Model predictive control for linear systems: adaptive, distributed and switching implementations

By:

Bernardo Andrés Hernández Vicente

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
Department of Automatic Control and Systems Engineering

November 2018

Dedicated to my family,
for all their love and support.

Abstract

Thanks to substantial past and recent developments, model predictive control has become one of the most relevant advanced control techniques. Nevertheless, many challenges associated to the reliance of MPC on a mathematical model that accurately depicts the controlled process still exist. This thesis is concerned with three of these challenges, placing the focus on constructing mathematically sound MPC controllers that are comparable in complexity to standard MPC implementations.

The first part of this thesis tackles the challenge of model uncertainty in time-varying plants. A new dual MPC controller is devised to robustly control the system in presence of parametric uncertainty and simultaneously identify more accurate representations of the plant while in operation. The main feature of the proposed dual controller is the partition of the input, in order to decouple both objectives. Standard robust MPC concepts are combined with a persistence of excitation approach that guarantees the closed-loop data is informative enough to provide accurate estimates. Finally, the adequacy of the estimates for updating the MPC's prediction model is discussed.

The second part of this thesis tackles a specific type of time-varying plant usually referred to as switching systems. A new approach to the computation of dwell-times that guarantee admissible and stable switching between mode-specific MPC controllers is proposed. The approach is computationally tractable, even for large scale systems, and relies on the well-known exponential stability result available for standard MPC controllers.

The last part of this thesis tackles the challenge of MPC for large-scale networks composed by several subsystems that experience dynamical coupling. In particular, the approach devised in this thesis is non-cooperative, and does not rely on arbitrarily chosen parameters, or centralized initializations. The result is a distributed control algorithm that requires one step of communication between neighbouring subsystems at each sampling time, in order to properly account for the interaction, and provide admissible and stabilizing control.

Acknowledgements

First and foremost I would like to thank GOD for the opportunities I have received in life, and in particular for the PhD adventure.

Secondly, I would like to express my deepest gratitude to Dr. Paul Trodden, for his supervision during this project. Throughout my doctoral studies he took an active interest in my research, and was always available to discuss whenever I was struggling to move forward or in need of a different perspective. His continuous advice, encouragement and guidance where instrumental in helping me shape my research efforts and take this project to a fruitful end.

I would also like to thank several of my PhD candidate colleagues. In one way or another you helped me down this PhD path, and I will forever appreciate your friendship.

Finally, I would like to thank my family for their endless love and continuous encouragement. In particular to Monica, for her love and patience. Your company throughout these years made the bad look good, and the good feel wonderful.

I would also like to thank the Chilean Ministry of Education for the financial support that made this PhD possible (CONICYT PFCHA/Concurso para Beca de Doctorado en el Extranjero 2014 - 72150125).

Contents

List of Figures

List of Tables

Nomenclature

$\mathbb{R}^n$	Real coordinates space of dimension n
$\mathbb{N}_0$	Set of natural numbers inclunding 0
$\times$	Cartesian product of sets
$\prod$	Cartesian product of a collection of sets
$\subset$	Strict set inclusion
$\text{conv}(\cdot)$	Convex hull (of a set or collection of points)
$\text{int}(\cdot)$	Interior of a compact set
$\mathcal{B}_r$	Ball of radius r centred at the origin (of appropriate dimension)
$\oplus$	Minkowski sum
$\ominus$	Pontryagin difference
$\boldsymbol{I}_n$	Identity matrix of dimension n
$\mathbf{0}$	Zero matrix of appropriate dimension
$Q^\top$	Transpose of matrix Q
$Q(\geq) > 0$	Matrix Q is positive (semi) definite
$Q(\leq) < 0$	Matrix Q is negative (semi) definite
$\xi_m(Q)$	Minimum eigenvalue of matrix Q
$\xi_M(Q)$	Maximum eigenvalue of matrix Q
$\lvert x\rvert_p$	p-norm of vector x
$\lvert\lvert x\rvert\rvert_Q^2$	2-norm of vector x weighted by matrix Q

$d_H(x, \mathbb{S})$ Haussdorf distance between vector x and set $\mathbb{S}$

$\vec{u}_t$ Sequence of values of variable $u(t)$ up until time t

q^{-1} Backwards shift operator

π Pi constant

$\mathcal{A}_z$ z-transform of transfer function $\mathcal{A}$

Chapter 1

Introduction

1.1 Motivation

The last couple of decades have come with a steep increase in society's demand for improving the performance of several complex processes needed for its correct functioning. Some of them are evident to everyone, such as the need for reliable transmission of power from producers to consumers and the efficient organization of people and goods' transport networks. Others are rather concealed to the wider population, such as the production of said power and the manufacturing of such goods.

Feedback control, if implemented correctly, has the capability to greatly improve the output performance of many such processes and in many different aspects. The most standard approach is to design a feedback controller to track a particular operating point which is deemed "optimal" for a particular process, however there are several different measures that can be employed to define the latter. Economic approaches to feedback control, for example, aim to reduce directly the economic cost of a process.

Independent of the process, plant or system under consideration, and the measure used to drive its performance, it is highly likely that any real world problem will be subject to constraints. These are, in most cases, associated to the limited capabilities of the controlling variables, also called inputs. Take for example the task of coordinating several automated forklifts in a warehouse. The torque available to move them around is limited by the type of engine they are fitted with. Several processes are also constrained in their controlled variables, due to physical limitations and/or safety considerations. The temperature of a chemical reaction, for example, may have an upper limit to avoid damaging the physical elements in contact with the reactants (such as valves and pipes).

Constraints pose a major difficulty in the design of feedback controllers, and render several control techniques ineffective for certain applications. There are, on the other hand, some feedback control design approaches that have been particularly devised to account for the constraints affecting the plant; one of them is Model Predictive Control (MPC). MPC, when properly implemented, stabilizes the control target and ensures a constraint satisfying closed-loop, however these highly desirable properties depend almost solely on having a mathematical model that accurately depicts the controlled process.

The last three decades have seen major and extensive developments in the framework of MPC, regarding both theoretical and implementability issues. Nevertheless, the need for a model that accurately represents the plant is, still, one of the main obstacles in the real world implementation of theoretically sound MPC methodologies. Indeed, the complexity of some plants/processes makes it difficult to obtain an accurate mathematical depiction of it, even if all elements involved are assumed invariant. Furthermore, it is highly unlikely that a plant will remain invariant throughout its entire life cycle, suffering changes that are not necessarily easy to foretell, meaning that any model will eventually become outdated and inaccurate.

There are other obstacles associated to the need of MPC for a model that are not directly related to the latter's accuracy. Indeed, there exist several applications in which a single model is not enough to fully represent a particular process due to foreseeable changes in the plant's behaviour. The different representations of the plant may be accurately known, but not the order in which they become active throughout an operation cycle, nor the length of time they remain active, complicating the construction of an MPC controller.

Even if an accurate model of the process is available, this is only the first step in implementing proper MPC controllers. Indeed, the many desirable properties of MPC controllers rely, most commonly, on a centralized control and design of many model-dependent elements. Several processes, however, are formed by a network of subsystems which may already have controllers in place and that may be physically widespread. Centralized control of the network may then be physically or economically challenging, while the size of the plant may lead to computational intractability in the design process. Distributed control, and particularly distributed MPC, is the solution in such cases, however there are several difficulties associated to guaranteeing the usual properties that centralized MPC implementations enjoy. The latter are mostly associated to the design process, which in most cases is local and with some degree of arbitrariness, but needs to accommodate to global demands and remain optimal.

This thesis describes three new approaches to the design of MPC controllers that tackle some of the issues discussed above for constrained linear systems in discrete time representation. Standard MPC controllers enjoy simplicity of design and straightforward operation, thereby the predominant goal throughout this thesis is to devised MPC controllers that are comparable in complexity to standard MPC, both in the design and operation stages. The main tool employed in doing so is robustness, carefully implemented in a specific way depending on the issue at hand, in order to reduce the conservativeness introduced.

1.2 Organization of this thesis

This thesis is concerned with the general topic of MPC, but in particular with three novel MPC controllers devised to overcome three different and specific challenges. In view of this, each chapter is provided with its own literature review, in order to properly frame the contribution depicted therein. This thesis is organized as follows:

Chapter 2: Tube Model Predictive Control for Linear Discrete Time Systems

The main tool used in the design of the proposed MPC controllers is robustness. In particular, the robust approach known as tube MPC is employed. This is due to its implementation complexity being comparable to standard non-robust MPC, and its design complexity being slightly more demanding. This chapter describes two variants of tube MPC, the most familiar one initially depicted in [2] and a less known one described in [1]. The difference between both is related to how the nominal state trajectories are computed, and has important implications in the rationale behind the obtained results. Furthermore, a combination of both approaches is proposed in order to improve the size of the region of attraction associated to the variant described in [1].

Chapter 3: Tube-Based Adaptive Model Predictive Control with Persistence of Excitation

One of the main obstacles in implementing MPC controllers is the uncertainty in the model parameters during the controller design stage. Moreover, although a model can be accurate at a certain time instant, changes in the surrounding environment and internal modifications to the plant may result in changes in the dynamical properties of the process.

In order to properly control said time varying plants and capture future changes in the model, this chapter puts forward a new dual controller that simultaneously guarantees constraint satisfaction, robust stabilizability of the control target, and convergent recursive least squares estimates. This is achieved by partitioning the input and separating both conflicting objectives. The control/constraint related objectives are then achieved by a standard implementation of tube MPC, while the estimation objectives are achieved by a novel MPC-like receding horizon optimization that guarantees a persistently exciting regressor vector [3].

Chapter 4: Robust MPC for switching systems: Minimum dwell-time for feasible and stabilizing switching

Plants with dynamics varying in an uncertain fashion are not the only obstacle in implementing stabilizing and constraint admissible MPC controllers. There exist several examples of plants whose dynamics leap within a finite set of known conditions or modes. These are usually referred to as switching systems, and the difficulties in designing an appropriate MPC controller grow with the heterogeneity of the different modes, and the uncertainty of the sequence in which they become active.

This chapter proposes a new approach to the computation of mode-dependent dwell-times that guarantee constraints satisfaction and stability when each heterogeneous mode is controlled by a local MPC controller. Several different possibilities are studied, including non-disturbed and disturbed systems, and independent or coupled design approaches. The proposed approach relies in the well-known exponential stability result available for standard MPC and tube MPC controllers.

Chapter 5: Distributed MPC for dynamically coupled systems: a chain of tubes

This chapter proposes a new non-cooperative distributed MPC algorithm for controlling a network of dynamically coupled subsystems. The algorithm requires one step of communication and two steps of optimization at each sampling time, in order to provide a constraint satisfying and stabilizing control action for each subsystem in the network. The general design approach requires several additional elements when compared to standard MPC implementations however simple design choices result in a design marginally more complex than standard tube MPC. Most of the design procedure is local to each subsystem, requiring only some information about state and input constraints of its neighbours (that is, those subsystems that interact with it).

One of the main drawbacks of the proposed approach is, as in many other distributed MPC algorithms proposed to date, the global stabilizability requirements that force the design process to be centralized. This disadvantage is thoroughly discussed in this chapter, concluding that it is redundant in the framework of tube-based distributed MPC.

Chapter 6: Concluding remarks
Finally, this chapter summarizes the contributions put forward by this thesis, and discusses several avenues for future work.

1.3 Publications and presentations

Some of the work presented in this thesis has also been published and/or presented at various events:

- **Hernandez, B.**, Baldivieso, P., Trodden, P., "Distributed MPC: Guaranteeing Global Stabilizability from Locally Designed Tubes", *IFAC PapersOnLine*, 50, 1, pp. 11829–11834 (presented at the *20th IFAC World Congress*, July 2017).

- **Hernandez, B.**, Trodden, P., "Distributed Model Predictive Control Using a Chain of Tubes", *Proceedings of the 11th UKACC Conference on Control*, Belfast, August 2016.

- **Hernandez, B.**, Trodden, P., "Persistently Exciting Tube MPC", *Proceedings of the 2016 American Control Conference*, pp. 948–953, Boston, July 2016.

- **Hernandez, B.**, Trodden, P., "Design of reconfigurable and non-centralized MPC implementations", *UK Automatic Control Council, PhD showcase*, May 2016.

- **Hernandez, B.**, Trodden, P., "Distributed Model Predictive Control Using a Chain of Tubes", *University of Sheffield, ACSE Research Symposium*, October 2016.

- **Hernandez, B.**, Trodden, P., "Persistently Exciting Tube MPC (PETMPC)", *University of Sheffield, ACSE Research Symposium* (poster), October 2015.

- **Hernandez, B.**, Trodden, P., "Persistently Exciting Tube MPC (PETMPC)", *University of Oxford, Workshop on Control and Optimisation* (poster), September 2015.

Chapter 2

Tube Model Predictive Control for Linear Discrete Time Systems

2.1 Introduction

Model predictive control (MPC) is an advanced control technique that relies on having access to an accurate model of the plant or system being controlled in order to predict its future behaviour and make optimal control decisions on-line [4]. MPC has been successfully employed by the chemical industry since the 1950s [4] to improve the performance of its processes, however it is only in the recent decades that most of the theoretical guarantees regarding its implementation have been devised [1].

One of the key features of MPC is its inherent ability to account for hard constraints (unlike classical techniques such as LQR or PID) that may arise, for example, from safety requirements or actuator limits [5]. In order to do so MPC solves a constrained optimization problem in which the optimized function represents the control objective not only at the current time instant, but over a certain time span into the future. A model of the plant is employed to predict its behaviour and evaluate the objective, and by enforcing the constraints throughout this time span (horizon) it is possible to ensure future constraint satisfaction. However, MPC is not an open-loop control strategy, indeed at each time instant the current state/output of the system is updated (from measurements) and the relevant forecasting optimization problem is re-solved. This is why MPC is also referred to as receding horizon control.

The analysis of the closed-loop stability of MPC controllers is usually based on the concept of Lyapunov stability [6]. This is because the explicit consideration of the constraints turns the closed-loop non-linear, even if the plant itself is linear. Many different approaches have been devised in the last three decades [7–19], however the current standard approach is to follow a purposeful design [1,4] in order to make the objective function itself a Lyapunov function for the controlled system, yielding a straightforward path for a stability guarantee.

Although the receding horizon implementation of MPC controllers introduces feedback at each time instant, the predictions within each optimization are made independently. This highlights the importance of an accurate prediction model in order to maintain stability and keep the system within its constraints. However, even if an accurate model of the system is available, external disturbances may cause constraint violations or even render the plant unstable if actuator constraints are present. Although MPC does posses an inherent degree of robustness [1, 20], the consensus is that to embed robustness in MPC controllers, while maintaining optimality, it is necessary to optimize over control policies instead of just control actions. This, in the most general case, increases the size of the optimization problem beyond practicality.

In order to obtain a robust MPC controller that is feasibly implementable, many robust MPC implementations have been proposed that settle for a trade-off between optimality and computational complexity at different levels [2,20–29]. The technique called tube MPC [2], devised to control linear systems under the effect of bounded disturbances, has attracted considerable attention due to its relative ease of implementation and lower number of extra design variables. Furthermore, the generality of its design requirements allows its use in solving control problems that are not necessarily robust problems but can be reformulated as one, such as the control of distributed/large-scale systems [30–36].

The remainder of this chapter introduces the class of systems that are going to be considered and presents the preliminaries and necessary definitions required for the implementation of two tube MPC variants: standard [2] and modified [1, Chapter 3]. These are the main tools employed in developing the adaptive, switching and distributed MPC implementations described in Chapters 3–5, therefore are revised here.

2.2 Constrained Linear Time Invariant Systems

The fundamental state space representation of a discrete, linear time invariant (LTI) system affected by additive disturbances is

$$x(t+1) = Ax(t) + Bu(t) + Ew(t) \tag{2.1a}$$

$$y(t) = Cx(t) + Du(t), \tag{2.1b}$$

where $t \in \mathbb{N}_0$ represents the discrete time instant, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$ and $y(t) \in \mathbb{R}^p$ are correspondingly the state, input and output of the system, and $w(t) \in \mathbb{R}^q$ represents an uncontrolled and uncorrelated external disturbance affecting the plant. Given the dimensions of the corresponding vectors, the system matrices fulfil $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$, $E \in \mathbb{R}^{n\times q}$, $C \in \mathbb{R}^{p\times n}$ and $D \in \mathbb{R}^{p\times m}$. In order to simplify notation, and without loss of generality, in the following it is assumed that $p = n$ and $E = \boldsymbol{I}_n$. Furthermore, in what follows it is assumed that full state measurements are available and there is no feed-forward thus $D = \mathbf{0}$ and $C = \boldsymbol{I}_n$. Finally, it is assumed that (2.1) is a minimal representation of the plant, meaning that n is the McMillan dregree of (2.1).

The linear state space model in (2.1a) may represent an almost infinite variety of linear systems, therefore, depending on the application, both state and input may be subject to various types of constraints (for example the fully open/closed position of a valve, or the safe maximum operation temperature). In general, any constraint can be represented by a particular subset of the relevant coordinate space, however optimizing over non-convex sets, for example, is a challenging task. In what follows all constraint sets are assumed convex and compact unless otherwise stated. This implies that a degree of conservatism is introduced for certain applications, say for example, if it is necessary to constraint the problem within a convex subset of the true non-convex constraint set. Moreover, consider the relevant definition.

Definition 2.1. A $\mathcal{C}$-set is a convex and compact set that contains the origin and a $\mathcal{PC}$-set is a $\mathcal{C}$-set that contains the origin in its interior.

System (2.1a) is then assumed to be subject to the following constraints for all $t \in \mathbb{N}_0$

$$x(t) \in \mathbb{X} \subseteq \mathbb{R}^n \tag{2.2a}$$

$$u(t) \in \mathbb{U} \subseteq \mathbb{R}^m, \tag{2.2b}$$

where $\mathbb{X}$ and $\mathbb{U}$ are $\mathcal{PC}$-sets .

Remark 2.1. The stabilizability guarantees of standard MPC controllers generally require $\mathbb{X}$ to be closed but not necessarily compact. In what follows, however, the boundedness of $\mathbb{X}$ is required for other purposes and therefore it is assumed throughout.

In most cases the input constraint corresponds to a saturation of the actuators, which implies that this constraint is enforced by the physicality of the plant. State constraints, on the other hand, are in most cases associated to safety limits, for example the maximum pressure in a boiler, and have to be enforced through the proper selection of the control actions. However, independent of the control technique employed, it is not possible to guarantee satisfaction of state constraints under unbounded (unstructured) disturbances. For the particular case of tube-based robust MPC implementations, it is assumed that the disturbance $w(t)$ is unknown but belongs, at all times, to a $\mathcal{C}$-set $\mathbb{W} \subseteq \mathbb{X}$.

2.2.1 Invariant sets

Generally speaking, tube MPC requires a single additional design parameter when compared to standard MPC implementations: the existence and knowledge of a stabilizing linear feedback gain. Consider then the following assumption.

Assumption 2.1. The system (A, B) is stabilizable. Furthermore, a linear feedback gain $K \in \mathbb{R}^{m\times n}$ is available such that the closed-loop system $A_K = A + BK$ is stable, i.e. the closed-loop system matrix A_K is Schur.

Tube MPC controllers employ a variety of invariant sets, associated to the additional design parameter, in order to guarantee robust constraint satisfaction and robust regulation of (2.1a) under the effects of the disturbance $w(t)$. For the implementations revised here, [2] and [1, Chapter 3], the definitions of two such sets are required.

Definition 2.2. Given a linear feedback gain K that satisfies Assumption 2.1, a set $\mathbb{S}$ is a robust positive invariant (RPI) set for system (2.1a) in closed-loop with K if for every $x \in \mathbb{S}$, $A_K x + w \in \mathbb{S}$ for all $w \in \mathbb{W}$. Equivalently if $A_K\mathbb{S} \oplus \mathbb{W} \subseteq \mathbb{S}$. If $\mathbb{W} = \{0\}$ the adjective *robust* is dropped from the definition.

Definition 2.3. An RPI (PI) set is called admissible for system (2.1a) in closed loop with K with respect to constraints (2.2) if $\mathbb{S} \subseteq \mathbb{X}$ and $K\mathbb{S} \subseteq \mathbb{U}$.

2.3 Tube Model Predictive Control

The main feature of the tube MPC algorithm proposed in [2] is that the full control action is composed by a nominal MPC controller, to regulate the trajectories of a fictitious undisturbed nominal system, and a linear feedback, to act over the prediction error induced by the disturbance. The MPC controller drives the nominal trajectories towards the control target while the linear feedback keeps the true trajectories within a *tube* centred around the nominal predictions, despite the disturbances.

