Master Thesis 47

Distributed Economic Model Predictive Control under inexact minimization with application to Power Systems

Master Thesis

Johannes Köhler

January 25, 2017

Examiner: *Prof. Dr.-Ing. Frank Allgöwer*
Supervisor: *Dr.-Ing. Matthias Müller*
Prof. Na Li, Ph.D.

Institute for Systems Theory and Automatic Control
University of Stuttgart
Prof. Dr.-Ing. Frank Allgöwer

John A. Paulson School of Engineering and Applied Sciences
Harvard University

Abstract

This thesis investigates distributed economic model predictive control (DEMPC) for linear coupled systems under inexact distributed minimization. The theoretical results are applied to solve the real time economic dispatch problem for distributed power systems. The goal is to provide a comprehensive framework for DEMPC with iterative dual algorithms, starting with offline computations, going through online computations and stopping conditions.

This work considers both DEMPC without terminal constraints and DEMPC with terminal costs and sets. Three relevant aspects of DEMPC are covered. The first question is how the optimization problem should be posed, such that desired properties like recursive feasibility and stability can be guaranteed. The main challenge for distributed systems is to facilitate scalable offline computations for the required distributed matrices and sets (e.g. structured stabilizing controller).

The next question is how such an optimization problem can be solved online in a distributed fashion. Here we argue that dual distributed optimization algorithms present a scalable solution and show one algorithm that can be used to solve the DEMPC optimization problem online.

Studying the effects of inexact minimization on DEMPC and corresponding modifications to the optimization problem is the third and most relevant aspect in this work. Most theoretical MPC results do not consider this in great detail. But for dual distributed optimization in combination with real time requirements, such considerations are highly relevant and should not be neglected. Here we present a modification to the optimization problem similar to robust MPC and give guarantees for constraint satisfaction and recursive feasibility despite constraint violations in the optimization. Furthermore, the impact of suboptimality on stability and performance is studied.

By combining these three aspects we get a comprehensive theory on how to use DEMPC. This theory involves the required offline computations, the formulation of the optimization problem, the iterative online solution and corresponding stopping conditions for the iterations.

The final contribution of this work is the application of the derived theory to solve the real time economic dispatch problem. This serves two purposes. On the one side it shows at a practical example, how the theoretical guarantees of constraint satisfaction and stability hold. On the other side this is also intended as a contribution to the power systems community, to show

that superior results can be achieved by considering DEMPC, compared to other state of the art solutions to the economic dispatch problem.

Contents

1 Introduction

1.1 Problem Description

Model predictive control (MPC) is a well established control method, that can handle complex dynamics and hard state and input constraints. The field of distributed MPC (DMPC) in comparison is relatively young and consists of multiple different approaches. The most difficult setup for DMPC, is when both coupling dynamics and coupling constraints are present. Such a setup can be found in power systems. In addition to these challenges the operation of power systems has strict real time requirements. Thus idealizing the underlying distributed optimization algorithm is not an option. Instead, we have to assume inexact solutions and study the implication on the closed-loop operation. Lastly the overarching goal in the operation of power systems is not a mere stabilization, but the improvement of the overall economic performance. This can also be incorporated by using Economic MPC (EMPC). If we combine all these real requirements for power systems, we have a solid motivation to study distributed economic model predictive control (DEMPC) under inexact minimization. With the corresponding theory we can use MPC to improve the economic performance of power systems.

1.2 Contribution

This thesis makes contributions to the theory of DEMPC under inexact minimization. This includes constraint satisfaction, recursive feasibility, stability and performance results for a given constraint violation and suboptimality of a distributed algorithm. In addition, these results are applied to a realistic power system setup to solve the real time economic dispatch problem with DEMPC.

The first set of contributions corresponds to nominal DEMPC results, that do not consider inexact minimization. These are mostly extensions of existing results, that can be used for the offline computations.

For DEMPC with terminal costs and sets, the existing results for stabilizing DMPC [ConteEtAl16] are extended to DEMPC by using the corresponding

EMPC results in [AmritRawlingsAngeli11].
For Robust DMPC there is an existing procedure to compute a distributed RPI set [ConteEtAl13] based on a given stabilizing controller and ellipsoidal bounds on the disturbances. We augment this procedure by providing a distributed optimization to compute a stabilizing controller, that minimizes the resulting constraint tightening. This is done by combining results in [ConteEtAl13] with [LimonEtAl08a] and by using less conservative terminal sets based on [AlvaradoEtAl10].
The main contribution correspond to DEMPC with inexact minimization and considers the closed-loop effect of constraint violations and suboptimality in the optimization.
Here, a novel approach to ensure closed-loop constraint satisfaction and recursive feasibility for DEMPC with terminal sets and costs, despite constraint violations within the optimization problem, is presented. Compared to existing results this does not require online adaptation as in [FerrantiEtAl15] and results in a less conservative tightening than [KögelFindeisen14] by using a different candidate solution. Based on this candidate solution, stability guarantees in dependence of the suboptimality are derived. Performance guarantees for DEMPC with terminal sets and costs under inexact minimization are also derived, which are an extension to [GrünePanin15].
For DEMPC without terminal constraints we only consider the effect of the suboptimality and do not investigate recursive feasibility. For this setup stability and performance guarantees under inexact minimization are derived, which are an extension of [Grüne13,GrüneStieler14].
The last contribution of this thesis involves the application of DEMPC to solve the real time economic dispatch problem. Here DEMPC under inexact minimization is compared to 'classical' power control methods [ZhangLiPapachristodoulou15] in realistic scenarios. This demonstrates the effectiveness of this approach by considering a realistic setup. At the same time this shows that DEMPC can significantly improve the economic performance of power systems.

1.3 Outline

This thesis is structured as follows. In chapter 2, the nominal DEMPC theory and corresponding offline computations are described. Distributed optimization algorithms that can solve the DEMPC optimization problem online are presented in chapter 3. In chapter 4, the effects of inexact minimization on the closed-loop properties of DEMPC are derived. In chapter 5, the DEMPC is applied to the real time economic dispatch problem and the performance is compared to other controllers. Since the topics of the different chapters vary substantially, a corresponding literature review is presented in the beginning of each chapter.

1.4 Notation

In this thesis, the following notation is used.

Graph

In the following, we consider distributed systems, which can be represented as a graph $(\mathcal{V},\mathcal{E})$, with the global state $x \in \mathbb{R}^n$. Each node $i \in \{1,\ldots,M\}$ in the graph corresponds to a subsystem i with corresponding local states $x_i \in \mathbb{R}^{n_i}$. Since the global state x is composed of the local states x_i we have $n = \sum_{i=1}^{M} n_i$. We denote the subset of all nodes, that are connected to subsystem i, as the strict neighborhood $\overline{\mathcal{N}}_i = \{j|(i,j) \in \mathcal{E}\}$. By $\mathcal{N}_i = \mathcal{N}_i \cup \{i\}$ we denote the neighborhood (including i) with the corresponding states $x_{\mathcal{N}_i} \in \mathbb{R}^{|\mathcal{N}_i|n_i}$.

Discrete Distributed Dynamics

In this work we consider the linear discrete time dynamics with neighboring coupling, which can be written as

$$x_i^+ = A_{\mathcal{N}_i} x_{\mathcal{N}_i} + B_i u_i, \quad i = 1,\ldots,M.$$

Here x^+ denotes the state in the next time step.

Lifted Matrices

To simplify the notation when considering local matrices $Q_i \in \mathbb{R}^{n_i \times n_i}$ we denote the lifted matrix by $\overline{Q}_i \in \mathbb{R}^{n \times n}$, which is generated by appending zeroes, i.e.

$$\overline{Q}_i = \begin{pmatrix} 0 & & & & 0 \\ & \ddots & & & \\ & & Q_i & & \\ & & & \ddots & \\ 0 & & & & 0 \end{pmatrix}.$$

The lifted matrix satisfies $x^\top \overline{Q}_i x = x_i^\top Q_i x_i$.

Set Operations

For two sets $S,T \subset \mathbb{R}^n$

$$S \bigoplus T = \{x | \exists s \in S, t \in T : x = s + t\}$$

denotes the Minkowski sum and the Pontryagin difference is denoted by

$$S \bigominus T = \{x | x + t \in S, \forall t \in T\}.$$

We also denote a ball around the origin with radius r as $\mathcal{B}_r = \{x | \|x\| \leq r\}$.

2 Distributed Economic Model Predictive Control - Theory

This chapter describes the theory concerning the used DEMPC schemes. The first section gives a general overview of relevant MPC theory. In the second section we discuss DEMPC with terminal costs and terminal sets. In the third section DEMPC without terminal constraint (i.e. unconstrained DEMPC) is considered. Finally, tube based robust DEMPC is discussed.

2.1 Overview

This section gives a short overview of relevant theory for MPC. A good overview of existing MPC theory can be found in [RawlingsMayne09]. The basic idea of MPC is to solve a control task by posing it as an optimization problem. This stands in contrast to classical control design, in which the controller is expressed explicitly as a dynamical system and desired properties are established based on the interconnected closed-loop dynamics. One of the greatest advantages of MPC is its ability to handle complex nonlinear dynamics and to guarantee constraint satisfaction during transient operation for both state and input constraints.

Standard MPC Optimization Problem

In MPC the control law is only defined implicitly as the solution to an optimization problem. Standard MPC optimizes over a finite horizon with a stage cost, that penalizes the deviation from the desired equilibrium. This can be accomplished with the following optimization problem:

$$\begin{aligned}\min_{u} \; & \sum_{k=0}^{N-1} l(x(k),u(k)) \\ \text{st.} \; & (x(k),u(k)) \in \mathcal{X} \times \mathcal{U} \\ & x(k+1) = f(x(k),u(k)) \\ & x(0) = x\end{aligned}$$

where $l(x,u)$ is the stage cost, $\mathcal{X}, \mathcal{U}$ are general constraint sets, $x(\cdot)$, $u(\cdot)$ are the predicted states and inputs, and $f(x,u)$ represents the system dynamics. This optimization problem is solved in each time step with the current state x and the first part of the resulting optimal input is applied in each time step.

Model Predictive Control Theory

Most of the classical MPC literature is devoted to centralized tracking MPC with a positive definite stage cost with respect to the desired steady state. In centralized MPC, the optimization problem is solved at one central entity and the overall system is driven to the desired steady state (u_s,x_s) while respecting state and input constraints: $(x,u) \in \mathcal{Z}$. Hereby, the optimization is usually regarded ideally as a solved problem and we only consider how to pose the optimization problem, such that the closed-loop system has the desired qualities.
One of the most important issues in MPC is to guarantee that the optimization problem is recursivly feasible, i.e. that the optimizer is always able to find a feasible solution. The other related desired property is that the recursive application of the MPC solution leads to a stable closed-loop system, that does not violate the constraints. This can be guaranteed with different approaches.

Zero-Terminal Constraint MPC

For general nonlinear dynamics and constraints, we can establish recursive feasibility and asymptotic stability by adding a zero-terminal constraint. This demands, that the predicted state at the end of the optimization horizon is the desired steady state, i.e. $x(N) = x_s$.

MPC with terminal cost and terminal set

This terminal constraint can be relaxed by using a terminal set constraint and adding a terminal cost [ChenAllgöwer97]. Here the terminal set and terminal cost need to be chosen, such that the terminal cost is a control Lyapunov function (clf) inside the terminal set, with a corresponding terminal controller.
The typical choice for terminal cost/set MPC is a linear terminal controller $u = Kx$ and a quadratic terminal cost $V_f(x) = x^\top P_f x$, which can be computed with the linear quadratic regulator (LQR) equation, assuming that the linearized system is stabilizable. There are also approaches that do not include the terminal set constraint explicitly in the optimization problem, but increase the terminal cost such that the terminal set constraint is implicitly satisfied.

MPC without terminal constraints

MPC without terminal constraints, i.e. unconstrained MPC, can avoid restrictive terminal constraints, which can deteriorate performance and increase the online computational complexity. An additional advantage of unconstrained MPC is that no complex offline computations are required. For general unconstrained MPC, asymptotic stability can be established with a long enough prediction horizon, if an asymptotic controllability assumption is satisfied [GrünePannek11]. However, this condition is in general hard to verify for most nonlinear systems or only bad estimates can be given.
Other MPC methods include generalized terminal constraints [FagianoTeel12, MüllerAngeliAllgöwer14], reference tracking MPC [LimónEtAl08b] and Lyapunov based MPC [HeidarinejadLiuChristofides12], which are however not considered in this work. Output feedback MPC [MayneEtAl09] is another important issue in MPC theory, that explicitly considers state estimation errors, which is also not included in this work.

Economic Model Predictive Control

Economic MPC (EMPC) can be regarded as a generalization of standard MPC theory [AngeliAmritRawlings12]. Standard MPC theory uses a stage cost $l(x,u)$, that is positive definite with respect to the desired equilibrium (x_s,u_s). Economic MPC does not need this assumption and operates with a general stage cost function that can directly incorporate the economic cost instead of simply tracking the desired steady state. For most standard MPC methods under additional assumptions, similar stability properties and in addition economic performance guarantees can be derived for EMPC . Those include mainly results for unconstrained EMPC [GrüneStieler14] and EMPC with a terminal cost [AmritRawlingsAngeli11].

Robust MPC

Robust MPC is a modification to standard MPC, which ensures that the desired properties still hold under unpredictable, but bounded disturbances. One such modification is tube based robust MPC [LangsonEtAl04]. Here the system trajectory is confined to be in a tube around the nominal predicted trajectory. This is accomplished by adding a controller that keeps the error bound, and tightening the constraints in the optimization. Robust MPC can lead to conservative control action, which is one of the major drawbacks. Alternative robust MPC approaches for EMPC can improve the economic performance [BayerEtAl16], but usually increase the computational demand and are thus not considered here.
In the remainder of this chapter we only consider EMPC with terminal cost and set, unconstrained EMPC and robust tube based MPC.

Distributed Model Predictive Control

Distributed MPC (DMPC) deals with the application of MPC to distributed systems. The idea is that each subsystem has its own MPC, that computes the corresponding input for this subsystem. DMPC in general is a newer field and there are a lot of different approaches for DMPC [MaestreNegenbornothers14, MüllerAllgöwer17]. These approaches differ generally in the kind of settings for which they can be used and the kind of information that needs to be exchanged. This diversity of methods stems from a diversity of applications: from physically independent agents that communicate on the one side, to large coupled plants for which a global optimization is computationally intractable on the other side. The main unifying component in DMPC is that the individual MPCs use locally available information and exchange information with their neighbors. The scope of the shared information varies greatly for different schemes and different control tasks. Here we focus on subsystems that are both physically coupled and have coupling constraints. For such setups there are mainly two different kind of approaches: iterative DMPC and non-iterative DMPC.

Iterative DMPC

In iterative DMPC we consider the global optimization problem as in the centralized case, which means that we can recover centralized performance. The difference is that the optimization problem is solved with iterative distributed optimization as in [StewartWrightRawlings11] or [KögelFindeisen12]. Distributed optimization methods for DMPC are described in detail in chapter 3. In the context of distributed optimization, we can in general not presume that the optimization problem is solved exactly due to the limited time. Thus we have to take inexact optimization into account, which is described in detail in chapter 4.

Non-Iterative DMPC

Non-Iterative DMPC does not attempt to obtain the global optimum, as it requires a large amount of cooperation and thus iterative information exchange. In [FarinaScattolini12] each subsystem adds an additional constraint, ensuring that the optimized state trajectory only changes by a certain amount compared to the initial candidate input. By using robust MPC constraint tightening and treating the change in the neighbor prediction as a disturbance, constraint satisfaction can be guaranteed without iterative communication. In the following, we consider iterative DMPC due to several limitations of non-iterative DMPC, such as difficulty of finding a feasible initialization and the conservative performance.

2.2 Distributed Economic Model Predictive Control with terminal sets and terminal costs

This section discusses Distributed Economic Model Predictive Control (DEMPC) with terminal sets and terminal costs. This approach is less restrictive than zero-terminal constraint MPC, while giving the same theoretical guarantees of recursive feasibility and asymptotic stability. Therefore, a shorter prediction horizon can be used and thus the computational demand can be decreased.
In [ConteEtAl16,ConteEtAl12b], a method to compute distributed terminal

costs and sets for Distributed MPC with quadratic tracking stage cost via distributed LMIs was presented. In [AmritRawlingsAngeli11], a method to compute a terminal cost and set for Economic MPC was presented. Here, the results presented in [ConteEtAl12b] are generalized to a linear-quadratic economic cost, using the results derived in [AmritRawlingsAngeli11].
In the following we consider linear system dynamics

$$x^+ = Ax + Bu$$

and a linear-quadratic economic stage cost

$$l(x,u) = x^\top Qx + x^\top q + u^\top Ru + u^\top r,$$

which can be interpreted as a tracking cost for an unreachable set point with $Q,R \geq 0$. Without loss of generality we further assume that the optimal steady state

$$\begin{aligned}(x_s,u_s) = \arg\min l(x,u)\\ \text{st. } x \in \mathcal{X},\, u \in \mathcal{U}, \quad x = Ax + Bu\end{aligned}$$

is given by $(x_s,u_s) = (0,0)$.
In the first part asymptotic stability of EMPC with terminal set and terminal cost is shown based on [AmritRawlingsAngeli11]. We also show how a suitable terminal cost and terminal set can be computed in the centralized case.
The second part extends these results to distributed systems. In particular a distributed computation of terminal sets and terminal costs is presented, which is an extension of the procedure in [ConteEtAl12b].

2.2.1 Economic Model Predictive Control with terminal set and terminal cost

For the centralized EMPC, all results are based on [AmritRawlingsAngeli11]. In each sampling step the following (E)MPC optimization problem is solved:

$$\begin{aligned}\mathcal{V}^*(x) = \min \sum_{k=0}^{N-1} l(x(k),u(k)) + V_f(x(N)) \qquad (2.1)\\ \text{st. } x(k+1) = Ax(k) + Bu(k), \quad k = 1,..,N\\ u(k) \in \mathcal{U}, \quad x(k) \in \mathcal{X}, \quad k = 1,..,N\\ x(N) \in \mathcal{X}_f, \quad x(0) = x\end{aligned}$$

The MPC feedback $u = \mu(x)$ consists of applying the first part of the optimal input sequence $u^*(0)$, which leads to the following closed-loop dynamics

$$x(t+1) = Ax(t) + B\mu(x).$$

We will use the following assumptions to establish asymptotic stability:

Assumption 1. *The constraint set $\mathcal{X} \times \mathcal{U}$ is compact.*

Assumption 2. *The system $x^+ = Ax + Bu$ is strictly dissipative with respect to the supply rate $s(x,u) = l(x,u) - l(0,0)$, i.e. there exists a storage function λ, and a function $\alpha_l \in \mathcal{K}_\infty$ such that:*

$$\lambda(Ax + Bu) - \lambda(x) \leq -\alpha_l(\|x\|) + l(x,u) - l(0,0). \tag{2.2}$$

Assumption 3. *The storage function λ is continuous on $\mathcal{Z}$.*

Assumption 4. *There exists a terminal set $\mathcal{X}_f \subseteq \mathcal{X}$, containing the origin in its interior, and a control law $u = Kx$ such that:*

$$(A + BK)x \in \mathcal{X}_f, \qquad \forall x \in \mathcal{X}_f \tag{2.3}$$

$$V_f((A + BK)x) \leq V_f(x) - l(x,Kx) + l(0,0), \qquad \forall x \in \mathcal{X}_f \tag{2.4}$$

$$u = Kx \in \mathcal{U}, \qquad \forall x \in \mathcal{X}_f \tag{2.5}$$

Remark 5. *For linear systems with a strictly convex quadratic cost ($Q,R > 0$), there exists a linear storage function, that satisfies assumption 2 and thus also assumption 3 [DiehlAmritRawlings11, DammEtAl14]. Furthermore this assumption implies that the economically optimal operation is steady state operation, thus motivating asymptotic stability of the optimal steady state [MüllerGrüneAllgöwer15].*

The following theorem addresses the stability of the EMPC.

Theorem 6. *([AmritRawlingsAngeli11],Theorem 15)*
Let assumptions 1 -4 hold. If the initial state is feasible, then the EMPC (2.1) is recursivly feasible and the optimal steady state (0,0) is an asymptotically stable equilibrium point for the resulting closed-loop system.

Proof. The stability proof of economic MPC relies on the rotated stage cost $\tilde{l}$, with

$$\tilde{l}(x,u) = l(x,u) + \lambda(x) - \lambda(Ax + Bu) - l(0,0),$$

where λ is the storage function (2.2). Correspondingly, we define the rotated terminal cost $\tilde{V}_f$ and the rotated cost function $\tilde{\mathcal{V}}$ as

$$\tilde{V}_f(x) = V_f(x) + \lambda(x) - V_f(0) - \lambda(0),$$
$$\tilde{\mathcal{V}}(x,\mathbf{u}) = \sum_{k=0}^{N-1} \tilde{l}(x(k),u(k)) + \tilde{V}_f(x(N)).$$

Now we use the candidate Lyapunov function

$$\tilde{\mathcal{V}}^*(x) = \tilde{\mathcal{V}}(x,\mathbf{u}^*(x)),$$

where $\mathbf{u}^*(x) = \{u^*(0;x),u^*(1;x),\dots,u^*(N-1;x)\}$ is the optimal control sequence. We denote $x_{u^*(x)}(k,x)$ as the optimal predicted state trajectory at time step k and the corresponding input as $u^*(k,x)$. Due to the bounded constraint set $\mathcal{X} \times \mathcal{U}$, the continuity of the storage function λ and the positive definiteness of the rotated stage cost $\tilde{l}$, the rotated cost $\tilde{\mathcal{V}}^*$ is upper and lower bounded by a $\mathcal{K}$-function ([RawlingsMayne09], proposition 2.17,2.18) and thus a candidate Lyapunov function.
As a candidate input sequence $\tilde{\mathbf{u}}(x)$ we use the shifted optimal input with the terminal control law appended:

$$\tilde{\mathbf{u}}(x) = \{u^*(1;x),u^*(2;x),\dots,u^*(N-1;x), Kx_{u^*(x)}(N;x)\},$$
$$\tilde{\mathbf{x}} = \{x_{u^*(x)}(1;x),x_{u^*(x)}(2;x),\dots,x_{u^*(x)}(N;x),(A+BK)x_{u^*(x)}(N;x)\}.$$

Due to the terminal constraint we have $x_{u^*(x)}(N;x) \in \mathcal{X}_f$, which implies $(A+BK)x_{u^*(x)}(N;x) \in \mathcal{X}_f$ due to the invariance of the terminal set $\mathcal{X}_f$.
To establish asymptotic stability we need the following auxiliary lemma about the rotated terminal cost:

Lemma 7. *([AmritRawlingsAngeli11] Lemma 9) The following two statements are equivalent:*

Condition (2.4) *in assumption* 4 *holds*

$$\Leftrightarrow$$

$$\tilde{V}_f((A+BK)x) \leq \tilde{V}_f(x) - \tilde{l}(x,Kx), \quad \forall x \in \mathcal{X}_f.$$

Proof. Adding $\lambda((A+BK)x) + \lambda(x)$ on both sides of condition (2.4) yields

$$\begin{aligned}\tilde{V}_f((A+BK)x) - \tilde{V}_f(x) &\leq -(l(x,Kx) + \lambda(x) - \lambda((A+BK)x) - l(0,0)) \\ &= -\tilde{l}(x,Kx).\end{aligned}$$

□

Using the previous definitions with an arbitrary input sequence $u(\cdot)$ and corresponding state sequence $x_u(\cdot)$, we have

$$\begin{aligned}\tilde{\mathcal{V}}(x,u) &= \underbrace{\sum_{k=1}^{N-1} \tilde{l}(x_u(k),u(k))}_{\sum_{k=1}^{N-1} l(x_u(k),u(k))+\lambda(x)-\lambda(x_u(N))} + \underbrace{\tilde{V}_f(x_u(N))}_{V_f(x_u(N)+\lambda(x_u(N))-\lambda(0)} \\ &= \mathcal{V}(x) + \lambda(x) - \lambda(0).\end{aligned}$$

Thus the optimal input u^* minimizes both the original cost $\mathcal{V}(x)$ and the rotated cost $\tilde{\mathcal{V}}(x)$. We abbreviate the MPC feedback as $u^*(0;x) = \mu(x)$ and the next state as $x^+ = Ax + B\mu(x)$. With this the decrease in the candidate Lyapunov function can be shown, using the fact that the optimal input leads to a smaller rotated cost $\tilde{\mathcal{V}}$, than the candidate input sequence $\tilde{u}$:

$$\begin{aligned}\tilde{\mathcal{V}}^*(x^+) &\leq \tilde{\mathcal{V}}(x^+,\tilde{\mathbf{u}}) \\ &= \sum_{k=1}^{N-1} \tilde{l}(x_{u^*(x)}(k;x),u^*(k;x)) \\ &\quad + \tilde{l}(x_{u^*(x)}(N;x),Kx_{u^*(x)}(N;x)) + \tilde{V}_f((A+BK)x_{u^*(x)}(N;x)) \\ &= \tilde{\mathcal{V}}^*(x) - \tilde{l}(x,u^*(0;x)) + \tilde{l}(x_{u^*(x)}(N;x),Kx_{u^*(x)}(N;x)) \\ &\quad - \tilde{V}_f(x_{u^*(x)}(N;x)) + \tilde{V}_f((A+BK)x_{u^*(x)}(N;x)) \\ &\overset{\text{lemma } 7}{\leq} \tilde{\mathcal{V}}^*(x) - \tilde{l}(x,u^*(0;x)).\end{aligned}$$

With this we have a decrease condition on the candidate Lyapunov function to proof asymptotic stability:

$$\tilde{\mathcal{V}}^*(x^+) - \tilde{\mathcal{V}}^*(x) \leq -\tilde{l}(x,\mu(x)) \leq -\alpha_l(\|x\|) < 0 \quad \forall x \neq 0.$$

The last inequality follows from the strict dissipativity in assumption 2. By standard Lyapunov arguments, the closed-loop state trajectory $x_\mu(k,x)$ under the MPC feedback μ satisfies $x_\mu(k,x) \leq \beta(\|x\|,k)$ with $\beta \in \mathcal{KL}$ and the system is asymptotically stable. □

This EMPC scheme also has economic performance guarantees, which are discussed in section 4.4.

Compute Terminal Cost - Discrete Linear Quadratic Regulator

The computation of a terminal set $\mathcal{X}_f$, terminal cost $V_f(x)$ and a terminal controller K that satisfy assumption 4 can be accomplished by computing the discrete linear quadratic regulator (DLQR), or by solving a semi-definite program (SDP). Due to the linear-quadratic stage cost, we use a linear-quadratic approach for the terminal cost

$$V_f(x) = x^\top P_f x + x^\top p_f.$$

The terminal cost and controller then need to satisfy the following inequality:

$$\begin{aligned} &x^\top (A+BK)^\top P(A+BK)x + p^\top (A+BK)x - x^\top Px - p\top x \\ &\leq -x^\top Qx - x^\top K^\top RKx - q^\top x - r^\top Kx. \end{aligned}$$

Since this inequality should hold for all $x \in \mathcal{X}_f$, the linear part needs to be zero. This can be shown with a proof by contradiction: Assume the i-th component of the linear vector has a non-zero component, pick a vector x with $x_i = \epsilon$ and $x_j = 0$, $j \neq i$ and let $\epsilon \to 0$. Then the linear part exceeds any quadratic terms and the inequality does not hold. By eliminating the linear part, we get the usual tracking MPC condition for the quadratic terms with an additional equality constraint for the linear terms:

$$\begin{aligned} (A+BK)^\top P_f (A+BK) - P_f &\leq -Q - K^\top RK, \\ p_f^\top (A+BK-I) &= -q^\top - r^\top K. \end{aligned}$$

The solution to the discrete linear quadratic regulators (DLQR) K, P_f fulfills the first inequality with equality. The linear part p_f can then be computed as

$$p_f^\top = -(q^\top + r^\top K)(A+BK-I)^{-1}. \tag{2.6}$$

Since $(A+BK)$ is Hurwitz and has eigenvalues with a magnitude smaller one, $(A+BK-I)$ is invertible. The economic stage cost can be interpreted as a tracking cost for unreachable set points, for which in [RawlingsEtAl08] this terminal cost has been derived.

Compute Terminal Cost - LMI approach

A more general approach to compute a terminal cost is to pose the first inequality as a linear matrix inequality (LMI), [BoydEtAl94]. This way

additional criteria, for example maximizing the volume of the terminal set, can be optimized.
For the quadratic part we have the following inequality

$$(A+BK)^\top P(A+BK) - P \leq -Q - K^\top RK.$$

Defining $E = P^{-1}$ and multiplying left and right with E we get

$$E(A+BK)^\top E^{-1}(A+BK)E - E \leq -EQE - EK^\top RKE.$$

Now defining $Y = KE$ and moving all terms to one side we get

$$E - (AE+BY)^\top E^{-1}(AE+BY) - EQE - Y^\top RY \geq 0.$$

To get a LMI condition in the optimization variables (E,Y) we rewrite this inequality as

$$E - \begin{pmatrix} AE+BY \\ Q^{1/2}E \\ R^{1/2}Y \end{pmatrix}^\top \begin{pmatrix} E & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix}^{-1} \begin{pmatrix} AE+BY \\ Q^{1/2}E \\ R^{1/2}Y \end{pmatrix} \geq 0$$

and use the Schur complement to get

$$\begin{pmatrix} E & EA^\top + Y^\top B^\top & EQ^{1/2} & Y^\top R^{1/2} \\ AE+BY & E & 0 & 0 \\ Q^{1/2}E & 0 & I & 0 \\ R^{1/2}Y & 0 & 0 & I \end{pmatrix} \geq 0.$$

With this we can minimize $-\log\det E$, subject to this LMI to compute a small terminal cost. The computation of p_f does not change, since it does not change the size and therefore does not need to be included in the optimization criterion.

Compute Terminal Set

For the terminal set we use $\mathcal{X}_f = \{x | x^\top P_f x \leq \alpha\}$, instead of the level sets of the terminal cost $V_f(x)$, since they are not necessarily centered around the origin. Note that this set satisfies the invariance condition (2.3) due to the quadratic inequality and since $R,Q \geq 0$. Once P_f,K are computed, we need to compute the scalar parameter α to determine the size of the terminal set, such that it satisfies the input constraints and state constraints

$$Kx \in \mathcal{U} \quad \forall x \in \mathcal{X}_f, \quad \mathcal{X}_f \subseteq \mathcal{X}.$$

We consider polytopic state and input constraints

$$\mathcal{X} = \{x|Hx \leq h\}, \quad \mathcal{U} = \{u|Lu \leq l\}.$$

With these constraints, the maximization of the terminal set that satisfies these constraints can be posed as a linear program (LP).

Lemma 8. *([ConteEtAl12b]) The size $\alpha_{\max}$ of the terminal set $\mathcal{X}_f = \{x|x^\top P_f x \leq \alpha_{\max}\}$ that satisfies the input and state constraints in assumption 4, i.e. $\mathcal{X}_f \subseteq \mathcal{X}, K\mathcal{X}_f \subseteq \mathcal{U}$, can be computed with the following linear program.*

$$\begin{aligned} \alpha_{\max} = \max_{\alpha} \ & \alpha && (2.7) \\ st. \ & |P_f^{-1/2} H_i^\top|^2 \alpha \leq h_i^2, && i = 1 \dots n_x \\ & |P_f^{-1/2} K^\top L_i^\top|^2 \alpha \leq l_i^2, && i = 1 \dots n_u \end{aligned}$$

Proof. Without loss of generality we focus on the state constraints:

$$\forall x : x^\top P_f x \leq \alpha \Rightarrow H_i x \leq h_i, \quad i = 1 \dots n_x.$$

Substituting $x = P_f^{-1/2} \tilde{x}$ we get:

$$\tilde{x}^\top \tilde{x} \leq \alpha \Rightarrow H_i P_f^{-1/2} \tilde{x} \leq h_i, \quad i = 1 \dots n_x.$$

Using the symmetry in the terminal set, the squared inequality is equivalent:

$$\tilde{x}^\top \tilde{x} \leq \alpha \Rightarrow |H_i P_f^{-1/2} \tilde{x}|^2 \leq h_i^2, \quad i = 1 \dots n_x.$$

The left side of the equation can then be rewritten as:

$$|H_i P_f^{-1/2} \tilde{x}|^2 \leq |H_i P_f^{-1/2}|^2 |\tilde{x}|_2^2 \leq |P_f^{-1/2} H_i^\top|^2 \alpha.$$

The condition for the input constraints can be derived equivalently, starting from

$$\forall x : x^\top P_f x \leq \alpha \Rightarrow L_i K x \leq l_i, \quad i = 1 \dots n_u.$$

□

With this we can pose the computation of the terminal cost, terminal controller and terminal set as optimization problems.

2.2.2 Distributed Economic Model Predictive Control with terminal sets and costs

Now we extend these results to the distributed case. The stability result of theorem 6 is not directly impacted by transitioning to a distributed setup. The challenge is to construct a terminal cost V_f, terminal controller K and terminal set $\mathcal{X}_f$, that satisfy assumption 4 in the distributed setting.
The change compared to the central setup is twofold. To solve the MPC optimization problem online in a distributed setup, we use distributed optimization methods based on dual decomposition (see chapter 3). This requires a distributed structure of the optimization problem, which in turn requires a distributed structure for the terminal cost V_f, the terminal set $\mathcal{X}_f$ and the terminal controller K. The other difficulty for distributed systems is that the offline computation procedures must be posed in such a way, that they can also be solved by distributed optimization. This ensures, that these computations can still be carried out for large scale systems without a central unit. For the standard distributed MPC, these challenges have been addressed in [ConteEtAl12b] and here these results are extended to distributed economic MPC, with linear-quadratic stage cost.

Distributed Setup

For the distributed setup we assume a decomposable stage cost, consisting of local linear quadratic stage costs:

$$\begin{aligned} l(x,u) &= \sum_{i=1}^{M} l_i(x_i,u_i) = \sum_{i=1}^{M} x_i^\top Q_i x_i + x_i^\top q_i + u_i^\top R_i u_i + u_i^\top r_i \\ &= x^\top Q x + x^\top q + u^\top R u + u^\top r, \end{aligned}$$

where x_i,u_i are the local states and input variables of subsystem i. We also have a linear distributed system dynamics

$$x_i^+ = A_{\mathcal{N}_i} x_{\mathcal{N}_i} + B_i u_i.$$

Hence for the terminal cost we use a sum of local terminal costs with linear and quadratic terms:

$$V_f(x) = \sum_{i=1}^{M} V_{f_i}(x_i) = \sum_{i=1}^{M} x_i^\top P_{f_i} x_i + x_i^\top p_{f_i} = x^\top P_f x + x^\top p_f.$$

For the terminal controller K we also impose a distributed structure, where the local input only depends on the neighboring states $x_{\mathcal{N}_i}$, i.e.

$$u_i = K_{\mathcal{N}_i} x_{\mathcal{N}_i}.$$

Sufficient distributed LMI conditions

With this approach, the task is to specify a distributed optimization problem, that can be used to compute the corresponding matrices and vectors, such that (2.4) in assumption 4 is satisfied. First, we plug in the ansatz, to derive sufficient local inequalities.

Lemma 9. *If the following two conditions are satisfied*

$$V_{f_i}(x_i^+) - V_{f_i}(x_i) \leq -l_i(x_i,u_i) + l_i(0,0) + \gamma_i(x_{\mathcal{N}_i}), \quad i = 1,\ldots,M \tag{2.8}$$

$$\sum_{i=1}^{M} \gamma_i(x_{\mathcal{N}_i}) \leq 0 \tag{2.9}$$

with $x_i^+ = (A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i}) x_{\mathcal{N}_i}$, *then the Lyapunov decrease condition* (2.4) *in assumption* 4 *is satisfied.*

Proof.

$$V_f((A+BK)x) - V_f(x) = \sum_{i=1}^{M} V_{f_i}(x_i^+) - V_{f_i}(x_i)$$
$$\leq \sum_{i=1}^{M} -l_i(x_i,u_i) + l_i(0,0) + \gamma_i(x_{\mathcal{N}_i}) = -l(x,u) + l(0,0) + \sum_{i=1}^{M} \gamma_i(x_{\mathcal{N}_i})$$
$$\leq -l(x,u) + l(0,0).$$

□

Since we have linear-quadratic stage cost and terminal cost, we make a linear-quadratic approach for γ_i:

$$\gamma_i(x_{\mathcal{N}_i}) = x_{\mathcal{N}_i}^\top \Gamma_{\mathcal{N}_i} x_{\mathcal{N}_i} + x_{\mathcal{N}_i}^\top \gamma_{\mathcal{N}_i}.$$

For notational purposes we introduce lifting matrices that pick subsystem values of the overall system and lift subsystem values to the global dimension, such that

$$x_{\mathcal{N}_i} = W_i x, \quad W_i \in \{0,1\}^{n_{\mathcal{N}_i} \times n}.$$

In addition, we denote lifted matrices by overlining them, i.e. $\overline{E}_{\mathcal{N}_i} = W_i^\top E_i W_i$. Given these local inequalities, we derive sufficient LMI conditions, that can be used to compute the corresponding terminal costs and controllers with distributed optimization. The following lemma is an extended version of theorem IV.3 in [ConteEtAl12b].

