can be derived from respective statements for (2.13) using the techniques from [36, Proof of Theorem 8.36] or [41, Proof of Proposition 1]. Whenever the measurement error is small compared to the external disturbance, we expect the beneficial effects of re-optimization analyzed in the thesis to hold analogously. Otherwise, large measurement errors may lead to

[1] As indicated in Chapter 1, the feedback value $\mu(\tilde{x}(\tilde{k}), k)$ depends on the system state $x(\tilde{k})$ at a time $\tilde{k} = \tilde{k}(k) \leq k$ which may be strictly smaller than k.

adverse effects particularly under fast sampling as analyzed in [61]. The trade-off analysis between the benefits of re-optimization and fast sampling is, however, beyond the scope of this thesis.

Due to the perturbations experienced by the system, the succeeding entries of the m-step feedback may no longer be suitable since the succeeding current states may be different from the predicted as the loop is not closed within m time instants. This thesis aims to investigate the benefit of updates (i.e. re-optimization) in this situation.

In the following discussion, we use the notation $\tilde{x}_\mu(\cdot, x_0)$ to denote a solution of (2.13) in order to distinguish it from the nominal trajectory $x_\mu(\cdot, x_0)$. Furthermore, we consider the set

$$S_{\overline{d}}(x_0) := \left\{ \tilde{x}_\mu(\cdot, x_0) \mid \|d(k)\| \leq \overline{d} \text{ for all } k \in \mathbb{N}_0 \right\}$$

of all possible solutions steered by μ starting in x_0 with perturbations bounded by $\overline{d}$.

Remark 2.2.2. In our discussion, we assume that for the initial values x_0, perturbation levels $\overline{d}$ and feedback laws μ under consideration, any trajectory $\tilde{x}_\mu(\cdot, x_0) \in S_{\overline{d}}(x_0)$ exists and satisfies $\tilde{x}_\mu(k, x_0) \in \mathbb{X}$ for all $k \in \mathbb{N}$. Techniques which allow to rigorously ensure this property are discussed, e.g., in [36, Sections 8.8–8.9] and the references therein.

Asymptotic stability is in general too strong a requirement for a system to achieve under perturbations. Nevertheless, it is often still possible to prove an appropriate relaxation of the stability properties for the feedback law μ. To this end, we make use of the so-called practical stability defined in the following.

Definition 2.2.3. Given $\overline{d} > 0$. Consider sets $\widehat{P} \subset Y \subseteq \mathbb{X}$. A point $x_* \in \widehat{P}$ is called **$\widehat{P}$-practically uniformly asymptotically stable on** Y if there exists $\beta \in \mathcal{KL}$ such that

$$\|\tilde{x}_\mu(k, x_0)\|_{x_*} \leq \beta(\|x_0\|_{x_*}, k)$$

holds for all $x_0 \in Y$, all $\tilde{x}_\mu(\cdot, x_0) \in S_{\overline{d}}(x_0)$ and all k with $\tilde{x}(k, x_0) \notin \widehat{P}$.

The definition requires the system to have asymptotically stable behavior until it reaches the set $\widehat{P}$. We can interpret $\widehat{P}$ as the region of the state space wherein the effects of the perturbations become dominant.

Definition 2.2.4. We say that x_* is **semi-globally practically asymptotically stable with respect to perturbation** d if there exists $\beta \in \mathcal{KL}$ such that the following property holds: for each $\delta > 0$ and $\Delta > \delta$ there exists $\overline{d} > 0$ such that

$$\|\tilde{x}_\mu(k, x_0)\|_{x_*} \leq \max\{\beta(\|x_0\|_{x_*}, k), \delta\} \tag{2.14}$$

holds for all $x_0 \in \mathbb{X}$ with $\|x_0\|_{x_*} \leq \Delta$, all $\tilde{x}_\mu(\cdot, x_0) \in S_{\overline{d}}(x_0)$ and all $k \in \mathbb{N}_0$.

In words, this definition demands that for initial values within a distance of Δ from the equilibrium, the system behaves with asymptotic stability until the trajectory is within a distance of δ from the equilibrium. Here, Δ and δ determine the admissible bound $\overline{d}$ on the perturbation. In what follows, we relate Definitions 2.2.3 and 2.2.4 via the following lemma.

Lemma 2.2.5. *The m-step MPC closed-loop system* (1.8) *is semi-globally practically asymptotically stable with respect to d in the sense Definition 2.2.4 if for every $\delta > 0$ and every $\Delta > \delta$ there exists $\overline{d} > 0$ and sets $\widehat{P} \subset Y \subseteq \mathbb{X}$ with*

$$\overline{\mathcal{B}}_{\Delta}(x_*) \cap \mathbb{X} \subseteq Y \quad \text{and} \quad \widehat{P} \subseteq \overline{\mathcal{B}}_{\delta}(x_*)$$

such that for each solution $\tilde{x}_\mu(\cdot, x_0) \in S_{\overline{d}}(x_0)$ the system is $\widehat{P}$-practically uniform asymptotically stable on Y in the sense of Definition 2.2.3 .

Proof. The proof follows from the fact that according to Definition 2.2.3 for each $k \in \mathbb{N}_0$ either $\|\tilde{x}_\mu(k, x_0)\|_{x_*} \leq \beta(\|x_0\|_{x_*}, k)$ or $\tilde{x}_\mu(k, x_0) \in \widehat{P}$. Since the latter implies $\|\tilde{x}_\mu(k, x_0)\|_{x_*} \leq \delta$, we observe the assertion. □

Now that we have defined the suitable notion of stability for our setting, we can also define the appropriate performance measure. We have described $\widehat{P}$ as the region of the state space in which the perturbations become predominant. Hence, when considering the performance of such a solution, it only makes sense to consider the trajectory until it first hits the set $\widehat{P}$. Thus, we need to truncate the infinite horizon closed-loop cost $J_\infty^{\mathrm{cl}}(x_0, \mu)$ from (1.3) as follows.

Definition 2.2.6. Consider a set $\widehat{P} \subset \mathbb{X}$. Then the performance associated to a perturbed solution $\tilde{x}_\mu(\cdot, x_0)$ of a closed-loop system outside $\widehat{P}$ is defined as

$$J_{\widehat{P}}^{\mathrm{cl}}(\tilde{x}_\mu(\cdot, x_0), \mu) := \sum_{k=0}^{k^*-1} \ell\left(\tilde{x}_\mu(k, x_0), \mu(\tilde{x}_\mu(\tilde{k}, x_0), k)\right) \tag{2.15}$$

where $k^* \in \mathbb{N}_0$ is minimal with $\tilde{x}_\mu(k, x_0) \in \widehat{P}$ for all $k \geq k^*$.

As a technical ingredient, we additionally need the following set properties.

Definition 2.2.7. Let $m \in \mathbb{N}$.

(i) A set $Y \subseteq \mathbb{X}$ is said to be **m-step forward invariant for** (2.13) **with respect to** $\overline{d}$ if for all $x_0 \in Y$ and all $\tilde{x}_\mu(\cdot, x_0) \in S_{\overline{d}}(x_0)$, it holds that $\tilde{x}_\mu(pm, x_0) \in Y$ for all $p \in \mathbb{N}$.

(ii) For an m-step forward invariant set Y with respect to $\overline{d}$ we call $\widehat{Y} \supseteq Y$ an **intermediate set** if $\tilde{x}_\mu(k, x_0) \in \widehat{Y}$ for all $k \in \mathbb{N}$ and all $x_0 \in Y$.

We are now in the position to state the following theorem which extends Proposition 2.1.3 to the perturbed setting.

Theorem 2.2.8. *(i) Consider a stage cost $\ell : X \times U \to \mathbb{R}_0^+$, an integer $m \in \mathbb{N}$ and a function $V : X \to \mathbb{R}_0^+$. Let $\mu : \mathbb{X} \times \mathbb{N} \to U$ be an admissible m-step feedback law of the form* (1.7) *or* (1.9) *and let $Y \subseteq \mathbb{X}$ and $P \subset Y$ be m-step forward invariant for* (2.13) *with respect to some $\overline{d} > 0$. Let $\widehat{P} \supseteq P$ be an intermediate set for P. Assume there exists $\alpha \in (0, 1]$ such that the relaxed dynamic programming inequality*

$$V(x_0) \geq V(\tilde{x}_\mu(m, x_0)) + \alpha \sum_{k=0}^{m-1} \ell(\tilde{x}_\mu(k, x_0), \mu(\tilde{x}_\mu(\lfloor k \rfloor_m, x_0), k)) \tag{2.16}$$

holds for all $x_0 \in Y \setminus P$ and all $\tilde{x}_\mu(\cdot, x_0) \in S_{\overline{d}}(x_0)$. Then the performance estimate

$$J_{\widehat{P}}^{cl}(\tilde{x}_\mu(\cdot, x_0), \mu) \leq V(x_0)/\alpha \tag{2.17}$$

holds for all $x_0 \in Y \setminus \widehat{P}$ *and all* $\tilde{x}_\mu(k, x_0) \in S_{\overline{d}}(x_0)$.

(ii) If, moreover, Assumption 2.1.2 holds and there exists $\alpha_3, \alpha_4 \in \mathcal{K}_\infty$ *with* $\alpha_3(\|x\|_{x_*}) \leq V(x) \leq \alpha_4(\|x\|_{x_*})$ *for all* $x \in \mathbb{X}$, *then the closed-loop system* (2.13) *is* $\widehat{P}$*-practically asymptotically stable on* Y *in the sense of Definition 2.2.3 .*

Proof. (i) Similar to the proof of Proposition 2.1.3 (i), for proving (2.17), by a straightforward induction from (2.16) we obtain

$$\alpha \sum_{k=0}^{pm-1} \ell(\tilde{x}_\mu(k, x_0), \mu(\tilde{x}_\mu(\lfloor k \rfloor_m, x_0), k)) \leq V(x_0) - V(\tilde{x}_\mu(pm, x_0)) \leq V(x_0)$$

for all $p \in \mathbb{N}$ for which $\tilde{x}_\mu(k, x_0) \notin P$ for $k = 0, m, 2m, \ldots, (p-1)m$. In particular, since $P \subseteq \widehat{P}$, this inequality holds for the smallest p satisfying $pm \geq k^*$ for k^* from Definition 2.2.6, implying

$$J^{\mathrm{cl}}_{\widehat{P}}(\tilde{x}_\mu(\cdot, x_0), \mu) \leq \sum_{k=0}^{pm-1} \ell(\tilde{x}_\mu(k, x_0), \mu(\tilde{x}_\mu(\lfloor k \rfloor_m, x_0), k)) \leq V(x_0)/\alpha$$

(ii) For proving practical asymptotic stability, as in first part of the proof of Proposition 2.1.3 (ii) we find a function $\rho \in \mathcal{KL}$ such that $V(x_\mu(pm, x_0)) \leq \rho(V(x_0), p)$ holds for all $x_0 \in Y$ and all $p \in \mathbb{N}$ with $pm \leq k^*$ for k^* from Definition 2.2.6. Now for $k \in \{1, \ldots, k^*\}$ which is not an integer multiple of m, (2.16) with $\tilde{x}_\mu(\lfloor k \rfloor_m, x_0)$ in place of x_0 and nonnegativity of ℓ imply

$$\ell(\tilde{x}_\mu(k, x_0), \mu(\tilde{x}_\mu(\lfloor k \rfloor_m, x_0), k)) \leq V(\tilde{x}_\mu(\lfloor k \rfloor_m, x_0))/\alpha.$$

Using the same technique as in Proposition 2.1.3 (ii) to construct the required β, we obtain $\|\tilde{x}_\mu(k, x_0)\|_{x_*} \leq \beta(\|x_0\|_{x_*}, k)$ for all $k = 0, \ldots, k^*$ with

$$\beta(r, k) := \alpha_3^{-1} \circ \alpha_4 \circ \alpha_1^{-1}(\rho(\alpha_4(r), \lfloor k \rfloor_m)/\alpha) + e^{-k}$$

which is easily extended to a $\mathcal{KL}$-function by linear interpolation in its second argument. Lastly, since $\tilde{x}_\mu(k^*, x_0) \in P$ implies that for all $k \geq k^*$ we have $\tilde{x}_\mu(k, x_0) \in \widehat{P}$, this shows the claimed $\widehat{P}$-practical asymptotic stability. □

This chapter gives the essential theorems required for the analysis of the robustness of a feedback-controlled closed loop. In the same manner Proposition 2.1.3 provides the key for the nominal system, Theorem 2.2.8 provides a pivotal theorem for the analysis of the stability rendered and performance of the feedback law μ for a system under perturbation. The analysis relies on the evaluation of the index α, whose positiveness indicates asymptotic stability and whose value gives the degree of suboptimality of the closed loop.

While Theorem 2.1.8 shows that $\mu_{N,m}$ renders the nominal system asymptotically stable, we aim to show analogous statements for $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ (and later on, $\overline{\mu}_{n,m}$, as defined in Chapter 6) for the perturbed system. As mentioned at the end of Section 1.3, we will rigorously show that the shrinking horizon update mechanism of the updated MPC algorithm (yielding $\hat{\mu}_{N,m}$) enhances the robustness of the closed loop.

3 Benefits of re-optimization on finite horizon OCPs

In this chapter, we divert the attention away from MPC and approximating the solution to the infinite horizon problem and instead focus on the finite horizon problem $\mathcal{P}_N(x_0)$ (recall the definition in Section 1.1). In Sections 3.1 to 3.3, we consider control algorithms for finite horizon problems and compare the so-called *nominal*, *perturbed* and *re-optimized* trajectories. We compare the trajectories by assigning each a performance measure that will allow us to quantify the benefit of re-optimization. Section 3.4 discusses controllability and stability and reports situations where the benefits of re-optimizations become significant. Illustrative examples are given in Section 3.5. The results we present in this chapter give essential tools for the analysis of the stability and performance of the feedback laws $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ for the perturbed setting that we will conduct in Chapter 4. A preliminary version of the said results is published in [34].

3.1 Control algorithms for finite horizon OCPs

In the rest of this chapter, we only consider the finite horizon problem with controls u^* and μ defined in the following algorithms.

Algorithm 3.1.1. (Open-loop control)

(1) Given x_0, solve the finite horizon problem $\mathcal{P}_N(x_0)$. Let $u^*(\cdot) \in \mathbb{U}^N(x_0)$ denote the optimal control sequence.

(2) For $k = 0, \ldots, N-1$, apply the control value $u^*(k)$ to the system.

This generates a nominal trajectory given by the rule

$$x(k+1) = f\left(x(k), u^*(k)\right) \quad k = 0, \ldots, N-1 \tag{3.1}$$

with $x(0) = x_0$ and the corresponding open-loop trajectory denoted by $x_{u^*}(\cdot, x_0)$.

Next, we design a controller that uses a shrinking horizon strategy.

Algorithm 3.1.2. (Shrinking horizon strategy)

For $k = 0, \ldots, N-1$,

(1) Solve $\mathcal{P}_{N-k}(x(k), k)$, i.e., we perform a re-optimization giving an optimal control sequence $u_k^*(j), j = 0, \ldots, N-1-k$ corresponding to the initial

value $x_0 = x(k)$ and a resulting trajectory $x_{u_k^*}(j), j = 0, \dots, N-k$. Note that for each sampling time k, the control horizon shrinks.

(2) Define the time-dependent feedback

$$\mu(x(k), k) := u_k^*(0)$$

and apply the control value to the system.

The closed-loop controlled system is described by

$$x(k+1) = f(x(k), \mu(x(k), k)) \quad k = 0, \dots, N-1 \tag{3.2}$$

Due to the dynamic programming principle in Theorem 1.2.1, in the nominal case where no uncertainties are present, (3.1) and (3.2) coincide. But as already mentioned in Chapter 1, this is not the case in the presence of perturbations. Due to the perturbations described in Section 2.2, a mismatch between the predicted states and those that are measured from the real plant is inevitable. We write the perturbed system controlled by the same open-loop controller used in (3.1) as

$$\tilde{x}(k+1) = f(\tilde{x}(k), u^*(k)) + d(k) \tag{3.3}$$

with $d(k) \in X$ representing external perturbation and modeling errors. This suggests, however, that the open-loop optimal control sequence obtained from the OCP solved at time 0, may not give the best control strategy as the system evolves in time.

Now we aim to scrutinize the effects of the disturbance and the advantage of using the shrinking horizon strategy in which we perform a re-optimization in each time step. In order to simplify the exposition, in the sequel we assume the existence of an optimal control sequence $u^*(\cdot)$ for each $x \in \mathbb{X}$ with $\mathbb{U}^N(x) \neq \emptyset$ and we examine the perturbed system using a shrinking horizon control strategy given by

$$\tilde{x}(k+1) = f(\tilde{x}(k), \mu(\tilde{x}(k), k)) + d(k) \tag{3.4}$$

We investigate whether the re-optimization in the shrinking horizon strategy addresses the drawbacks that the control design suffers from upon using open-loop control. To this end, we closely examine and compare the trajectories described above, namely, the **nominal trajectory** (3.1), the **perturbed trajectory** (3.3) and the **re-optimized trajectory** (3.4).

3.2 Nominal and perturbed trajectories

To analyze the nominal, the perturbed and the re-optimized trajectories, we introduce an intuitive and rigorous notation reflecting perturbations and performed re-optimizations.

Notation 3.2.1. Let $x_{j,p,r}$ denote the state trajectory element at time j that has gone through p perturbations at time instants $t = 1, \dots, p$ where $j \geq p$, and r re-optimizations have been performed at time instants $t = 1, \dots, r$ where $p \geq r$.

In this setting, we only put our attention to the trajectories generated by (3.1), (3.3) and (3.4) given by $x_{j,0,0}$, $x_{j,j,0}$ and $x_{j,j,j}$, respectively, for $j = 0, \dots, N$.

Notation 3.2.2. Let $u^*_{j,p,r}(\cdot)$ denote the optimal control sequence obtained by performing a re-optimization with an initial value $x_{j,p,r-1}$ and optimization horizon $N-j$, i.e., $u^*_{j,p,r}(\cdot)$ is obtained by solving $\mathcal{P}_{N-j}(x_{j,p,r-1})$.

Since the initial value does not change when performing a re-optimization, the identity $x_{j,p,r-1} = x_{j,p,r}$ holds. For our purposes, we will only consider states of the form $x_{j,p,r}$ with $r = 0, p, p-1$.

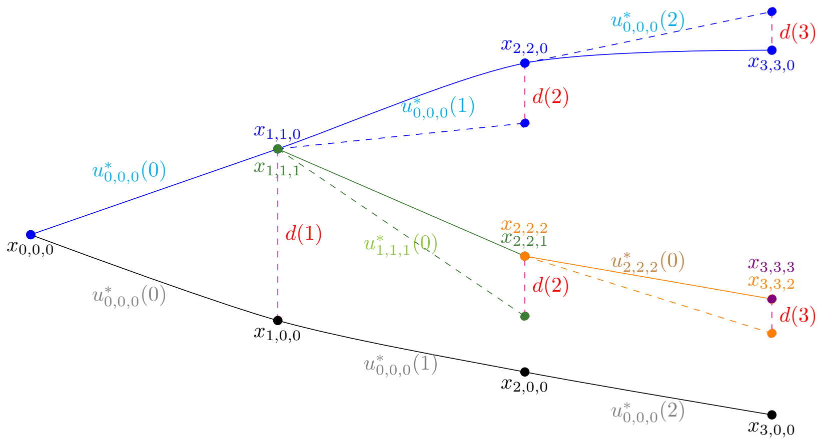

Figure 3.1: Trajectories through time where perturbations occur and re-optimizations are performed

Figure 3.1 illustrates the trajectories through time where perturbations occur and re-optimizations are performed for the control horizon $m = 3$. At time $t = 0$, by solving $\mathcal{P}_3(x_{0,0,0})$, we obtain an open-loop optimal control sequence $u^*_{0,0,0}(j) = u^*(j)$, $j = 0, 1, 2$ for which we can generate a nominal multistep trajectory $x_{j,0,0}, j = 0, \ldots, 3$ via (3.1) shown in black in the sketch. For an additive perturbation $d(\cdot)$, the blue trajectory in Figure 3.1 indicates the perturbed multistep trajectory $x_{j,j,0}$, $j = 0, \ldots, 3$ generated by (3.3). Here, each transition (shown in solid blue) is composed of the nominal transition $f(x_{j,j,0}, u^*_{0,0,0}(j))$ (blue dashed) followed by the addition of the perturbation $d(1), d(2), d(3)$ (red dashed). Finally, the trajectory $x_{j,j,j}$ obtained by re-optimization in each step and generated by (3.4) with perturbation d is shown piecewise in blue, green and orange, with the different colors indicating the different control sequences $u^*_{j,j,j}, j = 0, \ldots, 2$ whose first pieces are used in the transition. Again, the nominal transition and the effect of the perturbation $d(j)$ are indicated as dashed lines and the resulting perturbed transitions from $x_{j,j,j}$ to $x_{j+1,j+1,j} = x_{j+1,j+1,j+1}$ as solid lines.

Similar to how $x_{j,p,r}$ and $u^*_{j,p,r}$ were defined, we define the following stage cost.

Notation 3.2.3. For time instants $j \in \{0, \ldots, N-1\}$ and for $j \geq p,\ p \geq r,\ r = 0, p, p-1$ we define

$$\lambda_{j,p,r} := \ell\left(x_{j,p,r}, u^*_{r,r,r}(j-r)\right) \tag{3.5}$$

Observe that in order to determine the control needed to evaluate the stage cost for the state $x_{j,p,r}$, we go back to the last instant of the optimization, namely to time r and use the optimal control sequence obtained there for horizon $N-r$ and initial value $x_{r,r,r}$.

In order to simplify the numbering in the subsequent computations, we extend (3.5) to give meaning to the notation when $j<p,\ p\geq r,\ r=0,p,p-1$ through

$$\lambda_{j,p,r} := \begin{cases} \lambda_{j,j,j} & \text{if } r\neq 0 \\ \lambda_{j,j,0} & \text{if } r=0 \end{cases} \tag{3.6}$$

Remark 3.2.4. Although the previous discussion yields $x_{j,j,j-1}=x_{j,j,j}$, we see that $\lambda_{j,j,j-1}\neq\lambda_{j,j,j}$ since $\lambda_{j,j,j-1}=\ell\left(x_{j,j,j-1},u^*_{j-1,j-1,j-1}(1)\right)$ while $\lambda_{j,j,j}=\ell\left(x_{j,j,j},u^*_{j,j,j}(0)\right)$.

3.3 Re-optimizing versus not re-optimizing

In the presence of uncertainties or perturbations, we perform re-optimization in the hope of having a coping mechanism against the differences between the real system and the nominal model to redirect the trajectory back to the desired behavior aiming to stay 'close' to the nominal situation. We investigate whether re-optimization indeed gives such an advantage.

The idea is to find quantifiable relations among the various trajectory scenarios. More precisely, we compare the following scenarios.

Definition 3.3.1. Given an initial value $x_{0,0,0}=x_0\in\mathbb{X}$, we define the following performance measures.

i. The value of the nominal optimal trajectory

$$J_N^{\text{nmult}}(x_0) := \sum_{j=0}^{N-1}\lambda_{j,0,0}$$

ii. The value of the perturbed trajectory with nominal optimal control sequence

$$J_N^{\text{pmult}}(x_0) := \sum_{j=0}^{N-1}\lambda_{j,j,0}$$

iii. The value of the perturbed trajectory with re-optimized control

$$J_N^{\text{upd}}(x_0) := \sum_{j=0}^{N-1}\lambda_{j,j,j}$$

We recall that in Figure 3.1 the trajectories corresponding to these performance indices are indicated in black (i.), blue (ii.) and piecewise in blue, green and orange (iii.) and that they are generated by (3.1), (3.3) and (3.4), respectively. Further, it is easily seen that $J_N^{\text{nmult}}(x_0)=V_N(x_0)$ holds. This nominal optimal value will be our reference in the following analysis.

The following theorem provides the basis for comparing $J_N^{\text{nmult}}(x_0)$ to $J_N^{\text{pmult}}(x_0)$. This comparison is then stated in the subsequent corollary.

Theorem 3.3.2. *Assume $x_{j,j,0} \in \mathbb{X}$ for all $j = 0, \dots, N-1$. For $m = 1, \dots, N-1$,*

$$\left| V_N(x_{0,0,0}) - \sum_{j=0}^{N-1} \lambda_{j,m,0} \right| \leq \sum_{j=1}^{m} \Big| J_{N-j}(x_{j,j-1,0}, u^*(\cdot + j)) - J_{N-j}(x_{j,j,0}, u^*(\cdot + j)) \Big| \tag{3.7}$$

Proof. Let $u^* = u^*_{0,0,0}$. First, for each time step, we compare the total cost along nominal trajectory to the trajectory where perturbation is introduced in the next time step wherein optimization is performed. Using (3.5) and (3.6), we obtain the following identities and inequalities.

$$\begin{aligned} \left| \sum_{j=0}^{N-1} \lambda_{j,0,0} - \sum_{j=0}^{N-1} \lambda_{j,1,0} \right| &= \left| \lambda_{0,0,0} + \sum_{j=1}^{N-1} \lambda_{j,0,0} - \lambda_{0,1,0} - \sum_{j=1}^{N-1} \lambda_{j,1,0} \right| \\ &= |J_{N-1}(x_{1,0,0}, u^*(\cdot + 1)) - J_{N-1}(x_{1,1,0}, u^*(\cdot + 1))| \end{aligned}$$

and

$$\begin{aligned} \Big| \sum_{j=0}^{N-1} \lambda_{j,1,0} - \sum_{j=0}^{N-1} \lambda_{j,2,0} \Big| &= \Big| \lambda_{0,1,0} + \lambda_{1,1,0} + \sum_{j=2}^{N-1} \lambda_{j,1,0} \\ &\qquad - \lambda_{0,2,0} - \lambda_{1,2,0} - \sum_{j=2}^{N-1} \lambda_{j,2,0} \Big| \\ &= \Big| J_{N-2}(x_{2,1,0}, u^*(\cdot + 2)) \\ &\qquad - J_{N-2}(x_{2,2,0}, u^*(\cdot + 2)) \Big| \end{aligned}$$

Inductively, for $m = 1, \dots, N-1$,

$$\begin{aligned} \left| \sum_{j=0}^{N-1} \lambda_{j,m-1,0} - \sum_{j=0}^{N-1} \lambda_{j,m,0} \right| &= \Big| \lambda_{0,m-1,0} + \dots + \lambda_{m-1,m-1,0} \\ &\qquad + \sum_{j=m}^{N-1} \lambda_{j,m-1,0} - \lambda_{0,m,0} - \dots \\ &\qquad - \lambda_{m-1,m,0} - \sum_{j=m}^{N-1} \lambda_{j,m,0} \Big| \\ &= \Big| J_{N-m}(x_{m,m-1,0}, u^*(\cdot + m)) \\ &\qquad - J_{N-m}(x_{m,m,0}, u^*(\cdot + m)) \Big| \end{aligned}$$

With these above, for $m = 1, \ldots, N-1$,

$$\begin{aligned}
\left|\sum_{j=0}^{N-1} \lambda_{j,0,0} - \sum_{j=0}^{N-1} \lambda_{j,m,0}\right| &= \left|\sum_{j=0}^{N-1} \lambda_{j,0,0} - \sum_{j=0}^{N-1} \lambda_{j,1,0} + \sum_{j=0}^{N-1} \lambda_{j,1,0} - \right. \\
&\qquad \left. \ldots + \sum_{j=0}^{N-1} \lambda_{j,m-1,0} - \sum_{j=0}^{N-1} \lambda_{j,m,0}\right| \\
&\leq \left|\sum_{j=0}^{N-1} \lambda_{j,0,0} - \sum_{j=0}^{N-1} \lambda_{j,1,0}\right| + \\
&\qquad \ldots + \left|\sum_{j=0}^{N-1} \lambda_{j,m-1,0} - \sum_{j=0}^{N-1} \lambda_{j,m,0}\right| \\
&= \Big| J_{N-1}(x_{1,0,0}, u^*(\cdot+1)) \\
&\qquad - J_{N-1}(x_{1,1,0}, u^*(\cdot+1)) \Big| + \\
&\quad \Big| J_{N-2}(x_{2,1,0}, u^*(\cdot+2)) \\
&\qquad - J_{N-2}(x_{2,2,0}, u^*(\cdot+2)) \Big| \\
&\quad + \ldots + \Big| J_{N-m}(x_{m,m-1,0}, u^*(\cdot+m)) \\
&\qquad - J_{N-m}(x_{m,m,0}, u^*(\cdot+m)) \Big| \\
&= \sum_{j=1}^{m} \Big| J_{N-j}(x_{j,j-1,0}, u^*(\cdot+j)) \\
&\qquad - J_{N-j}(x_{j,j,0}, u^*(\cdot+j)) \Big|
\end{aligned}$$

□

The theorem above estimates the difference between the nominal value and the value corresponding to a trajectory that experiences perturbations up until time instant m. The difference depends on the objective functions of elements $x_{j,j-1,0}$ and $x_{j,j,0}$ as sketched in Figure 3.1.

Using uniform continuity assumptions on the objective, the following statement directly follows.

Corollary 3.3.3. *Suppose J_i, $i = 1, \ldots, N$, is uniformly continuous uniformly in u on $\mathbb{X}$ with modulus of continuity ω_{J_i}. Consider an initial value $x_0 \in \mathbb{X}$ and a perturbation sequence $d(\cdot)$ such that $x_{j,j,0} \in \mathbb{X}$ for all $j = 0, \ldots, N-1$. Then*

$$\left| J_N^{nmult}(x_0) - J_N^{pmult}(x_0) \right| \leq \sum_{j=1}^{N-1} \omega_{J_{N-j}} \left(\|d(j)\| \right) \tag{3.8}$$

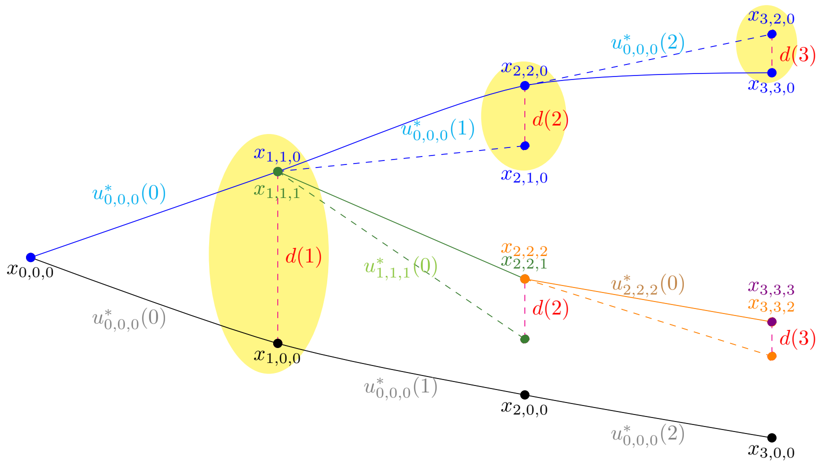

Figure 3.1: Dependence on the value functions of elements $x_{j,j-1,0}$ and $x_{j,j,0}$

Proof. The statement follows from Theorem 3.3.2 applied with $m = N-1$ observing that $J_N^{\text{nmult}} = V_N$ and $x_{j,j,0} - x_{j,j-1,0} = d(j)$. □

Next, we provide the analogous analysis for comparing $J_N^{\text{nmult}}(x_0)$ to $J_N^{\text{upd}}(x_0)$.

Theorem 3.3.4. *Assume $\mathcal{P}_{N-j}(x_{j,j,j})$, $j = 0, \dots, N-1$ is feasible. For $m = 1, \dots, N-1$,*

$$\left| V_N(x_{0,0,0}) - \sum_{j=0}^{N-1} \lambda_{j,m,m} \right| \leq \sum_{j=1}^{m} |V_{N-j}(x_{j,j-1,j-1}) - V_{N-j}(x_{j,j,j})| \tag{3.9}$$

Proof. Similar to the proof of Theorem 3.3.4, we obtain the following set of identities.

$$\begin{aligned} \left| \sum_{j=0}^{N-1} \lambda_{j,0,0} - \sum_{j=0}^{N-1} \lambda_{j,1,1} \right| &= \left| \lambda_{0,0,0} + \sum_{j=1}^{N-1} \lambda_{j,0,0} - \lambda_{0,1,1} - \sum_{j=1}^{N-1} \lambda_{j,1,1} \right| \\ &= |V_{N-1}(x_{1,0,0}) - V_{N-1}(x_{1,1,1})| \end{aligned}$$

$$\begin{aligned} \left| \sum_{j=0}^{N-1} \lambda_{j,1,1} - \sum_{j=0}^{N-1} \lambda_{j,2,2} \right| &= \Bigg| \lambda_{0,1,1} + \lambda_{1,1,1} + \sum_{j=2}^{N-1} \lambda_{j,1,1} \\ &\qquad - \lambda_{0,2,2} - \lambda_{1,2,2} - \sum_{j=2}^{N-1} \lambda_{j,2,2} \Bigg| \\ &= |V_{N-2}(x_{2,1,1}) - V_{N-2}(x_{2,2,2})| \end{aligned}$$

Inductively, for $m = 1, \ldots, N-1$,

$$\begin{aligned}
\left|\sum_{j=0}^{N-1} \lambda_{j,m-1,m-1} - \sum_{j=0}^{N-1} \lambda_{j,m,m}\right| &= \Big| \lambda_{0,m-1,m-1} + \ldots + \lambda_{m-1,m-1,m-1} \\
&\quad + \sum_{j=m}^{N-1} \lambda_{j,m-1,m-1} - \lambda_{0,m,m} - \ldots \\
&\quad - \lambda_{m-1,m,m} - \sum_{j=m}^{N-1} \lambda_{j,m,m} \Big| \\
&= |V_{N-m}(x_{m,m-1,m-1}) - V_{N-m}(x_{m,m,m})|
\end{aligned}$$

Now with the aid of the identities above, we have the following estimate. For $m = 1, \ldots, N-1$,

$$\begin{aligned}
\left|\sum_{j=0}^{N-1} \lambda_{j,0,0} - \sum_{j=0}^{N-1} \lambda_{j,m,m}\right| &= \Big| \sum_{j=0}^{N-1} \lambda_{j,0,0} - \sum_{j=0}^{N-1} \lambda_{j,1,1} + \sum_{j=0}^{N-1} \lambda_{j,1,1} - \ldots \\
&\quad \ldots + \sum_{j=0}^{N-1} \lambda_{j,m-1,m-1} - \sum_{j=0}^{N-1} \lambda_{j,m,m} \Big| \\
&\leq \left| \sum_{j=0}^{N-1} \lambda_{j,0,0} - \sum_{j=0}^{N-1} \lambda_{j,1,1} \right| + \\
&\quad \ldots + \left| \sum_{j=0}^{N-1} \lambda_{j,m-1,m-1} - \sum_{j=0}^{N-1} \lambda_{j,m,m} \right| \\
&= |V_{N-1}(x_{1,0,0}) - V_{N-1}(x_{1,1,1})| + \\
&\quad |V_{N-2}(x_{2,1,1}) - V_{N-2}(x_{2,2,2})| + \\
&\quad \ldots + \Big| V_{N-m}(x_{m,m-1,m-1}) \\
&\qquad -V_{N-m}(x_{m,m,m}) \Big| \\
&= \sum_{j=1}^{m} |V_{N-j}(x_{j,j-1,j-1}) - V_{N-j}(x_{j,j,j})|
\end{aligned}$$

□

The preceding theorem, on the other hand, estimates the difference between the nominal value and the value corresponding to a trajectory that have undergone perturbations and re-optimizations up until time instant m. The difference depends on the optimal value functions of elements $x_{j,j-1,j-1}$ and $x_{j,j,j}$ as sketched in Figure 3.2.

Using uniform continuity assumptions on the optimal value function we arrive

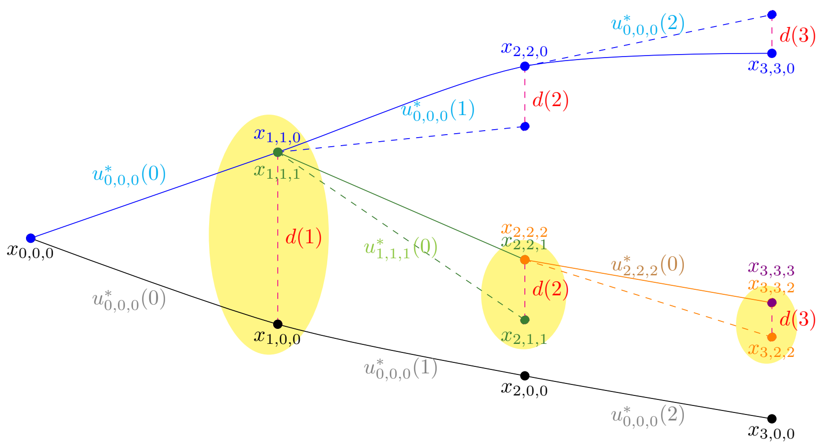

Figure 3.2: Dependence on the optimal value functions of elements $x_{j,j-1,j-1}$ and $x_{j,j,j}$

at the following corollary.

Corollary 3.3.5. *Suppose V_i, $i = 1, \dots, N$, is uniformly continuous on $\mathbb{X}$ with modulus of continuity ω_{V_i}. Consider an initial value $x_0 \in \mathbb{X}$ and a perturbation sequence $d(\cdot)$ such that $\mathcal{P}_{N-j}(x_{j,j,j})$, $j = 0, \dots, N-1$ is feasible. Then*

$$\left| J_N^{nmult}(x_0) - J_N^{upd}(x_0) \right| \leq \sum_{j=1}^{N-1} \omega_{V_{N-j}} \left(\|d(j)\| \right) \tag{3.10}$$

Proof. The statement follows immediately from Theorem 3.3.4 applied with $m = N-1$ observing that $J_N^{\text{nmult}} = V_N$ and $x_{j,j,j} - x_{j,j-1,j-1} = d(j)$. $\square$

Summarizing the results, the analysis reveals that the difference between re-optimizing and not re-optimizing can be quantitatively expressed by the difference between the moduli of continuity ω_{V_i} of the optimal value functions compared to the moduli of continuity ω_{J_i} of the objective functions J_i. Indeed, while the difference between J_N^{nmult} and J_N^{upd} is determined by the ω_{V_i}, the difference between J_N^{nmult} and J_N^{pmult} depends on the ω_{J_i}. In Theorem 1.2.4, we have already seen that $\omega_{V_i} \leq \omega_{J_i}$ holds, which implies that re-optimization should not worsen the performance — modulo the conservatism introduced in our analysis due to the triangle inequalities used in the proofs of Theorems 3.3.2 and 3.3.4.