The nominal reference system is defined by

$$\bar{x}(t+1) = A\bar{x}(t) + B\bar{u}(t),$$

hence the state deviation induced by the disturbance is $e(t) = x(t) - \bar{x}(t)$. The true input is composed by a nominal MPC action and a linear feedback acting over the error, that is $u(t) = \bar{u}(t) + Ke(t)$. It follows then that

$$e(t+1) = A_K e(t) + w(t). \tag{2.3}$$

It is clear that the nominal MPC controller cannot make use of the whole constraint sets (2.2) if robustness is to be attained. Assume then that the true constraints are tightened by a certain set $\mathcal{C}$-set $\mathbb{S}$, yet to be defined

$$\bar{\mathbb{X}} = \mathbb{X} \ominus \mathbb{S} \tag{2.4a}$$

$$\bar{\mathbb{U}} = \mathbb{U} \ominus K\mathbb{S}, \tag{2.4b}$$

it follows that if $e(t) \in \mathbb{S}$ and $(\bar{x}(t), \bar{u}(t)) \in \bar{\mathbb{X}} \times \bar{\mathbb{U}}$ then

$$x(t) = \bar{x}(t) + e(t) \in \bar{\mathbb{X}} \oplus \mathbb{S} \quad \subseteq \mathbb{X} \tag{2.5a}$$

$$u(t) = \bar{u}(t) + Ke(t) \in \bar{\mathbb{U}} \oplus K\mathbb{S} \subseteq \mathbb{U}. \tag{2.5b}$$

Equation (2.5) shows that, as long as the trajectory deviation $e(t)$ remains in $\mathbb{S}$ and the nominal variables are kept inside the tightened constraint sets in (2.4), the true constraints are met by the disturbed system (2.1a) regardless of the disturbance. The set $\mathbb{S}$ represents the cross section of the tube, and in order to ensure that the trajectory deviation remains inside it at all times, it is sufficient to select $\mathbb{S}$ as an RPI set for A_K and $\mathbb{W}$ (although, many different options with various degrees of complexity exist [37,38]). Furthermore, in order for the tightened constraints in (2.4) to not be empty, it is sufficient to choose

$\mathbb{S}$ as an admissible RPI with respect to the constraints (2.2).

In order to keep the nominal state trajectories within the tightened constraints, the nominal control action $\bar{u}(t)$, is driven by a nominal-tightened MPC controller, that is one that ignores the disturbances and enforces the tightened constraints in (2.4). However, setting up the appropriate (nominal) optimization problem requires the fictitious nominal state $\bar{x}(t)$ at each time instant. There are, generally, two approaches to do so, either optimize the nominal trajectories at each time instant [2] or allow them to evolve independently [1, Chapter 3]. The former is more general and therefore is addressed first.

2.3.1 Optimizing trajectories

The control law that defines the nominal control action is defined by a standard MPC controller. The objective function (performance index) employed by most MPC implementations is the finite horizon LQR with a terminal penalty that approximates (perfectly in the absence of constraints) the infinite horizon LQ problem. This selection allows to achieve significant theoretical results with simplicity. Consider then a cost function defined by

$$J_N\left(\bar{\boldsymbol{u}}, \bar{x}_0\right) = \sum_{k=0}^{N-1} \ell\left(\bar{x}_k, \bar{u}_k\right) + V_f\left(\bar{x}_N\right) = \sum_{k=0}^{N-1}\left(||\bar{x}_k||_Q^2 + ||\bar{u}_k||_R^2\right) + ||\bar{x}_N||_P^2, \quad (2.6)$$

where $(\bar{x}_k, \bar{u}_k)$ are the nominal predicted values at prediction time k and N is the prediction horizon. For a given predicted sequence of control actions $\bar{\boldsymbol{u}} = \{\bar{u}_0, \ldots, \bar{u}_{N-1}\}$, the predicted state trajectory associated to it is defined by $\bar{\boldsymbol{x}}\left(\bar{\boldsymbol{u}}\right) = \{\bar{x}_0, \cdots, \bar{x}_N\}$ with $\bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k$. The optimization problem solved at each time instant to compute the optimal, constraint admissible, nominal control action is

$$\mathbb{P}_N(x(t)): \qquad \min_{\bar{\boldsymbol{u}}, \bar{x}_0} J_N\left(\bar{\boldsymbol{u}}, \bar{x}_0\right) \quad (2.7a)$$

s.t. (for $k = 0, \ldots, N-1$)

$$x(t) - \bar{x}_0 \in \mathbb{S} \quad (2.7b)$$

$$\bar{x}_{k+1} = A\bar{x}_k + B\bar{u}_k \quad (2.7c)$$

$$\bar{x}_k \in \bar{\mathbb{X}} \quad (2.7d)$$

$$\bar{u}_k \in \bar{\mathbb{U}} \quad (2.7e)$$

$$\bar{x}_N \in \bar{\mathbb{X}}_f. \quad (2.7f)$$

It is clear from (2.7b) that the initial state of the nominal predictions, and therefore the value of $\bar{x}(t)$ at time t, is not externally defined (via measurements for example), but optimized by the controller at each time instant. This is possible because, as stated before, these are fictitious, and do not correspond to the true plant state trajectories. Furthermore, constraint (2.7b) ensures that the trajectory deviation is always contained inside the set $\mathbb{S}$ as long as such $\bar{x}_0 \in \bar{\mathbb{X}}$ exists. Indeed, the optimization problem (2.7) is not feasible for all initial states, but only for a subset of the associated state constraint set. Define

$$\bar{\mathcal{X}}_N = \{\bar{x}_0 \in \mathbb{R}^n \mid \exists\, \bar{\boldsymbol{u}} \text{ s.t. for } k = 0, \ldots, N-1, \text{ (2.7c)–(2.7f) hold}\}, \qquad (2.8)$$

it follows that, independent of the actual size or shape of $\bar{\mathcal{X}}_N$, for all x inside $\mathcal{X}_N = \bar{\mathcal{X}}_N \oplus \mathbb{S}$, there exists $\bar{x}_0 \in \bar{\mathcal{X}}_N$ such that (2.7b) holds. $\mathcal{X}_N$ is the feasibility region of the optimization problem (2.7) and it is also called the region of attraction (RoA) of the associated controller. Let the optimum and optimal value of the cost function be respectively defined by

$$\big(\bar{\boldsymbol{u}}^\star(x(t)), \bar{x}_0^\star(x(t))\big) = \arg \mathbb{P}_N(x(t)) \qquad (2.9a)$$

$$V_N(x(t)) = J_N(\bar{\boldsymbol{u}}^\star(x(t)), \bar{x}_0^\star(x(t))), \qquad (2.9b)$$

set the nominal input to the associated receding horizon control law $\bar{u}(t) = \bar{\kappa}_N(x(t)) = \bar{u}_0^\star(x(t))$ and let the nominal trajectories be updated with $\bar{x}(t) = \bar{x}_0^\star(x(t))$. The following result holds [1, 2].

Proposition 2.1. If (a) Assumption 2.1 holds with a certain K, (b) the sets $\mathbb{S}$ and $\bar{\mathbb{X}}_f$ are, correspondingly, admissible RPI and PI sets for A_K with respect to constraints (2.2), disturbance set $\mathbb{W}$ and tightened constraint (2.4a), (c) $Q, R > 0$, and P fulfils $A_K^\top P A_K + Q + K^\top R K - P \leq 0$ and (d) the loop is closed with $u(t) = \kappa_N(x(t)) = \bar{\kappa}_N(x(t)) + Ke(t)$, then (1) the optimization problem (2.7) is recursively feasible, (2) state and input constraints are met at all times despite the disturbances, and (3) there exist constant scalars $b, d, f > 0$ such that for all $x \in \mathcal{X}_N$ and $w \in \mathbb{W}$ it holds that

$$b|\bar{x}_0^\star(x)|_2^2 \leq V_N(x) \leq d|\bar{x}_0^\star(x)|_2^2 \qquad (2.10a)$$

$$V_N\left(Ax + B\kappa_N(x) + w\right) - V_N(x) \leq -f|\bar{x}_0^\star(x)|_2^2. \qquad (2.10b)$$

The proof is omitted as it can be found in [1]. Proposition 2.1 implies that for any feasible initial state, the closed-loop remains feasible, i.e. the optimization has a solution and the constraints are satisfied at all time instants. Furthermore, (2.10) implies that the optimal cost function is a Lyapunov function for the nominal closed-loop, resulting in the following Corollary.

Corollary 2.1. There exist constant scalars $c > 0$ and $\gamma \in (0, 1)$ such that for all feasible $x(0) \in \mathcal{X}_N$ it holds that

$$|\bar{x}(t)|_2 \leq c\gamma^t |\bar{x}(0)|_2 \tag{2.11a}$$

$$d_H(x(t), \mathbb{S}) \leq c\gamma^t |\bar{x}(0)|_2. \tag{2.11b}$$

Equivalently, the origin is exponentially stable for the closed-loop nominal trajectories (2.11a) and the true trajectories converge exponentially fast to $\mathbb{S}$ (2.11b).

The proof to Corollary 2.1 is also available in [1] and hence omitted here. In practice, and as pointed out in [1], Corollary 2.1 does not guarantee (robust) exponential stability of the set $\mathbb{S}$ for the true trajectories, but only for the set difference equation associated to the disturbed dynamics (i.e. the set that contains the true trajectory approaches $\mathbb{S}$ exponentially fast rather than the trajectory itself).

Note, however, that given (2.6), the value $\bar{x}_0^\star(x(0))$ depends solely on the initial state $x(0)$ because no disturbance has affected the system yet, thus the following stronger corollary to Proposition 2.1 holds. This is a new result.

Corollary 2.2. There exists a constant scalar $\delta > 0$ such that for all $x(0) \in \mathcal{X}_N$

$$d_H(x(t), \mathbb{S}) \leq c\gamma^t \delta d_H(x(0), \mathbb{S})$$

Corollary 2.2 provides a result that is not explicitly provided in [1,2], hence a proof is contributed here.

Proof. Given the definition of the trajectory deviation $e(t)$ it holds that

$$\begin{aligned} d_H(x(0), \mathbb{S}) &= d_H(\bar{x}(0) + e(0), \mathbb{S}) = d_H(\bar{x}_0^\star(x(0)) + e(0), \mathbb{S}) \\ &\leq d_H(\bar{x}_0^\star(x(0)) + e(0), e(0)) = |\bar{x}_0^\star(x(0))|_2 \end{aligned}$$

where the inequality follows from the fact that $e(0) \in \mathbb{S}$. Given an initial state, $\bar{x}_0^\star(x(0))$ can be computed off-line, hence the values of $d_H(x(0), \mathbb{S})$ and $|\bar{x}_0^\star(x(0))|_2$ are also available before initialization. Define then $\delta > \bar{\delta}$ with

$$\bar{\delta} = \max_{x \in \mathcal{X}_N} {}^{|\bar{x}_0^\star(x)|_2} / {}_{d_H(x, \mathbb{S})},$$

it follows that

$$\begin{aligned} &|\bar{x}_0^\star(x(0))|_2 < \delta d_H\left(x(0), \mathbb{S}\right) \\ \implies &c\gamma^t|\bar{x}_0^\star(x(0))|_2 < c\gamma^t \delta d_H\left(x(0), \mathbb{S}\right) \\ \implies &d_H\left(x(t), \mathbb{S}\right) \leq c\gamma^t \delta d_H\left(x(0), \mathbb{S}\right) \end{aligned}$$

which completes the proof. ■

Corollary 2.1 ensures that the set $\mathbb{S}$ is indeed exponentially stable for the true trajectories when the loop is closed with the tube MPC composite control law $\kappa_N(\cdot)$.

2.3.2 Independently evolving trajectories

The nominal state trajectory is regarded as an optimization variable in (2.7), therefore its value is not known a priori by the controller, despite the model dynamics and the values of $\bar{x}(t-1)$ and $\bar{u}(t-1)$ being available at time t. For several reasons particular to the challenges addressed in Chapters 3–5, the uncertainty introduced by the continuous re-optimization of $\bar{x}(t)$ may be detrimental. It is shown in [1, Chapter 3] that constraint (2.7b) can be replaced by

$$\bar{x}_0 = A\bar{x}(t-1) + B\bar{\kappa}_N(x(t-1)) \tag{2.12}$$

while keeping all the results from Proposition 2.1. Note, however, that this reduces the degrees of freedom of the optimization, thus resulting in a possibly higher cost at each time instant.

This approach requires the user to initialize the nominal trajectories. It is proposed in [1, Chapter 3] to do so by setting $\bar{x}(0) = x(0)$, however this significantly reduces the RoA to $\bar{\mathcal{X}}_N$ in (2.8). It is proposed here to initialize the nominal trajectories by solving $\mathbb{P}_N(x(t))$ for $t = 0$ and subsequently replacing constraint (2.7b) with (2.12) for all $t > 0$. This results in the nominal and true states evolving separately after the first time instant, however the feasibility region remains as $\mathcal{X}_N$. The latter approach is implicitly observed in [1] by saying that the RoA of the modified controller is $\mathcal{X}_N$, however their initialization proposal does not truly achieve this.

Note that, for all $t > 0$, the optimization does not depend on the true state, therefore the cost function and its optimal value depend only on the nominal state trajectories.

Proposition 2.2. If Proposition 2.1 holds and (a) constraint (2.7b) is replaced by (2.12) for all $t > 0$, then (1) the optimization problem (2.7) is recursively feasible, (2) state and input constraints are met at all times despite the disturbances, and (3) there exist constant scalars $b, d, f > 0$ such that for all $\bar{x} \in \bar{\mathcal{X}}_N$ it holds that:

$$\begin{aligned} b|\bar{x}|_2^2 \leq & V_N(\bar{x}) \leq d|\bar{x}|_2^2 \\ V_N\left(A\bar{x} + B\bar{u}_0^\star(\bar{x})\right) - & V_N(\bar{x}) \leq -f|\bar{x}|_2^2. \end{aligned}$$

Furthermore, define the composite state vector $\chi(t) = (x(t), \bar{x}(t))$, there exists constant scalars $c > 0$ and $\gamma \in (0, 1)$ such that for all $x(0) \in \mathcal{X}_N$ it holds that

$$|\bar{x}(t)|_2 \leq c\gamma^t |\bar{x}(0)|_2 \tag{2.13a}$$

$$d_H\left(\chi(t), \mathbb{S} \times \{0\}\right) \leq 2c\gamma^t d_H\left(\chi(0), \mathbb{S} \times \{0\}\right). \tag{2.13b}$$

Equivalently, the origin is exponentially stable for the closed-loop nominal trajectories (2.13a) and the set $\mathcal{R} = \mathbb{S} \times \{0\}$ is exponentially stable for the composite trajectories $\chi(t)$ (2.13b).

Proof. If $u(0)$ is set by solving $\mathbb{P}_N(x(0))$, it follows from Proposition 2.1 that $x(1) \in \mathcal{X}_N$ and $\bar{x}(1) \in \bar{\mathcal{X}}_N$. The rest of the proof follows from the proof in [1]. ■

Again, note that (2.13b) does not guarantee exponential stability of the set $\mathbb{S}$ for the true closed-loop, but exponential stability of the set $\mathcal{R}$ for the composite trajectories $\chi(t)$, which is a slightly weaker result. However, Corollary 2.2 holds for Proposition 2.2 also, thus $\mathbb{S}$ is indeed exponentially stable for the true trajectories when in closed-loop with $\kappa_N(\cdot)$.

2.4 Summary

This chapter introduced the general class of systems and robust controllers that will be employed to tackle some of the challenges related to controlling uncertain and changing constrained linear system. Well known results related to tube-based MPC controllers were presented and a minor modification was proposed to enlarge the RoA of the tube MPC variant that does not re-optimize the nominal trajectories, described in [1].

It is important to emphasize that although Section 2.2 introduces constrained LTI systems as the object of study, the tools presented here can be exploited to control constrained LTV systems. The key feature that allows this is to observe the changing/uncertainty of the system as a disturbance affecting an, otherwise, invariant system. The following chapters address the advantages and drawbacks of this rationale in different arrangements.

Chapter 3

Tube-Based Adaptive Model Predictive Control with Persistence of Excitation

3.1 Introduction

A key element of any MPC controller is, certainly, the mathematical model used to make the necessary predictions. As discussed in Chapter 2.1 standard MPC controllers have a certain degree of robustness against prediction errors, however uncertainties in the model can lead to large mismatches and therefore have a significant impact on the overall performance of the controller [39,40]. Moreover, several MPC implementations, including those presented in Chapter 2.1, rely on the computation of invariant sets. The latter are highly model dependent, therefore model mismatch can not only decrease performance but also result in constraint violation or even unstable behaviour of the closed-loop.

It seems then that the necessity of a precise prediction model may be one of the main drawbacks of any MPC implementation due to the challenges related to its acquisition. First-principle approaches, for example, may result in models that are too complex for controller design [41] or hinge on a-priori simplifications that neglect important input-output interactions. Even if an adequate model is obtained through such techniques several obstacles remain on the path to a successful MPC implementation: large non-linearities, expected degradation due to normal operation (resulting in changes to the values of the model parameters), expected structural changes due to operation conditions (such as payload changes) and the explicit realization of the plant (i.e. the uncertainties related to its manufacture [42]). On the other hand, the cost of system identification experiments may be prohibitively large [43], particularly if they ought to be regularly repeated due to degradation or plant changes.

In order to improve the performance of MPC controllers in the context of uncertain and changing systems, it is necessary to obtain new and more accurate descriptions of the current condition of the plant throughout its life-time. Moreover, in order to avoid expensive experiments and account for continually changing systems, it is necessary to obtain these models on-line. To do so, a form of system identification ought to be coupled with the controller. This combination is commonly known as the dual control problem [44], given that it confronts two incompatible objectives: the controller aims to regulate the system towards a desired optimal operation point while the identifier requires the system to be constantly disturbed, in order to accurately estimate it [3]. Moreover, it is necessary not only to identify a better model on-line, but also use this new and better description of the system to provide more accurate predictions for the MPC optimization. Changing the prediction model of an MPC controller while maintaining its key control properties, however, is not trivial due to the high model-dependency of most of the controller's design parameters. This adds a second layer of complexity to the challenges inherent to the dual control problem.

The techniques devised to solve these problems, within the MPC context, are usually grouped under the label of Adaptive MPC (AMPC) [45], however this concept has been occasionally misused. For example, the techniques presented in [46–48] are referred to as AMPC approaches, however gain-scheduling or time-varying MPC might be more appropriate. Indeed, a look-up table is constructed in [46–48] by successive linearisation of a non-linear model around different operating points. At each time instant, the model used for predictions is chosen depending on the current state, however no stability guarantees are provided. Similar misconception is found in [49], where the prediction model is computed as a fuzzy weighted combination of an ensemble of linear models that represent the plant throughout the entire state space. A more rigorous approach is found in [50, 51], where the prediction model is properly defined by a linear time varying model, however this is obtained via constant linearisation of a known, thus not uncertain, non-linear model.

There are, generally, three properties that can be used to categorize AMPC algorithms:

- Are there any closed-loop stability and/or constraint satisfaction guarantees?

- Is the system identification (parameter estimation) algorithm guaranteed to converge to an accurate description of the true plant?
- Is the new model effectively used to update the MPC prediction model?

The latter is of paramount importance, since it is the characteristic that allows to take advantage of the information provided by the system identification algorithm. Adaptation of the prediction model is present in almost all current AMPC algorithms, however the updating is performed in different ways owing to the type of uncertainty considered and the MPC technique employed. In general, most approaches proposed to date either fail to guarantee closed-loop stability, constraint satisfaction and/or estimation convergence guarantees. Nonetheless, many provide good solutions for certain aspects of the problem, and thus are important to contextualize the solution proposed here.

3.1.1 AMPC without estimation guarantees

There exist many application driven solutions to the AMPC problem that avoid almost any type of rigorous analysis [52–57]. These usually resort to suitable assumptions on system behaviour (such as open-loop stability) that fit the purpose at hand, but greatly reduce their applicability. In [53], for example, a fuzzy supervisor overviews the closed-loop behaviour and, based on some arbitrary performance criteria which include a numeric evaluation of stability, *adapts* the controller by modifying some of its design parameters (weights and terminal conditions).

A simple, yet effective way of formally addressing a possible mismatch between plant and model is to characterize it with respect to a predefined model structure, bound it, and treat it as a disturbance affecting an, otherwise, invariant plant [40, 58–60]. The model structure and the particular selection of its nominal parameters are not trivial to obtain but may be available from first-principle approaches and previous identification experiments. Furthermore, quantifying the mismatch may prove to be a challenging task, however a priori knowledge about the plant and its operation program are usually enough to properly estimate the expected mismatch (for example the stiffness variation of certain mechanical parts due to increased operational temperature, or the expected payload changes on unmanned vehicles). If the mismatch has been accurately estimated, robust approaches such as tube-based ones can be employed to regulate a nominal representation of the plant subject to the prediction error induced by model mismatch. Although this approach addresses the uncertainty, if no form of system estimation is coupled with it, the mismatch itself remains un-addressed. This implies that only robust stabilizability and robust constraint satisfaction can be achieved, albeit true external disturbances may be completely absent.