Lemma 10. *The conditions in lemma 9 are equivalent to the following set of LMIs and equality constraints:*

$$\begin{pmatrix} \overline{E}_i + F_{\mathcal{N}_i} & E_{\mathcal{N}_i} A_{\mathcal{N}_i}^\top + Y_{\mathcal{N}_i}^\top B_i^\top & E_{\mathcal{N}_i} \overline{Q}_i^{1/2} & Y_{\mathcal{N}_i}^\top R_i^{1/2} \\ A_{\mathcal{N}_i} E_{\mathcal{N}_i} + B_i Y_{\mathcal{N}_i} & E_i & 0 & 0 \\ \overline{Q}_i^{1/2} E_{\mathcal{N}_i} & 0 & I & 0 \\ R_i^{1/2} Y_{\mathcal{N}_i} & 0 & 0 & I \end{pmatrix} \geq 0 \quad (2.10)$$

$$p_{f_i}^\top (A_{\mathcal{N}_i} E_{\mathcal{N}_i} + B_i Y_{\mathcal{N}_i}) - \overline{p}_{f_i} E_{\mathcal{N}_i} + \overline{q}_i^\top E_{\mathcal{N}_i} + r_i Y_{\mathcal{N}_i} - \gamma_{\mathcal{N}_i} E_{\mathcal{N}_i} = 0 \quad (2.11)$$

$$i = \{1,...,M\}$$

$$\sum_{i=1}^{M} W_i^\top F_{\mathcal{N}_i} W_i \leq 0 \quad (2.12)$$

$$\sum_{i=1}^{M} W_i \gamma_{\mathcal{N}_i}^\top = 0 \quad (2.13)$$

with $E_i = P_{f_i}^{-1}$, $E = P_f^{-1}$, $F_{\mathcal{N}_i} = E_{\mathcal{N}_i} \Gamma_{\mathcal{N}_i} E_{\mathcal{N}_i}$ *and* $Y_{\mathcal{N}_i} = K_{\mathcal{N}_i} E_{\mathcal{N}_i}$

Proof. The proof consists of two parts :
Part 1 shows that condition (2.8) is equivalent to the two conditions (2.10),(2.11).
Part 2 shows that condition (2.9) is equivalent to the two conditions (2.12),(2.13).
This is done by separating the linear and quadratic terms and rewriting the inequality conditions as LMIs.
Part 1: Show that condition (2.8) is equivalent to the two conditions (2.10) and (2.11). Starting with (2.8) we have

$$V_{f_i}(x_i^+) - V_{f_i}(x_i) \leq -l_i(x_i,u_i) + l_i(0,0) + \gamma_i(x_{\mathcal{N}_i}).$$

Inserting the linear system dynamics, the stage cost and the terminal cost results in

$$\begin{aligned} & x_{\mathcal{N}_i}^\top (A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i})^\top P_{f_i} (A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i}) x_{\mathcal{N}_i} - x_{\mathcal{N}_i}^\top \overline{P}_{f_i} x_{\mathcal{N}_i} \\ & + p_{f_i}^\top (A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i}) x_{\mathcal{N}_i} - \overline{p}_{f_i}^\top x_{\mathcal{N}_i} \\ \leq & - x_{\mathcal{N}_i}^\top (\overline{Q}_i + K_{\mathcal{N}_i}^\top R_i K_{\mathcal{N}_i}) x_{\mathcal{N}_i} - (\overline{q}_i^\top + r_i^\top K_{\mathcal{N}_i}) x_{\mathcal{N}_i} + x_{\mathcal{N}_i}^\top \Gamma_{\mathcal{N}_i} x_{\mathcal{N}_i} + \gamma_{\mathcal{N}_i}^\top x_{\mathcal{N}_i}, \end{aligned}$$

with $x_i^\top P_{f_i} x_i = x_{\mathcal{N}_i}^\top \overline{P}_{f_i} x_{\mathcal{N}_i}$, $p_{f_i}^\top x_i = \overline{p}_{f_i}^\top x_{\mathcal{N}_i}$, $x_i^\top Q_i x_i = x_{\mathcal{N}_i}^\top \overline{Q}_i x_{\mathcal{N}_i}$ and $q_i^\top x_i = \overline{q}_i^\top x_{\mathcal{N}_i}$. Analogous to the centralized EMPC case the linear terms need to be zero, to ensure that this condition holds for all x. This yields the following two conditions:

$$(A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i})^\top P_{f_i} (A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i}) - \overline{P}_{f_i} \leq -(\overline{Q}_i + K_{\mathcal{N}_i}^\top R_i K_{\mathcal{N}_i}) + \Gamma_{\mathcal{N}_i}$$
$$p_{f_i}^\top (A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i}) - \overline{p}_{f_i}^\top + \overline{q}_i^\top + r_i K_{\mathcal{N}_i} - \gamma_{\mathcal{N}_i} = 0.$$

By multiplying the second condition from the right with $E_{\mathcal{N}_i}$, we get (2.11). Now we have to transform the matrix inequality condition into the LMI (2.10). By multiplying the matrix inequality with $E_{\mathcal{N}_i}$ from left and right we get:

$$\begin{aligned} E_{\mathcal{N}_i}(A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i})^\top P_{f_i} (A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i}) E_{\mathcal{N}_i} - E_{\mathcal{N}_i} \overline{P}_{f_i} E_{\mathcal{N}_i} \\ \leq -E_{\mathcal{N}_i}(\overline{Q}_i + K_{\mathcal{N}_i}^\top R_i K_{\mathcal{N}_i}) E_{\mathcal{N}_i} + E_{\mathcal{N}_i} \Gamma_{\mathcal{N}_i} E_{\mathcal{N}_i} \end{aligned}$$

Defining $F_{\mathcal{N}_i} = E_{\mathcal{N}_i} \Gamma_{\mathcal{N}_i} E_{\mathcal{N}_i}$ and $Y_{\mathcal{N}_i} = K_{\mathcal{N}_i} E_{\mathcal{N}_i}$ we get

$$\begin{aligned} -(E_{\mathcal{N}_i} A_{\mathcal{N}_i}^\top + Y_{\mathcal{N}_i}^\top B_i^\top) E_i^{-1} (E_{\mathcal{N}_i} A_{\mathcal{N}_i}^\top + Y_{\mathcal{N}_i}^\top B_i^\top) + \overline{E}_i + F_{\mathcal{N}_i} \\ -(E_{\mathcal{N}_i} \overline{Q}_i E_{\mathcal{N}_i} + Y_{\mathcal{N}_i}^\top R_i Y_{\mathcal{N}_i}) \geq 0. \end{aligned}$$

Due to the block diagonal structure of P_f/E the variables Y, F maintain the distributed structure. This inequality can be rewritten as

$$\overline{E}_i + F_{\mathcal{N}_i} - \begin{pmatrix} A_{\mathcal{N}_i} E_{\mathcal{N}_i} + B_i Y_{\mathcal{N}_i} \\ \overline{Q}_i^{1/2} E_{\mathcal{N}_i} \\ R_i^{1/2} Y_{\mathcal{N}_i} \end{pmatrix}^\top \begin{pmatrix} E_i & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix}^{-1} \begin{pmatrix} A_{\mathcal{N}_i} E_{\mathcal{N}_i} + B_i Y_{\mathcal{N}_i} \\ \overline{Q}_i^{1/2} E_{\mathcal{N}_i} \\ R_i^{1/2} Y_{\mathcal{N}_i} \end{pmatrix} \geq 0.$$

By applying the Schur complement, we get (2.10):

$$\begin{pmatrix} \overline{E}_i + F_{\mathcal{N}_i} & E_{\mathcal{N}_i} A_{\mathcal{N}_i}^\top + Y_{\mathcal{N}_i}^\top B_i^\top & E_{\mathcal{N}_i} \overline{Q}_i^{1/2} & Y_{\mathcal{N}_i}^\top R_i^{1/2} \\ A_{\mathcal{N}_i} E_{\mathcal{N}_i} + B_i Y_{\mathcal{N}_i} & E_i & 0 & 0 \\ \overline{Q}_i^{1/2} E_{\mathcal{N}_i} & 0 & I & 0 \\ R_i^{1/2} Y_{\mathcal{N}_i} & 0 & 0 & I \end{pmatrix} \geq 0.$$

Part 2: Show that condition (2.9) is equivalent to the two conditions (2.12) and (2.13). Starting with (2.9) we have:

$$\sum_{i=1}^{M} \gamma_i(x_{\mathcal{N}_i}) \leq 0.$$

Inserting the linear and quadratic approach we get:

$$\sum_{i=1}^{M} x_{\mathcal{N}_i}^{\top} \Gamma_{\mathcal{N}_i} x_{\mathcal{N}_i} + \gamma_{\mathcal{N}_i}^{\top} x_{\mathcal{N}_i} \leq 0.$$

To ensure that this condition holds for all x, the linear terms need to be zero, which yields the two conditions:

$$\sum_{i=1}^{M} W_i^{\top} \Gamma_{\mathcal{N}_i} W_i \leq 0,$$
$$\sum_{i=1}^{M} \gamma_{\mathcal{N}_i}^{\top} W_i = 0.$$

The linear part already equals (2.13). To show equivalence to (2.12) we need to make a transformation:

$$\begin{aligned} & \sum_{i=1}^{M} W_i^{\top} \Gamma_{\mathcal{N}_i} W_i \leq 0 \\ \Leftrightarrow E(\sum_{i=1}^{M} W_i^{\top} \Gamma_{\mathcal{N}_i} W_i) E = & \sum_{i=1}^{M} E W_i^{\top} \Gamma_{\mathcal{N}_i} W_i E \leq 0 \\ \Leftrightarrow \sum_{i=1}^{M} W_i^{\top} E_{\mathcal{N}_i} \Gamma_{\mathcal{N}_i} E_{\mathcal{N}_i} W_i = & \sum_{i=1}^{M} W_i^{\top} F_{\mathcal{N}_i} W_i \leq 0. \end{aligned}$$

□

Remark 11. *The LMI conditions correspond to the quadratic terms and the equality constraints correspond to the linear terms. Due to (2.11) the conditions are not linear in the optimization variables (E_i,$F_{\mathcal{N}_i}$,Y_i,p_{f_i},$\gamma_{\mathcal{N}_i}$). Therefore we propose finding E_i,$F_{\mathcal{N}_i}$,Y_i that satisfy condition (2.10) and (2.12), and then with these variables fixed computing p_{f_i},$\gamma_{\mathcal{N}_i}$ with equation (2.11) and (2.13). Then the whole computation consists mainly of the procedure in [ConteEtAl12b] with an additional step to compute the linear terms p_{f_i},$\gamma_{\mathcal{N}_i}$. Since p_{f_i} and $\gamma_{\mathcal{N}_i}$ do not influence the shape of the terminal set, they also are not required in the objective function. A typical choice for the cost function would be $-\sum_{i=1}^{M} \det(\log(E_i))$, which leads to a small terminal cost. This computation can be achieved by using distributed semi-definite programming, see for example [PakazadEtAl15].*

It is important to note that p_{f_i},$\gamma_{\mathcal{N}_i}$ that satisfy condition (2.11) and (2.13) always exist. By stacking the equality constraints, we can see that p_f needs to satisfy:

$$p_f^\top (A + BK - I) + q + rK = 0.$$

This is the same condition we derived for the centralized case. Since $A + BK$ is asymptotically stable and has eigenvalues with a magnitude smaller than one, $(A + BK - I)$ is invertible and thus there exists a p_f that satisfies the equality constraint. Correspondingly $\gamma_{\mathcal{N}_i}$ is given by:

$$\gamma_{\mathcal{N}_i} = p_{f_i}^\top (A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i}) - \overline{p}_{f_i}^\top + \overline{q}_i^\top + r_i K_{\mathcal{N}_i}.$$

Compute Terminal Sets

Once the conditions in lemma 10 are fulfilled, a scalar parameter $\alpha_{\max}$ needs to be determined such that the terminal set $\mathcal{X}_f = \{x | x^\top P_f x \leq \alpha_{\max}\}$ satisfies the input and states constraints:

$$\begin{aligned} K_{\mathcal{N}_i} x_{\mathcal{N}_i} &\in \mathcal{U}_i & \forall x \in \mathcal{X}_f \\ x_{\mathcal{N}_i} &\in \mathcal{X}_{\mathcal{N}_i} & \forall x \in \mathcal{X}_f \end{aligned}$$

For this we assume local polytopic constraints:

$$\begin{aligned} \mathcal{X}_{\mathcal{N}_i} &= \{x_{\mathcal{N}_i} | H_{\mathcal{N}_i} x_{\mathcal{N}_i} \leq h_{\mathcal{N}_i}\}, \\ \mathcal{U}_i &= \{u_i | L_i u_i \leq l_i\}. \end{aligned}$$

The maximization of the terminal set, that satisfies these constraints can be posed as the following linear program (LP)

$$\begin{aligned} \alpha_{\max} = \max_{\alpha} \alpha & \\ \text{st. } |P_{f,\mathcal{N}_i}^{-1/2} H_{\mathcal{N}_i,j}^\top|_2^2 \alpha \leq h_{\mathcal{N}_i,j}^2, \quad & i = 1, \dots, M \quad j = 1, \dots, n_{x_{\mathcal{N}_i}} \\ |P_{f,\mathcal{N}_i}^{-1/2} K_{\mathcal{N}_i}^\top L_{i,j}^\top|_2^2 \alpha \leq l_{i,j}^2, \quad & i = 1, \dots, M \quad j = 1, \dots, n_{u_i} \end{aligned} \tag{2.14}$$

The derivation is equivalent to the central case in lemma 8. With $\alpha_{\max}$ the size of the overall terminal set is given. The structure of this LP allows for a distributed solution.

Time varying terminal sets

The terminal set

$$\mathcal{X}_f(\alpha_{\max}) = \left\{ x | \sum_{i=1}^{M} x_i^\top P_{f_i} x_i \le \alpha_{\max} \right\},$$

is positive invariant under the terminal controller by construction. However, imposing such a coupled ellipsoid constraint on the online optimization is unreasonable. Instead we consider a terminal set $\mathcal{X}_f$, structured into local terminal sets $\mathcal{X}_{f_i}$:

$$\mathcal{X}_f(\alpha_1^t,...,\alpha_M^t) = \mathcal{X}_{f_1}(\alpha_1^t) \times ... \times \mathcal{X}_{f_M}(\alpha_M^t),$$
$$\mathcal{X}_{f_i}(\alpha_i^t) = \{x_i | x_i^\top P_{f_i} x_i \le \alpha_i^t\}.$$

Here the size α_i^t is time-varying, in order to reduce conservatism. This approach is also used in [ConteEtAl12b] and is based on the analysis in [JokicLazar09]. For $\sum_{i=1}^{M} \alpha_i^t \le \alpha_{\max}, \forall t \ge 0$, we can guarantee $\mathcal{X}_f(\alpha_1^t, \ldots, \alpha_M^t) \subseteq \mathcal{X}_f(\alpha_{\max})$. Therefore the size of the local terminal sets can change such that recursive feasibility is guaranteed, as long as the overall size does not increase. The update of the local terminal set size is done with the parameter update according to the follow lemma.

Lemma 12. *([ConteEtAl16] Lemma 8,9)) If the conditions in lemma 10 are satisfied, the parameter update*

$$\alpha_i^+ = \alpha_i + x_{\mathcal{N}_i}^\top \Gamma_{\mathcal{N}_i} x_{\mathcal{N}_i},$$

ensures

$$x_i \in \mathcal{X}_{f,i}(\alpha_i) \Rightarrow x_i^+ = (A_{\mathcal{N}_i} + B_i K_{\mathcal{N}_i}) x_{\mathcal{N}_i} \in \mathcal{X}_{f,i}(\alpha_i^+),$$
$$\mathcal{X}_f(\alpha_1, \ldots, \alpha_M) \subseteq \mathcal{X}_f(\alpha_{\max}) \Rightarrow \mathcal{X}_f(\alpha_1^+, \ldots, \alpha_M^+) \subseteq \mathcal{X}_f(\alpha_{\max}),$$
$$\alpha_i^+ \ge 0.$$

Proof. From $x_i \in \mathcal{X}_{f,i}(\alpha_i)$ and the positive definitness of P_{f_i} we know

$$0 \le x_i^\top P_{f_i} x_i \le \alpha_i.$$

Using the quadratic terms in (2.8) we have

$$\begin{aligned} x_i^{+} P_{f_i} x_i^{+} &\leq x_i^\top P_{f_i} x_i + x_{\mathcal{N}_i}^\top (\Gamma_{\mathcal{N}_i} + \overline{Q}_i + K_{\mathcal{N}_i}^\top R_i K_{\mathcal{N}_i}) x_{\mathcal{N}_i} \\ &\leq \alpha_i + x_{\mathcal{N}_i}^\top \Gamma_{\mathcal{N}_i} x_{\mathcal{N}_i} \leq \alpha_i^{+}, \end{aligned}$$

which proves the first and third assertion. Due to (2.9) we have $\sum_{i=1}^{M} \alpha_i^{+} \leq \sum_{i=1}^{M} \alpha_i \leq \alpha_{\max}$, which proves the second assertion. □

This update ensure recursive feasibility. One way to initialize the local sets size is then: $\alpha_i^0 = \alpha_{\max}/M$. In general any initialization with $\sum_{i=1}^{M} \alpha_i^0 \leq \alpha_{\max}$, $\alpha_i^0 \geq 0$ works. An alternative method for the initialization can be found in [ConteEtAl12b] Remark IV.5.

Summary - DEMPC with terminal costs and terminal sets

The following two algorithms summarize the offline and online distributed computations:

Offline distributed Synthesis of terminal costs and terminal sets

1. Terminal cost: solve LMI (2.10), (2.12) by distributed optimization
   $\rightarrow$ each system gets $K_{\mathcal{N}_i}, \Gamma_{\mathcal{N}_i}, P_{f_i}$
2. Terminal cost II: solve (2.11), (2.13) by distributed optimization
   $\rightarrow$ each system gets $\gamma_{\mathcal{N}_i}, p_{f_i}$
3. Terminal set size: solve LP (2.14) with tightened constraints $\overline{\mathcal{X}}, \overline{\mathcal{U}}$
   $\rightarrow$ each system gets $\alpha_{\max}$
4. Init Terminal: initialize α_i, such that $\sum_{i=1}^{M} \alpha_i \leq \alpha_{\max}$ (e.g. $\alpha_i = \alpha_{\max}/M$)

Online DEMPC, execute at every time step

1. Each system $i \in \{1,..,M\}$ measures local state x_i^t
2. Solve DEMPC problem (2.1) by distributed optimization
3. Each system $i \in \{1,..,M\}$ applies control input $u_i^*(0)$ computed in step 2
4. Each system $i \in \{1,..,M\}$ updates local terminal set:
   $\alpha_i^{t+1} = \alpha_i^t + x_{\mathcal{N}_i}^*(N+t|t)^\top \Gamma_{\mathcal{N}_i} x_{\mathcal{N}_i}^*(N+t|t)$

2.3 Distributed Economic Model Predictive Control without terminal constraints

In this section we discuss the usage of distributed economic model predictive control (DEMPC) without terminal constraint, also referred to as unconstrained DEMPC. Therefore the MPC optimization problem that needs to be solved in each time step is given by

$$\begin{aligned} \mathcal{V}_N^*(x) = \min \sum_{k=0}^{N-1} l(x(k),u(k)) \\ \text{st. } x(k+1) = Ax(k) + Bu(k), \quad k = 1,..,N \\ u(k) \in \mathcal{U}, \quad x(k) \in \mathcal{X}, \quad k = 1,..,N \\ x(0) = x \end{aligned} \tag{2.15}$$

In contrast to the DEMPC with terminal cost and terminal set, here we have no terminal cost and no terminal set constraint. Therefore one of the greatest advantages of DEMPC without terminal constraints is that complicated offline computations like computing a suitable terminal cost and terminal set can be avoided. In addition, due to the absence of restrictive terminal constraints more states are initially feasible, which increases the region of attraction. The main drawback is that the resulting theoretical guarantees for stability are usually weaker and the corresponding proofs are more involved. In the previously discussed EMPC, that uses a terminal cost and terminal set, we have the property that feasibility of the MPC problem implies stability, due to the stabilizing candidate solution. In EMPC without terminal constraints, this implication does not hold in general and the proofs of stability are of a different nature. In particular recursive feasibility is harder to establish and requires additional assumptions.
Since the DEMPC without terminal constraints requires no complex offline computations, there is no need to change the theoretical results to fit the distributed setup. Instead we can directly use available results for EMPC without terminal constraints.

Stability Results

We will first discuss stability results for EMPC without terminal constraints. These results are all taken from [GrüneStieler14].
To this end we first need to define some notation: Assume without loss of

generality that the optimal equilibrium point is the origin $(x_s,u_s) = (0,0)$. The control law computed by the MPC with a prediction horizon of N is given by $u = \mu_N(x)$. The state trajectory at step k resulting from applying the controller μ_N with an initial state x is denoted by $x_{\mu_N}(k,x)$.
For the EMPC without terminal constraints we will in general not be able to achieve asymptotic stability of the optimal steady state. Therefore we formally define practical asymptotic stability, a stable convergence to a small neighborhood of the optimal steady state.

Definition 13. *The origin is practically asymptotically stable with respect to $\epsilon \geq 0$ on a set $S \subseteq \mathcal{X}$ if there exists a function $\beta \in \mathcal{KL}$ such that*

$$|x_{\mu_N}(k,x)| \leq \max\{\beta(|x|,k),\epsilon\},$$

for all $x \in S$, $k \in \mathbb{N}$.

This can be established with a suitable practical Lyapunov function.

Lemma 14. *(Theorem 2.4 [GrüneStieler14])*
Given a practical Lyapunov function V and functions $\alpha_1,\alpha_2 \in \mathcal{K}_\infty$, $\alpha_3 \in \mathcal{K}$ that satisfy

$$\begin{aligned} \alpha_1(|x|) \leq V(x) \leq \alpha_2(|x|) \quad \forall x \in \mathcal{X} \\ V(Ax + B\mu_N(x)) - V(x) \leq -\alpha_3(|x|) + \delta \quad \forall x \in S \end{aligned}$$

with either $\mathcal{X} = \mathcal{S}$ or $S = \{x|V(x) \leq L\}$ with $L > \alpha_2(\alpha_3^{-1}(\delta)) + \delta$.
Then the origin is practically asymptotically stable on the set S with respect to $\epsilon = \alpha_1^{-1}(\alpha_2(\alpha_3^{-1}(\delta)) + \delta)$.

To establish asymptotic practical stability of the EMPC without terminal constraints we use the following assumptions.

Assumption 15. *Strict Dissipativity*
The optimal control problem is strictly dissipative, i.e. there exists a function $\alpha_l \in \mathcal{K}_\infty$ and a storage function λ, that satisfy

$$\min_{u \in \mathcal{U}} \tilde{l}(x,u) \geq \alpha_l(|x|) \quad \forall x \in \mathcal{X} \tag{2.16}$$

where $\tilde{l}(x,u) = l(x,u) + \lambda(x) - \lambda(Ax + Bu) - l(0,0)$ is the rotated stage cost.

Assumption 16. *Continuity + Compactness:*
The state and input constraints $\mathcal{X}$ and $\mathcal{U}$ are compact.
The functions f,l,λ are continuous and λ is Lipschitz continuous on a ball B_δ.
There is a $\alpha \in \mathcal{K}_\infty$ such that the rotated stage cost satisfies

$$\tilde{l}(x,u) \leq \alpha(|x|) + \alpha(|u|) \quad \forall x \in \mathcal{X}, u \in \mathcal{U}.$$

Assumption 17. *Local controllability on $\mathcal{B}_\epsilon$*
There exists a $\epsilon > 0$, $M' \in \mathbb{N}$, $C > 0$ such that $\forall x \in \mathcal{B}_\epsilon$ $\exists u_1 \in \mathcal{U}^{M'}(x)$, $u_2 \in \mathcal{U}^{M'}(0)$ with $x_{u_1}(M',x) = 0$, $x_{u_2}(M',0) = x$ and

$$\max\{\|x_{u_1}(k,x)\|, \|x_{u_2}(k,0)\|, \|u_1(k)\|, \|u_2(k)\|\} \leq C\|x\|, \quad k = 0, \ldots, M' - 1.$$

Assumption 18. *Finite time controllability into $\mathcal{B}_\epsilon$*
For $\epsilon > 0$ from assumption 17 there is a $K \in \mathbb{N}$, such that for all $x \in \mathcal{X}$ there is a $k \leq K$ and $u \in \mathcal{U}^k(x)$ with

$$x_u(k,x) \in \mathcal{B}_\epsilon.$$

The following theorem establishes the practical asymptotic stability of EMPC without terminal constraints.

Theorem 19. *([GrüneStieler14] Theorem 3.7) Let assumption 15-18 be satisfied. Then there exists a $N_0 \in \mathbb{N}$ and functions $\delta \in \mathcal{L}$, $\alpha_v \in \mathcal{K}_\infty$ such that*

$$\alpha_l(|x|) \leq \tilde{\mathcal{V}}_N^*(x) \leq \alpha_V(|x|)$$
$$\tilde{\mathcal{V}}_N^*(Ax + B\mu_N(x)) \leq \tilde{\mathcal{V}}_N^*(x) - \alpha_l(\|x\|) + \delta(N)$$

is satisfied for all $N \geq N_0$ and $x \in \mathcal{X}$.
This implies practical asymptotic stability of the origin with

$$\|x_{\mu_N}(k,x)\| \leq \max\{\beta(\|x\|,k), \epsilon(N)\},$$
$$\epsilon(N) = \alpha_l^{-1}(\alpha_V(\alpha_l^{-1}(\delta(N))) + \delta(N)) \in \mathcal{L}, \quad \beta \in \mathcal{KL}.$$

Proof. The stability proof consists of two parts: First the bounds on $\tilde{\mathcal{V}}_N^*$ are computed. Then the Lyapunov decrease condition is derived.
Part 1: The lower bound on the practical Lyapunov function $\tilde{\mathcal{V}}_N^*$ follows directly from the strict dissipativity property. The upper bound comes from assumptions 16, 17 and 18. If $x \notin \mathcal{B}_\epsilon$ assumption 18 guarantees that there

exists a u, that steers the system to the equilibrium with $M' + K$ steps. This implies

$$\tilde{\mathcal{V}}_N^*(x) \leq \tilde{J}_N(x,u) \leq (M' + K) \max_{x \in \mathcal{X}, u \in \mathcal{U}} \tilde{l}(x,u) =: \overline{C},$$

where $\tilde{J}_K(x,u)$ describes the open-loop rotated cost of a trajectory. If $x \in \mathcal{B}_\epsilon$ we can use assumption 16 and 17 to get

$$\tilde{\mathcal{V}}_N^*(x) \leq \tilde{J}_{M'}(x,u_1) \leq \sum_{k=0}^{M'-1} \alpha(|x_{u_1}(k,x)|) + \alpha(|u_1(k,x)|) \leq 2M'\alpha(C|x|) \leq \tilde{\alpha}(|x|)$$

with $\tilde{\alpha} \in \mathcal{K}_\infty$. By choosing a $\overline{K}$ such that $\overline{K}\tilde{\alpha}(|x|) \geq \overline{C}$, we get $\alpha_V(r) = \max(1,\overline{K})\tilde{\alpha}(r)$.
Part 2: In [Grüne13] it was shown, that for a sufficiently large N we have

$$\tilde{J}_K(x,\mu_N(x)) \leq \tilde{\mathcal{V}}_N^*(x) - \tilde{\mathcal{V}}_{N-K}^*(x_{\mu_N}(K,x)) + \tilde{\delta}(N)$$

with $\tilde{\delta} \in \mathcal{L}$. Inserting K=1 and using the bound on the rotated cost function we get

$$\tilde{\mathcal{V}}_N^*(Ax + B\mu_N(x)) \leq \tilde{\mathcal{V}}_N^*(x) - \alpha_l(|x|) + \tilde{\delta}(N).$$

Practical asymptotical stability then follows by applying lemma 14. □

Remark 20. *We can use this theorem to establish practical asymptotic stability. A deeper investigation in the stability properties will be carried out in section 4.3. This result also holds for a more general setup with nonlinear stage cost and dynamics, as long as the corresponding assumptions hold. There is also no need to change any of the conditions for distributed EMPC in comparison to standard EMPC.*
Assumption 18 can be very difficult to satisfy in practical applications, due to the control invariance. One way to satisfy this is by replacing the state constraint set $\mathcal{X}$ by the recursivly feasible set $\mathcal{X}_N$. Alternatively due to the Lyapunov proof we have an invariant level set, which can be used to guarantee recursive feasibility (see [BocciaGrüneWorthmann14] for details).
This theorem only guarantees the existence of a large enough prediction horizon, that guarantees practical asymptotic stability. In practice the estimates for N_0 can be very conservative.

Turnpike Property

It should be noted that under appropriate controllability assumptions, there is an equivalence between the strict dissipativity and the turnpike property [DammEtAl14,GrüneMüller16]. Loosely speaking, the turnpike property says, that for a long enough prediction horizon N the predicted state trajectory initially converges close to the optimal steady state x_s and then moves to a region with a smaller stage cost. Another similar property, is that the system is suboptimally operated outside the optimal steady state.
While this property is equivalent to the strict dissipativity property and harder to check, it can provide a better interpretation and insight in the behavior of EMPC without terminal constraints and explain why too short prediction horizons might not lead to stability.

Average Constraints

In EMPC without terminal constraints, average constraints can be used to ensure convergence to a steady state. This mainly relates to assumption 15. An average constraint on a linear auxiliary output

$$y = h_x x + h_u u + c$$

is defined as

$$\mathrm{Av}[y] \subset \mathcal{Y} = \{y | A_{\mathrm{av}} y \leq b_{\mathrm{av}}\},$$

where

$$\mathrm{Av}[y] = \left\{ \bar{y} | \exists t_n \to \infty : \lim_{n\to\infty} \frac{\sum_{k=0}^{t_n} y(k)}{t_n + 1} = \bar{y} \right\}$$

defines the asymptotic average [AngeliAmritRawlings12]. This average constraint is incorporated in the EMPC with the constraint

$$\sum_{k=0}^{N-1} y(k) \in \mathcal{Y}_t,$$

where $\mathcal{Y}_t$ is updated in each time step with

$$\mathcal{Y}_{t+1} = \mathcal{Y}_t \bigoplus \mathcal{Y} \bigominus y_t.$$

The average constraint can be used to enforce convergence by considering dissipativity under average constraints:

Assumption 21. *Strict Dissipativity under average constraints*
The optimal control problem is strictly dissipative under the average constraints, i.e. there exists a storage function λ*, a multiplier* $\overline{\lambda} \in [0,\infty)^p$ *and a function* $\alpha_l \in \mathcal{K}_\infty$ *such that we have*

$$\min_{u \in \mathcal{U}} \tilde{l}(x,u) - \overline{\lambda}^\top y(x,u) \geq \alpha_l(|x|), \quad \forall x \in \mathcal{X}.$$

With additional average constraints, we can ensure that the system is suboptimally operated off steady state, which in turn implies that the EMPC will converge close to the optimal steady state. An additional difficulty, that arises by using average constraints, is establishing recursive feasibility. If this is given, we can again establish practical asymptotic stability of the optimal steady state.

Recursive Feasibility of unconstrained EMPC with average constraints

In [MüllerAngeliAllgöwer13] a stability proof for zero-terminal constraint EMPC with average constraints is derived and in [MüllerEtAl14] it was extended to terminal cost/set EMPC. But for EMPC without terminal constraint guaranteeing recursive feasibility with average constraints is still an unresolved issue. The usual way of guaranteeing feasibility due to stability, becomes more complex with average constraints. In particular assumptions 17 and 18 need to include the average constraints.
This issue can be avoided by defining an appropriate dummy state variable and using corresponding state constraints. For example let us assume we have the average constraint $av[y] = 0$. Then we can define the dummy state $\tilde{x}^+ = \tilde{x} + y$ and impose simple bounds $\tilde{x} \in [-\Delta y N, \Delta y N]$. If the extended system $\tilde{x}$ is controlled with the extended state constraints, the system satisfies the average constraint $av[y] = 0$. Similarly for $av[y] \in [y_{\min}, y_{\max}]$ we can use

$$\tilde{x}_1 = \tilde{x}_1 + y - y_{\min}, \quad \tilde{x}_2 = \tilde{x}_2 + y - y_{\max}$$

with state constraints on $\tilde{x}_1$, $\tilde{x}_2$. For transient constraints this is more difficult.

2.4 Robust Distributed Model Predictive Control

This section investigates tube based robust distributed model predictive control (RDMPC). In the first part the general idea of tube based robust MPC is introduced and two robust MPC variants based on [MayneSeronRaković05] and [ChisciRossiterZappa01] are presented . In the second part these variants are discussed in the context of distributed MPC and the corresponding distributed computations are derived and summarized. In particular, one approach to compute ellipsoidal robust positive invariant (RPI) sets with ellipsoidal bounds on the disturbances based on [ConteEtAl13] is presented. Furthermore, a second approach, that computes both a local ellipsoidal RPI set and a stabilizing controller for polytopic disturbance is derived, which is a combination of [ConteEtAl13] and [LimonEtAl08a].

2.4.1 Tube based Robust Model Predictive Control

Here we discuss the usage of tube based robust MPC. We first give a short motivation, why robust MPC modifications are considered. Then we explain the basic idea and discuss the robust positive invariant (RPI) set approach and the growing tubes approach.

Motivation

In the previous two sections, we assumed ideal conditions, i.e. that the predicted trajectories in the optimization are equal to the real resulting system trajectory. In practice there are two common issues, that lead to an error in the predictions. There is often a model mismatch, due to the use of simplified models, unpredictable disturbances or inexact model parameters. In addition, we often have an additional error term due to the dual inexact optimization, see chapter 3 and chapter 4.
To ensure that the previously derived results hold despite such prediction errors, we investigate robust distributed model predictive control (RDMPC). In particular, we focus on tube based robust DMPC.

Basic Idea

In robust tube based MPC, the MPC algorithm is augmented with a stabilizing controller to ensure that the state trajectory under disturbances is confined to a tube around the nominal predicted trajectory. Due to the asymptotic stability of the nominal trajectory, the system converges to the tube around the optimal steady state. To ensure that the system constraints are still satisfied, a robust constraint tightening is performed, which takes the worst case disturbance into account. This modification can be used for both

terminal cost/set DEMPC and DEMPC without terminal constraints. We will discuss these modifications in the context of terminal cost/set DEMPC, since this requires some additional steps for the terminal set. We do not change the robust modifications for the economic MPC, even though there exists other approaches [BayerEtAl16] to improve the performance.
We assume an uncertain, but bounded additive disturbance $w \in \mathcal{W}$, which results in the model

$$x^+ = Ax + Bu + Ew.$$

Due to the uncertain disturbance w we distinguish between the nominal predicted input and state trajectories v, z and the actual trajectories u, x impacted by the disturbances.
The main components of tube based robust MPC are RPI tubes $\mathcal{Z}$, a stabilizing control law K_t and a tightening of the nominal constraints.

Robust MPC with RPI tubes

In [MayneSeronRaković05] a robust positive invariant (RPI) set $\mathcal{Z}$ is computed and used as a constant tube around the nominal trajectory. The RPI set $\mathcal{Z}$ has to satisfy

$$(A + BK_t)\mathcal{Z} \bigoplus E\mathcal{W} \subseteq \mathcal{Z},$$

with the stabilizing controller K_t, to ensures stability and robust positive invariance. With the PRI set we have the property that the feedback

$$u = v + K_t(x - z)$$

keeps the state trajectory x confined to the tube around the nominal predicted state trajectory z:

$$x \in z \bigoplus \mathcal{Z} \;\Rightarrow\; x^+ \in z^+ \bigoplus \mathcal{Z},$$
$$z^+ = Az + Bv, \quad x^+ = Ax + Bu + Ew.$$

The tightened constraints on the nominal trajectory can be computed as

$$\overline{\mathcal{X}} = \mathcal{X} \bigominus \mathcal{Z}, \quad \overline{\mathcal{U}} = \mathcal{U} \bigominus K_t\mathcal{Z}.$$

This way we can ensure satisfaction of the original constraints $(x,u) \in \mathcal{X} \times \mathcal{U}$, by requiring that the nominal predictions lie in the tightened constraint set

$(z,v) \in \overline{\mathcal{X}} \times \overline{\mathcal{U}}$.
The computation of the terminal set $\overline{\mathcal{X}}_f$ is done with the usual procedure in (2.7), by using the tightened constraint sets $\overline{\mathcal{X}}, \overline{\mathcal{U}}$ instead of the original constraint sets $\mathcal{X}, \mathcal{U}$.
Since the tube is invariant, the initial state of the nominal state trajectory $z(0)$ can be used as an optimization variable, as long as it lies within the tube around the real state. With this the new robust MPC optimization problem with terminal set and terminal cost is given by

$$
\begin{aligned}
\min_{z(0),v} & \sum_{k=0}^{N-1} l(z(k),v(k)) + V_f(z(N)) \\
\text{st. } & z(0) \in x \bigoplus \mathcal{Z} \\
& z(k+1) = Az(k) + Bv(k), \quad k = 1,\dots,N-1 \\
& (z(k),v(k)) \in \overline{\mathcal{X}} \times \overline{\mathcal{U}}, \quad k = 1,\dots,N-1 \\
& z(N) \in \overline{\mathcal{X}}_f,
\end{aligned}
$$

where x is the current state and $z(k), v(k)$ are the predicted nominal states and inputs. In each time step the input

$$u = v^*(0) + K_t(x - z^*(0))$$

is applied. The following theorem establishes the stability properties of this robust MPC approach.

Theorem 22. *([MayneSeronRaković05] Theorem 1) Let assumptions 2 and 4 hold with the tightened constraints $\overline{\mathcal{U}}$, $\overline{\mathcal{X}}$, $\overline{\mathcal{X}}_f$.*
Then the state of the system $x^+ = Ax + B(v^*(0) + K_t(x - z^*(0))) + Ew$, $w \in \mathcal{W}$ exponentially converges to $\mathcal{Z}$.

The main idea of the proof is to show convergence of the nominal state z to the optimal steady state. Since the real state x lies within a tube $\mathcal{Z}$ around z, x converges to $\mathcal{Z}$. By optimizing over the initial state $z(0)$ we can reduce conservatism and the MPC has a degree of freedom to react to disturbances instead of just stabilizing the nominal trajectory, but we also have a higher computational demand online. Note that the computation of the terminal set and robust tightening can be carried out in succession, i.e. first compute the robust tightening and then the terminal set.

Robust MPC with growing tubes

In [ChisciRossiterZappa01] the growing tube robust MPC approach is derived, which uses pre-stabilized dynamics, with the stabilizing controller K_t. This approach does not require the computation of a RPI set, but instead uses a growing tube along the prediction horizon, that approaches the size of the RPI set $\mathcal{Z}$ in the limit. The constraints are then tightened using the growing tube along the prediction horizon.
The corresponding tightened nominal constraints for each prediction step k are computed as:

$$\overline{\mathcal{X}}_k = \mathcal{X} \ominus \left(\bigoplus_{j=0}^{k-1} (A + BK_t)^j E\mathcal{W} \right), \qquad k = 0, \dots, N-1,$$
$$\overline{\mathcal{U}}_k = \mathcal{U} \ominus K_t \left(\bigoplus_{j=0}^{k-1} (A + BK_t)^j E\mathcal{W} \right), \qquad k = 0, \dots, N-1.$$

The tightened terminal set uses the nominal terminal set $\mathcal{X}_f$ and then tightens it with

$$\overline{\mathcal{X}}_f = \mathcal{X}_f \ominus \left(\bigoplus_{j=0}^{N-1} (A + BK_t)^j E\mathcal{W} \right).$$

For comparison the minimal RPI set is defined as

$$\mathcal{Z}_\infty := \bigoplus_{j=0}^{\infty} (A + BK_t)^j E\mathcal{W},$$

which means that the constraint tightening in this approach is smaller, compared to the RPI set approach. With this notation, the robust MPC optimization problem for the growing tubes approach is given by

$$\begin{aligned}
\min_v \; & \sum_{k=0}^{N-1} l(z(k), v(k)) + V_f(z(N)) \\
\text{st. } & z(0) = x \\
& z(k+1) = (A + BK_t)z(k) + Bv(k), \quad k = 1, \dots, N-1 \\
& (z(k), v(k) + K_t z(k)) \in \overline{\mathcal{X}}_k \times \overline{\mathcal{U}}_k, \quad k = 0, \dots, N-1 \\
& z(N) \in \overline{\mathcal{X}}_f
\end{aligned}$$

and the resulting MPC input is

$$u = v^*(0) + K_t x.$$

Due to the use of a pre-stabilizing controller K_t, the input constraints involve the state trajectory. The following theorem establishes the convergence of this robust MPC approach.

Theorem 23. *([ChisciRossiterZappa01] Theorem 8) Let assumption 2 and 4 hold with the constraints $\mathcal{U}$, $\mathcal{X}$, $\mathcal{X}_f$.*
Then the state of the system $x^+ = Ax + B(v^*(0) + K_t x) + Ew$, $w \in \mathcal{W}$ converges to the minimum RPI set of the system $x^+ = (A + BK_t)x + Ew$, $w \in \mathcal{W}$.

The resulting optimization problem has the same complexity as the original MPC problem, which makes it a very attractive add-on. Both theorems are originally for stablizing MPC, but for EMPC with terminal cost/set the same argument holds for the rotated cost (see proof theorem 6).