3.4 Improvement due to re-optimization

The analysis so far reports that re-optimization will "not worsen" the performance of the control design. However, in practice, we hope to determine when does re-optimization not only "not worsen" the performance but rather when does re-optimization improve the situation. For this reason, in this section, we analyze the moduli of continuity obtained for linear quadratic problems in order to identify situations in which an improvement due to re-optimization can indeed be expected.

To this end, we consider linear finite dimensional systems of the form

$$x(k+1) = Ax(k) + Bu(k)$$

with $X = \mathbb{X} = \mathbb{R}^n$, $U = \mathbb{U} = \mathbb{R}^m$ and matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$. The stage cost is given by the quadratic function

$$\ell(x,u) = x^T Qx + u^T Ru$$

with symmetric and positive definite matrices $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$.

The simplifying assumptions of linear dynamics, positive definite quadratic costs and no constraints are mainly imposed in order to simplify the presentation of the two key properties controllability and stability in this section. Similar results can also be obtained for nonlinear and constrained problems at the expense of more technically involved definitions and proofs.

We first estimate the modulus of continuity ω_{J_N}.

Proposition 3.4.1. *Let σ be the eigenvalue of A with maximal modulus $|\sigma|$. Let $S \subset \mathbb{R}^n$ be a bounded set, $N \in \mathbb{N}$ and $\varepsilon > 0$. For a constant $K > 0$ consider the set of control sequences*

$$\mathbb{U}_K^N := \{u(\cdot) \in \mathbb{U}^N \mid \|u(k)\| \le K \text{ for all } k = 0,\dots,N-1\}. \tag{3.11}$$

Then there exists real constants $c_1 > 0$ and $c_2 = c_2(\varepsilon) > 0$ such that the modulus of continuity ω_{J_N} of J_N on S, uniformly in $u(\cdot) \in \mathbb{U}_K^N$ satisfies

$$c_1 r^2 \sum_{k=0}^{N-1} |\sigma|^{2k} \le \omega_{J_N}(r) \le c_2 \sum_{k=0}^{N-1} |\sigma|^k r.$$

Proof. For any two initial values x_1, $x_2 \in \mathbb{R}^n$ and any control sequence $u(\cdot) \in \mathbb{U}^N$, observe

$$\begin{aligned} e(k) &:= x_u(k,x_2) - x_u(k,x_1) = A^k x_2 - A^k x_1 \\ &= A^k(x_2 - x_1) = A^k e(0) \end{aligned}$$

Setting $x_1 := 0$ and $x_2 := rv$ where v is an eigenvector for σ with $\|v\| = 1$ then yields $e(0) = rv$ and thus $e(k) = \sigma^k rv$. Since Q is positive definite there exists $c_1 > 0$ with $v^T Qv = c_1$. Then for $u(\cdot) :\equiv 0$ we obtain

$$\begin{aligned} \ell(x_u(k,x_2),u(k)) - \ell(x_u(k,x_1),u(k)) &= x_u(k,x_2)^\top Q x_u(k,x_2) \\ &\quad - x_u(k,x_1)^\top Q x_u(k,x_1) \\ &= e(k)^T Q e(k) = \sigma^k r v^T Q v r \sigma^k \\ &= c_1 r^2 (\sigma^k)^2 \end{aligned}$$

Since (1.5) holds for all $u(\cdot) \in \mathbb{U}^N$, by choosing $u(\cdot) \equiv 0$, it follows that

$$\begin{aligned} \omega_{J_N}(\|x_1 - x_2\|) &\geq \left\| \sum_{k=0}^{N-1} (\ell(x_u(k,x_2),u(k)) - \ell(x_u(k,x_1),u(k))) \right\| \\ &= c_1 r^2 \sum_{k=0}^{N-1} |\sigma|^{2k} \end{aligned}$$

This yields the lower bound.

To show the upper bound, we use the fact that on one hand, for $\varepsilon > 0$ there exists $\tilde{c}_2 > 0$ such that $\|A^k x\| \leq \tilde{c}_2(|\sigma| + \varepsilon)^k \|x\|$ holds (this follows, e.g., from [59, Satz 11.6]). On the other hand, there exists a compact set $D \subset \mathbb{R}^n$ such that for all $x_0 \in S$ and all $u(\cdot) \in \mathbb{U}_K^N$ the inclusion $x_u(k,x_0) \in D$ holds for all $k = 0, \dots, N-1$. On this set D, the stage cost ℓ is Lipschitz continuous in x with a constant $L > 0$, i.e.,

$$\begin{aligned} \|\ell(x_u(k,x_2),u(k)) - \ell(x_u(k,x_1),u(k))\| &\leq L\left(\|x_u(k,x_2) - x_u(k,x_1)\|\right) \\ &= L\|e(k)\| \;=\; L\|A^k e(0)\| \\ &= L\|A^k(x_2 - x_1)\| \\ &\leq L\tilde{c}_2(|\sigma| + \varepsilon)^k \|x_2 - x_1\| \end{aligned}$$

for all $x_1, x_2 \in S$, leading to

$$\begin{aligned} \|J_N(x_1,u(\cdot)) - J_N(x_2,u(\cdot))\| &\leq \sum_{k=0}^{N-1} \|\ell(x_u(k,x_2),u(k)) - \ell(x_u(k,x_1),u(k))\| \\ &\leq c_3\|x_2 - x_1\| \end{aligned}$$

with $c_3 = L\tilde{c}_2 \sum_{k=0}^{N-1} |\sigma|^k$. Since ω_{J_N} is the modulus of continuity of J_N, it must be that

$$\omega_{J_N}(\|x_2 - x_1\|) \leq c_3\|x_2 - x_1\|$$

This yields the claimed upper bound with $c_2 = L\tilde{c}_2$. □

Observe that the lower bound on $\omega_{J_N}(r)$ is independent of the choice of S, ε, K and N while the upper bound typically depends on these parameters.

Proposition 3.4.1 states that the modulus of continuity ω_{J_N} is large whenever $|\sigma|$ is large and small if $|\sigma|$ is small. In particular, ω_{J_N} grows unboundedly in N if the system is not open-loop asymptotically stable, i.e., if $|\sigma| \geq 1$.

From Theorem 1.2.4, we have $\omega_{V_N} \leq \omega_{J_N}$. Hence the upper bound on ω_{J_N} from Proposition 3.4.1 also applies to ω_{V_N}. In addition, under suitable conditions, ω_{V_N} can be considerably smaller than ω_{J_N}, as the following proposition shows.

Proposition 3.4.2. *Assume that the pair (A, B) is controllable. Let $S \subset \mathbb{R}^n$ be a bounded set. Then there exists a real constant $c > 0$ such that the modulus of continuity ω_{V_N} on S satisfies*

$$\omega_{V_N}(r) \leq cr$$

for all $N \in \mathbb{N}$.

Proof. Controllability implies that there exists a constant $\tilde{c} > 0$ such that for any $x_0 \in \mathbb{R}^n$ we can find a control $u_{x_0}(\cdot) \in \mathbb{U}^{\tilde{n}}$ with $\|u_{x_0}(k)\| \leq \tilde{c}\|x_0\|$ for all $k = 0, \ldots, \tilde{n}-1$ and $x_{u_{x_0}}(\tilde{n}, x_0) = 0$. This implies that on the bounded set S there exists a uniform upper bound M of V_N which can be chosen independent of N. Then, positive definiteness of Q and R implies that the optimal trajectories remain in a compact set D and that the optimal control sequences lie in the set $\mathbb{U}_K^N$ from (3.11), where D and K can also be chosen independent of N.

Now for $N \leq \tilde{n}$, using Proposition 3.4.2 in conjunction with Theorem 1.2.4 we have

$$\begin{aligned} \|V_N(x_1) - V_N(x_2)\| &\leq \omega_{V_N}(\|x_1 - x_2\|) \;\leq\; \omega_{J_N}(\|x_1 - x_2\|) \\ &\leq c_2 \sum_{k=0}^{N-1} |\sigma|^k \|x_1 - x_2\| \end{aligned}$$

The assertion follows with $c = c_2 \sum_{k=0}^{N-1} |\sigma|^k$.

For $N > \tilde{n}$, consider two initial values $x_1, x_2 \in S$ and let $u^\star(\cdot)$ be the optimal control for x_1. Let $x_0 := x_2 - x_1$ and pick the control sequence $u_{x_0} \in \mathbb{U}^{\tilde{n}}$ from the controllability property, which we extend with $u_{x_0}(k) := 0$ for all $k = \tilde{n}, \ldots, N-1$, implying $x_{u_{x_0}}(k, x_2 - x_1) = 0$ for all $k = \tilde{n}, \ldots, N-1$. Then for $\tilde{u}^\star = u^\star + u_{x_0}$, we get

$$x_{\tilde{u}^\star}(k, x_2) = x_{u^\star}(k, x_1) + x_{u_{x_0}}(k, x_2 - x_1) = x_{u^\star}(k, x_1)$$

for all $k \geq \tilde{n}$. Since ℓ is Lipschitz on $S \times \mathbb{U}_K$, we can find a constant $\hat{c} > 0$ such that

$$\ell(x_{\tilde{u}^\star}(k, x_2), \tilde{u}^\star(k)) - \ell(x_{u^\star}(k, x_1), u^\star(k)) \leq \hat{c}\|x_2 - x_1\|$$

for all $k = 0, \ldots, \tilde{n}-1$, while for $k \geq \tilde{n}$ this difference equals 0. Therefore,

$$\begin{aligned} \|V_N(x_2) - V_N(x_1)\| &\leq \|J_N(x_2, \tilde{u}^\star(\cdot)) - J_N(x_1, u^\star(\cdot))\| \\ &\leq \left\| \sum_{k=0}^{N-1} \ell(x_{\tilde{u}^\star}(k, x_2), \tilde{u}^\star(k)) - \sum_{k=0}^{N-1} \ell(x_{u^\star}(k, x_1), u^\star(k)) \right\| \\ &\leq \sum_{k=0}^{N-1} \hat{c}\|x_2 - x_1\| \;=\; \tilde{n}\hat{c}\|x_2 - x_1\| \end{aligned}$$

With $c = \tilde{n}\hat{c}$, this implies the desired estimate. □

Remark 3.4.3. As a consequence of the results above, we expect the difference between ω_{J_N} and ω_{V_N} to be particularly large when the system is open-loop unstable (implying a large ω_{J_N}) and controllable (implying a small ω_{V_N}).

In the next section, we present examples which numerically illustrate the result in Remark 3.4.3.

3.5 Numerical example: a linear quadratic problem

Here we consider an illustrative numerical example for which we compare the nominal case J_N^{nmult}, the case when nominal solution is applied to perturbed systems J_N^{pmult} and the shrinking horizon MPC J_N^{upd} where the re-optimization is carried out at each time step due to the mentioned perturbation. Consider the nominal system described by

$$x^+ = \alpha x + u \tag{3.12}$$

and the corresponding perturbed system

$$x^+ = \alpha x + u + d \tag{3.13}$$

where d is an additive perturbation. Consider the cost function

$$\ell(x, u) = x^2 + u^2.$$

Note that the stage cost ℓ forces the optimal trajectory to converge to the origin 0. Hence, the distance of the perturbed trajectory from the origin can be used to visualize the performance.

If $|\alpha| < 1$, then (3.12) is asymptotically stable with 0 as the equilibrium, and for $|\alpha| > 1$, it is unstable. In both cases, the system is controllable. Taking final time $N = 7$, Figure 3.1 provides a visualization of the trajectories throughout time for a chosen α and $x_0 = -4$. With $i = 0, \dots N$, $x_{i,0,0}$ represents the nominal trajectory related to $J_N^{\text{nmult}}(x_0)$, while $x_{i,i,0}$ denotes the trajectory corresponding to $J_N^{\text{pmult}}(x_0)$, i.e., when the nominal open-loop control is applied to the perturbed system (3.13). Finally, $x_{i,i,i}$ represents the trajectory with re-optimization, corresponding to $J_N^{\text{upd}}(x_0)$. The perturbations $d(\cdot)$ are randomly generated from the interval $[-0.1, 0.1]$.

We compare the three described trajectories in Figure 3.1. Figure 3.1(top) illustrates the case when $\alpha = 0.5$ for which (3.12) is is open-loop stable. Therefore, one would expect not much improvement from re-optimization, which is exactly what is visible in the figure, as the deviation from the nominal solution is only mildly improved by re-optimization. In contrast to this, Figure 3.1(bottom) shows the case $\alpha = 1.5$, in which the system is open-loop unstable and controllable. Here, our analysis predicts a large benefit of the re-optimization procedure which is clearly visible in the simulation. The similar effect is visible in Table 3.5.1 in which the values of $J_N^{\text{nmult}}(x_0)$, $J_N^{\text{pmult}}(x_0)$ and $J_N^{\text{upd}}(x_0)$ for $x_0 = -4$ are shown. In the open-loop unstable and controllable system with $\alpha = 1.5$, one can notice a better performing $J_N^{\text{upd}}(x_0)$ compared to $J_N^{\text{nmult}}(x_0)$. This is due to the fact that the introduced random perturbations here do by chance have a positive effect on the performance because they drive the system faster towards 0.

Figure 3.2 and Table 3.5.2 illustrate a case when re-optimization does not give much benefit because the system is not controllable. In this example, we set $\alpha = 1.5$ and impose a control constraint $u \geq 0$ which renders the system uncontrollable. Compared to Figure 3.1(bottom), one sees that the performance of the re-optimization significantly deteriorates, though it still provides some improvement over using the open-loop optimal trajectory. The numerical values in Table 3.5.2 confirm this behavior. In order to increase the visibility of this effect, we used here the constant perturbations $d(k) = 0.1$, i.e., the maximum positive additive perturbation, at each time step.

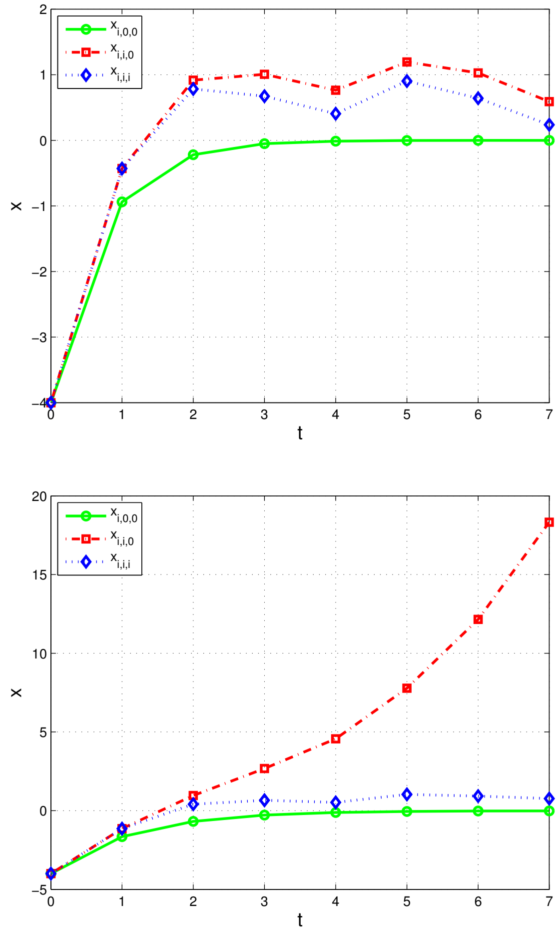

Figure 3.1: State trajectories for the stable and controllable system with $\alpha = 0.5$ (top) and for the unstable and controllable system with $\alpha = 1.5$ (bottom)

Table 3.5.1: Comparison of control scheme performance

	$\alpha = 0.5$	$\alpha = 1.5$
$J_N^{\text{nmult}}(x_0)$	18.1245	42.0829
$J_N^{\text{pmult}}(x_0)$	22.6457	613.1214
$J_N^{\text{upd}}(x_0)$	18.8812	24.8458

Table 3.5.2: Comparison of control scheme performance

	$\alpha = 1.5$
$J_N^{\text{nmult}}(x_0)$	42.0829
$J_N^{\text{pmult}}(x_0)$	1763.9
$J_N^{\text{upd}}(x_0)$	581.7244

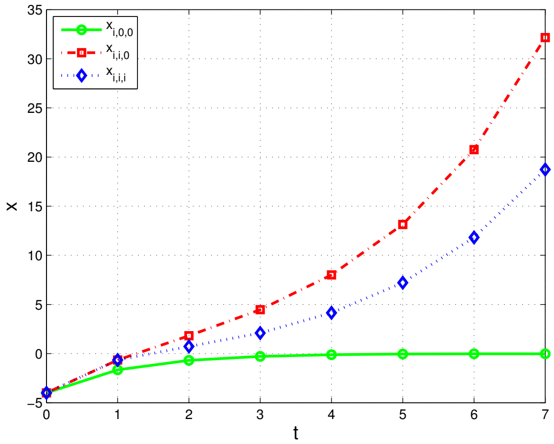

Figure 3.2: State trajectories for the unstable and uncontrollable system with $\alpha = 1.5$ with constraint $u \geq 0$ and maximum positive perturbation at each time step

4 Multistep and updated multistep MPC schemes

The benefits brought by re-optimization to counteract perturbations is presented in Chapter 3 for finite horizon problems. It is shown that over the finite horizon N, the performance difference between the nominal and perturbed system controlled by the nominal optimal control is determined by ω_{J_N} while the difference between the nominal and the shrinking horizon updated feedback controller is determined by ω_{V_N}. For open loop unstable and controllable systems, wherein ω_{V_N} is considerably smaller than ω_{J_N}, the benefit of updates becomes significant.

This chapter builds upon the framework of Chapter 3 and extends the results to the infinite horizon problem (approximately) solved by MPC through a moving horizon approach. We show in this chapter that the re-optimization in shrinking horizon update strategy performed in the updated m-step MPC does indeed allow for improved stability and performance estimates compared to non-updated scheme. We focus our attention to the evolution described by the perturbed multistep MPC closed-loop system

$$\tilde{x}_{\mu_{N,m}}(k+1) = f(\tilde{x}_{\mu_{N,m}}(k), \mu_{N,m}(\tilde{x}_{\mu_{N,m}}(\lfloor k \rfloor_m), k)) \ + \ d(k) \tag{4.1}$$

and the perturbed updated multistep MPC closed-loop system

$$\tilde{x}_{\hat{\mu}_{N,m}}(k+1) = f(\tilde{x}_{\hat{\mu}_{N,m}}(k), \hat{\mu}_{N,m}(\tilde{x}_{\hat{\mu}_{N,m}}(k), k)) \ + \ d(k) \tag{4.2}$$

where perturbation occurs and re-optimization is performed. The feedback controls $\mu_{N,m}$ and $\hat{\mu}_{N,m}$ are defined in (1.7) and (1.9), respectively.

Section 4.1 and Section 4.2 give statements analogous to some properties in Section 2.1 extended to MPC trajectories for which perturbations and possibly re-optimizations occur. These statements serve as ingredients for the main stability and performance results formulated and proved in Section 4.3 for the multistep and the updated multistep MPC. Lastly, an illustrative numerical example is given in Section 4.4. A preliminary version of the results we present in this chapter is published in [33].

4.1 Properties resulting from perturbations and re-optimizations

This section provides a counterpart of Proposition 2.1.7 for the perturbed closed-loop. To this end, we again make use of the notation introduced in Section 3.2

and derive a number of inequalities along the different trajectories.

4.1.1 Estimates involving $V_N(x_{m,m,0})$ and $V_N(x_{m,m,m})$

We derive in this subsection some implications of Assumption 2.1.4 on trajectories involving occurrence of perturbation and re-optimization. The following lemmas provide an upper bound for $V_N(x_{m,m,0})$ and for $V_N(x_{m,m,m})$ which can be viewed as extensions of (2.5) to the perturbed setting.

Lemma 4.1.1. *Let Assumption 2.1.4 hold and consider $x_{0,0,0} = x \in \mathbb{X}$ and an optimal control $u^*(\cdot) \in \mathbb{U}^N$ for the finite horizon optimal control problem $\mathcal{P}_N(x)$ with optimization horizon N. Then for each $m = 1, \ldots, N-1$ and each $j = 0, \ldots, N-m-1$,*

$$V_N(x_{m,m,0}) \leq \sum_{n=0}^{j-1} \lambda_{n+m,m,0} + B_{N-j}(\lambda_{j+m,m,0}) \tag{4.3}$$

$$V_N(x_{m,m,m}) \leq \sum_{n=0}^{j-1} \lambda_{n+m,m,m} + B_{N-j}(\lambda_{j+m,m,m}) \tag{4.4}$$

Proof. To show (4.3), we take the trajectory element $x_{m,m,0}$ whose evolution is steered by the optimal control $u^*(\cdot)$ along the perturbed system (4.1) within m-steps. We consider $x_{j+m,m,0}$ for some $j \in \{0, \ldots, N-m-1\}$.

We define

$$\tilde{u}(n) = \begin{cases} u^*(n+m) & n \in \{0, \ldots, j-1\} \\ u_{\check{x}}(n-j) & n \in \{j, \ldots, N-1\} \end{cases} \tag{4.5}$$

where $u_{\check{x}}(\cdot)$ results from solving the optimization problem $\mathcal{P}_{N-j}(\check{x})$ with initial value $\check{x} = x_{j+m,m,0} = x_{u^*(\cdot+m)}(j, x_{m,m,0})$. This yields

$$\begin{aligned} V_N(x_{m,m,0}) &\leq J_N(x_{m,m,0}, \tilde{u}(\cdot)) \\ &= J_j(x_{m,m,0}, u^*(\cdot+m)) + J_{N-j}(x_{j+m,m,0},\ u_{\check{x}}(\cdot)) \\ &= \sum_{n=0}^{j-1} \ell(x_{n+m,m,0}, u^*(n+m)) + \sum_{n=0}^{N-j-1} \ell(x_{u_{\check{x}}}(n,\check{x}), u_{\check{x}}(n)) \\ &= \sum_{n=0}^{j-1} \lambda_{n+m,m,0} + V_{N-j}(\check{x}) \ \leq \sum_{n=0}^{j-1} \lambda_{n+m,m,0} + B_{N-j}(\ell^*(\check{x})) \\ &= \sum_{n=0}^{j-1} \lambda_{n+m,m,0} + B_{N-j}(\lambda_{j+m,m,0}). \end{aligned}$$

To show (4.4), we proceed analogously with $\check{x} = x_{j+m,m,m} = x_{u_{m,m,m}}(j, x_{m,m,m})$. □

4.1.2 Estimates involving uniform continuity

The following are generalizations of Theorems 3.3.2 and 3.3.4 allowing an arbitrary time instant $k \in \{0, 1, \ldots, N-1\}$ to be the reference point in place of $k = 0$. These results eventually provide a basis for comparing, in the finite horizon OCP setting, the nominal system, the perturbed system controlled by the nominal optimal control and the perturbed system under the shrinking horizon updated feedback controller.

Theorem 4.1.2. *Given $k \in \{0, \ldots, N-1\}$. For any $p \in \{1, \ldots, N-k-1\}$,*

$$\left| \sum_{j=k}^{N-1} \lambda_{j,k,0} - \sum_{j=k}^{N-1} \lambda_{j,k+p,0} \right| \leq \sum_{j=1}^{p} |J_{N-k-j}(x_{k+j,k+j-1,0}, u^*(\cdot + k + j)) - J_{N-k-j}(x_{k+j,k+j,0}, u^*(\cdot + k + j))| \quad (4.6)$$

and

$$\left| \sum_{j=k}^{N-1} \lambda_{j,k,k} - \sum_{j=k}^{N-1} \lambda_{j,k+p,k+p} \right| \leq \sum_{j=1}^{p} |V_{N-k-j}(x_{k+j,k+j-1,k+j-1}) - V_{N-k-j}(x_{k+j,k+j,k+j})| \quad (4.7)$$

Proof. The proof follows using the same technique as the proofs of Theorems 3.3.2 and 3.3.4 with the appropriate changes in the indices. □

Following directly is a corollary that sizes up the differences among values associated with the tails of the nominal trajectory, the tails of the perturbed trajectory with nominal control and the tails of the perturbed trajectory with re-optimized control.

Corollary 4.1.3. *Let $k \in \{0, \ldots, N-1\}$. Suppose J_i, $i = 1, \ldots, N$, is uniformly continuous on a set A containing $x_{j,k,0}$ and $x_{j,j,0}$ for $j = k, \ldots, N-1$ uniformly in u on $\mathbb{X}$ with modulus of continuity ω_{J_i}. Suppose V_i, $i = 1, \ldots, N$, is uniformly continuous on a set A containing $x_{j,k,k}$ and $x_{j,j,j}$ for $j = k, \ldots, N-1$ with modulus of continuity ω_{V_i}. Then*

$$\left| \sum_{j=k}^{N-1} \lambda_{j,k,0} - \sum_{j=k}^{N-1} \lambda_{j,j,0} \right| \leq \sum_{j=1}^{N-k-1} \omega_{J_{N-k-j}} (\|d(k+j)\|) \quad (4.8)$$

and

$$\left| \sum_{j=k}^{N-1} \lambda_{j,k,k} - \sum_{j=k}^{N-1} \lambda_{j,j,j} \right| \leq \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}} (\|d(k+j)\|) \quad (4.9)$$

Proof. Straightforward from (4.6) and (4.7) with $p = N - k - 1$. □

Note that on the right hand side of the estimates the perturbations that occur before time step k do not appear since in both schemes they have cancelled each other. Also, for the special case of $k = 0$, we recover results of Corollaries 3.3.3 and 3.3.5.

In the next lemma, we combine the preceding results to derive an upper bound for the values corresponding to the tails of the perturbed trajectory with nominal control and for the tails of the perturbed trajectory with re-optimized control. Resulting estimates can be viewed as extensions of (2.4) to the perturbed setting.

Lemma 4.1.4. *Let the assumptions of Corollary 4.1.3 hold. Suppose further B_K, $K = 1, \dots, N$, is uniformly continuous on $\mathbb{R}_0^+$ with modulus of continuity ω_{B_K}. Then for $k = 0, \dots, N-2$, we have the inequalities*

$$\begin{aligned}
\sum_{j=k}^{N-1} \lambda_{j,j,0} \quad &\leq \quad B_{N-k}(\lambda_{k,k,0}) + \omega_{B_{N-k}}(\lambda_{k,k,0} - \lambda_{k,0,0}) \\
&\quad + \omega_{J_{N-k}}(x_{k,k,0} - x_{k,0,0}) + \sum_{j=1}^{N-k-1} \omega_{J_{N-k-j}}\left(\|d(k+j)\|\right)
\end{aligned} \tag{4.10}$$

$$\sum_{j=k}^{N-1} \lambda_{j,j,j} \quad \leq \quad B_{N-k}(\lambda_{k,k,k}) + \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}}\left(\|d(k+j)\|\right). \tag{4.11}$$

Proof. Inequality (4.10) follows since

$$\begin{aligned}
\sum_{j=k}^{N-1} \lambda_{j,j,0} \quad &\leq \quad \sum_{j=k}^{N-1} \lambda_{j,k,0} + \sum_{j=1}^{N-k-1} \omega_{J_{N-k-j}}\left(\|d(k+j)\|\right) \\
&= \quad J_{N-k}(x_{k,k,0}, u^*_{0,0,0}(k+\cdot)) + \sum_{j=1}^{N-k-1} \omega_{J_{N-k-j}}\left(\|d(k+j)\|\right) \\
&\leq \quad J_{N-k}(x_{k,0,0}, u^*_{0,0,0}(k+\cdot)) + \omega_{J_{N-k}}(x_{k,k,0} - x_{k,0,0}) \\
&\quad + \sum_{j=1}^{N-k-1} \omega_{J_{N-k-j}}\left(\|d(k+j)\|\right) \\
&\leq \quad B_{N-k}(\ell^*(x_{k,0,0})) + \omega_{J_{N-k}}(x_{k,k,0} - x_{k,0,0}) \\
&\quad + \sum_{j=1}^{N-k-1} \omega_{J_{N-k-j}}\left(\|d(k+j)\|\right) \\
&= \quad B_{N-k}(\lambda_{k,0,0}) + \omega_{J_{N-k}}(x_{k,k,0} - x_{k,0,0}) \\
&\quad + \sum_{j=1}^{N-k-1} \omega_{J_{N-k-j}}\left(\|d(k+j)\|\right) \\
&\leq \quad B_{N-k}(\lambda_{k,k,0}) + \omega_{B_{N-k}}(\lambda_{k,k,0} - \lambda_{k,0,0}) \\
&\quad + \omega_{J_{N-k}}(x_{k,k,0} - x_{k,0,0}) + \sum_{j=1}^{N-k-1} \omega_{J_{N-k-j}}\left(\|d(k+j)\|\right).
\end{aligned}$$

To show (4.11) we compute

$$\begin{aligned}
\sum_{j=k}^{N-1} \lambda_{j,j,j} \quad &\leq \quad \sum_{j=k}^{N-1} \lambda_{j,k,k} + \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}}\left(\|d(k+j)\|\right) \\
&= \quad V_{N-k}(x_{k,k,k}) + \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}}\left(\|d(k+j)\|\right) \\
&\leq \quad B_{N-k}(\ell^*(x_{k,k,k})) + \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}}\left(\|d(k+j)\|\right) \\
&= \quad B_{N-k}(\lambda_{k,k,k}) + \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}}\left(\|d(k+j)\|\right).
\end{aligned}$$

□

4.1.3 Counterpart of Proposition 2.1.7

By combining the results of Sections 4.1.1 and 4.1.2, we can now state the following counterpart of Proposition 2.1.7. It yields necessary conditions which hold if these values λ_n coincide with either $\lambda_{n,n,0}$ or $\lambda_{n,n,n}$, $n = 0, \ldots, N-1$, and ν with either $V_N(x_{m,m,0})$ or $V_N(x_{m,m,m})$.

Corollary 4.1.5. *Consider $N \geq 1, m \in \{1, \ldots, N-1\}$ and let the assumptions of Lemmas 4.1.1 and 4.1.4 hold. Let $x = x_{0,0,0} \in \mathbb{X}$ and consider a perturbation sequence $d(\cdot)$ where $d(k) = 0$ for $k \geq m$ generating the trajectories $\tilde{x}_{\mu_{N,N-1}}(n,x) = x_{n,n,0}$ and $\tilde{x}_{\hat{\mu}_{N,N-1}}(n,x) = x_{n,n,n}$. Consider a sequence $\lambda_n \geq 0$, $n = 0, \ldots, N-1$ and a value $\nu \geq 0$ such that either*

(i) $\lambda_n = \lambda_{n,n,0}$, $n = 0, \ldots, N-1$ *and* $\nu = V_N(x_{m,m,0})$ *or*

(ii) $\lambda_n = \lambda_{n,n,n}$, $n = 0, \ldots, N-1$ *and* $\nu = V_N(x_{m,m,m})$ *holds.*

Then the inequalities

$$\sum_{n=k}^{N-1} \lambda_n \leq B_{N-k}(\lambda_k) + \xi_k, \quad k = 0, \ldots, N-2 \tag{4.12}$$

$$\nu \leq \sum_{n=0}^{j-1} \lambda_{n+m} + B_{N-j}(\lambda_{j+m}), \quad j = 0, \ldots, N-m-1 \tag{4.13}$$

hold for

$$\begin{aligned}
(i) \quad \xi_k \quad &= \quad \xi_k^{pmult} \quad := \quad \textstyle\sum_{j=1}^{N-k-1} \omega_{J_{N-k-j}} \left(\|d(k+j)\|\right) \\
&\qquad\qquad + \omega_{B_{N-k}}(\lambda_{k,k,0} - \lambda_{k,0,0}) + \omega_{J_{N-k}}(x_{k,k,0} - x_{k,0,0}) \\
(ii) \quad \xi_k \quad &= \quad \xi_k^{upd} \quad := \quad \textstyle\sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}} \left(\|d(k+j)\|\right).
\end{aligned}$$

Proof. For case (i), inequality (4.13) follows immediately from (4.3) while (4.12) follows directly from (4.10). For case (ii), (4.13) follows from (4.4), and (4.12) from (4.11). □

Remark 4.1.6. We will later use Corollary 4.1.5 in order to establish inequality (2.16). Since all quantities in this inequality only depend on the perturbation values $d(0), \ldots, d(m-1)$, we could make the simplifying assumption $d(k) = 0$ for $k \geq m$ in Corollary 4.1.5.

4.2 The perturbed versions of $\mathcal{P}_\alpha$

Inequalities (2.4) and (2.5) comprise the constraints in the minimization problem $\mathcal{P}_\alpha$ for finding the suboptimality index of the nominal m-step MPC scheme with respect to the infinite horizon problem. For the perturbed and the perturbed updated m-step MPC, the preceding corollary yields analogous 'perturbed' inequalities (4.12) and (4.13). In this section, we investigate how much the values α resulting from the corresponding perturbed versions of $\mathcal{P}_\alpha$ may differ from the nominal case. To this end, we first state the three problems under consideration. Here, for the subsequent analysis it turns out beneficial to include perturbation terms in both inequalities (4.12) and (4.13).

First, the optimization problem $\mathcal{P}_\alpha$ corresponding to the nominal multistep MPC can be written in terms of the latterly introduced notation as

$$\alpha^{\text{nmult}} := \inf_{\lambda_{n,0,0}, n=0,\ldots,N-1, \nu^{\text{nmult}}} \frac{\sum_{n=0}^{N-1} \lambda_{n,0,0} - \nu^{\text{nmult}}}{\sum_{n=0}^{m-1} \lambda_{n,0,0}}$$

$$\text{subject to} \qquad \mathcal{P}_\alpha^{\text{nmult}}$$

$$\sum_{n=k}^{N-1} \lambda_{n,0,0} \leq B_{N-k}(\lambda_{k,0,0}), \quad k = 0, \ldots, N-2$$

$$\nu^{\text{nmult}} \leq \sum_{n=0}^{j-1} \lambda_{n+m,0,0} + B_{N-j}(\lambda_{j+m,0,0}), \quad j = 0, \ldots, N-m-1$$

$$\sum_{n=0}^{m-1} \lambda_{n,0,0} > 0, \ \lambda_{m,0,0}, \ \ldots, \ \lambda_{N-1,0,0}, \nu^{\text{nmult}} \geq 0$$

For the perturbed multistep MPC without update, we define α^{pmult} via

$$\alpha^{\text{pmult}} := \inf_{\lambda_{n,n,0}, n=0,\ldots,N-1, \nu^{\text{pmult}}} \frac{\sum_{n=0}^{N-1} \lambda_{n,n,0} - \nu^{\text{pmult}}}{\sum_{n=0}^{m-1} \lambda_{n,n,0}}$$

$$\text{subject to} \qquad \mathcal{P}_\alpha^{\text{pmult}}$$

$$\sum_{n=k}^{N-1} \lambda_{n,n,0} \leq B_{N-k}(\lambda_{k,k,0}) + \xi^{\text{pmult}}, \quad k = 0, \ldots, N-2$$

$$\nu^{\text{pmult}} \leq \sum_{n=0}^{j-1} \lambda_{n+m,m,0} + B_{N-j}(\lambda_{j+m,m,0}) + \xi^{\text{pmult}}, \quad j = 0, \ldots, N-m-1$$

$$\sum_{n=0}^{m-1} \lambda_{n,n,0} \geq \zeta, \ \lambda_{m,m,0}, \ \ldots, \lambda_{N-1,N-1,0}, \ \nu^{\text{pmult}} \geq 0$$

where

$$\xi^{\text{pmult}} := \max_{k \in \{0,\ldots,N-2\}} \xi_k^{\text{pmult}} \quad \text{with } \xi_k^{\text{pmult}} \text{ from Corollary 4.1.5(i)} \tag{4.14}$$

Finally, for the perturbed updated multistep MPC, we define α^{upd} by

$$\alpha^{\text{upd}} := \inf_{\lambda_{n,n,n}, n=0,\ldots,N-1, \nu^{\text{upd}}} \frac{\sum_{n=0}^{N-1} \lambda_{n,n,n} - \nu^{\text{upd}}}{\sum_{n=0}^{m-1} \lambda_{n,n,n}}$$

subject to $\qquad\qquad \mathcal{P}_\alpha^{\text{upd}}$

$$\sum_{n=k}^{N-1} \lambda_{n,n,n} \le B_{N-k}(\lambda_{k,k,k}) + \xi^{\text{upd}}, \quad k = 0, \ldots, N-2$$

$$\nu^{\text{upd}} \le \sum_{n=0}^{j-1} \lambda_{n+m,m,m} + B_{N-j}(\lambda_{j+m,m,m}) + \xi^{\text{upd}}, \quad j = 0, \ldots, N-m-1$$

$$\sum_{n=0}^{m-1} \lambda_{n,n,n} \ge \zeta, \ \lambda_{m,m,m}, \ \ldots, \lambda_{N-1,N-1,N-1}, \ \nu^{\text{upd}} \ge 0$$

with

$$\xi^{\text{upd}} = \max_{k \in \{0,\ldots,N-2\}} \xi_k^{\text{upd}} \quad \text{with } \xi_k^{\text{upd}} \text{ from Corollary 4.1.5(ii)} \tag{4.15}$$

Remark 4.2.1. The constraint bound $\zeta > 0$ is assigned to prevent the quotients with denominator $\sum_{n=0}^{m-1} \lambda_{n,n,0}$ and $\sum_{n=0}^{m-1} \lambda_{n,n,n}$ appearing in the analysis from blowing up.

The subsequent lemma, inspired by of [36, Lemma 6.32], is the key technical step to show how α^{nmult}, α^{pmult} and α^{upd} are related. It provides an estimate for the difference between the solutions to two abstract optimization problems of the type introduced above.

Lemma 4.2.2. *Consider increasing functions $B_k^i : \mathbb{R}_0^+ \to \mathbb{R}_0^+$ for $k \in \mathbb{N}$ and $i = 1, 2$ for which $B_k^2(r)$ is linear. Assume that these functions satisfy $B_k^i(r) \ge r$ for all $k \in \mathbb{N}, r \ge 0$ and that there exists a real constant $\xi > 0$ with*

$$B_k^1(r) \le B_k^2(r) + \xi \tag{4.16}$$

For $i = 1, 2$ and a constant $\zeta \ge 0$ consider the optimization problems

$$\alpha^i := \inf_{\lambda_0,\ldots,\lambda_{N-1},\nu} \frac{\sum_{n=0}^{N-1} \lambda_n - \nu}{\sum_{n=0}^{m-1} \lambda_n}$$

subject to

$$\sum_{n=k}^{N-1} \lambda_n \le B_{N-k}^i(\lambda_k), \ k = 0, \ldots, N-2 \tag{4.17}$$

$$\nu \le \sum_{n=0}^{j-1} \lambda_{n+m} + B_{N-j}^i(\lambda_{j+m}), \ j = 0, \ldots, N-m-1 \tag{4.18}$$

$$\sum_{n=0}^{m-1} \lambda_n \ge \zeta, \lambda_0, \ldots, \lambda_{N-1}, \nu \ge 0 \tag{4.19}$$

Then the following holds.