A number of recent robust-based approaches have focused on the control guarantees when implementing AMPC algorithms. Uncertain continuous time state space models subject to state and input constraints are considered in [59]. The adaptive estimation algorithm provides not only an estimate of the model parameters, but also an estimate of the error bound. This bound is included in a comparison model robust MPC [61] in order to reduce uncertainty and guarantee robust stability and constraint satisfaction via standard Lyapunov arguments. The prediction model is updated by the estimates only when the information provided by the data reduces the uncertainty on the parameters. This is quantified by checking whether the smallest eigenvalue of the inverse of the information matrix is larger than at the previous step. However, this is not guaranteed to happen at any time instant.

A similar approach is developed in [60] but for a class of non-linear continuous time systems with a parametric affine type of uncertainty. In this case the estimation algorithm guarantees unbiased estimates [3] alongside with a non-increasing parameter error bound, however, as in [59], the latter is not guaranteed to decrease at each time instant. A min-max robust MPC implementation is proposed to account for the uncertainty arising from the model mismatch. The recursive estimation algorithm is included in the MPC predictions in order to account for future estimation and reduce conservativeness. The latter, however, may lead to constraint violation because predictions may be far off the true plant behaviour. This becomes evident in that the terminal constraint set and associated terminal cost have to be computed accounting for all possible values of the true estimation error. Overall, the min-max optimization is considered a computationally intractable problem, and therefore replaced by a Lipschitz-based worst-case approach similar to the robust MPC technique in [23]. This is extended to account for external disturbances in [39] and to discrete time systems in [62].

The algorithm proposed in [63] also resorts to a robust approach similar to that in [23], but the algorithm is only applicable for open-loop stable plants. At initialization a set of all models that may represent the plant is supposed to be known in polytopic form. Every time step a set membership identification algorithm updates this set in a recursive inclusion fashion, and the prediction model used for the MPC is selected as the centre of the largest ball contained in this set. Recursive feasibility of the optimisation is secured by an additional group of constraints designed to ensure that the output of any model inside the current set satisfies the output constraints. For MIMO systems, these additional constraints are first introduced as a set of linear programming problems (similar to a min-max optimization problem) but then transformed into a set of auxiliary decision variables. This, although straightforward, increases the computational complexity of the problem.

It should be clear that predictions cannot be considered as reliable data for the parameter estimates computation, since they are not real measurements. In [64], however, a-priori knowledge on the rate of change of the uncertainty and the error bound provided by the estimator are used to predict a set that contains the uncertainty throughout the prediction horizon, given the current conditions. This time-varying error bound is fed to the MPC controller, unlike in [39, 60] where the bound was fixed, and therefore uncertainty is decreased. A min-max MPC algorithm is shown to guarantee ultimate boundedness of the closed-loop under several assumptions that include the existence of an invariant set to be used as terminal constraint. However, constraints may be violated, and again, the parameter error is not guaranteed to decrease at each time instant.

In all of the above, the obtained estimates are not necessarily convergent to the true plant parameters, because proper excitation of the closed-loop system is assumed rather than guaranteed. A similar set-up leads to an analogous outcome in [65, 66], where constrained polytopic linear difference inclusion (pLDI) systems are considered. In this case the pLDI structure is exploited in order to address the uncertainty in a more structured way when compared to [39,60]. Assuming a convex and bounded set of possible parameters, arguments from the robust controller proposed in [26] are employed to compute parameter dependent terminal conditions. A considerable advantage with respect to [60, 64] is that constraints are satisfied even if the parameters are slowly changing within their initially assumed bound. A drawback of embedding a strong structure in the controller's design is that, to maintain feasibility of the optimization, newly estimated parameters can only be included at the end of the prediction, moving forward one step at each time instant. Furthermore, there is no discussion about the associated estimator, and it is only assumed that a convergent one exists.

The same type of systems is studied in [58], however the standard MPC cost function is enhanced in order to push for convergent estimates by adding a term that depends on the covariance matrix of the estimates. It is expected, rather than guaranteed, that this will promote input sequences that reduce the uncertainty on the estimates, hence reducing the size of the covariance and ultimately yielding a standard MPC cost once the true model is known. Closed-loop stability and constraint satisfaction are guaranteed by means of a robust invariant set for the pLDI structure (similar to [65]), and a Lyapunov type constraint on the first prediction step. The modification of the cost results in a non-convex problem which is addressed by either the inclusion of relaxation variables, or the separation of the problem into two optimizations. The second approach is discussed in more detail in [67], where the additional term in the cost function penalizes the deviation of the input trajectory from an optimal probing sequence. The latter is obtained through a preceding step of non-convex optimization that maximizes the minimum eigenvalue of the inverse of the information matrix as in [59].

A noticeably different architecture, yet also lacking a proper convergence guarantee for the estimator, is employed in [40]. The core idea is to decouple the control and performance objectives by maintaining two models of the plant. A nominal model is employed to characterize the parametric uncertainty and design a robustly stabilizing tube MPC controller, while a second model, initialized as the nominal model, is constantly updated by an estimator. Both are employed by the controller to make predictions; those from the first model are used to ensure robust constraint satisfaction, while those from the second are used to compute the cost. In this way robust stability is maintained even if the estimates render the initial tube controller infeasible, but performance is possibly improved by using a more accurate model for evaluating the cost.

A similar approach is presented in [68], where nonlinear systems are studied. A machine-learning approach is used to obtain, off-line, a nominal model of the plant given some previous data. This is accompanied by an estimation of a Hölder constant, which allows to compute a bound on the prediction error associated to this nominal model. This bound depends on the length of the prediction, and is used to properly tighten the constraint sets in an otherwise standard MPC optimization, in order to guarantee constraint satisfaction and input-to-state stability with respect to the prediction error. This is in similar fashion to the robust MPC approach depicted in [23], but without parametrizing the control action. It is then proposed to use the closed-loop data to continuously obtain more accurate models, however the different MPC elements used to guarantee recursive feasibility of the optimization have already been computed for the model obtained off-line. This results in that newly obtained models can only be used to evaluate the cost, while the off-line obtained one continues to be used to guarantee constraint satisfaction.

In a similar fashion, the controller proposed in [69] attempts to decouple the goals of control and system identification in order to provide a solution to the dual control problem for uncertain LTI systems. The main drawback is that both tasks are executed separately, employing the whole input capabilities, thus open-loop stability is a required assumption. A technique known as zone-tracking MPC [70, 71] is employed to steer the uncertain system towards an invariant set for identification, or more simply a set that is robustly invariant against model mismatch and persistently exciting inputs. While inside, a previously defined persistently exciting sequence can be implemented, for as long as necessary, in order to accurately estimate the model parameters. Attractivity of the target set for identification is guaranteed for the nominal model of the plant; although open-loop stability guarantees convergence of the true plant to the set, only nominal trajectories are shown to be constraint admissible, thus the mismatch may result in constraint violation. This issue is dealt with in [72], by employing a robust MPC formulation, but several other drawbacks remain, such as the requirement for open-loop stability.

The concept of homothetic tube MPC is coupled with a set-membership identification algorithm in [73] to produce an adaptive MPC scheme that guarantees constraint satisfaction and practical stability of the control target. The set-membership identification approach, guarantees that the estimated set that contains the true parameters is non-increasing in size. This allows the homothetic tube MPC, which is also a set based MPC controller, to guarantee recursive feasibility of the optimization problem (and hence recursive constraint satisfaction). As opposed to standard tube MPC, the homothetic approach reduces conservativeness as it updates the tightening on the constraint sets at each time instant and throughout the prediction horizon. This update takes into account the new information provided by the estimator, and also the fact that the parametric uncertainty decreases in size as the state and input trajectories approach the origin. Nevertheless, practical stability is achieved, by modifying the standard MPC optimization problem into a min-max problem, and the set-membership approach only guarantees a non-increasing set that contains the true parameters. Furthermore, this implies that the proposed approach is not applicable for uncertain time-varying plants.

3.1.2 AMPC with estimation guarantees

The main motivation for designing and implementing an AMPC algorithm is to address uncertainties in the model and changes in the plant while the latter is in operation, thereby improving the performance of a predictive controller without incurring expensive or prohibitively long down times. Therefore, it is necessary to guarantee that the algorithm set forth to obtain the estimates will effectively yield a more accurate representation of the plant (provided it is allowed by the model structure selected [3]). Furthermore, particularly relevant to robust AMPC approaches is to compute an a-priori bound on the prediction error, ultimately meaning that a valid estimate ought to lie inside a bounded set of models. Guaranteeing convergence, uniqueness and boundedness of the estimates is not a trivial task, especially since the on-line nature of the process implies that closed-loop system identification is to be performed. Indeed, it is easily shown that many parameter estimation algorithms may result in biased and non-convergent estimates if the system is in closed-loop or not properly excited [3, 74].

Selecting and properly implementing a particular identification algorithm is fundamental for the success of any AMPC controller. Given its computational tractability, similarity to the Kalman filter [1] and inherent receding implementation, a recursive least square (RLS) estimation algorithm is the choice in several AMPC controllers [53, 65, 69, 74]. Nevertheless, the successful identification of an accurate description of the plant via an RLS algorithm relies on the input/output data being sufficiently rich, equivalently it requires data produced by a persistently excited system [3, 75, 76]. In the most general sense, an input signal (sequence) is said to persistently excite a system if the generated input/output data yields, via a certain parameter estimation algorithm, a unique model [3] within the considered model structure. However, many equivalent definitions exist [3, 75, 76].

The first attempt to guarantee persistence of excitation (PE) within an MPC set-up is, probably, the one presented in [77]. Input constrained FIR models are considered, and an additional constraint is added to an otherwise standard MPC optimization problem in order to force the predicted input sequence to persistently excite the system [75]. Although several interpretations exist, the PE constraint can be translated into a positive autocovariance demand, and thus into a positive definiteness requirement, which is a non-convex constraint. The MPC optimization problem is usually formulated as a convex quadratic programming (QP) problem subject to linear convex constraints, therefore the addition of a non-convex constraint considerably increases the complexity of the optimization. In order to tackle the non-convexity, a relaxation variable is introduced in [77]. Since PE is sought over regulation, a periodic terminal constraint is proposed instead of the usual equality constraint. This denies the usual approach to guaranteeing stability however, to contextualize this approach, the usual stability arguments employed in MPC were being developed at the time.

This approach is extended to SISO-ARX systems in [44] and to MIMO-ARX systems in [78]. The ARX structure results in that the RLS regressor includes the output, and hence a standard PE constraint would also. Since the only explicit optimization variable in MPC is the input, output reachability arguments are employed in [44] to avoid using the full regressor vector in the PE analysis. In [78] the PE constraint is posed in the frequency domain and readily replaced by a collection of reverse convex constraints. In [44, 77, 78], however, a standard (nominal) MPC is proposed for the control task, hence the prediction uncertainty arising from model mismatch is not accounted for, resulting in the lack of stability or feasibility guarantees.

An interesting additional development is proposed in [74] where the PE constraint is enforced only over the first element of the predicted input sequence, while the required time window is completed by looking backwards into past control actions. This conforms with the receding horizon fashion of MPC, in which only the first action of the optimized input sequence is ever applied to the plant. In fact, even without a relaxation variable, it could happen that the true sequence of inputs in [77] is not persistently exciting because the optimization relies on the predictions to fulfil the related constraint. This choice also allows to reformulate the PE constraint as a single non-convex constraint which represents the outside of an ellipsoid (in the appropriate dimension). A recursive feasibility proof based on periodicity is provided for the PE constraint, however it is not fully discussed whether this remains valid under non-periodic input sequences; furthermore, no guarantee of constraint satisfaction is provided and closed-loop stability is a standing assumption.

An alternative approach is proposed in [79–83] where a two-step optimization procedure is devised to solve the dual control problem for constrained linear time invariant systems. The first step solves a standard (nominal) MPC optimization problem, possibly subject to constraints, in order to obtain an optimal sequence of predicted input actions. The second step modifies the previously obtained optimum by maximizing the minimum eigenvalue of the information matrix related to the predicted data. This is nothing more than another interpretation of the PE condition. The second optimization is constrained by an arbitrarily defined allowable cost increase, in order to bound the optimality decrease.

The main difference between [79,80] and [81–83] is that the latter explicitly consider the receding horizon fashion of MPC when computing the probing action. This is done by either modifying only the first element of the optimal control sequence obtained by the first step of optimization or by modifying more than one and using a semi-receding horizon approach in order to apply the entirety of the probing sequence. Although constraints may be considered, the model mismatch is not addressed structurally, thus constraint satisfaction while the estimates are not accurate is not guaranteed.

Different authors tackle different challenges of the dual control problem, and the same is valid in the MPC context. In [43,84–86] the focus is placed upon the challenges related to the parametric uncertainty and the estimation algorithms. In [43] a standard MPC controller (without stability guarantees) is augmented with an additional constraint designed to maintain the performance degradation due to the model mismatch within a certain acceptable bound. Similar concepts are implemented in the context of stochastic MPC for nonlinear systems in [84]. The approach taken in [85] for SISO-ARX systems differs in many aspects, although it follows the same principle. The excitation in [85] is enforced by a modification of the MPC cost in order to minimize the covariance of the estimates, as in [58]. A sensitivity analysis shows that enforcing the exciting constraint over more than one time step in the prediction horizon produces negligible effects on the closed-loop performance, which is explained by the receding horizon strategy of MPC. Similar ideas are employed in [86] but from an stochastic probabilistic approach that results in less unnecessary excitation of the system. This allows to account for the uncertainty in the optimization and therefore produce probabilistic feasibility guarantees (chance constraints). Apart from [86], constraint satisfaction is not studied in these approaches, neither is the stability of the closed-loop.

3.1.3 Persistently exciting tube MPC

A new and simple robust-based solution to the dual control problem within the MPC framework is proposed in this chapter. The focus is placed on the regulation of plants for which a model structure is known to be (2.1) but the value of its parameters is uncertain and may experience changes throughout its operation. The latter means that $(A, B) = (A(t), B(t))$, but the time dependency is dropped almost everywhere in order to simplify notation. The uncertainty in the model parameters could have many sources, such as mass product variability [63] and changes in operation or environmental conditions (for example variable payloads or operating temperatures). On the other hand, all plants are bound to experience changes during its running time, most likely due to degradation of mechanical parts or small modifications introduced purposefully (for example the stiffness loss of certain supports due to extreme events or the replacement of an actuator). In general terms then, the systems under study are assumed to be time-varying, however not necessarily in any parameter-affine structure such as in [87] for example. Furthermore, it is assumed that the changes are not instantaneously known or measurable, such as in the case of many switching systems controllers, and that the plant is subject to constraints such as those in (2.2).

The proposed AMPC algorithm follows a reasoning similar to [40,69] in that the objectives of the dual control problem are decoupled, albeit in an inherently different way. The algorithm presented here, henceforth called persistently exciting tube MPC (PETMPC), attains guaranteed properties for the control and estimation objectives by allocating a certain portion of the system's input for each task. Furthermore, in order to obtain the aforementioned guarantees with relatively low levels of complexity (both computational and in design), it is determined that the feedback laws that guide each task should be independent of each other.

With this general set-up established, consider system (2.1a) but neglecting external disturbances for simplicity of exposition (they can be added later without additional algebraic complexity); the pivotal feature of the proposed controller is the separation of the input $u(t)$ into a controlling part $\hat{u}(t)$ and an exciting part $\hat{w}(t)$, resulting in the following state-input dynamics

$$x(t+1) = Ax(t) + B\hat{u}(t) + B\hat{w}(t).$$

Furthermore, it is assumed that the the true dynamics, although uncertain and possibly changing, lie at all time instants within a compact set $\mathcal{M}$. If constraints (2.2) are assumed to be satisfied, the effect of the uncertainty can be lumped into a single disturbing term acting over a nominal representation of the plant $\left(\bar{A}, \bar{B}\right) \in \mathcal{M}$, i.e.

$$x(t+1) = \bar{A}x(t) + \bar{B}u(t) + \underbrace{\left(A - \bar{A}\right)x(t) + \left(B - \bar{B}\right)u(t)}_{w_p(t)}.$$

with $w_p(t) \in \mathbb{W}_p = \mathbb{W}_p\left(\mathcal{M}, \mathbb{X}, \mathbb{U}, \bar{A}, \bar{B}\right)$. Note that the true dynamics may change in time, however the nominal dynamics $\left(\bar{A}, \bar{B}\right)$ are chosen by the user (possibly in order to decrease the size of $\mathbb{W}_p$) and remain fixed until a valid better estimate has been found.

Section 3.2 of this Chapter introduces the estimation algorithm employed to find a better estimate for the model parameters and discusses the requirements over $\hat{w}(t)$ and (A, B) in order to guarantee convergent estimates independent of $\hat{u}(t)$ and initial conditions. Similarly, Section 3.3 proposes a tube-based robust MPC controller in order to drive the controlling part $\hat{u}(t)$. The goal is to guarantee constraint satisfaction and robust stabilizability of the closed-loop despite the modelling uncertainties $\mathbb{W}_p$, and the excitation induced by $\hat{w}(t)$. Finally, Section 3.4 presents a novel MPC-like receding horizon optimization to drive $\hat{w}(t)$ and fulfil the requirements set in Section 3.2. Although not necessary, the rationale behind the proposed optimization is to minimize the disturbing impact of the excitation part. A receding horizon MPC-like optimization attains this by: firstly enforcing the control objective (MPC cost function) upon the exciting part $\hat{w}(t)$, albeit independently of $\hat{u}(t)$; secondly, by introducing feedback through the receding horizon implementation. Section 3.5 provides a detailed description of the verification process required to guarantee that updating the prediction model will not break the tube MPC stabilizability and feasibility properties. Finally, a numerical example is provided in Section 3.6 to showcase the performance of the proposed AMPC controller and the future work avenues that remain open to improve the proposed algorithm.

The control properties discussed in Section 3.3 are robust to the initially estimated size of the uncertainty $\mathbb{W}_p$, therefore they remain robust to future changes as long as they are bounded to lie within the estimated range. This is in contrast to other AMPC algorithms that rely on a constant decrease on the error bound to guarantee asymptotic stability of the origin such as [60, 63, 64]. Furthermore, the proposed algorithm guarantees robust stabilizability for open-loop unstable plants, as opposed to [43,63,69,85,86]. The MPC-like optimization proposed to drive the exciting part also guarantees the convergence of the estimates, provided the plant changes slowly, unlike [39,64,65]. Finally, although the arguments used to ensure convergence of the estimates follow the lines of those provided in [74, 77], the decoupling of the objectives results in an standard convex QP problem for the control objective, allowing the non-convexity of the PE conditions to be dealt with separately.

The main drawback of the proposed approach is that, in order to maintain all the guarantees provided, the current estimates have to undergo a verification step before being used to update the MPC prediction model, as opposed to [59, 60, 66] where the update is done instantly. The advantage, however, is that PETMPC does not require the on-line re-computation of controller elements in order to provide a feasible and stabilizing solution (such as PI sets in [60] and stabilizing linear feedbacks in [59]). Section 3.5 discusses the conditions that need to be verified before an update of the prediction model is allowed. It may result that the current estimates require a full re-design of the controller in order to be deployed as a prediction model, however it is shown that this does not require one to halt operation. In summary, the adaptive control approach proposed in this paper provides control and excitation guarantees alongside with a simple design procedure, but at the price of not necessarily being able to update the prediction model with the true plant parameters once these have been obtained.

3.2 RLS and persistence of excitation

3.2.1 Recursive least squares algorithm

The PETMPC algorithm is proposed to control uncertain linear systems that can be accurately represented by (2.1) therefore a standard linear RLS algorithm fits the parameter estimation requirements while keeping the computational demands low (the matrix inversion associated to RLS algorithms can also be relaxed [75]). This is especially important given that in AMPC the new estimates have to be obtained right after measurement acquisition and before the optimization step of the MPC controller. This is in order to obtain a more accurate prediction model before making the predictions, thus the overall computational-time required at each step increases. An important thing to note is that no constraints are considered for the RLS algorithm, meaning that the estimates could eventually lie outside of the initially computed valid bound $\mathcal{M}$. This becomes an obstacle if the estimates are used to instantaneously update the MPC prediction model, however Section 3.5 discusses how the control properties of the PETMPC rely on a verification step on the estimates before updating. This, alongside with the convergence guarantees that are developed for the estimates in Section 3.2 render RLS constraints unnecessary.

It is also important to note that the proposed setting diverges from the classical framework in system identification problems. First of all, it is assumed that a full measurement of the state is available at each time instant. Secondly, no process noise is considered and measurement error is entirely neglected. This results in an inherently deterministic approach, which paves the way for the control related guarantees that the proposed robust dual controller enjoys. Note however that a mock of process noise can be considered within the tube MPC framework (in a similar way to [88] for robust output-feedback MPC), however the set-up must remain deterministic for its properties to hold. Furthermore, although the plant may experience changes, it will be considered time invariant by the recursive estimation algorithm. This is in alignment with the expected changes to be slow in nature, allowing for the forgetting capabilities of the RLS algorithm to account for them.