2.4.2 Robust Distributed Model Predictive Control

Now we focus on the distributed case. As in section 2.2, the theoretical results do not change when we transition to a distributed setup. The challenge is to ensure, that all the offline computations required in the central robust MPC case can be carried out with distributed optimization and result in distributed structures. This includes in particular the computation of a stabilizing controller K_t, the computation of a structured RPI set $\mathcal{Z}$ and the tightening of the constraints. The outlined procedures are very similar to the distributed optimization used in section 2.2 for the terminal cost/set DEMPC.
The distributed linear system subject to local uncertain bounded additive disturbances is described by

$$x_i^+ = A_{\mathcal{N}_i} x_{\mathcal{N}_i} + B_i u_i + E_i w_i, \quad w_i \in \mathcal{W}_i.$$

First, a procedure to compute a distributed stabilizing controller K_t with local ellipsoidal RPI sets $\mathcal{Z}$ is presented, which is a distributed version of the computations in [LimonEtAl08a]. Then a distributed optimization to obtain less conservative ellipsoidal RPI sets $\mathcal{Z}$ from [ConteEtAl13] is presented, which assumes a given stabilizing controller K_t. Finally all the offline and online computations for RPI tubes and growing tubes robust DMPC are summarized.

Distributed controller synthesis and structured RPI sets for polytopic disturbances

The following procedure computes a distributed stabilizing controller K_t and a structured distributed ellipsoid RPI set $\mathcal{Z}$. This is a distributed version of the procedure in [LimonEtAl08a] and uses ideas of the distributed RPI set from [ConteEtAl13]. For the disturbances w_i we assume local polytopic constraints

$$\mathcal{W}_i = \{w_i | A_{w_i} w_i \leq b_{w_i}\}, \quad \mathcal{W} = \mathcal{W}_1 \times \cdots \times \mathcal{W}_M.$$

For the stabilizing controller in the distributed setup, we assume that the local input only depends on neighboring states, similar to the distributed terminal controller:

$$u_i = K_{t,\mathcal{N}_i} x_{\mathcal{N}_i}.$$

For the RPI set $\mathcal{Z}$ we use a distributed ellipsoid approach

$$\mathcal{Z} = \{x | x^\top P_{\mathcal{Z}} x = \sum_{i=1}^{M} x_i^\top P_{\mathcal{Z}_i} x_i \leq 1\}.$$

By using the approach from [ConteEtAl13] the RPI set constraint is replaced by local RPI constraints $\mathcal{Z}_i$, with

$$x_i \in \mathcal{Z}_i = \{x | x_i^\top P_{\mathcal{Z}_i} x_i \leq \beta_i\}, \quad \sum_{i=1}^{M} \beta_i \leq 1, \quad \beta_i \geq 0.$$

We assume local polytopic constraints, which can be written as

$$\mathcal{X}_{\mathcal{N}_i} = \{x_{\mathcal{N}_i} || h_{\mathcal{N}_i,j} x_{\mathcal{N}_i}| \leq 1, j = 1, \ldots, n_{x_i}\},$$
$$\mathcal{U}_i = \{u_i || l_{i,j} u_i| \leq 1, j = 1, \ldots, n_{u_i}\}.$$

The following lemma gives a distributed SDP, that can be used to compute a suitable controller K_t and RPI set $\mathcal{Z}$.

Lemma 24. *The structured RPI set* $\mathcal{Z} = \{x | \sum_{i=1}^{M} x_i^\top P_{\mathcal{Z}_i} x_i \leq 1\}$ *with*

$$\mathcal{Z}_{\mathcal{N}_i} \subseteq \sqrt{\gamma_i} \mathcal{X}_{\mathcal{N}_i}, \quad K_{t,\mathcal{N}_i} \mathcal{Z}_{\mathcal{N}_i} \subseteq \sqrt{\rho_i} \mathcal{U}_i,$$

can be computed with the following distributed SDP:

$$
\begin{aligned}
&\min_{Y_i, W_i, \gamma_i} \sum_{i=1}^{M} \gamma_i \\
st.\ &\begin{pmatrix} \rho_i & l_{i,j}^\top Y_i \\ Y_i^\top l_{i,j} & W_{\mathcal{N}_i} \end{pmatrix} \geq 0, \quad j = 1, \ldots, n_{u_i}, \\
&\begin{pmatrix} \gamma_i & h_{\mathcal{N}_i,j}^\top W_{\mathcal{N}_i} \\ W_{\mathcal{N}_i} h_{\mathcal{N}_i,j} & W_{\mathcal{N}_i} \end{pmatrix} \geq 0, \ldots, M, \quad j = 1, \ldots, n_{x_i}, \\
&\begin{pmatrix} \lambda \overline{W}_i & 0 & (A\overline{W}_i + B\overline{Y}_i)^\top \\ 0 & \beta_i(1-\lambda) & \overline{w}_i^\top \overline{E}_i^\top \\ A\overline{W}_i + B\overline{Y}_i & \overline{E}_i \overline{w}_i & \overline{W}_i \end{pmatrix} \geq \overline{S}_i, \quad \forall w_i \in vert(\mathcal{W}_i), \\
&\sum_{i=1}^{M} \overline{S}_i \geq 0, \quad \lambda \geq 0, \quad \sum_{i=1}^{M} \beta_i \leq 1, \beta_i \geq 0, \\
&i = 1, \ldots, M,
\end{aligned}
$$

with

$$
W = \sum_{i=1}^{M} \overline{W}_i, \quad W_i = P_{\mathcal{Z}_i}^{-1}, \quad Y = \sum_{i=1}^{M} \overline{Y}_i, \quad Y_i = K_{t,\mathcal{N}_i} W_{\mathcal{N}_i}, \quad \overline{W}_{\mathcal{N}_i} = \sum_{j \in \mathcal{N}_i} \overline{W}_i.
$$

The RPI set is given by $P_z = W^{-1}$, and the local controller is given by $K_t = YP_{\mathcal{Z}}$.

Proof. This proof consists of two parts. First, the variables are defined and the RPI condition is written as a LMI. In the second part the state and input constraints are included in the optimization, with the variables ρ_i, γ_i.
Part I: In this part we derive a distributed LMI condition, that is sufficient for the RPI condition. Denote the closed-loop controlled dynamics by $A_k = A + BK_t$. The RPI condition can be written as

$$
(A_k x + Ew)^\top P_{\mathcal{Z}} (A_k x + Ew) \leq 1, \quad \forall x^\top P_{\mathcal{Z}} x \leq 1, \quad \forall w \in \mathcal{W}.
$$

Due to the convexity in w it is sufficient if this condition is satisfied for all vertices w of $\mathcal{W}$, $w \in \text{vert}(\mathcal{W})$:

$$
(A_k x + Ew)^\top P_{\mathcal{Z}} (A_k x + Ew) \leq 1, \quad \forall x^\top P_{\mathcal{Z}} x \leq 1, \quad \forall w \in \text{vert}(\mathcal{W}).
$$

By using the S-procedure this can be written as

$$(A_k x + Ew)^\top P_{\mathcal{Z}} (A_k x + Ew) - 1 + \lambda(1 - x^\top P_{\mathcal{Z}} x) \leq 0, \quad \forall w \in \mathrm{vert}(\mathcal{W}), \quad \lambda \geq 0,$$

which can be expressed as the following matrix inequality

$$\lambda \begin{pmatrix} P_{\mathcal{Z}} & 0 \\ 0 & -1 \end{pmatrix} - \begin{pmatrix} A_k^\top P_{\mathcal{Z}} A_k & A_k^\top P_{\mathcal{Z}} Ew \\ w^\top E^\top P_{\mathcal{Z}} A_k & w^\top E^\top P_{\mathcal{Z}} Ew - 1 \end{pmatrix} \geq 0, \quad \lambda \geq 0, \quad \forall w \in \mathrm{vert}(\mathcal{W}).$$

Now we define $W = P_{\mathcal{Z}}^{-1}$ and $Y = K_t P_{\mathcal{Z}}^{-1}$. Multiplying the inequality with $\begin{pmatrix} W & 0 \\ 0 & 1 \end{pmatrix}$ from left and right yields

$$\lambda \begin{pmatrix} W & 0 \\ 0 & -1 \end{pmatrix} - \begin{pmatrix} (AW+BY)^\top W^{-1}(AW+BY) & (AW+BY)^\top W^{-1} Ew \\ w^\top E^\top W^{-1}(AW+BY) & w^\top E^\top W^{-1} Ew - 1 \end{pmatrix} \geq 0 \quad \lambda \geq 0, \quad \forall w \in \mathrm{vert}(\mathcal{W}).$$

By rewriting this inequality as

$$\begin{pmatrix} \lambda W & 0 \\ 0 & 1-\lambda \end{pmatrix} - \begin{pmatrix} (AW+BY)^\top \\ w^\top E^\top \end{pmatrix} W^{-1} \begin{pmatrix} (AW+BY) & Ew \end{pmatrix} \geq 0, \quad \lambda \geq 0, \quad \forall w \in \mathrm{vert}(\mathcal{W}),$$

we can apply the Schur complement to get the following LMI condition

$$\begin{pmatrix} \lambda W & 0 & (AW+BY)^\top \\ 0 & 1-\lambda & w^\top E^\top \\ AW+BY & Ew & W \end{pmatrix} \geq 0, \quad \lambda \geq 0, \quad \forall w \in \mathrm{vert}(\mathcal{W}).$$

By using the distributed structure of the RPI set and the controller we also get a distributed structure in the new variables W,Y:

$$W = \sum_{i=1}^{M} \overline{W}_i, \quad W_i = P_{\mathcal{Z}_i}^{-1}$$
$$Y = \sum_{i=1}^{M} \overline{Y}_i, \quad Y_i = K_{t,\mathcal{N}_i} W_{\mathcal{N}_i}, \quad \overline{W}_{\mathcal{N}_i} = \sum_{j \in \mathcal{N}_i} \overline{W}_i,$$

where the $\overline{\text{overline}}$ denotes a lifting of the local matrices to the global model, as described in section 2.2 for the terminal cost/set. Plugging in this structure yields

$$\begin{pmatrix} \lambda \sum_{i=1}^{M} \overline{W}_i & 0 & \sum_{i=1}^{M} (A\overline{W}_i + B\overline{Y}_i)^\top \\ 0 & \sum_{i=1}^{M} \beta_i(1-\lambda) & \sum_{i=1}^{M} \overline{w}_i^\top \overline{E}_i^\top \\ \sum_{i=1}^{M} A\overline{W}_i + B\overline{Y}_i & \sum_{i=1}^{M} \overline{E}_i\overline{w}_i & \sum_{i=1}^{M} \overline{W}_i \end{pmatrix} \geq 0,$$

$$\lambda \geq 0, \quad \sum_{i=1}^{M} \beta_i \leq 1, \beta_i \geq 0, \quad \forall w_i \in \text{vert}(\mathcal{W}_i).$$

Then the following set of distributed LMI conditions is sufficient

$$\begin{pmatrix} \lambda\overline{W}_i & 0 & (A\overline{W}_i + B\overline{Y}_i)^\top \\ 0 & \beta_i(1-\lambda) & \overline{w}_i^\top \overline{E}_i^\top \\ A\overline{W}_i + B\overline{Y}_i & \overline{E}_i\overline{w}_i & \overline{W}_i \end{pmatrix} \geq \overline{S}_i$$

$$\sum_{i=1}^{M} \overline{S}_i \geq 0, \quad \lambda \geq 0, \quad \sum_{i=1}^{M} \beta_i \leq 1, \beta \geq 0, \quad \forall w_i \in \text{vert}(\mathcal{W}_i), \quad i = 1, \ldots, M,$$

where S_i can for example be chosen as a block diagonal matrix with appropriate dimensions. The matrix S_i corresponds to the coupling between subsystems to reduce conservatism, and plays a similar role to $F_{\mathcal{N}_i}$ in lemma 10.
Part II: In the second part the constraints are included in the optimization, in order to lead to small constraint tightening. The conditions on the state and input constraints

$$\mathcal{Z}_{\mathcal{N}_i} \subseteq \sqrt{\gamma}_i \mathcal{X}_{\mathcal{N}_i}, \quad K_{t,\mathcal{N}_i}\mathcal{Z}_{\mathcal{N}_i} \subseteq \sqrt{\rho}_i \mathcal{U}_i,$$

can be explicitly written as

$$\begin{aligned} |l_{i,j}^\top K_{t,\mathcal{N}_i}, x_{\mathcal{N}_i}| \leq \sqrt{\rho}_i, \quad j = 1, \ldots, n_{u_i} \\ |h_{\mathcal{N}_i,j}^\top x_{\mathcal{N}_i}| \leq \sqrt{\gamma}_i, \quad j = 1, \ldots, n_{x_i} \\ \forall x_{\mathcal{N}_i}^\top P_{\mathcal{Z},\mathcal{N}_i} x_{\mathcal{N}_i} \leq 1, \quad \overline{P}_{\mathcal{Z},\mathcal{N}_i} = \sum_{j\in\mathcal{N}_i} \overline{P}_{\mathcal{Z}_i}. \end{aligned}$$

By applying the Schur complement this can be written as the following matrix inequalities

$$\begin{pmatrix} \gamma_i & h_{\mathcal{N}_i,j}^\top \\ h_{\mathcal{N}_i,j} & P_{\mathcal{Z},\mathcal{N}_i} \end{pmatrix} \geq 0, \quad \begin{pmatrix} \rho_i & l_{i,j}^\top K_{t,\mathcal{N}_i} \\ K_{t,\mathcal{N}_i}^\top l_{i,j} & P_{\mathcal{Z},\mathcal{N}_i} \end{pmatrix} \geq 0.$$

Now we multiply the inequalities with $\begin{pmatrix} 1 & 0 \\ 0 & W_{\mathcal{N}_i} \end{pmatrix}$ from left and right to get LMIs in the new variables

$$\begin{pmatrix} \rho_i & l_{i,j}^\top Y_i \\ Y_i^\top l_{i,j} & W_{\mathcal{N}_i} \end{pmatrix} \geq 0, \quad i = 1, \ldots, M, \quad j = 1, \ldots, n_{u_i},$$

$$\begin{pmatrix} \gamma_i & h_{\mathcal{N}_i,j}^\top W_{\mathcal{N}_i} \\ W_{\mathcal{N}_i} h_{\mathcal{N}_i,j} & W_{\mathcal{N}_i} \end{pmatrix} \geq 0, \quad i = 1, \ldots, M, \quad j = 1, \ldots, n_{x_i},$$

which finishes the proof. □

Remark 25. *The minimization of γ_i ensures, that the robust constraint tightening leads to a small constraint tightening in the state constraints. The parameter ρ_i sets the constraint tightening in the input constraints. The variables $\rho_i \in (0,1]$ and $\lambda \geq 0$ are not included as optimization variables, but are set as design parameters (or updated in an outer loop). Alternatively ρ_i can also be included in the objective function, to lead to a small input constraint tightening. The parameter β_i can be used in the optimization (since its linear), which leads to a global constraint or it can simply be set to $\beta_i = 1/M$. Compared to the procedure in [ConteEtAl13] the computation here is larger and does not require a stabilizing controller a priori. But we are limited to local RPI sets $P_{\mathcal{Z}_i}$ without coupling to neighbors, which can lead to more conservative RPI sets.*

Distributed RPI set computation for ellipsoid disturbances

Now an approach to compute less conservative RPI sets based on [ConteEtAl13] is presented. This approach assumes a given stabilizing controller K_t and only computes the RPI set. Here the structured RPI sets take the full neighborhood into account with

$$\mathcal{Z} = \{x | x^\top P_{\mathcal{Z}} x = \sum_{i=1}^{M} x_{\mathcal{N}_i}^\top P_{\mathcal{Z},\mathcal{N}_i} x_{\mathcal{N}_i} \leq 1\}.$$

Instead of considering local polytopic disturbances, we consider local ellipsoidal disturbances $\mathcal{W}_i = \{w_i | w_i^\top w_i \leq v_i\}$. As in the previous method, the RPI set can be enforced by local RPI sets

$$x_{\mathcal{N}_i} \in \mathcal{Z}_{\mathcal{N}_i} = \{x_{\mathcal{N}_i} | x_{\mathcal{N}_i}^\top P_{\mathcal{Z},\mathcal{N}_i} x_{\mathcal{N}_i} \leq \beta_i\},$$
$$\sum_{i=1}^{M} \beta_i \leq 1, \quad \beta_i \geq 0.$$

We will denote this tighter distributed RPI set in dependence of the parameters β_i as $\mathcal{Z}(\beta)$. The following lemma shows how the corresponding matrices can be computed by distributed optimization.

Lemma 26. *([ConteEtAl13]) For a given stabilizing controller K_t, the structured RPI set $\mathcal{Z} = \{x | \sum_{i=1}^{M} x_{\mathcal{N}_i}^\top P_{z,\mathcal{N}_i} x_{\mathcal{N}_i} \leq 1\}$ with*

$$\mathcal{Z}_{\mathcal{N}_i} \subseteq \sqrt{\gamma}_i \mathcal{X}_{\mathcal{N}_i}, \quad K_{t,\mathcal{N}_i} \mathcal{Z}_{\mathcal{N}_i} \subseteq \sqrt{\rho}_i \mathcal{U}_i,$$

can be computed with the following distributed SDP:

$$\begin{aligned}
&\min_{P_{\mathcal{N}_i}, S_i, s_i, \gamma_i} \sum_{i=1}^{M} \gamma_i \\
&\quad st. \begin{pmatrix} \gamma_i & h_{\mathcal{N}_i,j}^\top \\ h_{\mathcal{N}_i,j} & \tilde{P}_{\mathcal{Z},\mathcal{N}_i} \end{pmatrix} \geq 0, \quad i = 1, \ldots, M, \quad j = 1, \ldots, n_{x_i}, \\
&\quad \begin{pmatrix} \rho_i^2 & l_{i,j}^\top K_{t,\mathcal{N}_i} \\ K_{t,\mathcal{N}_i}^\top l_{i,j} & \tilde{P}_{\mathcal{Z},\mathcal{N}_i} \end{pmatrix} \geq 0, \quad i = 1, \ldots, M, \quad j = 1, \ldots, n_{u_i}, \\
&\quad \begin{pmatrix} -(A+BK_t)^\top \overline{P}_{\mathcal{Z},\mathcal{N}_i} (A+BK_t) & -(A+BK_t)^\top \overline{P}_{\mathcal{Z},\mathcal{N}_i} E & 0 \\ -E^\top \overline{P}_{\mathcal{Z},\mathcal{N}_i} (A+BK_t) & -E^\top \overline{P}_{\mathcal{Z},\mathcal{N}_i} E & 0 \\ 0 & 0 & \beta_i \end{pmatrix} \\
&\quad - s_0 \begin{pmatrix} -\overline{P}_{\mathcal{Z},\mathcal{N}_i} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \beta_i \end{pmatrix} - s_i \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\overline{I} & 0 \\ 0 & 0 & v_i \end{pmatrix} \geq \overline{S}_i, \quad i = 1, \ldots, M, \\
&\quad s_i \geq 0, i = 0, \ldots, M, \quad \sum_{i=1}^{M} \overline{S}_i \geq 0,
\end{aligned}$$

with $\tilde{P}_{\mathcal{Z},\mathcal{N}_i}$ such that

$$\overline{x}_{\mathcal{N}_i}^\top \left(\sum_{j \in \mathcal{N}_i} \overline{P}_{\mathcal{Z},\mathcal{N}_j} \right) \overline{x}_{\mathcal{N}_i} = x_{\mathcal{N}_i}^\top \tilde{P}_{\mathcal{Z},\mathcal{N}_i} x_{\mathcal{N}_i}.$$

Proof. This proof consists of two parts. First, the RPI condition is posed as an LMI condition. Then the constraints are included in the minimization, similar to lemma 24.

Part I: The RPI condition is given by

$$\begin{aligned} ((A+BK_t)x+Ew)^\top P_{\mathcal{Z}}((A+BK_t)x+Ew) \leq 1, \\ \forall x^\top P_{\mathcal{Z}} x \leq 1 \wedge w_i^\top w_i \leq v_i, i = 1,\dots,M. \end{aligned}$$

By applying the S-procedure this can be written as

$$\begin{aligned} 0 \leq & 1 - ((A+BK_t)x+Ew)^\top P_{\mathcal{Z}}((A+BK_t)x+Ew) \\ & - s_0(1 - x^\top P_{\mathcal{Z}} x) - \sum_{i=1}^{M} s_i(v_i - w_i^\top w_i), \\ & s_i \geq 0, i = 0,\dots,M, \end{aligned}$$

which can be written as the following matrix inequality

$$\begin{pmatrix} -(A+BK_t)^\top \sum_{i=1}^{M} \overline{P}_{\mathcal{Z},\mathcal{N}_i}(A+BK_t) & -(A+BK_t)^\top \sum_{i=1}^{M} \overline{P}_{\mathcal{Z},\mathcal{N}_i} E & 0 \\ -E \sum_{i=1}^{M} \overline{P}_{\mathcal{Z},\mathcal{N}_i}(A+BK_t) & -E^\top \sum_{i=1}^{M} \overline{P}_{\mathcal{Z},\mathcal{N}_i} E & 0 \\ 0 & 0 & \sum_{i=1}^{M} \beta_i \end{pmatrix}$$

$$-s_0 \begin{pmatrix} -\sum_{i=1}^{M} \overline{P}_{\mathcal{Z},\mathcal{N}_i} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sum_{i=1}^{M} \beta_i \end{pmatrix} - \sum_{i=1}^{M} s_i \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\overline{I} & 0 \\ 0 & 0 & v_i \end{pmatrix} \geq 0,$$

$$s_i \geq 0, i = 0,\dots,M,$$

with $\sum_{i=1}^{M} \overline{P}_{\mathcal{Z},\mathcal{N}_i} = P_{\mathcal{Z}}$. A sufficient condition for this LMI is given by the following local coupled LMIs

$$\begin{pmatrix} -(A+BK_t)^\top \overline{P}_{\mathcal{Z},\mathcal{N}_i}(A+BK_t) & -(A+BK_t)^\top \overline{P}_{\mathcal{Z},\mathcal{N}_i}E & 0 \\ -E^\top \overline{P}_{\mathcal{Z},\mathcal{N}_i}(A+BK_t) & -E^\top \overline{P}_{\mathcal{Z},\mathcal{N}_i}E & 0 \\ 0 & 0 & \beta_i \end{pmatrix}$$
$$-s_0 \begin{pmatrix} -\overline{P}_{\mathcal{Z},\mathcal{N}_i} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \beta_i \end{pmatrix} - s_i \begin{pmatrix} 0 & 0 & 0 \\ 0 & -\overline{I} & 0 \\ 0 & 0 & v_i \end{pmatrix} \geq \overline{S}_i,$$
$$s_i \geq 0, i = 0, \ldots, M, \quad \sum_{i=1}^{M} \overline{S}_i \geq 0.$$

Part II: In the second part the constraints are included in the optimization, in order to lead to small constraint tightening. Due to the coupled local RPI set, we define the matrix $\tilde{P}_{\mathcal{Z},\mathcal{N}_i}$ such that

$$\overline{x}_{\mathcal{N}_i}^\top \left(\sum_{j \in \mathcal{N}_i} \overline{P}_{\mathcal{Z},\mathcal{N}_j} \right) \overline{x}_{\mathcal{N}_i} = x_{\mathcal{N}_i}^\top \tilde{P}_{\mathcal{Z},\mathcal{N}_i} x_{\mathcal{N}_i}.$$

Analogous to lemma 24 we can apply the Schur complement to get sufficient LMI conditions for the set size with respect to the constraint set.

$$\begin{pmatrix} \gamma_i & h_{\mathcal{N}_i,j}^\top \\ h_{\mathcal{N}_i,j} & \tilde{P}_{\mathcal{Z},\mathcal{N}_i} \end{pmatrix} \geq 0, \quad \begin{pmatrix} \rho_i & l_{i,j}^\top K_{t,\mathcal{N}_i} \\ K_{t,\mathcal{N}_i}^\top l_{i,j} & \tilde{P}_{\mathcal{Z},\mathcal{N}_i} \end{pmatrix} \geq 0.$$

□

Remark 27. *Alternatively, the size of the RPI set can be directly minimized, by minimizing* $-\sum_{i=1}^{M} \det \log(P_{\mathcal{N}_i})$*. Unlike the previous optimization problem, the controller* K_t *is assumed to be known. The parameter* s_i *can be used as a local optimization variable, but* $s_0 \in (0,1)$ *needs to be set a priori, which plays the same role as* λ *in lemma 24.*

With this optimization problem we have two competing procedures to compute a RPI set: lemma 24 and lemma 26. The first procedure with polytopic disturbances can be used to compute both local RPI sets $P_{\mathcal{Z},i}$ and a corresponding stabilizing controller $K_{t,\mathcal{N}_i}$. The second procedure with ellipsoid constrained disturbances, needs a given stabilizing controller $K_{t,\mathcal{N}_i}$

to compute local RPI sets $P_{\mathcal{Z},\mathcal{N}_i}$, which are less conservative due to the neighboring coupling in the local RPI sets. By executing both procedures successively, we can combine the advantages to compute both stabilizing controller $K_{t,\mathcal{N}_i}$ and RPI sets $P_{\mathcal{Z},\mathcal{N}_i}$ with coupling to neighbors. For local scalar disturbances $w_i \in \mathbb{R}$, the polytopic and ellipsoid constraints $\mathcal{W}_i$ are equivalent.

Robust Distributed Model Predictive Control with RPI Tubes

Now we consider the Robust DMPC with RPI tubes [MayneSeronRaković05]. In particular all the offline computations including the robust constraint tightening and the online operation are summarized.
We first compute the stabilizing controller K_t and RPI set $\mathcal{Z}$ with lemma 24 and/or lemma 26. To distinguish the two we refer to the RPI set $\mathcal{Z}_i$ from lemma 24 as option 1 and the coupled RPI set $\mathcal{Z}_{\mathcal{N}_i}$ from lemma 26 as option 2. Now we have to compute the tightened state and input constraints with respect to the RPI tube. The constraints are tightened with the support function $\sigma_{\mathcal{Z}}$ [ConteEtAl12c]. The tightened local constraints

$$\overline{\mathcal{X}}_{\mathcal{N}_i} = \{x_{\mathcal{N}_i} | H_{\mathcal{N}_i} x_{\mathcal{N}_i} \leq \overline{h}_{\mathcal{N}_i}\}, \quad \overline{\mathcal{U}}_i = \{u_i | L_i u_i \leq \overline{l}_i\},$$

can be computed with the local support function $\sigma_{\mathcal{Z},\mathcal{N}_i}$

$$\overline{h}_{\mathcal{N}_i,j} = h_{\mathcal{N}_i,j} - \sigma_{\mathcal{Z},\mathcal{N}_i}(H_{\mathcal{N}_i,j}^\top), \quad i = 1,\ldots,M \quad j = 1,\ldots,n_{x_i} \tag{2.17}$$

$$\overline{l}_{i,j} = l_{i,j} - \sigma_{\mathcal{Z},\mathcal{N}_i}(K_{t,\mathcal{N}_i}^\top L_{i,j}^\top), \quad i = 1,\ldots,M \quad j = 1,\ldots,n_{u_i}. \tag{2.18}$$

Due to the structure of the constraints and the RPI set, the support function can be evaluated. For option 1 the support function is given by

$$\sigma_{\mathcal{Z},\mathcal{N}_i}(G_{\mathcal{N}_i,j}^\top) = \sup_{\overline{x}_{\mathcal{N}_i} \in \mathcal{Z}} G_{\mathcal{N}_i,j}^\top x_{\mathcal{N}_i} = |P_{\mathcal{Z},\mathcal{N}_i}^{-1} G_{\mathcal{N}_i,j}|$$

and for option 2 we have

$$\sigma_{\mathcal{Z},\mathcal{N}_i}(G_{\mathcal{N}_i,j}^\top) = \sup_{\overline{x}_{\mathcal{N}_i} \in \mathcal{Z}} G_{\mathcal{N}_i,j}^\top x_{\mathcal{N}_i} = |\tilde{P}_{\mathcal{Z},\mathcal{N}_i}^{-1} G_{\mathcal{N}_i,j}|.$$

The tightened distributed terminal set $\overline{\mathcal{X}}_f$ can be obtained by solving the optimization problem in lemma 10 and by using the tightened input and states constraints instead of the nominal constraints to compute the terminal

set size in (2.14).
For the initialization of the local RPI sets, any $\beta_i(0)$ that satisfies $\beta_i \geq 0$ and $\sum_{i=1}^{M} \beta_i \leq 1$ (eg. $\beta_i = 1/M$) can be chosen.
The optimization problem that needs to be solved in each time step is given by

$$\begin{aligned} \min_{z(0),v,z} & \sum_{k=0}^{N-1} \sum_{i=1}^{M} l_i(z_i(k),v_i(k)) + V_{f_i}(z_i(N)) \\ \text{st. } & z(0) \in x \bigoplus \mathcal{Z}(\beta) \\ & z_i(k+1) = A_{\mathcal{N}_i} z_{\mathcal{N}_i}(k) + B_i v_i(k), \quad k = 0,\dots,N-1 \\ & z_{\mathcal{N}_i}(k) \in \overline{\mathcal{X}}_{\mathcal{N}_i}, \quad v_i(k) \in \overline{\mathcal{U}}_i, \quad k = 0,\dots,N-1 \\ & z_i(N) \in \overline{\mathcal{X}}_{f_i}(\alpha_i) \end{aligned} \tag{2.19}$$

For option 1 the update for β is given by

$$\beta_i(t) = (x_i(t) - z_i^*(t|t-1))^\top P_{\mathcal{Z}_i}(x_i(t) - z_i^*(t|t-1)), \tag{2.20}$$

and for option 2 by

$$\beta_i(t) = (x_{\mathcal{N}_i}(t) - z_{\mathcal{N}_i}^*(t|t-1))^\top P_{\mathcal{Z},\mathcal{N}_i}(x_{\mathcal{N}_i}(t) - z_{\mathcal{N}_i}^*(t|t-1)). \tag{2.21}$$

This update ensures recursive feasibility as outlined in theorem V.1 in [ConteEtAl13].
As a summary the following operations need to executed offline.

Offline distributed synthesis - terminal cost/set with RPI modification
1. Stabilizing Controller: solve LMI in lemma 24 by distributed optimization
   → each system gets $P_{\mathcal{Z}_i}$,$K_{t,\mathcal{N}_i}$
  option 2: solve LMI in lemma 26 by distributed optimization using $K_{\mathcal{N}_i}$
   → each system gets $P_{\mathcal{Z},\mathcal{N}_i}$
2. Tighten constraints: compute tightened $\overline{\mathcal{X}}$ and $\overline{\mathcal{U}}$ with (2.17) and (2.18)
   → each system gets $\overline{\mathcal{X}}_{\mathcal{N}_i}, \overline{\mathcal{U}}_i$
3. Terminal cost: solve LMI (2.10), (2.12) by distributed optimization
   → each system gets $K_{\mathcal{N}_i}, \Gamma_{\mathcal{N}_i}, P_{f_i}$
4. Terminal cost II: solve (2.11), (2.13) by distributed optimization
   → each system gets $\gamma_{\mathcal{N}_i}$,p_{f_i}
5. Terminal set size: solve LP (2.14) with tightened constraints $\overline{\mathcal{X}}, \overline{\mathcal{U}}$
   → each system gets $\alpha_{\max}$
6. Init Terminal: initialize α_i, such that $\sum_{i=1}^{M} \alpha_i \leq \alpha_{\max}$ (e.g. $\alpha_i = \alpha_{\max}/M$)
7. Init RPI: initialize β_i such that $\sum_{i=1}^{M} \beta_i \leq 1$ (e.g. $\beta_i = 1/M$)

For the closed-loop operation we have the following operations in each time step:

Online Robust DEMPC-RPI, execute at every time step
1. Each system $i \in \{1,..,M\}$ measures local state x_i^t
2. Each system $i \in \{1,..,M\}$ updates RPI size β_i with (2.20)/(2.21) and x^t
3. Solve RDEMPC problem (2.19) via distributed optimization
4. Each system $i \in \{1,..,M\}$ applies control input
   $u_i^t = v_i^*(0) + K_{t,\mathcal{N}_i}(x_{\mathcal{N}_i}^t - z_{\mathcal{N}_i}^*(0))$
5. Each system $i \in \{1,..,M\}$ updates local terminal set:
   $\alpha_i^{t+1} = \alpha_i^t + z_{\mathcal{N}_i}(N+t|t)^{*\top}\Gamma_{\mathcal{N}_i} z_{\mathcal{N}_i}(N+t|t)^*$

From a computational point of view, the optimization over the initial state increases the complexity. In the case of coupled local RPI sets ($P_{\mathcal{Z},\mathcal{N}_i}$) we have quadratic constraints between neighboring systems, which excludes many distributed optimization methods, that are tailored to linear coupling constraints between neighbors, see chapter 3.

Robust Distributed Model Predictive Control with growing Tubes

For the robust DMPC with growing tubes [ChisciRossiterZappa01], we can use lemma 24 to obtain K_t via distributed optimization. Alternatively any stabilizing K_t can be used, since the RPI set $\mathcal{Z}$ is not explicitly used.
As a next step the local constraints need to be tightened over the prediction horizon. We again assume polytopic state and input constraints

$$\mathcal{X} = \{x|Hx \leq h\}, \quad \mathcal{U} = \{u|Lu \leq l\}.$$

For the constraint tightening we define a k-step support function $\sigma_{\mathcal{W}}(a,k)$ [ConteEtAl13]

$$\begin{aligned}\sigma_{\mathcal{W}}(a,k) = &\sup_{w \in \mathcal{W}^k} a^\top y(k)\\ &\text{st. } y(0) = 0,\\ &\quad y(l+1) = (A+BK_t)y(l) + w(l), \quad l = 0,\dots,k-1.\end{aligned}$$

Then the tightened constraints

$$\overline{\mathcal{X}}_k = \{x|Hx \leq \bar{h}_k\}, \quad \overline{\mathcal{U}}_k = \{u|Lu \leq \bar{l}_k\},$$

can be computed with

$$\bar{h}_{k,j} = h_j - \sigma_{\mathcal{W}}(H_j^\top,k), \qquad j = 1,\dots,n_x, \tag{2.22}$$
$$\bar{l}_{k,j} = l_j - \sigma_{\mathcal{W}}(K_t^\top L_j^\top,k), \qquad j = 1,\dots,n_u. \tag{2.23}$$

For polytopic disturbances $\mathcal{W}$, the computation of the k-step support function is a distributed LP and is equivalent to a distributed MPC problem.
The nominal terminal set $\mathcal{X}_f$ is obtained by solving the optimization problem in lemma 10 and determining the size α with (2.14).
The tightened terminal set $\overline{\mathcal{X}}_f$ can then be computed by tightening the nominal terminal set $\mathcal{X}_f$ with

$$\overline{\mathcal{X}}_f = \mathcal{X}_f \ominus \left(\bigoplus_{j=0}^{N-1} (A+BK_t)^j E\mathcal{W} \right). \tag{2.24}$$

In [AlvaradoEtAl10] a less conservative terminal set $\overline{\mathcal{X}}_f$ is computed. Consider the tightened set of disturbances

$$\mathcal{W}_N' = (A+BK_t)^{N-1} E\mathcal{W}$$

and the tightened state and input constraints $\overline{\mathcal{X}}_N, \overline{\mathcal{U}}_N$. Then the size of the terminal set $\overline{\mathcal{X}}_f$, needs to be such that

$$\begin{aligned} \overline{\mathcal{X}}_f \subseteq \overline{\mathcal{X}}_N, \quad K\overline{\mathcal{X}}_f \subseteq \overline{\mathcal{U}}_N, \\ (A+BK)\overline{\mathcal{X}}_f \bigoplus \mathcal{W}'_N \subseteq \overline{\mathcal{X}}_f. \end{aligned} \tag{2.25}$$

The first two inequalities are included in the usual computation of the terminal set size (2.14). The third condition can always be satisfied by choosing a large enough prediction horizon N. Alternatively, for a fixed smaller N, this condition could be enforced by increasing Q in the Lyapunov decrease inequality (2.4) in the computation of the terminal controller. This condition cannot be satisfied by simply decreasing the size of the terminal set.

In the following, we will refer to this choice of terminal set size (2.25) as option 2 and the usual tightening in (2.24) as option 1. The optimization problem that needs to be solved in each step is given by

$$\begin{aligned} &\min_{v,z} \sum_{k=0}^{N-1} \sum_{i=1}^{M} l_i(z_i(k),v_i(k)) + V_{f_i}(z_i(N)) \qquad (2.26)\\ &\text{st. } z(0) = x \\ &\quad x_i(k+1) = (A_{\mathcal{N}_i} + B_i K_{t,\mathcal{N}_i}) z_{\mathcal{N}_i}(k) + B_i v_i(k), \quad k = 0,\dots,N-1 \\ &\quad (z(k), K_t z(k) + v(k)) \in \overline{\mathcal{X}}_k \times \overline{\mathcal{U}}_k, \quad k = 0,\dots,N-1 \\ &\quad z_i(N) \in \overline{\mathcal{X}}_{f_i}(\alpha_i) \end{aligned}$$

As a summary the following operations need to executed offline.

Offline distributed synthesis - terminal cost/set with growing tubes

1. Stabilizing Controller: solve LMI in lemma 24 by distributed optimization
   $\rightarrow$ each system gets $K_{t,\mathcal{N}_i}$
2. Tighten constraints: compute tightened constraints with (2.22) and (2.23)
   $\rightarrow$ each system gets $\overline{\mathcal{X}}_{\mathcal{N}_i,k}, \overline{\mathcal{U}}_{i,k}, k = 1, \dots, N-1$
3. Terminal Cost: solve LMI (2.10), (2.12) by distributed optimization
   $\rightarrow$ each system gets $K_{\mathcal{N}_i}, \Gamma_{\mathcal{N}_i}, P_{f_i}$
4. Terminal Cost II: solve (2.11), (2.13) by distributed optimization
   $\rightarrow$ each system gets $\gamma_{\mathcal{N}_i}, p_{f_i}$
5. Tightened terminal set size $\overline{\alpha}_{\max}$:
   Option 1: solve LP (2.14) with nominal constraints $\rightarrow \alpha_{\max}$
   tighten $\overline{\alpha}_{\max}$ with (2.24)
   Option 2: solve LP (2.14) with tightened constraints $\overline{\mathcal{X}}_N, \overline{\mathcal{U}}_N \rightarrow \overline{\alpha}_{\max}$
   increase N such that (2.25) is satisfied
6. Init Terminal: initialize α_i, such that $\sum_{i=1}^{M} \alpha_i \leq \alpha_{\max}$ (e.g. $\alpha_i = \alpha_{\max}/M$)

For the closed-loop operation, we have the following operations in each time step:

Online Robust DEMPC-II, execute at every time step

1. Each system $i \in \{1,..,M\}$ measures local state x_i^t
2. Solve RDEMPC problem (2.26) by distributed optimization
3. Each system $i \in \{1,..,M\}$ applies control input $u_i = v_i^*(0) + K_{t,\mathcal{N}_i} x_{\mathcal{N}_i}$
4. Each system $i \in \{1,..,M\}$ updates local terminal set:
   $\alpha_i^{t+1} = \alpha_i^t + z_{\mathcal{N}_i}(N+t|t)^{*\top} \Gamma_{\mathcal{N}_i} z_{\mathcal{N}_i}(N+t|t)^*$

For this approach the robust modification does not increase the computational demand, which makes it a very attractive add-on.

3 Distributed Optimization for Distributed Model Predictive Control

In this chapter we discuss different distributed algorithms to solve the optimization problem arising in distributed MPC. The first section gives an overview of distributed optimization algorithms. In the second section the alternating direction method of multipliers (ADMM) and its application to DMPC are discussed.