(i) If $\zeta > 0$, then the inequality $\alpha^2 \leq \alpha^1 + \dfrac{B^2_{m+1}(\xi) + \xi}{\zeta}$ holds.

(ii) If $\zeta = 0$ and $\alpha^2 \geq 0$, then for all values $\lambda_0, \dots, \lambda_{N-1}, \nu$ satisfying (4.17)–(4.19) for $i = 1$ the inequality $\nu \leq \sum_{n=0}^{N-1} \lambda_n + B^2_{m+1}(\xi) + \xi$ holds.

Proof. (i) Fix $\varepsilon > 0$. Consider ε-optimal values $\lambda^1_0, \dots, \lambda^1_{N-1}, \nu^1$ satisfying the constraints (4.17)–(4.19) for $i = 1$ and

$$\frac{\sum_{n=0}^{N-1} \lambda^1_n - \nu^1}{\sum_{n=0}^{m-1} \lambda^1_n} \leq \alpha^1 + \varepsilon$$

Case 1: Suppose $\lambda^1_{N-1} - \xi > 0$. In the following we construct $\lambda^2_0, \dots, \lambda^2_{N-1}, \nu^2$ satisfying the constraints (4.17)–(4.19) for $i = 2$ and

$$\frac{\sum_{n=0}^{N-1} \lambda^2_n - \nu^2}{\sum_{n=0}^{m-1} \lambda^2_n} \leq \alpha^1 + \varepsilon + \frac{B^2_{m+1}(\xi)}{\zeta}$$

Set $\lambda^2_n := \lambda^1_n$, $n = 0, \dots, N-2$, $\lambda^2_{N-1} := \lambda^1_{N-1} - \xi$. Set $\nu^2 := \max\{0, \nu^1 - B^2_{m+1}(\xi) - \xi\}$. Notice that by this construction, $\lambda^2_0, \dots, \lambda^2_{N-1}, \nu^2$ satisfies constraint (4.19). For $k = 0, \dots, N-2$ this implies

$$\sum_{n=k}^{N-1} \lambda^2_n \;=\; \sum_{n=k}^{N-1} \lambda^1_n - \xi \;\leq\; B^1_{N-k}(\lambda^1_k) - \xi \;\leq\; B^2_{N-k}(\lambda^1_k) + \xi - \xi \;=\; B^2_{N-k}(\lambda^2_k)$$

where the last equality holds since k ranges only from 0 to $N-2$. This implies (4.17) for $B_k = B^2_k$.

Next observe that for $j = 0, \dots, N-m-2$

$$\begin{aligned}
\nu^1 &\leq \sum_{n=0}^{j-1} \lambda^1_{n+m} + B^1_{N-j}(\lambda^1_{j+m}) \;\leq\; \sum_{n=0}^{j-1} \lambda^1_{n+m} + B^2_{N-j}(\lambda^1_{j+m}) + \xi \\
&= \sum_{n=0}^{j-1} \lambda^2_{n+m} + B^2_{N-j}(\lambda^2_{j+m}) + \xi
\end{aligned}$$

holds. Further observe that for $j = N-m-1$ we have

$$\begin{aligned}
\nu^1 &\leq \sum_{n=0}^{N-m-2} \lambda^1_{n+m} + B^1_{m+1}(\lambda^1_{N-1}) \;\leq\; \sum_{n=0}^{N-m-2} \lambda^1_{n+m} + B^2_{m+1}(\lambda^1_{N-1}) + \xi \\
&= \sum_{n=0}^{N-m-2} \lambda^2_{n+m} + B^2_{m+1}(\lambda^2_{N-1} + \xi) + \xi \\
&= \sum_{n=0}^{N-m-2} \lambda^2_{n+m} + B^2_{m+1}(\lambda^2_{N-1}) + B^2_{m+1}(\xi) + \xi
\end{aligned}$$

with the last equality due to linearity of B^2_{N-k}. In case $\nu^2 = 0$ we get

$$\nu^2 \leq \sum_{n=0}^{j-1} \lambda^2_{n+m} + B^2_{N-j}(\lambda^2_{j+m}),\ j = 0, \ldots, N-m-2$$

and in case $\nu^2 = \nu^1 - B^2_{m+1}(\xi) - \xi$ the inequalities

$$\begin{aligned} \nu^2 &\leq \sum_{n=0}^{N-m-2} \lambda^2_{n+m} + B^2_{m+1}(\lambda^2_{N-1}) \\ \nu^2 &\leq \nu^1 - \xi \leq \sum_{n=0}^{j-1} \lambda^2_{n+m} + B^2_{N-j}(\lambda^2_{j+m}), \quad j = 0, \ldots, N-m-2 \end{aligned}$$

hold. Thus, for $j = 0, \ldots, N-m-1$, we have $\nu^2 \leq \sum_{n=0}^{j-1} \lambda^2_{n+m} + B^2_{N-j}(\lambda^2_{j+m})$. This implies (4.18) for $B_k = B^2_k$.

Since $\sum_{n=0}^{m-1} \lambda^1_n = \sum_{n=0}^{m-1} \lambda^2_n \geq \zeta > 0$ and $\xi > 0$, the values $\lambda^2_m, \ldots, \lambda^2_{N-1}, \nu^2$ satisfy all constraints (4.17)–(4.19) for $i = 2$ and we obtain

$$\begin{aligned} \alpha^2 &\leq \frac{\sum\limits_{n=0}^{N-1} \lambda^2_n - \nu^2}{\sum\limits_{n=0}^{m-1} \lambda^2_n} = \frac{\sum\limits_{n=0}^{N-1} \lambda^1_n - \xi - \nu^2}{\sum\limits_{n=0}^{m-1} \lambda^2_n} \leq \frac{\sum\limits_{n=0}^{N-1} \lambda^1_n - \xi - \nu^1 + B^2_{m+1}(\xi) + \xi}{\sum\limits_{n=0}^{m-1} \lambda^1_n} \\ &\leq \alpha^1 + \varepsilon + \frac{B^2_{m+1}(\xi)}{\zeta}. \end{aligned}$$

Case 2: Now suppose $\lambda^1_{N-1} - \xi \leq 0$. Let $\mu := \sum_{n=0}^{N-m-2} \lambda^1_{n+m} + B^1_{m+1}(\lambda^1_{N-1})$. Then

$$\begin{aligned} \alpha^1 + \varepsilon &\geq \frac{\sum_{n=0}^{N-1} \lambda^1_n - \nu^1}{\sum_{n=0}^{m-1} \lambda^1_n} \geq \frac{\sum_{n=0}^{N-1} \lambda^1_n - \mu}{\sum_{n=0}^{m-1} \lambda^1_n} \\ &= \frac{\sum_{n=0}^{m-1} \lambda^1_n + \sum_{n=m}^{N-2} \lambda^1_n + \lambda^1_{N-1} - \mu}{\sum_{n=0}^{m-1} \lambda^1_n} \\ &= 1 + \frac{\mu - B^1_{m+1}(\lambda^1_{N-1}) + \lambda^1_{N-1} - \mu}{\sum_{n=0}^{m-1} \lambda^1_n} \\ &= 1 + \frac{B^1_{m+1}(\lambda^1_{N-1}) - \lambda^1_{N-1}}{-\sum_{n=0}^{m-1} \lambda^1_n} \geq 1 + \frac{B^1_{m+1}(\lambda^1_{N-1}) - \lambda^1_{N-1}}{-\zeta} \\ &\geq 1 - \frac{B^1_{m+1}(\lambda^1_{N-1})}{\zeta} \geq 1 - \frac{B^1_{m+1}(\xi)}{\zeta} \geq \alpha^2 - \frac{B^1_{m+1}(\xi)}{\zeta} \\ &\geq \alpha^2 - \frac{B^2_{m+1}(\xi) + \xi}{\zeta}. \end{aligned}$$

Hence, in both cases we obtain $\alpha^2 \leq \alpha^1 + \varepsilon + \frac{B^2_{m+1}(\xi)+\xi}{\zeta}$ which shows the assertion since $\varepsilon > 0$ was arbitrary.

(ii) We proceed by contradiction. Assume there are values $\lambda_0^1, \ldots, \lambda_{N-1}^1, \nu^1$ satisfying (4.17)–(4.19) for $i = 1$ and $\nu^1 > \sum_{n=0}^{N-1} \lambda_n^1 + B_{m+1}^2(\xi) + \xi$. Then the same construction as in (i) yields $\lambda_0^2, \ldots, \lambda_{N-1}^2, \nu^2$ satisfying (4.17)–(4.19) for $i = 2$ and

$$\alpha^2 \leq \frac{\sum_{n=0}^{N-1} \lambda_n^2 - \nu^2}{\sum_{n=0}^{m-1} \lambda_n^2} \leq \frac{\sum_{n=0}^{N-1} \lambda_n^1 - \nu^1 + B_{m+1}^2(\xi) + \xi}{\sum_{n=0}^{m-1} \lambda_n^1} < 0$$

which contradicts the assumption $\alpha^2 \geq 0$. □

The following theorem finally applies Lemma 4.2.2 to the problems $\mathcal{P}_\alpha^{\text{nmult}}$, $\mathcal{P}_\alpha^{\text{pmult}}$ and $\mathcal{P}_\alpha^{\text{upd}}$.

Theorem 4.2.3. *Consider problems $\mathcal{P}_\alpha^{nmult}$, $\mathcal{P}_\alpha^{pmult}$ and $\mathcal{P}_\alpha^{upd}$, let the assumptions of Theorem 2.1.8 hold and assume that the B_k, $k \in \mathbb{N}$ from $\mathcal{P}_\alpha^{nmult}$ are linear functions. Then*

$$\alpha^{pmult} \geq \alpha^{nmult} - \frac{B_{m+1}(\xi^{pmult}) + \xi^{pmult}}{\zeta}$$

$$\alpha^{upd} \geq \alpha^{nmult} - \frac{B_{m+1}(\xi^{upd}) + \xi^{upd}}{\zeta}$$

where ξ^{pmult} and ξ^{upd} are defined in (4.14) *and* (4.15), *respectively. Here, α^{nmult} can be replaced by the right hand side of Equation* (2.12).

Proof. We apply Lemma 4.2.2 setting $\alpha^2 := \alpha^{\text{nmult}}$, $B_k^2(r) := B_k(r)$, $\alpha^1 := \alpha^{\text{pmult}}$ and $B_k^1(r) := B_k(r) + \xi^{\text{pmult}}$. Then $\alpha^{\text{nmult}} \leq \alpha^{\text{pmult}} + \frac{B_{m+1}^2(\xi^{\text{pmult}})+\xi^{\text{pmult}}}{\zeta}$. Similarly, taking $\alpha^2 := \alpha^{\text{nmult}}$, $B_k^2(r) := B_k(r)$, $\alpha^1 := \alpha^{\text{upd}}$ and $B_k^1(r) := B_k(r) + \xi^{\text{upd}}$, we have that $\alpha^{\text{nmult}} \leq \alpha^{\text{upd}} + \frac{B_{m+1}^2(\xi^{\text{upd}})+\xi^{\text{upd}}}{\zeta}$. The fact that α^{nmult} can be replaced by the right hand side of (2.12) follows immediately from Theorem 2.1.10. □

The preceding theorem gives lower bounds for the values α^{pmult} and α^{upd} of the perturbed problems in terms of the performance index α^{nmult} of the nominal problem.

4.3 Asymptotic stability and performance

In this section we combine all previous results in order to prove the 'perturbed' counterpart to Theorem 2.1.8. To this end, we start with a preparatory lemma.

Lemma 4.3.1. *Let the assumptions of Corollary 4.1.5 hold. (a) Consider a perturbation sequence $d(\cdot)$ with $d(k) = 0$ for all $k \geq m$ and a trajectory $\tilde{x}_{\mu_{N,m}}(\cdot, x_0)$ of* (4.1) *which corresponds to a perturbation sequence $\tilde{d}(\cdot)$ with $\tilde{d}(k) = d(k)$ for $k = 0, \ldots, m-1$,* *(i) Let α^{pmult} be the solution of $\mathcal{P}_\alpha^{pmult}$ for $d(\cdot)$ and some $\zeta > 0$ and assume $\sum_{k=0}^{m-1} \ell(\tilde{x}_{\mu_{N,m}}(k, x_0), \mu_{N,m}(x_0, k)) \geq \zeta$. Then the inequality*

$$V_N(x_{\mu_{N,m}}(m, x_0)) \leq V_N(x_0) - \tilde{\alpha}^{pmult} \sum_{k=0}^{m-1} \ell(\tilde{x}_{\mu_{N,m}}(k, x_0), \mu_{N,m}(x_0, k)) \quad (4.20)$$

holds for

$$\tilde{\alpha}^{pmult} = \alpha^{pmult} - \frac{\sigma}{\zeta} \qquad \text{where } \sigma = \sum_{j=1}^{m-1} \omega_{J_{N-j}}(\|d(j)\|) \quad (4.21)$$

(ii) Assume that all values $\lambda_0, \ldots, \lambda_{N-1}, \nu^{pmult}$ satisfying the constraints from $\mathcal{P}_\alpha^{pmult}$ satisfy $\nu \leq \sum_{n=0}^{N-1} \lambda_n + B_{m+1}(\xi^{pmult}) + \xi^{pmult}$. Then the inequality

$$V_N(x_{\mu_{N,m}}(m, x_0)) \leq V_N(x_0) + B_{m+1}(\xi^{pmult}) + \xi^{pmult} + \sigma$$

holds for σ from (i).

(b) The analogous statements hold for the trajectories $\tilde{x}_{\hat{\mu}_{N,m}}(\cdot, x_0)$ of (4.2) with $\mathcal{P}_\alpha^{pmult}$, $\tilde{\alpha}^{pmult}$ etc. replaced by $\mathcal{P}_\alpha^{upd}$, $\tilde{\alpha}^{upd}$ etc. and $\sigma = \sum_{j=1}^{N-1} \omega_{V_{N-j}}(\|d(j)\|)$.

Proof. (a)(i) Consider the trajectory $x_{j,j,0}$ corresponding to the perturbation $d(\cdot)$ starting in $x_{0,0,0} = x_0$, and the corresponding values $\lambda_{j,j,0}$. Note that for $j = 0, \ldots, m$ the identities $\tilde{x}_{\mu_{N,m}}(j, x_0) = x_{j,j,0}$ and for $j = 0, \ldots, m-1$ the identities $\ell(\tilde{x}_{\mu_{N,m}}(j, x_0), \mu_{N,m}(x_0, j)) = \lambda_{j,j,0}$ hold.

By Corollary 4.1.5(i), the values $\lambda_n = \lambda_{n,n,0}$ and $\nu = V_N(x_{m,m,0})$ satisfy the constraints of $\mathcal{P}_\alpha^{\mathrm{pmult}}$. This implies

$$\nu^{\mathrm{pmult}} \leq \sum_{n=0}^{N-1} \lambda_{n,n,0} - \alpha^{\mathrm{pmult}} \sum_{n=0}^{m-1} \lambda_{n,n,0}$$

from which using (4.8) we obtain

$$\begin{aligned} V_N(x_{\mu_{N,m}}(m, x_0)) &\leq \sum_{n=0}^{N-1} \lambda_{n,n,0} - \alpha^{\mathrm{pmult}} \sum_{n=0}^{m-1} \lambda_{n,n,0} \\ &\leq \underbrace{\sum_{n=0}^{N-1} \lambda_{n,0,0}}_{=V_N(x)} + \underbrace{\sum_{n=1}^{N-1} \omega_{J_{N-n}}(\|d(n)\|)}_{=\sigma \leq \sigma \frac{\zeta}{\sum_{n=0}^{m-1} \lambda_{n,n,0}}} - \alpha^{\mathrm{pmult}} \sum_{n=0}^{m-1} \lambda_{n,n,0} \\ &\leq V_N(x) - \tilde{\alpha}^{\mathrm{pmult}} \sum_{n=0}^{m-1} \lambda_{n,n,0}, \end{aligned}$$

i.e., the assertion, since $d(m) = \ldots, d(N-1) = 0$.

(a)(ii) Similar to (i) we obtain

$$V_N(x_{\mu_{N,m}}(m, x_0)) \leq \sum_{n=0}^{N-1} \lambda_{n,n,0} + B_{m+1}(\xi^{\mathrm{pmult}}) + \xi^{\mathrm{pmult}}.$$

From this the assertion follows using the same estimates as in (i).

(b) Follows by analogous arguments using $x_{j,j,j}$, $\lambda_{j,j,j}$, Corollary 4.1.5(ii) and (4.9). $\square$

The following theorem – together with the subsequent remark – comprises the main result of this chapter. For its formulation we need an additional property of f.

Definition 4.3.2. We say that f is *uniformly bounded on each ball* $\overline{\mathcal{B}}_\Delta(x_*)$ if for any $\Delta > 0$ the value $\sup_{\|x\|_{x_*} \leq \Delta, u \in \mathbb{U}(x)} \|f(x,u)\|$ is finite.

Theorem 4.3.3. *(i) Let $N \geq 1$ and consider the MPC Algorithm 1.3.2 with stage cost $\ell : X \times U \to \mathbb{R}_0^+$ satisfying Assumption 2.1.2, yielding the m-step feedback law $\mu_{N,m}$. Assume that f is uniformly bounded on each ball $\overline{\mathcal{B}}_\Delta(x_*)$ and that J_K, $K = 1, \ldots, N$, f and ℓ are uniformly continuous on each ball $A = \overline{\mathcal{B}}_\eta(x_*)$ around x_* uniformly in u with their respective moduli of continuity $\omega^\eta_{J_K}$, ω^η_f and ω^η_ℓ. Let Assumption 2.1.4 hold with B_K being linear and that the optimization problem $\mathcal{P}^{nmult}_\alpha$ has an optimal value $\alpha^{nmult} \in (0,1]$, implying that the nominal closed-loop system is asymptotically stable.*

Then the perturbed m-step closed-loop system (4.1) with feedback law $\mu_{N,m}$ is semi-globally practically asymptotically stable on $\mathbb{X}$ with respect to d.

Moreover, for $\tilde{\alpha}^{pmult} > 0$ with $\tilde{\alpha}^{pmult}$ defined in Lemma 4.3.1, the performance estimate

$$J^{cl}_{k^*}(\tilde{x}_{\mu_{N,m}}(\cdot,x), \mu_{N,m}) \leq V_N(x)/\tilde{\alpha}^{pmult}.$$

holds for all $\tilde{x}_{\mu_{N,m}}(\cdot,x) \in S_{\overline{d}}(x)$.

(ii) The same statements hold for the MPC Algorithm 1.3.4, with the feedback law $\hat{\mu}_{N,m}$ and the corresponding closed-loop system (4.2) when we replace the moduli of continuity $\omega^\eta_{J_K}$ by $\omega^\eta_{V_K}$ and $\tilde{\alpha}^{pmult}$, α^{pmult} by $\tilde{\alpha}^{upd}$, α^{upd}, respectively, with $\tilde{\alpha}^{upd}$ defined in Lemma 4.3.1.

Proof. (i) To show that $\mu_{N,m}$ is semi-globally practically asymptotically stable on $\mathbb{X}$ with respect to d, via Lemma 2.2.5, for every $\delta > 0$ and every $\Delta > \delta$, we need to show existence of $\overline{d} > 0$ and sets Y and P with intermediate set $\widehat{P}$ satisfying $P \subseteq \widehat{P} \subseteq Y \subseteq \mathbb{X}$ and

$$\overline{\mathcal{B}}_\Delta(x_*) \cap \mathbb{X} \subseteq Y \quad \text{and} \quad \widehat{P} \subseteq \overline{\mathcal{B}}_\delta(x_*)$$

such that for each solution $\tilde{x}_\mu(\cdot,x_0) \in S_{\overline{d}}(x_0)$ the system is $\widehat{P}$-practically uniformly asymptotically stable on Y.

We can prove this through Theorem 2.2.8, i.e., by showing (a) the existence of $\alpha \in (0,1]$ such that the relaxed dynamic programming inequality (2.16) with $V = V_N$, $\mu = \mu_{N,m}$ holds for all $x_0 \in Y \backslash P$ and all $\tilde{x}_\mu(\cdot,x_0) \in S_{\overline{d}}(x_0)$, and (b) that (2.1) holds and there exists $\alpha_3, \alpha_4 \in \mathcal{K}_\infty$ with $\alpha_3(\|x\|_{x_*}) \leq V(x) \leq \alpha_4(\|x\|_{x_*})$

First, observe that by taking $\alpha_3 := \alpha_1$ and $\alpha_4 := B_N \circ \alpha_2$ with α_2 from (2.1) we obtain

$$\alpha_3(\|x_*\|) \leq \ell^*(x) \leq V_N(x) \leq B_N(\ell^*(x)) \leq B_N(\alpha_2(\|x\|_{x_*})) = \alpha_4(\|x\|_{x_*}) \tag{4.22}$$

showing (b).

To show (a), fix $\Delta > \delta > 0$ and an arbitrary $\kappa \in (0,1)$.

The next step consists of showing the existence of sets Y, P and $\widehat{P}$ and value $\overline{d} > 0$. We show this by the following construction.

Construction of Y: Consider first some arbitrary $\tilde{d} > 0$. Due to the uniform continuity of f on balls around x_*, there exists $\eta_1 > 0$ such that $f(x,u) + d \in B_{\eta_1}(x_*)$ for all $x \in \overline{\mathcal{B}}_\Delta(x_*)$ and all d with $\|d\| \le \tilde{d}$. Then inductively for $i = 2,\dots,N$, with η_{i-1} in place of Δ, there exists η_N such that $\tilde{x}_\mu(k,x_0) \in \overline{\mathcal{B}}_{\eta_N}(x_*)$ for all $k = 0,\dots,N$ for any solution $\tilde{x}_\mu(\cdot,x_0) \in S_{\tilde{d}}(x_0)$ and for any $x_0 \in \overline{\mathcal{B}}_\Delta(x_*)$.

We set $L := \alpha_4(\eta_N)$. Suppose $x \in \overline{\mathcal{B}}_{\eta_N}(x_*)$. Then $\|x\|_{x_*} \le \eta_N$ which implies $\alpha_4(\|x\|_{x_*}) \le L$. Since $V_N(x) \le \alpha_4(\|x\|_{x_*}) \le L$, $x \in Y := V_N^{-1}([0,L])$. Thus,

$$\overline{\mathcal{B}}_\Delta(x_*) \cap \mathbb{X} \;\subseteq\; \overline{\mathcal{B}}_{\eta_N}(x_*) \cap \mathbb{X} \;\subseteq\; Y.$$

Setting $\eta := \alpha_1^{-1}(L)$ implies $Y \subset \overline{\mathcal{B}}_\eta(x_*)$. We let $\omega_{J_K} = \omega^\eta_{J_K}$, $K = 0,\dots,N$, $\omega_f = \omega^\eta_f$ and $\omega_\ell = \omega^\eta_\ell$ denote the moduli of continuity of J_K, f and ℓ, respectively, on $A = \overline{\mathcal{B}}_\eta(x_*)$.

Construction of P and $\widehat{P}$: We set $p := \alpha \cdot \alpha_3 \circ \alpha_4^{-1} \circ \alpha_3(\delta)$ with $\alpha = \kappa\alpha^{\text{nmult}}$ and define $P := V_N^{-1}([0,p])$. Suppose $x \in P$. Since $\alpha_3(\|x\|_{x_*}) \le V_N(x) \le p$, $\|x\|_{x_*} \le \alpha_3^{-1}(p)$, i.e., $x \in \overline{\mathcal{B}}_{\alpha_3^{-1}(p)}(x_*)$. Furthermore

$$\alpha\alpha_3(\|x\|_{x_*}) \le \alpha_3(\|x\|_{x_*}) \le V_N(x) \le \alpha_4(\|x\|_{x_*}) \le \alpha_4(\alpha_3^{-1}(p))$$

giving $\|x\|_{x_*} \le \frac{1}{\alpha}\alpha_3^{-1}(\alpha_4(\alpha_3^{-1}(p)))$, i.e., $x \in \overline{\mathcal{B}}_\delta(x_*)$. All this gives

$$P \;\subseteq\; \overline{\mathcal{B}}_{\alpha_3^{-1}(p)}(x_*) \;\subseteq\; \overline{\mathcal{B}}_\delta(x_*)$$

for which we define $\widehat{P} := \overline{\mathcal{B}}_\delta(x_*)$. For later use, we also define $q := p/2$, $Q := V_N^{-1}([0,q]) \subset P$ and $\zeta := \alpha_1(\alpha_4^{-1}(q))$. Observe that if $x \notin Q$, then $\alpha_4(\|x\|_{x_*}) \ge V_N(x) \ge q$ which yields $\ell^*(x) \ge \alpha_1(\|x\|_{x_*}) \ge \alpha_1(\alpha_4^{-1}(q))$. This implies the choice of ζ ensures $\ell^*(x) \ge \zeta$.

Choice of $\overline{d}$: We choose $\overline{d} \in (0, \min\{\tilde{d}, q\}]$ maximal such that the two conditions

$$B_{m+1}(\xi^{\text{pmult}}) + \xi^{\text{pmult}} + \sigma \le q \qquad \text{and} \qquad \tilde{\alpha}^{\text{pmult}} \ge \kappa\alpha^{\text{nmult}}$$

hold for ξ^{pmult} from Corollary 4.1.5(i), and σ and $\tilde{\alpha}^{\text{pmult}}$ from Lemma 4.3.1(a)(i) with ζ from above. Such $\overline{d} > 0$ exists due to Lemma 4.3.1 and Theorem 4.2.3: Due to the uniform continuity assumption on the J_K, f and ℓ and the linearity of B_K, all terms in the definition of ξ^{pmult} in Corollary 4.1.5(i) vanish as $\overline{d} \to 0$. We note that $\overline{d}$ depends on δ via q and ζ (which depends on δ via the construction of P) and on Δ via the moduli of continuity ω_{J_K}, ω_f and ω_ℓ (which depend on Δ via the construction of Y). By Lemma 4.3.1, this choice of $\overline{d}$ ensures (4.20) and thus (2.16) with $V = V_N$, $\mu = \mu_{N,m}$ and $\alpha = \tilde{\alpha}^{\text{pmult}} = \kappa\alpha^{\text{nmult}} > 0$ for all $x_0 \in Y$ with $\ell^*(x_0) \ge \zeta$. By the choice of ζ, this includes all $x_0 \in Y \setminus Q$.

Now what remains is to verify that Y and P are m-step forward invariant with respect to $\overline{d}$ and that $\widehat{P}$ is an intermediate set of P.

***m*-step forward invariance of Y:** It suffices to show the implication $x_0 \in Y \;\Rightarrow\; \tilde{x}_{\mu_{N,m}}(m,x_0) \in Y$ for all $\tilde{x}_{\mu_{N,m}}(\cdot,x_0) \in S_{\overline{d}}(x_0)$ since $\tilde{x}_{\mu_{N,m}}(rm,x_0) \in Y$

for $r \geq 2$ then follows by induction. For $x_0 \in Y \setminus Q$, we know that (4.20) applies, yielding $V_N(\tilde{x}_{\mu_{N,m}}(m, x_0)) \leq V_N(x_0)$ which implies $\tilde{x}_{\mu_{N,m}}(m, x_0) \in Y$. For $x_0 \in Q$, we know that $\|x_0\|_{x_*} \leq \delta < \Delta$. By construction of Y, all perturbed trajectories starting in $\overline{\mathcal{B}}_\Delta(x_*)$ remain in Y for at least N steps, which implies $\tilde{x}_{\mu_{N,m}}(m, x_0) \in Y$ since $m < N$.

m-step forward invariance of P: It suffices, once again, to show the implication $x_0 \in P \;\Rightarrow\; \tilde{x}_{\mu_{N,m}}(m, x_0) \in P$ for all $\tilde{x}_{\mu_{N,m}}(\cdot, x_0) \in S_{\overline{d}}(x_0)$. We thus consider arbitrary $x_0 \in P$ and $\tilde{x}_{\mu_{N,m}}(\cdot, x_0) \in S_{\overline{d}}(x_0)$ and distinguish two cases:

Case 1: $x_0 \notin Q$. Then (4.20) applies, yielding $V_N(\tilde{x}_{\mu_{N,m}}(m, x_0)) \leq V_N(x_0)$ which implies $\tilde{x}_{\mu_{N,m}}(m, x_0) \in P$.

Case 2: $x_0 \in Q$. Since $\alpha^{nmult} > 0$, Lemma 4.2.2(ii) applies and ensures that the assumptions of Lemma 4.3.1(a)(ii) are satisfied. Then the choice of Q, q and $\overline{d}$ yields

$$V_N(\tilde{x}_{\mu_{N,m}}(m, x_0)) \leq V_N(x_0) + B_{m+1}(\xi^{\mathrm{pmult}}) + \xi^{\mathrm{pmult}} + \sigma \leq q + q = p$$

which again implies $\tilde{x}_{\mu_{N,m}}(m, x_0) \in P$.

$\widehat{P}$ is an intermediate set: It remains to show that $\tilde{x}_{\mu_{N,m}}(k, x_0) \in \widehat{P} = \overline{\mathcal{B}}_\delta(x_*)$ for all $k \geq 0$ and $x_0 \in P$. To this end, we use the inequality

$$V_N(\tilde{x}_{\mu_{N,m}}(k, x_0)) \leq \alpha_4 \circ \alpha_1^{-1}(V_N(\tilde{x}_{\mu_{N,m}}(\lfloor k \rfloor_m, x_0))/\alpha)$$

derived in the proof of Theorem 2.2.8(ii). Since P is m-step forward invariant, we know $\tilde{x}_\mu(\lfloor k \rfloor_m, x_0) \in P$ and thus

$$V_N(\tilde{x}_{\mu_{N,m}}(k, x_0)) \leq \alpha_4 \circ \alpha_1^{-1}(p/\alpha)$$

which by (4.22) and choice of p implies

$$\|\tilde{x}_{\mu_{N,m}}(k, x_0)\|_{x_*} \leq \alpha_3^{-1} \circ \alpha_4 \circ \alpha_1^{-1}(p/\alpha) = \delta$$

and thus shows $\tilde{x}_{\mu_{N,m}}(k, x_0) \in \widehat{P}$.

(ii) The proof is completely identical to (i), observing that throughout the proof of (i), we have only used properties of Algorithm 1.3.2 and system (4.1) which have also been proven for Algorithm 1.3.4 and system (4.2). □

Remark 4.3.4. (a) The decisive difference between the cases (i) and (ii) in Theorem 4.3.3 which determine both the bound for $\overline{d}$ and the suboptimality index α lies in the error terms. For Algorithm 1.3.2 yielding index $\tilde{\alpha}^{\mathrm{pmult}}$, the error terms depend on ω_{J_K} and for Algorithm 1.3.4 yielding index $\tilde{\alpha}^{\mathrm{upd}}$ the error terms depend on ω_{V_K} .

(b) The bound $\overline{d}$ depending on Δ and δ in Definition 2.2.4 can be chosen to satisfy the condition $\tilde{\alpha}^{\mathrm{pmult}} > \kappa\alpha^{\mathrm{nmult}}$ for arbitrary $\kappa \in (0, 1)$, with $\tilde{\alpha}^{\mathrm{pmult}}$ from Lemma 4.3.1(a)(i). Here, the moduli of continuity ω_{J_N} involved in the estimates for $\tilde{\alpha}^{\mathrm{pmult}}$ and α^{pmult} are chosen as $\omega_{J_N} = \omega^\eta_{J_N}$ with η depending on Δ. The value ζ in these estimates depends on δ. An analogous statement hold for $\tilde{\alpha}^{\mathrm{upd}}$.

(c) Recall that a larger value of the suboptimality index α indicates better performance of the scheme. Theorem 4.2.3 limits the performance loss of α^{pmult}

and α^{upd} to the values $\frac{B_{m+1}(\xi^{\text{pmult}})+\xi^{\text{pmult}}}{\zeta}$ and $\frac{B_{m+1}(\xi^{\text{upd}})+\xi^{\text{upd}}}{\zeta}$, respectively with

$$\frac{B_{m+1}(\xi^{\text{pmult}})+\xi^{\text{pmult}}}{\zeta} \geq \frac{B_{m+1}(\xi^{\text{upd}})+\xi^{\text{upd}}}{\zeta}$$

since $\xi^{\text{pmult}} \geq \xi^{\text{upd}}$ (recall their definitions in (4.14) and (4.15)) and $\omega_{J_k} \geq \omega_{V_k}$. This means that although we can not conclude that $\alpha^{\text{upd}} > \alpha^{\text{pmult}}$, the theorem nevertheless guarantees that the estimated performance of the updated scheme can not be worse than that of the non-updated scheme. In Section 4.4, we give an example in which updated m-step indeed performs better than the m-step scheme.

(d) Now the definition of the performance indices $\tilde{\alpha}^{\text{pmult}}$ and $\tilde{\alpha}^{\text{upd}}$ in Theorem 4.3.3, where $\omega_{J_k} \geq \omega_{V_k}$, with the difference being significant, e.g., in case of open loop unstable and controllable systems (see Section 3.4, in particular, Remark 3.4.3 and examples in Section 3.5), explains and quantifies the better robustness properties of the updated MPC scheme.

4.4 Numerical example: inverted pendulum

In order to illustrate our results, we consider a nonlinear inverted pendulum model consisting of a cart mounted on a track where it can move and attached to it is a rigid pendulum that is able to rotate freely. We use the different MPC controllers discussed in our study to swing up the pendulum to the unstable upright or inverted position and stabilize it there. We consider the model used in [37]

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= -\frac{g}{\ell}\sin(x_1(t)) - \frac{k_L}{l}\arctan(1000x_2(t))x_2^2(t) - \frac{u(t)}{l}\cos(x_1(t)) \\
&\quad -k_R\left(\frac{4ax_2(t)}{1+4(ax_2(t))^2} + \frac{2\arctan(bx_2(t))}{\pi}\right) \\
\dot{x}_3(t) &= x_4(t) \\
\dot{x}_4(t) &= u(t)
\end{aligned}$$

where $x_i, i = 1, \ldots, 4$ represents pendulum angular displacement, angular velocity, cart position and cart velocity, respectively, with gravitational constant $g = 9.81$, pendulum length $l = 1.25$ and friction parameters $k_L = 0.007$ and $k_R = 0.197$. In order to convert the continuous time system to a discrete time model (1.1) we sample it with zero order hold and sampling period $T = 0.2$. To stabilize the upright position $x_* = ((2k+1)\pi, 0, 0, 0),\ k \in \mathbb{N}$, we consider the stage cost also used in [37]

$$\begin{aligned}
\ell(x(i), u(i)) &= \int_{t_i}^{t_{i+1}} 10^{-4}u(t)^2 + \big(3.51\sin(x_1(t)-\pi)^2 \\
&\quad + 4.82\sin(x_1(t)-\pi)x_2(t) + 2.31x_2(t)^2 \\
&\quad + 0.1\left((1-\cos(x_1(t)-\pi))\cdot(1+\cos(x_2(t))^2)\right)^2 \\
&\quad + 0.01x_3(t)^2 + 0.1x_4(t)^2\big)^2\,dt
\end{aligned}$$

where $t_i = iT$, leading to a cost functional of $J_N(x_0, u) = \sum_{i=0}^{N-1} \ell(x(i), u(i))$. We aim to compare simulations resulting from the multistep and updated multistep feedback controllers both on nominal and perturbed setting. We set the length of the optimization horizon to $N = 15$, set the initial value $x_0 = (-\pi - 0.1, 0, -0.1, 0)$ and for the perturbed system (2.13) we use a fixed randomly generated perturbation sequence of the form $d(k) = [0, 0, d_3(k), 0]^\top, k \in \mathbb{N}$, (i.e., perturbations occur on the cart position x_3 and are identical for each simulation) with values in the interval $[-\overline{d}_3, 0]$ for $\overline{d}_3 = 0.05$. Note that for demonstration purposes, i.e., to see clear trends, we present an example where perturbations have uniform signs (similar results are also obtained for $[0, \overline{d}_3]$), in contrast to arbitrarily signed perturbations which could also be chosen otherwise.

Figure 4.1 illustrates the trajectories for $m = 1$, where the 1-step MPC scheme (shown in blue) renders the nominal system asymptotically stable at $(-\pi, 0, 0, 0)$ while, as expected, the 1-step perturbed solution (cyan) is only practically asymptotically stable, i.e., only converges to a neighborhood of x_*. We remark that for $m = 1$, the trajectories generated by (4.1) and (4.2) coincide, hence only the former is shown in the figure. For $m = 7$, trajectories resulting from the nominal 7-step (blue), perturbed 7-step (red), and perturbed updated 7-step (green) are plotted in Figure 4.2. The larger m is chosen, the longer the multistep controller does not counteract the effect of the perturbation preventing the trajectory to arrive closer to the equilibrium which is exactly what we see in the plots (shown in red). Improvement is manifested by applying the updates to the multistep scheme allowing the trajectory to move towards the equilibrium against the perturbations (shown in green). Finally, the figure also illustrates how all the schemes mentioned compare to the 1-step scheme – the most robust scheme (shown in cyan).