Given the above considerations, define the following 1-step ahead predictor for the state-input model in (2.1a)

$$\hat{x}^\top(t) = \varphi^\top(t-1)\hat{\theta}(t-1) \in \mathbb{R}^{1\times n} \tag{3.1a}$$

$$\varphi^\top(t) = \left[x^\top(t)\ \ u^\top(t)\right] \qquad \in \mathbb{R}^{n+m} \tag{3.1b}$$

$$\hat{\theta}(t) = [\mathcal{A}(t)\ \ \mathcal{B}(t)]^\top \qquad \in \mathbb{R}^{(n+m)\times n}, \tag{3.1c}$$

where $\mathcal{A}(t) \in \mathbb{R}^{n\times n}$ and $\mathcal{B}(t) \in \mathbb{R}^{n\times m}$ are the parameter estimates of (A, B) at time t (element-wise). The estimate $\hat{\theta}(t)$ is computed at each time instant following a standard recursion that minimizes the squared error of the 1-step ahead predictions [3]

$$\hat{\theta}(t) = \hat{\theta}(t-1) + E(t)^{-1}\varphi(t-1)\left[x^\top(t) - \varphi^\top(t-1)\hat{\theta}(t-1)\right] \tag{3.2a}$$

$$E(t) = \lambda E(t-1) + \varphi(t-1)\varphi^\top(t-1). \tag{3.2b}$$

Initialization of the recursion in (3.2) requires the definition of $\hat{\theta}(0)$ and $E(0)$. The former can be set to the currently known values of the plant parameters, possibly obtained from a previous identification experiment or first-principle approaches. The latter can be shown to be inversely proportional to the covariance matrix of the parameter estimates, thus a standard initialization is to set $E(0) = \delta \boldsymbol{I}_{n+m}$ with δ proportional to the confidence on the accuracy of $\hat{\theta}(0)$. Finally, λ is an arbitrarily defined forgetting factor whose purpose is to decrease the relevance of older data in the computation of the current estimates. This is relevant since the assumption is that the controlled system may undergo changes in the future, thus old data needs to weight less in the overall estimation scheme.

3.2.2 Convergence of the estimates

The recursive algorithm (3.2) provides a new value of the estimates at each time instant, however without any further consideration it is not readily guaranteed that the new values are ever going to reach the true plant parameters. Nonetheless, the recursion in (3.2) can be guaranteed to converge to a unique and unbiased estimate through a proper excitation of the system. In general terms it is said that the estimates (asymptotically) converge if $\hat{\theta}(t) - \hat{\theta}(t-1) \to 0$ as $t \to \infty$, furthermore it is said that the estimates converge to the true plant parameters θ if $\hat{\theta}(t) \to \theta$ as $t \to \infty$ with $\theta = [A \;\; B]^\top$.

A necessary condition for the true plant to be identifiable is that the model structure chosen contains the true plant structure [3]. This is the case by assumption, thus convergence of the RLS estimates to the unique true plant parameters requires the input-output data to be informative enough with respect to the model structure [3]. Informative enough is a purposely imprecise term because it depends heavily on the system structure and the type of experiments performed to obtain the input-output data (open or closed-loop). In the case of an RLS recursion, however, the concept of persistence of excitation (PE) is usually employed to characterise the information carried by the regressor.

The general notion of persistence of excitation, or persistently exciting sequences, has many equivalent definitions, however all of them arise from the necessity of guaranteeing the convergence of the estimates to a unique value. Ultimately, the goal here is to obtain a condition of excitation over the input partition $\hat{w}(t)$ that: (1) guarantees convergent estimates, (2) is verifiable at each time instant in a receding horizon fashion, and (3) it can be enforced through a constraint of an optimization problem.

3.2.2.1 Persistence of excitation

In [3] a frequency domain definition that is independent of the estimation algorithm is provided, albeit valid only for scalar signals. In what follows the backwards shift operator is denoted by q^{-1}.

Definition 3.1. A quasi-stationary scalar sequence $\vec{u}_t$ with spectrum $\Phi_u(\omega)$ is said to be persistently exciting of order n if, for all filters of the form $M_n(-q) = m_1 q^{-1} + \ldots + m_n q^{-n}$ and for all $-\pi < \omega \leq \pi$ the relation

$$|M_n(e^{j\omega})|^2 \Phi_u(\omega) = 0 \tag{3.3}$$

implies that $M_n(e^{i\omega}) = 0$.

Given Definition 3.1 it is fairly simple to derive a sufficient condition.

Lemma 3.1. A quasi-stationary scalar sequence $\vec{u}_t$ is said to be persistently exciting of order n if, its spectrum $\Phi_u(\omega)$ is non-zero for at least n different points in the interval $-\pi < \omega \leq \pi$.

Proof. Given that $m_0 = 0$, the polynomial $M_n(r)$ has one zero at the origin. This means that $M_n(r)$ has, at most, $n-1$ zeros *on* the unit circle (i.e. such that the zero can be represented as $r_0 = e^{j\omega_0}$). Hence, if $\Phi_u(\omega)$ is non-zero at at least n different points in the interval $-\pi < \omega \leq \pi$, then (3.3) implies $M_n(e^{i\omega}) = 0$. ■

In [3] general linear time invariant systems in open-loop are shown to generate convergent RLS estimates provided that the input to the system is persistently exciting of a certain order (following Definition 3.1). An example is also provided to demonstrate the non-uniqueness of the convergence when the data is generated in closed-loop operation. Note however that the spectrum of a sequence is defined in [3] as the Fourier Transform (FT) of its autocorrelation function. The latter is given by

$$R_u(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} u(t)u^\top(t-\tau), \tag{3.4}$$

therefore it is not possible to evaluate whether the regressor vector fulfils (3.3) at a certain time instant t, since the regressor sequence is finite.

A time domain definition of persistence of excitation, targeting least squares techinques, is provided in [75, 76].

Definition 3.2. A sequence $\vec{u}_t$ is called (weakly) persistently exciting of order n if the following matrix exists and is positive-definite (PD)

$$C_n = \begin{bmatrix} c(0) & c(1) & \cdots & c(n-1) \\ c(1) & c(0) & \cdots & c(n-2) \\ \vdots & \vdots & \ddots & \vdots \\ c(n-1) & c(n-2) & \cdots & c(0) \end{bmatrix} \tag{3.5}$$

where

$$c(\tau) = \lim_{N\to\infty} \frac{1}{N} \sum_{t=1}^{N} u(t)u^\top(t-\tau) \tag{3.6}$$

The parallel between Definitions 3.1 and 3.2 becomes obvious when comparing (3.4) and (3.6). In fact, it can be shown that both definitions are equivalent by noticing that for a certain input-output model $y(t) = M_n(-q)u(t)$,

$$R_y(0) = m^\top C_n m$$

with $m = [m_0\ m_1 \dots m_n]$; and also that, by definition, the inverse FT of the spectrum is equal to the correlation function, thus

$$m^\top C_n m = \frac{1}{2\pi}\int_{-\pi}^{\pi} M_n(e^{j\omega})\Phi_u(\omega)M_n^\top(e^{-j\omega})d\omega \tag{3.7}$$

Definition 3.2 demands C_n to be PD, equivalently $m^\top C_n m = 0 \implies m = 0$. Given (3.7), this can be readily rephrased as (3.3) (provided that $u(t)$ is a scalar).

Definition 3.2 still requires the computation of autocorrelations which is not possible due to the regressor vector being finite at time t. An alternative definition that provides a solution to this issue is also provided in [76] as well as in [75].

Lemma 3.2. A sequence $\vec{u}_t$ is called (strongly) persistently exciting of order n if for all t, there exists an integer l and scalars $\rho_1, \rho_2 > 0$ such that

$$\rho_2 I > \sum_{j=0}^{l-1}\left(\boldsymbol{u}_{t-j}\boldsymbol{u}_{t-j}^\top\right) > \rho_1 I \tag{3.8}$$

where,

$$\boldsymbol{u}_{t-j} = \begin{bmatrix} u(t-j) \\ u(t-j-1) \\ \vdots \\ u(t-j-n+1) \end{bmatrix} \tag{3.9}$$

Proof. The summation in (3.8) amounts to a portion, in time, of the summation in (3.6). Analogously, a time dependent C_n can be constructed

$$C_n = \lim_{N\to\infty}\frac{1}{N}\sum_{t=1}^{N} C_n(t),$$

with $C_n(t)$ defined as in (3.5) but with $c(\tau)$ replaced by the time dependent portions in (3.8). It follows then that if $C_n(t)$ is PD and bounded above by a constant value for all t then C_n exists and is PD. ■

Note that the definition in Lemma 3.2 is reported in a slightly different manner when compared to [75, 76]. The difference is in the position of the time window that is observed in order to compute the PE order. In (3.9) this window is placed so that the resulting vector $\boldsymbol{u}_{t-j}$ depends only on current and past time instances. This is done to simplify the inclusion of such a requirement on the receding horizon control framework that is to be employed. Furthermore, the upper limit in the summation is set to $l-1$ as opposed to l as in [75, 76] in order to simplify the recursive feasibility guarantees discussed in Section 3.4. Since l from [75,76] can be arbitrarily set to $l-1$ from (3.9) this does not affect the PE definition. Finally, it is important to note that, in order to meet (3.8) it is necessary that $l \geq n$.

Definition 3.2 is used in [75] to guarantee unbiased convergence of the RLS estimates of scalar FIR models when the input is (weakly) persistently exciting. The same is shown to hold for open-loop stable ARX models, however unstable plants may lead to invalid estimates due to an additional term in the LS minimization that weighs the initial value of the parameters $\hat{\theta}(0)$. Similar results are presented in [76] where the LS estimate is shown to be unique provided a certain matrix is non-singular. The parallel of this matrix in the RLS algorithm is $E(t)$ in (3.2), therefore Lemma 3.2 represents a suitable excitation condition over the input of scalar FIR models to guarantee convergent recursive estimates. The result in [76] however hints to a more general excitation condition to be placed over the regressors instead of simply over the inputs.

Other authors prefer to avoid endowing an order property when defining a persistently exciting signal. In [89] for example, the order is replaced by simply defining an auxiliary variable that contains n past elements of the sequence being analysed. In [90] the concept of sufficient richness is introduced to refer to a persistently exciting sequence of a certain required order. The results in [89,90] are particularly useful for the problem at hand since they are applicable to multi-variable open-loop unstable plants in state space form [89] and ARX form [90]. The concept of input-output reachability is employed in [89, 90] to guarantee that persistently exciting inputs will transfer its probing properties towards the regressor vectors. Moreover, it is shown in [90] that persistently exciting regressor vectors are sufficient to guarantee the convergence of certain recursive estimation schemes such as a RLS. In what follows, the results presented in [89, 90] are reproduced and employed to guarantee convergence of the recursion in (3.2) through an appropriate design of $\hat{w}(t)$ that follows Lemma 3.2

3.2.3 Persistence of excitation of the regressor vector

Convergence of the RLS estimates requires the regressor sequence to be persistently exciting of order 1 [90]. Although Lemma 3.2 provides a suitable definition to verify this at each time instant, it is not directly applicable to the regressor sequence (3.1a) in the proposed AMPC framework. Firstly, the regressor is formed by the full input and state vectors, however only $\hat{w}(t)$ is available for the controller to meet the PE requirements. Secondly, it is shown in Section 3.4 that the recursive feasibility of the overall AMPC controller hinges on including the PE condition over the prediction horizon. If the full regressor is employed then this results in a collection of non-linear constraints that couple the predicted states and inputs.

To devise a practical constraint that depends only on the exciting part of the input, and that can be included in an MPC-like receding horizon optimization problem, the subject of transmissibility of the PE condition is studied. Particularly, it is necessary to guarantee that the PE condition will transmit from $\hat{w}(t)$ to $u(t)$ and from the input to the regressor vector $\varphi(t)$.

3.2.3.1 Input-to-regressor PE transmissibility

The first objective is to guarantee that PE characteristics from the input are passed onto the regressor. Consider the following standard definitions and results.

Definition 3.3. System (2.1) with $E = \mathbf{0}$ is said to be output reachable (equivalently y is reachable from u), if for any y and arbitrary initial state, there exists an input sequence $\vec{u}_k$ with $0 \leq k < \infty$, such that $y(k) = y$. Equivalently, the system's output reachability matrix

$$M_o = \begin{bmatrix} D & CB & CAB \cdots CA^{n-1}B \end{bmatrix}$$

is of full row rank.

Definition 3.4. If $C = \boldsymbol{I}_n$ and $E, D = \mathbf{0}$, system (2.1) is said to be state reachable (equivalently x is reachable from u), if the state reachability matrix

$$M_s = \begin{bmatrix} B & AB \cdots A^{n-1}B \end{bmatrix}$$

is of full row rank.

Lemma 3.3. If $C = \boldsymbol{I}_n$ and $E, D = \mathbf{0}$, a system of the form (2.1) is said to be state unreachable if and only if the matrix $[z\boldsymbol{I}_n - A \;\; B]$ loses rank for some $z = \lambda$ where λ is an eigenvalue of A.

According to Definitions 3.3 and 3.4, the following PE results can be established for reachable systems [89].

Proposition 3.1 (Corollary 2.1 in [89]). A necessary and sufficient condition for the output of any output reachable time invariant linear system of McMillan degree n to be persistently exciting of order 1, independent of initial conditions, is that the input to the system is persistently exciting of order $n + 1$.

The proof to Proposition 3.1 can be found in [89]. A remark is provided in [89] to address systems with full state availability, but it is cast here as a Corollary to Proposition 3.1.

Corollary 3.1. If $C = \boldsymbol{I}_n$ and $E, D = \mathbf{0}$, the reachability requirement is reduced to that of state reachability, and the input is only required to be persistently exciting of order n.

Proof. If $C = \boldsymbol{I}_n$ and $D = \mathbf{0}$ there is no feed-forward to the current measured output (state), thus the reduction by 1 in the order requirement. ∎

Proposition 3.1 is a precise account of the result presented in [89] but rephrased to fit the PE definition given in Lemma 3.2. The following is also an exact depiction of a result in [89] but shaped to account for systems with full state measurement.

Lemma 3.4 (Corollary 3.2 in [89]). Consider the multivariable ARMA model

$$x(t) + \mathcal{A}_1 x(t-1) + \cdots + \mathcal{A}_{\bar{n}} x(t-\bar{n}) = \mathcal{B}_1 u(t-1) + \cdots + \mathcal{B}_{\bar{m}} u(t-\bar{m})$$

and the corresponding regressor vector

$$\varphi_{\bar{n}\bar{m}}(t) = \left[x^\top(t-1) \cdots x^\top(t-\bar{n}) \; u^\top(t-1) \cdots u^\top(t-\bar{m})\right]^\top.$$

Define the input-state transfer function as $\mathcal{T}(z)$ and assume it is proper. Moreover define

$$\mathcal{A}(z) = \left[\boldsymbol{I}_n z^{\bar{n}} \;\; \mathcal{A}_1 z^{\bar{n}-1} \cdots \mathcal{A}_{\bar{n}}\right]$$
$$\mathcal{B}(z) = \left[\mathcal{B}_1 z^{\bar{m}-1} \;\; \mathcal{B}_2 z^{\bar{m}-2} \cdots \mathcal{B}_{\bar{m}}\right]$$

and assume that $\mathcal{A}_{\bar{n}}$ and $\mathcal{B}_{\bar{m}}$ are not zero. The regressor vector $\varphi_{\bar{n}\bar{m}}(t)$ is reachable from $u(t)$ if and only if, $[\mathcal{A}(z) \mid \mathcal{B}(z)]$ is of full row rank for all z.

Lemma 3.4 and Corollary 3.1 give way to the following result that guarantees transmissibility of the persistence of excitation condition from the input to the regressor.

Theorem 3.1. Consider the system in (2.1) with $C = \boldsymbol{I}_n$, $E, D = \mathbf{0}$ and assume that the system is state reachable. The corresponding regressor vector (3.1a) is persistently exciting of order 1 if and only if the input to the system is persistently exciting of order $n + m$ according to Lemma 3.2.

Proof. The proof is based on Corollary 3.1, for which it is necessary to show that the regressor vector is reachable from the input. Given $C = \boldsymbol{I}_n$ and $E, D = \mathbf{0}$, the regressor vector (3.1a) meets the definition of Lemma 3.4 by setting

$$\begin{aligned} \bar{n} &= 1, & \mathcal{A}_1 &= A \\ \bar{m} &= 1, & \mathcal{B}_1 &= B \\ \mathcal{A}(z) &= z\boldsymbol{I}_n - A, & \mathcal{B}(z) &= B. \end{aligned}$$

It follows that $\mathcal{T}(z)$ is proper and that $[\mathcal{A}(z) \mid \mathcal{B}(z)] = [z\boldsymbol{I}_n - A \mid B]$. Lemma 3.3 and the reachability assumption on the system imply then that $[\mathcal{A}(z) \mid \mathcal{B}(z)]$ is of full row rank for all z. Thereby all the hypotheses of Lemma 3.4 are met and the corresponding regressor vector $\varphi(t) = \left[x^\top(t-1)\, u^\top(t-1)\right]^\top$ is reachable from the input $u(t)$. Once reachability has been established, the PE order is guaranteed by Corollary 3.1. In fact, consider the following minimal state space model

$$\varphi(t) = \begin{bmatrix} A & B \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \varphi(t-1) + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{I}_m \end{bmatrix} u(t), \tag{3.10}$$

where $u(t)$ is the input and the regressor vector $\varphi(t)$ is the state. The McMillan degree of (3.10) is $n + m$, hence since the input $u(t)$ is persistently exciting of order $n + m$, the regressor vector (output of (3.10)) is persistently exciting of order 1 by Corollary 3.1. ■

Theorem 3.1 guarantees that the regressor vector (3.1b) defined for system (2.1a) is persistently exciting of order 1 given a persistently exciting input of order $n+m$. This result, however, requires that no disturbances are affecting the system. Although external disturbances, if assumed white noise, are persistently exciting of infinite order, they excite the closed-loop thereby feeding biased information to the RLS algorithm. Furthermore, if $Bu(t) = -Ew(t)$ for any given time instant t then the probing capabilities of the input are hindered by the disturbance. In what follows it is assumed that the size of the possible disturbances is small enough to neglect both these effects.

3.2.3.2 Exciting part-to-input PE transmissibility

Theorem 3.1 establishes the necessary assumptions and provides the required design tools to guarantee that the regressor (3.1b) for system (2.1a) is persistently exciting, thus that the RLS estimates will converge. Particularly, it is required for the system's input $u(t)$ to be persistently exciting of order $n+m$. However in the proposed AMPC algorithm part of the input sequence, namely $\hat{u}(t)$, is governed by the tube MPC controller and hence not explicitly available for excitation purposes. It is not sufficient to impose the same PE demands over $\hat{w}(t)$, given that $\hat{u}(t)$ could cancel out the probing capabilities of $\hat{w}(t)$. This is due to the fact that both portions are designed independently. Particularly the robust control action achieves robust stabilizability and constraint satisfaction by regarding the excitation as a disturbance, hence it is inherently designed to reject it.

Nevertheless, given the characteristics of the tube MPC robust controller described in Section 2.3, the same reachability arguments established in Section 3.2.3.1 can be employed to guarantee PE of the regressor starting from PE of the exciting part $\hat{w}(t)$. Indeed, assume that a tube MPC controller is employed to drive the controlling part of the input, meaning that $\hat{u}(t) = \kappa_N(x(t)) = \bar{\kappa}_N(x(t)) + Ke(t) = \bar{u}(t) + K(x(t) - \bar{x}(t))$, it follows that

$$
\begin{aligned}
u(t) &= \hat{u}(t) + \hat{w}(t) \\
u(t) &= \bar{u}(t) + K\left(x(t) - \bar{x}(t)\right) + \hat{w}(t) \\
u(t) &= \left(\bar{u}(t) - K\bar{x}(t)\right) + \left(Kx(t) + \hat{w}(t)\right).
\end{aligned}
$$

Now, according to Propositions 2.1 and 2.2, independent of whether the nominal trajectories are optimized or not, the origin is exponentially stable for the undisturbed closed-loop dynamics, hence the quantity $\bar{u}(t) - K\bar{x}(t)$ converges to the origin exponentially fast. Consider now the following result from [90].

Lemma 3.5 (Lemma 4 in [90])**.** Suppose that $s(t)$ and $r(t)$ are two bounded sequences taking values in $\mathbb{R}^m$, and assume that $s(t)$ is persistently exciting of order n and that:

$$\lim_{t\to\infty} r(t) = 0$$

then $s(t) + r(t)$ is also persistently exciting of order n.

Lemma 3.5 establishes that if the controlling part of the input is defined by an exponentially convergent tube MPC controller, then the excitation properties of the overall input $u(t)$ depend purely on the excitation properties of $Kx(t) + \hat{w}(t)$. In view of this set $\hat{u}(t) = \hat{x}(t) = \mathbf{0}$, then the following input-output state space model can be constructed

$$x(t+1) = (A + BK)\, x(t) + B\hat{w}(t) \tag{3.11a}$$

$$u(t) = Kx(t) + \hat{w}(t). \tag{3.11b}$$

System (3.11) has $x(t)$ as the state, $\hat{w}(t)$ as the input, and $u(t)$ as the output. By analysing this auxiliary system, the matter of transmissibility of the PE conditions from $\hat{w}(t)$ to $u(t)$, reduces to analysing whether $u(t)$ is reachable from $\hat{w}(t)$ via the dynamics in (3.11).