3.1 Overview

This section gives an overview of different distributed optimization methods and algorithms, that can be used to solve the optimization problem arising in distributed MPC. As mentioned in section 2.1 we consider iterative DMPC, which means that the global MPC optimization problem has to be solved with distributed optimization. In chapter 2 the corresponding structure of the MPC optimization problem has been derived. For DEMPC without terminal constraints the resulting optimization problem is a distributed quadratic program (QP). If terminal constraints or RPI tubes are used, we have additional quadratic constraints, which increases the computational complexity to distributed quadratically constraint quadratic programs (QCQP).
In the following we discuss various distributed optimization methods based on primal and dual decomposition with respect to their application to DMPC. In [Boyd07] a short description of primal and dual decomposition methods is given and a good overview over convex optimization methods can be found in [BoydVandenberghe04].

Primal Decomposition

In the primal decomposition the MPC optimization problem is distributed among the subsystems by optimizing the local inputs u_i independently and iteratively exchanging information. In [StewartEtAl10] this method was applied to DMPC by combining the locally optimized inputs u_i in a Jacobi

iteration. For linear system dynamics with convex cost and constraints the convex combination of the respective optimal inputs decreases the global objective function and ensures constraint satisfaction due to the convexity. This iteration converges to the Pareto optimum and ensures constraint satisfaction with an arbitrary number of iterations.
One disadvantage of this method lies in the nature of the local optimization problems, that needs to be solved in each iteration. The local input minimizes the global cost, which requires knowledge of the models, states and stage cost of all subsystems affected within the prediction horizon. This means that this method scales badly, since the information exchange is required with non-neighboring subsystems, which makes it less appropriate for large scale distributed systems.
Another limitation is the computation of a feasible solution, by only optimizing one input. Especially for DEMPC without terminal constraints we cannot give an initially feasible solution and thus we cannot guarantee recursive feasibility. This is the reason why primal optimization and also the non-iterative approaches are not further considered and we focus on dual decomposition.

Dual Decomposition

In dual decomposition the central optimization problem is solved by decomposing the optimization problem into local optimization problems that are coupled by linear (in-)equality constraints. A good introduction to distributed dual optimization can be found in [BoydEtAl11].

Introduction to Distributed Dual Optimization

Since several different dual decomposition algorithms are discussed, the common framework for all of them is first introduced. Let us assume a linearly constrained convex optimization problem

$$
\begin{aligned}
p^* = \min_{z} & f(z) \\
\text{st. } & A_{eq} z - b_{eq} = 0, \\
& A_{ineq} z - b_{ineq} \leq 0,
\end{aligned}
$$

where $f(z)$ is a strict convex function. For this optimization problem the Lagrange function $\mathcal{L}$ is defined as

$$\mathcal{L}(z,\lambda,\nu) = f(z) + \nu^\top (A_{eq}z - b_{eq}) + \lambda^\top (A_{ineq}z - b_{ineq}),$$

where $\nu \in R^{m_{\text{eq}}}, \lambda \in R_{\geq 0}^{m_{\text{ineq}}}$ are the dual variables. The dual function g is then defined as

$$g(\lambda,\nu) = \min_{x \in \mathcal{R}^n} \mathcal{L}(x,\lambda,\nu),$$

and the dual optimization problem is given by

$$d^* = \max_{\lambda \geq 0,\nu} g(\lambda,\nu).$$

The most important result in this context is strong duality:

Theorem 28. *([BoydVandenberghe04]) Strong Duality: If the primal optimization problem is convex and the constraint qualifier condition (CQC) holds, then $d^* = p^*$.*

For linear constraints the constraint qualifier condition always holds and for more general convex constraints $f_i(z) \leq 0$ (e.g. quadratic constraints) the slater condition/strong feasibility is a sufficient condition. The strong duality property is also equivalent to the saddle point property:

$$\min_z \max_{\lambda \geq 0,\nu} \mathcal{L}(z,\lambda,\nu) = \max_{\lambda \geq 0,\nu} \min_z \mathcal{L}(z,\lambda,\nu).$$

With these properties, algorithms to solve convex optimization problems can be discussed. One of the fundamental algorithms in this context is the dual-ascent algorithm, which converges to the saddle point, by iteratively minimizing with respect to the primal variable and making a projected gradient (ascent) step in the dual variables:

Dual gradient ascent

$z^{k+1} = \arg\min_z \mathcal{L}(z,\lambda^k,\nu^k)$

$\nu^{k+1} = \nu^k + \alpha_k(A_{eq}z^{k+1} - b_{eq})$

$\lambda^{k+1} = \mathcal{P}_{\geq 0}\left(\lambda^k + \alpha_k(A_{ineq}z^{k+1} - b_{ineq})\right)$

Here α_k denotes the step size and $\mathcal{P}_{\geq 0}$ the projection.

Distributed Dual Gradient

In DMPC we have separable cost functions $f(z) = \sum_{i=1}^{M} f_i(z_i)$ and structured constraints A_{eq}, A_{ineq}, which enables a distributed computation of the corresponding primal and dual update. Based on this a distributed dual gradient algorithm [NecoaraNedich15] can be formulated.

Distributed Dual Gradient Algorithm

$z^{k+1} = \arg\min_z \mathcal{L}(z,\lambda^k,\nu^k)$

$\nu^{k+1} = \nu^k + W_\nu^{-1}(A_{eq}z^{k+1} - b_{eq})$

$\lambda^{k+1} = \mathcal{P}_{\geq 0}\left(\lambda^k + W_\lambda^{-1}(A_{ineq}z^{k+1} - b_{ineq})\right)$

This algorithm achieves a linear convergence rate by using block diagonal matrices W_ν, W_λ for the step size, which requires a strictly convex local cost.

Proximal Center based Decomposition

A difficulty of the distributed dual gradient method is the requirement of a strictly convex cost, which also affects the convergence rate. The proximal center method uses a smoothing technique for the Lagrangian, that improves the convergence and keeps the distributed structure. The smoothed Lagrangian is given by

$$\mathcal{L}_c(z,\nu,\lambda) = \sum_{i=1}^{M} f_i(z_i) + \nu^\top(A_{\text{eq}}z - b_{\text{eq}}) + \lambda^\top(A_{\text{ineq}}z - b_{\text{ineq}}) + c\sum_{i=1}^{M} \|z_i\|^2,$$

with the smoothing constant $c > 0$. This idea of using a prox term to smooth the Lagrangian is based on [Nesterov05] and ensures strict convexity, which is a pre-requisit for fast gradient algorithms. With this smoothed Lagrangian the distributed optimization consists of a minimization with respect to the primal variables and a fast gradient step in the dual variables. Compared to the dual gradient algorithm, a faster convergence can be reached. In order to converge to the true optimum the smoothing constant c has to decrease ([NecoaraSuykens08] Thm 3.6).
The proximal center based method has two benefits: It increases the convexity, which improves the convergence, and it can treat linear coupling

inequality constraints with dual variables. In [NecoaraDoanSuykens08] this method was used for distributed MPC, in [NecoaraEtAl09] it was extended to nonlinear systems by using sequential convex programming and in [Tran-DinhNecoaraDiehl16] it was extended to inexact optimization.

Alternating Direction Method of Multipliers

The alternating direction method of multipliers (ADMM) uses the augmented Lagrangian to increase the convexity. Here the primal variables are split in y, z and the primal minimization is done in an alternating fashion, i.e. first minimizing y and then minimizing z. This method can only treat equality constraints with dual variables. The corresponding augmented Lagrangian $\mathcal{L}_\rho$ is given by

$$\mathcal{L}_\rho(y,z,\lambda) = f(z,y) + \lambda^\top(A_z z + A_y y - b) + \frac{\rho}{2}\|A_z z + A_y y - b\|_2^2.$$

The algorithm then consists of an alternating primal minimization and a dual update.

Alternating Directional Method of Multiplier

$y^{k+1} = \arg\min_y \mathcal{L}_\rho(y,z^k,\lambda^k)$

$z^{k+1} = \arg\min_z \mathcal{L}_\rho(y^{k+1},z,\lambda^k)$

$\lambda^{k+1} = \lambda^k + \rho(A_z z^{k+1} + A_y y^{k+1} - b)$

The convergence of this algorithm is studied in detail in [BoydEtAl11]. In [ConteEtAl12a] and [KögelFindeisen12] different possibilities are presented to write the distributed MPC optimization problem in this fashion. The variable z is used as a global variable and the equality constraint is used to demand consistency between different predictions, while the state and input constraints are included in the primal update. If the tunable weight ρ is chosen properly, this method can achieve reasonable results with few iterations.

Summary

All the presented dual algorithms can be implemented fully distributed, but require a strict convex Lagrangian. If the original optimization problem is only convex, we can either use the smoothed Lagrangian or the augmented Lagrangian to ensure strict convexity. One of the major differences between the augmented Lagrangian and the smoothed Lagrangian is that the prox term introduces an error, while the augmentation does not change the minimum. On the other side, the smoothed Lagrangian enables the usage of fast gradient methods, that can give an upper bound on the number of required iterations, while such estimates are hard to obtain for ADMM. Another difference is how the constraints are handled. In the ADMM approach, all the constraints are included in the local optimization, while only equality constraints can be enforced with dual variables. On the other side, for the dual fast gradient the coupling constraints are enforced with dual variables, while the linear local constraints can be either included in the primal optimization or with dual variables.

3.2 Alternating Direction Method of Multipliers

In this section the Alternating Direction Method of Multipliers (ADMM) and its application to distributed MPC is discussed in detail. First, the optimization problem is formulated in such a way, that it fits into the ADMM framework. Then the corresponding online ADMM iteration is described in detail and properties of the algorithm are discussed.

3.2.1 Rewrite central optimization problem

We first write the optimization problem with only coupling equality constraints, to fit the ADMM framework. To this end we consider the formulation used in [ConteEtAl12a]. The variable z contains a copy of all the global predictions u, x over the whole prediction horizon N. The variable y consists of local predictions y_i, $i = 1, \ldots, M$. Each local prediction y_i consists of the state trajectory of the neighboring systems as predicted by subsystem i, i.e. $x^i_{\mathcal{N}_i}$, and the predicted input u_i over the prediction horizon N. This way the same variable $x_i(k)$ is contained as independent optimization variables in y_j, $j \in \mathcal{N}_i$ and in z. To ensure that all the optimization variables have the same value, a consistency constraint

$$E_i z = y_i$$

is added, that enforces $x_j^i = x_j$. With this notation we can rewrite the central optimization problem for DEMPC with terminal costs/sets as

$$\min_{y,z} \sum_{i=1}^{M} J_i(y_i) = \sum_{i=1}^{M} \sum_{k=0}^{N-1} l_i(x_i^i(k), u_i(k)) + V_{f_i}(x_i^i(N)) \tag{3.1}$$
$$\text{st. } y_i \in \mathcal{Y}_i(x_i), \quad y_i = E_i z,$$

where $\mathcal{Y}_i(x_i)$ is a convex set, that enforces all the local constraints. The local constraint set is given by

$$\mathcal{Y}_i(x_i) = \Big\{ y_i | x_i(0) = x_i,\, x_i^i(k+1) = A_{\mathcal{N}_i} x_{\mathcal{N}_i}^i(k) + B_i u_i(k), \ldots$$
$$x_i^i(N) \in \mathcal{X}_{f_i},\, x_{\mathcal{N}_i}^i(k) \in \mathcal{X}_{\mathcal{N}_i},\, u_i(k) \in \mathcal{U}_i \Big\}.$$

In this formulation we can also consider nonlinear dynamics and ellipsoidal terminal constraints within the local constraint $\mathcal{Y}_i$.

3.2.2 Distributed Algorithm

With this formulation we can now formulate the ADMM iteration. As mentioned this algorithm relies on the augmented Lagrangian $\mathcal{L}_\rho$:

$$\mathcal{L}_\rho(y,z,\lambda) = \sum_{i=1}^{M} \mathcal{L}_{i,\rho}(y_i,z,\lambda_i) = \sum_{i=1}^{M} J_i(y_i) + \lambda_i^\top (y_i - E_i z) + \frac{\rho}{2} \| E_i z - y_i \|_2^2.$$

Only the simple consistency constraint is treated with dual variables, while the complex constraints $\mathcal{Y}_i$ are treated locally. The distributed ADMM iteration is given by the following algorithm:

Alternating Directional Method of Multiplier - Iteration

solve local optimization: $y_i^{k+1} = \arg \min_{y_i \in \mathcal{Y}_i(x_0^i)} \mathcal{L}_{i,\rho}(y_i, z^k, \lambda_i^k)$

communicate y_i^{k+1} to neighbors $j \in \mathcal{N}_i$

average global variable: $z_i^{k+1} = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} E_{ji}^\top (y_{ji}^{k+1} + \frac{1}{\rho} \lambda_{ji})$

communicate z_i^{k+1} to neighbors: $j \in \mathcal{N}_i$

update dual variable $\lambda_i^{k+1} = \lambda_i^k + \rho(y_i^{k+1} - E_i z^{k+1})$

Each iteration requires two communication steps and one local optimization step (y_i). The update of z is equivalent to an unconstrained minimization of the augmented Lagrangian, which can be shown by writing it as a quadratic program (QP) in z_i.

Convergence

The convergence of this algorithm is studied in detail in [BoydEtAl11]. If the unaugmented Lagrangian $\mathcal{L}$ has a saddle point and the cost $J_i(y_i)$ is closed, proper and convex, then the iteration converges asymptotically to a feasible solution that minimizes the global cost. The convergence proof relies on a Lyapunov function

$$V = \frac{1}{\rho}\|\lambda - \lambda^*\|_2^2 + \sum_{i=1}^{M} \rho \|E_i(z - z^*)\|_2^2$$

with

$$V^+ - V \leq -\rho\|r^+\| - \rho \sum_{i=1}^{M} \|E_i(z^+ - z)\|_2^2$$

and the residual $r = y - E_i z$. In practice, ADMM is able to achieve medium accuracy with few iterations, but then requires an increasing amount of iterations to achieve high accuracy.

3.2.3 Stopping Condition

Due to real time requirements and/or constraints on the communication, we might have to stop the online distributed iteration prior to computing the optimum. Therefore we consider the use of stopping conditions and investigate the properties of the resulting inexact solution. For ADMM, a simple stopping condition is

$$\|E_i z - y_i\|_\infty \leq \epsilon_z,$$

with the accuracy ϵ_z. In general, we are not able to give any prior bound to the number of required iterations, in order to satisfy the stopping condition. But the stopping condition can easily be checked online.

Prediction error

One important question is which bounds we can give on the prediction error. To this end we consider the prediction error w_i, with

$$w_i(k) = x_i^i(k+1) - (A_{\mathcal{N}_i} x_{\mathcal{N}_i}(k) + B_i u_i(k)),$$

where $x_{\mathcal{N}_i}$ is composed of x_j^j, $j \in \mathcal{N}_i$. From the constraint $y_i \in \mathcal{Y}_i(x_0)$ we know

$$x_i^i(k+1) = A_{\mathcal{N}_i} x_{\mathcal{N}_i}^i(k) + B_i u_i(k).$$

Thus we have

$$w_i(k) = A_{\mathcal{N}_i}(x_{\mathcal{N}_i}^i - x_{\mathcal{N}_i}).$$

Due to the stopping condition we know $\|x_j^i - x_j^j\|_\infty \leq 2\epsilon_z$ and thus

$$\|w_i(k)\|_\infty \leq 2 \sum_{j \in \overline{\mathcal{N}}_i} \|A_{ij}\|_\infty \epsilon_z.$$

This bound can be seen as a distributed version of the bounds derived in [FerrantiEtAl15]. In chapter 4 we will consider the closed-loop properties of using DEMPC with such inexact solutions.

4 Distributed Economic Model Predictive Control with inexact optimization

This chapter deals with the effect of inexact optimization on the closed-loop properties of DEMPC. In particular, we are interested in guaranteeing constraint satisfaction, stability and performance despite inexact optimization resulting from finite dual iterations.
The first section gives an overview over some existing results for inexact optimization and motivates a further investigation. In the second section, a robust MPC approach to inexact optimization for terminal constraint DEMPC is considered and recursive feasibility and stability guarantees are derived. Stability results for DEMPC without terminal constraints under inexact optimization are investigated in the third section . In the last section, performance guarantees for DEMPC under inexact minimization with and without terminal constraints are studied.

4.1 Overview

This section gives an overview over some existing results for MPC with inexact optimization. These methods are discussed in the context of distributed optimization and possible guarantees on feasibility and stability. Also the relevance of investigating such results is highlighted.

Motivation

We first establish the relevance of considering the results of inexact optimization and argue why it needs to be incorporated. One of the main advantages of MPC are the satisfaction of constraints, guaranteed stability and an improved economic performance. In chapter 2, we established corresponding theoretical results under the assumption that the underlying optimization problem is solved in each time step. Here the online optimization is accomplished with iterative dual distributed optimization algorithms, which converge asymptotically to the true optimum, see chapter 3. In real time

applications of distributed MPC, the optimum is in general not attained due to limited communication and computational resources.
For stabilizing MPC with primal optimization, few iterations are usually sufficient to stabilize the system and satisfy the constraints. Therefore the effect of inexact optimization can often be neglected. This is however not the case for DEMPC with dual optimization.
The usage of distributed dual optimization methods leads to constraint violations and errors in the predicted trajectories that only vanish asymptotically. This in turn complicates the analysis of MPC schemes, leads to constraint violations and can even cause loss of stability. This is why the deteriorating effects of inexact minimization should not be ignored in the context of DEMPC with dual optimization, particularly for DEMPC without terminal constraints.
With this in mind, the goal is to modify the optimization problem, such that we have sufficient stopping conditions on the online iterations, that guarantee feasibility, stability and performance with the current suboptimal solution. In particular, those stopping conditions should be easy to check and are preferably based directly on properties of the optimization, which enables an a priori upper bound on the number of needed iterations.
We will now introduce some existing results regarding suboptimality, stability and constraint satisfaction for MPC with inexact optimization.

Results for Primal Decomposition

We start by discussing suboptimality in primal decomposition due to its simplicity. The main reason for this is that the predicted state trajectories in the optimization are consistent with the dynamic equality constraint.

Feasibility implies Stability

For the primal decomposition, we can ensure feasibility of the predicted trajectories in each iteration step. In particular, if we consider terminal constraint MPC, the candidate solution is already both feasible and stabilizing. This feature was exploited in [StewartEtAl10] to use a primal decomposition algorithm, that guarantees feasibility and stability with any number of iterations.
So we can see that for primal decomposition, the issue of inexact optimization is less difficult. Note that we discussed in chapter 3 the main drawbacks of primal decomposition and why we use dual decomposition instead.

Stabilizing Linear MPC with projected gradient steps

In [RubagottiPatrinosBemporad14], a stabilizing terminal cost/set MPC with a projected dual gradient algorithm is used. Here the dynamic constraints are satisfied in each iteration step and only a violation ϵ in the state and input constraints need to be considered. To ensure both constraint satisfaction and recursive feasibility of the algorithm the constraints are tightened over the prediction horizon k with $\mathcal{Z}_k = (1 - k\epsilon)\mathcal{Z}$. This has similarities to the constraint tightening in robust MPC with growing tubes. The algorithm is able to guarantee a maximum constraint violation of ϵ with a fixed number of iterations, which in turn establishes feasibility and stability of the original problem. The main limitation of this method is the projected minimization, which makes it unsuitable for distributed optimization.

Suboptimal Distributed MPC without terminal constraint

In [GiselssonRantzer14] stabilizing distributed MPC without terminal constraints and with dual distributed optimization is investigated. The optimization problem uses an online iterative adaptive constraint tightening to achieve feasibility of the system trajectory, despite primal constraint violations in the optimization. To ensure stability despite suboptimality, a sufficient stopping condition based on a consistent candidate solution in the next time step is used.

MPC based on time splitting with adaptive constraint tightening

In [FerrantiEtAl15] a different adaptive constraint tightening is proposed and the optimization problem is split along the prediction horizon (time splitting) with an ADMM like approach. This algorithm is augmented with an adaptive constraint tightening, that chooses the accuracy ϵ and the corresponding tightening based on the candidate solution, i.e. such that the candidate is feasible with respect to the tightened constraints. By using a terminal constraint MPC, the satisfaction of the constraints can be used to establish stability by bounding the suboptimality. The key advantages of this method are both a parallelization of the computation due to the time splitting and the adaptive tightening that guarantees feasibility and stability with a finite number of iterations. In order to use fixed tightening as in [RubagottiPatrinosBemporad14], the parameters would need to be computed offline for each initial state ([FerrantiEtAl15] Remark 6).

Distributed Hierarchical MPC

In [DoanKeviczkyDe Schutter11] a distributed hierarchical MPC scheme is presented, that enables the computation of a feasible and stabilizing solution with finite iterations. The algorithm also uses adaptive tightening based on a slater vector. The feasibility is guaranteed with a constraint tightening and stability with a bound on the suboptimality.

Stabilization of inexact MPC

In [KögelFindeisen14] a framework to ensure recursive feasibility and stability for inexact MPC is proposed. This method takes the constraint violation in the dynamic constraint explicitly into account with a robust MPC approach. Compared to the previous approaches a constant tightening can be used, but the tightening of the constraints is significantly more conservative. In particular, the constraints are tightened in such a way, that the consistent candidate solution is feasible with respect to the tightened constraints. Stability is guaranteed by ensuring a decrease in the optimized cost and bounding the cost increase due to the error in the equality constraint.

Summary

From these different approaches we can see some common features, which are needed to guarantee feasibility and stability.
To guarantee feasibility with inexact dual optimization, a tightening of the constraint similar to robust MPC is inevitable. If we use a distributed dual optimization, the constraint tightening has to consider both the error in the dynamic constraint and the error in the constraint satisfaction.
We have seen, that ensuring recursive feasibility of the tightened optimization problem can be accomplished by adapting the tolerance and the constraint tightening. This adaptation however requires a certain amount of global communication and is thus not suited for large scale distributed systems. On the other hand, approaches considering a fixed tolerance lead to a constant but significantly larger constraint tightening.
With respect to the stability, this can be posed as a condition on the suboptimality η of the optimization problem. For EMPC, especially without terminal constraints, this issue seems largely unexplored and will also be investigated.
In conclusion, a suitable method should include stopping conditions on

the constraint violation ϵ and suboptimality η, that ensures feasibility and stability. Furthermore, guaranteeing recursive feasibility for the tightened optimization problem under inexact minimization is an important, nontrivial task. In the following a new method is presented, that ensure recurisve feasibility with a less conservative constraint tightening.

4.2 A Robust Model Predictive Control approach

In this section a robust MPC approach to ensure recursive feasibility and closed-loop stability with inexact minimization is presented. We consider constant constraint tightening, but compared to established results we use a different candidate solution. First, recursive feasibility with constraint tightening and an the new candidate solution is established. Then stability guarantees in dependence of suboptimality guarantees are discussed.

4.2.1 Constraint Violations - Recursive Feasibility

An important and difficult issue for DMPC is to guarantee recursive feasibility under inexact minimization resulting from finite dual iterations. To this end we will use methods similar to robust MPC and the growing tubes approach, as outlined in section 2.4, based on [ChisciRossiterZappa01, ConteEtAl13].

Pre-Stabilized Dynamics

We consider pre-stabilized dynamics

$$A_K = A + BK_t,$$

with a structured distributed stabilizing feedback K_t, computed using lemma 24. As mentioned in chapter 3 and section 4.1 the state and input trajectory (z,v) resulting from inexact dual optimization does not satisfy the dynamic constraint. Thus, to study feasibility we define the consolidated predicted state trajectory $\overline{x}$, that is consistent with the optimized input trajectory v and the system dynamics, i.e.

$$\begin{aligned} \overline{x}(k+1) &= A\overline{x}(k) + B(K_t\overline{x}(k) + v(k)) = A_K\overline{x}(k) + v(k), \\ \overline{x}(0) &= x. \end{aligned}$$

Recursive Feasibility under inexact minimization - Available Results

As outlined in section 4.1 there are two main approaches, to insure recursive feasibility despite inexact predictions. Both approaches use the same candidate solution $\overline{x}^+$, which consists of the consolidated predicted state trajectory $\overline{x}$ shifted by one time step and with the terminal controller K appended (the usual candidate solution). In [FerrantiEtAl15] the constraints are tightened, such that this consolidated trajectory is feasible with respect to the original constraint $\mathcal{X}$, $\mathcal{U}$. Recursive feasibility is then established by adapting the constraint tightening and thus the accuracy ϵ, such that the problem is feasible, based on the consolidated candidate solution $\overline{x}^+$ (slater vector). In order to use a fixed tightening, the parameters need to be computed offline for each initial state ([FerrantiEtAl15] Remark 6), which is impracticable.
In [KögelFindeisen14], a constant constraint tightening is used with the same consolidated candidate solution $\overline{x}^+$. To ensure recursive feasibility without adapting the constraint, the tightening of the constraints is significantly more conservative. In particular, the constraints are tightened in such a way, that the consolidated candidate solution $\overline{x}^+$ is feasible with respect to the tightened constraints. This problem is illustrated in figure 4.1 for a one dimensional system $x \in \mathbb{R}$. Here z is the inexact trajectory, the red lines correspond to the error in the prediction, $\overline{x}$ is the consolidated trajectory, X is the state constraint and $\overline{X}$ is the tightened state constraint.
The goal now is to use a fixed constraint tightening and ensure recursive feasibility, without using overly conservative approaches. To this end we use a different candidate solution and consider a relaxed optimization problem.

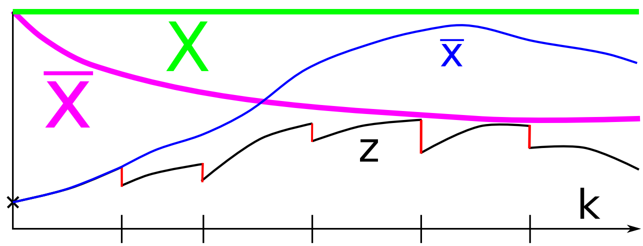

Figure 4.1: Illustration: inexact prediction z and consolidated trajectory $\overline{x}$.

Tightened Optimization Problem - Feasibility

Let the overall state and input constraint be denoted by

$$\mathcal{X} = \{x|Hx \leq h\}, \quad \mathcal{U} = \{u|Lu \leq l\}$$

and the tightened constraints by

$$\overline{\mathcal{X}}_k = \{x|Hx \leq \overline{h}_k\}, \quad \overline{\mathcal{U}}_k = \{u|Lu \leq \overline{l}_k\}.$$

The corresponding tightening will be discussed later. The tightened optimization problem, which will be solved in each time step, is then given by

$$\begin{aligned} &\min \sum_{i=1}^{M} \sum_{k=0}^{N-1} l_i(z_i(k), v_i(k)) + V_{f_i}(z_i(N)) \qquad (4.1) \\ \text{st. } &\|(A_{\mathcal{N}_i} + B_i K_{t,\mathcal{N}_i}) z_{\mathcal{N}_i}(k) + B_i v_i(k) - z_i(k+1)\|_\infty \leq \epsilon_{z_i,k}, \\ &z_{\mathcal{N}_i}(k) \in \overline{\mathcal{X}}_{\mathcal{N}_i,k}, \quad v_i(k) + K_{t,\mathcal{N}_i} z_{\mathcal{N}_i}(k) \in \overline{\mathcal{U}}_{i,k}, \\ &z_i(N) \in (1-\epsilon_f)\overline{\mathcal{X}}_{f_i}, \quad z_i(0) = x_i, \\ &i = 1, \ldots, M, \quad k = 0, \ldots, N-1, \end{aligned}$$

with the current measured local state x_i. This approach uses ideas from [FerrantiEtAl15] and directly considers the fact, that we cannot expect the dynamic constraint to be satisfied exactly.

Inexact Minimization

Given the optimization problem (4.1), we can define an inexact solution.

Assumption 29. *The inexact solution (z,v) to the optimization problem* (4.1) *satisfies:*

$$\begin{aligned} &\|(A_{\mathcal{N}_i} + B_i K_{t,\mathcal{N}_i}) z_{\mathcal{N}_i}(k) + B_i v_i(k) - z_i(k+1)\|_\infty \leq \epsilon_{z_i,k} + \epsilon_i \\ &H_{\mathcal{N}_i} z_{\mathcal{N}_i}(k) \leq \overline{h}_{i,k} + 1_p \epsilon_{x_i}, \quad L_i(v_i(k) + K_{t,\mathcal{N}_i} z_{\mathcal{N}_i}(k)) \leq \overline{l}_{i,k} + 1_q \epsilon_{u_i}, \\ &z(N) \in \overline{\mathcal{X}}_f, \quad z_i(0) = x_i \end{aligned}$$

Compared to the posed optimization problem, this allows a violation ϵ_i in the dynamic constraint, ϵ_{u_i}, ϵ_{x_i} in the input and state constraints and ϵ_f for the terminal constraint. Only the initial state constraint is assumed to be satisfied exactly, which can be accomplished by most algorithms by considering $z_i(0)$ as a fixed parameter instead of a variable. This bound can be guaranteed after finite iterations and a corresponding stopping condition can be checked distributedly online.

Tightened Constraints

Given assumption 29 we can discuss how to tighten the constraints in order to ensure constraint satisfaction and recursive feasibility of the algorithm. The error in prediction can be interpreted as a bounded unknown disturbance

$$x_i(k+1) = A_{\mathcal{N}_i} x_{\mathcal{N}_i}(k) + B_i u_i(k) + w_i(k), \quad w_i(k) \in \mathcal{W}_{i,k},$$

with $\mathcal{W}_{i,k} = \{w_i | \|w_i\|_\infty \leq \epsilon_{z_i,k} + \epsilon_i\}$ and $\mathcal{W}_k = \mathcal{W}_{1,k} \times \cdots \times \mathcal{W}_{M,k}$. This setup is opposite to the robust MPC setting, since the disturbance lies in the prediction and not the actual system behavior. To ensure recursive feasibility, the size of the relaxation decreases over the prediction horizon with

$$\epsilon_{z_i,k+1} = \epsilon_{z_i,k} - \epsilon_i. \tag{4.2}$$

The tightening of the state and input constraints uses the k-step support function $\sigma_{\mathcal{W}}(a,k)$:

$$\begin{aligned} \sigma_{\mathcal{W}}(a,k) = \sup_{w \in \mathcal{W}_1^k} \; & a^\top y(k), \\ \text{st. } & y(0) = 0, \\ & y(l+1) = A_K y(l) + w(l), \quad l = 1, \ldots, k-1. \end{aligned}$$

We consider $\mathcal{W}_1^k$ instead of $\mathcal{W}_1 \times \cdots \times \mathcal{W}_k$, which is slightly more conservative, but required for recursive feasibility (see proof of theorem 30). The tightened state and input constraints are computed with:

$$\begin{aligned} \bar{h}_{i,k,j} &= h_{i,j} - \sigma_{\mathcal{W}}(\overline{H}_{\mathcal{N}_i,j}^\top, k) - (k+1)\epsilon_{x_i}, \\ \bar{l}_{i,k,j} &= l_{i,j} - \sigma_{\mathcal{W}}(\overline{L_{i,j} K_{t,\mathcal{N}_i}}^\top, k) - (k+1)\epsilon_{u_i}. \end{aligned}$$

Here $\bar{h}_{i,k,j}$ denotes the j-th component of $\bar{h}_{i,k}$. This tightening consists of two factors. The term $(k+1)\epsilon_{x_i}$ is similar to the tightening used in [RubagottiPatrinosBemporad14] ,with $\mathcal{Z}_k = (1-k\epsilon)\mathcal{Z}$. The other term $\sigma_{\mathcal{W}}$ reflects the dynamic constraint violation and the corresponding tightening is similar as in [FerrantiEtAl15]. Compared to [KögelFindeisen14] the constraint tightening is significantly less conservative. The tightened constraints maintain the distributed structure and the k-step support function can be computed by distributed optimization, see section 2.4.

Tightened terminal set

The terminal cost V_f and the terminal controller $u = Kx$ need to satisfy the typical Lyapunov decrease condition, (2.4) and can be computed using lemma 10. The size of the terminal set $\overline{\mathcal{X}}_f = \{x | x^\top P_f x \leq \overline{\alpha}\}$ needs to be such that

$$\begin{aligned} \overline{\mathcal{X}}_f \subseteq \overline{\mathcal{X}}_N, \quad K\overline{\mathcal{X}}_f \subseteq \overline{\mathcal{U}}_N, \\ (A+BK)(\overline{\mathcal{X}}_f \bigoplus A_K^{N-1}\mathcal{W}) \subseteq (1-\epsilon_f)\overline{\mathcal{X}}_f, \end{aligned} \tag{4.3}$$

which can be computed analog to (2.14). The last condition is similar to [AlvaradoEtAl10] and can always be ensured by choosing a large enough prediction horizon N and a small enough ϵ_f.

Theorem 30. *Assume that the predicted state and input trajectories z, v satisfy assumption 29, and the terminal set $\overline{\mathcal{X}}_f$ satisfies (4.3), with $\epsilon_i, \epsilon_{z_i,k}, \epsilon_f, \epsilon_{u_i} \geq 0$ and $\epsilon_{z_i,k}$ according to (4.2). Then the consolidated trajectory $\overline{x}$ satisfies*

$$\overline{x}(k) \in \mathcal{X}, \quad u(k) = v(k) + K_t\overline{x}(k) \in \mathcal{U}.$$

Furthermore, the candidate solution $(\tilde{z},\tilde{v})$

$$\begin{aligned} \tilde{v}(k) &= v(k+1), k = 0, \ldots, N-2, \\ \tilde{v}(N-1) &= Kz(N) \\ \tilde{z}(0) &= x = z(1) + w, \quad w \in \mathcal{W}_1, \\ \tilde{z}(k) &= z(k+1) + A_K^k w, k = 0, \ldots, N-1, \\ \tilde{z}(N) &= (A+BK)\tilde{z}(N-1), \end{aligned}$$

based on the previous solution (z, v) and the current state x is a feasible solution to the optimization problem (4.1), which ensures recursive feasibility.

Proof. The proof is composed of 4 parts. First, constraint satisfaction of the consolidated trajectory is established. In the second part the candidate solution is constructed and we show satisfaction of the relaxed dynamic constraints. In the third part we show that this candidate solution satisfies

the tightened state and input constraints over the prediction horizon. Finally we show that this solution also satisfies the terminal constraint, and thus establishes recursive feasibility.
Part I: Show constraint satisfaction of consolidated trajectory.
The constraint satisfaction of the consolidated trajectory $\overline{x}$ follows directly from the construction of the constraint tightening (see Robust MPC). The disturbance considered in $\sigma_{\mathcal{W}}$ is slightly larger than $\mathcal{W}_1 \times \ldots \mathcal{W}_k$, due to the decrease $\epsilon_i \geq 0$ and (4.2). Thus we have:

$$\begin{aligned} H_{\mathcal{N}_i,j}\overline{x}_{\mathcal{N}_i}(k) &\leq H_{\mathcal{N}_i,j}z(k) + \sigma_{\mathcal{W}}(\overline{H}_{\mathcal{N}_i,j}^{\top},k) \\ &\leq \overline{h}_{i,k,j} + \sigma_{\mathcal{W}}(\overline{H}_{\mathcal{N}_i,j}^{\top},k) + \epsilon_{x_i} \leq h_{i,j}, \\ L_{i,j}(v_i(k) + K_{t,\mathcal{N}_i}\overline{x}_{\mathcal{N}_i}(k)) &\leq L_{i,j}(v_i(k) + K_{t,\mathcal{N}_i}z_{\mathcal{N}_i}(k)) + \sigma_{\mathcal{W}}(\overline{L_{i,j}K_{t,\mathcal{N}_i}}^{\top},k) \\ &\leq \overline{l}_{i,k,j} + \sigma_{\mathcal{W}}(\overline{L_{i,j}K_{t,\mathcal{N}_i}}^{\top},k) + \epsilon_{u_i} \leq d_{i,j}. \end{aligned}$$

Part II: Show that candidate sequence satisfies relaxed dynamic constraint.
We construct a candidate state and input trajectory $\tilde{u}$, $\tilde{z}$, that satisfies the relaxed equality constraint

$$\|(A_{\mathcal{N}_i} + B_i K_{t,\mathcal{N}_i})z_{\mathcal{N}_i}(k) + B_i v_i(k) - z_i(k+1)\|_\infty \leq \epsilon_{z_i,k}.$$

To this end we shift the previous solution by one time step and append the terminal controller K. We also have to add an error term, due to the previous constraint violation. With the measured state x_{meas} in the next time step, we have

$$\tilde{z}(0) = x_{\text{meas}} = z(1) + w, \quad w \in \mathcal{W}_1,$$

with z being the approximate solution to the optimization problem in the last time step according to assumption 29. With this we can set

$$\begin{aligned} \tilde{z}(k) &= z(k+1) + A_K^k w, \quad k = 0, \ldots, N-1, \\ \tilde{v}(k) &= v(k+1), \quad k = 0, \ldots, N-2. \end{aligned}$$

To show that this trajectory satisfies the relaxed dynamic constraint, we use

$$z(k+1) = A_K z(k) + Bv(k) + w_k, \quad w_k \in \mathcal{W}_k.$$

Now substituting $\tilde{z}$ we have

$$\begin{aligned}\tilde{z}(k) &= z(k+1) + A_K^k w = A_K z(k) + Bv(k) + A_K^k w + w_k \\ &= A_K(\tilde{z}(k-1) - A_K^{k-1} w) + B\tilde{v}(k-1) + A_K^k w + w_k \\ &= A_K \tilde{z}(k-1) + B\tilde{v}(k-1) + w_k,\end{aligned}$$

for $k = 1, \ldots, N-1$. Now we have the new equality constraint violation $\tilde{w}_{i,k-1} = w_{i,k}$, which satisfies

$$\|\tilde{w}_{i,k-1}\|_\infty \leq \epsilon_{z_i,k} + \epsilon_i = \epsilon_{z_i,k-1},$$

and thus the required bounds on the dynamic constraint over the prediction horizon. The terminal state of the candidate solution

$$\tilde{z}(N) = (A + BK)\tilde{z}(N-1), \quad \tilde{v}(N-1) = (K - K_t)\tilde{z}(N-1)$$

exactly satisfies the dynamic equality constraint, by using the terminal controller K.
Part III: Show that the candidate sequence satisfies state and input constraints.
For the state constraints we know that

$$H_{\mathcal{N}_i,j}\tilde{z}_{\mathcal{N}_i}(k) = H_{\mathcal{N}_i,j} z_{\mathcal{N}_i}(k+1) + \overline{H}_{\mathcal{N}_i,j} A_K^k w,$$

and due to assumption 29 we have

$$H_{\mathcal{N}_i,j} z_{\mathcal{N}_i}(k+1) \leq \overline{h}_{i,k+1,j} + \epsilon_{x_i} = h_{i,j} - \sigma_{\mathcal{W}}(\overline{H}_{\mathcal{N}_i,j}^\top, k+1) - (k+1)\epsilon_{x_i},$$

which implies

$$H_{\mathcal{N}_i,j}\tilde{z}_{\mathcal{N}_i}(k) \leq h_{i,j} + \overline{H}_{\mathcal{N}_i,j} A_K^k w - \sigma_{\mathcal{W}}(\overline{H}_{\mathcal{N}_i,j}^\top, k+1) - (k+1)\epsilon_{x_i}.$$

Due to the definition of the support function and linear superposition we have

$$\sigma_{\mathcal{W}}(\overline{H}_{\mathcal{N}_i,j}^\top, k+1) \geq \sigma_{\mathcal{W}}(\overline{H}_{\mathcal{N}_i,j}^\top, k) + \overline{H}_{\mathcal{N}_i,j} A_K^k w.$$

Note that this does not hold, if we use $\mathcal{W}_1 \times \cdots \times \mathcal{W}_k$ for the k-step support function. This leads to

$$H_{\mathcal{N}_i,j}\tilde{z}_{\mathcal{N}_i}(k) \leq h_{i,j} - \sigma_{\mathcal{W}}(\overline{H}_{\mathcal{N}_i,j}^\top, k) - (k+1)\epsilon_{x_i} = \overline{h}_{i,k,j}.$$

For the input constraints the same arguments hold with

$$\begin{aligned} L_{i,j}(\tilde{v}_i(k) + K_{t,\mathcal{N}_i,j}\tilde{z}_{\mathcal{N}_i}(k)) &= L_{i,j}(v_i(k+1) + K_{t,\mathcal{N}_i}z_{\mathcal{N}_i}(k+1) + \overline{K}_{t,\mathcal{N}_i}A_K^k w) \\ &\leq \overline{d}_{i,k+1,j} + \epsilon_{u_i} + \overline{L_{i,j}K_{t,\mathcal{N}_i}}A_K^k w \leq \overline{d}_{i,k,j}. \end{aligned}$$

Part IV: Now we consider the terminal constraint and the effect of the terminal controller. We have

$$\begin{aligned} \tilde{z}(N) &= (A+BK)\tilde{z}(N-1) = (A+BK)(z(N) + A_K^{N-1}w), \\ \tilde{u}(N-1) &= K\tilde{z}(N-1). \end{aligned}$$

From assumption 29 we have $z(N) \in \overline{\mathcal{X}}_f \subseteq \overline{\mathcal{X}}_N$ and thus we can use the arguments from Part III to establish $\tilde{z}(N-1) \in \overline{\mathcal{X}}_{N-1}$. Similarly for the input $u_i(N) = Kz(N) \in K\overline{\mathcal{X}}_f \subseteq \overline{\mathcal{U}}_N$, we can establish $\tilde{u}(N-1) \in \overline{\mathcal{U}}_{N-1}$. For the terminal set we have

$$\begin{aligned} \tilde{z}(N) &= (A+BK)(z(N) + A_K^{N-1}w) \\ &\in (A+BK)(\overline{\mathcal{X}}_f \bigoplus A_K^{N-1}\mathcal{W}) \subseteq (1-\epsilon_f)\overline{\mathcal{X}}_f, \end{aligned}$$

due to (4.3). □

This result allows an additional violation ϵ in the constraints and still ensures recursive feasibility with a suitable candidate solution $\tilde{z}$, $\tilde{v}$. The initial state constraint is assumed to be satisfied, which most algorithms can accomplish. This formulation is general and independent of the used optimization algorithm, assuming that the algorithm can handle linear coupling inequality constraints. The linear decreasing relaxation $\epsilon_{z_i,k}$ over the prediction horizon, has some similarities to the adaptive exponential relaxation in [FerrantiEtAl15]. A suitable choice for $\epsilon_{z_i,k}$ might be $\epsilon_{z_i,N} = \epsilon_i$.
If we compare the required constraint tightening to ensure recursive feasibility, with the constraint tightening to ensure constraint satisfaction of the consolidated trajectory, the difference only lies in the error terms ϵ_{x_i}, ϵ_{u_i}, ϵ_i, but is independent of ϵ_{z_i}.