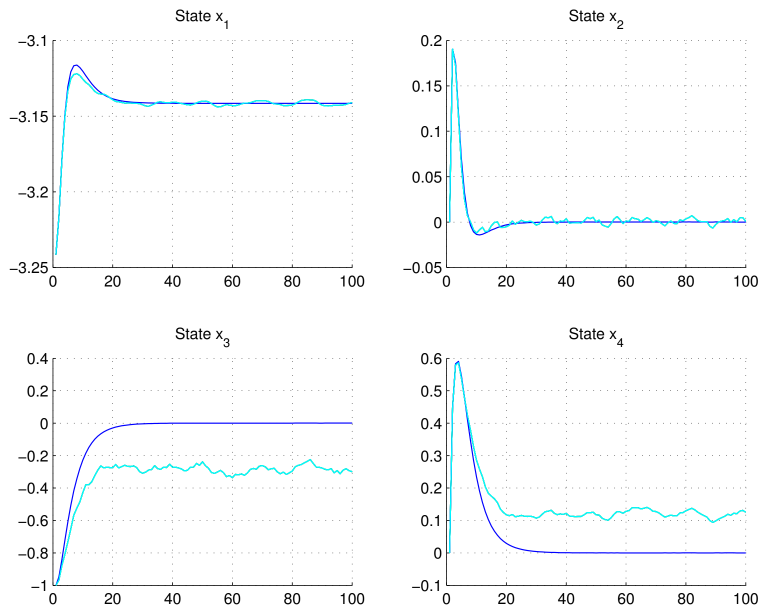

Figure 4.1: State trajectories driven by the 1-step MPC scheme for nominal (blue) and perturbed system (cyan)

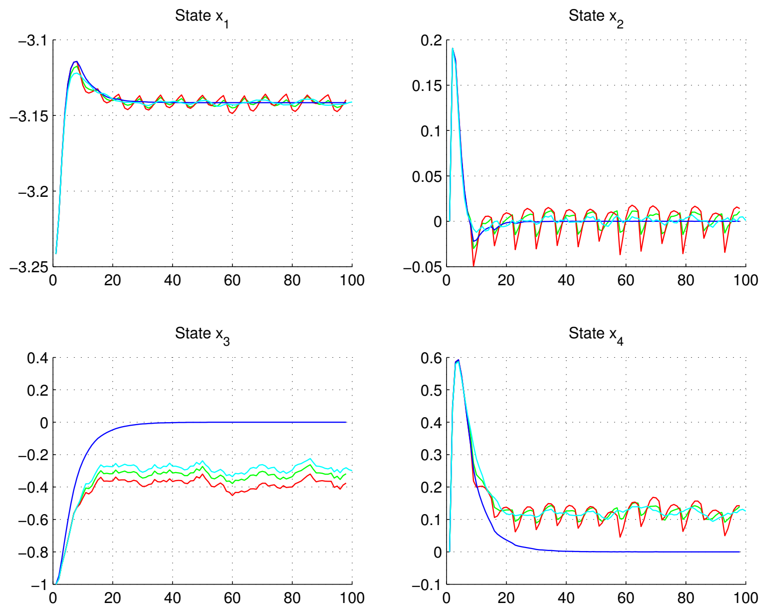

Figure 4.2: State trajectories driven by the 7-step MPC scheme for nominal system (blue), the 1-step (cyan), 7-step (red) and updated 7-step (green) MPC schemes for the perturbed system

Table 4.4.1 shows the comparison of time requirements in CPU time among the multistep and the updated multistep schemes for increasing multisteps m. To allow comparison, time instants 0 to 100 are considered for which for each scheme, $\texttt{floor}(100/m)$ optimizations with full horizon N are performed and the times needed are recorded. As expected, since neither a control has to be computed nor an optimization has to be performed for the multistep scheme, the larger m is chosen, the larger the savings in time becomes. For each m, due to the sequence of optimization with shrinking horizon that has to be performed, the corresponding updated scheme requires more time which one can easily notice in the table. Although optimization for each time step is still required for the updated multistep scheme, savings in time is nevertheless achieved in contrast to the 1-step MPC – the most expensive scheme – which performs optimization with full horizon N at each time instant. We note that no warm-start was used in the simulations. Otherwise, time requirements would have been lowered for all schemes but the trend is expected to remain the same. Also, the slight difference between the updated and the non-updated scheme for $m = 1$ (same schemes) appears on the table because the simulation was run twice.

Finally, Table 4.4.2 presents performance indices α of the schemes which are computed from the generated trajectories using the approach presented in [35]. To estimate the α values α^{nmult}, $\tilde{\alpha}^{\mathrm{pmult}}$ and $\tilde{\alpha}^{\mathrm{upd}}$, we use (2.11), (4.20) and its counterpart for the perturbed updated m-step scheme, respectively. The values in these formulas are available at runtime giving a computationally feasible and inexpensive *a posteriori* α estimator. We vary m and list the values of α for the first three iterations of each scheme. In our simulation, the values of α for the nominal multistep scheme indicates that the feedback is 'close' to being infinite horizon optimal having values $\alpha > 0.9$. Furthermore, along increasing m, the α

Table 4.4.1: Comparison of time requirements in CPU time

m	multistep	updated
1	11.0447	11.0967
2	5.6484	10.4687
3	3.6762	10.3646
4	2.5522	10.1046
5	2.1921	9.3766
6	1.8241	8.6125
7	1.5801	7.7765
8	1.2321	7.7845
9	1.0881	7.2405
10	1.0641	6.5404
11	0.9521	6.1124
12	0.8601	5.7884
13	0.8681	5.2243

values increase, peak and then deteriorate exemplifying the parabolic profile of the α's of the multistep MPC scheme reported in [37]. For the perturbed system with $\overline{d}_3 = 0.05$, for the multistep scheme, α values are observably lower and even worsen on the second and third iteration where negative values are also seen. These negative values indicate that the region $\widehat{P}$ of practical asymptotic stability has been reached, see [35, Section 4]. Most importantly, Table 4.4.2 shows a noticeable improvement to the values of α for the updated multistep brought about by the re-optimization that counteracts the effect of the perturbation as seen in the last three columns of the table. Weighing in all benefits after examining the time requirements and suboptimality estimates, by updating the multistep feedback for the perturbed system, we clearly gain time savings compared to the classical MPC scheme, and improve robustness in comparison with the multistep feedback scheme.

Table 4.4.2: Suboptimality index α of the schemes for various m and iterations

	nominal multistep			perturbed multistep			updated multistep		
m	0	2m	3m	0	2m	3m	0	2m	3m
1	0.9908	0.9917	0.9935	0.8667	0.8699	0.6032	0.8667	0.8699	0.6032
2	0.9911	0.9937	0.9950	0.8678	0.6322	0.8479	0.8681	0.6383	0.8538
3	0.9915	0.9944	0.9948	0.7936	0.7713	0.5857	0.7955	0.7810	0.6203
4	0.9917	0.9942	0.9937	0.7672	0.6870	0.5282	0.7729	0.7139	0.5647
5	0.9916	0.9933	0.9916	0.7632	0.6898	0.4171	0.7734	0.7307	0.4882
6	0.9913	0.9916	0.9880	0.7724	0.3915	0.3810	0.7868	0.4974	0.4037
7	0.9908	0.9887	0.9829	0.7404	0.4850	-0.0954	0.7629	0.5695	-0.0251
8	0.9902	0.9843	0.9755	0.7103	0.4233	-0.0370	0.7414	0.4981	0.0228
9	0.9895	0.9778	0.9662	0.7066	0.1941	-0.0328	0.7423	0.2845	-0.0129
10	0.9888	0.9698	0.9561	0.6988	0.0840	-0.2314	0.7379	0.1718	-0.2125
11	0.9883	0.9622	0.9461	0.6477	0.1414	-0.0467	0.6953	0.1394	0.0009
12	0.9880	0.9576	0.9400	0.6183	0.1227	-0.1213	0.6688	0.0776	-0.0356
13	0.9879	0.9584	0.9372	0.6133	-0.0139	-0.1130	0.6609	-0.0474	-0.0468

5 NLP and sensitivity analysis

Although the updated m-step MPC already results in a noticeable reduction in terms of computational cost compared to the standard MPC, we see that optimization still needs to be carried out at each time step. We design in this thesis another MPC variant that results in another significant reduction in the computational expense where the updates (i.e., re-optimizations) in Algorithm 1.3.4 are replaced by approximative updates obtained through sensitivity analysis (as will be detailed in the next chapter). To this end, we first present some results of the study on sensitivity analysis by Fiacco in [23, 24] and Robinson [58] which will serve as the foundation of the described MPC variant.

As a prerequisite, we revisit in this chapter essential concepts from nonlinear programming found in classic optimization textbooks (e.g., [49, 11, 64]). Basic definitions and theorems are given in Section 5.1. Sections 5.2 and 5.3 focus on optimality conditions and solving unconstrained optimization problems using gradient-based methods. Section 5.4 deals with constrained optimization. In this section, we derive an algorithm to solve the constrained optimization problem and investigate optimality conditions and additional crucial properties for sensitivity analysis. Section 5.5 presents the sensitivity theorem on which the next chapter will essentially be based. The theorems, along with their proofs, and formulations taken from the classic literature are written in this chapter in nomenclature that allows involved quantities to be easily incorporated to the MPC setting discussion in the next chapter.

5.1 Basic definitions and theorems

Suppose $z \in \mathbb{R}$. Consider the scalar function $\varphi : \mathbb{R} \to \mathbb{R}$. Assuming the limit given below exists, we define the **(first) derivative** as

$$\frac{d\varphi}{dz}(z) := \lim_{h \to 0} \frac{\varphi(z+h) - \varphi(z)}{h}$$

Suppose now $z = (z_1, \dots, z_n)^\top \in \mathbb{R}^n$ and define the unit vector

$$e_i = (0, \dots, 1, \dots, 0)^\top \in \mathbb{R}^n,$$

i.e., a vector with 0 entries except for a 1 on the ith position.

For a scalar function $\varphi : \mathbb{R}^n \to \mathbb{R}$, assuming the limit given below exists, we define the **partial derivative** as

$$\frac{\partial \varphi}{\partial z_i}(z) := \lim_{h \to 0} \frac{\varphi(z + he_i) - \varphi(z)}{h}$$

The operator ∇ is defined as

$$\nabla := \left(\frac{\partial}{\partial z_1}, \ldots, \frac{\partial}{\partial z_n} \right)^\top$$

giving a column vector of partial derivatives. Unless otherwise specified, ∇ is understood to be the derivative with respect to z, i.e., ∇_z.

The **gradient of a scalar function** $\varphi : \mathbb{R}^n \to \mathbb{R}$ is given by

$$\nabla \varphi := \left(\frac{\partial \varphi}{\partial z_1}, \ldots, \frac{\partial \varphi}{\partial z_n} \right)^\top$$

The **gradient of a vector function** $\varphi : \mathbb{R}^n \to \mathbb{R}^m$ with

$$\varphi(z) = (\varphi_1(z), \ldots, \varphi_m(z))^\top$$

is given by

$$\nabla \varphi := (\nabla \varphi_1 \ \nabla \varphi_2 \ \ldots \ \nabla \varphi_m) = \begin{pmatrix} \frac{\partial \varphi_1}{\partial z_1} & \cdots & \frac{\partial \varphi_m}{\partial z_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \varphi_1}{\partial z_n} & \cdots & \frac{\partial \varphi_m}{\partial z_n} \end{pmatrix}$$

The **Jacobian of a vector function** $\varphi : \mathbb{R}^n \to \mathbb{R}^m$ is given by

$$\nabla \varphi^\top := \begin{pmatrix} (\nabla \varphi_1)^\top \\ \vdots \\ (\nabla \varphi_m)^\top \end{pmatrix}$$

In the case of a twice continuously differentiable scalar function $\varphi : \mathbb{R}^n \to \mathbb{R}$, the matrix

$$\nabla^2 \varphi := \nabla \nabla \varphi = \begin{pmatrix} \frac{\partial^2 \varphi}{\partial z_1^2} & \frac{\partial^2 \varphi}{\partial z_1 \partial z_2} & \cdots & \frac{\partial^2 \varphi}{\partial z_1 \partial z_n} \\ \frac{\partial^2 \varphi}{\partial z_2 \partial z_1} & \frac{\partial^2 \varphi}{\partial z_2^2} & \cdots & \frac{\partial^2 \varphi}{\partial z_2 \partial z_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \varphi}{\partial z_n \partial z_1} & \frac{\partial^2 \varphi}{\partial z_n \partial z_2} & \cdots & \frac{\partial^2 \varphi}{\partial z_n^2} \end{pmatrix}$$

is called the **Hessian matrix**.

Let $\varepsilon > 0$. An **ε-neighborhood** $\mathcal{N}_\varepsilon(z)$ of $z \in \mathbb{R}^n$ is defined as

$$\mathcal{N}_\varepsilon(z) = \{\tilde{z} \in \mathbb{R}^n \mid \|z - \tilde{z}\| < \varepsilon\}$$

We may also use the notation $\mathcal{N}(z)$ whenever ε is not specified.

A function $\varphi : \mathbb{R}^n \to \mathbb{R}$ is said to be **convex** if

$$\alpha\varphi(x) + (1-\alpha)\varphi(y) \;\geq\; \varphi(\alpha x + (1-\alpha)y)$$

holds for all $\alpha \in (0,1)$ and all points $x, y \in \mathbb{R}^n$. If the strict inequality '>' is imposed instead of '$\geq$', then we have **strict convexity**.

A region Y is **convex** if for all points $x, y \in Y$,

$$\alpha x + (1-\alpha)y \in Y$$

holds for all $\alpha \in (0,1)$.

In addition, we need the following theorems giving important tools for the subsequent sections.

Theorem 5.1.1 (Taylor's theorem)**.** *Suppose that $\varphi(z)$ is continuously differentiable, then we have for all $z, p \in \mathbb{R}^n$,*

$$\varphi(z+p) = \varphi(z) + \nabla\varphi(z+tp)^\top p \qquad \text{for some } t \in (0,1)$$

Moreover, if $f(z)$ is twice continuously differentiable, then we have for all $z, p \in \mathbb{R}^n$,

$$\varphi(z+p) = \varphi(z) + \nabla\varphi(z)^\top p + \frac{1}{2}p^\top\nabla^2\varphi(z+tp)p \qquad \text{for some } t \in (0,1)$$

Theorem 5.1.2 (Farkas' lemma)**.** *Given $C \in \mathbb{R}^{m\times n}, D \in \mathbb{R}^{m\times k}$ and $b \in \mathbb{R}^m$. Exactly one of the following statements is true:*

(i) There exists $x \in \mathbb{R}^n, y \in \mathbb{R}^k$ such that $Cx + Dy = b$ with $x \geq 0$.

(ii) There exists $v \in \mathbb{R}^m$ such that $C^\top v \geq 0,\; D^\top v = 0$ and $b^\top v < 0$.

5.2 Unconstrained optimization

Consider an objective function $f : \mathbb{R}^n \to \mathbb{R}$ and the unconstrained optimization problem

$$\min_{z\in\mathbb{R}^n} f(z) \tag{5.1}$$

In this section, we investigate necessary and sufficient conditions for the solution of (5.1) which we first formally define in the following.

Definition 5.2.1.
A point $z^* \in \mathbb{R}^n$ is a **global minimizer** of f if

$$f(z^*) \leq f(z) \quad \text{for all } z \in \mathbb{R}^n$$

A point $z^* \in \mathbb{R}^n$ is a **local minimizer** of f if there exists a neighborhood $\mathcal{N}(z^*)$ of z^* such that

$$f(z^*) \leq f(z) \quad \text{for all } z \in \mathcal{N}(z^*)$$

A point $z^* \in \mathbb{R}^n$ is a **strict local minimizer** of f if there exists a neighborhood $\mathcal{N}(z^*)$ of z^* such that

$$f(z^*) < f(z) \quad \text{for all } z \in \mathcal{N}(z^*),\; z \neq z^*$$

Theorem 5.2.2. *Suppose f is convex and z^* is local minimizer of f. Then z^* is a global minimizer.*

Proof. Suppose z^* is not the global minimizer. Then there exists y such that $f(z^*) > f(y)$. Since z^* is a local minimizer, $f(z^*) \leq f(x)$ for all $x \in \mathcal{N}_\varepsilon(z^*)$ for some positive ε. Since f is convex,

$$f(\alpha y + (1-\alpha)z^*) \;\leq\; \alpha f(y) + (1-\alpha)f(z^*)$$

for all $\alpha \in (0,1)$. Taking α such that $x = \alpha y + (1-\alpha)z^* \in \mathcal{N}_\varepsilon(z^*)$, then

$$f(x) \;\leq\; f(z^*) + \alpha \underbrace{(f(y) - f(z^*))}_{<0} \;<\; f(z^*)$$

giving a contradiction. □

We provide in the following two theorems which are standard results on unconstrained optimization.

Theorem 5.2.3 (First and second-order necessary conditions for local optimality)**.** *Suppose f is twice continuously differentiable and z^* is a local minimizer of f, then $\nabla f(z^*) = 0$ and $\nabla^2 f(z^*)$ is positive semidefinite.*

Proof. See, e.g., proofs of [11, Theorem 2.17] or [64, Theorem 1.4.4-5] which make use of Taylor's theorem. □

Theorem 5.2.4 (Sufficient conditions for local optimality)**.** *Suppose $f(z)$ is twice continuously differentiable and there exists $z^* \in \mathbb{R}^n$ where $\nabla f(z^*) = 0$ and $\nabla^2 f(z^*)$ is positive definite, then z^* is a strict local minimizer.*

Proof. See, e.g., proofs of [11, Theorem 2.18] or [64, Theorem 1.4.6]. □

5.3 Optimization methods requiring derivatives

To solve (5.1), for twice continuously differentiable f, we consider iterative algorithms that generate a sequence of iterates z^k that converges to z^*. Consider f and apply Taylor's theorem at $z + p$. Then we have

$$f(z+p) \approx f(z) + \nabla f(z)^\top p + \frac{1}{2} p^\top \nabla^2 f(z) p$$

giving a quadratic Taylor expansion of f at z. Computing the gradient, we have

$$\nabla f(z+p) \approx \nabla f(z) + \nabla^2 f(z) p$$

At point z, to determine the vector p that locates the stationary point, i.e., p that satisfies $\nabla f(z+p) = 0$, we have

$$\begin{aligned} 0 &= \nabla f(z) + \nabla^2 f(z) p \qquad (5.2) \\ \Leftrightarrow \quad p &= -(\nabla^2 f(z))^{-1} \nabla f(z) \end{aligned}$$

if $\nabla^2 f(z)$ is nonsingular. This gives a general optimization method in the form of an update rule

$$z^{k+1} = z^k + p^k \quad \text{where } p^k = -(\nabla^2 f(z^k))^{-1} \nabla f(z^k) \tag{5.3}$$

Here, p^k is called the **Newton direction** obtained by solving the linear system (5.2) which involves the matrix $\nabla^2 f(z)$ called the **Hessian matrix**.

The update rule (5.3) can be modified to include a step length γ to enforce a sufficient reduction of the objective function through

$$z^{k+1} = z^k + \gamma^k p^k \tag{5.4}$$

Techniques such as line search method (see e.g., [49, 15, 11]) are employed to compute the step length γ^k indicating how far z^k should move along the direction p^k.

The algorithm is formally given in the following.

Algorithm 5.3.1. (Newton's method with line search)
Choose a starting point z^0.

(1) At z^k, evaluate $\nabla f(z^k)$ and $\nabla^2 f(z^k)$.

(2) Solve the linear system $\nabla^2 f(z^k) p^k = -\nabla f(z^k)$. If $p^k = 0$, stop.

(3) Determine a step length γ^k. Set $z^{k+1} = z^k + \gamma^k p^k$ and $k = k + 1$. Go to (1).

The described Newton's method for optimization is essentially a root-finding approach for the system $\nabla f(z) = 0$ and hence does not distinguish among local minimizers, local maximizers and saddle points.

From Algorithm 5.3.1 (2), one realizes that certain challenges arising from this method come from the exact computation and storage of the Hessian matrix that has to be done at each iteration. Calculating derivatives can be done, e.g., using finite differences or automatic differentiation [49, Chapter 8]. In addition, the positive definiteness of the Hessian matrix is necessary for the method to work. A major advantage, however, in using the method is the quadratic convergence of the scheme (see optimization textbooks, e.g., [49, 15, 11] for the convergence proof). In addition, if f is actually a quadratic function, then the second Taylor approximation is exact yielding $z + p$ to be the global minimizer.

Newton-type methods refer to schemes which can be written as

$$z^{k+1} = z^k - B_k^{-1} \nabla f(z^k) \tag{5.5}$$

Algorithm 5.3.1 uses $B_k = \nabla^2 f(z^k)$, i.e., the exact Hessian matrix which is also the reason the scheme is also called the exact Newton's method. Other well-known Newton-type methods are the steepest descent/ascent method ($B_k = \gamma^k I$), Gauss-Newton and Levenberg-Marquardt method, Quasi-Newton methods and inexact Newton methods, to name a few, each exhibiting different properties such as order of convergence, under which setting they are most suitable, etc. See discussions in e.g., [49, 15, 11]. These Newton-type methods serve as adaptation techniques to Algorithm 5.3.1 allowing approximation of the Hessian matrix at a lower cost giving substantial advantages in the case of large-scale systems.

5.4 Constrained optimization and SQP

We now examine the corresponding optimality conditions for the constrained problem

$$\begin{array}{rl} \min & f(z) \\ \text{s.t.} & g(z) \le 0 \\ & h(z) = 0 \end{array} \tag{5.6}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function, $g : \mathbb{R}^n \to \mathbb{R}^{N_i}$ represents the inequality constraints while $h : \mathbb{R}^n \to \mathbb{R}^{N_e}$ the equality constraints. We set $N_c = N_i + N_e$. The process of solving (5.6) is referred to as **nonlinear programming (NLP)**. Some properties arising from optimality conditions in this section turn out to be required properties for the sensitivity analysis to be discussed afterwards. In addition, we also present an algorithm to solve (5.6) which we use throughout the thesis.

We call the set

$$\Sigma := \left\{ z \;\middle|\; \begin{array}{l} g_j(z) \le 0, \; j = 1, \ldots, N_i \\ h_j(z) = 0, \; j = N_i + 1, \ldots, N_c \end{array} \right\}$$

the **admissible set** or the **feasible set**. Note that with the defined indexing, no index j repeats. The function

$$\mathcal{L}(z, \lambda, \mu) := f(z) + \mu^\top g(z) + \lambda^\top h(z)$$

is called the **Lagrangian function** and $\mu \in \mathbb{R}^{N_i}, \lambda \in \mathbb{R}^{N_e}$ are called **Lagrange multipliers** corresponding to the inequality and equality constraints, respectively.

Definition 5.4.1.
A point $z^* \in \Sigma$ is a **global minimizer** of (5.6) if

$$f(z^*) \le f(z) \quad \text{for all } z \in \Sigma$$

A point $z^* \in \Sigma$ is a **local minimizer** of (5.6) if there exists a neighborhood $\mathcal{N}(z^*)$ such that

$$f(z^*) \le f(z) \quad \text{for all } z \in \mathcal{N}(z^*) \cap \Sigma$$

A point $z^* \in \Sigma$ is a **strict local minimizer** of (5.6) if there exists a neighborhood $\mathcal{N}(z^*)$ such that

$$f(z^*) < f(z) \quad \text{for all } z \in \mathcal{N}(z^*) \cap \Sigma, \; z \ne z^*$$

Consider the following set of indices associated with an optimal solution z^* of (5.6)

$$\begin{array}{rcl} \mathfrak{Eq} & := & \{N_i + 1, \ldots, N_c\} \\ \mathfrak{In}(z^*) & := & \{j \in \{1, \ldots, N_i\} \mid g_j(z^*) = 0\} \\ \mathcal{A}(z^*) & := & \mathfrak{Eq} \cup \mathfrak{In}(z^*) \\ \mathcal{I}(z^*) & := & \{j \in \{1, \ldots, N_i\} \mid g_j(z^*) < 0\} \end{array}$$

The notation $\mathcal{A}(z^*)$ denotes the **index set of active constraints** while the notation $\{h_i, g_i \mid i \in \mathcal{A}(z^*)\}$ gives the **set of active constraints** for $z^* \in \Sigma$. The set $\mathcal{I}(z^*)$ is the **index set of inactive constraints** for z^* and $\{ g_i \mid i \in \mathcal{I}(z^*)\}$

is the **set of inactive constraints** for $z^* \in \Sigma$.

Suppose $\mathcal{I}(z^*) \neq \emptyset$, i.e., there exists $i_0 \in \{1, \dots, N_i\}$ such that $g_{i_0}(z^*) < 0$. Deleting the i_0-th inequality constraint does not change z^* from being the local minimizer of the problem (5.6). Thus assuming $\mathcal{A}(z^*)$ is the index set of active constraints for z^* for (5.6), then z^* is also the local minimizer of the equality constrained problem

$$\begin{array}{ll} \min & f(z) \\ \text{s.t.} & g_i(z) = 0, \; i \in \mathcal{A}(z^*) \\ & h(z) \; = 0 \end{array} \tag{5.7}$$

Convex problems

The optimization problem (5.6) is said to be a **convex problem** if it has a convex objective function and a convex feasible region.

If g is convex and h is linear, then the feasible region Σ is convex. Indeed, suppose Σ is not convex. Then there exist $x, y \in \Sigma$ and $\alpha \in (0, 1)$ such that $z := \alpha x + (1-\alpha)y \notin \Sigma$ which means either $g(z) > 0$ or $h(z) \neq 0$. Since g is convex, $0 < g(z) = g(\alpha x + (1-\alpha))y \leq \alpha g(x) + (1-\alpha)g(y) \leq 0$ giving a contradiction. Since h is linear, $0 \neq h(z) = h(\alpha x + (1-\alpha)y) = \alpha h(x) + (1-\alpha)h(y) = 0$ which also gives a contradiction. If, in addition, $f(z)$ is convex, then (5.6) is a convex problem.

Linear and quadratic programming problems

Problems of the form

$$\begin{array}{lrcl} \min\limits_z & & & c^\top z \\ \text{s.t.} & Az + b & \leq & 0 \\ & A_{\text{eq}} z + b_{\text{eq}} & = & 0 \end{array}$$

called linear programming (LP), and

$$\begin{array}{lrcl} \min\limits_z & & & h^\top z + \dfrac{1}{2} z^\top B z \\ \text{s.t.} & Az + b & \leq & 0 \\ & A_{\text{eq}} z + b_{\text{eq}} & = & 0 \end{array}$$

with positive semidefinite matrix B, called quadratic programming (QP), are convex problems.

Nonconvex problems

For general NLP, nonlinear equality constraints render a problem nonconvex even if $f(z)$ is convex. A nonconvex problem may have multiple local minimizer increasing the complexity to identify whether the problem has no solution or has a global minimizer. In this case, one can then limit the analysis to a local setting.

The key advantage when (5.6) is a convex problem is given in the following theorem.

Theorem 5.4.2. *If the optimization problem* (5.6) *is convex, then every local minimizer in* Σ *is a global minimizer.*

Proof. Similar to the proof of Theorem 5.2.2. In this case, the convexity of Σ guarantees that the point $\alpha x + (1-\alpha)y$ is feasible for feasible points x and y. $\square$

Definition 5.4.3. Let $z' \in \Sigma$ and $d \in \mathbb{R}^n$. Then d is said to be a **descent direction at** z' if $\nabla f(z')^\top d < 0$. We define the set

$$\mathcal{D}(z') = \{d \in \mathbb{R}^n \mid \nabla f(z')^\top d < 0\}$$

as the set of all descent directions at z'.

Definition 5.4.4. Let $z' \in \Sigma$ and $d \in \mathbb{R}^n \backslash \{0\}$. If there exists $\delta > 0$ such that

$$z' + td \in \Sigma \text{ for all } t \in [0, \delta]$$

then d is said to be a **feasible direction of** Σ **at** z'. The set

$$\mathcal{F}_\Sigma(z') = \{d \in \mathbb{R}^n \backslash \{0\} \mid \exists \delta > 0 \text{ s.t. } z' + td \in \Sigma \ \forall t \in [0, \delta]\}$$

contains all feasible directions of Σ at z'.

In the following, we define certain cone conditions derived from the linearization of the active constraints.

Definition 5.4.5. Let $z' \in \Sigma$. The set of all **linearized feasible directions** given by

$$\mathcal{C}_\Sigma(z') = \left\{ d \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} \nabla h_i(z')^\top d = 0, & i \in \mathfrak{Eq} \\ \nabla g_i(z')^\top d \leq 0, & i \in \mathcal{A}(z') \end{array} \right\}$$

is called the **linearized feasible cone**.

Definition 5.4.6. Let $z' \in \Sigma$ and $d \in \mathbb{R}^n$. If there exist a sequence $\{d^k\}$ and a positive sequence $\{\delta^k\}$ such that $z' + \delta^k d^k \in \Sigma$ for all k with $d^k \to d$ and $\delta^k \to 0$, then the limiting direction d is called the **sequential feasible direction of** Σ **at** z'. The set

$$\mathcal{S}_\Sigma(z') = \left\{ d \in \mathbb{R}^n \;\middle|\; \begin{array}{l} z' + \delta^k d^k \in \Sigma \quad \forall k \\ d^k \to d, \delta^k \to 0 \end{array} \right\}$$

is the set of all sequential feasible directions of Σ at z'.

From Definition 5.4.6, setting $z^k := z' + \delta^k d^k$, we obtain $z^k \to z'$. In addition, setting $\delta^k := \|z^k - z'\|$ gives $d^k = \dfrac{z^k - z'}{\|z^k - z'\|} \to d$. Thus, $\{z^k\}$ is a feasible point sequence with limiting direction d.

Definition 5.4.7. We define the **tangent cone** of Σ at z'

$$\mathcal{T}_\Sigma(z') = \mathcal{S}_\Sigma(z') \cup \{0\}$$

Lemma 5.4.8. *Let $z' \in \Sigma$. If g, h are differentiable at z', then*

$$\mathcal{F}_\Sigma(z') \subseteq \mathcal{S}_\Sigma(z') \subseteq \mathcal{C}_\Sigma(z')$$

Proof. See proof in [64, Lemma 8.2.4]. $\square$

Theorem 5.4.9. *Let z^* be a local minimizer of* (5.6)*. If f, g, h are differentiable at z^*, then*

$$\nabla f(z^*)^\top d \geq 0 \quad \textit{for all } d \in \mathcal{S}_\Sigma(z^*)$$

Proof. See proof in [64, Lemma 8.2.5]. □

Lemma 5.4.10 (Restatement of Farkas' lemma)**.** *The equality*

$$S := \left\{ d \in \mathbb{R}^n \;\middle|\; \begin{array}{ll} \nabla f(z^*)^\top d < 0, & \\ \nabla h_i(z^*)^\top d = 0, & i \in \mathfrak{Eq} \\ \nabla g_i(z^*)^\top d \leq 0, & i \in \mathcal{A}(z^*) \end{array} \right\} = \emptyset$$

holds if and only if there exist $\lambda_i \in \mathbb{R}, i \in \mathfrak{Eq}$ and $\mu_i \geq 0, i \in \mathfrak{In}(z^*)$ such that

$$\nabla f(z^*) + \sum_{i \in \mathfrak{Eq}} \lambda_i \nabla h_i(z^*) + \sum_{i \in \mathfrak{In}(z^*)} \mu_i \nabla g_i(z^*) = 0$$

Proof. By using Theorem 5.1.2, with $v = -d$, $b = -\nabla f(z^*)$, $C = \nabla g(z^*)$, $D = \nabla h(z^*)$, $x = \mu$ and $y = \lambda$. □

We next introduce a constraint qualification that ensures that the sequential feasible direction at a solution can be represented by the linearizations of active constraints at that point.

Definition 5.4.11. Given a local solution z^* of (5.6) and the index set of active constraints $\mathcal{A}(z^*)$, **linear independence constraint qualification (LICQ)** holds if the constraint gradients

$$\nabla g_i(z^*), \nabla h_i(z^*), i \in \mathcal{A}(z^*)$$

are linearly independent.

Lemma 5.4.12. *If LICQ holds at z^*, then $\mathcal{T}_\Sigma(z^*) = \mathcal{C}_\Sigma(z^*)$.*

Proof. See proof in [49, Lemma 12.2]. □

Now, we are ready to state the first-order necessary condition for (5.6).

Theorem 5.4.13 (First-order necessary condition)**.** *If z^* is a local minimizer of* (5.6) *at which LICQ holds, there exists $\lambda^* \in \mathbb{R}^{N_e}$ and $\mu^* \in \mathbb{R}^{N_c}$ such that*

$$\nabla \mathcal{L}(z^*, \lambda^*, \mu^*) := \nabla f(z^*) + \nabla g(z^*)^\top \mu^* + \nabla h(z^*)^\top \lambda^* = 0 \tag{5.8}$$

$$g(z^*) \leq 0, \quad h(z^*) = 0 \tag{5.9}$$

$${\mu^*}^\top g(z^*) = 0, \quad \mu^* \geq 0 \tag{5.10}$$

Proof. Since $z^* \in \Sigma$, (5.9) follows. Let $d \in \mathcal{T}_\Sigma(z^*)$. Since z^* is a local minimizer, by Definition 5.4.7 and Theorem 5.4.9, $\nabla f(z^*)^\top d \geq 0$. In addition, by LICQ and Lemma 5.4.12, $d \in \mathcal{C}_\Sigma(z^*)$. This means that the system

$$\left\{ \begin{array}{ll} \nabla f(z^*)^\top d < 0 & \\ \nabla h_i(z^*)^\top d = 0, & i \in \mathfrak{Eq} \\ \nabla g_i(z^*)^\top d \leq 0, & i \in \mathfrak{In}(z^*) \end{array} \right.$$

has no solution in $\mathbb{R}^n$. Then by Farkas' Lemma,

$$\nabla f(z^*) + \sum_{i\in\mathfrak{Eq}} \lambda_i^* \nabla h_i(z^*) + \sum_{i\in\mathfrak{In}(z^*)} \mu_i^* \nabla g_i(z^*) = 0$$

where $\lambda_i^* \geq 0, i \in \mathfrak{Eq}$ and $\mu_i^* \geq 0, i \in \mathfrak{In}(z^*)$. Set $\mu_i^* = 0$ for $i \in \mathcal{I}(z^*)$, then

$$\nabla f(z^*) + \nabla g(z^*)^\top \mu^* + \nabla h(z^*)^\top \lambda^* = 0$$

which shows (5.8) and that $\mu^* \geq 0$. Lastly, if $i \in \mathfrak{In}(z^*)$, then $g_i(z^*) = 0$ giving ${\mu^*}^\top g(z^*) = 0$ and if $i \in \mathcal{I}(z^*)$, since we set $\mu_i^* = 0$, then ${\mu^*}^\top g(z^*) = 0$. This shows (5.10). □

Conditions (5.8)–(5.10) are called the **Karush-Kuhn-Tucker (KKT) conditions**. This set of conditions is comprised of condition (5.8) called the stationary point condition, (5.9) called the feasibility conditions and (5.10) giving the nonnegativity of the multipliers and complementarity condition.

Next we examine the second-order necessary conditions. To this end, we first refine the definition of the index set $\mathcal{A}$. We define

$$\mathcal{A}_S(z^*) = \{\ i \in \mathfrak{In}(z^*) \mid \mu_i^* > 0\}$$

as the **index set of strongly active constraints** and

$$\mathcal{A}_W(z^*) = \{\ i \in \mathfrak{In}(z^*) \mid \mu_i^* = 0\}$$

as the **index set of weakly active constraints**. Given a local minimizer z^* of (5.6) together with multipliers λ^*, μ^* satisfying (5.8) and (5.10), **strict complementarity** is said to occur if $\mathcal{A}_W(z^*) = \emptyset$.

We now consider the critical cone

$$\mathcal{G}_\Sigma(z^*) = \left\{ d \in \mathbb{R}^n \;\middle|\; \begin{array}{l} \nabla h(z^*)^\top d = 0, \\ \nabla g_i(z^*)^\top d = 0,\ i \in \mathcal{A}_S(z^*) \\ \nabla g_i(z^*)^\top d \leq 0,\ i \in \mathcal{A}_W(z^*) \end{array} \right\} \tag{5.11}$$

We now state the following constrained optimization second-order conditions.

Theorem 5.4.14 (Second-order necessary condition)**.** *If z^* is a local minimizer of* (5.6) *at which LICQ holds together with λ^*, μ^* satisfying the KKT conditions* (5.8)*–*(5.10)*, then*

$$d^\top \nabla^2 \mathcal{L}(z^*, \lambda^*, \mu^*) d \geq 0 \quad \textit{for all } d \in \mathcal{G}_\Sigma(z^*)$$

Proof. See, e.g., proofs of [11, Theorem 4.17] or [64, Theorem 8.3.3] □

Theorem 5.4.15 (Second-order sufficient conditions (SOSC))**.** *If z^* and the multipliers λ^*, μ^* satisfy the KKT conditions* (5.8)*–*(5.10) *and*

$$d^\top \nabla \mathcal{L}(z^*, \lambda^*, \mu^*) d > 0 \quad \textit{for all nonzero } d \in \mathcal{G}_\Sigma(z^*) \tag{5.12}$$

then z^* is a strict local minimizer of (5.6)*.*

Proof. See, e.g., proofs of [11, Theorem 4.18] or [64, Theorem 8.3.4] □

5.4.1 Equality constrained optimization problems

We now adapt the Newton-based method (see Algorithm 5.3.1) that solves unconstrained optimization problems to the constrained setting. We first consider the constrained problem

$$\begin{array}{ll} \min & f(z) \\ \text{s.t.} & h(z) = 0 \end{array} \tag{5.13}$$

with only equality constraints.

Assuming LICQ, the KKT conditions for (5.13) read

$$\begin{aligned} \nabla \mathcal{L}(z, \lambda) = \nabla f(z) + \nabla h(z)^\top \lambda &= 0 \\ h(z) &= 0 \end{aligned}$$

Let

$$w = \begin{bmatrix} z \\ \lambda \end{bmatrix} \quad \text{and} \quad F(w) = \begin{bmatrix} \nabla \mathcal{L}(z, \lambda) \\ h(z) \end{bmatrix} = \begin{bmatrix} \nabla f(z) + \nabla h(z)^\top \lambda \\ h(z) \end{bmatrix}$$

We apply Newton's method to solve $F(w) = 0$. At a point of interest w^k, the linearized system can be written as

$$\begin{aligned} F(w^k) + \nabla_w F(w^k)^\top (w - w^k) &= 0 \\ \begin{bmatrix} \nabla \mathcal{L}(z^k, \lambda^k) \\ h(z^k) \end{bmatrix} + \begin{bmatrix} (\nabla_w F_1(w^k))^\top \\ (\nabla_w F_2(w^k))^\top \end{bmatrix} \begin{bmatrix} z - z^k \\ \lambda - \lambda^k \end{bmatrix} &= 0 \\ \begin{bmatrix} \nabla \mathcal{L}(z^k, \lambda^k) \\ h(z^k) \end{bmatrix} + \begin{bmatrix} \nabla^2 \mathcal{L}(z^k, \lambda^k) & \nabla h(z^k) \\ \nabla h(z^k)^\top & 0 \end{bmatrix} \begin{bmatrix} z - z^k \\ \lambda - \lambda^k \end{bmatrix} &= 0 \quad (5.14) \end{aligned}$$

where the coefficient matrix $\begin{bmatrix} \nabla^2 \mathcal{L}(z^k, \lambda^k) & \nabla h(z^k) \\ \nabla h(z^k)^\top & 0 \end{bmatrix}$ is called the **KKT matrix**. Finding an update rule $z^{k+1} = z^k + \Delta z^k$ and $\lambda^{k+1} = \lambda^k + \Delta \lambda^k$, since $\nabla \mathcal{L}(z^k, \lambda^k) = \nabla f(z^k) + \nabla g(z^k) \lambda^k$, from (5.14) we obtain

$$\begin{bmatrix} \nabla f(z^k) \\ h(z^k) \end{bmatrix} + \begin{bmatrix} \nabla^2 \mathcal{L}(z^k, \lambda^k) & \nabla h(z^k) \\ \nabla h(z^k)^\top & 0 \end{bmatrix} \begin{bmatrix} \Delta z^k \\ \lambda^{k+1} \end{bmatrix} = 0 \tag{5.15}$$

Solving the linear system (5.15) allows to compute z^{k+1} and λ^{k+1}. This gives us the Newton Lagrange method [2] for solving the equality constrained problem (5.13).