Theorem 3.2 (Transmissibility of the persistence of excitation through the robust control action)**.** Consider the system in (2.1a) and assume $C = \boldsymbol{I}_n$, $E, D = \mathbf{0}$ and that the system is state reachable. Assume also that the controlling part of the input $\hat{u}(t)$ is driven by a robustly exponential stable tube MPC controller such as those described in Section 2.3. The corresponding regressor vector (3.1a) is persistently exciting of order 1 if and only if the exciting part of the input $w(t)$ is persistently exciting of order $2n + m + 1$.

Proof. The proof hinges on showing that the input $u(t)$ is persistently exciting of order $n + m$, which according to Theorem 3.1 is necessary and sufficient for the regressor (3.1a) to be persistently exciting of order 1. First note that the output reachability matrix for system (3.11) is

$$M_o = \begin{bmatrix} \hat{D} & \hat{C}\hat{B} & \hat{C}\hat{A}\hat{B} \cdots \hat{C}\hat{A}^{n-1}\hat{B} \end{bmatrix}$$

with $\hat{A} = A + BK \in \mathbb{R}^{n \times n}$, $\hat{B} = B \in \mathbb{R}^{n \times m}$, $\hat{C} = K \in \mathbb{R}^{m \times n}$ and $\hat{D} = \boldsymbol{I}_m$. Since, $\hat{D}$ has rank m, then M_o has full row rank, thus $u(t)$ is reachable from $\hat{w}(t)$. Moreover, the McMillan degree of the system (3.11) is n, therefore $\hat{w}(t)$ being persistently exciting of order $2n + m + 1$ is necessary and sufficient for $u(t)$ to be persistently exciting of order $n + m$ by Proposition 3.1. Finally, by Theorem 3.1, it follows that the regressor vector in (3.1a) is persistently exciting of order 1. ■

3.2.3.3 Time-varying systems

A shortcoming of Theorems 3.1 and 3.2 is that the system is required to be time-invariant due to Proposition 3.1, however the goal was to also account for slowly changing systems. A extension of these results to time-varying systems is discussed in [91]; in order to properly employ the results therein it is necessary to explicitly define the uncertainty on the model parameters. In order to do so let $\beta(t) \in \mathbb{R}^{n(n+m)}$ be a vector that contain each element of the state and input matrices, and that explicitly states the time-varying nature of the system through its time dependency. It follows that $A = A(\beta(t))$ and $B = B(\beta(t))$. Furthermore, suppose that $\beta(t)$ is constrained, at all time instances, to lie inside a compact set $\mathbb{B} \subseteq \mathbb{R}^{n(n+m)}$, which is no more than the set $\mathcal{M}$ represented in a different space.

Consider now the following results from [90, 91].

Lemma 3.6. Consider two sequences $v(t) \in \mathbb{R}^n$ and $\hat{v}(t) \in \mathbb{R}^n$ and suppose there exists constants $\varepsilon > 0$ and $l \in \mathbb{N}$ such that

$$\sum_{j=0}^{l-1} | \left(v(t-j) - \hat{v}^\top(t-j)\right) |_2^2 \leq \varepsilon.$$

If $\hat{v}(t)$ is persistently exciting of order n according to Lemma 3.2, then so is $v(t)$.

Lemma 3.6 establishes that, as long as the deviation between two sequences is bounded, then PE of one implies PE of the other. This property is used to ensure PE characteristics of a time-varying system by looking at the PE order of a suitable piece-wise constant approximation of it, thus the latter are first established. Define $f(\beta) = \xi_m(BB^\top)$, then the following holds.

Lemma 3.7. Consider a PWC system $\left(\hat{A}(\beta(t)), \hat{B}(\beta(t))\right)$ such that $\beta(t)$ remains constant over intervals $[t, t+l-1]$ with $l \in \mathbb{N}$. Assume there exists constants $l_1, l_2 > 0$ such that

$$\max_{\beta \in \mathbb{B}} ||A(\beta)|| \leq l_1$$
$$\min_{\beta \in \mathbb{B}} \xi_m \left(B(\beta)B^\top(\beta)\right) = l_2.$$

If the input $u(t)$ is persistently exciting of order n according to Lemma 3.2, then $x(t)$ is persistently exciting of order 1 according to Definition 3.2.

Lemma 3.7 guarantees PE of the state of a state reachable PWC system given a persistently exciting input of appropriate order. Furthermore, $\left(\hat{A}(\beta(t)), \hat{B}(\beta(t))\right)$ will be referred to as a reachable PWC (l, ε)-approximation of (A, B) if it fulfils the conditions of Lemma 3.7 and if for any arbitrary input sequence it holds that for all t

$$\sum_{j=0}^{l-1} | \left(x(t-j) - \hat{x}^\top(t-j)\right) |_2^2 \leq \varepsilon, \tag{3.12}$$

where $x(t)$ and $\hat{x}(t)$ are correspondingly the state trajectories of (A, B) and $\left(\hat{A}(\beta(t)), \hat{B}(\beta(t))\right)$, and l is from Lemma 3.7. The PE characteristics for the output of time-varying systems can now be established.

Lemma 3.8. Let a reachable PWC (l, ε)-approximation of (A, B) exist. If $u(t)$ is persistently exciting of order n according to Lemma 3.2 with l from Lemma 3.7, then the state $x(t)$ is PE of order 1 according to Definition 3.2, provided that $\rho_1 > {}^{2\varepsilon l_1^2}/_{l_2}$ where l_1 and l_2 are from Lemma 3.7 and ε is from (3.12).

Lemma 3.8 implies that given a large enough lower bound in Lemma 3.2, persistently exciting inputs guarantee persistently exciting outputs even if the system is time-varying. This results can be analogously extended to guarantee excitation of the regressor vector given a persistently exciting sequence $\hat{w}(t)$, provided that the lower bound ρ_1 for $\hat{w}(t)$ in Lemma 3.2 is large enough. Note that l_1 and l_2 depend on $\mathbb{B}$ (or $\mathcal{M}$) and thus are defined by the particular plant, ε depends on l according to (3.12) and hence can be set by the designer. It may seem intuitive that, given that the exciting part of the input is bounded to $\hat{\mathbb{W}}$, the best option is to choose a small l in order to obtain a small ε according to (3.12), and thereby a smaller lower bound for ρ_1 needed to guarantee PE transmissibility. However, note that l has to be equal to, or greater than the order of PE sought, otherwise it is not possible to meet constraint (3.5). Furthermore, a shorter time window l also makes it harder to guarantee (3.5) given that the exciting part of the input is bounded inside $\hat{\mathbb{W}}$.

From the above discussion it is straightforward that: (1) there is a lower bound to l, (2) the relationship between l and the minimum ρ_1 required to guarantee transmissibility is non-decreasing since the left-hand-side of (3.12) is a norm, and that (3) the relation between l and the maximum ρ_1 that allows for (3.5) to be met is non-decreasing. It follows then that existence of l that fulfils all requirements is not guaranteed and depends on the uncertainty $\mathcal{M}$

which defines ε according to (3.12). Furthermore, if such an l exists, the most likely case is that it is not unique. In such a case l is chosen to be as close as possible to its lower bound, in order to reduce the computational complexity of the PE related optimization (see Section 3.4).

Theorems 3.1 and 3.2 provide the necessary tools to design the exciting part of the input $\hat{w}(t)$ in a such a way that it guarantees uniqueness and convergence of the RLS recursion (3.2). In summary, it is necessary to design $\hat{w}(t)$ such that it is persistently exciting of order $2n + m + 1$ where n is the number of states and m is the number of inputs of the system (Theorem 3.2). In this way the excitation is transmitted to the regressor vector through the control part of the input and through the systems dynamics. Moreover, the results in Lemma 3.8 allow to extend Theorem 3.2 to account for systems that are varying in time, either continuously or in a piece-wise nature. However, all of the above require the system (A, B) to be state reachable. Although testing for reachability amounts to a rank verification, the issue is that the true system is assumed uncertain, and even if the estimates converge at some time instant, the plant may experience changes in the future, thereby requiring reachability as an standing assumption.

Assumption 3.1. The true plant is state reachable at every time instant.

3.3 Tube-based MPC

The proposed architecture for the PETMPC is based on the partition of the input. Section 3.2 describes the requirements over the exciting part in order to achieve convergent estimates independently of the regulatory part $\hat{u}(t)$. Similarly, it is now proposed to implement a standard tube MPC controller with independently evolving trajectories (Section 2.3.2) in order to robustly regulate the plant despite the presence of purposeful excitation $\hat{w}(t)$ and parametric uncertainty.

Before setting up the optimization problem related to the tube MPC controller it is necessary to define the appropriate input constrains. The overall input is subject to constraint (2.2b), notwithstanding the partition then, it must happen that $\hat{u}(t) + \hat{w}(t) \in \mathbb{U}$ at all times. To maintain the independency between the input partitions set $0 < \alpha < 1$ and define

$$\hat{\mathbb{U}} = \alpha \mathbb{U} \tag{3.13a}$$

$$\hat{\mathbb{W}} = \mathbb{U} \ominus \hat{\mathbb{U}} = (1 - \alpha)\mathbb{U}. \tag{3.13b}$$

It follows that if $\hat{u}(t) \in \hat{\mathbb{U}}$ and $\hat{w}(t) \in \hat{\mathbb{W}}$ at all times, then $u(t) \in \mathbb{U}$. The sets $\hat{\mathbb{U}}$ and $\hat{\mathbb{W}}$ are independent scaling based partition of the input constraint set, and although many different division schemes could be used, this one allows to maintain the properties of the source set $\mathbb{U}$. In this context, another option for partitioning the input set would be to arbitrarily define $\hat{\mathbb{W}} \subset \mathbb{U}$ and then set $\hat{\mathbb{U}} = \mathbb{U} \ominus \hat{\mathbb{W}}$, however this might result in a portion of the input not being used since $\hat{\mathbb{U}} \oplus \hat{\mathbb{W}}$ would not necessarily be equal to $\mathbb{U}$.

Given (3.13a) then, it holds that both $\hat{\mathbb{U}}$ and $\hat{\mathbb{W}}$ are $\mathcal{PC}$-sets. The selection of α may seem a trivial step in the design of the proposed controller, but it represents, almost on its own, the trade-off between both objectives of the dual control problem. Indeed, it might seem intuitive to favour smaller values of α to increase the RoA of the MPC controller; however, a small set $\hat{\mathbb{W}}$ implies that the exciting sequence $\hat{w}(t)$ might not be able to meet the lower bound requirements in Lemma 3.8.

3.3.1 Admissible RPI set for tightening

After the partition is executed, the dynamics of the uncertain system can be recast as

$$x(t+1) = Ax(t) + B\hat{u}(t) + B\hat{w}(t). \tag{3.14}$$

where the pair (A, B) is contained in the compact set $\mathcal{M}$ and the state and input constraints $\mathbb{X}$ and $\hat{\mathbb{U}}$ are in effect. In order to employ a tube MPC controller to robustly control such a system it is necessary to define: (1) a nominal prediction model that does not require knowledge of the exciting part $\hat{w}(t)$ or exact knowledge of (A, B), and (2) an admissible RPI set for the associated error dynamics given the standard composite tube control law and constraints $\mathbb{X}$ and $\hat{\mathbb{U}}$. The latter requires the computation of a linear gain that stabilizes the error dynamics.

In Section 3.1.3 it was anticipated that the uncertainty on the values of (A, B) can be dealt with by choosing an arbitrary model $\left(\bar{A}, \bar{B}\right) \in \mathcal{M}$ to represent the plant's dynamics while regarding the mismatch due to parametric uncertainty as a disturbance w_p contained in the set $\mathbb{W}_p\left(\mathcal{M}, \mathbb{X}, \mathbb{U}, \bar{A}, \bar{B}\right)$. In view of this the plant's dynamics can be recast as

$$x(t+1) = \bar{A}x(t) + \bar{B}\hat{u}(t) + w_p(t) + \bar{B}\hat{w}(t), \tag{3.15}$$

with an obvious selection of nominal prediction model for the MPC optimization

$$\bar{x}(t+1) = \bar{A}\bar{x}(t) + \bar{B}\bar{u}(t). \tag{3.16}$$

This approach, however, might be unnecessarily conservative since the constraint sets are employed in their entirety to compute $\mathbb{W}_p$, yet for any given pair (x, u) it follows that $w_p = \left(A - \bar{A}\right) x + \left(B - \bar{B}\right) u$. This prompts the question of whether a more constructive consideration of the set $\mathcal{M}$ can lead to a considerable decrease in conservatism when computing the RPI set associated to the tube MPC implementation.

To answer this question note that, although the true values of (A, B) are unknown, they are bound to be inside the compact set $\mathcal{M}$ at any time instant. Assume that $\mathcal{M}$ is (or can be outer approximated by) a polyhedron with vertices $m_i = (A_i, B_i)$ for $i \in [1, \dots, M]$, say $\tilde{\mathcal{M}}$, and define $\bar{w} = B\hat{w}(t) \in \bar{\mathbb{W}}$ where $\bar{\mathbb{W}}$ is a function of $\hat{\mathbb{W}}$ and $\tilde{\mathcal{M}}$ (since B is unknown). It follows that the uncertain system (3.14) can be recast as disturbed polytopic linear difference inclusion (pLDI)

$$x(t+1) = \underbrace{\sum_{i=1}^{M} \lambda_i(t) A_i}_{A} x(t) + \underbrace{\sum_{i=1}^{M} \lambda_i(t) B_i}_{B} \hat{u}(t) + \bar{w}(t),$$

with $\lambda_i(t) \geq 0$ for all $i \in [1, \dots, M]$ and $t \geq 0$ such that

$$\sum_{i=1}^{M} \lambda_i(t) = 1.$$

It may seem appropriate then to take this pLDI framework explicitly into account to compute a robust invariant set for $(A + BK)$ when disturbed by perturbations bounded in $\bar{\mathbb{W}}$. It is only reasonable to expect that, by doing so, a less conservative RPI set will be obtained when compared to what would emanate from the lumped uncertainty approach depicted by (3.15).

It is usual in tube MPC that, given the composite control law, the error dynamics are driven by $A_K = (A + BK)$ with K being a stabilizing gain for (A, B) (see (2.3)). In this case, as opposed to Chapter 2, A_K is not time invariant, nevertheless several approaches exist to compute a single K that stabilizies the entirety of models in $\bar{\mathcal{M}}$, such as the LMI approach proposed in [26] or the relaxation in [92]. A similar procedure is presented in [93], where an admissible PI set is computed alongside the corresponding linear gain K. Given an appropriate stabilizing gain there exists several approaches to compute invariant sets for pLDI systems [93–95] and in particular to compute RPI sets that are minimal (or approximations of it that are invariant) [94, 95]. Once a stabilizing gain K and associated RPI set $\mathbb{S}$ have been computed it follows that $(A_i + B_i K)\,\mathbb{S} \oplus \bar{\mathbb{W}} \subseteq \mathbb{S}$ for all $i \in [1, \dots, M]$ and, by convexity of $\tilde{\mathcal{M}}$, $(A + BK)\,\mathbb{S} \oplus \bar{\mathbb{W}} \subseteq \mathbb{S}$ for all $(A, B) \in \mathcal{M}$.

There is, however, an obstacle in using such an RPI set for tightening in a standard tube MPC approach. Indeed, as described in Chapter 2, tube MPC requires an RPI set for the dynamics of the error between the true trajectories and those predicted by the prediction model, but selecting the latter is not trivial. If the prediction model is set to

$$\bar{x}(t+1) = A\bar{x}(t) + B\bar{u}(t) \tag{3.17}$$

it follows that, with $e(t) = x(t) - \bar{x}(t)$ and $u(t) = \bar{u}(t) + K\left(x(t) - \bar{x}(t)\right)$, the error dynamics are given by

$$e(t+1) = (A + BK)\, e(t) + \bar{w}(t),$$

and the RPI set computed for the pLDI closed-loop is indeed an RPI set for the error dynamics. However, (3.17) cannot be employed as a prediction model because the pair (A, B) is unknown. If (3.16) is employed as a prediction model it follows that

$$e(t+1) = (A + BK)\, e(t) + \bar{w}(t) + \bar{w}_p(t), \tag{3.18}$$

with $\bar{w}_p(t) = \left(A - \bar{A}\right)\bar{x}(t) + \left(B - \bar{B}\right)\bar{u}(t)$ and so $\mathbb{S}$, computed as an RPI set for the pLDI, is not RPI for the error dynamics. This is the reason why in [65, 66] the invariant sets (controlled and positive) computed for the pLDI systems considered therein are not employed for tightening, but directly as a constraint over the MPC optimization.

Still, an RPI set for the pLDI system could be computed to account for both disturbance $\bar{w}$ and $\bar{w}_p$, but the latter depends on the nominal prediction variables, and so any bounding set would depend on $\bar{\mathbb{X}}$ and $\bar{\mathbb{U}}$. The tightened sets, however, are usually defined as a function of the RPI set to increase the RoA of the resulting controller (see (2.4)) which makes the whole design procedure intertwined. A workaround would be to arbitrarily define the tightened constraint sets as a scaling of the true constraint sets, say

$$\bar{\mathbb{X}} = \alpha_x \mathbb{X}$$

$$\bar{\mathbb{U}} = \alpha_u \mathbb{U}.$$

with $\alpha_x, \alpha_u \in (0, 1)$ and then to compute $\bar{\mathbb{W}}_p$ to contain $\bar{w}_p$ at all times (in a similar way as to how $\mathbb{W}_p$ would be computed). After that an RPI set $\bar{\mathbb{S}}$ for the pLDI (3.18) could be computed, and used to verify constraint satisfaction only if

$$\alpha_x \mathbb{X} \oplus \bar{\mathbb{S}} \subseteq \mathbb{X} \tag{3.19a}$$

$$\alpha_u \mathbb{U} \oplus K\bar{\mathbb{S}} \subseteq \mathbb{U}. \tag{3.19b}$$

If not, the whole process needs to be repeated with different scaling factors α_x, α_u until (3.19) are verified. Although it is expected that the necessary inclusions will be met as $(\alpha_x, \alpha_u) \to (0, 0)$ this design approach does not necessarily produce considerably less conservative RPI sets for the error dynamics since the set $\bar{\mathbb{W}}_p$ would still depend on a nominal model chosen arbitrarily as a prediction model for the MPC optimization. Furthermore, by defining the tightened constraint sets as a scale of the true ones, as opposed to (2.4), it is most likely that the overal RoA of the controller will suffer some shrinkage, even if the obtained RPI set is indeed less conservative.

Since it is not clear whether the explicit consideration of a pLDI structure is advantageous, it is proposed to move forward with the lumped uncertainty approach proposed in Section 3.1.3 and characterized by the model (3.15). The latter can be further simplified to

$$x(t+1) = \bar{A}x(t) + \bar{B}\hat{u}(t) + w(t), \tag{3.20}$$

where the term $w(t)$ lumps all the effects that the controller will regard as external and uncontrollable disturbances, that is

$$w(t) = w_p(t) + \bar{B}\hat{w}(t) \in \mathbb{W}_p \oplus \bar{B}\hat{\mathbb{W}} = \mathbb{W}. \tag{3.21}$$

By assumption $\mathbb{X}$, $\mathbb{U}$ are $\mathcal{PC}$-sets and $\mathcal{M}$ is a compact set, thus the set $\mathbb{W}_p$ can be constructed as a $\mathcal{C}$-set. Nevertheless, since $\mathcal{M}$ is not necessarily convex (just compact) it follows that a certain degree of conservatism is introduced in order to make $\mathbb{W}_p$ convex. Finally, if $\mathbb{W}_p$ is a $\mathcal{C}$-set, it follows that $\mathbb{W}$ is a $\mathcal{C}$-set given that and $\hat{\mathbb{W}}$ is a $\mathcal{PC}$-set [96].

As previously discussed, and given (3.20), (3.16) represents an obvious choice for the nominal prediction model used in the MPC optimization. Define again then $e(t) = x(t) - \bar{x}(t)$ and $u(t) = \bar{u}(t) + K(x(t) - \bar{x}(t))$, the error dynamics follow

$$e(t+1) = \left(\bar{A} + \bar{B}K\right) e(t) + w(t),$$

and so it is only left to find a stabilizing K for the pair $\left(\bar{A}, \bar{B}\right)$ and an associated (admissible) RPI set. There is, however, one remaining obstacle. Since the model is uncertain, there is no immediate guarantee that a stabilizing feedback for the nominal dynamics $\left(\bar{A}, \bar{B}\right) \in \mathcal{M}$ also stabilizes the true dynamics. This is of paramount importance given the exponential stability guarantee available for the nominal closed-loop in standard tube MPC. Indeed, consider that tube MPC as in Section 2.3.2 is implemented with an associated disturbance rejection gain K. Suppose then that at some time t it holds that the nominal state and input are $\bar{x}(t) = \bar{u}(t) = \mathbf{0}$. It follows

$$x(t+1) = \bar{A}_K x(t) + w(t) = A_K x(t) + B\hat{w}(t) \tag{3.22}$$

with $\bar{A}_K = \left(\bar{A} + \bar{B}K\right)$ and $A_K = (A + BK)$. It is clear then that, since $\hat{w}(t)$ is driven solely with excitation purposes, the undisturbed closed-loop (3.22) could be unstable.