Remark 31. *To the best knowledge of the author, this is the only result, that uses a candidate sequence, which does not satisfy the dynamics exactly, to establish recursive feasibility. By posing the constraints for the optimization problem with an intermediate tightening ($\epsilon/2$), we can guarantee the existence of a strictly feasible candidate solution (slater vector). Thus we can use methods like [FerrantiEtAl15] to bound the dual variables and give prior bounds on the number of iterations for* *fast gradient schemes. Note that here we do not require any online adaptation of the parameters and it is even possible for some algorithms to given a general upper bound on the number of iterations, independent of the current state.*

4.2.2 Suboptimality - Stability

With this constraint tightening we can ensure feasibility of the consolidated trajectory. Now we focus on stability. Let $\mathcal{V}_\epsilon(v,x)$ denote the cost associated with the predicted consolidated trajectory $\overline{x}$ with the input trajectory v and the initial state x:

$$\mathcal{V}_\epsilon(v,x) = \sum_{i=1}^{M} \sum_{k=0}^{N-1} l_i(\overline{x}_i(k),v_i(k)) + V_{f_i}(\overline{x}_i(N)).$$

Let $\mathcal{V}^*_\epsilon(x)$ denote the optimal cost given the constraints in assumption 29:

$$\begin{aligned}
\mathcal{V}^*_\epsilon(x) &= \min_{z,v} \mathcal{V}_\epsilon(v,x) \\
\text{st. } & \overline{x}(k+1) = (A + BK_t)\overline{x}(k) + Bv(k), \quad \overline{x}(0) = x, \\
& \|(A_{\mathcal{N}_i} + B_i K_{t,\mathcal{N}_i}) z_{\mathcal{N}_i}(k) + B_i v_i(k) - z_i(k+1)\|_\infty \leq \epsilon_{z_i,k} + \epsilon_i, \\
& H_{\mathcal{N}_i} z_{\mathcal{N}_i}(k) \leq \overline{h}_{i,k} + 1_p \epsilon_{x_i}, \quad L_i(v_i(k) + K_{t,\mathcal{N}_i} z_{\mathcal{N}_i}(k)) \leq \overline{l}_{i,k} + 1_q \epsilon_{u_i}, \\
& z(N) \in \overline{\mathcal{X}}_f, \quad z_i(0) = x_i
\end{aligned}$$

By definition we have $\mathcal{V}^*_\epsilon(x) \leq \mathcal{V}_\epsilon(\tilde{v},x)$. In the following we denote the suboptimal sequence, that results from inexact minimization by (z_ϵ,v_ϵ) and assume that they satisfy assumption 29. The resulting suboptimal MPC state feedback is denoted by $\mu_\epsilon(x)$.
Using the candidate solution $(\tilde{z}_\epsilon,\tilde{v}_\epsilon)$ based on the previous solution (z_ϵ,v_ϵ), a stability result for the special case of $K = 0$,i.e. if the stabilizing controller can also be used as a terminal controller, is presented.

Lemma 32. *Let the assumptions in theorem 30 be satisfied and let the pre-stabilized system satisfy* (2.4) *with $K = 0$, i.e.*

$$V_f(A_K x) \leq V_f(x) - l(x,0).$$

Then the candidate solution from theorem 30 satisfies

$$\mathcal{V}_\epsilon(\tilde{v},A_k x + Bv) - \mathcal{V}_\epsilon(v,x) \leq -l(x,v).$$

Assume furthermore, that the suboptimal solution v_ϵ has a suboptimality $\eta(x)$, i.e.

$$\mathcal{V}_\epsilon(v_\epsilon(x),x) \leq \mathcal{V}_\epsilon^*(x) + \eta(x),$$

and that the strict dissipativity in assumption 2 is satisfied.
Then we have

$$\tilde{\mathcal{V}}_\epsilon^*(A_K x + B\mu_\epsilon(x)) - \tilde{\mathcal{V}}_\epsilon^*(x) \leq \eta(x) - \tilde{l}(x,\mu_\epsilon(x)).$$

Proof. This proof consists of 2 parts. First, the decrease in $\mathcal{V}_\epsilon$ with the candidate input $\tilde{v}$ is shown. Then assuming a suboptimality η the decrease in the candidate Lyapunov function $\tilde{\mathcal{V}}_\epsilon$ is derived.
Part I: Show cost decrease based on candidate input $\tilde{v}$.
To show the cost decrease we use the feasible input sequence $\tilde{v}$ from theorem 30, based on the previous solution v and the measured state x and thus w. With the terminal controller $K = 0$ we have

$$\tilde{v}(N-1) = 0.$$

We study the difference $\mathcal{V}_\epsilon(\tilde{v},A_K x + B\mu_\epsilon(x)) - \mathcal{V}_\epsilon(v,x)$, where $\tilde{v}$ is the candidate input sequence. Abbreviate $x^+ = A_K x + B\mu_\epsilon(x)$ to get

$$\begin{aligned}\mathcal{V}_\epsilon(\tilde{v},x^+) - \mathcal{V}_\epsilon(v,x) &= \sum_{k=0}^{N-1} l(\overline{x}(k+1),\tilde{v}(k)) - l(\overline{x}(k),v(k)) + V_f(\overline{x}(N+1)) - V_f(\overline{x}(N) \\ &= -l(x,v) + l(\overline{x}(N),\tilde{v}(N-1)) + V_f(\overline{x}(N+1)) - V_f(\overline{x}(N)).\end{aligned}$$

For the terminal state $\overline{x}(N+1)$ we have

$$\overline{x}(N+1) = A_K\overline{x}(N) + B\tilde{v}(N-1) = A_K\overline{x}(N).$$

Thus we have

$$V_f(\overline{x}(N+1)) - V_f(\overline{x}(N) + l(\overline{x}(N),\tilde{v}(N-1)) \leq 0$$

and

$$\mathcal{V}_\epsilon(\tilde{v},x^+) - \mathcal{V}_\epsilon(v,x) \leq -l(x,v).$$

Part II: Show closed-loop decrease in candidate Lyapunov function $\tilde{\mathcal{V}}_\epsilon(x)$.
First, note that $\tilde{\mathcal{V}}_\epsilon(x)$ is a valid candidate Lyapunov function. To study the

closed-loop stability based on the suboptimality η we consider $\mathcal{V}$ under the feedback μ_ϵ. Due to optimality we clearly have

$$\mathcal{V}_\epsilon^*(x^+) \leq \mathcal{V}_\epsilon(\tilde{v}_\epsilon, x^+),$$

with the candidate solution $\tilde{v}_\epsilon$ based on the suboptimal solution v_ϵ. Now using the decrease condition derived in the Part I we get

$$\mathcal{V}_\epsilon^*(x^+) \leq \mathcal{V}_\epsilon(v_\epsilon, x) - l(x, \mu_\epsilon(x)).$$

Using the suboptimality bound $\eta(x)$ we can conclude

$$\mathcal{V}_\epsilon^*(x^+) \leq \mathcal{V}_\epsilon^*(x) + \eta(x) - l(x, \mu_\epsilon(x)).$$

Similar to the proof of theorem 6 we know use the strict dissipativity assumption and add $\lambda(x^+) - \lambda(x)$ to get

$$\underbrace{\lambda(x^+) - \lambda(x) + \mathcal{V}_\epsilon^*(x^+) - \mathcal{V}_\epsilon^*(x)}_{\tilde{\mathcal{V}}_\epsilon(x^+) - \tilde{\mathcal{V}}_\epsilon(x)} \leq +\eta(x) + \underbrace{\lambda(x^+) - \lambda(x) - l(x, \mu_\epsilon(x))}_{-\tilde{l}(x, \mu_\epsilon(x))}.$$

□

The restriction $K = 0$ simplifies the procedure by using the terminal controller K as the stabilizing controller K_t, but can lead lead to a larger constraint tightening. One way to avoid this, is to impose additional constraints based on lemma 24 with some fixed ρ, γ as tuning parameters on the computation of P_f, K in lemma 10.
It is possible to derive a similar result for general terminal controller, by appending $\tilde{v}(N-1) = K\overline{x}(N)$ for the candidate trajectory. This requires the explicit computation of the consolidated trajectory $\overline{x}$ and also has higher demands on the terminal set.

Corollary 33. *If an upper bound on the suboptimality $\eta(x) \leq \overline{\eta}$ is guaranteed, we have practical asymptotic stability (see lemma 14). If the suboptimality satisfies $\eta(x) < \alpha_l(x) = \min_u \tilde{l}(x, u)$ the optimal steady state is asymptotically stable.*

Ensuring a certain degree of suboptimality can be a challenging task, especially since here the suboptimality is defined with respect to the consolidated trajectory. One way to bound the suboptimality is described in [FerrantiEtAl15], which in general requires an adaptive solver tolerance. The condition $\eta < \alpha_l(\|x\|)$ leads to high accuracy demands close to the origin. Ensuring some upper bound $\overline{\eta}$ on the suboptimality is a lot more

reasonable.
Alternatively the tightening could be chosen small enough, such that the suboptimality η is smaller than $\alpha_l(\|x\|)$ for all x outside of the terminal set $\overline{\mathcal{X}}_f$ and use dual-mode MPC, see [KögelFindeisen14].

4.3 Economic Model Predictive Control without terminal constraints

In this section we study the stability of EMPC without terminal constraints under inexact optimization. In [Grüne13] the stability of EMPC without terminal constraints is studied. Here we extend these results to include a suboptimality η due to the an inexact optimization. The final result of this section is a generalized version of theorem 3.7 in [GrüneStieler14], that includes the suboptimality η. As in chapter 2, we consider linear system dynamics with a linear-quadratic stage cost $l(x,u)$. In addition, we assume without loss of generality that the optimal steady state is the origin and that the optimal steady state cost is zero, i.e. $l(0,0) = 0$. The theoretical results are also valid for general nonlinear system dynamics and general economic stage cost $l(x,u)$, as long as the assumptions used in theorem 3.7 in [GrüneStieler14] are satisfied.
Contrary to section 4.2 we only consider a suboptimality η in the cost and do not prove recursive feasibility of the tightened optimization problem under inexact minimization. For EMPC without terminal constraints this is a significantly more difficult and so far unexplored task.

Stability Proof with inexact minimization

If we consider the stability results in section 4.2, we always rely on the candidate solution $\tilde{v}$. For the terminal set DEMPC this candidate solution can always be obtained due to the properties of the terminal set and feasibility. For the EMPC without terminal constraints we also use a candidate solution to show the stability properties. The (main) difference is that this candidate solution uses properties that are implicitly satisfied due to strict dissipativity and optimality, but not enforced by any constraint. This requires a more detailed investigation of the stability proof in [Grüne13] to derive similar stability properties.

Suboptimal EMPC

We first define some terms. We assume that the MPC optimization problem is solved by some algorithm, that can ensure feasibility, but will not give the optimal input, see chapter 3. The suboptimal input can be written as

$$u_\epsilon(x) = \{u_\epsilon(0,x), u_\epsilon(1,x), \ldots, u_\epsilon(N-1,x)\}$$

and the corresponding state feedback of the suboptimal EMPC is defined by $\mu_{N,\epsilon}(x) = u_\epsilon(0,x)$. Furthermore the closed-loop state trajectory under the suboptimal EMPC feedback can is written $x_{\mu_{N,\epsilon}}(k,x)$. The open-loop predicted cost over the prediction horizon N of a input u is defined as $J_N(x,u)$. The resulting suboptimal MPC feedback is defined as follows.

Definition 34. *The η-suboptimal MPC solution $u_\epsilon(x)$ satisfies the original state and input constraints, i.e. $u_\epsilon(x) \in \mathcal{U}^N(x)$. Furthermore, the corresponding value function $\mathcal{V}_{N,\epsilon}(x)$ satisfies*

$$\mathcal{V}_{N,\epsilon}(x) = J_N(x, u_\epsilon(x)) \leq \mathcal{V}_N^*(x) + \eta.$$

Overview

The proof of the practical asymptotic stability of inexact EMPC without terminal constraints is rather long and consists of several intermediate results. Large portions of the proof are taken from [Grüne13]. The main results, that include the suboptimality, are lemma 35 and theorem 36.
The proof is structured as follows. In lemma 35, bounds on the closed-loop cost with suboptimal EMPC are derived, given the existence of a candidate control sequence $u_{N,x}$. In theorem 36, this candidate control sequence is constructed based on some assumptions on the optimal cost function $\mathcal{V}_N^*$ and a turnpike property. In theorem 39, sufficient assumptions to satisfy condition c) in theorem 36 are stated. In theorem 42 sufficient assumptions to satisfy condition b) in theorem 36 are stated. In theorem 47 the bounds of theorem 36 on $\mathcal{V}_N^*$ are used to derive bounds on the rotated cost $\tilde{\mathcal{V}}_N^*$. In theorem 48, all the previous results are combined to show practical asymptotic stability.

1. Performance with candidate input $u_{N,x}$

The first part of the proof consists in showing some properties of the optimal value function $\mathcal{V}_N^*$ under the suboptimal MPC feedback $\mu_{N,\epsilon}$, based on the existence of a candidate control sequence $u_{N,x}$. This lemma is an extension of proposition 4.1 in [Grüne13] to EMPC with suboptimality. The corresponding control sequence will be constructed in theorem 36. This lemma also gives us an upper bound on the average infinite horizon closed-loop performance $\bar{J}^{\text{cl}}_{\infty}$

$$\bar{J}^{\text{cl}}_{\infty}(x,\mu_{N,\epsilon}) = \lim_{K\to\infty} \frac{1}{K} J^{\text{cl}}_{K}(x,\mu_{N,\epsilon}),$$
$$J^{\text{cl}}_{K}(x,\mu_{N,\epsilon}) = \sum_{k=0}^{K} l(x_{\mu_{N,\epsilon}}(k,x),\mu_{N,\epsilon}(x_{\mu_{N,\epsilon}}(k,x))).$$

Lemma 35. *Assume there exists a $N_0 > 0$, $\delta_1, \delta_2 \in \mathcal{L}$ such that $\forall x \in \mathcal{X}$, $N \geq N_0$, there exists a control sequence $u_{N,x} \in \mathcal{U}^{N+1}(x)$ and $k_{N,x} \in \{0,\ldots,N\}$ such that the following conditions hold:*

$$(1)\;\; J'_N(x) := \sum_{k=0,k\neq k_{N,x}}^{N} l(x_{u_{N,x}}(k,x),u_{N,x}(k)) \leq \mathcal{V}^*_N(x) + \delta_1(N)$$

$$(2)\;\; l(x_{u_{N,x}}(k_{N,x},x),u_{N,x}(k_{N,x})) \leq \delta_2(N)$$

Then the suboptimal EMPC $\mu_{N,\epsilon}$ with suboptimality η satisfies the following inequalities
for all $x \in \mathcal{X}$, all $N \in \mathbb{N}$, with $N \geq N_0 + 1$ and for all $K \in \mathbb{N}$:

$$J^{cl}_{K}(x,\mu_{N,\epsilon}) \leq \mathcal{V}^*_N(x) - \mathcal{V}^*_N(x_{\mu_{N,\epsilon}}(K)) + K(\delta_1(N-1) + \delta_2(N-1) + \eta),$$
$$\bar{J}^{cl}_{\infty}(x,\mu_{N,\epsilon}) \leq \delta_1(N-1) + \delta_2(N-1) + \eta.$$

Proof. We start with the dynamic programming principle

$$\begin{aligned}\mathcal{V}^*_N(x) &= l(x,\mu_N(x)) + \mathcal{V}^*_{N-1}(Ax + B\mu_N(x)) \leq l(x,\mu_\epsilon(x)) + \mathcal{V}^*_{N-1}(Ax + B\mu_{N,\epsilon}(x))\\ &\leq l(x,\mu_\epsilon(x)) + J_{N-1}(Ax + B\mu_{N,\epsilon}(x),u_\epsilon) = \mathcal{V}_{N,\epsilon}(x) \leq \mathcal{V}^*_N(x) + \eta,\end{aligned}$$

which implies

$$l(x,\mu_{N,\epsilon}(x)) \leq \mathcal{V}^*_N(x) - \mathcal{V}^*_{N-1}(Ax + B\mu_{N,\epsilon}(x)) + \eta.$$

In the following we denote $x_{N,\epsilon}(k) = x_{\mu_{N,\epsilon}}(k,x)$ and $\mu_{N,\epsilon}(x_{N,\epsilon}(k)) = \mu_{N,\epsilon}(k)$. From this we have

$$\begin{aligned} J_K^{\text{cl}}(x,\mu_{N,\epsilon}) &= \sum_{k=0}^{K-1} l(x_{N,\epsilon}(k),\mu_{N,\epsilon}(k))) \qquad (4.4) \\ &\leq \sum_{k=0}^{K-1} \left(\mathcal{V}_N^*(x_{N,\epsilon}(k)) - \mathcal{V}_{N-1}^*(x_{N,\epsilon}(k+1)) + \eta\right) \\ &= \mathcal{V}_N^*(x) - \mathcal{V}_{N-1}^*(x_{N,\epsilon}(K)) + K\eta + \sum_{k=1}^{K-1} \mathcal{V}_N^*(x_{N,\epsilon}(k)) - \mathcal{V}_{N-1}^*(x_{N,\epsilon}(k)). \end{aligned}$$

Now we will bound these terms. Using property (1) with $N-1$ in place of N we get

$$\mathcal{V}_{N-1}^*(x) \geq J'_{N-1}(x) - \delta_1(N-1)$$

and due to optimality we have

$$\mathcal{V}_N^*(x) \leq J_N(x,u_{N-1,x}).$$

By combining these bounds we get

$$\begin{aligned} \mathcal{V}_N^*(x) - \mathcal{V}_{N-1}^*(x) &\leq J_N(x,u_{N-1,x}) - J'_{N-1}(x) + \delta_1(N-1) \qquad (4.5) \\ &= l(x_{u_{N-1,x}}(k_{N-1,x},x),u_{N-1,x}(k_{N-1,x})) + \delta_1(N-1) \\ &\leq \delta_2(N-1) + \delta_1(N-1). \end{aligned}$$

By pluging in (4.5) in (4.4) with $x = x_{N,\epsilon}(k)$ we get

$$J_K^{\text{cl}}(x,\mu_{N,\epsilon}) \leq \mathcal{V}_N^*(x) - \mathcal{V}_{N-1}^*(x_{N,\epsilon}(K)) + (K-1)(\delta_2(N-1) + \delta_1(N-1)) + K\eta.$$

Using (4.5) with $k = K$ we have

$$\mathcal{V}_N^*(x_{N,\epsilon}(K)) - \mathcal{V}_{N-1}^*(x_{N,\epsilon}(K)) \leq \delta_2(N-1) + \delta_1(N-1),$$

which yields

$$J_K^{\text{cl}}(x,\mu_{N,\epsilon}) \leq \mathcal{V}_N^*(x) - \mathcal{V}_N^*(x_{N,\epsilon}(K)) + K(\delta_2(N-1) + \delta_1(N-1) + \eta).$$

The average bound $\overline{J}_\infty$ follows by dividing by K and letting $K \to \infty$, since $\mathcal{V}_N^*$ is finite. □

This lemma states, that we can bound the average infinite horizon performance $\overline{J}_\infty$, if there is a control sequence $u_{N,x}$, that is close to optimal (δ_1) and reaches a state with a cost close to the optimal steady state cost (δ_2) within $k_{N,x}$ steps. If the suboptimality η goes to zero, we recover the original result from [Grüne13].

2. Construct Candidate Input

The following theorem is the main theoretical results for EMPC without terminal constraints. Here a candidate control sequence is constructed, based on the previous optimal input u^*.
This is a modified version of theorem 4.2 in [Grüne13], that uses lemma 35 and the suboptimal MPC $\mu_{N,\epsilon}$ instead of proposition 4.1 in [Grüne13].

Theorem 36. *Assume there exists $\overline{\delta} > 0$ such that the following properties holds:*

a) There exists $\gamma_f, \gamma_l \in \mathcal{K}_\infty$ such that $\forall \delta \in (0,\overline{\delta}]$, $\forall x \in \mathcal{B}_\delta$, there is a $u_x \in \mathcal{U}$ such that $Ax + Bu_x \in \mathcal{X}$ and

$$\|Ax + Bu_x\| \leq \gamma_f(\delta) \quad \text{and} \quad l(x,u_x) \leq \gamma_l(\delta)$$

hold.

b) There exists a $N_0 \in \mathbb{N}_0$ and $\gamma_V \in \mathcal{K}_\infty$ such that for each $\delta \in (0,\overline{\delta}]$, for each $N \in \mathbb{N}$ with $N \geq N_0$ and $x \in \mathcal{B}_\delta$ the inequality

$$|\mathcal{V}_N^*(x) - \mathcal{V}_N^*(0)| \leq \gamma_V(\delta)$$

holds.

c) There exists a $\sigma \in \mathcal{L}$ and $N_1 \in \mathbb{N}$ with $N_1 \geq N_0$ for $N_0 \in \mathbb{N}_0$ from b), such that for each $x \in \mathcal{X}$ and each $N \geq N_1$ there exists an optimal trajectory $x_{u^*_{N,x}}(\cdot,x)$ satisfying $\|x_{u^*_{N,x}}(k_x,x)\| \leq \sigma(N)$ for some $k_x \in \{0,\ldots,N-N_0\}$.

Then there exists a $N_2 \in \mathbb{N}$ such that the suboptimal EMPC $\mu_{N,\epsilon}$ with suboptimality η satsifies the following inequalities

$$J_K^{cl}(x,\mu_{N,\epsilon}) \leq \mathcal{V}_N^*(x) - \mathcal{V}_N^*(x_{\mu_{N,\epsilon}}(K)) + K(\epsilon(N-1)+\eta),$$
$$\overline{J}_\infty^{cl}(x,\mu_{N,\epsilon}) \leq \epsilon(N-1)+\eta,$$

for all $x \in \mathcal{X}, K \in \mathbb{N}$, all $N \geq N_2 + 1$ with $\epsilon \in \mathcal{L}$ given by

$$\epsilon(N) = \gamma_V(\sigma(N)) + \gamma_V(\gamma_f(\sigma(N))) + \gamma_l(\sigma(N)).$$

If furthermore

$$\mathcal{V}_N^*(x) \geq \mathcal{V}_N^*(0)$$

holds for all $N \in \mathbb{N}$ and $x \in \mathcal{B}_{\overline{\delta}}$, then we have

$$\epsilon(N) = \gamma_V(\gamma_f(\sigma(N))) + \gamma_l(\sigma(N)).$$

Proof. This proof basically consist of constructing the control sequence for lemma 35 with

$$\delta_1(N) = \gamma_V(\sigma(N)) + \gamma_V(\gamma_f(\sigma(N)), \quad \delta_2(N) = \gamma_l(\sigma(N)).$$

To satisfy the assumptions of lemma 35 we choose $N_2 \geq N_1$ such that $\sigma(N_2) \leq \overline{\delta}$ and $\gamma_f(\sigma(N_2)) \leq \overline{\delta}$ holds for σ from c) and γ_f from a). Now pick $N \geq N_2$, $x \in \mathcal{X}$ and $u^*_{N,x} \in \mathcal{U}^N(x)$ from c) with $x' = x_{u^*_{N,x}}(k_x,x)$ and let $u_{x'}$ be the control value in a) for $x = x'$. Let $x'' = Ax' + Bu_{x'}$ and let $u^*_{N-k_x,x''}$ be an optimal control sequence for $x = x''$ and the horizon $N - k_x$. With this the candidate control sequence $u_{N,x} \in \mathcal{U}^{N+1}(x)$ is given by

$$u_{N,x}(k) := \begin{cases} u^*_{N,x}(k) & k = 0,\ldots,k_x - 1 \\ u_{x'} & k = k_x \\ u^*_{N-k_x,x''}(k - k_x - 1) & k = k_x + 1,\ldots,N. \end{cases}$$

This implies

$$\begin{aligned} x_{u_{N,x}}(k,x) = x_{u^*_{N,x}}(k,x), \qquad k = 0,\ldots,k_x, \\ \|x'\| \leq \sigma(N), \quad \|x''\| = \|Ax' + Bu_{x'}\| \leq \gamma_f(\sigma(N)), \\ l(x',u_{x'}) \leq \gamma_l(\sigma(N)). \end{aligned}$$

From this we can conclude using b)

$$\begin{aligned} \mathcal{V}^*_K(x'') &\leq \mathcal{V}^*_K(0) + \gamma_V(\gamma_f(\sigma(N))) \\ &\leq \gamma_V(\gamma_f(\sigma(N))) + \gamma_V(\sigma(N)) + \mathcal{V}^*_K(x') = \mathcal{V}^*_K(x') + \delta_1(N) \end{aligned} \tag{4.6}$$

for any $K \in \mathbb{N}$ with $K \geq N_0$. By c) we have $K = N - k_x \geq N_0$. Now we have to distinguish two cases:
In case $N - k_x \geq 1$ we use (4.6) to get

$$\begin{aligned} \sum_{k=k_x+1}^{N} l(x_{u_{N,x}}(k,x),u_{N,x}(k)) = J_{N-k_x}(x'',u^*_{N-k_x,x''}) = \mathcal{V}^*_{N-k_x}(x'') \\ \leq \mathcal{V}^*_{N-k_x}(x') + \delta_1(N). \end{aligned}$$

In order to satisfy condition (1) in lemma 35 we set $k_{N,x} = k_x$ and get

$$\begin{aligned} J'_N(x) &= \sum_{k=0,k\neq k_x}^{N} l(x_{u_{N,x}}(k,x),u_{N,x}(k)) \\ &= \sum_{k=0}^{k_x-1} l(x_{u_{N,x}}(k,x),u_{N,x}(k)) + \sum_{k=k_x+1}^{N} l(x_{u_{N,x}}(k,x),u_{N,x}(k)) \\ &= \mathcal{V}_N^*(x) - \mathcal{V}_{N-k_x}^*(x') + \sum_{k=k_x+1}^{N} l(x_{u_{N,x}}(k,u_{N,x}(k))) \\ &\leq \mathcal{V}_N^*(x) + \delta_1(N), \end{aligned}$$

which satisfies condition (1) of lemma 35 with $\delta_1(N) = \gamma_V(\sigma(N)) + \gamma_V(\gamma_l(\sigma(N)))$.
In case $N - k_x = 0$ we get

$$J'_N(x) = \sum_{k=0,k\neq k_x}^{N} l(x_{u_{N,x}}(k,x),u_{N,x}(k)) = \mathcal{V}_N^*(x),$$

which also satisfies condition (1) of lemma 35 with $\delta_1(N)$.
For condition (2) of lemma 35 we get

$$l(x_{u_{N,x}}(k_x,k),u_{N,x}(k_x)) = l(x',u_{x'}) \leq \gamma_l(\sigma(N)) = \delta_2(N).$$

With this we have satisfied all conditions of lemma 35 and get

$$\epsilon(N) = \delta_1(N) + \delta_2(N) + \eta = \gamma_l(\sigma(N)) + \gamma_V(\sigma(N)) + \gamma_V(\gamma_l(\sigma(N))) + \eta.$$

For the special case $\mathcal{V}_N^*(x) \geq \mathcal{V}_N^*(0)$ we get

$$\mathcal{V}_K^*(x'') \leq \mathcal{V}_K^*(0) + \gamma_V(\gamma_l(\sigma(N))) \leq \mathcal{V}_K^*(x') + \gamma_V(\gamma_l(\sigma(N))),$$

which implies $\delta_1(N) = \gamma_V(\gamma_f(\sigma(N)))$ and thus the lower bound $\epsilon(N)$. □

This theorem contains the desired inequality for the cost $\mathcal{V}_N^*$ under suboptimal MPC $\mu_{N,\epsilon}$, but the conditions b) and c) are rather hard to verify. The following theorems will give simpler sufficient assumptions for condition b) and c), and use strict dissipativity to conclude practical asymptotic stability. The tighter bound for $\mathcal{V}_N^*(x) \geq \mathcal{V}^*(0)$ is in general only achievable for a positive (semi-)definite tracking cost, and not for 'real' economic cost functions.

3. Asymptotic Controllability

We first discuss sufficient assumptions, that can be used to satisfy condition c) in theorem 36. Therefore we define $\mathcal{KLS}$ functions:

Definition 37. *([Grüne13], Def. 5.4)*
A class $\mathcal{KLS}$ is a class of summable $\mathcal{KL}$ functions, that sum up to a $\mathcal{K}$ function, i.e. $\beta \in \mathcal{KL}$ with

$$\gamma_\beta(r) := \sum_{k=0}^{\infty} \beta(r,k)$$

satisfies $\gamma_\beta(r) \in \mathcal{K}$.

The following asymptotic controllability condition will be used.

Assumption 38. *([Grüne13], Assumpt. 5.5)*
There exists a $\beta \in \mathcal{KLS}$ such that $\forall x \in \mathcal{X}$, $\forall N \in \mathbb{N}$ there exists a $u \in \mathcal{U}^N(x)$ such that

$$l(x_u(k,x),u(k)) \leq \beta(\|x\|,k)$$

holds for all $k = 0, \ldots, N-1$.

Theorem 39. *([Grüne13], Thm. 5.6)*
Assume that the strict dissipativity conditions holds and $C := 2\sup_{x\in\mathcal{X}} \|\lambda(x)\|$. Assume furthermore, that the asymptotic controllability assumption 38 is satisfied. Then condition c) of theorem 36 is satisfied.

4. Local Controllability

Now corresponding assumptions for condition b) in theorem 36 are discussed. The following intermediate result gives convergence results to the optimal state for close to optimal inputs.

Theorem 40. *([Grüne13],Thm. 5.3)*
Assume that there exists a λ that satisfies the strict dissipativity condition in assumption 15 and $C := 2\sup_{x\in\mathcal{X}} \|\lambda(x)\|$. Then for each $x \in \mathcal{X}$, $\delta > 0$, each control sequence $u \in \mathcal{U}^N(x)$ satisfying $J(x,u) \leq \delta$ and each $\epsilon > 0$ the value

$$Q_\epsilon := \#\{k \in \{0,\ldots,N-1\} \mid \|x_u(k,x)\| \leq \epsilon\}$$

satisfies the inequality $Q_\epsilon \geq N - (\delta + C)/\alpha_l(\epsilon)$.

Now a local controllability condition on $\mathcal{B}_\delta$ sufficient for condition b) in theorem 36 is used.

Assumption 41. *([Grüne13],Assumption 6.1)*
There exists a $\delta_c > 0$, $d \in \mathbb{N}$, γ_x, γ_u, $\gamma_c \in \mathcal{K}_\infty$ such that for each $x_{u_1}(k,x)$ with $u_1 \in \mathcal{U}^d(x)$ satisfying $x_{u_1}(k,x) \in \mathcal{B}_{\delta_c}$ for all $k = 0,\dots,d$ and all x_1, $x_2 \in \mathcal{B}_{\delta_c}$ there exists a $u_2 \in \mathcal{U}^d(x)$ satisfying

$$x_{u_2}(d,x_1) = x_2$$

and the estimates

$$\begin{aligned} \|x_{u_2}(k,x_1) - x_{u_1}(k,x)\| &\leq \gamma_x(\max\{\|x_1 - x\|,\|x_2 - x_{u_1}(d,x)\|\}), \\ \|u_2(k) - u_1(k)\| &\leq \gamma_u(\max\{\|x_1 - x\|,\|x_2 - x_{u_1}(d,x)\|\}), \end{aligned}$$

and

$$\|l(x_{u_2}(k,x_1),u_2(k)) - l(x_{u_1}(k,x),u_1(k))\| \leq \gamma_c(\max\{\|x_1 - x\|,\|x_2 - x_{u_1}(d,x)\|\})$$

for all $k = 0,\dots,d-1$.

With this assumption we can ensure that the open-loop state trajectory temporary stays close to the optimal steady state. In addition to this assumption, we require a bound on the rotated stage cost

$$\tilde{l}(x,u) \leq \alpha_u(\|x\| + \|u\|). \tag{4.7}$$

This result can be used for the asymptotic controllability condition, assumption 38 in theorem 39, with $P(N) = \lceil N/2 \rceil$. Using the fact that for some $k \leq N/2$ we have $\|x_{u^*_{N,x}}(k,x)\| \leq \sigma(N)$ we can give sufficient conditions for condition b) in theorem 36.

Theorem 42. *([Grüne13],Thm. 6.4)*
Let the assumptions in theorem 40, assumption 41 and (4.7) hold. Then condition b) in theorem 36 is satisfied.

5. Rotated Cost decrease

Now the decrease in the cost $\mathcal{V}_N^*$ of theorem 36 is used to get a decrease in the rotated cost $\tilde{\mathcal{V}}_N^*$. We first need some intermediate results:

Assumption 43. *([Grüne13], Assumption 7.2)*
We assume that there exists $C' \geq 0$ such that for each $x \in \mathcal{X}$, the optimal control sequence $u^*_{x,N} \in \mathcal{U}^N(x)$ and each $\epsilon > 0$ the value

$$Q_\epsilon := \#P_\epsilon, \quad P_\epsilon := \{k \in \{0, \dots, N-1\} | \|x_{u^*_{N,x}}(k,x)\| \leq \epsilon\}$$

satisfies $Q_\epsilon \geq N - C'/\alpha_l(\epsilon)$.

Lemma 44. *([Grüne13], Lemma 7.4)*
Assume that the MPC problem satisfies assumption 43 and condition (b) from theorem 36 for some $N_0 \in \mathbb{N}$ and $\overline{\delta} > 0$. Then for all $x \in \mathcal{X}$, $N \geq N_0$ and $\epsilon \leq \overline{\delta}$ satisfying

$$\epsilon \geq \alpha_l^{-1}\left(\frac{C'}{N - N_0 - 1}\right),$$

the set $P'_\epsilon = P_\epsilon \cap \{0, \dots, N - N_0\}$ is nonempty and for all $P \in P'_\epsilon$ we have

$$\mathcal{V}^*_N(x) = J_P(x, u^*_{N,x}) + \mathcal{V}^*_{N-P}(0) + R_1(N, \epsilon)$$

with

$$|R_1(N, \epsilon)| \leq \gamma_V(\epsilon).$$

Lemma 45. *([Grüne13], Lemma 7.5)*
Assume $\lambda(x)$ is Lipschitz on $B_{\overline{\delta}}$ with Lipschitz constant L_λ. Then for all $u \in \mathcal{U}^P_{\mathcal{B}_\epsilon}(x)$ with $\epsilon < \overline{\delta}$ we have

$$\begin{aligned} \tilde{J}_P(x, u) = J_P(x, u) + \lambda(x) + R_2(u, P, \epsilon), \\ |R_2(u, P, \epsilon)| \leq L_\lambda \epsilon. \end{aligned}$$

Lemma 46. *([Grüne13], Lemma 8.5)*
Suppose that the assumptions of lemma 44 and lemma 45 hold for some $\overline{\delta} > 0$, for both the original and the rotated problem, and that

$$\epsilon \geq \alpha_l^{-1}\left(\frac{2C'}{N - N_0 - 1}\right).$$

Then $P'_\epsilon := P_\epsilon \cap \tilde{P}_\epsilon \cap \{0, \dots, N - N_0\} \neq \emptyset$ and for each $P \in P'_\epsilon$ we have

$$\begin{aligned} \tilde{J}_P(x, \tilde{u}^*_{N,x}) = J_P(x, u^*_{N,x}) + \lambda(x) + R_3(P, \epsilon), \\ |R_3(P, \epsilon)| \leq 4(\gamma_V(\epsilon) + L_\lambda \epsilon). \end{aligned}$$

With these intermediate results we can uses the result in theorem 36 for the suboptimal MPC $\mu_{N,\epsilon}$ and gives bounds on the rotated cost $\tilde{\mathcal{V}}_N^*$. The proof uses large parts of the proof of theorem 7.6 [Grüne13].

Theorem 47. *Assume that the original problem and the rotated problem satisfy assumption 43 and condition (b) of theorem 36 for some* $N_0 \in \mathbb{N}_0$ *and* $\overline{\delta} > 0$. *Assume furthermore, that* λ *is Lipschitz on* $\mathcal{B}_{\overline{\delta}}$ *with Lipschitz constant* L_λ . *Let* N_1 *be such that*

$$\epsilon = \overline{\delta} \geq \alpha_l^{-1}\left(\frac{6C'}{N - N_0 - 1}\right)$$

for all $N \geq N_1$.
Assume that

$$J_K^{cl}(x,\mu_{N,\epsilon}) \leq \mathcal{V}_N^*(x) - \mathcal{V}_N^*(x_{\mu_{N,\epsilon}}(K)) + \delta.$$

Then for all $N \geq N_1$ *we have*

$$\tilde{J}_K^{cl}(x,\mu_{N,\epsilon}) \leq \tilde{\mathcal{V}}_N^*(x) - \tilde{\mathcal{V}}_N^*(x_{\mu_{N,\epsilon}}(K)) + \tilde{\delta},$$

with $\tilde{\delta} = \delta + 2\gamma_V(\epsilon) + 8(\gamma_V(\epsilon) + L_\lambda \epsilon)$.