Algorithm 5.4.16. (Newton Lagrange method)
Choose a starting point z^0, λ^0 and tolerance ϵ.

(1) If $\|F(w^k)\| < \epsilon$, stop. For z^k, λ^k, solve the linear system (5.15).

(2) Set $z^{k+1} = z^k + \Delta z^k$ and $k = k + 1$.

Since the Algorithm 5.4.16 is applying the root-finding Newton's Method to $F(w) = 0$, similar to Algorithm 5.3.1, one can also employ a choice of step length γ^k yielding instead an update rule $z^{k+1} = z^k + \gamma^k \Delta z^k$.

It is easy to see that $\begin{bmatrix} \Delta z^k \\ \lambda^{k+1} \end{bmatrix}$ also happens to be the solution of the quadratic programming problem

$$\begin{array}{ll} \min\limits_{\Delta z^k} & \nabla f(z^k)^\top \Delta z^k + \frac{1}{2}\Delta z^{k\top} \nabla^2 \mathcal{L}(z^k, \lambda^k)\Delta z^k \\ \text{s.t.} & \nabla h(z^k)^\top \Delta z^k + h(z^k) = 0 \end{array} \tag{5.16}$$

Indeed, in determining the KKT conditions for (5.16), we recover (5.15).

Remark 5.4.17. One can see that solving an arbitrary optimization problem of the form (5.13) by the Newton Lagrange method is equivalent to solving a sequence of quadratic programming problems (5.16) until convergence to the solution.

Similar to Algorithm 5.3.1, Algorithm 5.4.16 can also be adapted to tackle challenges in calculating derivatives and handling large-scale problems resulting in large matrices in order to effectively keep the computational costs to a tolerable level.

5.4.2 Inequality constrained optimization problems

We consider first the QP

$$\begin{array}{ll} \min\limits_{z} & h^\top z + \frac{1}{2} z^\top B z \\ \text{s.t.} & Az + b \le 0 \end{array} \tag{5.17}$$

where B is positive semidefinite making the problem convex. The corresponding Lagrangian function is $\mathcal{L}(z, \mu) = h^\top z + \frac{1}{2} z^\top B z + \mu^\top b + \mu^\top Az$. The KKT conditions are

$$\begin{array}{rcl} \nabla \mathcal{L}(z, \mu) = Bz + h + A^\top \mu & = & 0 \\ Az + b & \le & 0 \\ \mu \ \ge \ 0, \ \ (Az + b)^\top \mu & = & 0 \end{array} \tag{5.18}$$

Suppose z^* is a global minimizer of (5.17). Now the left-hand side of inequality (5.18) can be decomposed as

$$\begin{bmatrix} A_{\mathcal{A}} \\ A_{\mathcal{I}} \end{bmatrix} z^* + \begin{bmatrix} b_{\mathcal{A}} \\ b_{\mathcal{I}} \end{bmatrix}$$

where $A_{\mathcal{A}} z^* + b_{\mathcal{A}} = 0$ represents the active constraints while $A_{\mathcal{I}} z^* + b_{\mathcal{I}} < 0$ the inactive.

Theorem 5.4.18. *z^* is a global minimizer of (5.17) if and only if there exist index sets $\mathcal{A}$ and $\mathcal{I}$ and a vector $\mu^*_{\mathcal{A}}$ such that*

$$Bz^* + h + A_{\mathcal{A}}^\top \mu^*_{\mathcal{A}} = 0 \tag{5.19}$$
$$A_{\mathcal{A}} z^* + b_{\mathcal{A}} = 0 \tag{5.20}$$
$$A_{\mathcal{I}} z^* + b_{\mathcal{I}} < 0 \tag{5.21}$$
$$\mu^*_{\mathcal{A}} \ge 0 \tag{5.22}$$

with $\mu^*_{\mathcal{I}} = 0$ where $\mu^* = \begin{bmatrix} \mu^*_{\mathcal{A}} \\ \mu^*_{\mathcal{I}} \end{bmatrix}$.

Equations (5.19) and (5.20) become

$$\begin{bmatrix} B & A_{\mathcal{A}}^\top \\ A_{\mathcal{A}} & 0 \end{bmatrix} \begin{bmatrix} z^* \\ \mu_{\mathcal{A}}^* \end{bmatrix} = - \begin{bmatrix} h \\ b_{\mathcal{A}} \end{bmatrix} \tag{5.23}$$

We can then apply the so-called **active-set strategy** to solve (5.17). First we choose an initialization of set $\mathcal{A}$, we solve for the solution z^* and $\mu_{\mathcal{A}}^*$ of (5.23) and check if these satisfy (5.21) and (5.22). If (5.21) and (5.22) are satisfied, then the correct index set $\mathcal{A}$ has been found. Through the process, either we have found the correct $\mathcal{A}$ or we keep on modifying $\mathcal{A}$ until the correct one is found. Details of the active-set strategy are presented in an MPC implementation in Section 7.2.

5.4.3 Active-set sequential quadratic programming

We are now in the position to finally solve the NLP (5.6). The Lagrangian function is given by $\mathcal{L}(z,\lambda,\mu) = f(z) + \mu^\top g(z) + \lambda^\top h(z)$. Let $C(z) = \begin{bmatrix} g(z) \\ h(z) \end{bmatrix}$ and $\eta = \begin{bmatrix} \mu \\ \lambda \end{bmatrix}$. As in the discussion above, the constraint $g(z) \leq 0$ can also be decomposed into its active and inactive components. From Remark 5.4.17 and from the discussed active-set strategy, we solve the sequence of QPs

$$\begin{array}{ll} \min\limits_{\Delta z^k} & \nabla f(z^k)^\top \Delta z^k + \frac{1}{2} {\Delta z^k}^\top \nabla^2 \mathcal{L}(z^k,\eta^k) \Delta z^k \\ \text{s.t.} & \nabla g(z^k)^\top \Delta z^k + g(z^k) \leq 0 \\ & \nabla h(z^k)^\top \Delta z^k + h(z^k) = 0 \end{array} \tag{5.24}$$

and similar to how we solve (5.17), the optimal solution z^* and $\eta_{\mathcal{A}}^*$ is obtained by solving the system

$$\begin{bmatrix} \nabla^2 \mathcal{L}(z^k,\eta^k) & \nabla C_{\mathcal{A}}(z^k) \\ \nabla C_{\mathcal{A}}(z^k)^\top & 0 \end{bmatrix} \begin{bmatrix} \Delta z^k \\ {\eta^{k+1}}_{\mathcal{A}} \end{bmatrix} = - \begin{bmatrix} \nabla f(z^k) \\ C_{\mathcal{A}}(z^k) \end{bmatrix} \tag{5.25}$$

with the active-set strategy where

$$\nabla C(z) = (\nabla g(z), \nabla h(z)) \text{ and } \nabla C_{\mathcal{A}}(z) = (\{\nabla g_i(z)\}_{i\in\mathcal{A}}, \nabla h(z))$$

where $\mathcal{A}$ is defined to be the index set of all active constraints. In (5.25), $\eta_{\mathcal{A}}$ denotes the multipliers and $\nabla C_{\mathcal{A}}(z^k)^\top$ the Jacobian corresponding to the active constraints. The method results in an iterative update $z^{k+1} = z^k + \Delta z^k$. Here, a sequence of QPs (5.24) is solved until the iterates converge. This procedure is the so-called **sequential quadratic programming (SQP)**. In this work, as detailed in the next chapter, we use SQP to solve (5.6) and exploit the matrix structures arising from the formulation in order to design an MPC approach based on sensitivity analysis.

5.5 Sensitivity analysis

In this section, we present some results on **parametric sensitivity analysis** (studies originally conducted in [23, 24, 58]) which refers to the impact of a change in the design parameters on the optimal solution vector and the objective function. From the mentioned works, differentiability of optimal solutions as functions of parameters are shown. The main result from these works that we will use in our study is the explicit formula for computing the sensitivity derivatives of the optimal solution and the corresponding Lagrange multipliers.

We now consider the parametric NLP problem

$$\begin{array}{lll} \min\limits_{z} & f(z,p), & \\ \text{such that} & g_j(z,p) \leq 0, & j = 1,\dots,N_i, \\ & h_j(z,p) = 0, & j = N_i+1,\dots,N_c. \end{array} \qquad P(p)$$

with optimization variable $z \in \mathbb{R}^n$ depending on the parameter $p \in \mathbb{R}^q$. Let the functions f, g_j, $j = 1,\dots,N_i$ and h_j, $j = N_i+1,\dots,N_c$ be twice continuously differentiable on $\mathbb{R}^n \times \mathbb{R}^q$.

Problem $P(p)$ is of the form (5.6) additionally featuring the dependence of the functions and therefore the problem, on the parameter p. Let μ and λ be the Lagrange multipliers corresponding to the inequality and equality constraints, respectively, with $\eta = \begin{bmatrix} \mu \\ \lambda \end{bmatrix}$. For a fixed parameter p, the definition of the feasible set $\Sigma(p)$, global, local and strict minimizer and index set $\mathcal{A}(z,p)$ of active constraints at optimal solution $z^* \in \Sigma(p)$ defined in Section 5.4 still hold for problem $P(p)$. The theorems pertaining to the constrained optimization still hold, namely, the first and second-order necessary conditions, Theorems 5.4.13 and 5.4.14, respectively, and the second-order sufficient conditions Theorem 5.4.15.

We now present the differential properties of the optimal solutions to the perturbed problems $P(p)$. The following theorem shows that the optimal solutions are differentiable functions of the parameter.

Theorem 5.5.1 (Sensitivity theorem, Fiacco [23, 24])**.** *Consider the problem $P(p_0)$ with a nominal parameter p_0, optimal solution z^* and corresponding multiplier $\eta^*_{\mathcal{A}}$ for the active constraints. Suppose f, g and h are twice continuously differentiable in a neighborhood of z^* and SOSC, LICQ and strict complementarity hold at z^*. Then for a neighborhood $\mathcal{N}(p_0)$ of p_0 and a neighborhood $\mathcal{N}(z^*,\eta^*)$ of (z^*,η^*), there exist unique, continuously differentiable functions $z : \mathcal{N}(p_0) \to \mathbb{R}^n$ and $\eta : \mathcal{N}(p_0) \to \mathbb{R}^{N_c}$ with the following properties:*

(i) $z(p_0) = z^*, \quad \eta(p_0) = \eta^*$

(ii) the index set of active constraints are constant in $\mathcal{N}(p_0)$

(iii) LICQ holds for $z(p)$ for all $p \in \mathcal{N}(p_0)$

(iv) for all $p \in \mathcal{N}(p_0)$, $(z(p),\eta(p))$ satisfies SOSC for $P(p)$. In particular, $z(p)$ is a strict local minimizer of $P(p)$.

Proof. The proof follows the proofs presented in [27, Theorem 6.1.4] and [62, Satz 2.5.1]. First, let

$$\begin{aligned} \Delta &:= \operatorname{diag}(\mu_1,\dots,\mu_{N_i}) \\ \Delta^* &:= \operatorname{diag}(\mu^*_1,\dots,\mu^*_{N_i}) \\ \Gamma^* &:= \operatorname{diag}(g_1(z^*,p_0),\dots,g_{N_i}(z^*,p_0)) \end{aligned}$$

The KKT conditions (5.8)–(5.10) for an arbitrary pair $(z, \eta) = (z(p), \eta(p))$ are given by[1]

$$\nabla \mathcal{L}(z, \eta, p) := \nabla f(z, p) + \nabla g(z, p)^\top \mu + \nabla h(z, p)^\top \lambda = 0 \tag{5.26}$$
$$\Delta g(z, p) = 0 \tag{5.27}$$
$$h(z, p) = 0 \tag{5.28}$$

By letting $w = \begin{bmatrix} z \\ \eta \end{bmatrix} = \begin{bmatrix} z \\ \mu \\ \lambda \end{bmatrix}$, (5.26)–(5.28) can be written as $K(w, p) = 0$. From the assumptions, K is continuously differentiable and $K(w^*, p_0) = 0$ where $w^* = \begin{bmatrix} z^* \\ \eta^* \end{bmatrix}$.

We show next that the implicit function theorem (see, e.g., [27, Theorem 2.1.14]) is applicable on $K(w, p) = 0$. To this end, we need to show that $\frac{\partial}{\partial w} K(w^*, p_0)$ is nonsingular. First, we have

$$\frac{\partial}{\partial w} K(w^*, p_0) = \begin{bmatrix} \nabla^2 \mathcal{L}(w^*, p_0) & \nabla g(z^*, p_0) & \nabla h(z^*, p_0) \\ \Delta^* \nabla g(z^*, p_0)^\top & \Gamma^* & 0 \\ \nabla h(z^*, p_0)^\top & 0 & 0 \end{bmatrix} \tag{5.29}$$

Without loss of generality, let $\{\ell + 1, \ldots, N_i\}$ be the set of indices of the active inequality constraints. Due to strict complementarity,

$$\Delta^* = \begin{bmatrix} 0 & 0 \\ 0 & \Delta_2^* \end{bmatrix} \text{ where } \Delta_2^* = \operatorname{diag}(\mu_{\ell+1}^*, \ldots, \mu_{N_i}^*) \text{ is nonsingular} \tag{5.30}$$

$$\Gamma^* = \begin{bmatrix} \Gamma_1^* & 0 \\ 0 & 0 \end{bmatrix} \text{ where } \Gamma_1^* = \operatorname{diag}(g_1(x^*, 0), \ldots, g_\ell(x^*, 0)) \text{ is nonsingular} \tag{5.31}$$

Consider

$$\frac{\partial}{\partial w} K(w^*, p_0) \begin{bmatrix} v_1 \\ v_{21} \\ v_{22} \\ v_3 \end{bmatrix} = 0 \tag{5.32}$$

where $v_1 \in \mathbb{R}^n, v_{21} \in \mathbb{R}^\ell, v_{22} \in \mathbb{R}^{N_i - \ell}$ and $v_3 \in \mathbb{R}^{N_e}$

Using equations (5.29)–(5.32), we obtain $\Gamma_1^* v_{21} = 0$ which implies $v_{21} = 0$ due to nonsingularity of Γ_1^*. Hence, it suffices to consider the system

$$\begin{bmatrix} A & N \\ N^\top & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_{22} \\ v_3 \end{bmatrix} = 0 \tag{5.33}$$

$$\iff \quad A v_1 + N \begin{bmatrix} v_{22} \\ v_3 \end{bmatrix} = 0 \tag{5.34}$$

$$N^\top v_1 = 0 \tag{5.35}$$

[1] We append p to the notation $\mathcal{L}(z, \lambda, \mu) = \mathcal{L}(z, \eta)$ giving the notation $\mathcal{L}(z, \eta, p)$. Whenever we only consider the multipliers of the active constraints, we use $\mathcal{L}(z, \eta_{\mathcal{A}}, p)$.

where $A = \nabla^2 \mathcal{L}(w^*, p_0)$ and

$$N = \nabla C_{\mathcal{A}}(z^*, p_0) = [\nabla g_{\ell+1}(z^*, p_0), \ldots, \nabla g_{N_i}(z^*, p_0), \nabla h(z^*, p_0)]$$

Due to (5.35), by strict complementarity at z^*, we have $v_1 \in \mathcal{G}_\Sigma(z^*)$ with $\mathcal{G}_\Sigma(z^*)$ defined in (5.11). Multiplying $v_1^\top$ to both sides of (5.34), we obtain $v_1^\top A v_1 + (N^\top v_1)^\top \begin{bmatrix} v_{22} \\ v_3 \end{bmatrix} = 0$ and by (5.35), $v_1^\top A v_1 = 0$, thus by (5.12) of SOSC, v_1 must be 0. Thus, (5.34) becomes $N \begin{bmatrix} v_{22} \\ v_3 \end{bmatrix} = 0$ and by LICQ, i.e., N has full column rank, $\begin{bmatrix} v_{22} \\ v_3 \end{bmatrix} = 0$. These show that $(v_1, v_{21}, v_{22}, v_3)^\top$ in (5.32) must be 0 which means $\dfrac{\partial}{\partial w} K(w^*, p_0)$ is nonsingular implying the applicability of the implicit function theorem.

By the implicit function theorem, there exist neighborhoods $\mathcal{N}(p_0)$ and $\mathcal{N}(w^*)$ and a uniquely defined function $w : \mathcal{N}(p_0) \to \mathcal{N}(w^*)$ satisfying $K(w(p), p) = 0$ for all $p \in \mathcal{N}(p_0)$. The total differentiation of the identity $K(w(p), p) = 0$ with respect to p then yields the following linear system

$$\left(\frac{\partial}{\partial w} K(w,p)\right) \frac{\partial}{\partial p} w(p) \; + \; \frac{\partial}{\partial p} F(w,p)\Big|_{w=w(p)} \; = \; 0$$

Thus, the function w is continuously differentiable in p with

$$\frac{\partial}{\partial p} w(p) = -\left(\frac{\partial}{\partial w} K(w,p)\right)^{-1} \frac{\partial}{\partial p} F(w,p)\Big|_{w=w(p)} \tag{5.36}$$

Now, we show properties (i) to (iv). Due to uniqueness of the function w, and since w^* is the optimal solution for p_0, then $w(p_0) = w^*$ showing (i). Since $\eta^*_{\ell+1}, \ldots, \eta^*_{N_c} > 0$, $g_1(z^*, p_0), \ldots, g_\ell(z^*, p_0) > 0$. Then for p sufficiently close to p_0,

$$\eta_{\ell+1}(p), \ldots, \eta_{N_c}(p) > 0 \text{ and } g_1(z(p), p), \ldots, g_\ell(z(p), p) > 0$$

and since $K(w(p), p) = 0$, we obtain

$$\eta_1(p) = \ldots = \eta_\ell(p) = 0 \text{ and } g_{\ell+1}(z(p), p) = \ldots = g_m(z(p), p) = 0$$

implying strict complementarity at $z(p)$, and since $h(z(p), p) = 0$, $z(p) \in \Sigma(p)$ and $\mathcal{A}(z(p)) = \mathcal{A}$ showing (ii). Due to the continuity of the first derivative, for p sufficiently close to p_0, $\nabla C_{\mathcal{A}}(z, p)$ has full column rank giving LICQ showing (iii). Lastly, to show (iv), since the critical cone $\mathcal{G}_\Sigma(z(p))$ varies with p, one needs to show that for p sufficiently close to p_0, $d^\top \nabla^2 \mathcal{L}(w(p), p) d$ remains positive for nonzero $d \in \mathcal{G}_\Sigma(z(p))$. We refer to [27, proof of Theorem 6.1.4] for the details of this final step. □

In the proof of Theorem 5.5.1, let us examine the case where we only consider the active constraints. The KKT conditions (5.8)–(5.10) for an arbitrary pair $(z, \eta_{\mathcal{A}}) = (z(p), \eta_{\mathcal{A}}(p))$ along with the definition of the active constraints give

$$\nabla \mathcal{L}(z, \eta_{\mathcal{A}}, p) := \nabla f(z, p) + \nabla C_{\mathcal{A}}(z, p)^\top \eta_{\mathcal{A}} = 0 \tag{5.37}$$

$$C_{\mathcal{A}}(z, p) = 0 \tag{5.38}$$

Redefine $w = \begin{bmatrix} z \\ \eta_{\mathcal{A}} \end{bmatrix}$. Then (5.37)–(5.38) can be written as $K(w,p) = 0$. From the assumptions, K is continuously differentiable and $K(w^*, p_0) = 0$ where $w^* = \begin{bmatrix} z^* \\ \eta^*_{\mathcal{A}} \end{bmatrix}$. Consider

$$\frac{\partial}{\partial w} K(w^*, p_0) = \begin{bmatrix} \nabla^2 \mathcal{L}(w^*, p_0) & \nabla C_{\mathcal{A}}(z^*, p_0) \\ \nabla C_{\mathcal{A}}(z^*, p_0)^\top & 0 \end{bmatrix} \tag{5.39}$$

Using the same technique as in the proof of Theorem 5.5.1, one shows that $\frac{\partial}{\partial w} K(w^*, p_0)$ is nonsingular implying the applicability of the implicit function theorem. Thus, there exist neighborhoods $\mathcal{N}(p_0)$ and $\mathcal{N}(w^*)$ and a uniquely defined function $w : \mathcal{N}(p_0) \to \mathcal{N}(w^*)$ satisfying $K(z(p), p) = 0$ for all $p \in \mathcal{N}(p_0)$. Moreover, the function w is continuously differentiable in p with (5.36) where $\frac{\partial}{\partial w} K(w, p)$ is given by (5.39). This provides an application of Theorem 5.5.1 to approximate solutions of perturbed OCPs as presented in the subsequent remark.

We make use of the following definition of order of approximation.

Definition 5.5.2. Let D be the domain of q and r. If for every compact $K \subset D$, there exists $C > 0$ such that $\|q(x) - r(x)\| \leq C h^{p+1}$ for every $x \in K$, then we write

$$q(x) = r(x) + \mathcal{O}(h^{p+1})$$

In this case, $r(x)$ is called a **pth order approximation of** $q(x)$ where the order of magnitude of the error is at most h^{p+1}, or in terms of big $\mathcal{O}$ notation, the error is $\mathcal{O}(h^{p+1})$.

Remark 5.5.3. (a) Based on (5.36) with $\frac{\partial}{\partial w} K(w, p)$ given in (5.39), the **sensitivity differentials** or simply, **sensitivities of the optimal solution** z^* **and corresponding multiplier** $\eta^*_{\mathcal{A}}$ is given by solving the system

$$\begin{bmatrix} \nabla^2 \mathcal{L}(w^*, p_0) & \nabla C_{\mathcal{A}}(z^*, p_0) \\ \nabla C_{\mathcal{A}}(z^*, p_0)^\top & 0 \end{bmatrix} \begin{bmatrix} \frac{\partial z}{\partial p}(p_0) \\ \frac{\partial \eta_{\mathcal{A}}}{\partial p}(p_0) \end{bmatrix} = - \begin{bmatrix} \nabla^2_{zp} \mathcal{L}(w^*, p_0)^\top \\ \nabla_p C_{\mathcal{A}}(z^*, p_0) \end{bmatrix} \tag{5.40}$$

(b) As an approach proposed in Büskens and Maurer [17], the sensitivity $\frac{\partial z}{\partial p}(p_0)$ allows for a first-order approximation of the optimal solution for a perturbed parameter via

$$z(p) = z^* + \frac{\partial z}{\partial p}(p_0)\,(p - p_0) + \mathcal{O}\left(\|p - p_0\|^2\right) \tag{5.41}$$

In reference to the nominal problem $P(p_0)$, we can regard $P(p)$ as a perturbed problem for which the solution can be approximated by (5.41) through the availability of the nominal solution $z^*(p_0)$, the perturbation $p - p_0$ and the sensitivity $\frac{\partial z}{\partial p}(p_0)$.

(c) Note that the coefficient matrix $\begin{bmatrix} \nabla^2\mathcal{L}(z^*, \eta^*_{\mathcal{A}}, p_0) & \nabla C_{\mathcal{A}}(z^*, p_0) \\ \nabla C_{\mathcal{A}}(z^*, p_0)^\top & 0 \end{bmatrix}$ of (5.40), also called as the **KKT-matrix**, coincides with the coefficient matrix of the system (5.25) as $k \to \infty$, i.e., $z_k \to z^*$. As a consequence, the sensitivity $\dfrac{\partial z}{\partial p}(p_0)$ can easily be obtained by solving a linear system with a coefficient matrix obtained when the SQP converges. Due to this, [23] (as reported in [17]) describes sensitivities as a byproduct of optimization. In other words, the sensitivity $\dfrac{\partial z}{\partial p}(p_0)$ is obtained by taking advantage of already available information without having to build a new coefficient matrix which, otherwise, usually entails considerable expense. We take advantage of this result in the subsequent chapters.

(d) The coincidence of the KKT matrix to the coefficient matrix of the SQP system upon convergence, as pointed in (c), no longer holds if one uses a Newton-type method (as discussed in Section 5.3) where the exact computation of the coefficient matrix of the SQP system is replaced by an approximation in order to reduce computational cost. In this case, the sensitivity differentials can be accurately computed by a *post-optimal analysis* detailed in [17] which involves an exact calculation of the KKT matrix and then computing the sensitivities through either an LR-factorization of the the KKT matrix or RQ-factorization of (5.40).

6 Sensitivity-based multistep MPC

In this chapter, we construct an MPC variant that offers a considerable reduction in the computational expense compared to the standard and the updated MPC through NLP sensitivity analysis. Similar studies have been conducted in the past. For instance, the works [17, 46, 53] use sensitivities to achieve real-time approximations of the perturbed solutions based on an open-loop control obtained from solving an OCP. In the works [70, 69], sensitivities are used to construct a so-called *advanced-step* MPC controller allowing for a scheme with reduced computational delay. In this thesis, we design a particular MPC variant wherein we approximate the re-optimization performed in the updated m-step MPC through using sensitivity analysis. This is detailed in Section 6.1. In Section 6.2, we address challenges due to changes in the active constraints in order to ensure a valid sensitivity-based control. The development of this sensitivity-based scheme from the re-optimization-based scheme distinguishes our approach and its analysis from the existing works on sensitivity-based MPC algorithms in the literature. In Section 6.3, due to the approximation property of the new scheme, we show that the stability and performance analysis for the updated m-step can be carried over to this setting.

6.1 Design of the scheme

Our goal is to apply the sensitivity theorem in the MPC setting. Recall that we solve an OCP at each time step of Algorithm 1.3.1. Therefore, applying (5.41) is suitable in the case we want to approximate a perturbed solution whenever the information on the solution of a reference (e.g. nominal) problem is available. Recall that by the dynamic programming principle, the tails of an optimal control are also optimal controls for succeeding time instants using shorter optimization horizons and modified initial values. Therefore, for succeeding time instants, the perturbed solutions can be approximated using these already available tails as the reference nominal solutions.

6.1.1 MPC OCP as a parametric NLP

In Algorithms 1.3.1 and 1.3.4, we solve OCPs at each time step. In this section, we first write the full details of the MPC OCP in order to determine exactly where sensitivity analysis can enter the setting. The OCP is composed of the objective function $J_N(x_0, u(\cdot))$ defined in (1.4), with constraints that each control value u must be admissible, i.e., $u(\cdot) \in \mathbb{U}^N(x_0)$. Admissibility requires that $x_u(k+1, x_0) \in \mathbb{X}$ and $u(k) \in \mathbb{U}(x_u(k, x_0))$ for all k. Here we lay out the details of the formulation and write it as a parametric NLP.

Let us consider a plant with dynamics given by the discrete-time model (1.2). Let us use the notation $x_j := x(j)$ and $u_j := u(j)$ giving

$$x_{j+1} = f(x_j, u_j)$$

We set p to be the parameter and assign it to be the initial state value. Recall first the definition of $\mathcal{P}_N(x_0)$ in Section 1.1. We consider the OCP $\mathcal{P}_N(p)$ given by

$$\begin{array}{llll}
\min\limits_{\substack{x_j, j=0,\dots,N \\ u_j, j=0,\dots,N-1}} & J_N(x_0, \dots, x_N, u_0, \dots, u_{N-1}) := \sum_{j=0}^{N-1} \omega_{N-j} \ell(x_j, u_j) + F(x_N) & & \\
\text{subject to} & & & (6.1) \\
\text{the initial value} & x_0 = p, & & \\
\text{dynamics} & x_{j+1} = f(x_j, u_j), & j = 0, \dots, N-1, & \\
\text{additional equality constraints} & h_j(x_j, u_j) = 0, & j = 0, \dots, N-1, & \\
\text{terminal equality constraint} & r^{\text{eq}}(x_N) = 0, & & \\
\text{inequality constraints} & g_j(x_j, u_j) \leq 0, & j = 0, \dots, N-1, & \\
\text{terminal inequality constraint} & r^{\text{in}}(x_N) \leq 0 & &
\end{array}$$

In this formulation, $\ell : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ represents the stage cost function, $\omega_j, j = 1, \dots, N$, are the weights of the stage cost function and F is the terminal cost function. Notice that we present here a more general objective function (compare with (1.4)) due to the included weights and terminal costs which under certain assumptions guarantee stability (see discussion in [36, Chapter 5 and 7]). The case (1.4) is obtained with $\omega_j = 1, j = 1, \dots, N$, and $F \equiv 0$.

For structural advantages that we will see later on, we arrange the optimization variables into a vector

$$z := \left[x_0^\top, u_0^\top, x_1^\top, u_1^\top, \dots, x_{N-1}^\top, u_{N-1}^\top, x_N^\top, u_N^\top\right]^\top \tag{6.2}$$

and define

$$z_j := \left[x_j^\top, u_j^\top\right]^\top$$

where u_N is an auxiliary variable to complete the notation (as in [18], $u_N := u_{N-1}$ is introduced for notational convenience and does not affect the derivatives we will later need due to linearity). With the parameter p, the OCP $\mathcal{P}_N(p)$ in (6.1) is of the form $P(p)$ defined in Section 5.5, i.e., a parametric NLP – a minimization problem of an objective function subject to equality and inequality constraints – that depends on the parameter p which is, in this case, assigned to be the current measured state of the OCP.

6.1.2 Resulting matrix structures

The Lagrangian function $\mathcal{L}$ for (6.1) is given by

$$\begin{aligned}
\mathcal{L}(z, \eta, p) \quad = \quad & \sum_{j=0}^{N-1} \omega_{N-j} \ell(x_j, u_j) + F(x_N) + \lambda^a (x_0 - p) \\
& + \sum_{j=0}^{N-1} \lambda_{j+1}^b (x_{j+1} - f(x_j, u_j)) + \sum_{j=0}^{N-1} \lambda_j^c h_j(x_j, u_j) + \lambda^d r^{\text{eq}}(x_N) \\
& + \sum_{j=0}^{N-1} \mu_j^e g_j(x_j, u_j) + \mu^f r^{\text{in}}(x_N))
\end{aligned}$$

where $\eta = [\lambda^\top, \mu^\top]^\top$, $\lambda = [\lambda^{a\top}, \lambda^{b\top}, \lambda^{c\top}, \lambda^{d\top}]^\top$ and $\mu = [\mu^{e\top}, \mu^{f\top}]^\top$. Observe that the function $\mathcal{L}$ can be decomposed into subfunctions that each depend on particular multipliers and only on the variable z_j, i.e., the pair (x_j, u_j). We obtain

$$\mathcal{L}(z, \eta, p) = \sum_{j=0}^{N} \mathcal{L}_j(z_j, \eta, p) \tag{6.3}$$

where

$$\begin{aligned}
\mathcal{L}_0(z_0, \eta, p) &= \omega_N \ell(x_0, u_0) + \lambda^a(x_0 - p) - \lambda_1^b f(x_0, u_0) \\
&\quad + \lambda_0^c h_0(x_0, u_0) + \mu_0^e g_0(x_0, u_0) \\
\mathcal{L}_j(z_j, \eta, p) &= \omega_{N-j} \ell(x_j, u_j) + \lambda_j^b\, x_j - \lambda_{j+1}^b f(x_j, u_j) \\
&\quad + \lambda_j^c h_j(x_j, u_j) + \mu_j^e g_j(x_j, u_j), \qquad j = 1, \ldots, N-1, \\
\mathcal{L}_N(z_N, \eta, p) &= F(x_N) + \lambda_N^b x_N + \lambda^d r^{\mathrm{eq}}(x_N) + \mu^f r^{\mathrm{in}}(x_N)
\end{aligned}$$

Such a property is called **partial separability** of the Lagrangian under which

$$\nabla_{z_i} \mathcal{L}(z, \eta, p) = \nabla_{z_i} \sum_{j=0}^{N} \mathcal{L}_j(z_j, \eta, p) = \nabla_{z_i} \mathcal{L}_i(z_i, \eta, p), \quad i = 0, \ldots, N,$$

holds, making the gradient of the Lagrangian

$$\nabla \mathcal{L}(z, \eta, p) = \begin{bmatrix} \nabla_{z_0} \mathcal{L}(z, \eta, p) \\ \nabla_{z_1} \mathcal{L}(z, \eta, p) \\ \vdots \\ \nabla_{z_N} \mathcal{L}(z, \eta, p) \end{bmatrix} = \begin{bmatrix} \nabla_{z_0} \mathcal{L}_0(z_0, \eta, p) \\ \nabla_{z_1} \mathcal{L}_1(z_1, \eta, p) \\ \vdots \\ \nabla_{z_N} \mathcal{L}_N(z_N, \eta, p) \end{bmatrix}$$

and the Hessian of the Lagrangian

$$\nabla^2 \mathcal{L} = \begin{bmatrix} \dfrac{\partial^2 \mathcal{L}}{\partial z_0^2} & \dfrac{\partial^2 \mathcal{L}}{\partial z_0\, \partial z_1} & \cdots & \dfrac{\partial^2 \mathcal{L}}{\partial z_0\, \partial z_N} \\ \dfrac{\partial^2 \mathcal{L}}{\partial z_1\, \partial z_0} & \dfrac{\partial^2 \mathcal{L}}{\partial z_1^2} & \cdots & \dfrac{\partial^2 \mathcal{L}}{\partial z_1\, \partial z_N} \\ \vdots & \vdots & \ddots & \vdots \\ \dfrac{\partial^2 \mathcal{L}}{\partial z_N\, \partial z_0} & \dfrac{\partial^2 \mathcal{L}}{\partial z_N\, \partial z_1} & \cdots & \dfrac{\partial^2 \mathcal{L}}{\partial z_N^2} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial^2 \mathcal{L}}{\partial z_1^2} & & & 0 \\ & \dfrac{\partial^2 \mathcal{L}}{\partial z_2^2} & & \\ & & \ddots & \\ 0 & & & \dfrac{\partial^2 \mathcal{L}}{\partial z_n^2} \end{bmatrix}$$

since all derivatives $\dfrac{\partial^2 \mathcal{L}}{\partial z_i\, \partial z_j}(z, \eta, p) = 0$ for any $i \neq j$. In addition, $\dfrac{\partial^2 \mathcal{L}}{\partial z_i^2}(z, \eta, p) = \dfrac{\partial^2 \mathcal{L}_i}{\partial z_i^2}(z_i, \eta, p)$.

6.1.3 Solving $\mathcal{P}_N(p)$ by the active-set SQP strategy

We solve (6.1) which is in the form (5.6) by the active-set SQP strategy. Using this approach, we iteratively solve a sequence of QPs until convergence. At current iterate (z^k, η^k) i.e., in one iteration of an SQP method, a QP needs to be solved. This QP can be written in the form of (5.24). For simplicity, we drop the iteration index k on the variables $z, \eta, \Delta z$. We have

$$\min_{\Delta z_0,\ldots,\Delta z_N} \quad \frac{1}{2}\,\Delta z^\top \nabla^2 \mathcal{L}(z,\eta,p)\Delta z \;+\; \nabla\left(\sum_{j=0}^{N-1} \omega_{N-j}\ell\left(x_j,u_j\right) + F\left(x_N\right)\right)^\top \Delta z$$

subject to

$$\begin{aligned}
x_0 - p + \Delta x_0 &= 0, \\
x_{j+1} - f(x_j,u_j) + \Delta x_{j+1} - \nabla_{z_j} f(x_j,u_j)^\top \Delta z_j &= 0, \quad j = 0,\ldots,N-1, \\
h(x_j,u_j) + \nabla_{z_j} h(x_j,u_j)^\top \Delta z_j &= 0, \quad j = 0,\ldots,N, \\
r^{\mathrm{eq}}(x_N) + \nabla_{z_N} r^{\mathrm{eq}}(x_N)^\top \Delta z_N &= 0, \\
g(x_j,u_j) + \nabla_{z_j} g(x_j,u_j)^\top \Delta z_j &\le 0, \quad j = 0,\ldots,N, \\
r^{\mathrm{in}}(x_N) + \nabla_{z_N} r^{\mathrm{in}}(x_N)^\top \Delta z_N &\le 0, \\
\Delta u_N - \Delta u_{N-1} &= 0
\end{aligned}$$

The solution may be obtained by solving the corresponding system in the form of (5.25), i.e.,

$$\begin{bmatrix} \nabla^2\mathcal{L}(z,\eta,p) & \nabla C_{\mathcal{A}}(z,p) \\ \nabla C_{\mathcal{A}}(z,p)^\top & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \eta_{\mathcal{A}} \end{bmatrix} = -\begin{bmatrix} \nabla f(z,p) \\ C_{\mathcal{A}}(z,p) \end{bmatrix} \tag{6.4}$$

where the submatrices of the coefficient matrix are constructed using

$$\nabla^2\mathcal{L}(z,\eta,p) = \begin{bmatrix} \frac{\partial^2 \mathcal{L}}{\partial z_0^2}(z_0,\eta,p) & & & 0 \\ & \frac{\partial^2 \mathcal{L}}{\partial z_1^2}(z_1,\eta,p) & & \\ & & \ddots & \\ 0 & & & \frac{\partial^2 \mathcal{L}}{\partial z_N^2}(z_N,\eta,p) \end{bmatrix} \tag{6.5}$$

$$\nabla C(z,p)^\top = \begin{bmatrix} I_x & & & & \\ \Psi_0 & I_x & & & \\ & & \ddots & & \\ & & & \Psi_{N-1} & I_x \\ \Phi_0 & & & & \\ & & \ddots & & \\ & & & \Phi_{N-1} & \\ & & & & \Lambda \\ \Theta_0 & & & & \\ & & \ddots & & \\ & & & \Theta_{N-1} & \\ & & & & \Upsilon \\ & & & I_u & -I_u \end{bmatrix} \tag{6.6}$$

where $\Psi_j = -\nabla_{z_j} f(x_j,u_j)^\top$, $\Phi_j = \nabla_{z_j} h(x_j,u_j)^\top$, $\Theta_j = \nabla_{z_j} g(x_j,u_j)^\top$, $j = 0,\dots,N-1$, $\Lambda = \nabla_{z_N} r^{\mathrm{eq}}(z_N)^\top$, $\Upsilon = \nabla_{z_N} r^{\mathrm{in}}(z_N)^\top$, $I_x = [I\ 0]$, $I_u = [0\ I]$. The right-hand side of (5.25) (or (6.4)) is composed of

$$\nabla\left(\sum_{j=0}^{N-1} \omega_{N-j}\ell\left(x_j,u_j\right) + F\left(x_N\right)\right) = \begin{bmatrix} \nabla_{z_0}\omega_N\ell\left(x_0,u_0\right) \\ \nabla_{z_1}\omega_{N-1}\ell\left(x_1,u_1\right) \\ \vdots \\ \nabla_{z_{N-1}}\omega_1\ell\left(x_{N-1},u_{N-1}\right) \\ \nabla_{z_N}F\left(x_N\right) \end{bmatrix}$$

and $C_{\mathcal{A}}(z)$ obtained from

$$C(z) = \begin{bmatrix} x_0 - p \\ x_1 - f\left(x_0,u_0\right) \\ \vdots \\ x_N - f\left(x_{N-1},u_{N-1}\right) \\ h(x_0,u_0) \\ \vdots \\ h(x_{N-1},u_{N-1}) \\ r^{\mathrm{eq}}(x_N) \\ g(x_0,u_0) \\ \vdots \\ g(x_{N-1},u_{N-1}) \\ r^{\mathrm{in}}(x_N) \\ u_N - u_{N-1} \end{bmatrix}$$

We then obtain Δz (actually, Δz^k since the superscripts were dropped). This allows for the iterative update $z^{k+1} = z^k + \Delta z^k$. We update until convergence. From Remark 5.5.3, the resulting coefficient matrix of the system (6.4) coincides with the KKT matrix needed to compute the sensitivity differentials with respect to p for the problem $\mathcal{P}_N(p)$.