As previously discussed, and assuming the vertices of an outer bounding polyhedron for $\mathcal{M}$ are known, a set of LMIs can be constructed to compute a single linear gain K that stabilizes the entire model set $\mathcal{M}$ following the developments in [26]. It might be the case, however, that there is an infinite number of linear gains that fulfil the required inequalities, making it difficult to chose an appropriate one. In [26] the sole purpose of K is to generate a cost decrease and an optimization problem is set accordingly to drive the aforementioned LMIs. In the present context, however, the purpose of K is to compute an associated (admissible) RPI set which is usually sought to be minimal in size, hence not allowing for a straightforward definition for a driving optimization.

To avoid the implementation of an additional optimization problem and the aforementioned LMIs, an iterative procedure is proposed here that allows to freely select the linear gain K and to test its adequacy after. First note that for any nominal model $\left(\bar{A}, \bar{B}\right) \in \mathcal{M}$ that fulfils Assumption 2.1 a stabilizing linear gain K and RPI set $\mathbb{S}$ (with respect to and $\mathbb{W}_p$) can be computed. Furthermore, admissibility of $\mathbb{S}$, as defined in Definition 2.3, is necessary to guarantee that $\bar{\mathbb{X}} \neq \emptyset$ and thus that the nominal optimization has a non-empty RoA. In view of this, and ignoring for now the partition of the input, the following result holds.

Proposition 3.2. Consider a certain nominal model $(\bar{A}, \bar{B}) \in \mathcal{M}$ that fulfils Assumption 2.1, and a stabilizing linear feedback K. If the RPI set $\mathbb{S}$ corresponding to $\mathbb{W}_p$ is admissible with respect to constraint sets (2.2), then A_K is Schur for all $(A, B) \in \mathcal{M}$.

Proof. Suppose $x(0) \in \mathbb{S}$ and so $\bar{x}(t) = \bar{u}(t) = \mathbf{0}$ for all $t \geq 0$. Since $\mathbb{S}$ is constraint admissible it holds that

$$\begin{aligned} &x(t) \in \mathbb{S} \implies x(t) \in \mathbb{X} \\ &x(t) \in \mathbb{S} \implies u(t) = Kx(t) \in K\mathbb{S} \implies u(t) \in \mathbb{U}, \end{aligned}$$

and so $w_p(t) \in \mathbb{W}_p$ for all $t \geq 0$. According to Definition 2.2 then $\bar{A}_K x + w_p \in \mathbb{S}$ for any $x \in \mathbb{S}$. It is straightforward to show that $\bar{A}_K x + w_p = A_K x$, thus $A_K x \in \mathbb{S}$ for all $x \in \mathbb{S}$. By Definition 2.2 this implies that $\mathbb{S}$ is a PI set for A_K, hence A_K must be Schur. ■

Proposition 3.2 allows for an arbitrary selection of the linear (nominally) stabilizing gain K, followed by a verification of its eligibility through the admissibility of the RPI set $\mathbb{S}$. The proof to Proposition 3.2 ignores the input partition, but the latter can be easily included while maintaining the same outcome. Note however, that the admissibility of $\mathbb{S}$ depends on the size of $\mathbb{W}_p$ and $\hat{\mathbb{U}}$, hence given a collection $(\bar{A}, \bar{B}, \mathbb{X}, \mathbb{U}, \alpha)$, there is a bound on the parametric uncertainty that this approach can accept (i.e., a bound on the size of $\mathcal{M}$).

3.3.2 MPC Optimization

Following the proposed input partition, with its corresponding constraint allocation, the optimization problem to be solved at each time instant is

$$\mathbb{P}_N(x(t))\colon \qquad \min_{\bar{\boldsymbol{u}}, \bar{x}_0} J_N(\bar{\boldsymbol{u}}, \bar{x}_0) \tag{3.23a}$$

s.t. (for $k = 0, \ldots, N-1$)

$$x(t) - \bar{x}_0 \in \mathbb{S} \quad \text{if } t = 0 \tag{3.23b}$$

$$\bar{x}_0 = \bar{x}(t) \quad \text{if } t > 0 \tag{3.23c}$$

$$\bar{x}_{k+1} = \bar{A}\bar{x}_k + \bar{B}\bar{u}_k \tag{3.23d}$$

$$\bar{x}_k \in \bar{\mathbb{X}} \tag{3.23e}$$

$$\bar{u}_k \in \bar{\mathbb{U}} \tag{3.23f}$$

$$\bar{x}_N \in \bar{\mathbb{X}}_f \subseteq \bar{\mathbb{X}}. \tag{3.23g}$$

All the elements in (3.23) are defined according to Section 2.3.2, except the tightened input constraint set which is $\bar{\mathbb{U}} = \hat{\mathbb{U}} \ominus K\mathbb{S}$ given the constraint allocation defined by (3.13). The following result establishes the stabilizability guarantees of the closed-loop when the above optimization problem is used to drive the nominal input. This is a replica of Proposition 2.2, however a detailed proof is provided here for completeness.

Theorem 3.3 (Robust stabilizability)**.** If (a) Assumption 2.1 holds with a certain K for the nominal system $(\bar{A}, \bar{B})$, (b) the set $\mathbb{S}$ is an admissible RPI set for $\bar{A}_K$ with respect to constraints (2.2) and disturbance set $\mathbb{W}$, (c) the set $\bar{\mathbb{X}}_f$ is an admissible PI set for $\bar{A}_K$ and tightened constraint (2.4a), (d) $Q, R > 0$, and P fulfils $\bar{A}_K^\top P \bar{A}_K + Q + K^\top R K - P \leq 0$ and (e) the loop is closed with $u(t) = \kappa_N(x(t)) = \bar{\kappa}_N(x(t)) + Ke(t)$ and $\hat{u}(t) = \bar{\kappa}_N(x(t))$, then (i) the optimization problem (2.2) is recursively feasible, (ii) state and input constraints are met at all times despite the disturbances, and (iii) there exist constant scalars $b, d, f > 0$ such that for all $\bar{x} \in \bar{\mathcal{X}}_N$ it holds that

$$b|\bar{x}_0^\star(x)|_2^2 \leq V_N(\bar{x}) \leq d|\bar{x}_0^\star(x)|_2^2 \tag{3.24a}$$

$$V_N\left(\bar{A}\bar{x} + \bar{B}\bar{\kappa}_N(\bar{x})\right) - V_N(\bar{x}) \leq -f|\bar{x}_0^\star(x)|_2^2. \tag{3.24b}$$

Furthermore, the set $\mathbb{S}$ is exponentially stable with a region of attraction $\bar{\mathcal{X}}_N \oplus \mathbb{S}$ for the constrained closed-loop system

$$x(t+1) = \bar{A}x(t) + \bar{B}\kappa_N\left(x(t)\right) + w(t), \tag{3.25}$$

for all $w(t) \in \mathbb{W}$, while the origin is exponentially stable with a region of attraction $\bar{\mathcal{X}}_N$ for the nominal closed-loop system

$$\bar{x}(t+1) = \bar{A}\bar{x}(t) + \bar{B}\bar{\kappa}_N\left(x(t)\right). \tag{3.26}$$

Proof. For (i) suppose that the optimal solution at time t is $\bar{\boldsymbol{u}}^\star\left(\bar{x}(t)\right)$ with an associated predicted state trajectory $\bar{\boldsymbol{x}}^\star\left(\bar{\boldsymbol{u}}^\star\left(\bar{x}(t)\right)\right)$ or simply $\bar{\boldsymbol{x}}^\star\left(\bar{x}(t)\right)$. It follows that $\bar{x}_0^\star\left(\bar{x}(t)\right)$ meets (3.23b) if $t = 0$ and (3.23c) otherwise. Furthermore, the dynamics constraint (3.23d) is met by the pair of optimized sequences, hence $\bar{\boldsymbol{x}}^\star\left(\bar{x}(t)\right)$ fulfils (3.23e), $\bar{\boldsymbol{u}}^\star\left(\bar{x}(t)\right)$ fulfils (3.23f) and $\bar{x}_N^\star\left(\bar{x}(t)\right)$ meets (3.23g). In view of this, it is easy to show that

$$\tilde{\boldsymbol{u}} = \left\{\bar{u}_1^\star\left(\bar{x}(t)\right), \bar{u}_2^\star\left(\bar{x}(t)\right), \ldots, \bar{u}_{N-1}^\star\left(\bar{x}(t)\right), K\bar{x}_N^\star\left(\bar{x}(t)\right)\right\} \tag{3.27}$$

is a feasible solution at time $t+1$. Indeed note that by (3.26) it follows that $\bar{x}(t+1) = \bar{x}_1^\star\left(\bar{x}(t)\right)$, which fulfils constraint (3.23c). Moreover, the input sequence $\tilde{\boldsymbol{u}}$ in (3.27) is formed by the elements of $\bar{\boldsymbol{u}}^\star$ starting from prediction time $k = 1$ up to step $N-1$, therefore the state trajectory $\tilde{\boldsymbol{x}}$ associated to $\tilde{\boldsymbol{u}}$ fulfils

$$\tilde{\boldsymbol{x}} = \left\{\bar{x}_1^\star\left(\bar{x}(t)\right), \bar{x}_2^\star\left(\bar{x}(t)\right), \ldots, \bar{x}_N^\star\left(\bar{x}(t)\right), \bar{A}_K\bar{x}_N^\star\left(\bar{x}(t)\right)\right\}. \tag{3.28}$$

Since $\bar{x}_N^\star\left(\bar{x}(t)\right) \in \bar{\mathbb{X}}_f$ and the latter is an admissible PI set for the closed-loop $\bar{A}_K$ it holds that $K\bar{x}_N^\star\left(\bar{x}(t)\right) \in \bar{\mathbb{U}}$ and $\bar{A}_K\bar{x}_N^\star\left(\bar{x}(t)\right) \in \bar{\mathbb{X}}_f$, thus the prediction pair $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{x}})$ as defined by (3.27) and (3.28) fulfil constraints (3.23e)–(3.23g), thus making $\tilde{\boldsymbol{u}}$ a feasible solution at time $t+1$.

For (ii) note that (3.23b) forces $e(0) \in \mathbb{S}$. Closing the loop of the true and nominal trajectories with $\kappa_N(\cdot)$ and $\bar{\kappa}_N(\cdot)$, respectively, results in $e(t+1) = \bar{A}_K e(t) + w(t)$ for all $t \in \mathbb{N}_0$, thus $e(t) \in \mathbb{S}$ holds for all t by the robust invariance of $\mathbb{S}$. It follows then that

$$\begin{aligned} & x(t) - \bar{x}(t) \in \mathbb{S} && (3.29\text{a})\\ \Longrightarrow\ & x(t) \in \{\bar{x}(t)\} \oplus \mathbb{S} \subset \bar{\mathbb{X}} \oplus \mathbb{S} && (3.29\text{b})\\ \Longrightarrow\ & x(t) \in (\mathbb{X} \ominus \mathbb{S}) \oplus \mathbb{S} && (3.29\text{c})\\ \Longrightarrow\ & x(t) \in \mathbb{X}. && (3.29\text{d}) \end{aligned}$$

The first inclusion in (3.29) holds by the definition of $e(t)$, the second one by the recursive feasibility of the optimization which keeps the nominal trajectories inside $\bar{\mathbb{X}}$, the third by the definition of the tightened constraint (2.4a) and the last one due to the properties of the Pontryagin difference and the Minkowsky addition for convex sets [96]. The same chain of arguments can be made for $u(t)$.

For (iii) first note that for all $\bar{x} \in \bar{\mathbb{X}}_f$

$$\begin{aligned} V_f\left(\bar{A}_K\bar{x}\right) - V_f\left(\bar{x}\right) &= \left(\bar{A}_K\bar{x}\right)^\top P\left(\bar{A}_K\bar{x}\right) - \bar{x}^\top P\bar{x} && (3.30\text{a})\\ &= \bar{x}^\top\left(\bar{A}_K^\top P\bar{A}_K - P\right)\bar{x} && (3.30\text{b})\\ &\leq -\ell\left(\bar{x}, K\bar{x}\right) && (3.30\text{c}) \end{aligned}$$

where the inequality holds by hypothesis (d). Suppose now that the optimal solution at time t is defined by the pair $(\bar{\boldsymbol{u}}^{\star}(\bar{x}(t)), \bar{\boldsymbol{x}}^{\star}(\bar{x}(t)))$, then the associated optimal cost is

$$V_N(\bar{x}(t)) = \ell(\bar{x}_0^{\star}(\bar{x}(t)), \bar{u}_0^{\star}(\bar{x}(t))) + \sum_{k=1}^{N-1} \ell(\bar{x}_k^{\star}(\bar{x}(t)), \bar{u}_k^{\star}(\bar{x}(t))) + V_f(\bar{x}_N^{\star}(\bar{x}(t))). \tag{3.31}$$

Recall that the pair $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{x}})$ represents a feasible solution at time $t+1$, albeit not necessarily optimal. Suppose then that the cost associated to the feasible pair is defined by $\tilde{V}_N(\bar{x}(t+1))$, then similarly

$$\begin{aligned}\tilde{V}_N(\bar{x}(t+1)) = \sum_{k=1}^{N-1} \ell(\bar{x}_k^{\star}(\bar{x}(t)), \bar{u}_k^{\star}(\bar{x}(t))) + \ell(\tilde{x}_N^{\star}(\bar{x}(t)), K\bar{x}_N^{\star}(\bar{x}(t))) \\ + V_f\left(\left(\bar{A}_K\right)\bar{x}_N^{\star}(\bar{x}(t))\right).\end{aligned} \tag{3.32}$$

Define $\Delta V_N(\bar{x}(t)) = V_N(\bar{x}(t+1)) - V_N(\bar{x}(t))$, the possible sub-optimality of $(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{x}})$ implies $V_N(\bar{x}(t+1)) \leq \tilde{V}_N(\bar{x}(t+1))$, hence $\Delta V_N(\bar{x}(t)) \leq \tilde{V}_N(\bar{x}(t+1)) - V_N(\bar{x}(t))$. It follows then from (3.31) and (3.32) that

$$\begin{aligned}\Delta V_N(\bar{x}(t)) \leq \quad & -\ell(\bar{x}_0^{\star}(\bar{x}(t)), \bar{u}_0^{\star}(\bar{x}(t))) - V_f(\bar{x}_N^{\star}(\bar{x}(t))) \\ & + \ell(\tilde{x}_N^{\star}(\bar{x}(t)), K\bar{x}_N^{\star}(\bar{x}(t))) + V_f\left(\left(\bar{A}_K\right)\bar{x}_N^{\star}(\bar{x}(t))\right) \\ \Longrightarrow \Delta V_N(\bar{x}(t)) \leq \quad & -\ell(\bar{x}_0^{\star}(\bar{x}(t)), \bar{u}_0^{\star}(\bar{x}(t))),\end{aligned}$$

with the second inequality following from (3.30).

Given the cost function (2.6), it is straightforward to show that for all $\bar{x}(t) \in \bar{\mathcal{X}}_N$ it holds that $V_N(\bar{x}(t)) \geq \ell(\bar{x}_0^{\star}(\bar{x}(t)), \bar{u}_0^{\star}(\bar{x}(t)))$ and that $V_N(\bar{x}(t)) \leq V_f(\bar{x}_0^{\star}(\bar{x}(t)))$ for all $\bar{x}(t) \in \bar{\mathbb{X}}_f$. Moreover, according to (d) it is also trivial to show that

$$\begin{aligned}\ell(\bar{x}_0^{\star}(\bar{x}(t)), \bar{u}_0^{\star}(\bar{x}(t))) &\geq ||\bar{x}_0^{\star}(\bar{x}(t))||_Q^2 \\ &\geq \xi_m(Q)|\bar{x}_0^{\star}(\bar{x}(t))|_2^2 \\ V_f(\bar{x}_0^{\star}(\bar{x}(t))) &\leq \xi_M(P)|\bar{x}_0^{\star}(\bar{x}(t))|_2^2.\end{aligned}$$

In summary, it holds that

$$V_N(\bar{x}(t)) \geq \xi_m(Q)|\bar{x}_0^{\star}(\bar{x}(t))|_2^2 \quad \forall \bar{x}(t) \in \bar{\mathcal{X}}_N \tag{3.33a}$$

$$V_N(\bar{x}(t)) \leq \xi_M(P)|\bar{x}_0^{\star}(\bar{x}(t))|_2^2 \quad \forall \bar{x}(t) \in \bar{\mathbb{X}}_f \tag{3.33b}$$

$$\Delta V_N(\bar{x}(t)) \leq -\xi_m(Q)|\bar{x}_0^{\star}(\bar{x}(t))|_2^2 \quad \forall \bar{x}(t) \in \bar{\mathcal{X}}_N \tag{3.33c}$$

According to [1], it can be shown that if $\bar{\mathbb{X}}_f$ is compact, then there exists a constant $d > \xi_M(P)$ such that (3.33b) holds for all $\bar{x} \in \bar{\mathcal{X}}_N$. Thereby (3.24) holds and the optimal cost function is a Lyapunov function for the nominal closed-loop dynamics. Furthermore, there exists constants $c > 0$ and $\gamma \in (0,1)$ such that $|\bar{x}_0^\star(\bar{x}(t))|_2^2 \leq c\gamma^t |\bar{x}_0^\star(\bar{x}(0))|_2^2$, indeed, note that (3.24a) means that ${}^f/{}_d V_N(\bar{x}(t)) \leq f|\bar{x}_0^\star(\bar{x}(t))|_2^2$, thus (3.24b) implies

$$V_N(x(t+1)) \leq (1 - {}^f/{}_d)V_N(x(t)).$$

The same holds for all $t \geq 0$ thus

$$V_N(x(t)) \leq \gamma^t V_N(x(0)).$$

with $\gamma = (1 - {}^f/{}_d)$. Finally, the upper and lower bounds in (3.24a) result in $|\bar{x}_0^\star(\bar{x}(t))|_2^2 \leq c\gamma^t |\bar{x}_0^\star(\bar{x}(0))|_2^2$ with $c = {}^d/{}_b$, which ensures exponential stability of the origin for the nominal closed-loop (3.26)

Now, note that $x(t) = \bar{x}(t) + e(t)$ hence $d_H(x(t), \mathbb{S}) = d_H(\bar{x}(t) + e(t), \mathbb{S})$. Now, since $e(t) \in \mathbb{S}$ at all times, it holds that

$$\begin{aligned} & d_H(x(t), \mathbb{S}) \leq d_H(\bar{x}(t) + e(t), e(t)) = |\bar{x}_0^\star(\bar{x}(t))|_2^2 \\ \implies & d_H(x(t), \mathbb{S}) \leq c\gamma^i |\bar{x}_0^\star(\bar{x}(0))|_2^2 \end{aligned}$$

for all $t \geq 0$. According to Corollary 2.2 this implies exponential stability of the set $\mathbb{S}$ for the true closed-loop (3.25) for all $w(t) \in \mathbb{W}$, which concludes the proof. ■

In summary, Theorem 3.3 provides the necessary conditions to guarantee that if the loop is closed with the MPC control law $\kappa_N(\cdot)$, driven by optimization (3.23), then constraints are satisfied at all times despite the model uncertainty and the excitation that $\hat{w}(t)$ may induce. Furthermore, Theorem 3.3 also shows that the set $\mathbb{S}$ is exponentially stable for the true closed-loop dynamics irrespective of the disturbance $w(t) \in \mathbb{W}$.

It is important to remark that the size of the RoA is tightly related to the terminal constraint set $\bar{\mathbb{X}}_f$ and the tightening set $\mathbb{S}$. Indeed, a larger terminal set allows for a larger $\bar{\mathcal{X}}_N$ while a larger $\mathbb{S}$ results in a smaller $\bar{\mathbb{X}}$ and consequently a smaller $\bar{\mathcal{X}}_N$. It is clear then that to allow for a larger RoA for the controller, the best choices for terminal constraint set and tightening set are, correspondingly, the maximal PI set and the minimal RPI set related to $\bar{A}_K$ and $\mathbb{W}$ [96]. Furthermore note that, although the same linear feedback is employed in Theorem 3.3 to compute both invariant sets, this is not necessary. It could be argued that, given a cost defined to match the infinite horizon LQ problem, both should be chosen as the optimal LQR gain in order to be consistent. However, less aggressive feedback laws usually result in larger maximal PI sets, thus allowing for a larger RoA. Similarly more aggressive feedback laws result in smaller RPI sets, thus enlarging the corresponding RoA.