Proof. To derive this inequality lemma 44 is used for times and lemma 46 twice. Therefore assumption 43 is used eight times. Thus we can choose N such that we have

$$\overline{\delta} \geq \alpha_l^{-1}\left(\frac{8C'}{N - N_0 - 1}\right).$$

This ensures, that P'_ϵ is nonempty, for all six considered trajectories. We start by applying lemma 44 at x and $x_{\mu_\epsilon}(K)$ to get

$$\begin{aligned}
&\mathcal{V}_N^*(x) - \mathcal{V}_N^*(x_{\mu_{N,\epsilon}}(K)) \\
=&J_P(x,u_{N,x}^*) + \mathcal{V}_N^*(0) + R_1^1(N,\epsilon) \\
&- J_P(x_{\mu_{N,\epsilon}}(K),u^*_{N,x_{\mu_{N,\epsilon}}(K)}) - \mathcal{V}_N^*(0) - R_1^2(N,\epsilon) \\
=&J_P(x,u_{N,x}^*) - J_P(x_{\mu_{N,\epsilon}}(K),u^*_{N,x_{\mu_{N,\epsilon}}(K)}) + R_1^1(N,\epsilon) - R_1^2(N,\epsilon),
\end{aligned}$$

with $|R_1^i(N,\epsilon)| \leq \gamma_V(\epsilon)$. Similarly for the rotated problem we get

$$\begin{aligned}
&\tilde{\mathcal{V}}_N^*(x) - \tilde{\mathcal{V}}_N^*(x_{\mu_{N,\epsilon}}(K)) \\
=&\tilde{J}_P(x,\tilde{u}_{N,x}^*) - \tilde{J}_P(x_{\mu_{N,\epsilon}}(K),\tilde{u}^*_{N,x_{\mu_{N,\epsilon}}(K)}) + \tilde{R}_1^1(N,\epsilon) - \tilde{R}_1^2(N,\epsilon).
\end{aligned}$$

By applying lemma 46 to the points x and $x_{\mu_\epsilon}(K)$ we get

$$\begin{aligned}&J_P(x,u^*_{N,x}) - J_P(x_{\mu_{N,\epsilon}}(K),u^*_{N,x_{\mu_{N,\epsilon}}(K)})\\ =&\tilde{J}_P(x,\tilde{u}^*_{N,x}) - \tilde{J}_P(x_{\mu_{N,\epsilon}}(K),\tilde{u}^*_{N,x_{\mu_{N,\epsilon}}(K)})\\ &-\lambda(x) + \lambda(x_{\mu_{N,\epsilon}}(K)) + R^2_3(P,\epsilon) - R^1_3(P,\epsilon),\end{aligned}$$

with $|R^i_3(P,\epsilon)| \le 4(\gamma_V(\epsilon) + L_\lambda \epsilon)$. With this we get

$$\begin{aligned}&\tilde{J}^{\mathrm{cl}}_K(x,\mu_{N,\epsilon}) = J^{\mathrm{cl}}_K(x,\mu_{N,\epsilon}) + \lambda(x) - \lambda(x_{\mu_{N,\epsilon}}(K))\\ \le&\mathcal{V}^*_N(x) - \mathcal{V}^*_N(x_{\mu_{N,\epsilon}}(K)) + \delta + \lambda(x) - \lambda(x_{\mu_{N,\epsilon}}(K))\\ =&J_P(x,u^*_{N,x}) - J_P(x_{\mu_{N,\epsilon}}(K),u^*_{N,x_{\mu_{N,\epsilon}}(K)})\\ &+R^1_1(N,\epsilon) - R^2_1(N,\epsilon) + \delta(N) + \lambda(x) - \lambda(x_{\mu_{N,\epsilon}}(K))\\ =&\tilde{J}_P(x,\tilde{u}^*_{N,x}) - \tilde{J}_P(x_{\mu_\epsilon}(K),\tilde{u}^*_{N,x_{\mu_{N,\epsilon}}(K)})\\ &+R^1_1(N,\epsilon) - R^2_1(N,\epsilon) - R^2_3(P,\epsilon) + R^1_3(P,\epsilon) + \delta + \lambda(x) - \lambda(x_{\mu_{N,\epsilon}}(K))\\ =&\tilde{\mathcal{V}}^*_N(x) - \tilde{\mathcal{V}}^*_N(x_{\mu_{N,\epsilon}}(K)) + \underbrace{R^1_1(N,\epsilon) - R^2_1(N,\epsilon) - R^2_3(P,\epsilon) + R^1_3(P,\epsilon) + \delta}_{\tilde{\delta}}.\end{aligned}$$

□

6. Practical Asymptotic Stability

Now we combine all of the previous results to derive practical asymptotic stability under the suboptimal MPC $\mu_{N,\epsilon}$. This theorem is a modified version of theorem 3.7 in [GrüneStieler14], that includes suboptimality.

Theorem 48. *Let assumptions 15-18 be satisfied. Then there exists a $N_0 \in \mathbb{N}$, $\alpha_V \in \mathcal{K}_\infty$ such that*

$$\begin{aligned}\tilde{\mathcal{V}}^*_N(Ax + B\mu_{N,\epsilon}(x)) \le \tilde{\mathcal{V}}^*_N(x) - \alpha_l(|x|) + \tilde{\delta}(N) + \eta,\\ \alpha_l(|x|) \le \tilde{\mathcal{V}}^*_N(x) \le \alpha_V(|x|),\end{aligned}$$

for all $N \ge N_0$ with $\tilde{\delta} \in \mathcal{L}$. Correspondingly we have practical asymptotic stability with $\|x_{\mu_{N,\epsilon}}(k,x)\| \le \max\{\beta_{N,\epsilon}(k,x),\eta_\epsilon\}$ and $\beta_{N,\epsilon} \in \mathcal{KL}$.
Furthermore we have

$$l(x,\mu_{N,\epsilon}) \le \mathcal{V}^*_N(x) - \mathcal{V}^*_N(Ax + B\mu_{N,\epsilon}(x)) + \delta(N) + \eta,$$

with $\delta \in \mathcal{L}$.

Proof. This proof consists of four steps. First the bounds on the rotated cost $\tilde{\mathcal{V}}_N^*$ are derived. Then the bound on the stage cost $l(x,\mu_{N,\epsilon})$ is derived using theorem 36 with $K = 1$. This can be used in combination with theorem 47 with $K = 1$ to show the desired decrease in the rotated cost $\tilde{\mathcal{V}}_N^*$. Finally this decrease is used to show practical asymptotic stability.
Part I:
The bounds α_l, α_V on $\tilde{\mathcal{V}}_N^*$ are independent of the suboptimality and can thus be used directly from theorem 19.
Part II:
We need to show

$$l(x,\mu_{N,\epsilon}) \leq \mathcal{V}_N^*(x) - \mathcal{V}_N^*(x_{\mu_{N,\epsilon}}(1)) + \delta + \eta.$$

This can be done by satisfying the conditions a)-c) of theorem 36 with $K = 1$ and

$$\delta = \epsilon(N-1), \quad \epsilon(N) = \gamma_V(\sigma(N)) + \gamma_V(\gamma_f(\sigma(N))) + \gamma_l(\sigma(N)).$$

Condition a) follows directly from assumption 16.
For condition (b) of theorem 36 we know that the assumptions in theorem 42 are sufficient. The bound on the rotated stage cost (4.7) is clearly satisfied for the linear-quadratic stage cost. Assumption 41 is a local controllability assumption, that can be replaced by assumption 17 in combination with a Lipschitz bound on λ and the upper bound on the rotated stage cost in assumption 16. The Lipschitz continuity of λ is contained in assumption 16. The last assumption follows from the strict dissipativity in assumption 15 and the bounded set.
For condition c) we can use theorem 39, which requires the controllability assumption 38. Due to the upper bound on $\tilde{J}_N(x,u)$ we can conclude an upper bound on $J_N(x,u)$ by using the continuity of λ. With this bound we can establish the assumption.
Part III:
The bounds on the cost decrease in the rotated cost $\tilde{\mathcal{V}}_N^*$ can be concluded from the cost decrease in $\mathcal{V}_N^*$ by using theorem 47 with $K = 1$. In order to use theorem 47 we have to ensure, that assumption 43 and condition (b) of theorem 36 are satisfied. From Part II, we already know that condition (b) of theorem 36 is satisfied.
To show that assumption 43 is satisfied, we can use theorem 40. Due to the bound on $\tilde{\mathcal{V}}_N^*$ on the compact set we have

$$J_N(x,u_{N,x}^*) \leq \mathcal{V}_N^*(x) = \tilde{\mathcal{V}}_N^*(x) - \lambda(x) + \lambda(x_{u_{N,x}^*}) \leq C_1$$

and thus we can use theorem 40 with $\delta = C_1$ to get $C = C_1 + 2C$. Correspondingly for the rotated cost we have $\tilde{C}' = \max_{x \in \mathcal{X}} \alpha_V(\|x\|)$.
By choosing ϵ from theorem 47 as $\epsilon = \gamma_\epsilon(N)$, with $\gamma_\epsilon \in \mathcal{L}$, we get

$$\begin{aligned}\tilde{\delta}(N) =& \delta(N) + 2\gamma_V(\gamma_\epsilon(N)) + 8(\gamma_V(\gamma_\epsilon(N)) + L_\lambda \gamma_\epsilon(N)) \\ =& \gamma_V(\sigma(N)) + \gamma_V(\gamma_f(\sigma(N))) + \gamma_l(\sigma(N)) + 2\gamma_V(\gamma_\epsilon(N)) \\ &+ 8(\gamma_V(\gamma_\epsilon(N)) + L_\lambda \gamma_\epsilon(N))\end{aligned}$$

with $\tilde{\delta}(N) \in \mathcal{L}$.
Part IV:
By using lemma 14 we get practical asymptotic stability with

$$\epsilon_{\overline{\eta}} = \alpha_l^{-1}(\alpha_V(\alpha_3^{-1}(\tilde{\delta}(N) + \eta)) + \tilde{\delta}(N) + \eta).$$

□

Summary

Let us summarize this stability result and compare it to the nominal stability result obtained in [GrüneStieler14]. For a constant η and a large enough prediction horizon N we have practical asymptotic stability. The error η_ϵ converges to zero, if the prediction horizon N goes to infinity and the suboptimality η goes to zero.
If the suboptimality η converges to zero, we recover the original result for nominal MPC. If the prediction horizon N goes to infinity, $\tilde{\delta}$ goes to zero, but we still have only practical asymptotic stability, due to the constant suboptimality η.
If we compare this result, with the stability result in section 4.2 for terminal constraint DEMPC we have the additive term η in both cases. A big difference between the two results, is that for the terminal constraint DEMPC, the constraint tightening and the recursive feasibility of inconsistent trajectories is also taken into account.
In theory for a constant suboptimality η, the stability guarantees improve for longer prediction horizons N. On the other hand achieving a certain suboptimality η requires in general more iterations for a longer prediction horizon N. This problem exists for both the DEMPC with terminal constraints and without terminal constraints and is also discussed in section 4.4.

4.4 Performance Guarantees

In this section we discuss performance guarantees for DEMPC under inexact minimization. For practical applications we are mostly interested in the economic performance. As motivated in section 3.1, we should not neglect the effect of inexact minimization, if we consider dual distributed optimization. Therefore, extending economic performance guarantees to suboptimal DEMPC is a relevant contribution.
The first part of this section considers DEMPC with terminal cost and sets and extends the performance results in [GrünePanin15] to take inexact optimization into account. The second part considers DEMPC without terminal constraints and extends the transient performance guarantees given in [GrüneStieler14]. In the third part, these theoretical results are interpreted and consequences for practical applications are discussed.

4.4.1 Performance Guarantees with terminal sets and terminal costs

We first consider DEMPC with terminal costs and sets from section 2.2. For this setup we derive economic performance results under inexact minimization, which are extensions of the results given in [GrünePanin15]. A prerequisite for these results is a decrease condition in $\mathcal{V}^*$ under the suboptimal MPC-feedback μ_ϵ. To this end, the results in section 4.2 can be used.
We first need to define some notation. We denote the closed-loop cost of applying the suboptimal MPC by

$$J_K^{\mathrm{cl}}(x,\mu_\epsilon) = \sum_{k=0}^{K-1} l(x_{\mu_\epsilon}(k,x),\mu_\epsilon(x_{\mu_\epsilon}(k,x))).$$

Let $J_\infty^{\mathrm{cl}}(x,\mu_\epsilon)$ be the infinite closed-loop MPC cost. Furthermore, we define the control set

$$\mathcal{U}_{\mathcal{B}_c}^K(x) = \{u \in \mathcal{U}^K | x_u(K,x) \in \mathcal{B}_c, x_u(k,x) \in \mathcal{X}, k \in \{0,\dots,K\}\}.$$

Denote $\mathcal{X}_N$ as the feasible set for MPC with prediction horizon N, i.e.

$$\mathcal{X}_N = \{x | \mathcal{U}_{\mathcal{X}_f}^N(x) \neq \varnothing\},$$

and correspondingly $\mathcal{X}_\infty$ denotes the infinite horizon feasible set. $J_K^{\mathrm{uc}}(x,u)$ is the open-loop cost of applying the input sequence u over a prediction horizon

K, without considering a terminal cost or constraint. $V_\infty^{\mathrm{uc}}(x)$ is the corresponding optimal infinite horizon cost, i.e. $V_\infty^{\mathrm{uc}}(x) = \limsup_{K\to\infty} \inf_{u\in\mathcal{U}^K(x)} J_K^{\mathrm{uc}}(x,u)$. First, the average infinite horizon performance is analysed. Then in theorem 50, the un-averaged infinite horizon performance is bounded, by placing a specific bound on the suboptimality. In theorem 53, the transient performance is bounded, using the same assumption on the suboptimality. In theorem 55, an alternative transient performance guarantee is derived, with a less restrictive assumption on the suboptimality.

Averaged infinite horizon Performance

We first consider the averaged infinite horizon performance. Therefore we use the stability results of DEMPC with terminal sets and terminal cost under inexact minimization, as derived in section 4.2. For the nominal case the average performance has been analysed in [AngeliAmritRawlings12] and it was shown, that the averaged closed-loop performance is no worse, than the optimal steady state. For $\eta(x) < \tilde{l}(x,\mu_\epsilon(x))$ we can use Corollary 33 to conclude asymptotic stability of the optimal steady state, which implies the existence of a $\beta_\epsilon \in \mathcal{KL}$ such that $\|x_{\mu_\epsilon}(k,x)\| \leq \beta_\epsilon(\|x\|,k)$. This implies an average performance no worse, than the optimal steady state. For a more general suboptimality $\eta(k)$ we cannot guarantee asymptotic stability, but we still can bound the average infinite horizon performance with

$$\begin{aligned}\bar{J}_\infty^{\mathrm{cl}}(x,\mu_\epsilon) &= \limsup_{T\to\infty} \frac{1}{T}\sum_{k=0}^{T-1} l(x_{\mu_\epsilon}(k,x),\mu_\epsilon(x_{\mu_\epsilon}(k,x))) \\ &\leq \limsup_{T\to\infty} \frac{1}{T}\left(\mathcal{V}^*(x) - \mathcal{V}^*(x_{\mu_\epsilon}(T,x)) + \sum_{k=0}^{T-1}\eta(k)\right) = \sup av[\eta],\end{aligned}$$

where $av[\eta]$ is the average suboptimality. For both results we require that η converges to 0, to get an averaged performance equal to the optimal steady state.

Non-averaged infinite horizon Performance

Now we focus on the non-averaged infinite horizon performance and derive a bound on the suboptimality with respect to the infinite horizon MPC. In order to have a finite value, the average performance has to be zero and thus the suboptimality has to convergence to 0. We consider

$$\eta(x) \leq c\tilde{l}(x,\mu_\epsilon(x)) \leq c\alpha_l(\|x\|),$$

with $c \in (0,1)$. We require the following additional assumption:

Assumption 49. *(a) The terminal cost satisfies*

$$|V_f(x)| \leq \gamma_f(\|x\|).$$

(b) There exists $N_0 \in \mathbb{N}$, $\delta > 0$ such that $\mathcal{X}_{N_0}$ contains the Ball $\mathcal{B}_\delta$.

(c) There exists $\gamma_\lambda \in \mathcal{K}_\infty$ such that

$$|\lambda(x)| \leq \gamma_\lambda(\|x\|).$$

(d) There exists $\gamma_v \in \mathcal{K}_\infty$ such that

$$|\mathcal{V}^*(x)| \leq \gamma_V(\|x\|).$$

Assumption (d) might not seem intuitive, but by using the fact, that $\tilde{\mathcal{V}}^*(x)$ is a Lyapunov function and that the difference is bounded by $\gamma_\lambda(\|x\|)$, this assumption can be satisfied. Now we can state a theorem for the infinite horizon cost under suboptimal MPC feedback. This theorem is a modified version of theorem 5.1 in [GrünePanin15], that includes suboptimality.

Theorem 50. *Let assumptions 1,2,4 and 49 hold. Assume further, that*

$$\mathcal{V}^*(A_k x + B\mu_\epsilon(x)) - \mathcal{V}^*(x) \leq -l(x,\mu_\epsilon(x)) + \eta(x),$$

holds for all $x \in \mathcal{X}_N$ and that $\eta(x) \leq c\tilde{l}(x,\mu_\epsilon(x))$ with $c \in (0,1)$. Then we have

$$J_\infty^{cl}(x,\mu_\epsilon) \leq \mathcal{V}^*(x) + \frac{c}{1-c}\tilde{\mathcal{V}}^*(x) \leq \frac{1}{1-c}(V_\infty^{uc}(x) + \delta(N)) + \frac{c}{1-c}\gamma_\lambda(\|x\|)$$

for all $x \in \mathcal{X}_N$ with $\delta(N) \in \mathcal{L}$.

Proof. The proof consists of two parts. First, the closed-loop cost $J_\infty^{\mathrm{cl}}(x,\mu_\epsilon)$ is bounded by using the decrease condition in $\mathcal{V}^*(x)$. Then the infinite horizon cost $V_\infty^{\mathrm{uc}}(x)$ is bounded by using a candidate input sequence u_ϵ and the dynamic programming principle.

Part 1: Starting with

$$\mathcal{V}^*(A_k x + B\mu_\epsilon(x)) - \mathcal{V}^*(x) \leq -l(x,\mu_\epsilon(x)) + \eta(x),$$

we have

$$
\begin{aligned}
J_K^{\text{cl}}(x,\mu_\epsilon) =& \sum_{k=0}^{K-1} l(x_{\mu_\epsilon}(k,x),\mu_\epsilon(x_{\mu_\epsilon}(k,x))) \\
\leq& \mathcal{V}^*(x) - \mathcal{V}^*(x_{\mu_\epsilon}(K,x)) + \sum_{k=0}^{K-1} \eta(x_{\mu_\epsilon}(k,x)) \\
\leq& \mathcal{V}^*(x) - \mathcal{V}^*(x_{\mu_\epsilon}(K,x)) + c\sum_{k=0}^{K-1} \tilde{l}(x_{\mu_\epsilon}(k,x),\mu_\epsilon(x_{\mu_\epsilon}(k,x))) \\
=& \mathcal{V}^*(x) - \mathcal{V}^*(x_{\mu_\epsilon}(K,x)) + c\underbrace{\sum_{k=0}^{K-1} l(x_{\mu_\epsilon}(k,x)),\mu_\epsilon(x_{\mu_\epsilon}(k,x))}_{J_K^{\text{cl}}(x,\mu_\epsilon)} \\
&+ c\lambda(x) - c\lambda(x_{\mu_\epsilon}(K,x)) \\
\leq& \frac{1}{1-c}\left(\mathcal{V}^*(x) - \mathcal{V}^*(x_{\mu_\epsilon}(K,x)) + c\lambda(x) - c\lambda(x_{\mu_\epsilon}(K,x))\right). \quad (4.8)
\end{aligned}
$$

With $\eta(x) \leq c\tilde{l}(x,\mu_\epsilon(x))$ the system is asymptotically stable, which implies

$$\|x_{\mu_\epsilon}(k,x)\| \leq \beta_\epsilon(\|x\|,k) \leq \beta_\epsilon(M,k) =: \sigma_\epsilon(k),$$

with $M = \max_{x\in\mathcal{X}} \|x\|$ and $\sigma_\epsilon \in \mathcal{L}$, $\beta_\epsilon \in \mathcal{KL}$. Using assumption 49 (d) we have

$$|\mathcal{V}^*(x_{\mu_\epsilon}(K,x))| \leq \gamma_V(\sigma_\epsilon(K)). \quad (4.9)$$

By taking the limit $K \to \infty$ we have

$$\lim_{K\to\infty} \|\mathcal{V}^*(x_{\mu_\epsilon}(K,x))\| = \lim_{K\to\infty} \gamma_V(\sigma_\epsilon(K)) = 0.$$

Using $\tilde{\mathcal{V}}^*(x) = \mathcal{V}^*(x) + \lambda(x)$ we get

$$J_\infty^{cl}(x,\mu_\epsilon) \leq \frac{1}{1-c}\left(\mathcal{V}^*(x) + c\lambda(x)\right) = \mathcal{V}^*(x) + \frac{c}{1-c}\tilde{\mathcal{V}}^*(x),$$

which concludes the first part of the proof.
Part 2: For the second inequality we need an intermediate result.

Lemma 51. *([GrünePanin15] Corollary 4.4) Let assumption 1,2,4 and 49 hold. There exists $\sigma \in \mathcal{L}$ such that $\forall x \in \mathcal{X}_\infty, u \in \mathcal{U}^\infty(x)$ with $J_\infty^{uc}(x,u) \leq V_\infty^{uc}(x) + 1$ and for any K, $p \in \mathbb{N}$, there is a $k \in \mathbb{N}$ with $p \leq k \leq K + p$ such that $\|x_u(k,x)\| \leq \sigma(K)$.*

Now we choose N_0, δ based on assumption 49(b), fix $\epsilon \in (0,1)$ and an admissible control u_ϵ that satisfies $J_\infty^{\text{uc}}(x,u_\epsilon) \leq V_\infty^{\text{uc}} + \epsilon$. Then for $N \geq 2N_0$ we use lemma 51 with $K = \lfloor N/2 \rfloor$. Therefore the exists a $k \in \{0,\dots,K-1\}$ such that $\|x_{u_\epsilon}(k,x)\| \leq \sigma(K) \leq \sigma(N_0)$. For $\sigma(N_0) \leq \delta$ we have $x_{u_\epsilon}(k,x) \in \mathcal{X}_{N_0} \subset \mathcal{X}_{N_1}$ for $N_1 \geq N_0$. Correspondingly for $N_1 = N-k$ we have $u_\epsilon \in \mathcal{U}^k_{\mathcal{X}_{N-k}}(x)$. By using assumption 49 (d) and lemma 51 we have

$$|\mathcal{V}^*_{N-k}(x_{u_\epsilon}(k,x))| \leq \gamma_V(\sigma(K)).$$

To get a bound on the best possible infinite horizon performance, we need the following intermediate lemma.

Lemma 52. *([GrünePanin15] Lemma 4.5)*
Let assumption 1,2,4 and 49 hold. Then $V_\infty^{uc}(x) \geq -\lambda(x)$ hold $\forall x \in \mathcal{X}_\infty$.

Using lemma 52 with assumption 49 (c) we have

$$V_\infty^{\text{uc}}(x) \geq -\gamma_\lambda(\|x\|).$$

With this the candidate input from lemma 51 satisfies

$$V_\infty^{\text{uc}}(x) + \epsilon \geq J_k^{\text{uc}}(x,u_\epsilon) + V_\infty^{\text{uc}}(x_{u_\epsilon}(k,x)) \geq J_k^{\text{uc}}(x,u_\epsilon) - \gamma_\lambda(\sigma(K)).$$

Using the dynamic programming principle we get

$$\begin{aligned}\mathcal{V}^*(x) &= \inf_{u \in \mathcal{U}^k_{\mathcal{X}_{N-k}}(x)} \{J_k^{\text{uc}}(x,u) + \mathcal{V}^*_{N-k}(x_u(k,x))\} \\ &\leq J_k^{\text{uc}}(x,u_\epsilon) + \mathcal{V}^*_{N-k}(x_{u_\epsilon}(k,x)) \\ &\leq V_\infty^{\text{uc}}(x) + \gamma_V(\sigma(K)) + \gamma_\lambda(\sigma(K)) + \epsilon.\end{aligned}$$

Since ϵ was arbitrary, we have

$$\begin{aligned}&\mathcal{V}^*(x) \leq V_\infty^{\text{uc}}(x) + \delta(N), \\ &\delta(N) = \gamma_V(\sigma(\lfloor N/2 \rfloor)) + \gamma_\lambda(\sigma(\lfloor N/2 \rfloor)).\end{aligned}$$

For the rotated cost $\tilde{\mathcal{V}}^*(x) = \mathcal{V}^*(x) + \lambda(x)$ we have correspondingly

$$\tilde{\mathcal{V}}^*(x) \leq V_\infty^{\text{uc}}(x) + \delta(N) + \gamma_\lambda(\|x\|).$$

The final inequality is obtained by combing the two results. □

To satisfy the assumption of the cost decrease, the result in section 4.2 can be used. Note that if we use the pre-stabilized dynamics, the bound on the closed-loop performance considers only the additional input $v = \mu_\epsilon(x)$ and not the full input $u = \mu_{N,\epsilon}(x) + K_t x$, i.e. $l(x,u) \neq l(x,\mu_\epsilon)$.
For $c = 0$ ($\eta = 0$) we recover the result in [GrünePanin15], which states that the infinite horizon performance difference is bounded by a $\mathcal{L}$-function of N. This means that by increasing the prediction horizon N the MPC performance will approach the infinite horizon performance. From the suboptimality η, c we can see a deterioration of the performance, if $c \to 1$. This corresponds to the average result and stability result, that only holds for $c < 1$.

Transient Performance

Now we will consider the transient performance over a horizon K with inexact minimization. Recall from theorem 6 that the state trajectory under the nominal MPC feedback satisfies $\|x_\mu(k,x)\| \leq \beta(\|x\|,k)$ with $\beta \in \mathcal{KL}$. The following theorem gives a bound on the transient performance $J_K^{\mathrm{cl}}(x,\mu_\epsilon)$ with respect to the optimal performance $J_K^{\mathrm{uc}}(x,u)$. This theorem is a modified version of theorem 5.2 in [GrünePanin15], that includes the suboptimality η.

Theorem 53. *Let assumptions 1,2,4 and 49 hold. Assume further, that*

$$\mathcal{V}^*(A_k x + B\mu_\epsilon(x)) - \mathcal{V}^*(x) \leq -l(x,\mu_\epsilon(x)) + \eta(x).$$

and that $\eta(x) \leq c\tilde{l}(x,\mu_\epsilon(x))$, with $c \in (0,1)$.
Then there exists a $\delta_1,\delta_2 \in \mathcal{L}$ such that for all $x \in \mathcal{X}_N$ we have

$$J_K^{cl}(x,\mu_\epsilon) \leq \frac{1}{1-c}\left(\inf_{u \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J_K^{uc}(x,u) + \delta_1(N) + \delta_2(K) + c(\gamma_\lambda(\sigma_\epsilon(K)) + \lambda(x))\right),$$

with $\kappa = \beta(\|x\|,K)$ and

$$\delta_1(N) = \gamma_V(\sigma(\lfloor N/2 \rfloor)) + \gamma_\lambda(\sigma(\lfloor N/2 \rfloor)),$$
$$\delta_2(K) = \gamma_V(\sigma(K)) + \gamma_\lambda(\sigma(K)) + \gamma_V(\beta(M,K)) + \gamma_\lambda(\beta(M,k)) + \gamma_V(\beta_\epsilon(M,K)),$$

with $M = \max_{x\in\mathcal{X}} \|x\|$.

Proof. This proof consists of two parts. First, a bound on $\mathcal{V}^*(x)$ is derived. (This part is the same as in theorem 5.2 in [GrünePanin15].) Then a bound on the closed-loop cost with the inexact DEMPC is computed based on this bound.

Part 1: For the first part, we need a δ_1, $\tilde{\delta}_2 \in \mathcal{L}$ such that

$$\mathcal{V}^*(x) \leq \inf_{u \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J^{\mathrm{uc}}_K(x,u) + \delta_1(N) + \tilde{\delta}_2(K).$$

For this we require an intermediate lemma.

Lemma 54. *([GrünePanin15] Lemma. 4.6)*
Let assumptions 1,2,4 and 49 hold and fix $\kappa_0 \geq 0$. Then for any $\kappa \in (0,\kappa_0]$, any $x \in \mathcal{X}$, $K_0 \in \mathbb{N}$ with $\beta(\|x\|,K_0) \leq \kappa$ the following inequalities hold.

(a) For all $K \geq K_0$ the inequality

$$\inf_{u \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J^{uc}_K(x,u) \leq \gamma_V(\|x\|) + \gamma_V(\kappa)$$

holds with $\gamma_V \in \mathcal{K}_\infty$ from assumption 49 (d).

(b) For all $K \in \mathbb{N}$ with $\mathcal{U}^K_{\mathcal{B}_\kappa}(x) \neq \varnothing$ we have

$$\lambda(x) - \gamma_\lambda(\kappa) \leq \inf_{u \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J^{uc}_K(x,u).$$

(c) There exists a $\sigma \in \mathcal{L}$ such that for all $K \geq K_0$, all $P \in \mathbb{N}$, any $u \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)$ with

$$J^{uc}_K(x,u) \leq \inf_{u \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J^{uc}_K(x,u) + 1$$

there is a $k \leq \min\{P,K-1\}$ such that $\|x_u(k,x)\| \leq \delta(\min\{P,K\})$.

Using lemma 54 (c) with $P = \lfloor N/2 \rfloor$, we use a candidate input $u_\epsilon \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)$ with

$$J^{\mathrm{uc}}_K(x,u_\epsilon) \leq \inf_{u \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J^{\mathrm{uc}}_K(x,u) + \epsilon$$

and an arbitrary $\epsilon \in (0,1]$. Therefore there exists a $k \in \{0,\ldots,\lfloor N/2 \rfloor\}$, $k \leq K-1$ with $\|x_{u_\epsilon}(k,x)\| \leq \sigma(\min\{P,K-1\})$. Since u_ϵ steers x into $\mathcal{B}_\kappa$ the shifted input satisfies $u_\epsilon(k+\cdot) \in \mathcal{U}^{K-k}_{\mathcal{B}_\kappa}(x_{u_\epsilon}(k,x))$ and thus $\mathcal{U}^{K-k}_{\mathcal{B}_\kappa}(x_{u_\epsilon}(k,x)) \neq \varnothing$. Now we can apply lemma 54 (b) to conclude

$$J^{\mathrm{uc}}_{K-k}(x_{u_\epsilon}(k,x),u_\epsilon(k+\cdot)) \geq -\gamma_\lambda(\sigma(\min\{P,K-1\})) - \gamma_\lambda(\kappa).$$

This implies

$$\inf_{u\in\mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J^{\mathrm{uc}}_K(x,u)+\epsilon \geq J^{\mathrm{uc}}_K(x,u_\epsilon) = J^{\mathrm{uc}}_K(x,u_\epsilon) + J^{\mathrm{uc}}_{K-k}(x_{u_\epsilon}(k,x),u_\epsilon(k+\cdot))$$
$$\geq J^{\mathrm{uc}}_K(x,u_\epsilon) - \gamma_\lambda(\sigma(\min\{N,K-1\})) - \gamma_\lambda(\kappa)$$

By choosing a sufficiently large N, K we have $\sigma(\min\{P,K\}) \leq \delta$, with δ from assumption 49 (b), we have $u_\epsilon \in \mathcal{U}^k_{\mathcal{X}_Q}(x)$ for $Q \geq N_0$, with N_0 from assumption 49 (b). By choosing $N \geq 2N_0$, we have $N-k \geq N_0$ and thus $u_\epsilon \in \mathcal{U}^k_{\mathcal{X}_{N-k}}(x)$.
With this inequality and the dynamic programming principle, we have

$$\begin{aligned}\mathcal{V}^*(x) &= \inf_{u\in\mathcal{U}^k_{\mathcal{X}_{N-k}}(x)} \{J^{\mathrm{uc}}_k(x,u)+\mathcal{V}^*_{N-k}(x_u(k,x))\} \leq J^{\mathrm{uc}}_k(x,u_\epsilon)+\mathcal{V}^*_{N-k}(x_{u_\epsilon}(k,x)) \\ &\leq \inf_{u\in\mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J^{\mathrm{uc}}_K(x,u) + \gamma_V(\sigma(\min\{P,K-1\})) + \gamma_V(\kappa) \\ &\quad + \gamma_\lambda(\sigma(\min\{P,K-1\})) + \gamma_\lambda(\kappa) + \epsilon.\end{aligned}$$

With

$$\delta_1(N) = \gamma_V(\sigma(\lfloor N/2\rfloor)) + \gamma_\lambda(\sigma(\lfloor N/2\rfloor))$$
$$\tilde{\delta}_2(K) = \gamma_V(\sigma(K)) + \gamma_\lambda(\sigma(K)) + \gamma_V(\beta(M,K)) + \gamma_\lambda(\beta(M,k))$$

and $M = \sup_{x\in\mathcal{X}} \|x\|$ we have

$$\mathcal{V}^*(x) \leq \inf_{u\in\mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J^{\mathrm{uc}}_K(x,u) + \delta_1(N) + \tilde{\delta}_2(K).$$

Part 2:
Now we use the bound on $\mathcal{V}^*$ to derive a bound on the transient performance. From the proof of theorem 50: (4.8), (4.9) we have

$$J^{\mathrm{cl}}_K(x,\mu_\epsilon) \leq \frac{1}{1-c}\left(\mathcal{V}^*(x) - \mathcal{V}^*(x_{\mu_\epsilon}(K,x)) + c\lambda(x) - c\lambda(x_{\mu_\epsilon}(K,x))\right)$$

and

$$\|\mathcal{V}^*(x_{\mu_\epsilon}(K,x))\| \leq \gamma_V(\sigma_\epsilon(K)),$$

due to the stability $\|x_{\mu_\epsilon}(k,x)\| \leq \beta_\epsilon(\|x\|,k)$ and assumption 49 (d). This implies

$$J_K^{\text{cl}}(x,\mu_\epsilon) \leq \frac{1}{1-c}\Bigg(\inf_{u\in\mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J_K^{\text{uc}}(x,u) + \delta_1(N) + \tilde{\delta}_2(K) + \gamma_V(\sigma_\epsilon(K)) \\ + c\lambda(x) + c\gamma_\lambda(\sigma_\epsilon(K))\Bigg).$$

With $\delta_2(K) = \gamma_V(\sigma_\epsilon(K)) + \tilde{\delta}_2(K)$ we have

$$J_K^{\text{cl}}(x,\mu_\epsilon) \leq \frac{1}{1-c}\left(\inf_{u\in\mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J_K^{\text{uc}}(x,u) + \delta_1(N) + \delta_2(K) + c(\gamma_\lambda(\sigma_\epsilon(K)) + \lambda(x))\right).$$

□

For $K \to \infty$ this theorem is equivalent to the previous theorem 50 on the infinite horizon performance. If we compare this result, to the nominal result in [GrünePanin15], the main difference is the factor $\frac{1}{1-c}$. The other additive therm $c(\gamma_\lambda(\sigma_\epsilon(K)) + \lambda(x))$ depends also on the initial state x, and is linear in the suboptimality c. Furthermore, for $c = 0$ ($\eta = 0$) and thus $\beta = \beta_\epsilon$ we recover the result from [GrünePanin15] theorem 5.2.

Transient Performance - II

So far all performance results relied on asymptotic stability of the system under the suboptimal MPC feedback μ_ϵ with $\eta(x) < \tilde{l}(x,\mu_\epsilon(x))$. Now we study the performance by considering a more general constant upper bound $\overline{\eta}$ on the suboptimality, for which we can only establish practical asymptotic stability. As already discussed the averaged infinite performance is equal to the average suboptimality $av[\eta]$, which in general leads to a non-finite value for the non-averaged infinite horizon performance. The following theorem extends the transient performance result in theorem 55 to a constant suboptimality bound $\eta(k) \leq \overline{\eta}$.

Theorem 55. *Let assumptions 1,2,4 and 49 hold. Assume further, that*

$$\mathcal{V}^*(A_k x + B\mu_\epsilon(x)) - \mathcal{V}^*(x) \leq -l(x,\mu_\epsilon(x)) + \eta(x).$$

with a constant bound $\eta(x) \leq \overline{\eta}$. Then we have

$$\|x_{\mu_\epsilon}(k,x)\| \leq \max\{\beta_\epsilon(k,\|x\|),\epsilon_\eta(\overline{\eta})\}.$$

with $\beta_\epsilon \in \mathcal{KL}$ and $\epsilon_\eta \in \mathcal{K}$. Furthermore, there exists a δ_1, $\delta_2 \in \mathcal{L}$ such that for all $x \in \mathcal{X}_N$ we have

$$\begin{aligned} J_K^{cl}(x,\mu_\epsilon) \leq & \inf_{u \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J_K^{uc}(x,u) + \delta_1(N) + \tilde{\delta}_2(K) \\ & + \gamma_V(\max\{\sigma_\epsilon(K),\epsilon_\eta(\overline{\eta})\}) + \sum_{k=0}^{K-1} \eta(k), \end{aligned}$$

with $\kappa = \beta(\|x\|,K)$, $\sigma_\epsilon(k) = \beta_\epsilon(M,k)$ and

$$\begin{aligned} \delta_1(N) &= \gamma_V(\sigma(\lfloor N/2 \rfloor)) + \gamma_\lambda(\sigma(\lfloor N/2 \rfloor)), \\ \tilde{\delta}_2(K) &= \gamma_V(\sigma(K)) + \gamma_\lambda(\sigma(K)) + \gamma_V(\beta(M,K)) + \gamma_\lambda(\beta(M,k)), \end{aligned}$$

with $M = \max_{x \in \mathcal{X}} \|x\|$.

Proof. This proof consists of three parts. First, practical asymptotic stability with β_ϵ, ϵ_η is established. Then the optimal open loop cost J_K^{uc} is bounded based on the value function $\mathcal{V}^*$. In the third part the closed-loop cost J_K^{cl} is bounded based on $\mathcal{V}^*$.

Part 1:

For the practical asymptotic stability we consider the rotated version of the cost decrease in $\mathcal{V}^*$. By adding $\lambda(x^+) - \lambda(x)$ on both sides we get

$$\tilde{\mathcal{V}}^*(x^+) - \tilde{\mathcal{V}}^*(x) \leq -\tilde{l}(x,\mu_\epsilon(x)) + \eta(x) \leq -\alpha_l(\|x\|) + \overline{\eta}.$$

Recall that $\alpha_l \in \mathcal{K}_\infty$ and from theorem 6, that $\tilde{\mathcal{V}}^*$ is a candidate Lyapunov function (positive definite). Now by using lemma 14 ([GrüneStieler14] Thm. 2.4), we get

$$\|x_{\mu_\epsilon}(k,x)\| \leq \max\{\beta_\epsilon(k,\|x\|),\epsilon_\eta(\overline{\eta})\},$$

with $\beta_\epsilon \in \mathcal{KL}$ and $\epsilon_\eta \in \mathcal{K}$.