6.1.4 Incorporating sensitivity updates to the m-step MPC algorithm

In the nominal setting, by performing the re-optimization (as discussed in Algorithm 1.3.2) on a shrunken horizon using the current state of the system as the initial value, we recover as a solution a tail of the optimal solution obtained from full horizon optimization. This is due to the fact that at the current time instant, the current measured state coincides with the predicted state generated by the full horizon optimal control.

In the perturbed setting, using the updated m-step MPC, the current measured state that we use as the initial value in the re-optimization on a shrunken horizon can be viewed as a perturbation of the predicted value that would have been the initial value had there been no perturbations.

The setting allows for an alternative to re-optimization through the use of sensitivity analysis. This enables the approximation of the solution of the updated multistep MPC and the avoidance of solving all optimization problems on shrunken horizons and hence reducing computational cost. This gives us an

MPC variant which we refer to as **sensitivity-based m-step (SBM) MPC** for which the only optimizations performed are full-horizon optimizations done only every m steps.

First, we make the following observations.

Suppose x_0^{m} is the current measured state. Consider $\mathcal{P}_N(p)$ defined by (6.1) and let the parameter p take the value x_0^{m}, i.e., solve $\mathcal{P}_N(x_0^{\mathrm{m}})$. Let $x_0^*, \ldots, x_N^*$ be the *nominal* optimal trajectory and $u_0^*, \ldots, u_{N-1}^*$ be the *nominal* optimal control sequence.

Due to the structure of $\mathcal{P}_N(p)$ (that yields properties such as separability of the Lagrangian), we can easily construct the following problem $\mathcal{P}_{N-j}(p_j)$ by discarding terms with the variables $(x_0, u_0), \ldots, (x_{j-1}, u_{j-1})$ in the objective function and the constraints and shortening the horizon to $N - j$. Consider

$$
\begin{array}{lll}
\min\limits_{\substack{x_i, i=j,\ldots,N \\ u_i, i=j,\ldots,N-1}} & \sum\limits_{i=j}^{N-1} \omega_{N-i} \ell\left(x_i, u_i\right) \quad + F\left(x_N\right) & \\
\text{subject to} & x_j = p_j & \\
& x_{i+1} = f\left(x_i, u_i\right) \qquad i = j, \ldots, N-1, & \\
& h_i(x_i, u_i) = 0 \qquad i = j, \ldots, N-1, & \\
& r^{\mathrm{eq}}(x_N) = 0 & \\
& g_i(x_i, u_i) \le 0 \qquad i = j, \ldots, N-1, & \\
& r^{\mathrm{in}}(x_N) \le 0 &
\end{array}
\qquad \mathcal{P}_{N-j}(p_j)
$$

for all $j = 0, \ldots, N-1$. We particularly indexed the parameter p_j with j to indicate that it is the parameter for $\mathcal{P}_{N-j}(p_j)$. In addition, note that in the setup x_j is now the first element of the trajectory and u_j is now the first element of the control sequence.

Remark 6.1.1. From $\mathcal{P}_N(x_0^{\mathrm{m}})$ (equivalently, $\mathcal{P}_N(x_0^*)$), note that the tails $u_j^*, \ldots, u_{N-1}^*$ form the optimal control sequence for $\mathcal{P}_{N-j}(x_j^*)$ for all $j = 1, \ldots, N-1$.

Consider the optimization variable z defined in (6.2). In reference to that, define the tails

$$
z^j := \left[x_j^\top, u_j^\top, \ldots, x_{N-1}^\top, u_{N-1}^\top, x_N^\top, u_N^\top\right]^\top
$$

Define z^{j^*} accordingly. Let $\mathcal{L}^j(z^j, \eta, p_j)$ denote the corresponding Lagrangian and $\mathcal{A}^j$ the corresponding active set of $\mathcal{P}_{N-j}(p_j)$.

Note that $\mathcal{P}_{N-j}(p_j)$ can be written in the form $P(p_j)$ to clearly identify the objective, equality and inequality constraint left hand side functions (denoted as f, g and h, respectively, for $P(p)$ defined in Section 5.5). To be able to incorporate sensitivities in the discussion, we make the following assumption.

Assumption 6.1.2. For $j = 0, \ldots, N-1$, the objective, equality and inequality constraint left hand side functions of $\mathcal{P}_{N-j}(p_j)$ written in the form $P(p_j)$ are twice continuously differentiable in a neighborhood of the solution z^{j^*} and SOSC, LICQ and strict complementarity hold at z^{j^*}.

This assumption is precisely the assumption of the sensitivity theorem, Theorem 5.5.1, allowing the existence of a neighborhood $\mathcal{N}(x^{j^*})$ for all $j = 0, \ldots, N-1$, where the required sensitivities are defined.

Let x_j^{m} be the measured state. Let the parameter p_j take this value. Consider $\mathcal{P}_{N-j}(x_j^{\mathrm{m}})$ and denote the resulting optimal control sequence as $u_{j,0}^*, \ldots, u_{j,N-j-1}^*$.

Remark 6.1.3. Following Remark 5.5.3, for $j = 1, \ldots, N-1$, the already available information u_j^* from the nominal solution of the problem $\mathcal{P}_{N-j}(x_j^*)$ and the sensitivity differentials $\dfrac{\partial u_j}{\partial p_j}(x_j^*)$ provides $u_{j,0}^*$, i.e., the first element of the optimal control sequence of the *perturbed* problem $\mathcal{P}_{N-j}(x_j^{\mathrm{m}})$ through

$$u_{j,0}^* = u_j^* + \frac{\partial u_j}{\partial p_j}(x_j^*)(x_j^{\mathrm{m}} - x_j^*) + \mathcal{O}\left(\|x_j^{\mathrm{m}} - x_j^*\|^2\right), \quad j = 0, \ldots, m-1 \tag{6.7}$$

We are now in the position to provide the SBM MPC algorithm.

Algorithm 6.1.4. (SBM MPC)

Assume that for the initial time instant k, k is a multiple of m.

(1) measure the state $x(k) \in \mathbb{X}$ of the system at time instant k

(2) set $j := k - \lfloor k \rfloor_m,\ x_j^m := x(k)$.

- If $j = 0$, solve $\mathcal{P}_N(x_0^m)$. Store $u_0^*, \ldots, u_{N-1}^*$ and $x_0^*, \ldots, x_N^*$.
- Define the time-dependent MPC feedback

$$\overline{\mu}_{N,m}(x(k), k) := u_j^* + \frac{\partial u_j}{\partial p_j}(x_j^*)(x_j^{\mathrm{m}} - x_j^*) \tag{6.8}$$

(3) apply the control values $\overline{\mu}_{N,m}(x(k), k)$ to the system, set $k := k+1$ and go to (1)

Note that at $j = 0$, $x_0^{\mathrm{m}} = x_0^*$, thus the corrective term $\frac{\partial u_j}{\partial p_j}(x_j^*)(x_j^{\mathrm{m}} - x_j^*)$ vanishes, i.e., no update is performed during the first iteration.

Remark 6.1.5. From Remark 6.1.3 and the approximation (6.7), the feedback $\overline{\mu}_{N,m}(x(k), k)$ defined in (6.8) is a first-order approximation of $\hat{\mu}_{N,m}(x(k), k)$ defined in (1.9) having an error with order of magnitude of at most $\|x_j^{\mathrm{m}} - x_j^*\|^2$. A detailed analysis on the implications of this is given in Section 6.3.

To summarize, in using Algorithm 6.1.4 we first apply the obtained u_0^* and then we apply corrections on $u_1^*, u_2^*, \ldots, u_{m-1}^*$. Hence, at time instants $1, 2, \ldots, m-1$, instead of optimizing again (i.e., using SQP active-set strategy) as in the standard MPC, or instead of re-optimizing using shrinking horizons as in the updated m-step MPC, in the hopes of reducing the operation costs, we compute the sensitivities

$$\frac{\partial u_1}{\partial p_1}(x_1^*), \frac{\partial u_2}{\partial p_2}(x_2^*), \ldots, \frac{\partial u_{m-1}}{\partial p_{m-1}}(x_{m-1}^*)$$

from appropriate linear systems as detailed in the subsequent subsection and use them as corrective updates.

6.1.5 Computing sensitivities and exploiting matrix structures

To illustrate, let us compute first $\dfrac{\partial u_1}{\partial p_1}(x_1^*)$. In words, this is the sensitivity of the first element of the tail $u_1 \ldots, u_{N-1}, u_N$ with respect to the parameter p_1

of the shortened problem $\mathcal{P}_{N-1}(p_1)$ evaluated at the predicted state x_1^*. The sensitivity $\dfrac{\partial u_1}{\partial p_1}(x_1^*)$ is obtained from

$$\frac{\partial z^1}{\partial p_1}(x_1^*) = \left[\frac{\partial x_1}{\partial p_1}^\top, \frac{\partial u_1}{\partial p_1}^\top, \ldots, \frac{\partial x_N}{\partial p_1}^\top, \frac{\partial u_N}{\partial p_1}^\top \right]^\top (x_1^*)$$

which is solved using (5.40) via

$$\begin{aligned} &\begin{bmatrix} \nabla^2_{z^1 z^1} \mathcal{L}^1({z^1}^*, \eta^*, {x_1}^*) & \nabla_{z^1} C_{\mathcal{A}^1}({z^1}^*, {x_1}^*) \\ \nabla_{z^1} C_{\mathcal{A}^1}({z^1}^*, {x_1}^*)^\top & 0 \end{bmatrix} \begin{bmatrix} \dfrac{\partial z^1}{\partial p_1}(x_1^*) \\ \dfrac{\partial \eta_{\mathcal{A}^1}}{\partial p_1}(x_1^*) \end{bmatrix} \\ &\qquad = - \begin{bmatrix} \nabla^2_{z^1 p_1} \mathcal{L}^1({z^1}^*, \eta^*, {x_1}^*)^\top \\ \nabla_{p_1} C_{\mathcal{A}^1}({z^1}^*, {x_1}^*)^\top \end{bmatrix} \end{aligned} \tag{6.9}$$

The same applies for the sensitivities $\dfrac{\partial u_2}{\partial p_2}(x_2^*), \ldots, \dfrac{\partial u_{m-1}}{\partial p_{m-1}}(x_{m-1}^*)$, i.e., we need to construct and solve the corresponding system analogous to (6.9) to solve sensitivities $\dfrac{\partial z^j}{\partial p_j}(x_j^*)$, for all $j = 2, \ldots, m-1$ from which we obtain $\dfrac{\partial u_j}{\partial p_j}(x_j^*), j = 2, \ldots, m-1$. Therefore, to generalize, computing the updating or correcting sensitivities requires solving the sequence of linear systems

$$\begin{aligned} &\begin{bmatrix} \nabla^2_{z^j z^j} \mathcal{L}^j({z^j}^*, \eta^*, {x_j}^*) & \nabla_{z^j} C_{\mathcal{A}^j}({z^j}^*, {x_j}^*) \\ \nabla_{z^j} C_{\mathcal{A}^j}({z^j}^*, {x_j}^*)^\top & 0 \end{bmatrix} \begin{bmatrix} \dfrac{\partial z^j}{\partial p_j}(x_j^*) \\ \dfrac{\partial \eta_{\mathcal{A}^j}}{\partial p_j}(x_j^*) \end{bmatrix} \\ &\qquad = - \begin{bmatrix} \nabla^2_{z^j p_j} \mathcal{L}^j({z^j}^*, \eta^*, {x_j}^*)^\top \\ \nabla_{p_j} C_{\mathcal{A}^j}({z^j}^*, {x_j}^*)^\top \end{bmatrix} \end{aligned} \tag{6.10}$$

for $j = 1, \ldots, m-1$ corresponding to OCPs $\mathcal{P}_{N-j}(p_j)$ of decreasing horizons and adjusting parametric values.

Now, as pointed out in Remark 5.5.3, the coefficient matrix of the QP system (5.25) coincides with the KKT matrix in (5.40).

Consider the Hessian $\nabla^2_{z^j z^j} \mathcal{L}^j({z^j}^*, \eta^*, {x_j}^*)$ of the Lagrangian function for the problem $\mathcal{P}_{N-j}(p_j)$ evaluated at $p_j = x_j^*$. It has the same form but is smaller in size as the Hessian $\nabla^2_{zz} \mathcal{L}(z^*, \eta^*, {x_0}^*)$ for $\mathcal{P}_N(p_0)$ evaluated at $p_0 = x_0^{\mathrm{m}}$. It can be obtained from $\nabla^2_{zz} \mathcal{L}(z^*, \eta^*, {x_0}^*)$ (as in (6.5)) by discarding blocks of indices 0 to $j-1$ leaving $N+1-j$ blocks along the diagonal.

The Jacobian $\nabla_{z^j} C_{\mathcal{A}^j}({z^j}^*, {x_j}^*)^\top$ of the active constraints corresponding to $\mathcal{A}^j$ are obtained from (6.6) by also discarding blocks of indices 0 to $j-1$.

This means that the KKT matrix of the sensitivity system corresponding to the OCP $\mathcal{P}_{N-j}(p_j)$ with $p_j = x_j^*$ can be constructed through the submatrices of the coefficient matrix of the QP system obtained for $\mathcal{P}_N(p_0)$ with $p_o = x_0^{\mathrm{m}}$, i.e., from information that is already available.

What remains is to construct the right-hand side of the sensitivity system as in the right-hand side of (5.40). The parameter p_j only appears in $\mathcal{P}_{N-j}(p_j)$ in the equality constraints. Hence, the second derivatives $\nabla^2_{z^j p_j} \mathcal{L}^j (z^{j*}, \eta^*, x_j{}^*)^\top$ of the Lagrangian function is 0 since p_j enters the Lagrangian $\mathcal{L}^j$ of $\mathcal{P}_{N-j}(p_j)$ linearly through the equality constraint in $x_j - p_j$.

For $\nabla_{p_j} C_{\mathcal{A}^j}(z^{j*}, x_j{}^*)^\top$, we obtain a zero matrix except for the $-I$ corresponding to $x_j - p_j$.

In addition, the systems (6.10) are closely related to each other as the succeeding systems differ from the previous ones by deleting rows and columns. In particular, in solving the linear systems (6.10) by factorization, one does not need to factorize each coefficient matrix from scratch. Instead, the matrix factorization computed at $\mathcal{P}_N(x_0)$ is modified according to minor changes caused by deletion of rows and columns. Exploiting matrix structures is discussed in great details in [67], [29] and [30].

6.2 Changes in active constraints set

In applying the update (6.8), caution is necessary so that the updates due to the perturbed parameter do not change the set of active constraints as to not violate the assertion of the sensitivity theorem, Theorem 5.5.1. We discuss in this section some of the existing ideas in the literature, i.e., as presented in [17] and [6] addressing this issue.

First, we consider an approach given in [17] to determine for which values of p is the perturbation $p - p_0$ too large to render (5.41) a good approximation. Based on (5.41),

$$\eta_{\mathcal{A}}(p) \approx \eta^*_{\mathcal{A}} + \frac{\partial \eta_{\mathcal{A}}}{\partial p}(p_0)\,(p - p_0) \tag{6.11}$$

where the sensitivity $\dfrac{\partial \eta_{\mathcal{A}}}{\partial p}(p_0)$ is obtained by solving (5.40). A constraint will leave the active set when its corresponding Lagrange multiplier goes to zero. If one of the multipliers is close to zero, using (6.11) we obtain

$$0 = \eta_{\mathcal{A},i}(p) \approx \eta^*_{\mathcal{A},i} + \frac{\partial \eta_{\mathcal{A},i}}{\partial p}(p_0)\,(p - p_0) \tag{6.12}$$

The relation (6.12) allows for an approximation of the perturbed parameter $p^i = (p^i_1, \ldots, p^i_n)^\top$ that causes a constraint C_i to leave the active set as given by

$$p^i_j \approx p_{0,j} - \frac{\eta^*_{\mathcal{A},i}}{\frac{\partial \eta_{\mathcal{A},i}}{\partial p_j}(p_0)},\; i \in \mathcal{A},\; j \in \{1, \ldots, n\} \tag{6.13}$$

assuming $\frac{\partial \eta_{\mathcal{A},i}}{\partial p_j}(p_0) \neq 0$. Similarly, a constraint will enter the active set when its value goes to zero, i.e., a constraint $C_i, i \notin \mathcal{A}$, becomes zero if

$$0 = C_i(z(p), p) \approx C_i(z^*, p_0) + \frac{\partial C_i}{\partial p}(z^*, p_0)\,(p - p_0)$$

This yields an approximation of $p^i = (p^i_1, \ldots, p^i_n)^\top$ causing a constraint C_i to enter the active set as given by

$$p_j^i \approx p_{0,j} - \frac{C_i(z^*, p_0)}{\frac{\partial C_i}{\partial p_j}(z^*, p_0)}, \; i \notin \mathcal{A}, \; j \in \{1, \ldots, n\} \tag{6.14}$$

provided that $\frac{\partial C_i}{\partial p_j}(z^*, p_0) \neq 0$. The *sensitivity domain* P_0, which gives the range of perturbed parameter on which formula (6.8) is suitable to be applied, is determined by the values p_j^i in (6.13) and (6.14) closest to the nominal parameter $p_{0,j}$

$$\begin{aligned} P_0 &\approx P_0^1 \times P_0^2 \times \ldots \times P_0^{N_c} \\ P_0^j &:= \left[\max_{\overline{p}_j < p_{0,j}} \{\overline{p}_j \in \overline{P}_j\}, \min_{\overline{p}_j > p_{0,j}} \{\overline{p}_j \in \overline{P}_j\} \right], j = 1, \ldots, n, \\ \overline{P}_j &:= \{p_j^i \mid i = 1, \ldots, N_c\} \cup \{-\infty, +\infty\} \end{aligned}$$

After computing P_0, an alternative approximation to (5.41) proposed in [17] based on [6] may then be obtained. Suppose $\tilde{p}$ denotes a perturbed parameter that causes a constraint to enter or leave the active set. The new first-order approximation is given by

$$z(p) \approx z_0 + \Delta z, \quad \eta^u(p) \approx \eta^u + \Delta\eta^u, \quad \text{if } p - \tilde{p} \geq 0$$

where

$$\Delta z := \frac{\partial z}{\partial p}(p_0)(\tilde{p} - p_0) + \frac{\partial z}{\partial p}(\tilde{p})(p - \tilde{p}_0) \tag{6.15}$$

$$\Delta\eta^u := \frac{\partial \eta^u}{\partial p}(p_0)(\tilde{p} - p_0) + \frac{\partial \eta^u}{\partial p}(\tilde{p})(p - \tilde{p}_0) \tag{6.16}$$

and η^u is the updated Lagrange multiplier reflected by the change in active constraints.

In (6.15)–(6.16), observe the need for the sensitivities $\frac{\partial z}{\partial p}(\tilde{p})$ and $\frac{\partial \eta^u}{\partial p}(\tilde{p})$ which are not directly obtainable from the information available after solving $\mathcal{P}_N(p_0)$ as in the case when solving systems (6.10). We recall, however, in Remark 5.5.3(d) that the sensitivity differentials can also be computed by a so-called post-optimal analysis (as detailed in [17]) which is an approach typically used when the KKT matrix for computing sensitivities does not coincide to the coefficient matrix of the SQP system upon convergence (e.g., when approximations of the Hessian matrices are used).

In our implementation, since the motivation is to analyze the reduction of cost by taking advantage of information that are available through the SBM MPC strategy, the post optimal analysis is not applied. A much simpler rule is used as to not to violate the constraints when updating by sensitivities. We mention this in Section 7.2.

6.3 Stability and performance analysis of SBM MPC

The main motivation for considering sensitivity-based control is to have a less costly alternative to re-optimization. Now the aim of the section is to investigate how well the SBM MPC approximates the updated m-step MPC in terms of stability and suboptimality performance.

To meaningfully include sensitivity updates to the MPC discussion, we need to assume Assumption 6.1.2. Aside from guaranteeing the existence of the desired sensitivities, the assumption also implies that the active sets remain constant on neighborhoods where updates are applied. This simplifies the exposition.

To this end, we first consider the finite horizon setting as in Chapter 3. Let x_j^{m} be the measured state at time instant j and consider $\mathcal{P}_{N-j}(x_j^{\mathrm{m}})$. We denote the resulting optimal control sequence as

$$u_{j,\cdot}^* := \{u_{j,0}^*, u_{j,1}^*, \ldots, u_{j,N-j-1}^*\}$$

For time instants $j = 0, \ldots, N-1$, let the sequence

$$\overline{u}_{j,\cdot}^* := \{\overline{u}_{j,0}^*, \overline{u}_{j,1}^*, \ldots, \overline{u}_{j,N-j-1}^*\}$$

indicate the sensitivity-based approximation of the sequence $u_{j,\cdot}^*$.

As discussed in Remark 5.5.3(b) the optimal solution of a perturbed problem is given by

$$\begin{aligned} u_{j,k}^* \quad &= \quad u_{0,j+k}^* + \frac{\partial u_{j+k}}{\partial p_j}(x_{u_{0,\cdot}^*}(j,x_0))(x_j^{\mathrm{m}} - x_{u_{0,\cdot}^*}(j,x_0)) \\ &+ \ \mathcal{O}(\|x_j^{\mathrm{m}} - x_{u_{0,\cdot}^*}(j,x_0)\|^2), \qquad k = 0, \ldots, N-j-1 \end{aligned}$$

from which we obtain a sensitivity-based approximation (compare to (6.7) and (6.8)) given by the definition

$$\overline{u}_{j,k}^* := u_{0,j+k}^* + \frac{\partial u_{j+k}}{\partial p_j}(x_{u_{0,\cdot}^*}(j,x_0))(x_j^{\mathrm{m}} - x_{u_{0,\cdot}^*}(j,x_0)), \ k = 0, \ldots, N-j-1$$

with

$$u_{j,k}^* = \overline{u}_{j,k}^* + \mathcal{O}(\|x_j^{\mathrm{m}} - x_{u_{0,\cdot}^*}(j,x_0)\|^2) \tag{6.17}$$

In this definition, all quantities except x_j^{m} are computed at time $j = 0$ from $\mathcal{P}_N(x_0)$ with optimal control $u_{0,j}^*$, $j = 0, \ldots, N-1$, and $x_{u_{0,\cdot}^*}(j,x_0)$, $j = 0, \ldots, N-1$.

Next we define the cost

$$\overline{\lambda}_j = \ell\left(x_j^{\mathrm{m}}, \overline{u}_{j,0}^*\right)$$

which denotes the cost incurred at time j from the measured state x_j^{m} by applying the sensitivity-based control $\overline{u}_{j,0}^*$.

Similar to $J_N^{\mathrm{nmult}}(x_0)$, $J_N^{\mathrm{pmult}}(x_0)$ and $J_N^{\mathrm{upd}}(x_0)$ defined in Definition 3.3.1, we assign the quantity

$$J_N^{\mathrm{sens}}(x_0) := \sum_{j=0}^{N-1} \overline{\lambda}_j = \sum_{j=0}^{N-1} \ell(x_j^{\mathrm{m}}, \overline{u}_{j,0}^*) \tag{6.18}$$

to denote the value of the perturbed trajectory driven by the sensitivity-based control we will illustrate shortly.

We first make the following assumption allowing for error estimates on generated states and stage costs resulting from approximate optimal controls.

Assumption 6.3.1. The functions f and ℓ satisfy the Lipschitz condition both on x and on u, i.e., for all compact sets $K_1 \in \mathbb{X}, K_2 \in \mathbb{U}$, there exist $C_{11}, C_{12} > 0$ such that

$$\begin{array}{rcl} \|f(x,u) - f(\tilde{x},u)\| & \leq & C_{11}\|x - \tilde{x}\| \\ \|\ell(x,u) - \ell(\tilde{x},u)\| & \leq & C_{12}\|x - \tilde{x}\| \end{array}$$

for all $x, \tilde{x} \in K_1, u \in K_2$, and there exist $C_{21}, C_{22} > 0$ such that

$$\begin{array}{rcl} \|f(x,u) - f(x,\tilde{u})\| & \leq & C_{21}\|u - \tilde{u}\| \\ \|\ell(x,u) - \ell(x,\tilde{u})\| & \leq & C_{22}\|u - \tilde{u}\| \end{array}$$

for all $u, \tilde{u} \in K_2, x \in K_1$.

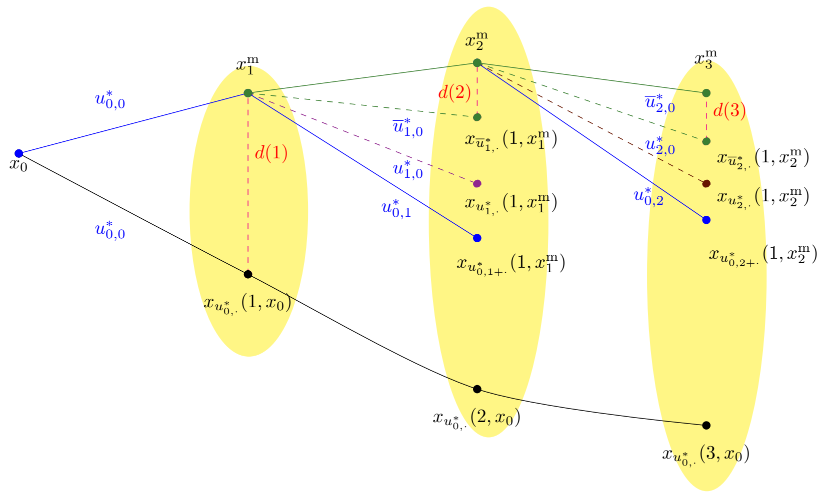

Figure 6.1: Resulting trajectories from approximately re-optimizing (through sensitivity-based updates) and the accumulating errors.

We illustrate the trajectories in Figure 6.1. At time $j = 0$, by optimization we obtain an optimal control sequence whose first element is $u^*_{0,0}$. The predicted state is $x_{u^*_{0,\cdot}}(1, x_0) = f(x_0, u^*_{0,0})$ but due to perturbation $d(1)$, the resulting measured state is x^{m}_1.

Consider time $j = 1$ and the measured state x^{m}_1. There are three ways we can proceed, namely, by applying the nominal open-loop control, by re-optimization or by a sensitivity-based update. By applying the nominal open-loop control element $u^*_{0,1}$, the predicted state will be $x_{u^*_{0,1+\cdot}}(1, x^{\mathrm{m}}_1) = f(x^{\mathrm{m}}_1, u^*_{0,1})$. Alternatively, we can also perform re-optimization to obtain and apply the control value $u^*_{1,0}$ resulting in the predicted state $x_{u^*_{1,\cdot}}(1, x^{\mathrm{m}}_1) = f(x^{\mathrm{m}}_1, u^*_{1,0})$. Lastly, we can apply the approximate control $\overline{u}^*_{1,0}$ given by

$$\overline{u}^*_{1,0} = u^*_{0,1} + \frac{\partial u_1}{\partial p_1}(x_{u^*_{0,\cdot}}(1, x_0))\underbrace{(x^{\mathrm{m}}_1 - x_{u^*_{0,\cdot}}(1, x_0))}_{d(1)} \tag{6.19}$$

where we have the relation

$$\begin{aligned} u^*_{1,0} &= \overline{u}^*_{1,0} + \mathcal{O}(\|x^{\mathrm{m}}_1 - x_{u^*_{0,\cdot}}(1, x_0)\|^2) \\ &= \overline{u}^*_{1,0} + \mathcal{O}(d_1^2) \ \text{ with } d_1 = \|d(1)\| \end{aligned} \tag{6.20}$$

The predicted state when $\overline{u}^*_{1,0}$ is applied will then be $x_{\overline{u}^*_{1,\cdot}}(1, x^{\mathrm{m}}_1) = f(x^{\mathrm{m}}_1, \overline{u}^*_{1,0})$. Observe that by Assumption 6.3.1,

$$\begin{aligned} \left\|x_{u^*_{1,\cdot}}(1, x^{\mathrm{m}}_1) - x_{\overline{u}^*_{1,\cdot}}(1, x^{\mathrm{m}}_1)\right\| &= \left\|f(x^{\mathrm{m}}_1, u^*_{1,0}) - f(x^{\mathrm{m}}_1, \overline{u}^*_{1,0})\right\| \\ &\leq C_1\|u^*_{1,0} - \overline{u}^*_{1,0}\| \end{aligned}$$

for some $C_1 > 0$, and by (6.20) we obtain

$$x_{u^*_{1,\cdot}}(1, x^{\mathrm{m}}_1) = x_{\overline{u}^*_{1,\cdot}}(1, x^{\mathrm{m}}_1) + \mathcal{O}(d_1^2)$$

Due to an additive perturbation $d(2)$, the resulting measured state is $x^{\mathrm{m}}_2 = x_{\overline{u}^*_{1,\cdot}}(1, x^{\mathrm{m}}_1) + d(2)$.

Consider next time $j = 2$ and the measured state x^{m}_2. We can apply the nominal open-loop control element $u^*_{0,2}$ to obtain $x_{u^*_{0,2+\cdot}}(1, x^{\mathrm{m}}_2) = f(x^{\mathrm{m}}_2, u^*_{0,2})$. We can also perform re-optimization to obtain and apply the control value $u^*_{2,0}$ resulting in the predicted state $x_{u^*_{2,\cdot}}(1, x^{\mathrm{m}}_2) = f(x^{\mathrm{m}}_2, u^*_{2,0})$. Or we can apply the approximate control $\overline{u}^*_{2,0}$ given by

$$\overline{u}^*_{2,0} = u^*_{0,2} + \frac{\partial u_2}{\partial p_2}(x_{u^*_{0,\cdot}}(2, x_0))(x^{\mathrm{m}}_2 - x_{u^*_{0,\cdot}}(2, x_0)) \tag{6.21}$$

Observe

$$\begin{aligned} &\|x^{\mathrm{m}}_2 - x_{u^*_{0,\cdot}}(2, x_0)\| \\ &\quad = \|d(2) + x_{\overline{u}^*_{1,\cdot}}(1, x^{\mathrm{m}}_1) - x_{u^*_{0,\cdot}}(2, x_0)\| \\ &\quad \leq \|d(2)\| + \|x_{\overline{u}^*_{1,\cdot}}(1, x^{\mathrm{m}}_1) - x_{u^*_{0,1+\cdot}}(1, x^{\mathrm{m}}_1)\| + \|x_{u^*_{0,1+\cdot}}(1, x^{\mathrm{m}}_1) - x_{u^*_{0,\cdot}}(2, x_0)\| \\ &\quad = \|d(2)\| + \|f(x^{\mathrm{m}}_1, \overline{u}^*_{1,0}) - f(x^{\mathrm{m}}_1, u^*_{0,1})\| + \|f(x^{\mathrm{m}}_1, u^*_{0,1}) - f(x_{u^*_{0,\cdot}}(1, x_0), u^*_{0,1})\| \\ &\quad \leq \|d(2)\| + C_2\|\overline{u}^*_{1,0} - u^*_{0,1}\| + C_3\|x^{\mathrm{m}}_1 - x_{u^*_{0,\cdot}}(1, x_0)\| \\ &\quad = \|d(2)\| + \tilde{C}_2 d_1 + C_3 d_1 \\ &\quad \leq C_4(\underbrace{\|d(2)\| + R_1}_{=:d_2}) \text{ with } R_1 = \mathcal{O}(d_1) \end{aligned} \tag{6.22}$$

for some $C_2, \tilde{C}_2, C_3, C_4 > 0$ with the last identity due to (6.19). Using (6.22), we obtain

$$\begin{aligned} u^*_{2,0} &= \overline{u}^*_{2,0} + \mathcal{O}(\|x^{\mathrm{m}}_2 - x_{u^*_{0,\cdot}}(2, x_0)\|^2) \\ &= \overline{u}^*_{2,0} + \mathcal{O}(d_2^2) \end{aligned} \tag{6.23}$$

The predicted state when $\overline{u}^*_{2,0}$ is applied will then be $x_{\overline{u}^*_{2,\cdot}}(1, x^{\mathrm{m}}_2) = f(x^{\mathrm{m}}_2, \overline{u}^*_{2,0})$ and by Assumption 6.3.1, we have

$$\begin{aligned} \left\|x_{u^*_{2,\cdot}}(1, x^{\mathrm{m}}_2) - x_{\overline{u}^*_{2,\cdot}}(1, x^{\mathrm{m}}_2)\right\| &= \left\|f(x^{\mathrm{m}}_2, u^*_{2,0}) - f(x^{\mathrm{m}}_2, \overline{u}^*_{2,0})\right\| \\ &\leq C_5\|u^*_{2,0} - \overline{u}^*_{2,0}\| \end{aligned}$$

for some $C_5 > 0$. We then obtain the relation

$$x_{u^*_{2,\cdot}}(1, x^{\mathrm{m}}_2) \quad = \quad x_{\overline{u}^*_{2,\cdot}}(1, x^{\mathrm{m}}_2) + \mathcal{O}(d_2^2)$$

due to (6.23). And due to perturbation $d(3)$, the resulting measured state is $x^{\mathrm{m}}_3 = x_{\overline{u}^*_{2,\cdot}}(1, x^{\mathrm{m}}_2) + d(3)$.

We repeat the exact process for time $j = 3$ and the measured state x^{m}_3 and obtain the key inequalities

$$x^{\mathrm{m}}_3 \leq x_{u^*_{0,\cdot}}(3, x_0) + \mathcal{O}(\underbrace{\|d(3)\| + R_2}_{=:d_3}) \tag{6.24}$$

where $R_2 = \mathcal{O}(d_2)$ and

$$x_{u^*_{3,\cdot}}(1, x^{\mathrm{m}}_3) = x_{\overline{u}^*_{3,\cdot}}(1, x^{\mathrm{m}}_3) + \mathcal{O}(d_3^2)$$

We now formalize the results of the discussion. We begin by defining the sequence $\{d_k\}$ by

$$\begin{aligned} &d_0 := 0 \\ &d_1 := \|d(1)\| \\ &d_k := \|d(k)\| + R_{k-1},\ k = 2, \ldots, N-1 \end{aligned}$$

where R_j is some term with order of magnitude d_j, i.e., $R_j = \mathcal{O}(d_j)$, $j = 1, \ldots, N-2$, following the way d_1, d_2 and d_3 are constructed in (6.20), (6.22) and (6.24), respectively.

The subsequent lemmas and corollary give a rigorous description of the preceding discussion.

Lemma 6.3.2. *Suppose that the Assumptions 6.1.2 and 6.3.1 hold. For $k = 1, \ldots, N-1$,*

$$\|x^m_k - x_{u^*_{0,\cdot}}(k, x_0)\| = \mathcal{O}(d_k)$$

for some $C > 0$.