3.4 PE optimization

Section 3.2 developed the required characteristics of the exciting input in order to guarantee convergent estimates within the proposed input partition architecture. Particularly, it is required for the exciting part of the input $\hat{w}(t)$ to be persistently exciting of order $2n + m + 1$ according to Lemma 3.2. This definition fits the best with the receding horizon fashion of the AMPC architecture proposed since it places the design requirements only on the current value of $\hat{w}(t)$. Indeed, in a receding horizon architecture future predicted values are not necessarily applied, hence characteristics that depend on them may be met in the open-loop predictions but not in the true closed-loop. Equation (3.8) presents the PE condition as a lower and upper bound enforced over a portion of the covariance matrix of the corresponding sequence, however in what follows the focus is placed solely over the lower bound. The upper bound is trivially met given that the summation in (3.8) is finite and that $\hat{\mathbb{W}}$ is a $\mathcal{PC}$-set.

It is important to note that the PE characteristics of the exciting input, according to Lemma 3.2, are completely independent of the actual model parameters and of the current state of the plant. Moreover, the tube MPC proposed in Section 3.3 does not need to know $\hat{w}(t)$ in order to compute a control action at time t, thus a persistently exciting sequence $\vec{\hat{w}}_t$ that fulfils the required PE conditions could be devised off-line (as in [69]). However, in order to reduce the impact that $\hat{w}(t)$ has over the control objective, a receding horizon optimization similar to (3.23) is now proposed to drive the exciting part. The goal of this approach is to decrease the unnecessary disturbing effects that an off-line computed $\hat{w}(t)$ may generate, by introducing feedback in its computation. Indeed, there are infinitely many persistently exciting sequences that fulfil Lemma 3.2, but possibly only a few that do so with minimal state disturbance for a particular state trajectory that depends on the control sequence $\vec{\hat{u}}_t$. First a general structure for the proposed optimization is presented, followed by a trivial recursively feasible solution and two specializations that include additional constraints in order to guarantee recursive feasibility under non-trivial solutions.

3.4.1 General structure of the PE optimization

First define, as in Section 3.2, the sequence of exciting inputs up to time t as $\vec{\hat{w}}_t = \{\hat{w}(0), \ldots, \hat{w}(t)\}$ and

$$\hat{\boldsymbol{w}}_{t-j} = \begin{bmatrix} \hat{w}(t-j) \\ \hat{w}(t-j-1) \\ \vdots \\ \hat{w}(t-j-h+1) \end{bmatrix}$$

where h is the order of persistent excitation sought. It follows form Lemma 3.2 that the exciting sequence $\vec{\hat{w}}_t$ is guaranteed to be PE of order h at time t if $\boldsymbol{\Gamma}(\hat{w}(t)) > 0$ with

$$\boldsymbol{\Gamma}(\hat{w}(t)) = \sum_{j=0}^{l-1} \left(\hat{\boldsymbol{w}}_{t-j} \hat{\boldsymbol{w}}_{t-j}^{\top} \right) - \rho_0 \mathcal{I} \tag{3.34}$$

and ρ_0 chosen according to the discussion on time-varying systems in Section 3.2.3.3. In order to introduce feedback in the selection of $\hat{w}(t)$ it is proposed that, at each time instant, the value of $\hat{w}(t)$ is obtained by solving an MPC-like optimization problem that minimizes the effect of such an exciting action over the control objective. The latter is defined by the cost function associated to (3.23), so the general structure of the proposed optimization problem is

$$\hat{\mathbb{P}}\left(\vec{\hat{w}}_{t-1}, x(t)\right): \qquad \min_{\hat{\boldsymbol{w}}} \sum_{k=0}^{\bar{N}} \ell(x_k, \hat{w}_k) \tag{3.35a}$$

s.t. (for $k < 0$)

$$\hat{w}_k = \hat{w}(t-k) \tag{3.35b}$$

and (for $k = 0, \ldots, \bar{N}$)

$$x_0 = x(t) \tag{3.35c}$$

$$x_{k+1} = \bar{A} x_k + \bar{B} \hat{w}_k \tag{3.35d}$$

$$\hat{w}_k \in \hat{\mathbb{W}} \tag{3.35e}$$

$$\boldsymbol{\Gamma}(\hat{w}_k) > 0, \tag{3.35f}$$

where the decision variable is $\hat{\boldsymbol{w}} = \{\hat{w}_0, \ldots, \hat{w}_{\bar{N}}\}$ and $\bar{N}$ is the prediction horizon. The optimum value at time t is defined by $\hat{\boldsymbol{w}}^\star(x(t)) = \left\{\hat{w}_0^\star(x(t)), \ldots, \hat{w}_{\bar{N}}^\star(x(t))\right\}$ or simply $\hat{\boldsymbol{w}}^\star(t)$ and the exciting input at time t is set by the feedback law $\hat{w}(t) = \mu(x(t)) = \hat{w}_0^\star(x(t))$.

It is important to note that PE, according to Lemma 3.2, cannot be guaranteed until a sufficient amount of exciting inputs have been applied. That is because $\mathbf{\Gamma}(\hat{w}(t))$ requires the knowledge of $h + l - 2$ past inputs in order to be constructed. This is reflected in (3.35) by the dependency of $\hat{\mathbb{P}}(\cdot, \cdot)$ on $\vec{\hat{w}}_{t-1}$ and in the constraints (3.35b) and (3.35f). Ultimately, this means that (3.35) is not a memoryless optimization like (3.23), hence a (persistently exciting) buffer sequence is required for initialization. The latter, however, is not necessarily applied to the system, but only used to evaluate the corresponding constraints during the initial time steps.

A major difficulty in solving (3.35) is that the PE constraint (3.35f) is non-convex. Indeed, it is shown in [74] that for $\bar{N} = 0$ (3.35f) defines the outside of an ellipsoid in the $\mathbb{R}^n$ space, but the complexity of such a constraint grows with the length of the horizon. It is out of the scope of this chapter to provide a comprehensive approach to tackle the non-convexity of (3.35f), however a standard solution would be to relax it by introducing a slack variable such as in [77].

The purpose of an optimization problem like (3.35) is to produce, at each time instant, an exciting input $\hat{w}(t)$ that guarantees PE of $\vec{\hat{w}}_t$ of appropriate order. As opposed to other approaches [44, 77, 78], the PE characteristics of $\vec{\hat{w}}_t$ at time t do not depend on predicted values, but only on the currently optimized exciting input. This implies that PE of the exciting sequence at the current time instant is guaranteed if constraint (3.35f) is met merely at $k = 0$, yet it is enforced over the whole horizon to better account for the effect of future excitations over the state trajectories and hence the cost. Nevertheless, given the finite horizon of the optimization, compliance of the PE constraint at time t does not necessarily guarantee the same at time $t + 1$. This is because the PE constraint (3.35f) is formed by looking at a rolling windows of past values, one of which ceases to be contained in this window at time $t + 1$. In simpler words, past portions of $\vec{\hat{w}}_t$ with high excitation content may permit the addition of little excitation in the present while maintaining the required PE order. If too low, however, and given the bound on the exciting input, it may become impossible to guarantee the necessary order of PE in the future. This issue is similar to the difficulties found in guaranteeing constraint satisfaction in standard finite-horizon MPC without terminal constraints (or cost).

Finally, note that the optimization problem (3.35) minimizes the running cost associated to a nominal state trajectory that would be generated by purely feeding the exciting part to the nominal model. This, however, is only an approximation of the true effect of the exciting part over the state trajectories. Setting aside the fact that (3.36b) uses the nominal representation of the plant, since it is the only one available, the true input exerted on the plant is $u = \hat{u} + \hat{w}$. Given (2.6), it follows that $\ell(x, u) \leq \ell(x, \hat{u}) + \ell(x, \hat{w})$ for any $x \in \mathbb{R}^n$, $\hat{u}, \hat{w} \in \mathbb{R}^m$. Nevertheless, it is not possible to predict the values of $\hat{u}(t)$ throughout the horizon since $\kappa_N(\cdot)$ depends on the true value of the state at each instant. Furthermore, even if $\bar{\boldsymbol{u}}^\star(\bar{x}(t))$ is employed, it would then be necessary to solve both optimization problems in sequence rather than in parallel, increasing the time required to compute the control action at each time instant. However, it is expected that this definition of the cost helps minimize the disturbing impact that $\hat{w}(t)$ has over the plant, while securing persistence of excitation through constraint (3.35f) and the feedback law $\mu(x(t))$.

3.4.2 Recursively feasibly PE optimization

From the above discussion it follows that the constrained optimization (3.35) is not necessarily suitable to introduce feedback in the selection of the exciting part of the input, given that it may become infeasible at any time instant t. This is true even if the horizon is set to $\bar{N} = 0$, that is, even if the optimization variable amounts to a single exciting element. Since this issue is similar to those that arise in MPC without terminal constraints, it is proposed here to augment (3.35) with a *terminal* constraint in order to guarantee recursive feasibility by inducing a periodically invariant behaviour towards the end of the horizon. This will result in a recursive feasibility proof that is very similar to that in standard MPC implementations, resorting to a previously feasible solution extended in some way. It follows then that this approach not only tackles the infeasibility issues, but also the non-convexity of the PE constraint, guaranteeing the existence and knowledge of a feasible solution without the need to optimize.

In what follows the additional constraint is referred to as the periodicity constraint and it will be broadly denoted as $\mathcal{C}(\hat{w}_k)$ with an specific definition that depends on the rational employed to guarantee recursive feasibility of (3.35). Two different rationales will be discussed but in both $\mathcal{C}(\hat{w}_k)$ will be an equality constraint applied over all elements of the optimization variable except the first one, letting a single input to be optimized throughout the prediction horizon at each sampling time. It may seem then that the horizon $\bar{N}$ could be set to 0, however this is not the case since the optimized value has a direct effect on the compliance of the PE constraint (3.35f) in future prediction times and this has to be accounted for in order to guarantee recursive feasibility. Furthermore, a longer horizon will also allow to better account for the effects of a exciting sequence (rather than an exciting input) over the running cost. The horizon then will remain as an arbitrary value for now, however it will be shown that a minimum is required to guarantee recursive feasibility by verifying the PE constraint well into the future.

In view of all these considerations, the optimization problem (3.35) is recast as

$$\hat{\mathbb{P}}\left(\overrightarrow{\hat{w}}_{t-1}, x(t)\right): \qquad \min_{\hat{w}_0} \sum_{k=0}^{\bar{N}} \ell\left(x_k, \hat{w}_k\right) \tag{3.36a}$$

s.t. (for $k < 0$)

$$\hat{w}_k = \hat{w}(t-k) \tag{3.36b}$$

(for $k = 0, \ldots, \bar{N}$)

$$x_0 = x(t) \tag{3.36c}$$

$$x_{k+1} = \bar{A}x_k + \bar{B}\hat{w}_k \tag{3.36d}$$

$$\hat{w}_k \in \hat{\mathbb{W}} \tag{3.36e}$$

$$\boldsymbol{\Gamma}\left(\hat{w}_k\right) > 0 \tag{3.36f}$$

and (for $k = 1, \ldots, \bar{N}$)

$$\mathcal{C}(\hat{w}_k) = 0. \tag{3.36g}$$

Note that, given the inclusion of the periodicity constraint, the decision variable is reduced from $\hat{\boldsymbol{w}} = \{\hat{w}_0, \ldots, \hat{w}_{\bar{N}}\}$ to $\hat{w}_0$. The associated optimum is then $\hat{w}_0^\star(t)$ rather than $\hat{\boldsymbol{w}}^\star(t)$, but the latter will still be used, acknowledging that $\bar{N}-1$ of its elements are not really optimized but fixed by constraint (3.36g).

Constraint (3.36b) is employed to endow the optimization problem with memory, so that constraint (3.36f) can be evaluated at all prediction times independent of l and h. Constraints (3.36c) and (3.36d) allow for the computation of the predicted state trajectory, given a particular exciting input sequence, but have no influence in the feasibility of the optimization and could be merge with (3.36a) in a more concise definition of the cost function. It follows that, in order to design the periodicity constraint, the focus is placed on constraints

(3.36e) and (3.36f).

3.4.2.1 Buffer sequence for recursive feasibility

Before delving into the periodicity constraint, the design of an appropriate buffer sequence for initialization is discussed. Indeed, given that the PE constraint is designed according to Lemma 3.2, a total of $h+l-2$ initialization exciting elements are required to evaluate $\mathbf{\Gamma}(\hat{w}_k)$ at time 0. The buffer sequence then must be one such that it allows for the PE constraint (3.36f) to be met with a single optimization variable that is $\hat{w}_0 \in \hat{\mathbb{W}}$. It is proposed then to design a buffer sequence that it is PE on its own, when applied with a certain period. This not only will allow for the optimization at time 0 to be feasible, but also to use the buffer as a fall-back for recursive feasibility. Indeed, despite (3.36f) being a non-convex constraint, it is possible to guarantee recursive feasibility of (3.36) resorting to a trivial periodic implementation of an appropriate buffer sequence, even when no terminal constraint $\mathcal{C}(\hat{w}_k)$ is employed. Define the buffer sequence as $\vec{\hat{w}}_b = \{\hat{w}_b(-h-l+2), \dots, \hat{w}_b(0)\}$ with negative time indexing in order to make explicit that this sequence is for initialization, and consider the following assumption on its properties.

Assumption 3.2. $\vec{\hat{w}}_b$ is such that (i) $\hat{w}_b(j) = \hat{w}_b(j-l)$ for all $j \geq -h$, (ii) $\hat{w}_b(t) \in \hat{\mathbb{W}}$ for all $t \in [-h-l+2, 0]$ and (iii) $\mathbf{\Gamma}(\hat{w}_b(0)) > 0$.

According to Assumption 3.2 only l elements of the buffer sequence are independent, the rest are generated by an l-periodic repetition of past values. Assumption 3.2 presents the design requirements for a buffer sequence that will allow for a recursive feasibility proof.

Proposition 3.3. Suppose $\vec{\hat{w}}_b$ meets Assumption 3.2 and that $\mathcal{C}(\hat{w}_k) := 0$. If the buffer sequence is used to verify constraint (3.36b) during the initial time steps, that is $\hat{w}(t) = \hat{w}_b(t)$ for all $t \in [-h-l+2, -1]$, then a feasible solution for the optimization (3.36) at time $t = 0$ is

$$\hat{\boldsymbol{w}}^f(t) = \left\{\hat{w}_0^f(t), \dots, \hat{w}_{\bar{N}}^f(t)\right\} \tag{3.37a}$$

$$\hat{w}_k^f(t) = \begin{cases} \hat{w}(t+k-l) & k \in [0,\ l-1] \\ \hat{w}_{k-l}^f(t) & k \in \left[l, \bar{N}\right] \end{cases} \tag{3.37b}$$

Furthermore, $\hat{w}(t) = \mu^f\,(x(t)) = \hat{w}(t-l)$ represents a feasible open-loop control law that guarantees $\vec{\hat{w}}_t$ is persistently exciting of order h at all t according to Lemma 3.2.

Proof. The proof hinges on the properties of the buffer sequence given by Assumption 3.2, and that the proposed feasible feedback law $\mu^f\left(x(t)\right)$ closes the loop while maintaining the same periodicity.

It is first shown that (3.37) is feasible at time $t = 0$. Indeed, given Assumption 3.2, the candidate solution (3.37) at time $t = 0$ results in $\hat{w}_0^f(0) = \hat{w}(-l) = \hat{w}_b(-l)$. Since Assumption 3.2 requires every element of the buffer sequence to be inside $\hat{\mathbb{W}}$, then the candidate solution meets (3.36e) at time 0. From (3.34) it easy to see that for any t

$$\begin{aligned}\mathbf{\Gamma}(\hat{w}(t)) &= \hat{\boldsymbol{w}}_t\hat{\boldsymbol{w}}_t^\top + \sum_{j=1}^{l-1}\left(\hat{\boldsymbol{w}}_{t-j}\hat{\boldsymbol{w}}_{t-j}^\top\right) - \rho_0\mathcal{I} + \left(\hat{\boldsymbol{w}}_{t-l}\hat{\boldsymbol{w}}_{t-l}^\top - \hat{\boldsymbol{w}}_{t-l}\hat{\boldsymbol{w}}_{t-l}^\top\right)\\ &= \mathbf{\Gamma}(\hat{w}(t-1)) + \hat{\boldsymbol{w}}_t\hat{\boldsymbol{w}}_t^\top - \hat{\boldsymbol{w}}_{t-l}\hat{\boldsymbol{w}}_{t-l}^\top,\end{aligned}$$

It follows then that, given (3.37),

$$\mathbf{\Gamma}(\hat{w}_0^f(0)) = \mathbf{\Gamma}(\hat{w}(-1)) + \hat{\boldsymbol{w}}_0\hat{\boldsymbol{w}}_0^\top - \hat{\boldsymbol{w}}_{-l}\hat{\boldsymbol{w}}_{-l}^\top$$

with

$$\hat{\boldsymbol{w}}_0 = \begin{bmatrix}\hat{w}_0^f(0)\\ \hat{w}(-1)\\ \vdots\\ \hat{w}(-h+1)\end{bmatrix} = \begin{bmatrix}\hat{w}(-l)\\ \hat{w}(-l-1)\\ \vdots\\ \hat{w}(-l-h+1)\end{bmatrix} = \hat{\boldsymbol{w}}_{-l},$$

where the equality of the first element follows from (3.37) and the rest from the l-periodicity of the buffer sequence established in Assumption 3.2. It follows that $\mathbf{\Gamma}(\hat{w}_0^f(0)) = \mathbf{\Gamma}(\hat{w}(-1)) = \mathbf{\Gamma}(\hat{w}_b(-1)) \geq 0$, where the positive definiteness follows from Assumption 3.2. The same arguments prove $\mathbf{\Gamma}(\hat{w}_k^f(0)) \geq 0$ for all $k \in \left[1, \bar{N}\right]$. The candidate then meets (3.36f), and so it is a feasible solution for (3.36) at time 0. The second part follows by induction, and the l-periodicity induced by $\mu^f\left(x(t)\right)$. ■

Proposition 3.3 ensures that the l-periodic repetition of a buffer signal that fulfils Assumption 3.2 is a feasible solution for (3.36), even without a periodicity constraint $\mathcal{C}(\hat{w}_k)$. This is not surprising since the control law $\mu^f(x(t))$ already induces periodicity, and periodic signals are persistently exciting of a certain order [3, 76]; the disadvantage, however, is clear. The buffer signal is computed entirely off-line hence its l-periodic implementation introduces no feedback. In what follows two rationales are explored in order to allow for the optimizer to introduce non-periodic values while maintaining feasibility and guaranteeing PE. Both stem out of the fact that the buffer sequence is a feasible, persistently exciting, l-periodic sequence, and rely on the appropriate design of the periodicity constraint (3.36g).

3.4.2.2 Recursive feasibility of a non-periodic signal: first approach

Ignore for now, as in Proposition 3.3, the periodicity constrain and assume that at time 0 the optimizer chosen to solve (3.36) finds a feasible optimum with $\hat{w}_0^\star(0) \neq \hat{w}(-l)$. Since by Proposition 3.3 the l-periodic repetition of the buffer sequence is feasible and persistently exciting, a valid question is whether this remains true, even after a non-periodic value, such as $\hat{w}_0^\star(0)$, has been introduced in the sequence

As it stands, it is not possible to ensure the above will happen, however an appropriate design of the periodicity constraint does allow it. In what follows note that the periodicity constraint is enforced over all elements of the predicted sequence except the first one, hence the optimization variable is reduced to $\hat{w}_0$.

Theorem 3.4. Suppose $\overrightarrow{\hat{w}_b}$ meets Assumption 3.2, that the horizon is set to $\bar{N} \geq h + l - 2$, that the buffer sequence is employed to verify constraint (3.36b) during the initial time steps, that $\hat{w}(t) = \hat{w}_b(t)$ for all $t \in [-h - l + 2, -1]$, and that the periodicity constraint (3.36g) is defined as

$$\mathcal{C}(\hat{w}_k) \coloneqq \begin{cases} \hat{w}_k - \hat{w}(t + k - l) & k \in [1,\ l - 1] \\ \hat{w}_k - \hat{w}(t + k - 2l) & k = l \\ \hat{w}_k - \hat{w}_{k-l} & k \in \left[l + 1,\ \bar{N}\right]. \end{cases} \tag{3.38}$$

If (a) the l-periodic feasible control law $\mu^f(x(t))$ from Proposition 3.3 is used to drive the exciting input until time instant $\bar{t} \geq 0$, and (b) at time $\bar{t}$ the optimizer in place chooses an optimum $\hat{\boldsymbol{w}}^\star(\bar{t})$ and sets $\hat{w}(\bar{t}) = \hat{w}_0^\star(\bar{t})$, then the control law

$$\hat{w}(t) = \nu^f(x(t)) = \begin{cases} \hat{w}(t - 2l), & t = \bar{t} + l \\ \mu^f(x(t)), & t \neq \bar{t} + l, \end{cases} \tag{3.39}$$

renders $\overrightarrow{\hat{w}}_t$ a PE sequence of order h at all $t \geq \bar{t} + 1$ according to Lemma 3.2 despite the possibility of $\hat{w}_0^\star(\bar{t}) \neq \mu^f(x(t)) = \hat{w}(\bar{t} - l)$.