Part 2:

This part equivalent to part 1 in the proof of theorem 55 and yields

$$\mathcal{V}^*(x) \leq \inf_{u \in \mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J_K^{\mathrm{uc}}(x,u) + \delta_1(N) + \tilde{\delta}_2(K).$$

Part 3:
Now we use the bound on $\mathcal{V}^*$ to derive a bound on the transient performance.

$$J_K^{\text{cl}}(x,\mu_\epsilon) = \sum_{k=0}^{K-1} l(x_{\mu_\epsilon}(k,x),\mu_\epsilon(x_{\mu_\epsilon}(k,x))) \leq \mathcal{V}^*(x) - \mathcal{V}^*(x_{\mu_\epsilon}(K,x)) + \sum_{k=0}^{K-1} \eta(k)$$

With assumption 49 (d) and the practical stability result of part 1, we have

$$\|\mathcal{V}^*(x_{\mu_\epsilon}(K,x))\| \leq \gamma_V(\max\{\sigma_\epsilon(K),\epsilon_\eta(\overline{\eta}\}),$$

with $\sigma_\epsilon(k) = \beta_\epsilon(M,k)$, $M = \max_{x\in\mathcal{X}} \|x\|$. By combining these bounds we get

$$J_K^{\text{cl}}(x,\mu_\epsilon) \leq \inf_{u\in\mathcal{U}^K_{\mathcal{B}_\kappa}(x)} J_K^{\text{uc}}(x,u) + \delta_1(N) + \tilde{\delta}_2(K)$$
$$+ \gamma_V(\max\{\sigma_\epsilon(K),\epsilon_\eta(\overline{\eta}\}) + \sum_{k=0}^{K-1} \eta(k).$$

□

For $\overline{\eta} = 0$ and thus $\sigma_\epsilon = \sigma$, $\epsilon_\eta = 0$ we recover the result from [GrünePanin15] theorem 5.2.
For $K \to \infty$ we recover the result on the average performance $av[\eta] \leq \overline{\eta}$.
The most critical error term is $\sum_{k=0}^{K-1} \eta(k)$, since it increases with K. This result is similar to the transient performance guarantee for EMPC without terminal constraints (Thm. 4.1 [GrüneStieler14]), where we have one term $K \cdot \delta_1(N)$, that represents the fact, that we do not attain the optimal steady state. Intuitively this means, that the effect of a suboptimality in the optimization η to terminal constraint EMPC, is similar to the effect of the prediction horizon N for EMPC without terminal constraints. Both lead to practical asymptotic stability, instead of asymptotic stability. Furthermore, due to the suboptimal operation off steady state, this leads to an additional cost term in the transient performance, that scales with the interval K.

4.4.2 Performance Guarantees without terminal constraints

Now we consider DEMPC without terminal constraints from section 2.3. First, the average infinite horizon performance is analysed. Then the transient performance is bounded, by assuming some bound on the suboptimality. The following result bounds the performance of DEMPC without terminal constraints under inexact optimization.

Averaged infinite horizon Performance

We first consider the averaged infinite horizon performance. To this end we use

$$J_K^{\mathrm{cl}}(x,\mu_{N,\epsilon}) \leq \mathcal{V}_N^*(x) - \mathcal{V}_N^*(x_{\mu_{N,\epsilon}}(K)) + \delta(N) + \eta$$

from theorem 48. This yields the following bound on the averaged infinite horizon performance:

$$\begin{aligned}\overline{J}_\infty^{\mathrm{cl}}(x,\mu_{N,\epsilon}) &= \limsup_{T\to\infty} \frac{1}{T} J_k^{\mathrm{cl}}(x,\mu_{N,\epsilon}) \\ &\leq \limsup_{T\to\infty} \frac{1}{T}\left(\mathcal{V}_N^*(x) - \mathcal{V}_N^*(x_{\mu_{N,\epsilon}}(T,x)) + \sum_{k=0}^{T-1}\eta(k)\right) + \delta(N) \\ &= \sup av[\eta] + \delta(N).\end{aligned}$$

Compared to the nominal result we have an additive term corresponding to the average suboptimality.

Transient Performance

Since we have a non-zero average infinite horizon performance, we do not investigate the infinite horizon performance further and directly consider the transient performance. The following result is an extension of theorem 4.1 in [GrüneStieler14], that includes a suboptimality η with a constant upper bound $\overline{\eta}$. To this end we use the stability result of theorem 48 derived in section 4.3.

Theorem 56. *Assume that the rotated cost function $\tilde{\mathcal{V}}^*$ satisfies*

$$\begin{aligned}\tilde{\mathcal{V}}_N^*(Ax + B\mu_{N,\epsilon}(x)) - \tilde{\mathcal{V}}_N^*(x) &\leq -\alpha_l(\|x\|) + \tilde{\delta}(N) + \eta(x), \\ \alpha_l(\|x\|) \leq \tilde{\mathcal{V}}_N^*(x) &\leq \alpha_V(\|x\|),\end{aligned}$$

with $\eta(x) \leq \overline{\eta}$. Assume further that the system is practical asymptotically stable

$$\|x_{\mu_{N,\epsilon}}(k,x)\| \leq \max\{\beta_{N,\epsilon}(\|x\|,k),\epsilon_\eta\},$$

with $\beta_{N,\epsilon} \in \mathcal{KL}$ and that assumption 49 (c) holds. Let $\kappa := \|x_{\mu_{N,\epsilon}}(K,x)\| \leq \max\{\beta_{N,\epsilon}(\|x\|,K),\epsilon_\eta\}$, then

$$J_K^{cl}(x,\mu_{N,\epsilon}) \leq \inf_{u\in\mathcal{U}_\kappa^K(x)} J_K(x,u) + \alpha_V(\kappa) + 2\gamma_\lambda(\kappa) + K\delta_1(N) + \sum_{k=0}^{K-1}\eta(k)$$

holds for all K, $N \in \mathbb{N}$, and all $x \in \mathcal{X}_N$.

Proof. Starting with the practical Lyapunov function $\tilde{\mathcal{V}}^*$ we have

$$\tilde{\mathcal{V}}_N^*(Ax + B\mu_{N,\epsilon}(x)) \leq \tilde{\mathcal{V}}_N^*(x) - \alpha_l(\|x\|) + \tilde{\delta}_1(N) + \eta(x).$$

Summing over the prediction horizons and using induction this yields

$$\sum_{k=0}^{K-1} \tilde{l}(x_{\mu_{N,\epsilon}}(k,x), \mu_{N,\epsilon}(k,x)) \leq \tilde{\mathcal{V}}_N^*(x) - \tilde{\mathcal{V}}_N^*(x_{\mu_{N,\epsilon}}(K)) + K\tilde{\delta}_1(N) + \sum_{k=0}^{K-1} \eta(k).$$

Using the dynamic programming principle

$$\tilde{\mathcal{V}}_N^*(x) = \inf_{u \in \mathcal{U}^k(x)} \left\{ \tilde{J}_K(x,u) + \tilde{\mathcal{V}}_{N-K}^*(x_u(K,x)) \right\}$$

and the bounds on $\tilde{\mathcal{V}}_N^*$

$$\alpha_l(\|x\|) \leq \tilde{\mathcal{V}}_N^*(x) \leq \alpha_V(\|x\|)$$

we get for all $K \in \{1, \ldots, N\}$ with $u \in \mathcal{U}_{\mathcal{B}_\epsilon}^K(x)$

$$\tilde{J}_K(x,u) = \underbrace{\tilde{J}_K(x,u) + \tilde{\mathcal{V}}_{N-K}^*(x_u(K,x))}_{\geq \tilde{\mathcal{V}}_N^*(x)} - \underbrace{\tilde{\mathcal{V}}_{N-K}^*(x_u(K,x))}_{\leq \alpha_V(\epsilon)} \geq \tilde{\mathcal{V}}_N^*(x) - \alpha_V(\epsilon).$$

Going back to the original cost we have

$$\sum_{k=0}^{K-1} \tilde{l}(x_u(k,x), u(k)) = \tilde{J}_K(x,u) = \lambda(x) + J_K(x,u) - \lambda(x_u(K,x)).$$

Now for all $u \in \mathcal{U}^{K}_{\mathcal{B}_\kappa}(x)$ we get

$$
\begin{aligned}
J^{\mathrm{cl}}_K(x,\mu_{N,\epsilon}) = &\sum_{k=0}^{K-1} \tilde{l}(x_{\mu_{N,\epsilon}}(k,x),\mu_{N,\epsilon}(k,x)) - \lambda(x) + \lambda(x_{\mu_{N,\epsilon}}(K,x)) \\
\leq &\tilde{\mathcal{V}}^*_N(x) - \tilde{\mathcal{V}}_N(x_{\mu_{N,\epsilon}}(K,x)) + K\delta_1(N) + \sum_{k=0}^{K-1} \eta(k) \\
&- \lambda(x) + \lambda(x_{\mu_{N,\epsilon}}(K,x)) \\
\leq &\tilde{J}_K(x,u) + \alpha_V(\kappa) - \tilde{\mathcal{V}}_N(x_{\mu_{N,\epsilon}}(K,x)) + K\delta_1(N)) + \sum_{k=0}^{K-1} \eta(k) \\
&- \lambda(x) + \lambda(x_{\mu_{N,\epsilon}}(K,x)) \\
= &J_K(x,u) + \alpha_V(\kappa) - \tilde{\mathcal{V}}^*_N(x_{\mu_{N,\epsilon}}(K,x)) + K\delta_1(N) + \sum_{k=0}^{K-1} \eta(k) \\
&- \lambda(x_u(K,x)) + \lambda(x_{\mu_{N,\epsilon}}(K,x)) \\
\leq &J_K(x,u) + \alpha_V(\kappa) + K\delta_1(N) + \sum_{k=0}^{K-1} \eta(k) + 2\gamma_\lambda(\kappa).
\end{aligned}
$$

□

The proof of the transient performance is similar to theorem 4.1 in [GrüneStieler14], by replacing $\delta_1(N)$ with $\tilde{\delta}_1(N) + \eta$. The main ingredient is the practical asymptotic stability proof under inexact minimization as derived in section 4.3. This theorem states, that the effect of the suboptimality due to inexact optimization η, on the transient performance, is similar to that of the suboptimality $\tilde{\delta}_1(N)$ with respect to the infinite horizon MPC.

4.4.3 Interpretation

Now we discuss the implications of these results for the application of DEMPC in comparison to the nominal results in [GrüneStieler14,GrünePanin15].

DEMPC without terminal constraints

Let us first consider the DEMPC without terminal constraints. The nominal results in [GrüneStieler14] imply, that by increasing the prediction horizon N, we can improve the performance and stability properties and even approach

the infinite horizon EMPC. This means, that if we for example use a time splitting algorithm [FerrantiEtAl15], we can improve the performance at the cost of increasing the computational demand.
Now if we consider the results presented in this section, we get a seemingly similar implication. By increasing the prediction horizon N, the stability properties and performance approach the infinite horizon EMPC, up to a constant depending on the suboptimality η. It is however unreasonable to assume a constant suboptimality η with an increasing prediction horizon N. Most likely that for a fixed number of iterations (limited time), the suboptimality η will increase with the prediction horizon N.
Thus increasing the prediction horizon N without increasing the number of iterations, will lead to a larger suboptimality η and depending on $\delta(N)$ potentially to a worse performance. Even if the computational recources grow proportional to the prediction horizon N, we can conclude, that there is a trade off for how large the prediction horizon N should be. If we can express η as a function of N, then the optimal prediction horizon N^* is such that $\eta(N) + \tilde{\delta}(N)$ is minimal.
More practically speaking this means that the prediction horizon should be chosen large enough, such that we would get a good enough nominal performance. But increasing the prediction horizon too much, will at some point result in worse performance (assuming limited computational resources). Alternatively, one might try to increase the prediction horizon without increasing the suboptimality. For [FerrantiEtAl15] this means that the number of CPUs has to grow faster thann linear with respect to the prediction horizon N, which can only be accomplished to a certain limit.

DEMPC with terminal constraints

Now we consider the results for the DEMPC with terminal cost and terminal constraints. Again starting from the nominal results in [GrünePanin15] we know, that the closed-loop infinite horizon performance compared to the infinite horizon MPC cost V_∞^{uc}, is only worse by a factor $\delta(N)$, which means that for $N \to \infty$ we approach infinite horizon performance. Furthermore, independent of the prediction horizon, we can guarantee asymptotic stability and thus an average optimal operation.
In the case of inexact minimization we distinguish two cases. We first consider $\eta(x) \leq c\alpha_l(\|x\|)$. To ensure this we need to know α_l and the stopping condition for the dual iteration has to be adapted in each time step, to ensure this bound. This requirement is quire similar to the bound on the

suboptimality required for stabilizing MPC [FerrantiEtAl15]. If we approach the optimal steady state this leads to high accuracy demands and thus high computational demands. In [FerrantiEtAl15] this issue is avoided by using the terminal controller inside the terminal set $\mathcal{X}_f$. If we can guarantee this suboptimality η we get similar performance bounds to the nominal case, with the additional factor $\frac{1}{1-c}$.

The more general case $\eta(x) \leq \overline{\eta}$ is a lot easier to facilitate computationally, since we can always assume some bound on the suboptimality η. The price we pay is the loss of asymptotic stability and thus a deterioration of the performance. If we consider short transient performance guarantees, the deterioration in the performance is not necessarily that large. Even though the performance guarantees look quite similar to the DEMPC without terminal constraints, the implications are quite different.

The effect of the prediction horizon N on the performance seems small compared to the suboptimality η. Therefore it might be better to use a short prediction horizon with a small suboptimality than a large prediction horizon with a larger suboptimality. The limitation is that the prediction horizon needs to be large enough to ensure a large domain of attraction.

5 Real time Economic Dispatch for Power Systems with Model Predictive Control

This chapter describes the economic power dispatch problem for power systems and compares the performance of DEMPC to existing 'classical' control structures. The first section introduces the economic power dispatch problem and the corresponding distributed power system dynamics. The second section introduces the classical distributed control structure for power systems, automatic generation control (AGC), and a state of the art extension: economic AGC (EAGC). In the third section we investigate how the theoretical results for DEMPC in the previous chapters and the corresponding assumptions fit into the power system setup. In the fourth section the performance of DEMPC is evaluated in simulation experiments and compared to classical controllers.

5.1 Overview

This section gives a motivation for the usage of DEMPC for power systems and outlines some of the existing controllers.

Motivation

Traditionally the control of power systems is divided into two time scales. The slow economic dispatch (ED) determines the economically optimal operation point based on a steady state optimization. The AGC regulates the frequency and power exchanges to drive the system to the desired operation point.

Several trends can be observed, that motivate the usage of more advanced control methods for distributed power systems. Power systems grow in size and complexity, making centralized control impractical due to the high computational demand and the sharing of confidential information. Therefore a distributed control structure that relies on minimal data exchange is preferable. In the future an increase in renewable energy resources and demand response will lead to larger and faster fluctuations and disturbances in the power grid. This means that transient performance becomes more relevant and the classical controller configuration needs to be modified.

Available Controllers

Several alternative controller configurations have been studied. In [ZhangLi-Papachristodoulou15, ZhaoEtAl16] AGC has been modified to incorporate constraints and performance in a steady state sense and achieve better stabilization. While these approaches can improve the performance, they are mainly steady state oriented and cannot give any guarantees during transient operation.
In contrast, MPC is a dynamic optimization method and can thus give guarantees for transient operation. In [VenkatEtAl08] a distributed MPC was considered to stabilize the power system, which improves the convergence speed. This approach however only considers input constraints, uses primal decomposition and does not consider the economic performance. In [ErsdalEtAl16] centralized MPC is used to stabilize the Nordic power grid and compared to AGC. This MPC considers the same kind of constraints, but only stabilizes the system and does not consider distributed optimization. In comparison, the derived DEMPC uses dual distributed optimization methods, that are scalable and directly considers the economic performance.

5.2 The Economic Dispatch Problem for Power Systems

This section describes the economic dispatch problem for power systems. First the corresponding system dynamics are described. Then the objectives and constraints are discussed.

5.2.1 Distributed Power System Model

A distributed power system network can be modeled as a graph $(\mathcal{N},\mathcal{E})$, where each node $i \in \mathcal{N}$ corresponds to a bus/generator and the edges $(i,j) \in \mathcal{E}$ correspond to the coupled dynamics. The following model is taken from [LiEtAl14]. Each generator/subsystem i is described by its local states

x_i, inputs u_i and disturbances d_i:

$$x_i = \begin{pmatrix} P_i^M \\ \omega_i \\ \delta_i \end{pmatrix}, \quad u_i = P_i^C, \quad d_i = P_i^L,$$

with the local mechanical Power P_i^M, the local frequency deviation ω_i, the local phase shift δ_i, the local power change command P_i^C and the local power load P_i^L. The dynamic coupling between neighboring subsystems (i,j) is based on the branch flow/tie-line power P_{ij}

$$P_{ij} = \frac{|V_i||V_j|}{x_{ij}^2 + r_{ij}^2}(x_{ij} \sin \delta_{ij} - r_{ij} \cos \delta_{ij}),$$

which is a nonlinear function of the phase shift $\delta_{ij} = \delta_i - \delta_j$. Here V_i is the bus voltage, x_{ij} the line inductance and r_{ij} the line resistance. The corresponding local dynamics are given by

$$\begin{aligned} \dot{\omega}_i &= -\frac{1}{M_i}(D_i\omega_i - P_i^M + P_i^L + \sum_{j:i\to j} P_{ij} - \sum_{j:j\to i} P_{ji}), \\ \dot{\delta}_i &= \omega_i, \\ \dot{P}_i^M &= -\frac{1}{T_i}(P_i^M - P_i^C + \frac{1}{R_i}\omega_i). \end{aligned}$$

The corresponding parameters are the generator inertia M_i, a damping constant D_i, the time constant T_i and a constant R_i. With this the distributed nonlinear model can be written as

$$\dot{x}_i = A_i x_i + \sum_{j \in \mathcal{N}_i} g_{ij}(x_{ij}) + B_i u_i + E_i d_i.$$

Controller Model

This nonlinear distributed model tends to be too complex for the controller design. In order to avoid a large model mismatch, the model is successively linearized around the current phase δ_{ij}:

$$P_{ij}(\Delta\delta_{ij}) \approx \underbrace{\frac{|V_i||V_j|}{x_{ij}^2 + r_{ij}^2}(x_{ij}\cos(\delta_{ij}(0)) + r_{ij}\sin(\delta_{ij}(0)))}_{B_{ij}} \Delta\delta_{ij} + P_{ij}(0),$$

with the current tie-line power $P_{ij}(0)$, the small change in the relative phase $\Delta\delta_{ij}$ and a parameter B_{ij}. Theoretically the parameter B_{ij} depends on the current relative phase $\delta_{ij}(0)$, but choosing a constant parameter is a sufficient approximation. With this we have a linear distributed model

$$\dot{x}_i = A_{\mathcal{N}_i}(x_{\mathcal{N}_i} + c_{\mathcal{N}_i}) + B_i u_i + E_i d_i,$$

where $c_{\mathcal{N}_i}$ and d_i are modeled as constant external disturbances. This model is controllable and marginally stable.
For DMPC we require a discrete time distributed linear model. Therefore we use a second order discretization:

$$\begin{aligned} \dot{x} &= A_c(x+c) + B_c u + E_c d, \\ x(t+h) &= (\underbrace{\frac{h^2}{2}A_c^2 + hA_c + I}_{A_d})(x+c) + (\underbrace{\frac{h^2}{2}AB_c + B_c h}_{B_d})u + (\underbrace{\frac{h^2}{2}AE_c + E_c h}_{E_d})d, \end{aligned}$$

which leads to the linear distributed model

$$x^+ = A_{\mathcal{N}_i}(x_{\mathcal{N}_i} + c_{\mathcal{N}_i}) + B_i u_i + E_i d_i.$$

The resulting discretization error is inevitable, since higher order discretizations lead to a further coupling in the dynamics, that destroys the distributed structure and poses difficulties to the distributed online optimization.

5.2.2 Constraints and Objectives

Now we consider the constraints and economic cost function used for the economic dispatch.

Constraints

Simple hyper box constraints on the power are considered

$$P_i^C \in [P_{i,\min}^C, P_{i,\max}^C], \quad P_i^M \in [P_{i,\min}^M, P_{i,\max}^M], \quad P_{ij} \in [P_{ij}^{\min}, P_{ij}^{\max}].$$

This corresponds to local input constraints $u_i \in \mathcal{U}_i$ and coupled local state constraints $x_{\mathcal{N}_i} \in \mathcal{X}_{\mathcal{N}_i}$.

Economic Cost

The economic cost is a quadratic cost for the mechanical power P_i^M, with

$$l_i(x_i,u_i) = a_i(P_i^M)^2.$$

Additional quadratic penalization of the angular frequency ω_i can also be used to ensure small frequency deviations during transients.

Optimal Operation at steady state

The classical economic dispatch problem is an optimization problem to compute the optimal steady state. In particular, a desired frequency ω^* is usually set by a higher level controller and should be reached, which corresponds to $\omega = 0$ and thus steady state operation. For steady state operation the supplied and demanded power must be balanced, which corresponds to

$$\sum_{i=1}^{M} P_i^L = \sum_{i=1}^{M} P_i^M.$$

Therefore the economic stage cost corresponds to a tracking cost of an unreachable set point [RawlingsEtAl08] (except for $P^L = 0$). By shifting the origin to the optimal steady state x^* we get linear dynamics with a linear-quadratic stage cost l_i, which corresponds to the setup in chapter 2. If we only consider this stage cost and the previous constraints the optimal operation would be a quasi-stationary state with constant $\omega \neq 0$ and constant power P_{ij}, P^M (frequency synchronized solution). To ensure that the DEMPC converges to the optimal steady state, we will discuss modifications to the cost and/or constraints, which correspond to assumption 2.

5.3 Classical Control Structure - Power Systems

This section introduces the classical control structure used for distributed power systems. In particular, classical automatic generation control (AGC) and the more recent economic automatic generation control (EAGC) are presented.

5.3.1 Automatic Generation Control

The classical control of power systems is achieved with AGC and a slow centralized steady state economic dispatch. The centralized economic dispatch solves the economic steady state optimization problem and assigns each subsystem a desired local power P_i^M. The area control error (ACE) based AGC is then used to stabilize this set point. This is done by computing the error

$$ACE_i = B_i\omega_i + \sum_{j:i\to j} P_{ij} - \sum_{i:j\to i} P_{ji},$$

and adjusting the power change command with an integral control

$$\dot{P}_j^C = -K_j ACE_j,$$

that controls the turbine-governor control. This controller always compensates the disturbances locally and cannot facilitate a more economic sharing of the load among the subsystems. A more thorough investigation of AGC can be found in [KumarKothariothers05].

5.3.2 Economic Automatic Generation Control

In [ZhangLiPapachristodoulou15] EAGC is proposed, a AGC like controller, that modifies the power flow to enable a more economic operation and load sharing. In particular, this controller is able adapt to changing disturbances and track the new optimal steady state, without requiring the centralized economic dispatch to compute the new optimal steady state. This way changing disturbances can be incorporated a lot faster, which is increasingly relevant due to emerging renewable energies and thus more dynamics in the disturbances.

Basic Idea

Without going into specifics, the basic idea of the EAGC lies in reverse engineering of the AGC and then re-engineering it to get economic AGC. This relies on the saddle point algorithm to solve the steady state optimization problem. The AGC can be reverse engineered to be equivalent to the saddle point dynamics of the steady state optimization problem with a local disturbance matching, i.e. $P_i^M = P_i^L$. This steady state optimization problem is then re-engineered by relaxing the disturbance matching, to allow an exchange in the power flow and imposing the hyperbox constraints on P^M, P_{ij}. By expressing the controller as a saddle point dynamic for this optimization problem we get the EAGC, which has a similar structure to the AGC, but incorporates the economic cost and has additional variables for the branch flow and the constraints.

Properties

Let us discuss some properties of the EAGC. The derivations all use the continuous time linearized system dynamics and the resulting controller is a continuous controller, similar to the AGC. The controller does not need any online optimization, consists of simple addition and multiplication and can be implemented in a distributed fashion. This controller steers the system to the economical optimal steady state, that satisfies the constraints. During transient operation no guarantees with respect to performance or constraint satisfaction can be made.

5.4 Distributed Economic Model Predictive Control

This section bridges the gap between the theoretical results for DEMPC on the one side and the distributed economic dispatch problem on the other side. First we show that the theoretical results in chapter 2 are applicable to the distributed power setup. Then we consider the previously ignored problem of output-feedback DEMPC and show how to compute an augmented distributed observer. Finally we discuss the applicability of different DEMPC approaches and argue for a specific one, which will be used for the comparison in section 5.5.

5.4.1 Assumptions

Here we show how the assumptions used for the theoretical results in chapter 2 can be satisfied for the power systems setup. As already pointed out in section 5.2, the system dynamics and the stage cost fit the setup used in chapter 2.

Optimal Operation at steady state

Strict dissipativity, assumption 2/16, is the most crucial assumption for DEMPC. In [MüllerGrüneAllgöwer15] it was shown, that this is satisfied, if the system is uniformly sub-optimally operated off steady state. As outlined in section 5.2 the pure economically optimal behavior would result in a constant $\omega \neq 0$. By adding a small penalty on δ_i^2 the optimal behavior must lead to a finite δ and thus $av[\omega] = 0$, which is desired. A more straight forward approach to this problem is to directly use an average constraint $av[\omega_i] = 0$ in the DEMPC, which ensures asymptotic stability with assumption 21, see [MüllerEtAl14]. An alternative way to avoid a constant offset in ω is to impose hyper box constraints on δ_i, which implicitly imposes an average constraint $av[\omega_i]$. This makes the analysis of recursive feasibility simpler, but has other disadvantages.

DEMPC Assumptions

For completeness, we also cover the other assumptions and show, how they can be satisfied for this setup. Assumption 1 requires compactness of the constraint set $\mathcal{X} \times \mathcal{U}$. To satisfy this assumption additional hyper box constraints can be imposed on δ_i and ω_i.
Assumption 2 and 3 relate to the storage function $\lambda(x)$. For linear systems with a strictly convex quadratic stage costs and convex constraints there exists a linear storage function $\lambda(x) = \lambda^\top x$ that satisfies the strict dissipativity, [DiehlAmritRawlings11]. This can be guaranteed by adding a small penalty on all state and input variables.
Assumption 4 relating to the terminal set $\mathcal{X}_f$ and the terminal cost V_f, can be satisfied by using the design procedures lemma 10 and (2.14). Even though we have a system that is controllable by distributed controllers, we can theoretically not guarantee that the LMI conditions in lemma 10 have a solution, due to the imposed structure. For the considered numerical examples it was always possible to solve the corresponding SDPs. This issue has been investigated in more detail in [HorssenLazarWeiland14].
For the DEMPC without terminal constraints we require some additional assumptions. Assumption 17 is equivalent to a local controllability condition. Assuming $\mathcal{B}_\epsilon \in \mathcal{X}$ and $\mathcal{B}_{\epsilon_2} \in \mathcal{U}$, this condition is satisfied due to the controllability of power systems.
Assumption 18 requires a finite time controllability into $\mathcal{B}_\epsilon$. This assumption can be satisfied by replacing the state constraints $\mathcal{X}$ by the feasible set of the MPC with terminal constraints. Or in other words, this assumption is satisfied for the feasible set and thus we can guarantee the stability property for all x in the feasible set instead of the whole state constraint set.

5.4.2 Distributed Observers

In MPC we usually assume, that the full state x is measurable. For distributed power systems we can only measure the local frequency deviation ω_i and the tie-line power P_{ij}. The local mechanical power P_i^M is not measured, but is required for the predictions in the MPC. In comparison, the AGC/EAGC does not require a measurement of the mechanical power P_i^M. Therefore we augment the DEMPC with a state observer, and use the estimated state to make the predictions. In general, the issue of operating MPC with limited observations is refereed to as output feedback MPC. Deriving theoretical guarantees for this can be very difficult, since it usually requires bounds on the observer error, and is beyond the scope of this investigation. Instead we just show how such a distributed observer can be computed and assume some upper bound on the transient estimation error, that can be used for the constraint tightening with the robust DMPC.

Distributed Lunenberg Observer

To estimate the state $\hat{x}_i$, a distributed Lunenberg observer is used. Denote the local measurements by $y_i \in \mathbb{R}^{m_i}$, then the structure of the observer is given by

$$\hat{x}_i^+ = A_{\mathcal{N}_i}\hat{x}_{\mathcal{N}_i} + B_i u_i + L_i \underbrace{(y_i - C_{\mathcal{N}_i}\hat{x}_{\mathcal{N}_i})}_{\tilde{y}_i}.$$

The main challenge here is to compute a distributed observer gain $L_i \in \mathbb{R}^{n_i \times m_i}$, that asymptotically stabilizes the observer dynamics. This task is dual to the computation of the terminal controller/cost in section 2.2. For the estimation error $e_i = \hat{x}_i - x_i$ we get the system dynamics

$$e_i^+ = (A_{\mathcal{N}_i} - L_i C_{\mathcal{N}_i})e_{\mathcal{N}_i}.$$

Computing a distributed observer gain L for a system $(A; C)$ with the system dynamics matrix A and the output matrix C, is equivalent to computing a distributed controller for $(A; B) = (A^\top; C^\top)$, with the input matrix B. This means that lemma 10 can be used to compute a distributed observer gain L_i with distributed LMI computations. Similar to a LQR or Kalman filter design, the matrices Q and R are weighting matrices, that can be used to shape the observer.

Tie-line Power based Observer

Computing a distributed observer based on the controller model, can pose some difficulties due to the non-observable marginally stable mode ($\delta_i = \delta_j$) and offsets in the nonlinear tie-line power P_{ij}. To avoid these issues we make a transformation to the local variables $[P_i^M, \omega_i, P_{ij}]$ and use the corresponding model for the estimation. This linear model is fully observable and we can compute a distributed linear observer gain L_i, that asymptotically stabilizes the observer dynamics. An additional advantage of this observer model, is that the nonlinear tie-line power P_{ij} is directly estimated, by only using the linear model.
The computation of this local observer relies again on the controller-observer duality and lemma 10. Since the state variables P_{ij} are shared, the structure of the terminal cost/Lyapunov function $V = x^\top P x$, needs to be further specified. We use the ansatz

$$V = x^\top P x = \sum_{i=1}^{M} x_i^\top P_i x_i + \sum_{(i,j)\in E} p_{ij} P_{ij}^2,$$

with the scalar variables $p_{ij} \in \mathbb{R}$ and the local cost $P_i \in \mathbb{R}^2$ only considering the purely local states (P_i^M,ω_i). This structure ensures that the transformation in lemma 10 ($Y = KP$), keeps the distributed structure of the local observer gain L_i. With this we can compute a distributed observer gain L_i, which asymptotically stabilizes the observer dynamics and is tunable with the weighting matrices Q, R.

Disturbance Observer

In addition to the local state measurement the MPC uses the predicted disturbance trajectory $d_i(t+k)$ in the optimization. The disturbance changes in general unpredictably, which means that we cannot presume the correctness of the predicted disturbance trajectory. Instead the MPC simply assumes, that the disturbance trajectory remains constant. A small smooth change in the disturbance can then be compensated by a small constraint tightening. This however still requires the knowledge of the current disturbance $d_i(t)$.
For large sudden load changes (step), we can assume some external knowledge, but small random changes need to be estimated. We use an augmented distributed state observer, that estimates both the current state $\hat{x}_i$ and disturbance $\hat{d}_i$. This is done by computing a distributed observer, for the

augmented state $\tilde{x}_i = [x_i, d_i] \in \mathbb{R}^3$. The augmented system dynamics are given by

$$\underbrace{\begin{pmatrix} x^+ \\ d^+ \end{pmatrix}}_{\tilde{x}^+} = \underbrace{\begin{pmatrix} A & E \\ 0 & I \end{pmatrix}}_{\tilde{A}} \underbrace{\begin{pmatrix} x \\ d \end{pmatrix}}_{\tilde{x}} + \underbrace{\begin{pmatrix} B \\ 0 \end{pmatrix}}_{\tilde{B}} u,$$

$$y = \underbrace{\begin{pmatrix} C & 0 \end{pmatrix}}_{\tilde{C}} \underbrace{\begin{pmatrix} x \\ d \end{pmatrix}}_{\tilde{x}}.$$

By using the previous method to compute a distributed observer for this augmented system, we get an observer, that estimates both the local state x_i and disturbance d_i.

5.4.3 Discussion

We have established, that the theoretical results can all be used for power systems. Now we discuss some of the advantages and disadvantages of different DEMPC methods, as they relate to the power system setup.

DEMPC without terminal constraints with average constraint ω

The usage of terminal costs and sets has the advantage of easily guaranteeing recursive feasibility and asymptotic stability. The main drawback is the need to recompute/adapt the optimal steady state and the terminal set size, due to the changing disturbances. Additional difficulties result from the ellipsoid constraint, which increases computational complexity for the online optimization. In contrast DEMPC without terminal constraints requires no offline computations and fluctuating disturbances do not pose a problem. Therefore we only consider DEMPC without terminal constraints in the comparison. For the DEMPC without terminal constraints we use average constraints $av[\omega_i] = 0$, to ensure convergence to steady state. This is mainly because the alternative options need to consider δ_i, a quantity which cannot be measured and has no real physical meaning.

Robust DMPC

For the robust DMPC modification we use the additive disturbance w_i to model different effects. This includes in particular the model mismatch and

the additional error caused by inexact dual optimization (see section 4.2). Furthermore, unpredictable fluctuations in the power load P_i^L can also be addressed this way. The state-estimation error can also be included here. If we want to guarantee constraint satisfaction despite all these inconveniences, a robust constraint tightening is inevitable.
For the robust modification we have two options: the growing tubes approach and the RPI tubes. For the RPI tubes, the RPI set needs to be computed explicitly and the online computational demand is increased due to the quadratic RPI constraint on the initial state. On the other side the growing tubes approach only requires a stabilizing controller and leads to a smaller constraint tightening along the prediction horizon. For power systems the economical optimal operation point lies often at the boundary of the constraint set. Therefore we use the growing tubes robust MPC approach for power systems.

Realistic Scenario

It is important to clarify, that here we go beyond the previous established results, to apply the inexact DEMPC to a realistic scenario. In particular we have not established recursive feasibility guarantees for DEMPC without terminal constraints under this setup (average constraints, inexact minimization, constraint tightening, output feedback). Neithertheless, we have recursive feasibility for this particular example and correspondingly we have constraint satisfaction, stability and and superior economic performance.

5.5 Simulation Experiment - Test Case based Comparison

In this section, the performance of the DEMPC for the real time economic dispatch for distributed power systems is studied in two scenarios and compared to classical controllers and the best possible performance. First the setup is introduced and the corresponding model parameters and controller configurations are described. Then the first scenario with a large disturbance change is considered and the resulting performance for operation along the constraints is compared. In the second scenario randomly changing disturbances are considered and the performance under constant fluctuations are analysed. Finally the qualitative differences between DEMPC and EAGC are summarized.

5.5.1 Setup - Parameters and Controller Configuration

Here the setup for the comparison is introduced and the corresponding model parameters and controller configurations are detailed. We consider a 4-area interconnected power system, which is represented in figure 5.1. For the comparison AGC, EAGC, DEMPC and an optimal controller DEMPC* (to be specified later) are used. The corresponding model parameters are shown

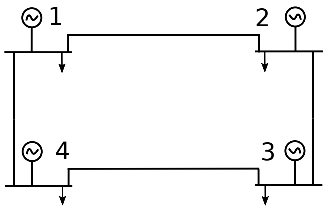

Figure 5.1: A 4-area interconnected power system.

in Table 5.1 and 5.2 and correspond to the model used in [LiEtAl14]. For the economic cost we use $l_i = \left(P_i^M\right)^2$, which means that the economic optimal behavior is to distribute the load as equally as possible. For the constraints we have $P_M^{\max} = -P_M^{\min} = 1$, and $P_{ij}^{\max} = -P_{ij}^{\min} = 0.5$. The EAGC is based on a continous time model, linearized around the origin, and the control parameters are $\gamma = 2$, $K_c = 15$, $K_\theta = K_\xi = K_\lambda = K_\mu = K_\nu = K_{l_{ij}} = 10$. The AGC is simulated with $K_i = 2$ and $B_i = 2D_i$.

DEMPC-ω

We consider DEMPC without terminal constraints, with the growing robust tube approach and with an average constraint on ω_i, i.e. DEMPC-ω. The online computation is carried out with ADMM, and the online stopping condition is based on the tolerance $\epsilon = 10^{-5}$. To get a behavior similar to EAGC we use the local stage cost $l_i(x_i,u_i) = (P_i^M)^2 + 100\omega_i^2$, that also penalizes frequency deviations and thus stabilizes the frequency dynamics. The model of the DEMPC uses a step size of $h = 0.1s$ and a second order discretization. For the constraint tightening the disturbance

$$\mathcal{W}_i = \{w_i | |w_{P_M}| \leq 10^{-3}, |w_\omega| \leq 3 \cdot 10^{-4}, |w_\delta| \leq 3 \cdot 10^{-4}\},$$

Table 5.1: Generator Parameters

Area i	M_i	D_i	$\lvert V_i \rvert$	T_i	R_i
1	3	1	1.045	4	0.05
2	2.5	1.5	0.98	4	0.05
3	4	1.2	1.033	4	0.05
4	3.5	1.4	0.997	4	0.05

Table 5.2: Line Parameters

line	1-2	2-3	3-4	4-1
r	0.004	0.005	0.006	0.0028
x	0.0386	0.0294	0.0596	0.0474

is considered. The corresponding stabilizing controller K_t is computed based on lemma 24 with $\lambda = 0.9$. An artificial input constraint $|P^C| \leq 10$ and the tightening ρ is included in the cost function, in order to avoid high gain controllers.
The terminal cost P_f and terminal controller K are computed based on lemma 10. To avoid high terminal costs and strong coupling both the terminal cost P_f and the coupling Γ are minimized.
A distributed tie-line based observer and an augmented version are computed based on the description in section 5.4.2 with the weights $R_i = 10^2$, $Q_i = [10^2, 10^{-2}, 10^2]$ and $p_{ij} = 10^2$. All the corresponding LMI computations are carried out with the MPT-3 toolbox [HercegEtAl13], Yalmip [Lofberg04] and MOSEK [Mosek10].
Due to the absence of any cost or constraints on the control input P^C, we do not need to know the optimal steady state x^* to implement the stabilizing controller $u = K_t(x - x^*) + v$.

Best possible Performance - DEMPC*-ω

In addition to the comparison between the different controllers, we compare the performance to the best theoretically achievable performance. In section 4.4, we have seen that the difference between the DEMPC and the best possible performance is bounded by a factor depending on the prediction

horizon N and the suboptimality η. An additional deterioration in the performance arises due to the model mismatch (discretization + linearization), the unpredictable change in the load (d_i) and the consequently tightened operational space (constraint tightening).
To get an estimate for the loss of performance due to these factors, we consider an idealized controller, denoted by DEMPC*-ω. For this we assume, that the disturbance trajectory d_i in the future is known, that there is no model mismatch (system model = linearized discretized controller model) and that the underlying optimization problem is solved exactly. This controller has the same constraint on the average frequency ω as DEMPC-ω. This is basically an optimal open loop control. It is important to note, that such a performance can never be attained in practice, but it is merely a lower bound.