Proof. Let $k \in \{1, \ldots, N-1\}$. Recall (6.22) and (6.24). Inductively, observe that

$$\begin{aligned} &\|x^{\mathrm{m}}_k - x_{u^*_{0,\cdot}}(k, x_0)\| \\ &\quad = \|d(k) + x_{\overline{u}^*_{k-1,\cdot}}(1, x^{\mathrm{m}}_{k-1}) - x_{u^*_{0,\cdot}}(k, x_0)\| \\ &\quad \leq \|d(k)\| + \|x_{\overline{u}^*_{k-1,\cdot}}(1, x^{\mathrm{m}}_{k-1}) - x_{u^*_{0,k-1+\cdot}}(1, x^{\mathrm{m}}_1)\| \\ &\qquad + \|x_{u^*_{0,k-1+\cdot}}(1, x^{\mathrm{m}}_{k-1}) - x_{u^*_{0,\cdot}}(k, x_0)\| \\ &\quad = \|d(k)\| + \|f(x^{\mathrm{m}}_{k-1}, \overline{u}^*_{k-1,0}) - f(x^{\mathrm{m}}_{k-1}, u^*_{0,k-1})\| \\ &\qquad + \|f(x^{\mathrm{m}}_{k-1}, u^*_{0,k-1}) - f(x_{u^*_{0,\cdot}}(k-1, x_0), u^*_{0,k-1})\| \\ &\quad \leq \|d(k)\| + C_6\|\overline{u}^*_{k-1,0} - u^*_{0,k-1}\| + C_7\|x^{\mathrm{m}}_{k-1} - x_{u^*_{0,\cdot}}(k-1, x_0)\| \\ &\quad = \|d(k)\| + \tilde{C}_6(d_{k-1}) + \tilde{C}_7(d_{k-1}) \leq C_8(\|d(k)\| + \mathcal{O}(d_{k-1})\|) \end{aligned}$$

for some $C_6, \tilde{C}_6, C_7, \tilde{C}_7, C_8 > 0$ □

Corollary 6.3.3. *Suppose that the Assumptions 6.1.2 and 6.3.1 hold. For* $k = 1, \ldots, N-1$,

$$\ell(x_k^m, u_{k,0}^*) = \ell(x_k^m, \overline{u}_{k,0}^*) + \mathcal{O}(d_k^2)$$

Proof. The assertion directly follows from Lemma 6.3.2 since for some $\tilde{C}, C > 0$

$$\|\ell(x_k^{\mathrm{m}}, u_{k,0}^*) - \ell(x_k^{\mathrm{m}}, \overline{u}_{k,0}^*)\| = \tilde{C}\|u_{k,0}^* - \overline{u}_{k,0}^*\| = C\|x_k^{\mathrm{m}} - x_{u_{0,\cdot}^*}(k, x_0)\|^2$$

□

Lemma 6.3.4. *Suppose that the Assumptions 6.1.2 and 6.3.1 hold. Then for* $k = 1, \ldots, N-1$,

$$\|x_k^m - x_{u_{k-1,\cdot}^*}(1, x_{k-1}^m))\| \le \|d(k)\| + \mathcal{O}(d_{k-1}^2)$$

Proof. Observe that for some $\tilde{C}, C > 0$

$$\begin{aligned}
\|x_k^{\mathrm{m}} - x_{u_{k-1,\cdot}^*}(1, x_{k-1}^{\mathrm{m}}))\| &= \|d(k) + x_{\overline{u}_{k-1,\cdot}^*}(1, x_{k-1}^{\mathrm{m}}) - x_{u_{k-1,\cdot}^*}(1, x_{k-1}^{\mathrm{m}}))\| \\
&\le \|d(k)\| + \|f(x_{k-1}^{\mathrm{m}}, \overline{u}_{k-1,0}^*) - f(x_{k-1}^{\mathrm{m}}, u_{k-1,0}^*)\| \\
&\le \|d(k)\| + \tilde{C}\|\overline{u}_{k-1}^* - u_{k-1}^*\| \\
&\le \|d(k)\| + C\|x_{k-1}^{\mathrm{m}} - x_{u_{0,\cdot}^*}(k-1, x_0)\|^2
\end{aligned}$$

We obtain the assertion by using Lemma 6.3.2. □

We are now in the position to compare the value $J_N^{\mathrm{sens}}(x_0)$ of the trajectory driven by the sensitivity-based controls $\overline{u}_{j,0}^*,\ j = 0, \ldots, N-1$ to the nominal value

$$J_N^{\mathrm{nmult}}(x_0) = \sum_{j=0}^{N-1} \ell(x_{u_{0,\cdot}^*}(j, x_0), u_{0,j}^*)$$

Based on the results above, we now provide the key theorem for the analysis in this section.

Theorem 6.3.5. *Let the Assumptions 6.1.2 and 6.3.1 hold. Suppose* $V_i,\ i = 1, \ldots, N$, *is uniformly continuous on* $\mathbb{X}$ *with modulus of continuity* ω_{V_i}. *Consider an initial value* $x_0 \in \mathbb{X}$ *and external perturbations represented by the sequence* $\{d(j)\}$ *such that* $\mathcal{P}_{N-j}(x_j^m),\ j = 1, \ldots, N-1$ *is feasible. Then*

$$\left|J_N^{nmult}(x_0) - J_N^{sens}(x_0)\right| \le \sum_{j=1}^{N-1} \omega_{V_{N-j}}\left(\|d(j)\| + \mathcal{O}(d_{j-1}^2)\right) + \sum_{j=1}^{N-1} \mathcal{O}(d_j^2) \quad (6.25)$$

Proof. From the value definition (6.18) and Corollary 6.3.3, we have

$$
\begin{aligned}
&\left| J_N^{\mathrm{nmult}}(x_0) - \sum_{j=0}^{N-1} \overline{\lambda}_j \right| \\
= \quad & \left| \sum_{j=0}^{N-1} \ell(x_{u^*_{0,\cdot}}(j, x_0^{\mathrm{m}}), u^*_{0,j}) \; - \; \sum_{j=0}^{N-1} \ell(x_j^{\mathrm{m}}, \overline{u}^*_{j,0}) \right| \\
\leq \quad & \left| \sum_{j=0}^{N-1} \ell(x_{u^*_{0,\cdot}}(j, x_0^{\mathrm{m}}), u^*_{0,j}) \; - \; \sum_{j=0}^{N-1} \ell(x_j^{\mathrm{m}}, u^*_{j,0}) \right| + \sum_{j=0}^{N-1} \mathcal{O}(d_j^2) \\
= \quad & \left| \sum_{j=0}^{N-1} \ell(x_{u^*_{0,\cdot}}(j, x_0^{\mathrm{m}}), u^*_{0,j}) \right. \\
& - \ell(x_{u^*_{0,\cdot}}(0, x_0^{\mathrm{m}}), u^*_{0,0}) \; - \; \sum_{j=0}^{N-2} \ell(x_{u^*_{1,\cdot}}(j, x_1^{\mathrm{m}}), u^*_{1,j}) \\
& + \sum_{j=0}^{N-2} \ell(x_{u^*_{1,\cdot}}(j, x_1^{\mathrm{m}}), u^*_{1,j}) \\
& - \ell(x_{u^*_{1,\cdot}}(0, x_1^{\mathrm{m}}), u^*_{1,0}) \; - \; \sum_{j=0}^{N-3} \ell(x_{u^*_{2,\cdot}}(j, x_2^{\mathrm{m}}), u^*_{2,j}) + \\
& \qquad \vdots \\
& + \sum_{j=0}^{1} \ell(x_{u^*_{N-2,\cdot}}(j, x_{N-2}^{\mathrm{m}}), u^*_{N-2,j}) \\
& - \ell(x_{u^*_{N-2,\cdot}}(0, x_{N-2}^{\mathrm{m}}), u^*_{N-2,0}) - \ell(x_{u^*_{N-1,\cdot}}(0, x_{N-1}^{\mathrm{m}}), u^*_{N-1,0}) \\
& + \ell(x_{u^*_{N-1,\cdot}}(0, x_{N-1}^{\mathrm{m}}), u^*_{N-1,0}) \\
& \left. - \ell(x_{u^*_{N-1,\cdot}}(0, x_{N-1}^{\mathrm{m}}), u^*_{N-1,0}) \right| \; + \sum_{j=1}^{N-1} \mathcal{O}(d_j^2) \\
\leq \quad & \left| V_{N-1}(x_{u^*_{0,\cdot}}(1, x_0^{\mathrm{m}})) - V_{N-1}(x_1^{\mathrm{m}}) \right| \\
& + \left| V_{N-2}(x_{u^*_{1,\cdot}}(1, x_1^{\mathrm{m}})) - V_{N-2}(x_2^{\mathrm{m}}) \right| + \\
& + \left| V_{N-3}(x_{u^*_{2,\cdot}}(1, x_2^{\mathrm{m}})) - V_{N-3}(x_3^{\mathrm{m}}) \right| \; + \ldots \\
& + \left| V_1(x_{u^*_{N-2,\cdot}}(1, x_{N-2}^{\mathrm{m}})) - V_1(x_{N-1}^{\mathrm{m}}) \right| \; + \; \sum_{j=1}^{N-1} \mathcal{O}(d_j^2) \\
\leq \quad & \sum_{j=1}^{N-1} \omega_{V_{N-j}} \left(\| (x_{u^*_{j-1,\cdot}}(1, x_{j-1}^{\mathrm{m}})) - x_j^{\mathrm{m}} \| \right) + \sum_{j=1}^{N-1} \mathcal{O}(d_j^2) \\
\leq \quad & \sum_{j=1}^{N-1} \omega_{V_{N-j}} \left(\| d(j) \| + \mathcal{O}(d_{j-1}^2) \right) + \sum_{j=1}^{N-1} \mathcal{O}(d_j^2)
\end{aligned}
$$

□

Theorem 6.3.5 allows to quantify the performance difference between approximate re-optimizing and not re-optimizing for the finite horizon problem similar to the results given by Corollaries 3.3.3 and 3.3.5.

Let us compare Theorem 6.3.5 to the results obtained in Section 3.3. In Section 3.3, we established that the difference between $J_N^{\text{nmult}}(x_0)$ and $J_N^{\text{pmult}}(x_0) = \sum_{j=0}^{N-1} \lambda_{j,j,0}$ depends on the modulus ω_{J_i} of elements of the perturbation sequence $d(\cdot)$ and that the difference between J_N^{nmult} and $J_N^{\text{upd}}(x_0) = \sum_{j=0}^{N-1} \lambda_{j,j,j}$ is determined by the ω_{V_i} of elements of $d(\cdot)$. In this section, we find that the difference between $J_N^{\text{nmult}}(x_0)$ and $J_N^{\text{sens}}(x_0) = \sum_{j=0}^{N-1} \overline{\lambda}_j$ also depends on the ω_{V_i} of elements of $d(\cdot)$ with additional error terms depending on squares of accumulated past errors. These results quantitatively characterize the 3 schemes. And due to the inequality $\omega_{V_i} \leq \omega_{J_i}$, provided that the system is unstable but controllable, for moderate perturbations $d(j)$ where $d(j)$ is much more dominant than $\mathcal{O}(d_{j-1}^2)$, whereas re-optimizing gives the most benefit, using sensitivity-based updates in place of re-optimizing still offers considerable benefit compared to not re-optimizing.

The following corollary generalizes Theorem 6.3.5 allowing an arbitrary time instant $k \in \{0, \ldots, N-1\}$ to be the reference point other than $k = 0$.

Corollary 6.3.6. *Let the assumptions of Theorem 6.3.5 hold. Then*

$$\left| J_{N-k}^{nmult}(x_k^m) - \sum_{j=k}^{N-1} \overline{\lambda}_j \right| \leq \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}} \left(\|d(k+j)\| + \mathcal{O}(d_{k+j-1}^2) \right) + \sum_{j=0}^{N-k-1} \mathcal{O}(d_{k+j}^2)$$

Proof. The proof follows using the same technique as in the proof of Theorem 6.3.5 with the appropriate changes in the indices. □

The next goal is to obtain a performance estimate for the SBM MPC feedback $\overline{\mu}_{N,m}$ similar to the analysis for the updated MPC feedback $\hat{\mu}_{N,m}$ in Chapter 4. As done in Chapter 4, the idea is to construct the corresponding perturbed version of $\mathcal{P}_\alpha$ for quantities involving costs incurred along the trajectory driven by the SBM MPC feedback $\overline{\mu}_{N,m}$. We begin with the following two lemmas.

Lemma 6.3.7. *Let the Assumption 2.1.4 and the assumptions of Theorem 6.3.5 hold. Suppose further B_K, $K = 1, \ldots, N$, is uniformly continuous on $\mathbb{R}_0^+$ with modulus of continuity ω_{B_K}. Consider $x_0 = x \in \mathbb{X}$ and an optimal control $u^*(\cdot) \in \mathbb{U}^N$ for the finite horizon optimal control problem $\mathcal{P}_N(x)$ with optimization horizon N. Then for each $m = 1, \ldots, N-1$ and each $j = 0, \ldots, N-m-1$,*

$$\begin{aligned} V_N(x_m^m) \quad \leq \quad & \sum_{n=0}^{j-1} \overline{\lambda}_{n+m} + B_{N-j}(\overline{\lambda}_{j+m}) + \omega_{B_{N-j}}(\mathcal{O}(d_{j+m}^2)) \\ & + \sum_{n=1}^{N-j-1} \omega_{V_{N-j-n}} \left(\|d(j+n+m)\| + \mathcal{O}(d_{j+n-1+m}^2) \right) \\ & + \sum_{n=j}^{N-1} \mathcal{O}(d_{n+m}^2) \end{aligned}$$

Proof. Observe that

$$
\begin{aligned}
&V_N(x^{\mathrm{m}}_m)\\
&\leq \sum_{j=0}^{N-1} \overline{\lambda}_{n+m} = \sum_{n=0}^{j-1} \overline{\lambda}_{n+m} + \sum_{n=j}^{N-1} \overline{\lambda}_{n+m}\\
&\leq \sum_{n=0}^{j-1} \overline{\lambda}_{n+m} + J^{\mathrm{nom}}_{N-j}(x^{\mathrm{m}}_{j+m})\\
&\quad + \sum_{n=1}^{N-j-1} \omega_{V_{N-j-n}} \left(\|d(j+n+m)\| + \mathcal{O}(d^2_{j+n-1+m})\right) + \sum_{n=j}^{N-1} \mathcal{O}(d^2_{n+m})\\
&\leq \sum_{n=0}^{j-1} \overline{\lambda}_{n+m} + B_{N-j}(\ell^*(x^{\mathrm{m}}_{j+m}))\\
&\quad + \sum_{n=1}^{N-j-1} \omega_{V_{N-j-n}} \left(\|d(j+n+m)\| + \mathcal{O}(d^2_{j+n-1+m})\right) + \sum_{n=j}^{N-1} \mathcal{O}(d^2_{n+m})\\
&\leq \sum_{n=0}^{j-1} \overline{\lambda}_{n+m} + B_{N-j}(\overline{\lambda}_{j+m}) + \omega_{B_{N-j}}(\mathcal{O}(d^2_{j+m}))\\
&\quad + \sum_{n=1}^{N-j-1} \omega_{V_{N-j-n}} \left(\|d(j+n+m)\| + \mathcal{O}(d^2_{j+n-1+m})\right) + \sum_{n=j}^{N-1} \mathcal{O}(d^2_{n+m})
\end{aligned}
$$

where the second inequality is due to Corollary 6.3.6, the third inequality due to Assumption 2.1.4 and the fourth due to Corollary 6.3.3. □

Lemma 6.3.8. *Let the assumptions of Corollary 4.1.3 and of Theorem 6.3.5 hold. Suppose further B_K, $K = 1, \ldots, N$, is uniformly continuous on $\mathbb{R}^+_0$ with modulus of continuity ω_{B_K}. Then for $k = 0, \ldots, N-2$, we have the inequalities*

$$
\begin{aligned}
&\sum_{j=k}^{N-1} \overline{\lambda}_j \leq B_{N-k}(\overline{\lambda}_k) + \omega_{B_{N-k}}(\mathcal{O}(d^2_k))\\
&\qquad + \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}} \left(\|d(k+j)\| + \mathcal{O}(d^2_{k+j-1})\right) + \sum_{j=0}^{N-k-1} \mathcal{O}(d^2_{k+j})
\end{aligned}
$$

Proof. From Corollary 6.3.6, we observe

$$\begin{aligned}\sum_{j=k}^{N-1} \overline{\lambda}_j \le V_{N-k}(x_k^{\mathrm{m}})& \\ &+ \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}} \big(\|d(k+j)\| + \mathcal{O}(d_{k+j-1}^2)\big) + \sum_{j=0}^{N-k-1} \mathcal{O}(d_{k+j}^2) \\ &\le B_{N-k}(\ell^*(x_k^{\mathrm{m}})) \\ &+ \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}} \big(\|d(k+j)\| + \mathcal{O}(d_{k+j-1}^2)\big) + \sum_{j=0}^{N-k-1} \mathcal{O}(d_{k+j}^2) \\ &\le B_{N-k}(\overline{\lambda}_k) + \omega_{B_{N-k}}(\mathcal{O}(d_k^2)) \\ &+ \sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}} \big(\|d(k+j)\| + \mathcal{O}(d_{k+j-1}^2)\big) + \sum_{j=0}^{N-k-1} \mathcal{O}(d_{k+j}^2)\end{aligned}$$

where the second inequality follows from to Assumption 2.1.4 and the third is due to Corollary 6.3.3. □

By combining these results, we can now state the following counterpart of Proposition 2.1.7. The statement yields necessary conditions which hold if the values λ_n coincide with $\overline{\lambda}_n$ and ν with $V_N(x_m^{\mathrm{m}})$.

Corollary 6.3.9. *Consider $N \ge 1, m \in \{1,\ldots,N-1\}$ and let the assumptions of Lemmas 6.3.7 and 6.3.8 hold. Let $x = x_0 \in \mathbb{X}$ and consider external perturbations represented by the sequence $\{d(k)\}$ where $d(k) = 0$ for $k \ge m$ generating the trajectories $\tilde{x}_{\overline{\mu}_{N,N-1}}(n,x) = x_n^m$. Consider a sequence $\lambda_n \ge 0,\ n = 0,\ldots,N-1$ and a value $\nu \ge 0$ such that $\lambda_n = \overline{\lambda}_n,\quad n = 0,\ldots,N-1$ and $\nu = V_N(x_m^m)$ holds. Then the inequalities*

$$\sum_{n=k}^{N-1} \lambda_n \le B_{N-k}(\lambda_k) + \xi_k^1,\ \ k = 0,\ldots,N-2$$

$$\nu \le \sum_{n=0}^{j-1} \lambda_{n+m} + B_{N-j}(\lambda_{j+m}) + \xi_j^2,\ \ j = 0,\ldots,N-m-1$$

hold for

$$\begin{aligned}\xi_k^1 = &\sum_{j=1}^{N-k-1} \omega_{V_{N-k-j}} \big(\|d(k+j)\| + \mathcal{O}(d_{k+j-1})^2\big) \\ &+\ \omega_{B_{N-j}}(\mathcal{O}(d_{j+m}^2)) + \sum_{j=0}^{N-k-1} \mathcal{O}(d_{k+j}^2) \\ \xi_j^2 = &\sum_{n=1}^{N-j-1} \omega_{V_{N-j-n}} \big(\|d(j+n+m)\| + \mathcal{O}(d_{j+n-1+m}^2)\big) \\ &+\ \omega_{B_{N-k}}(\mathcal{O}(d_k^2)) + \sum_{n=j}^{N-1} \mathcal{O}(d_{n+m}^2)\end{aligned}$$

This corollary allows us to formulate a corresponding perturbed version of $\mathcal{P}_\alpha$ (recall Theorem 2.1.8 for the definition of $\mathcal{P}_\alpha$ and Section 4.2 for the perturbed versions).

$$
\begin{aligned}
&\alpha^{\text{sens}} := \inf_{\overline{\lambda}_n, n=0,\dots,N-1, \nu^{\text{sens}}} \frac{\sum_{n=0}^{N-1} \overline{\lambda}_n - \nu^{\text{sens}}}{\sum_{n=0}^{m-1} \overline{\lambda}_n} \\
&\text{subject to} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \mathcal{P}_\alpha^{\text{sens}} \\
&\sum_{n=k}^{N-1} \overline{\lambda}_n \le B_{N-k}(\overline{\lambda}_k) + \xi^{\text{sens}}, \quad k = 0, \dots, N-2 \\
&\nu^{\text{sens}} \le \sum_{n=0}^{j-1} \overline{\lambda}_{n+m} + B_{N-j}(\overline{\lambda}_{j+m}) + \xi^{\text{sens}}, \quad j = 0, \dots, N-m-1 \\
&\sum_{n=0}^{m-1} \overline{\lambda}_n \ge \zeta, \ \overline{\lambda}_m, \dots, \overline{\lambda}_{N-1}, \ \nu^{\text{sens}} \ge 0
\end{aligned}
$$

with

$$
\xi^{\text{sens}} = \max_{\substack{k \in \{0,\dots,N-2\} \\ j \in \{0,\dots,N-m-1\}}} \{\xi_k^1, \xi_j^2\} \quad \text{with } \xi_k^1 \text{ and } \xi_j^2 \text{ from Corollary 6.3.9} \tag{6.26}
$$

The formulation of $\mathcal{P}_\alpha^{\text{sens}}$ implies the applicability of the statements we obtained in Sections 4.2 and 4.3, namely Theorem 4.2.3, Lemma 4.3.1 and Theorem 4.3.3, to obtain performance and stability properties of the closed-loop system driven by the SBM feedback $\overline{\mu}_{N,m}$.

Remark 6.3.10. (a) Lemma 4.2.2 is straightforwardly applied to obtain an analogous statement to Theorem 4.2.3 to estimate the solution α^{sens} in reference to the nominal case solution α^{nmult}, we have

$$
\alpha^{\text{sens}} \ge \alpha^{\text{nmult}} - \frac{B_{m+1}(\xi^{\text{sens}}) + \xi^{\text{sens}}}{\zeta}
$$

(b) An analagous statement to Lemma 4.3.1 is also obtained giving the corresponding relaxed dynamic programming inequality which provides the suboptimality index $\tilde{\alpha}^{\text{sens}}$ given by

$$
\tilde{\alpha}^{\text{sens}} = \alpha^{\text{sens}} - \frac{\sigma}{\zeta}
$$

where

$$
\sigma = \sum_{j=1}^{m-1} \omega_{V_{N-j}} \left(\|d(j)\| + \mathcal{O}(d_{j-1}^2) \right) + \sum_{j=1}^{N-1} \mathcal{O}(d_j^2)
$$

Using the relation $\omega_{V_k} \le \omega_{J_k}$, we conclude that SBM MPC yields better robustness properties than the non-updated m-step MPC.

(c) The same statement as Theorem 4.3.3 is also obtained showing that the perturbed sensitivity-based m-step closed-loop system with feedback law $\overline{\mu}_{N,m}$ is semi-globally practically asymptotically stable on $\mathbb{X}$ with respect to d.

7 Numerical examples

In this chapter, we present examples of the implementation of the SBM MPC and comparisons to the MPC variants discussed in the previous chapters. In Section 7.1, we consider once again the inverted pendulum where we illustrate how does SBM MPC compare against the schemes we discussed so far, examine the suboptimality performance of each schemes and analyze the influence of perturbations on approximating re-optimization. In Section 7.2, we apply MPC schemes on an electric circuit process. We demonstrate taking advantage of the matrix structures arising from the problem formulation, investigate the computational expense aspect of the schemes and finally, illustrate the usually opposing objectives of reducing computational expense and improving performance and how SBM MPC maintains a compromise between these two objectives. A preliminary version of the results presented in Section 7.2 is published in [51].

7.1 Case study: inverted pendulum

In order to illustrate our results, we consider once again the nonlinear inverted pendulum model presented in Section 4.4 depicting a cart on a track to which a rigid pendulum is attached and able to rotate freely. Recall the aim to stabilize the pendulum to the unstable inverted position and the previously defined stage cost as well.

We aim to compare SBM MPC to the other schemes, namely, the m-step and updated m-step feedback controllers. As in Section 4.4, we use here the same optimization horizon $N = 15$, initial value $x_0 = (-\pi - 0.1, 0, -0.1, 0)$ and a fixed randomly generated perturbation sequence of the form $d(k) = [0, 0, d_3(k), 0]^\top$, $k \in \mathbb{N}$, with values in the interval $[-\overline{d}_3, 0]$ for $\overline{d}_3 = 0.05$. Aside from the system dynamics and the initial condition, no further constraints (e.g., box contraints) are imposed on the states and the control.

The simulations are implemented using OCPIDDAE-1 (see the user manual [26]) which is a software package that discretizes an optimal control problem, transforms it into a finite-dimensional NLP and solves it using SQP method. The package uses sensitivity analysis of the discretized OCP with respect to the so-called real-time parameter and computes sensitivity differentials which we use for the computation of the approximate solution. Sensitivities are computed as in Remark 5.5.3 where the use of the exact Hessian is a requirement. Updates are performed as in Algorithm 6.1.4.

First, let us recall Figure 4.2 which shows that compared with the 7-step MPC, improvement is manifested by applying the updates to the multistep scheme allowing the trajectory to move towards the equilibrium against the perturbations. This is once again shown in Figure 7.2. Now in addition, Figure 7.2 also depicts the improvement brought about this time by SB updates to the multistep scheme confirming the results obtained in Section 6.3. One can also observe that the SB 7-step MPC (shown in black) behaves closely like the updated 7-step MPC (shown in green) as pointed out in Remark 6.1.3. The figure also shows how all the schemes discussed in the thesis compare to the 1-step scheme – the most robust MPC scheme (shown in cyan).

Recall that the sensitivity theorem limits its assertion to some neighborhood of the optimal solution. We next examine the effects of increasing the magnitude of perturbation to the quality of the approximation of the optimal solution and the robustness of the schemes. To this end, we vary the magnitude of $d_3(k)$ in the perturbation sequence of $d(k) = [0, 0, d_3(k), 0]^\top, k \in \mathbb{N}$. Figures 7.1–7.4 illustrates that the bigger the magnitude of $d_3(k)$, the larger the corresponding δ becomes in the robust stability Definition 2.2.4 where the system behaves like an asymptotic stable system until the trajectory is within a distance of δ from the equilibrium. We show here plots corresponding to $\|d_3(k)\| = 0.01, 0.05, 0.1, 0.5$. For $\|d_3(k)\| = 1$, the perturbations become so big that no meaningful trend can be reported for the resulting trajectories. In Figure 7.4, one can observe that despite a considerable perturbation magnitude of 0.5, the re-optimization provides an effective coping mechanism against the perturbation signifying robustness. This can also be said about the SBM feedback since in this case, it approximates well the updated scheme.

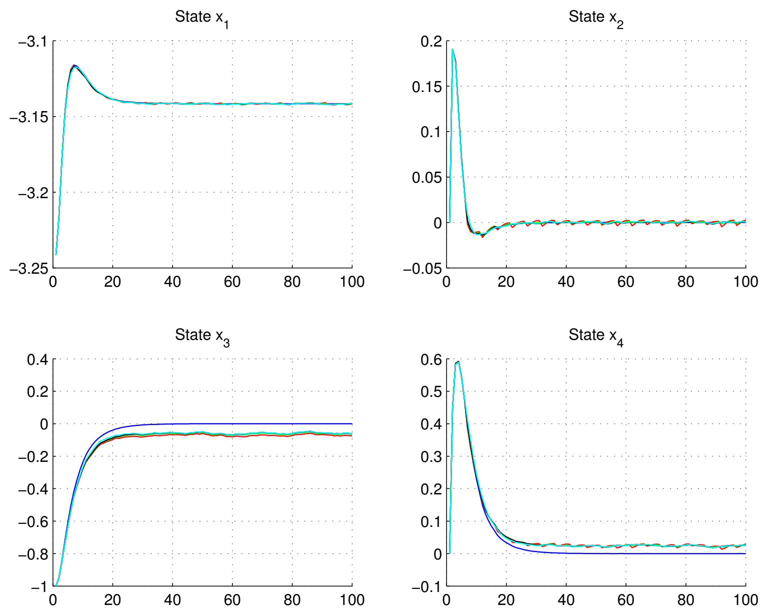

Figure 7.1: State trajectories driven by the 7-step MPC scheme for nominal system (blue), the 1-step (cyan), 7-step (red), updated 7-step (green) and SB 7-step (black) MPC schemes for the perturbed system with $\|d_3(k)\| = 0.01$.

Table 7.1.1 presents the performance index α^{sens} of the SBM MPC computed

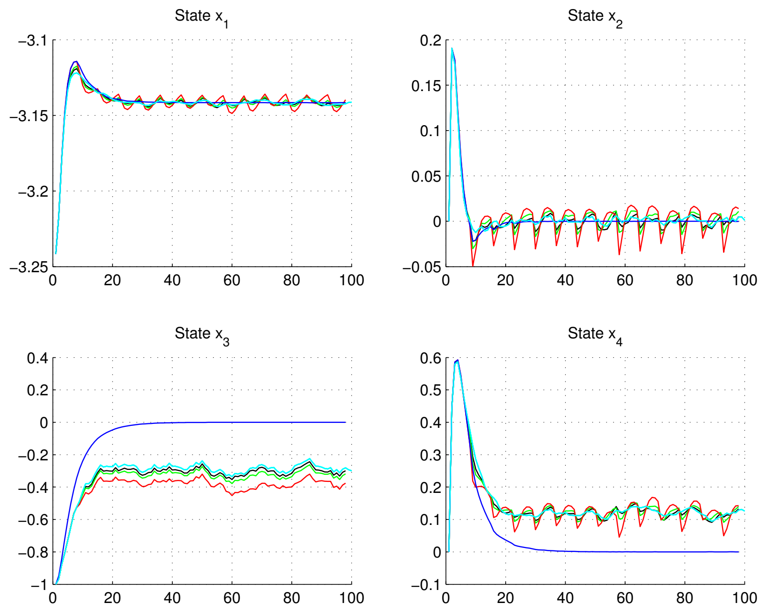

Figure 7.2: State trajectories driven by the 7-step MPC scheme for nominal system (blue), the 1-step (cyan), 7-step (red), updated 7-step (green) and SB 7-step (black) MPC schemes for the perturbed system with $\|d_3(k)\| = 0.05$.

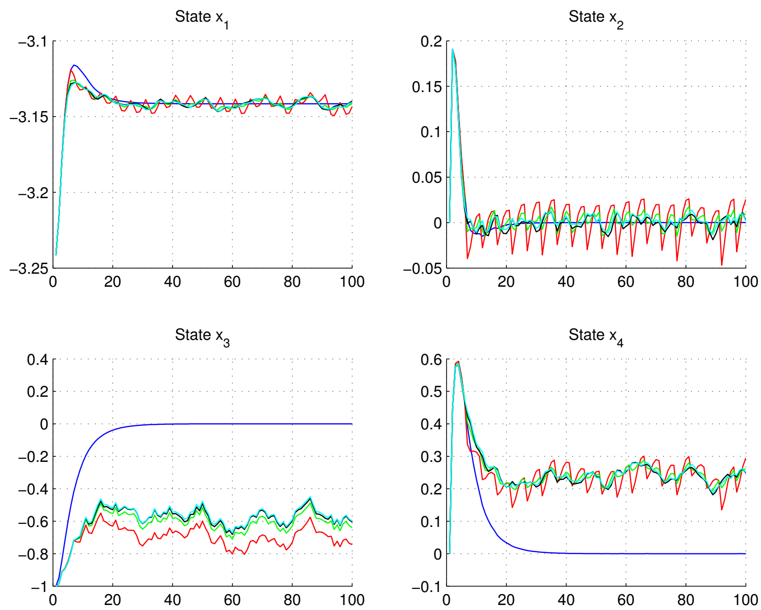

Figure 7.3: State trajectories driven by the 7-step MPC scheme for nominal system (blue), the 1-step (cyan), 7-step (red), updated 7-step (green) and SB 7-step (black) MPC schemes for the perturbed system with $\|d_3(k)\| = 0.1$.

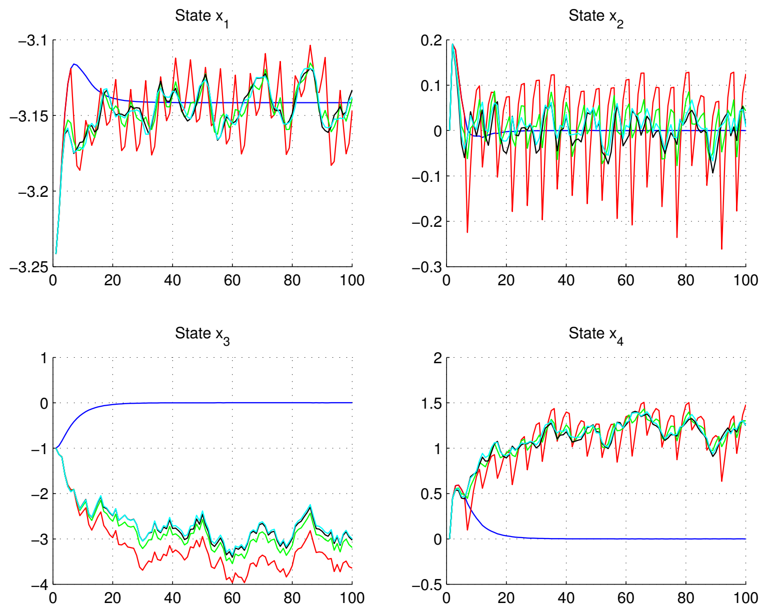

Figure 7.4: State trajectories driven by the 7-step MPC scheme for nominal system (blue), the 1-step (cyan), 7-step (red), updated 7-step (green) and SB 7-step (black) MPC schemes for the perturbed system with $\|d_3(k)\| = 0.5$.

from the generated trajectories using the approach presented in [35] as briefly explained in Section 4.4. We vary m and compute the values of α^{sens} for the first three iterations of each scheme. Recall Table 4.4.2 where the values of α for the nominal multistep scheme are 'close' to being infinite horizon optimal having values $\alpha > 0.9$ and that a degree improvement to the values of α due to re-optimization are observed by comparing the m-step and the updated m-step MPC. The values for the SBM MPC approximate very well those of the updated MPC wherein the SBM MPC is even slightly better than the updated scheme for $m > 2$. This is visible in Figure 7.2 where both the updated and SBM MPC behave very closely to the 1-step MPC with the SBM MPC behaving more closely to it. Here, the combination of the error in approximating the optimal control and the additive perturbation makes the SBM, by chance, a good approximation of the 1-step MPC. Also found in the table, the negative values indicate that the region $\widehat{P}$ in the definition of $\widehat{P}$-practical asymptotic stability has been reached where the effects of the perturbations become dominant.

7.2 Case study: DC-DC converter

In this section, we examine the computational effort in implementing the MPC schemes. We investigate the challenges in implementing an MPC scheme on an embedded system with limited computing power alongside the aim of accelerating the MPC procedure for fast systems applications.

We apply MPC in an electronic circuit process setting. We implement the m-step MPC for a DC-DC converter model motivated by the goal of saving computational costs. We examine the system under perturbation and address the reduced

Table 7.1.1: Suboptimality index α of the schemes for various m and iterations

	SBM		
m	0	2m	3m
1	0.8667	0.8699	0.6032
2	0.8681	0.6383	0.8538
3	0.7957	0.7819	0.6237
4	0.7734	0.7175	0.5705
5	0.7746	0.7380	0.5016
6	0.7888	0.5215	0.4167
7	0.7671	0.5931	-0.0041
8	0.7481	0.5264	0.0413
9	0.7508	0.3208	-0.0081
10	0.7486	0.2030	-0.2069
11	0.7099	0.1549	0.0273
12	0.6826	0.0996	-0.0004
13	0.6745	-0.0152	-0.0500

robustness by introducing updates on the controller through the SBM MPC. We illustrate here that SBM MPC considerably reduces the computing requirements in terms of floating point operations (FLOPs) compared to a standard MPC formulation, while fulfilling the expectations for better performance compared to the multistep MPC. We also show how a control-hardware designer can optimally trade-off closed-loop performance considerations with computing requirements in order to make the controller suitable for a tightly constrained embedded system.

A synchronous step-down converter, also referred to as a DC-DC converter, is a switching electronic circuit that converts an input voltage level V_s to satisfy a desired voltage requirement V_o. The circuit topology is illustrated in Figure 7.1. We follow the modeling presented in [28].

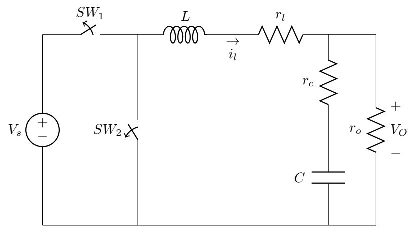

Figure 7.1: A DC-DC converter.

Here, r_0 denotes an ohmic output load, r_c is the equivalent series resistance of the capacitor with capacitance C and r_l is the internal resistance of the inductor with inductance L. The low-pass filter setup is comprised of two switches SW_1 and SW_2. In this setting, feedback control is used in order to stabilize the output voltage with respect to load, input voltage and component variations. At each switching period T_s, the output voltage and the current flowing in the inductor i_l are measured and used to control the opening and closing time of the two switches. When SW_1 is closed (i.e., at time $d(t) \cdot T_s$, where $d(t) \in [0, 1]$ is the duty cycle), SW_2 is opened and the input power is transferred to the output through the inductor. For the remaining time $(1 - d(t)) \cdot T_s$ of the switching period, the status of the switch are swapped providing a path for the inductor current i_l. This procedure is then repeated.

The described process leads to a set of affine time-invariant continuous-time state-space equations representing the two operating conditions. Defining the state vector as $x(t) := [i_l(t), V_o(t)]^\top$, the system behavior is modeled by

$$\dot{x}(t) = \begin{cases} A_c x(t) + b_c, & kT_s \le t \le (k + d(t))T_s \\ & (SW_1 \text{ is closed}) \\ A_c x(t), & (k + d(t))T_s \le t \le (k+1)T_s \\ & (SW_2 \text{ is closed}) \end{cases} \tag{7.1}$$

with output voltage given by $V_o(t) := c_c^T x(t)$ with A_c, b_c and c_c given by

$$A_c := \begin{bmatrix} -\frac{r_l}{L} & -\frac{1}{L} \\ \frac{1}{C}\frac{r_o}{r_o+r_c}\left(1 - Cr_c\frac{r_l}{L}\right) & -\frac{1}{C}\frac{1}{r_o+r_c}\left(1 + Cr_c\frac{r_o}{L}\right) \end{bmatrix}$$

$$b_c := \begin{bmatrix} \frac{1}{L} \\ \frac{r_o}{r_o+r_c}\frac{C}{L} \end{bmatrix}, \quad c_c := [0 \ 1]^\top$$

As reported in [63], this hybrid model may not be suitable for control purposes. To address this, a standard state-space averaging method [48] is used resulting in an average continuous-time model that merges the laws of the hybrid model and uses the duty cycle $d(t)$ as an input variable. This gives a nonlinear mathematical model to which linearization around an operating point can be carried out for further simplification of the controller design. This then leads to the state-space average model of the step-down converter (7.1) given by

$$\begin{array}{rcl} \dot{x}(t) & = & A_c x(t) + b_c \cdot d(t) \\ V_o(t) & = & c_c^T x(t) \end{array} \tag{7.2}$$

which is a linear system for which the states can be measured straightforwardly. Here, the input is the duty cycle $d(t)$ and the output is the output voltage $V_o(t)$. In addition, constraints arise from the converter topology, e.g., the duty cycle has to be between 0 and 1, and for safety reasons, the inductor current i_l be less than its saturation value $i_{l\max}$. This therefore implies the need for a controller design that can handle constraints.

7.2.1 Design of the controller

We consider the continuous-time finite horizon LQ problem defined by the cost function

$$J_c = x(T)^\top P_c x(T) + \int_0^T \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}^\top \begin{bmatrix} Q_c & 0 \\ 0 & R_c \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix} dt \tag{7.3}$$

where $Q_c = I, R_c = 1, P_c$ is the solution of continuous Ricatti equation and $T = 40\ \mu s$ is the prediction horizon. We assume zero-order hold. The function (7.3) represents the nominal closed-loop performance of the continuous-time model (7.2).

7.2.2 Discretization

We discretize the continuous-time model (7.2) and the continuous weighting matrices $\begin{bmatrix} Q_c & 0 \\ 0 & R_c \end{bmatrix}$ in (7.3) using the sample time T_s and zero-order hold approximation on the input. Let u_k denote the discrete domain counterpart of the input $d(t)$ in (7.2). Due to sampling (see, e.g., [4, Chapter 5] for discussion), (7.2) is transformed into

$$x_{k+1} \quad = \quad Ax_k + bu_k$$

where $A := e^{A_c T_s}$, $b := \left(\int_0^{T_s} e^{A_c \tau} d\tau\right) b_c$ and u_k is a constant control between sampling instants. The corresponding sampled-data cost function is given by

$$J_{T_s} = x_N^\top P x_N + \sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \begin{bmatrix} Q & M \\ M^\top & R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

where $N = \lceil T/T_s \rceil$ is the number of samples for the prediction horizon T.

7.2.3 MPC problem formulation

The MPC problem is defined by the core optimization problem solved at each time instant given by

$$\min_{x_k, u_k} \quad x_N^\top P x_N + \sum_{k=0}^{N-1} \begin{bmatrix} x_k \\ u_k \end{bmatrix}^\top \begin{bmatrix} Q & M \\ M^\top & R \end{bmatrix} \begin{bmatrix} x_k \\ u_k \end{bmatrix}$$

$$\begin{array}{lrcll} \text{s.t.} & x_0 & = & [\alpha, \beta]^\top & \\ & x_{j+1} & = & Ax_j + bu_j & j = \ 0, 1, \ldots, N-1 \\ & [0, 0]^\top & \leq & x_{j+1} \ \leq \ [i_{\text{lmax}}, V_s]^\top & j = \ 0, 1, \ldots, N-1 \\ & 0 & \leq & u_j \ \leq \ 1 & j = \ 0, 1, \ldots, N-1 \end{array} \tag{7.4}$$

We gauge the performance of the algorithm through the closed-loop cost function

$$J^{\text{cl}} \quad = \quad x_{N_T}^\top P x_{N_T} + \sum_{k=0}^{N_T - 1} \begin{bmatrix} x_k \\ \mu(x_k) \end{bmatrix}^\top \begin{bmatrix} Q & M \\ M^\top & R \end{bmatrix} \begin{bmatrix} x_k \\ \mu(x_k) \end{bmatrix} \tag{7.5}$$

for simulation time $N_T = \lceil T_T / T_s \rceil$ where T_T is the simulation time and μ is the MPC feedback (namely, $\mu_{N,m}, \hat{\mu}_{N,m}$ and $\overline{\mu}_{N,m}$.)

7.2.4 Matrix structures

Let us define the optimization variable

$$z := \left[x_0^{(1)} \ x_0^{(2)} \ u_0 \mid x_1^{(1)} \ x_1^{(2)} \ u_1 \mid \ \ldots \ \mid x_{N-1}^{(1)} \ x_{N-1}^{(2)} \ u_{N-1} \mid x_N^{(1)} \ x_N^{(2)} \ \right]^\top$$

It follows that the objective function has the form $\min\limits_z \ \frac{1}{2} z^\top H z$ given by

$$\min_z \quad \frac{1}{2} z^\top 2 \begin{bmatrix} Q & M & & & & \\ M^\top & R & & & & \\ & & \ddots & & & \\ & & & Q & M & \\ & & & M^\top & R & \\ & & & & & P \end{bmatrix} z$$

for which H has N blocks of $\begin{bmatrix} Q & M \\ M^\top & R \end{bmatrix}$ and a block of P. The equality constraints

$$\begin{array}{rcll} \begin{bmatrix} x_0^{(1)} \\ x_0^{(2)} \end{bmatrix} & = & \begin{bmatrix} \alpha \\ \beta \end{bmatrix} & \\ \begin{bmatrix} x_{j+1}^{(1)} \\ x_{j+1}^{(2)} \end{bmatrix} - A \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \end{bmatrix} - bu_j & = & 0 & j = 0, 1, \dots, N-1, \end{array} \tag{7.6}$$

composed of $2 \cdot (N+1)$ equations, can be written as

$$\begin{bmatrix} I_2 & & & & & \\ -A & -B & I_2 & & & \\ & & \ddots & & & \\ & & & -A & -B & I_2 \end{bmatrix} z = \begin{bmatrix} \alpha \\ \beta \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

which is of the form $C_{\text{eq}} z = d_{\text{eq}}$. The inequality constraints

$$\begin{array}{rcll} u_k - 0 & \geq & 0 & j = 0, 1, \dots, N-1 \\ \begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \geq & 0 & j = 1, 2, \dots, N \\ -u_k + u_{\text{ub}} & \geq & 0 & j = 0, 1, \dots, N-1 \\ -\begin{bmatrix} x_j^{(1)} \\ x_j^{(2)} \end{bmatrix} + \begin{bmatrix} x_{\text{ub}}^{(1)} \\ x_{\text{ub}}^{(2)} \end{bmatrix} & \geq & 0 & j = 1, 2, \dots, N, \end{array} \tag{7.7}$$

giving $(2+1) \cdot 2 \cdot N$ inequalities, can be written as

$$\left[\begin{array}{cccccc} 0 & 0 & 1 & & & \\ & & & 1 & & \\ & & & & \ddots & \\ & & & & & 1 \\ \hline 0 & 0 & -1 & & & \\ & & & -1 & & \\ & & & & \ddots & \\ & & & & & -1 \end{array}\right] z + \left[\begin{array}{c} 0 \\ \hline u_{\text{ub}} \\ x_{\text{ub}}^{(1)} \\ x_{\text{ub}}^{(2)} \\ \hline \vdots \\ \hline u_{\text{ub}} \\ x_{\text{ub}}^{(1)} \\ x_{\text{ub}}^{(2)} \end{array}\right] \geq 0$$

which we can write in the form $Cz \leq d$.

This shows that the problem (7.4) can be written in the form

$$\begin{array}{ll} \min\limits_{z} & \frac{1}{2} z^\top H z \\ \text{s.t.} & C_{\text{eq}} z - d_{\text{eq}} = 0 \\ & -Cz + d \geq 0 \end{array} \tag{7.8}$$

which is a QP wherein the constant matrix H happens to be the exact Hessian of the Lagrangian function of (7.4). Solving the optimization problem (7.8) is straightforward using `quadprog` in Matlab where active-set method can be chosen to solve the problem.

7.2.5 Implementing m-step and SBM MPC

For the standard MPC, at each time instant, we solve the problem (7.4) (or equivalently (7.8)) i.e., solve for the optimal solution z^* wherein we obtain the open-loop optimal control u^*. We apply u_0^* to the system and generate the next state. For the next time instant, the current state is measured and assigned as x_0 in (7.4). Then the process is repeated.

To reduce further the computational cost, we can use the m-step MPC in which we use the first m elements of the optimal control sequence u^*. In the hope of maintainaing robustness, we apply corrections on $u_1^*, u_2^*, \ldots, u_{m-1}^*$ using the sensitivity-based update rule (6.8). It is at the time instant m, where we solve an optimization problem again.

To solve the required updating/correcting sensitivities, we need to construct and solve the systems (6.10) for $j = 1, \ldots, m-1$. Consequently, by computing the sensitivities $\dfrac{\partial z^j}{\partial p_j}(x_j^*), j = 1, \ldots, m-1$, we obtain $\dfrac{\partial u^j}{\partial p_j}(x_j^*), j = 1, \ldots, m-1$. If we denote the problem formulation (7.4) (or (7.8)) by $\mathcal{P}_N(p_0)$, computing the sensitivities $\dfrac{\partial u^j}{\partial p_j}(x_j^*), j = 1, \ldots, m-1$ by (6.10) requires solving a sequence of systems for $j = 1, \ldots, m-1$, corresponding to the OCPs $\mathcal{P}_{N-j}(p_j)$ of decreasing horizon and adjusting parametric value.

It is worth mentioning that in this formulation, due to the nice structure of the matrices resulting from the OCP (7.4) (i.e., the involved Hessian and Jacobian matrices), adding the fact that these resulting matrices are constant matrices, the sequence of systems (6.10) can easily and immediately be constructed.

The exact Hessian $\nabla^2_{z^j z^j}\mathcal{L}^j({z^j}^*, \eta, x_j^*)$ of the Lagrangian function of $\mathcal{P}_{N-j}(p_j)$ evaluated at the nominal solution has the same form but smaller in size as H (i.e., the corresponding Hessian for $\mathcal{P}_N(p_0)$). It has $N-j$ blocks of $\begin{bmatrix} Q & M \\ M^\top & R \end{bmatrix}$ and a block of P. The submatrix $\nabla_{z^j} C_{\mathcal{A}^j}({z^j}^*, x_j^*)^\top$ denoting the Jacobian of the active constraints are obtained appropriately from the active constraints of $\mathcal{P}_N(p_0)$. This shows that the KKT matrix of the linear system corresponding to the OCP $\mathcal{P}_{N-j}(p_j)$ can be constructed through the submatrices of the KKT matrix solved for $P_N(p_0)$ which is already available. Finally, the right-hand side is a zero matrix except for the identity I_2 appearing in $\nabla_{p_j} C_{\mathcal{A}^j}({z^j}^*, {x_j}^*)^\top$ corresponding to $x_j - p_j$.

7.2.6 Numerical results

We consider a low-power (2 Watt) step-down converter setup with the following design parameters: $V_s = 6$ V, $r_l = 15.5$ mΩ, $V_o = 1$ V, $i_{lmax} = 4$ A, $r_o = 500$ mΩ, $C = 68$ μF, $L = 1.5$ μH and $r_c = 1.5$ mΩ.

We formulate different m-step and SBM MPC controllers by varying the sampling frequency $f_s \in [300\text{kHz}, 400\text{kHz}]$ (where $f_s := 1/T_s$) and the number of steps $m \in \{1, 2, ...10, 11\}$. Closed-loop simulations are performed in Matlab in order to measure the controller closed-loop performance and the required computing power in terms of floating point operations (FLOPs)[1].

Closed-loop performance

For each m-step or SBM MPC scheme, we perform 10^3 simulations of the plant evolution of different initial values (using a set of random and uniformly distributed feasible initial state values) and evaluate the closed-loop performance function (7.5). These values are then averaged and assigned to the scheme.

Figure 7.2 shows the trend of the performance of the algorithm along increasing sampling frequency f_s for varying multistep m both for m-step and SBM MPC. Note first that from the discretization of (7.3) using sampling frequency $f_s \in [300\text{kHz}, 400\text{kHz}]$ resulting in sampling time length T_s of magnitude 10^{-6} seconds, the entries of the resulting submatrices M, P, Q, R in (7.5) have magnitude 10^{-6}. With the prescribed state and control constraints of magnitudes 10^0 and 10^{-1}, respectively, we expect J^{cl} to be not far from magnitude 10^{-6} as confirmed by the figure. In addition, the differences between values of J^{cl} ranging from 1.382×10^{-5} to 1.402×10^{-5} can be, in this case, considered significant.

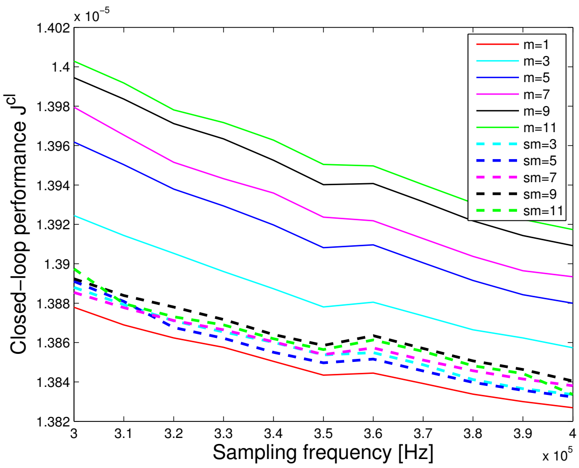

Figure 7.2: Performance J^{cl} for varying sampling frequency f_s. The symbol m stands for the number of steps of the m-step MPC while sm for the SBM MPC.

Observe that the scheme with $m = 1$ gives the standard MPC where we solve an OCP at every sampling instant. As expected, this gives the best performance where the feedback is able to react to the disturbance at each time step. Also shown is that higher sampling frequency yields better closed-loop performance since faster reaction implies faster disturbance rejection.

Furthermore, the closed-loop performance worsens upon using higher value of m (in solid lines). This is as expected since the system runs in open loop for a longer time causing further propagation of the deviation between the measured and the predicted states. However, improvement is achieved through the use of the sensitivity updates. Unlike the m-step feedback law, SBM MPC uses the perturbation magnitude and the sensitivity information to allow the controller to react to this measured and predicted state deviation. As seen in Figure 7.2 (in

[1]As opposed to FLOPS which means floating-point operations per second

dashed lines), the performance profiles get closer to that of the standard MPC indicating better closed-loop performance for the SBM MPC in comparison to the m-step MPC. The graph, however, gives little information to determine which number of steps sm gives for the best performance.

Computational complexity

We present here the details of the *computational complexity* of the active set method for solving a QP which is quantified by the number of FLOPs (i.e., addition, multiplication and division) to be executed per iteration as investigated in [44]. We quantify the number of FLOPs it takes for a fixed simulation time T_{sim} and aim to compare the number of FLOPs we save by increasing the multistep m and the additional operations we incur when updating the controls through computing sensitivities.

First, we consider the active set algorithm for solving a QP in each iteration of the SQP strategy as outlined in Algorithm 7.2.1 (given in [44]). The method begins with an initial guess $\mathcal{W}_0$ of the active set which is called the *working set*. The working set is then refined by adding or deleting a constraint until the exact active set is found.

Algorithm 7.2.1. (Active Set Method for Solving a QP)

Compute a feasible point z_0.
Set initial working set $\mathcal{W}_0$.
For $k = 0, 1, 2, \ldots$
    Solve the system $\begin{bmatrix} H & \nabla C_{\mathcal{W}_k} \\ \nabla C_{\mathcal{W}_k}^\top & 0 \end{bmatrix} \begin{bmatrix} \Delta z_k \\ \eta \end{bmatrix} = \begin{bmatrix} -f - Hz_k \\ 0 \end{bmatrix}$
    If $\Delta z_k = 0$, **then**
        If all $\eta_i \geq 0$, **then**
            Terminate, $z^* = z_k$.
        Else
            Remove i from $\mathcal{W}_k$ s.t. $\lambda_i = \min_{i \in \mathcal{W}_k} \lambda_i$ and then $z_{k+1} \leftarrow z_k$.
        End if
    Else
        $D_k \leftarrow \left\{ i \notin \mathcal{W}_k \mid \nabla C_i \Delta z_k > 0,\ \frac{r_i - \nabla C_i z_k}{\nabla C_i \Delta z_k} < 1 \right\}$
        If $D_k = \emptyset$, **then**
            $z_{k+1} \leftarrow z_k + \Delta z_k$ and $\mathcal{W}_{k+1} \leftarrow \mathcal{W}_k$
        Else
            $\alpha \leftarrow \min_{i \in D_k} \left\{ \frac{r_i - \nabla C_i z_k}{\nabla C_i \Delta z_k} \right\}$ and $z_{k+1} \leftarrow z_k + \alpha \Delta z_k$
            Construct $\mathcal{W}_{k+1}$ by adding one element of D_k to $\mathcal{W}_k$.
        End if
    End if
End for

We define the following variables

n_x	dimension of the state
n_u	dimension of the control
$n_o = (n_x + n_u)N + n_x$	number of optimization/decision variables
$n_e = n_x(N + 1)$	number of equality constraints
$n_i = 2(n_x + n_u)N$	number of inequality constraints
$n_c = n_e + n_i$	total number of constraints

We first consider the worst-case scenario which pertains to solving the system in line 4 of Algorithm 7.2.1 with the largest possible dimension, i.e., the maximum number of inequality constraints that can become active are active. This equals half of the box constraints in the formulation (7.4) which is $n_i/2$. Let $\xi := n_o + n_e + n_i/2$. Since systems with banded matrices are best solved by Gaussian elimination with pivoting as pointed in [67], we use this technique to solve the system in line 4. It requires the following amount of operations

	number of
$\mathcal{N}_{(\cdot)}(\xi) = (\xi - 1)\xi(\xi + 1)/2$	multiplication
$\mathcal{N}_{(+)}(\xi) = \xi^2(\xi + 1)/2$	addition
$\mathcal{N}_{(\div)}(\xi) = \xi$	division

Let $\mathcal{N}_{\mathrm{GE}}(N)$ be the total number of FLOPs needed to perform Gauss elimination as a function of the discrete time prediction horizon N. As $\xi = \xi(N)$, this is given by

$$\mathcal{N}_{\mathrm{GE}}(N) = \left(\mathcal{N}_{(\cdot)} + \mathcal{N}_{(+)} + \mathcal{N}_{(\div)}\right)(\xi)$$

Let us now estimate the number of operations for Algorithm 7.2.1. The following lines require the corresponding amount of operations

line	multiplication	addition	division
4	$n_0^2 +$ $N_{(\cdot)}(\xi)$	$n_0(n_0 - 1) + n_0 +$ $N_{(+)}(\xi)$	$N_{(\div)}(\xi)$
12	$n_c \cdot 2n_o$	$n_c((n_o - 1) + 1)$ $+ n_c(n_o - 1)$	n_c
16	n_o	n_o	

Therefore, letting $\mathcal{N}_{\mathrm{AS}}(N)$ be the total number of FLOPs performed in a single iteration of the active set method which is a function of the discrete time prediction horizon N, we obtain

$$\mathcal{N}_{\mathrm{AS}}(N) = 2n_o^2 + 2n_o(2n_c + 1) + \mathcal{N}_{\mathrm{GE}}(N)$$

which is a polynomial in N of degree 3 (i.e. $\mathcal{O}(N^3)$).

This allows us to compute the number of operations for an MPC scheme over a simulation period. If we fix prediction horizon T (from which we determine N) and simulation time T_{sim} (from which we determine $\tilde{N}$) and assume $\bar{k}$ is the average number of iterations it takes the active-set method to converge, for the m-step MPC, the FLOPs amount to

$$\frac{\tilde{N}}{m} \cdot \bar{k} \cdot \mathcal{N}_{\mathrm{AS}}(N),$$

while for SBM MPC, $(m \neq 1)$, the FLOPs total to

$$\frac{\tilde{N}}{m} \cdot (\bar{k} \cdot \mathcal{N}_{\mathrm{AS}}(N) + \underbrace{\sum_{i=1}^{m-1} (\mathcal{N}_{\mathrm{GE}}(N-i) + 2n_u(n_x+1))}_{(*)})$$

where $(*)$ is additional the expense due to solving a sequence of linear systems for smaller dimension to compute the required sensitivities.

Figure 7.3 shows the trend in the amount of FLOPs of the algorithm along increasing sampling frequency for varying multistep m both for MF and SBM MPC assuming $\bar{k} = 1$. The standard MPC $(m = 1)$ requires the most number of iterations. The number is divided by m as m increases and additional amount is added if sensitivity updates are performed. Note that Figure 7.3 shows the worst-case scenario FLOPs requirement, i.e., with maximum number of active inequality constraints. In the reality, the number of active constraints is significantly much less than the maximum possible. The SBM MPC requires significantly less computing power compared to standard MPC, but requires more compared to an m-step approach when $m > 1$. In addition, by increasing the sampling frequency f_s, the measured FLOPs increase for any controller. This is related to the discretization step (see Section 7.2.2) in the sense that increasing f_s means increasing the prediction horizon N and therefore the problem size and computational complexity.

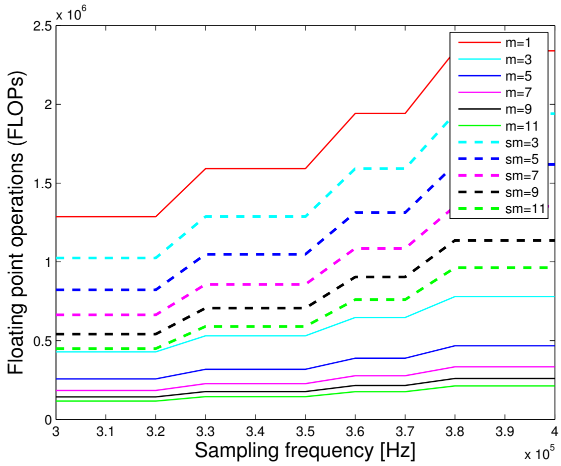

Figure 7.3: Worst case scenario FLOPs for varying sampling frequency f_s and various m-step MPC and sm for the SBM MPC.

In implementing SBM MPC, as mentioned in Section 6.2, one has to take care so as not to violate constraints or create changes in the active constraints when updating by sensitivities. To simplify the analysis on the reduction of cost by taking advantage of available information, we apply the following rule so as not to perform further computations (e.g. the post-optimal analysis in [17] for computing unavailable sensitivities, see Remark 5.5.3 (d)) when constraints are violated. At a given time step, if the control is already on the bound, we do not update in order to keep the corresponding constraint active. Otherwise, if upon updating, the resulting updated control goes on or beyond the constraints, we use a control with a difference of 10^{-6} from the concerned bound in order to keep the corresponding constraint inactive. Similarly, we also prevent the predicted states and perturbed states to go beyond the state constraints. This, however, do not occur in this particular example where perturbation of 5×10^{-3} is used. Figure 7.4 illustrates the state and control staying within the constraints indicated in (7.4) for SBM MPC implementations of varying sampling frequency f_s and multistep m.

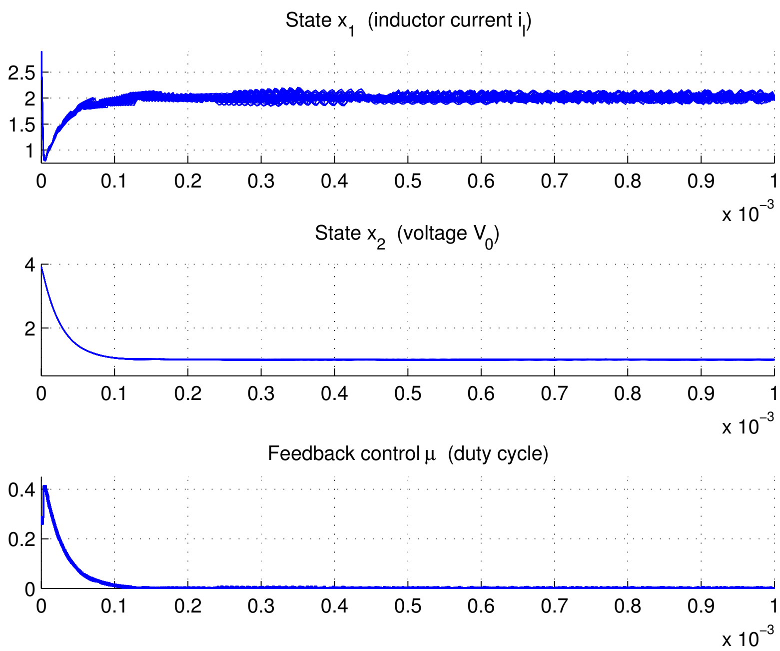

Figure 7.4: Perturbed state and updated control for SBM MPC implementations of varying sampling frequency f_s and multistep m.

Pareto optimality analysis

As shown in Figures 7.2 and 7.3, the closed-loop performance and computing power requirements are strongly correlated: (i) increasing the sampling frequency f_s and decreasing the number of multistep m lead to controllers with lower J^{cl} (i.e., better closed-loop performance) and higher computing power requirement; (ii) similarly, decreasing f_s and using higher multistep m yield controllers with worse closed-loop performance and limited computing power. This results in a design trade-off between closed-loop performance and computing power. We analyze these trade-offs and present them in terms of Pareto optimality and efficiency (for a single point solution) or compromise solutions (see the tutorial in [45]). Figure 7.5 shows the Pareto frontier, thus the design trade-off between closed-loop performance J^{cl} and computing power in terms of FLOPs. On one extreme, the points in red represent the m-step schemes with higher value of m which we observe to be less computationally demanding algorithms, while on the other extreme is the MPC scheme with $m = 1$ which is the one with the highest computing requirements but with the best closed-loop performance (indicated by the lowest J^{cl}). Moreover, the points in blue represent the SBM MPC schemes which we observe to be the algorithms compromising a 'balance' between the two opposing objectives of having a good algorithm performance and being computationally less demanding. This suggests a great potential for the suitability of the scheme for embedded systems with limited computing power.

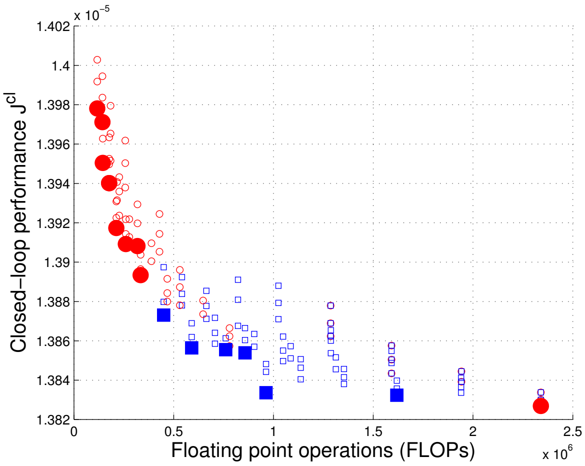

Figure 7.5: A Pareto efficiency plot (solid circles and squares forming the Pareto frontier) on a set of feasible options for m-step (red circles) and SBM (blue squares) MPC.

Bibliography

[1] Alamir, M., and Bornard, G. Stability of a truncated infinite constrained receding horizon scheme: the general discrete nonlinear case. *Automatica 31*, 9 (1995), 1353–1356.

[2] Alt, W., and Malanowski, K. The Lagrange-Newton method for nonlinear optimal control problems. *Computational Optimization and Applications 2*, 1 (1993), 77–100.

[3] Altmüller, N., Grüne, L., and Worthmann, K. Performance of NMPC schemes without stabilizing terminal constraints. In *Recent Advances in Optimization and its Applications in Engineering*, M. Diehl, F. Glineur, E. Jarlebring, and W. Michiels, Eds. Springer, 2010, pp. 289–298.

[4] Aström, K. J., and Murray, R. M. *Feedback systems: an introduction for scientists and engineers.* Princeton University Press, 2010.

[5] Bellman, R. *Dynamic Programming.* Princeton University Press, Princeton, 1957.

[6] Beltracchi, T., and Gabriele, G. *An investigation of new methods for estimating parameter sensitivities.* NASA contractor report. National Aeronautics and Space Administration, Office of Management, Scientific and Technical Information Division, 1989.

[7] Bemporad, A., and Morari, M. Robust model predictive control: A survey. In *Robustness in identification and control.* Springer London, 1999, pp. 207–226.

[8] Bertsekas, D. P. *Dynamic programming and optimal control*, 2nd ed., vol. II. Athena Scientific, 2001.

[9] Bertsekas, D. P. *Dynamic programming and optimal control*, 3rd ed., vol. I. Athena Scientific, 2005.

[10] Biegler, L. T. Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. *Computers & chemical engineering 8*, 3 (1984), 243–247.

[11] Biegler, L. T. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes.* MOS-SIAM Series on Optimization. SIAM, 2010.

[12] Binder, T., Blank, L., Bock, H., Bulirsch, R., Dahmen, W., Diehl, M., Kronseder, T., Marquardt, W., Schlöder, J., and v. Stryk, O. Introduction to model based optimization of chemical processes on moving horizons. In *Online Optimization of Large Scale Systems*, M. Grötschel, S. O. Krumke, and J. Rambau, Eds. Springer Berlin Heidelberg, 2001, pp. 295–339.

[13] Bock, H., Diehl, M., Kostina, E., and Schlöder, J. Constrained optimal feedback control of systems governed by large differential algebraic equations. In *Real-Time and Online PDE-Constrained Optimization*, L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. van Bloemen Waanders, Eds. SIAM, 2007, pp. 3–22.

[14] Bock, H. G., and Plitt, K.-J. A multiple shooting algorithm for direct solution of optimal control problems. In *9th IFAC World Congress Budapest* (1984), Pergamon Press, pp. 243–247.

[15] Boyd, S., and Vandenberghe, L. *Convex Optimization.* Cambridge University Press, 2004.

[16] Büskens, C., and Gerdts, M. Emergency landing of a hypersonic flight system: a corrector iteration method for admissible real-time optimal control approximations. In *Optimalsteuerungsprobleme in der Luft-und Raumfahrt, Workshop in Greifswald des Sonderforschungsbereichs* (2003), vol. 255, pp. 51–60.

[17] Büskens, C., and Maurer, H. Sensitivity analysis and real-time optimization of parametric nonlinear programming problems. In *Online Optimization of Large Scale Systems*, M. Grötschel, S. O. Krumke, and J. Rambau, Eds. Springer Berlin Heidelberg, 2001, pp. 3–16.

[18] Diehl, M. *Real-Time Optimization for Large Scale Nonlinear Processes.* PhD thesis, University of Heidelberg, 2001.

[19] Diehl, M., Bock, H. G., Diedam, H., and Wieber, P.-B. Fast direct multiple shooting algorithms for optimal robot control. In *Fast Motions in Biomechanics and Robotics.* Springer Berlin Heidelberg, 2006, pp. 65–93.

[20] Diehl, M., Bock, H. G., and Schlöder, J. P. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on control and optimization 43*, 5 (2005), 1714–1736.

[21] Diehl, M., Bock, H. G., Schlöder, J. P., Findeisen, R., Nagy, Z., and Allgöwer, F. Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. *Journal of Process Control 12*, 4 (2002), 577–585.

[22] Diehl, M., Uslu, I., Findeisen, R., Schwarzkopf, S., Allgöwer, F., Bock, H. G., Bürner, T., Gilles, E. D., Kienle, A., Schlöder, J. P., et al. Real-time optimization for large scale processes: Nonlinear model predictive control of a high purity distillation column. In *Online Optimization of Large Scale Systems*, M. Grötschel, S. O. Krumke, and J. Rambau, Eds. Springer Berlin Heidelberg, 2001, pp. 363–383.

[23] Fiacco, A. *Introduction to Sensitivity and Stability Analysis in Nonlinear Programming.* Mathematics in Science and Engineering. Academic Press, 1983.

[24] Fiacco, A. V. Sensitivity analysis for nonlinear programming using penalty methods. *Mathematical Programming 10*, 1 (1976), 287–311.

[25] Gerdts, M. Optimal control and real-time optimization of mechanical multi-body systems. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik 83*, 10 (2003), 705–719.

[26] Gerdts, M. *OCPID-DAE1: Optimal Control and Parameter Identification with Differential-Algebraic Equations of Index 1*. Universität der Bundeswehr München, 2011. http://www.optimal-control.de/.

[27] Gerdts, M. *Optimal Control of ODEs and DAEs*. De Gruyter Textbook. De Gruyter, 2012.

[28] Geyer, T., Papafotiou, G., Frasca, R., and Morari, M. Constrained optimal control of the step-down DC-DC converter. *Power Electronics, IEEE Transactions on 23*, 5 (2008), 2454–2464.

[29] Gill, P. E., Golub, G. H., Murray, W. A., and Saunders, M. A. Methods for modifying matrix factorizations. *Mathematics of computation 28*, 126 (1974), 505–535.

[30] Gondzio, J. Stable algorithm for updating dense LU factorization after row or column exchange and row and column addition or deletion. *Optimization 23*, 1 (1992), 7–26.

[31] Grimm, G., Messina, M. J., Tuna, S. E., and Teel, A. R. Model predictive control: for want of a local control Lyapunov function, all is not lost. *Automatic Control, IEEE Transactions on 50*, 5 (2005), 546–558.

[32] Grüne, L. Analysis and design of unconstrained nonlinear MPC schemes for finite and infinite dimensional systems. *SIAM Journal on Control and Optimization 48*, 2 (2009), 1206–1228.

[33] Grüne, L., and Palma, V. G. Robustness of performance and stability for multistep and updated multistep MPC schemes. Discrete and Continuous Dynamical Systems - Series A (Special Issue on New Trends for Optimal Control and Sensitivity Analysis) (2015), to appear.

[34] Grüne, L., and Palma, V. G. On the benefit of re-optimization in optimal control under perturbations. In *21st International Symposium on Mathematical Theory of Networks and Systems (MTNS 2014)* (2014), pp. 439 – 446.

[35] Grüne, L., and Pannek, J. Practical NMPC suboptimality estimates along trajectories. *Sys. & Contr. Lett. 58*, 3 (2009), 161–168.

[36] Grüne, L., and Pannek, J. *Nonlinear Model Predictive Control: Theory and Algorithms*. Communications and Control Engineering. Springer, 2011.

[37] Grüne, L., Pannek, J., Seehafer, M., and Worthmann, K. Analysis of unconstrained nonlinear MPC schemes with time varying control horizon. *SIAM Journal on Control and Optimization 48*, 8 (2010), 4938–4962.

[38] Grüne, L., and Rantzer, A. On the infinite horizon performance of receding horizon controllers. *Automatic Control, IEEE Transactions on 53*, 9 (2008), 2100–2111.

[39] Jadbabaie, A., and Hauser, J. On the stability of receding horizon control with a general terminal cost. *Automatic Control, IEEE Transactions on 50*, 5 (2005), 674–678.

[40] Jerez, J. L., Goulart, P. J., Richter, S., Constantinides, G. A., Kerrigan, E. C., and Morari, M. Embedded online optimization for model predictive control at megahertz rates. *arXiv preprint arXiv:1303.1090* (2013).

[41] Kellett, C. M., Shim, H., and Teel, A. R. Further results on robustness of (possibly discontinuous) sample and hold feedback. *Automatic Control, IEEE Transactions on 49*, 7 (2004), 1081–1089.

[42] Kerrigan, E. C., Jerez, J. L., Longo, S., and Constantinides, G. A. Number representation in predictive control. In *IFAC Conference on Nonlinear Model Predictive Control, Noordwijkerhout, NL* (2012), pp. 60–67.

[43] Khalil, H. *Nonlinear Systems.* Prentice Hall, 2002.

[44] Lau, M., Yue, S., Ling, K., and Maciejowski, J. A comparison of interior point and active set methods for FPGA implementation of model predictive control. In *Proc. European Control Conference* (Budapest, August 2009), European Union Control Association.

[45] Marler, R., and Arora, J. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization 26*, 6 (2004), 369–395.

[46] Maurer, H., and Pesch, H. Solution differentiability for parametric nonlinear control problems with control-state constraints. *Journal of Optimization Theory and Applications 86*, 2 (1995), 285–309.

[47] Mayne, D. Q., Rawlings, J. B., Rao, C. V., and Scokaert, P. O. Constrained model predictive control: Stability and optimality. *Automatica 36*, 6 (2000), 789–814.

[48] Middlebrook, R., and Cuk, S. A general unified approach to modeling switching-converter power stages. *Int. Journal of electronics 42*, 6 (1977), 521–550.

[49] Nocedal, J., and Wright, S. *Numerical Optimization.* Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[50] Palma, V. G., and Grüne, L. Stability, performance and robustness of sensitivity-based multistep feedback NMPC. In *20th International Symposium on Mathematical Theory of Networks and Systems (MTNS 2012)* (2012). Extended Abstract, CD-ROM, Paper No. 68, 4 pages.

[51] Palma, V. G., Suardi, A., and Kerrigan, E. C. Sensitivity-based multistep MPC for embedded systems. Submitted to the 5th IFAC Conference on Nonlinear Model Predictive Control 2015 (NMPC '15).

[52] Pannek, J., Michael, J., and Gerdts, M. A general framework for nonlinear model predictive control with abstract updates. *arXiv preprint arXiv:1309.1610* (2013).

[53] Pesch, H. J. Numerical computation of neighboring optimum feedback control schemes in real-time. *Applied Mathematics and Optimization 5*, 1 (1979), 231–252.

[54] Primbs, J. A., and Nevistić, V. Feasibility and stability of constrained finite receding horizon control. *Automatica 36*, 7 (2000), 965–971.

[55] Qin, S. J., and Badgwell, T. A. An overview of industrial model predictive control technology. *AIChE Symposium Series 93*, 316 (1997), 232–256.

[56] Qin, S. J., and Badgwell, T. A. A survey of industrial model predictive control technology. *Control engineering practice 11*, 7 (2003), 733–764.

[57] Rawlings, J. B., and Mayne, D. Q. *Model predictive control: Theory and design*. Nob Hill Pub., 2009.

[58] Robinson, S. M. Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms. *Mathematical Programming 7*, 1 (1974), 1–16.

[59] Schwarz, H., and Köckler, N. *Numerische Mathematik*. Lehrbuch Mathematik. B.G. Teubner Verlag / GWV Fachverlage GmbH, Wiesbaden (GWV), 2006.

[60] Shamma, J. S., and Xiong, D. Linear nonquadratic optimal control. *Automatic Control, IEEE Transactions on 42*, 6 (1997), 875–879.

[61] Sontag, E. D. Clocks and insensitivity to small measurement errors. *ESAIM: Control, Optimisation and Calculus of Variations 4* (1999), 537–557.

[62] Spellucci, P. *Numerische Verfahren der nichtlinearen Optimierung*. ISNM Lehrbuch. Birkhäuser, 1993.

[63] Suardi, A., Longo, S., Kerrigan, E. C., and Constantinides, G. A. Energy-aware MPC co-design for DC-DC converters. In *2013 European Control Conference (ECC)* (2013), IEEE, pp. 3608–3613.

[64] Sun, W., and Yuan, Y. *Optimization Theory and Methods: Nonlinear Programming*. Springer Optimization and Its Applications. Springer, 2006.

[65] Tuna, S. E., Messina, M. J., and Teel, A. R. Shorter horizons for model predictive control. In *Proceedings of the American Control Conference, Minneapolis, Minnesota, USA* (2006), pp. 863–868.

[66] Worthmann, K. *Stability Analysis of Unconstrained Receding Horizon Control Schemes*. PhD thesis, University of Bayreuth, 2011.

[67] Wright, S. J. Applying new optimization algorithms to model predictive control. In *Chemical Process Control-V, AIChE Symposium Series No. 316*, vol. 93. CACHE Publications, 1997, pp. 147–155.

[68] Würth, L., Hannemann, R., and Marquardt, W. Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization. *Journal of Process Control 19*, 8 (2009), 1277–1288.

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

[70] Zavala, V. M., and Biegler, L. T. The advanced-step NMPC controller: Optimality, stability and robustness. *Automatica 45*, 1 (2009), 86–93.

Ehrenwörtliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die von mir vorgelegte Dissertation mit dem Thema

"Robust Updated MPC Schemes"

selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Zudem erkläre ich, dass

- ich diese Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt habe und
- Hilfe von gewerblichen Promotionsberatern bzw. -vermittlern oder ähnlichen Dienstleistern weder in Anspruch genommen wurde noch künftig in Anspruch genommen wird.

Bayreuth, den 23. Februar 2015

....

Vryan Gil Palma