5.5.2 Scenario 1 - Operation along the Constraints

In the first scenario the disturbance d_4 increases with a 1 second ramp from $d_4 = 0$ to $d_4 = 1.4$, which pushes the system to the constraints. The resulting behavior of the classical AGC, the Economic AGC, DEMPC with average constraints for ω_i and the optimal DEMPC* are shown in figures 5.2- 5.4. In this scenario we assume, that the DEMPC knows the disturbance trajectory a priori, which is a reasonable assumption for large deterministic changes.

Here we can see, that the EAGC balances the load among the subsystems and converges to a steady state that respects the constraints. The AGC compensates the load locally and does not take any constraints into account. We can also see, that the AGC converges slowly, with stronger oscillations. The DEMPC is a lot faster in stabilizing the dynamics and is also able to respect the tie-line constraints during transients, while the EAGC violates the tie-line constraints. It is important to point out that the DEMPC only converges close to the optimal steady state due to the constraint tightening. Another big difference between MPC and AGC can be seen at the frequency deviation. The AGC stabilizes the frequency deviation like a linear controller with lage oscillations. The DEMPC quickly stabilizes all subsystems to the same, non-zero frequency (frequency synchronization). Then after some time, the frequency transitions smoothly to the nominal value. This reflects the fact, that purely minimizing the economic cost, will result in an optimal behavior with non-zero frequency. Then after some time (tuneable), the constraint on the average frequency $av[\omega] = 0$ becomes active, and thus the frequency is naturally driven back to the nominal value.

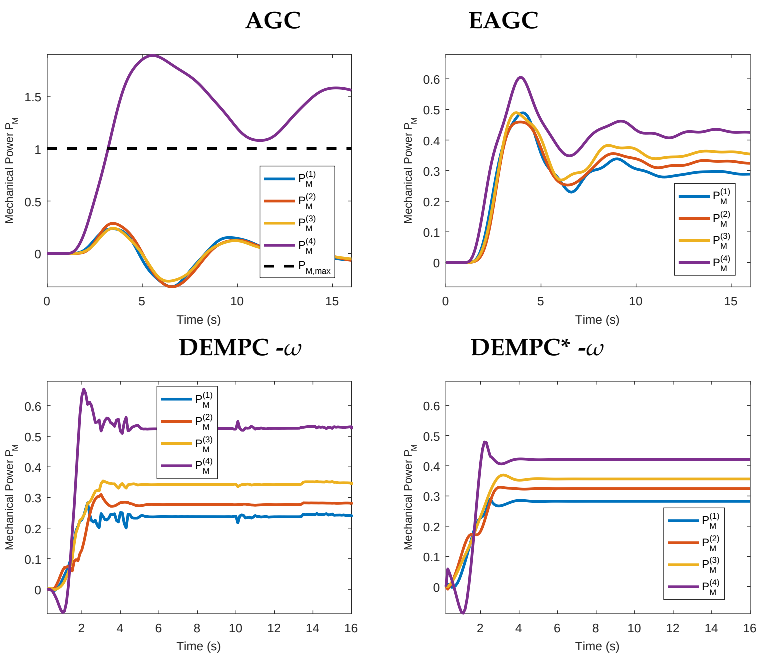

Figure 5.2: Power P^M - Scenario 1.

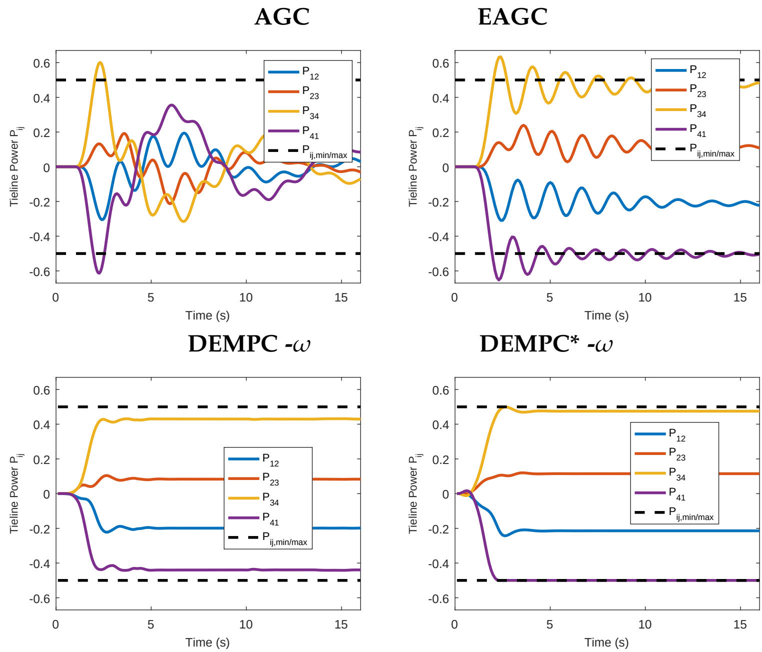

Figure 5.3: Tie-line Power P_{ij} - Scenario 1.

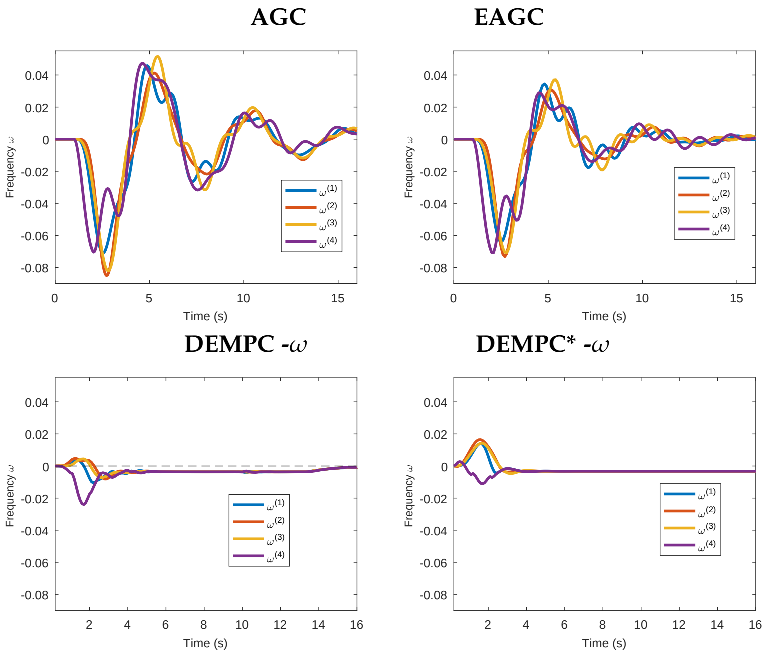

Figure 5.4: Frequency Deviation ω - Scenario 1.

The 'optimal' behavior, DEMPC*, is relatively similar to the DEMPC. One of the major difference is the steady state economic cost, due to the constraint tightening. The other big difference is in the frequency deviation ω. To achieve better performance, a negative frequency ω is stabilized in the quasi-stationary operation. To still satisfy the average constraint on ω, ω is initially artificially increased, where it only results in a small additional cost.
The corresponding differences in performance are summarized in table 5.3.

Table 5.3: Performance

Method	av[P_M^2]	av[$\lvert\omega\rvert$]	$\lVert\omega\rVert_\infty$	av[ω^2]	$P_M^2(end)$	con
AGC	1.7170	1.5e-2	8.5e-2	5.28e-4	2.4346	0.2156
EAGC	0.4622	1.1e-2	7.3e-2	3.84e-4	0.4945	0.0018
DEMPC-ω	0.4776	1.9e-3	2.4e-2	1.00e-5	0.5401	0
DEMPC*-ω	0.4359	3.7e-3	1.6e-2	1.90e-5	0.4886	-

Here av[P_M^2] is the average economic cost, av[$\lvert\omega\rvert$] the absolute average frequency offset, $\lvert\omega\rvert_\infty$ is the peak value of the frequency and av[ω^2] is the average quadratic frequency. $P_M^2(end)$ is the economic cost at the steady state, that is reached at the end of this simulation, and *con* measures how much the constraints have been violated. The difference in the economic stage cost over time can be seen in figure 5.5.

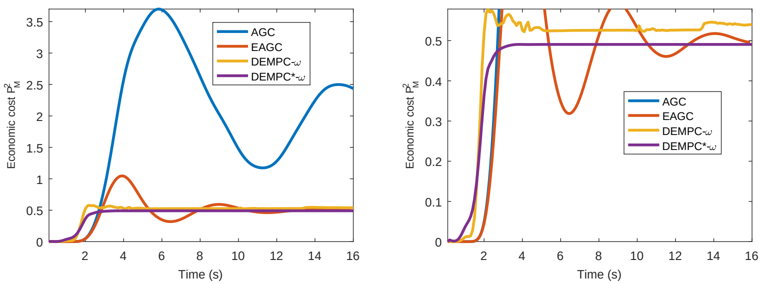

Figure 5.5: Economic Performance -Scenario 1

On different notations of Constraints

There is a general difference in the treatment of constraints between MPC and AGC, that should be highlighted. For (E)AGC the constraints are defined for steady state operation and violations over longer periods of time are expected. In contrast MPC always satisfies constraints at each point in time.
In the given scenario this property in combination with the constraint tightening prevents the MPC from reaching the optimal steady state. It would however be false to see this as a limitation of MPC. Instead it is a feature, that allows the operator to impose hard constraints in addition to steady state constraints. The steady state constraints can be incorporated in the MPC as average or transient constraints. This way the MPC has a larger domain of operation and can achieve even better results. More importantly this enables the operator to directly specify hard constraints and also to give hard bounds on how long/how much the steady state constraints are allowed to be violated.

5.5.3 Scenario 2 - Random load changes

In the second scenario we consider constantly fluctuating disturbances to better reflect the permanent changes and put more focus on the transient performance. Here the local loads d_i change randomly with

$$d_i(k+1) = d_i(k) + \text{unif}(-\Delta d, \Delta d),$$

with the uniform distribution unif and the maximum change $\Delta d = 0.01$. In this scenario the constraints are not relevant, and the focus is on the transient economic performance due to this changing disturbance. By using a random change in the disturbance, instead of just a random disturbance, we get trends in the disturbance. This means that just stabilizing the original steady state is not optimal. Furthermore, this enables large disturbances, while having a smooth change. Here the disturbance is unknown to the DEMPC and online estimated with the augmented state observer. The resulting random walk of the disturbance and the corresponding estimated disturbance can be seen in figure 5.6.
The resulting trajectories for the different controllers are shown in figures 5.7-5.9.

If we compare the EAGC with AGC, we can see a very similar frequency stabilization, while the economic cost of the EAGC is significantly lower,

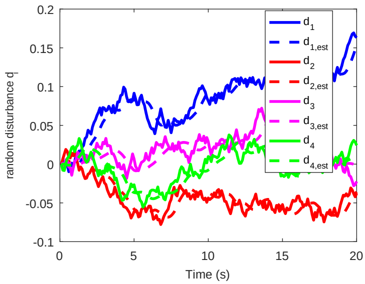

Figure 5.6: Random disturbance d_i - Scenario 2.

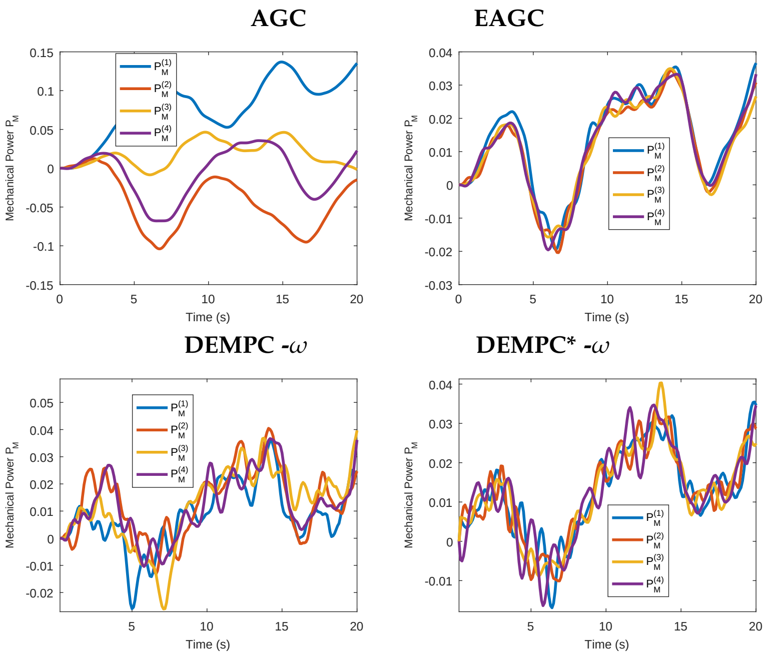

Figure 5.7: Power trajectory P^M - Scenario 2.

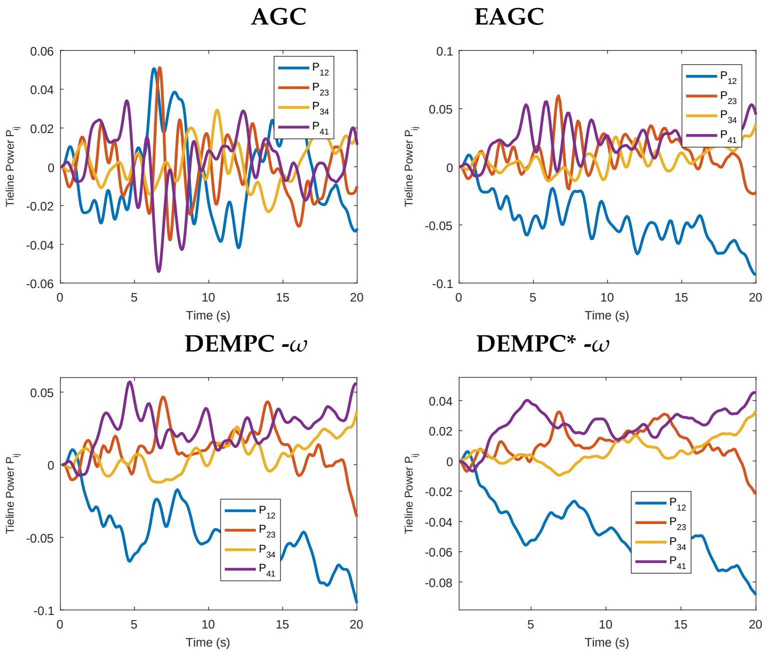

Figure 5.8: Tie-line Power P_{ij} - Scenario 2.

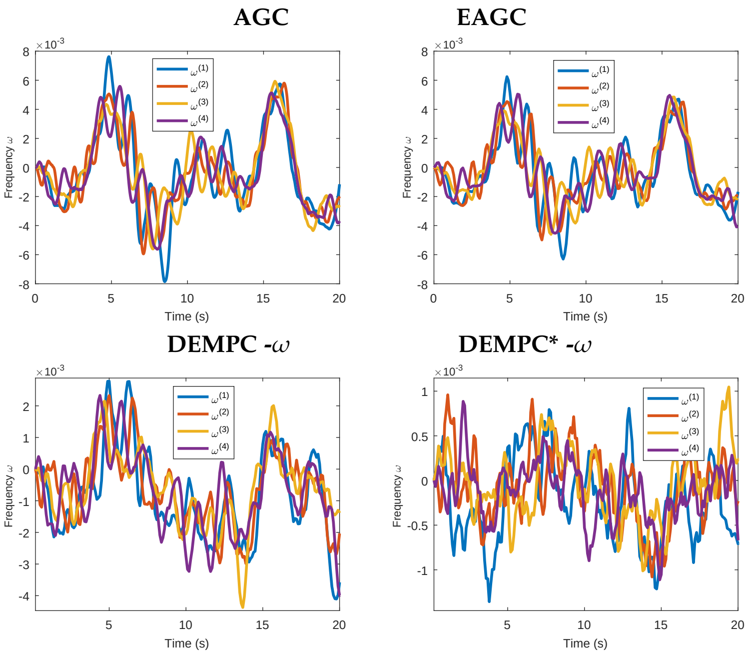

Figure 5.9: Frequency Deviation ω - Scenario 2.

due to the power exchange. Even though, there are no theoretical results available for the tracking performance of EAGC, we can see, that the load is shared almost equally over the full time span, which corresponds to the optimal steady state behavior.
If we look at the DEMPC, the frequency ω is kept significantly smaller, while the tie-line power P_{ij} has a similar trend to the EAGC. Also the power P^M follows a similar pattern to EAGC, but is less smooth.
For the optimal behavior DEMPC*, we can see little difference to the DEMPC. The quantitative performance is summarized in table 5.4.
The difference in the economic stage cost can be seen in figure 5.10.

Table 5.4: Performance

Method	av$[P_M^2]$	av$[\omega]$	av$[\|\omega\|]$	$\|\omega\|_\infty$	av$[\omega^2]$
AGC	1.242e-2	-4.3e-4	2.2e-3	7.8e-3	7.26e-6
EAGC	1.362e-3	-3.0e-4	1.8e-3	6.3e-3	5.06e-6
DEMPC-ω	1.144e-3	-4.6e-4	7.6e-4	4.4e-3	1.37e-6
DEMPC*-ω	1.139e-3	-1.1e-4	3.3e-4	1.4e-3	1.80e-7

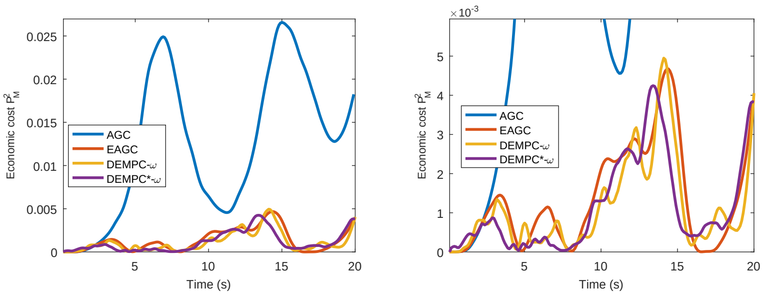

Figure 5.10: Economic Performance - Scenario 2

Numerical Effects

While numerical studies are not the focus of this work, some results regarding the convergence of the ADMM algorithm are presented. Therefore, we considered different tolerances ϵ and different penalty factors ρ and study the effect on the performance and the number of required iterations for this scenario. In table 5.5 the corresponding numerical results are summarized. For comparison, two simulation runs with a fixed number of iterations are included, for which no theoretical guarantees can be obtained.

Table 5.5: Numerical convergence results

ϵ	ρ	av[P_M^2]	av[ω^2]	av[iter]	max[iter]
10^{-5}	100	1.043e-3	1.76e-6	228	1042
10^{-4}	100	1.036e-3	1.81e-6	1.12	4
10^{-4}	10	1.066e-3	1.63e-6	92.5	470
10^{-5}	10	1.134e-3	1.40e-6	556	876
-	10	1.038e-3	1.79e-6	10	10
-	10	1.059e-3	1.65e-6	100	100

Here we can see that a well-tuned ADMM is able to give medium accuracy with few iterations. But we can also see, that the number of required iterations can grow large, if we have higher accuracy demands. Furthermore we can see that (at least in this scenario) the impact of the accuracy on the suboptimality is relatively small. In the first scenario the connection is more complicated since the medium accuracy ϵ also leads to a larger constraint tightening and thus worse performance. The number of iterations could be further reduced by scaling the optimization problem and selecting the parameters optimally [GiselssonBoyd14,GhadimiEtAl15].

5.5.4 Qualitative Comparison

As an addition to the quantitative performance comparison in the two scenarios we discuss some qualitative differences. Out of the four considered controllers we only compare DEMPC and EAGC. AGC seems by every measure worse then EAGC, and while DEMPC* is superior, it is not practically implementable.

The first obvious difference lies in the complexity and online computational demand associated with DEMPC in comparison to the 'simple' implementation of EAGC. For real time application of DEMPC, multiple distributed optimization iterations need to be carried out in each sampling time step, while EAGC only requires simple multiplications/additions. The most important advantage of MPC in general, is the satisfaction of state constraints during transient dynamics. In comparison, EAGC guarantees that the system converges to a steady state, that satisfies the state constraints, with no guarantees for the transient constraint violation.
For EAGC we know that the closed-loop system is asymptotically stable. For DEMPC with suboptimality, especially without terminal conditions, we can only guarantee practical asymptotic stability (see section 4.3).
A key advantage of DEMPC are the transient performance guarantees despite suboptimality derived in section 4.4, which are not available for EAGC. In the context of fluctuating disturbances the advantages of DEMPC become more prevalent. The distinction between asymptotic stability and practical stability becomes meaningless, while the advantage of transient performance and constraint satisfactions gains significance. On the other side the EAGC is mainly based on a steady state optimization, which gives no guarantees for transient phases, which are dominant under fluctuating disturbances.
Another important issue is the extension to nonlinear dynamics. By its very nature EAGC is based on duality and thus linear dynamics. For DEMPC we can use a successive linearized model, which seems to represent the nonlinear model well enough. For DEMPC without terminal constraints, all theoretical results are general enough to apply to nonlinear dynamics. The challenge lies mainly in the online distributed optimization, which requires the solution of a local nonlinear programming (NLP) in each iteration step. (An inexact ADMM method, ALADIN, could be used for this [HouskaFraschDiehl16].)
In summary, DEMPC is more complex than EAGC, but offers benefits for transient operation. Therefore, the more fluctuations and dynamics are present in the system, the higher is the potential improvement in performance, stability and constraint satisfaction by using DEMPC, a dynamic optimization method, instead of EAGC, a primarily stationary optimization method.

6 Summary and Outlook

This chapter summarizes the main results of this thesis in comparison to existing results. Also further research topics building on this work are proposed.

6.1 Conclusions

In this work, theoretical guarantees for DEMPC under inexact minimization were derived. We also showed, that all the required offline computations can be cast as distributed optimization problems, which facilitates an automatic design procedure. This is mainly an extension of previous results in [ConteEtAl16] and [ConteEtAl13].
Then assuming a certain accuracy (ϵ,η) of a distributed optimization algorithm, the original MPC optimization problem can be modified such that we have guarantees for constraint satisfaction and recursive feasibility. There exist several distributed algorithms, in particular dual fast gradient methods [FerrantiEtAl15,NecoaraNedelcu14,NecoaraSuykens08], that can compute such a solution with a finite number of iterations. Compared to existing results, we do not require online adaptation of the constraint tightening as in [FerrantiEtAl15,GiselssonRantzer14,DoanKeviczkyDe Schutter11], and the constant tightening is significantly less conservative compared to [KögelFindeisen14]. This is possible by considering an inexact trajectory for the candidate solution.
Stability and performance guarantees with suboptimality η in the optimization have been derived. This can be seen as an extension of the nominal results derived in [Grüne13,GrüneStieler14,GrünePanin15].
The other contribution of this thesis is the application of DEMPC to solve the real time economic dispatch problem. Here we demonstrated that DEMPC can be used for realistic power system scenarios. We also compared DEMPC to EAGC [ZhangLiPapachristodoulou15] and showed superior economic performance, faster frequency damping and constraint satisfaction.

6.2 Future Work

Although this work already contains a framework for DEMPC with inexact minimization, there are several promising directions for further research. The recursive feasibility result could be put together with a dual fast gradient algorithm like [FerrantiEtAl15]. By choosing the constraint tightening properly it should be possible to derive a general a priori upper bound on the number of required iterations.

Deriving a similar recursive feasibility results under constraint tightening for MPC without terminal constraints, would also be a valuable contribution. Here a RPI tightening instead of the growing tubes approach might be more adequate. Showing that the simulation results in thesis also hold for more complex power systems is another future endeavor. Therefore a more comprehensive simulation study, considering for example the 39 IEEE bus system used in [ZhaoEtAl16] would be worthwhile.

Eigenständigkeitserklärung

Ich versichere hiermit, dass ich, Johannes Köhler, die vorliegende Arbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und sowohl wörtliche, als auch sinngemäß entlehnte Stellen als solche kenntlich gemacht habe. Die Arbeit hat in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen. Weiterhin bestätige ich, dass das elektronische Exemplar mit den anderen Exemplaren übereinstimmt.

Ort, Datum

Unterschrift

Literaturverzeichnis

[AlvaradoEtAl10] Alvarado, I.; Limon, D.; de la Pena, D.M.; Alamo, T.; Camacho, E.: Enhanced ISS nominal MPC based on constraint tightening for constrained linear systems. In Control 2010, UKACC International Conference on, S. 1–6, IET, 2010.

[AmritRawlingsAngeli11] Amrit, R.; Rawlings, J.B.; Angeli, D.: Economic optimization using model predictive control with a terminal cost. Annual Reviews in Control, Bd. 35, Nr. 2, S. 178–186, 2011.

[AngeliAmritRawlings12] Angeli, D.; Amrit, R.; Rawlings, J.B.: On average performance and stability of economic model predictive control. IEEE transactions on automatic control, Bd. 57, Nr. 7, S. 1615–1626, 2012.

[BayerEtAl16] Bayer, F.A.; Lorenzen, M.; Müller, M.A.; Allgöwer, F.: Robust economic Model Predictive Control using stochastic information. Automatica, Bd. 74, S. 151–161, 2016.

[BocciaGrüneWorthmann14] Boccia, A.; Grüne, L.; Worthmann, K.: Stability and feasibility of state constrained MPC without stabilizing terminal constraints. Systems & Control Letters, Bd. 72, S. 14–21, 2014.

[Boyd07] Boyd, S.: Primal and dual decomposition. Lecture Notes, EE364b Convex Optimization, Stanford University, http://www. stanford. edu/class/ee364b/lectures. html, 2007.

[BoydEtAl94] Boyd, S.P.; El Ghaoui, L.; Feron, E.; Balakrishnan, V.: Linear matrix inequalities in system and control theory, Bd. 15. SIAM, 1994.

[BoydEtAl11] Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine Learning, Bd. 3, Nr. 1, S. 1–122, 2011.

[BoydVandenberghe04] Boyd, S.; Vandenberghe, L.: Convex optimization. Cambridge university press, 2004.

[ChenAllgöwer97] Chen, H.; Allgöwer, F.: A quasi-infinite horizon nonlinear model predictive control scheme with guaranteed stability. In Control Conference (ECC), 1997 European, S. 1421–1426, IEEE, 1997.

[ChisciRossiterZappa01] Chisci, L.; Rossiter, J.A.; Zappa, G.: Systems with persistent disturbances: predictive control with restricted constraints. Automatica, Bd. 37, Nr. 7, S. 1019–1028, 2001.

[ConteEtAl12a] Conte, C.; Summers, T.; Zeilinger, M.N.; Morari, M.; Jones, C.N.: Computational aspects of distributed optimization in model predictive control. In 2012 IEEE 51st IEEE Conference on Decision and Control (CDC), S. 6819–6824, IEEE, 2012.

[ConteEtAl12b] Conte, C.; Voellmy, N.R.; Zeilinger, M.N.; Morari, M.; Jones, C.N.: Distributed synthesis and control of constrained linear systems. In 2012 American Control Conference (ACC), S. 6017–6022, IEEE, 2012.

[ConteEtAl12c] Conte, C.; Voellmy, N.R.; Zeilinger, M.N.; Morari, M.; Jones, C.N.: Distributed synthesis and control of constrained linear systems. In 2012 American Control Conference (ACC), S. 6017–6022, IEEE, 2012.

[ConteEtAl13] Conte, C.; Zeilinger, M.N.; Morari, M.; Jones, C.N.: Robust distributed model predictive control of linear systems. In Proc. of the 12th European Control Conf, S. 2764–2769, 2013.

[ConteEtAl16] Conte, C.; Jones, C.N.; Morari, M.; Zeilinger, M.N.: Distributed synthesis and stability of cooperative distributed model predictive control for linear systems. Automatica, Bd. 69, S. 117–125, 2016.

[DammEtAl14] Damm, T.; Grüne, L.; Stieler, M.; Worthmann, K.: An exponential turnpike theorem for dissipative discrete time optimal control problems. SIAM Journal on Control and Optimization, Bd. 52, Nr. 3, S. 1935–1957, 2014.

[DiehlAmritRawlings11] Diehl, M.; Amrit, R.; Rawlings, J.B.: A Lyapunov function for economic optimizing model predictive control. IEEE Transactions on Automatic Control, Bd. 56, Nr. 3, S. 703–707, 2011.

[DoanKeviczkyDe Schutter11] Doan, M.D.; Keviczky, T.; De Schutter, B.: A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints. In 2011 50th IEEE Conference on Decision and Control and European Control Conference, S. 5236–5241, IEEE, 2011.

[ErsdalEtAl16] Ersdal, A.M.; Imsland, L.; Uhlen, K.; Fabozzi, D.; Thornhill, N.F.: Model predictive load–frequency control taking into account imbalance uncertainty. Control Engineering Practice, Bd. 53, S. 139–150, 2016.

[FagianoTeel12] Fagiano, L.; Teel, A.R.: Model Predictive Control with generalized terminal state constraint. IFAC Proceedings Volumes, Bd. 45, Nr. 17, S. 299–304, 2012.

[FarinaScattolini12] Farina, M.; Scattolini, R.: Distributed predictive control: a non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. Automatica, Bd. 48, Nr. 6, S. 1088–1096, 2012.

[FerrantiEtAl15] Ferranti, L.; et al.: A parallel dual fast gradient method for MPC applications. In 2015 54th IEEE Conference on Decision and Control (CDC), S. 2406–2413, IEEE, 2015.

[GhadimiEtAl15] Ghadimi, E.; Teixeira, A.; Shames, I.; Johansson, M.: Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems. IEEE Transactions on Automatic Control, Bd. 60, Nr. 3, S. 644–658, 2015.

[GiselssonBoyd14] Giselsson, P.; Boyd, S.: Diagonal scaling in Douglas-Rachford splitting and ADMM. In Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on, S. 5033–5039, IEEE, 2014.

[GiselssonRantzer14] Giselsson, P.; Rantzer, A.: On feasibility, stability and performance in distributed model predictive control. IEEE Transactions on Automatic Control, Bd. 59, Nr. 4, S. 1031–1036, 2014.

[Grüne13] Grüne, L.: Economic receding horizon control without terminal constraints. Automatica, Bd. 49, Nr. 3, S. 725–734, 2013.

[GrüneMüller16] Grüne, L.; Müller, M.A.: On the relation between strict dissipativity and turnpike properties. Systems & Control Letters, Bd. 90, S. 45–53, 2016.

[GrünePanin15] Grüne, L.; Panin, A.: On non-averaged performance of economic MPC with terminal conditions. Universität Bayreuth, 2015.

[GrünePannek11] Grüne, L.; Pannek, J.: Nonlinear model predictive control. In Nonlinear Model Predictive Control, S. 43–66. Springer, 2011.

[GrüneStieler14] Grüne, L.; Stieler, M.: Asymptotic stability and transient optimality of economic MPC without terminal conditions. Journal of Process Control, Bd. 24, Nr. 8, S. 1187–1196, 2014.

[HeidarinejadLiuChristofides12] Heidarinejad, M.; Liu, J.; Christofides, P.D.: Economic model predictive control of nonlinear process systems using Lyapunov techniques. AIChE Journal, Bd. 58, Nr. 3, S. 855–870, 2012.

[HercegEtAl13] Herceg, M.; Kvasnica, M.; Jones, C.; Morari, M.: Multi-Parametric Toolbox 3.0. In Proc. of the European Control Conference, S. 502–510, Zürich, Switzerland, 2013. http://control.ee.ethz.ch/~mpt.

[HorssenLazarWeiland14] van Horssen, E.P.; Lazar, M.; Weiland, S.: On Synthesis of Stabilizing Distributed Controllers with an Application to Power Systems. IFAC Proceedings Volumes, Bd. 47, Nr. 3, S. 11917–11925, 2014.

[HouskaFraschDiehl16] Houska, B.; Frasch, J.; Diehl, M.: An Augmented Lagrangian Based Algorithm for Distributed NonConvex Optimization. SIAM Journal on Optimization, Bd. 26, Nr. 2, S. 1101–1127, 2016.

[JokicLazar09] Jokic, A.; Lazar, M.: On decentralized stabilization of discrete-time nonlinear systems. In 2009 American Control Conference, S. 5777–5782, IEEE, 2009.

[KögelFindeisen12] Kögel, M.; Findeisen, R.: Cooperative distributed MPC using the alternating direction multiplier method. IFAC Proceedings Volumes, Bd. 45, Nr. 15, S. 445–450, 2012.

[KögelFindeisen14] Kögel, M.; Findeisen, R.: Stabilization of inexact MPC schemes. In 53rd IEEE Conference on Decision and Control, S. 5922–5928, IEEE, 2014.

[KumarKothariothers05] Kumar, P.; Kothari, D.P.; et al.: Recent philosophies of automatic generation control strategies in power systems. IEEE transactions on power systems, Bd. 20, Nr. 1, S. 346–357, 2005.

[LangsonEtAl04] Langson, W.; Chryssochoos, I.; Raković, S.; Mayne, D.Q.: Robust model predictive control using tubes. Automatica, Bd. 40, Nr. 1, S. 125–133, 2004.

[LiEtAl14] Li, N.; Chen, L.; Zhao, C.; Low, S.H.: Connecting automatic generation control and economic dispatch from an optimization view. In 2014 American Control Conference, S. 735–740, IEEE, 2014.

[LimonEtAl08a] Limon, D.; Alvarado, I.; Alamo, T.; Camacho, E.: On the design of Robust tube-based MPC for tracking. IFAC Proceedings Volumes, Bd. 41, Nr. 2, S. 15333–15338, 2008.

[LimónEtAl08b] Limón, D.; Alvarado, I.; Alamo, T.; Camacho, E.F.: MPC for tracking piecewise constant references for constrained linear systems. Automatica, Bd. 44, Nr. 9, S. 2382–2387, 2008.

[Lofberg04] Lofberg, J.: YALMIP: A toolbox for modeling and optimization in MATLAB. In Computer Aided Control Systems Design, 2004 IEEE International Symposium on, S. 284–289, IEEE, 2004.

[MaestreNegenbornothers14] Maestre, J.M.; Negenborn, R.R.; et al.: Distributed model predictive control made easy, Bd. 69. Springer, 2014.

[MayneEtAl09] Mayne, D.Q.; Raković, S.; Findeisen, R.; Allgöwer, F.: Robust output feedback model predictive control of constrained linear systems: Time varying case. Automatica, Bd. 45, Nr. 9, S. 2082–2087, 2009.

[MayneSeronRaković05] Mayne, D.Q.; Seron, M.M.; Raković, S.: Robust model predictive control of constrained linear systems with bounded disturbances. Automatica, Bd. 41, Nr. 2, S. 219–224, 2005.

[Mosek10] Mosek, A.: The MOSEK optimization software. Online at http://www. mosek. com, Bd. 54, S. 2–1, 2010.

[MüllerAllgöwer17] Müller, M.A.; Allgöwer, F.: Economic and Distributed Model Predictive Control: Recent Developments in Optimization-Based Control. SICE Journal of Control, Measurement and Systems Integration, Bd. 10, Nr. 2, 2017.

[MüllerAngeliAllgöwer13] Müller, M.A.; Angeli, D.; Allgöwer, F.: On convergence of averagely constrained economic MPC and necessity of dissipativity for optimal steady-state operation. In 2013 American Control Conference, S. 3141–3146, IEEE, 2013.

[MüllerAngeliAllgöwer14] Müller, M.A.; Angeli, D.; Allgöwer, F.: On the performance of economic model predictive control with self-tuning terminal cost. Journal of Process Control, Bd. 24, Nr. 8, S. 1179–1186, 2014.

[MüllerEtAl14] Müller, M.A.; Angeli, D.; Allgöwer, F.; Amrit, R.; Rawlings, J.B.: Convergence in economic model predictive control with average constraints. Automatica, Bd. 50, Nr. 12, S. 3100–3111, 2014.

[MüllerGrüneAllgöwer15] Müller, M.A.; Grüne, L.; Allgöwer, F.: On the role of dissipativity in economic model predictive control. IFAC-PapersOnLine, Bd. 48, Nr. 23, S. 110–116, 2015.

[NecoaraDoanSuykens08] Necoara, I.; Doan, D.; Suykens, J.A.: Application of the proximal center decomposition method to distributed model predictive control. In Decision and Control, 2008. CDC 2008. 47th IEEE Conference on, S. 2900–2905, IEEE, 2008.

[NecoaraEtAl09] Necoara, I.; Savorgnan, C.; Tran, D.Q.; Suykens, J.; Diehl, M.: Distributed nonlinear optimal control using sequential convex programming and smoothing techniques. In Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on, S. 543–548, IEEE, 2009.

[NecoaraNedelcu14] Necoara, I.; Nedelcu, V.: Rate analysis of inexact dual first-order methods application to dual decomposition. IEEE Transactions on Automatic Control, Bd. 59, Nr. 5, S. 1232–1243, 2014.

[NecoaraNedich15] Necoara, I.; Nedich, A.: A fully distributed dual gradient method with linear convergence for large-scale separable convex problems. In Control Conference (ECC), 2015 European, S. 304–309, IEEE, 2015.

[NecoaraSuykens08] Necoara, I.; Suykens, J.A.: Application of a smoothing technique to decomposition in convex optimization. IEEE Transactions on Automatic Control, Bd. 53, Nr. 11, S. 2674–2679, 2008.

[Nesterov05] Nesterov, Y.: Smooth minimization of non-smooth functions. Mathematical programming, Bd. 103, Nr. 1, S. 127–152, 2005.

[PakazadEtAl15] Pakazad, S.K.; Hansson, A.; Andersen, M.S.; Rantzer, A.: Distributed Semidefinite Programming with Application to Large-scale System Analysis. arXiv preprint arXiv:1504.07755, 2015.

[RawlingsEtAl08] Rawlings, J.B.; Bonné, D.; Jorgensen, J.B.; Venkat, A.N.; Jorgensen, S.B.: Unreachable setpoints in model predictive control. IEEE Transactions on Automatic Control, Bd. 53, Nr. 9, S. 2209–2215, 2008.

[RawlingsMayne09] Rawlings, J.B.; Mayne, D.Q.: Model predictive control: Theory and design. Nob Hill Pub., 2009.

[RubagottiPatrinosBemporad14] Rubagotti, M.; Patrinos, P.; Bemporad, A.: Stabilizing linear model predictive control under inexact numerical optimization. IEEE Transactions on Automatic Control, Bd. 59, Nr. 6, S. 1660–1666, 2014.

[StewartEtAl10] Stewart, B.T.; Venkat, A.N.; Rawlings, J.B.; Wright, S.J.; Pannocchia, G.: Cooperative distributed model predictive control. Systems & Control Letters, Bd. 59, Nr. 8, S. 460–469, 2010.

[StewartWrightRawlings11] Stewart, B.T.; Wright, S.J.; Rawlings, J.B.: Cooperative distributed model predictive control for nonlinear systems. Journal of Process Control, Bd. 21, Nr. 5, S. 698–704, 2011.

[Tran-DinhNecoaraDiehl16] Tran-Dinh, Q.; Necoara, I.; Diehl, M.: Fast inexact decomposition algorithms for large-scale separable convex optimization. Optimization, Bd. 65, Nr. 2, S. 325–356, 2016.

[VenkatEtAl08] Venkat, A.N.; Hiskens, I.A.; Rawlings, J.B.; Wright, S.J.: Distributed MPC strategies with application to power system automatic generation control. IEEE transactions on control systems technology, Bd. 16, Nr. 6, S. 1192–1206, 2008.

[ZhangLiPapachristodoulou15] Zhang, X.; Li, N.; Papachristodoulou, A.: Achieving real-time economic dispatch in power networks via a saddle point design approach. In 2015 IEEE Power & Energy Society General Meeting, S. 1–5, IEEE, 2015.

[ZhaoEtAl16] Zhao, C.; Mallada, E.; Low, S.; Bialek, J.: A unified framework for frequency control and congestion management. In Power Systems Computation Conference, 2016.