Note that the focus is placed on guaranteeing PE of the appropriate order for all $t \geq \bar{t} + 1$ since by assumption the implementation of $\mu^f(x(t))$ for all $t < \bar{t}$ and feasibility of the optimization at time $\bar{t}$ guarantee $\overrightarrow{\hat{w}}_t$ is a PE sequence of order h for all $t \in [0, \bar{t}]$. Furthermore, note that it is assumed that at time $\bar{t}$ the optimizer chooses an optimum $\hat{\boldsymbol{w}}^\star(\bar{t})$, however only the first element is a real decision variable, since the periodicity constraint fixes the rest.

The rationale of Theorem 3.4 is to guarantee feasibility of the l-periodic buffer sequence even after a new value disrupts the periodicity at a single time instant. Indeed, this rationale seeks to return to the original buffer sequence and only lets the optimizer interrupt its periodicity at a single time step. This is realized in the periodicity constraint (3.38) by setting $\hat{w}_k = \hat{w}(t+k-2l)$ when $k = l$, thus avoiding to include the new value in the future. It is also realized in the feasible solution (3.39) by setting $\hat{w}_0^f(t) = \hat{w}(t-2l)$ if $t = \bar{t} + l$. This is why Theorem 3.4 does not guarantee recursive feasibility of (3.36) with periodicity constraint as in (3.38), but the existence of a PE control law.

This might seem counter-intuitive since the objective of such an approach to define the exciting input is to introduce feedback, nevertheless it will be shown that Theorem 3.4 does allow for such a feedback. The proof of Theorem 3.4 follows.

Proof. First note that, given (3.34) and the periodicity constraint (3.38) that eliminates the newly optimized value from future periodic loops, the last prediction time in which $\hat{w}_0^\star(\bar{t})$ is part of the computation of $\boldsymbol{\Gamma}(\hat{w}_k)$ is $k = h+l-2$. Indeed, it is easy to show that, given (3.38), $\boldsymbol{\Gamma}(\hat{w}_k) = \boldsymbol{\Gamma}(\hat{w}_{k-2l})$ for all $k > h+l-2$, which by assumption are positive. This explains the lower bound on the prediction horizon $\bar{N}$, which forces the optimizer to find an optimum that guarantees compliance of the PE constraint until it stops being a function of such optimum.

Given the implementation of the feasible control law $\mu^f(x(t))$ up until $\bar{t}$ and feasibility of (3.36) at time $\bar{t}$ it follows straightforwardly that the candidate control law (3.39) fulfils the hard constraints over the exciting input (3.36e). Furthermore, given the consistency constraint (3.38) and the candidate control law (3.39) it also follows that

$$\boldsymbol{\Gamma}(\hat{w}(\bar{t}+1)) = \begin{cases} \boldsymbol{\Gamma}(\hat{w}_1^\star(\bar{t})) & l \neq 1 \\ \boldsymbol{\Gamma}(\hat{w}(\bar{t}+1-2l)) & l = 1 \end{cases} \tag{3.40a}$$

$$\boldsymbol{\Gamma}(\hat{w}(\bar{t}+k)) = \boldsymbol{\Gamma}(\hat{w}_k^\star(\bar{t})) \qquad \forall k \in \left[2, \bar{N}-1\right] \tag{3.40b}$$

$$\boldsymbol{\Gamma}(\hat{w}(\bar{t}+j)) = \boldsymbol{\Gamma}(\hat{w}(\bar{t}-\lambda_j)) \qquad \forall j \geq \bar{N} \tag{3.40c}$$

with $\lambda_j \in \mathbb{N}$ for all j. Since (3.36) is assumed feasible at time $\bar{t}$ and $\mu^f(x(t))$ is assumed used up until $\bar{t}$ then $\boldsymbol{\Gamma}(\hat{w}(\bar{t}+k)) \geq 0$ for all $k \geq 1$ as defined in

(3.40). It follows then that (3.39) renders $\vec{\hat{w}}_t$ a PE sequence of order h for all $t \geq \bar{t} + 1$. ■

Again, Theorem 3.4 does not guarantee recursive feasibility of the optimization problem (3.36) with consistency constraint (3.38), but the existence of a PE inducing control law that takes over after the optimization. This is because the rationale employed here is that of returning to an off-line computed buffer sequence, while allowing the optimization to be solved at consecutive time instances would go against that and against the periodicity constraint which only accounts for a single new element. In the extreme, if the optimization is allowed to be solved at each time instant, recursive feasibility would have to be guaranteed by means of invariance of the non-linear equation that drives the PE constraint. Nevertheless, it is possible to introduce feedback more than once under this rational.

Corollary 3.2. Assume all the hypotheses of Theorem 3.4 hold. Define t_i with $i \in \mathbb{N}$ as the time instances in which the optimizer is allowed to interrupt periodicity by setting $\hat{w}(\bar{t}) = \hat{w}_0^\star(\bar{t})$ possibly different to $\hat{w}(\bar{t} - l)$. If $\bar{t}_i > \bar{t}_{i-1} + h + l - 2$ and (3.39) is employed at all time instances $t \in (t_i,\, t_{i+1})$ then $\vec{\hat{w}}_t$ is a PE sequence of order h for all $t \geq 0$ according to Lemma 3.2

Proof. The proof follows straightforwardly by noting that at time $t_{i+1} \geq t_i + h + l - 2$ the oldest value of the exciting sequence needed to verify the PE constraint is $t_i + 1$, and hence only buffer sequence values are employed. The rest follows from the proof of Theorem 3.4. ■

Note that at any time instant t_i the candidate (3.39) is still a feasible solution. Thereby, even if the optimizer is not able to find the optimum due to the non-convexity of the optimization, a solution that guarantees PE of the exciting sequence is always available.

3.4.2.3 Recursive feasibility of a non-periodic signal: second approach

Theorem 3.4 guarantees the appropriate PE order on the exciting sequence by introducing feedback under the rationale of employing the newly optimized value only once, and then returning to a previously known feasible l-periodic solution. This approach is valid only if the optimizer is allowed to interrupt the buffer sequence every $h + l - 2$ time steps, which does not help to introduce appropriate feedback. To allow the optimizer to continually provide optimized values of the exciting input, a different rationale is now proposed.

Ignore again the periodicity constrain and assume that at time 0 the optimizer chosen to solve (3.36) finds a feasible optimum with $\hat{w}_0^\star(0) \neq \hat{w}(-l)$. Since the new value is feasible, and guarantees a $\overrightarrow{\hat{w}}(0)$ of appropriate order, a valid question is whether this new value can take a place in the l-periodic buffer sequence, replacing a former value of it for all time instances in the future (rather than just at a single time instance). Similarly to the previous rationale, it is not possible to guarantee such an outcome immediately, but an appropriate design of the periodicity constraint does allow it.

Theorem 3.5. Suppose $\overrightarrow{\hat{w}_b}$ meets Assumption 3.2, that the horizon is set to $\bar{N} \geq h-1$, that the buffer sequence is employed to verify constraint (3.36b) during the initial time steps and that $\hat{w}(t) = \hat{w}_b(t)$ for all $t \in [-h-l+2, -1]$. If the periodicity constraint (3.36g) is defined as

$$\mathcal{C}(\hat{w}_k) := \begin{cases} \hat{w}_k - \hat{w}(t+k-l) & k \in [1,\ l-1] \\ \hat{w}_k - \hat{w}_{k-l} & k \in \left[l,\ \bar{N}\right]. \end{cases} \tag{3.41}$$

then

$$\hat{w}_0^f(t) = \hat{w}_0(t-l) \tag{3.42}$$

is a feasible solution for the optimization (3.42) for all $t \geq 0$. Furthermore, the sequence $\overrightarrow{\hat{w}_t}$ is persistently exciting of order h at all $t \geq 0$ according to Lemma 3.2.

Opposed to Theorem 3.4, Theorem 3.5 does claim recursive feasibility of (3.36) with consistency constraint (3.41) at any time instant. This is thanks to the rationale employed, that seeks to replace the l-periodic values of the buffer sequence computed off-line by the newly optimized values obtained by solving (3.36). Note, however, that Theorem 3.5 cannot guarantee feasibility at time $t+1$ simply given feasibility at time t, since the PE optimization is not memoryless, hence the necessity of a PE buffer sequence. This is also why the following proof of Theorem 3.5 starts by showing recursive feasibility at time $t=1$ given a feasible optimization at time $t=0$.

Proof. At any time instant t, the periodic element to be replaced by a new optimal one is $\hat{w}(t-l)$. This element appears in the computation of $\mathbf{\Gamma}(\hat{w}_k)$ for all $k \in [0, h-1)$, making $k = h-1$ the first prediction step in which it has been completely removed of the l-periodic prediction. This explains the lower bound on the prediction horizon $\bar{N}$, which forces the optimizer to find an optimum that guarantees compliance of the PE constraint until the old value is completely removed from the l-periodic buffer sequence.

Suppose then that a feasible solution exists at time 0 and refer to it as $\hat{\boldsymbol{w}}^f(0) = \left\{\hat{w}_0^f(t), \ldots, \hat{w}_{\bar{N}}^f(t)\right\}$ although only the first element would have been available for the optimizer since the periodicity constraint fixes the rest. The candidate at time 1, as given by (3.42), is then $\hat{w}_0^f(1) = \hat{w}_0(1-l) = \hat{w}_b(l-1)$, which by Assumption 3.2 meets the hard constraint (3.36e). Furthermore, the candidate solution at time 1 and the consistency constraint (3.41) result in that

$$\boldsymbol{\Gamma}\left(\hat{w}_k^f(1)\right) = \boldsymbol{\Gamma}\left(\hat{w}_{k+1}^f(0)\right) \quad \forall k \in \left[0, \bar{N}-1\right] \tag{3.43a}$$

$$\boldsymbol{\Gamma}\left(\hat{w}_{\bar{N}}^f(1)\right) = \boldsymbol{\Gamma}\left(\hat{w}_{\bar{N}-1}^f(1)\right). \tag{3.43b}$$

Since (3.36) is assumed feasible at time 0, then (3.36f) is met at time 1 by the candidate (3.42). It follows then that (3.42) is feasible at time 1. Feasibility of (3.42) for $t > 1$ follows from induction and (3.43). ■

Theorem 3.5 guarantees recursive feasibility under the rationale of employing the newly optimized value to replace the l-steps prior value and generate a completely new buffer sequence which contains an optimal value computed through feedback. This result remains valid even if the optimizer is allowed to change the buffer sequence at every time instant, as opposed to the limit on the frequency of the optimization associated to the previous rationale. Furthermore, note that Theorem 3.4 requires a prediction horizon that is, at least, l steps longer than the one required by Theorem 3.5, resulting in l more instances of the non-convex constraint that need to be evaluated by the optimizer. Thereby the rationale of complete replacement of the old value (Theorem 3.5) is regarded as an overall better approach to recursive feasibility and feedback.

The result in Theorem 3.5 depends heavily on the periodically-invariant terminal constraint (3.41). Although theoretically pleasing, constraint (3.41) allows a single element to be optimized, hence putting a possibly large lower bound on the optimal value function. This deficiency can be tackled by allowing several elements of the predicted exciting sequence to be optimized. In such a case, extending the results in Theorem 3.5 is straightforward provided a comparable increase of the prediction horizon and the periodicity constraint (3.41).

However, increasing the amount of decision variables may result in the complexity of the optimization problem growing prohibitively large given that constraint (3.36f) is non-convex. The latter is a problem even in the case of a single optimization variable, and the time restrictions inherent to the receding horizon framework imply that a solution to (3.36) may not be found suitably fast. The non-convexity of constraint (3.36f) remains as a future challenge, and it is important to note that the claims in Theorem 3.5 do not depend on the quality of the solution provided by the optimizer at each time instant. In fact, even if no optimization is performed, a feasible solution is always available by looking at the exciting input applied l steps into the past.

In general, the approach presented here to drive the exciting part of the input has similarities with the adaptive MPC architecture proposed in [74]. The most relevant difference is that in [74] the entire input is used to excite the system, and hence stability is provided as an standing assumption. Another significant difference is that [74] enforces the PE constraint (3.41) only on the first element of the optimized sequence, independent of the prediction horizon, and no terminal constraints are included. Following the fact that their demands are weaker, their recursive feasibility results are also. A similar claim to that in Proposition 3.3 is provided, however there is no guidance as to how the buffer signal should be designed, as opposed to the structure described in Assumption 3.2. Furthermore, there is no explicit analysis on whether recursive feasibility is guaranteed after a time step in which the optimizer interrupts periodicity with a new optimal value.

3.5 Model Update

Section 3.2 presented the necessary and sufficient conditions to guarantee persistence of excitation of the regressor vector and hence convergence (in deterministic fashion) of the RLS estimates to the true plant parameters. Subsequently, Section 3.4.2.2 proposed a novel receding horizon control law, driven by an MPC-like optimization, in order to achieve these conditions while minimizing the disturbance of the exciting part on the control objective. However, the fundamental goal of any adaptive controller is to employ the current converged estimates, assumed more accurate, to improve the regulation performance of the controller. When it comes to control via an MPC controller, the control performance is tightly related to the accuracy of the prediction model. In Section 3.3 tube MPC is proposed to regulate the system hence achieving stability and feasibility despite model inaccuracy, however its performance could improve given a more accurate prediction model. Indeed, smaller disturbances lead to smaller robust invariant sets, yielding the exponential stability of a smaller neighbourhood of the origin according to Theorem 3.3 (even if the set in question is never explicitly used in the MPC design).

It would then be desirable to use the current converged estimates to perform the predictions in the optimizations problems (3.23) and (3.36). It was stressed in Section 3.4 that the recursive feasibility and PE guarantees provided by the receding horizon control law related to (3.36) are not dependent on the state nor the model dynamics. Hence updating the prediction model in (3.36) can be done instantaneously at any time instant at which a more accurate representation of the plant is available. This is not the case for the prediction model in (3.23). Indeed, most properties related to the feedback law $\kappa_N(\cdot)$ (such as stability and feasibility) depend heavily on elements that were designed specifically for a certain prediction model $\left(\bar{A}, \bar{B}\right)$. This is, possibly, the main issue associated to AMPC controllers with simultaneous excitation and control guarantees, and stems from the high model dependency of the MPC technique.

Consequently, if any converged set of estimates is to be employed to update the prediction model, it is necessary to verify whether the tube MPC elements retain their properties, or guarantee so by re-computing them. In what follows three options are explored, ranging from simple verification of the assumptions in Theorem 3.3, to complete controller re-design. The focus of these approaches is placed on the fact that the plant is considered to be time varying, hence a *new* controller, employing the converged estimates as prediction model, must remain exciting and be robust to possible future plant changes (within $\mathcal{M}$).

Furthermore, none of the three approaches proposed here is necessarily better than the others, and all three can be implemented concurrently. Hereafter, and according to (3.1), the model $(\mathcal{A}(t), \mathcal{B}(t))$ refers to the RLS estimates at time t, however the time index will be dropped almost everywhere to reduce notation. Furthermore, it will be assumed that an estimate, to be a valid prediction model candidate, has to fulfil $(\mathcal{A}, \mathcal{B}) \in \mathcal{M}$. The analysis is performed for a single instance of prediction model update, assumed to be attempted at time $\bar{t}$, however the results here presented are valid for any subsequent update attempt at any time instant for which $(\mathcal{A}, \mathcal{B}) \in \mathcal{M}$.

3.5.1 A-posteriori verification

Ultimately, the objective is to retain the properties of the feedback law $\kappa_N(\cdot)$, summarized by Theorem 3.3, under a new prediction model. If the controller parameters cannot be modified on-line, the current estimates can only be used to update the prediction model if they fulfil the assumptions required by Theorem 3.3 and if the arguments employed in its proof still hold. These elements can be classified in three groups depending on the issue they tackle: (i) constraint satisfaction, (ii) recursive feasibility of the MPC optimization and (iii) stability of the control target.

3.5.1.1 Constraint satisfaction

Robust constraint satisfaction, despite the parametric uncertainty and the perturbing effects of $\hat{w}(t)$, is achieved through the robust invariance properties of the tightening set $\mathbb{S}$. The latter is computed to be robust against any perturbation coming from the compound set of disturbances $\mathbb{W}$ defined in (3.21). However, if the prediction model changes from $\left(\bar{A}, \bar{B}\right)$ to $(\mathcal{A}, \mathcal{B})$, then $\mathbb{W}_p$ and $\mathbb{W}$ also change. In view of this the following assumption is required.

Assumption 3.3. For every pair $(\mathcal{A}, \mathcal{B}) \in \mathcal{M}$ there exists a $\mathcal{C}$-set $\mathcal{W}_p\left(\mathcal{A}, \mathcal{B}\right) \subset \mathbb{R}^n$ such that $w_p = \left(A - \mathcal{A}\right)x + \left(B - \mathcal{B}\right)u \in \mathcal{W}_p$ for all $(x, u, (A, B)) \in \mathbb{X} \times \mathbb{U} \times \mathcal{M}$.

In Assumption 3.3 $\mathcal{W}_p$ is a function of $(\mathcal{A}, \mathcal{B})$, i.e. a different $\mathcal{W}_p$ is considered for every pair $(\mathcal{A}, \mathcal{B})$, however such dependency is dropped to reduce notation. If the prediction model is to be updated by $(\mathcal{A}, \mathcal{B}) \in \mathcal{M}$ while maintaining the constraint satisfaction guarantees provided by the control law $\kappa_N(\cdot)$, $\mathbb{S}$ must remain invariant for the candidate $(\mathcal{A}, \mathcal{B}) \in \mathcal{M}$ when in closed-loop with K and subject to disturbances bounded in $\mathcal{W}_p$. This is summarized in the following criterion.

Criterion 3.1. The current candidate $(\mathcal{A}, \mathcal{B})$, with associated parametric uncertainty set $\mathcal{W}_p$, fulfils

$$(\mathcal{A} + \mathcal{B}K)\,\mathbb{S} \oplus \left(\mathcal{W}_p \oplus \mathcal{B}\hat{\mathbb{W}}\right) \subseteq \mathbb{S}.$$

3.5.1.2 Recursive feasibility

Theorem 3.3 guarantees that, if the optimization is feasible at time instant $t = 0$ and the nominal loop is closed with the MPC control law $\bar{\kappa}_N\left(\cdot\right)$, then the optimization remains feasible for any time instant $t > 0$. Similarly, in order to guarantee recursive feasibility of optimization (3.23) after a prediction model update, it is not only necessary to guarantee feasibility at the transition time instant $\bar{t}$, but also for every $t > \bar{t}$.

The latter can be done by checking whether $\bar{\mathbb{X}}_f$ remains invariant for the candidate prediction model. Provided that happens, if the optimization problem is feasible at the update time $\bar{t}$, then it remains feasible for all future time instances under the same arguments of Theorem 3.3. The feasibility of the optimization (3.23) at time $\bar{t}$, however, cannot be readily guaranteed. Indeed, changing the prediction model also changes the N-step stabilizability set [97] associated to the terminal constraint set $\bar{\mathbb{X}}_f$. It follows that the feasibility regions associated to both prediction models, say $\bar{\mathcal{X}}_N\left(\bar{A}, \bar{B}\right)$ and $\bar{\mathcal{X}}_N\left(\mathcal{A}, \mathcal{B}\right)$, are not necessarily equal. Thus, even if $\bar{x}(\bar{t}-1) \in \bar{\mathcal{X}}_N\left(\bar{A}, \bar{B}\right)$ implies $\bar{x}(\bar{t}) \in \bar{\mathcal{X}}_N\left(\bar{A}, \bar{B}\right)$, the latter does not guarantee $\bar{x}(\bar{t}) \in \bar{\mathcal{X}}_N\left(\mathcal{A}, \mathcal{B}\right)$. Recursive feasibility through and after an update then can be guaranteed if the following criterion is verified.

Criterion 3.2. The current candidate $(\mathcal{A}, \mathcal{B})$ and model update time $\bar{t}$ are such that (i) the set $\bar{\mathbb{X}}_f$ remains PI for the closed-loop dynamics $\mathcal{A} + \mathcal{B}K$ and (ii) $\bar{x}(\bar{t}) \in \bar{\mathcal{X}}_N\left(\mathcal{A}, \mathcal{B}\right)$. Equivalently, that (i) $(\mathcal{A} + \mathcal{B}K)\,\bar{\mathbb{X}}_f \subseteq \bar{\mathbb{X}}_f$ and (ii) there exists a feasible solution to (3.23) at time $\bar{t}$ with (3.23d) replaced by $\bar{x}_{k+1} = \mathcal{A}\bar{x}_k + \mathcal{B}\bar{u}_k$.

As mentioned previously, $\bar{x}(\bar{t}-1) \in \bar{\mathcal{X}}_N\left(\bar{A}, \bar{B}\right)$ guarantees $\bar{x}(\bar{t}) \in \bar{\mathcal{X}}_N\left(\bar{A}, \bar{B}\right)$, but not $\bar{x}(\bar{t}) \in \bar{\mathcal{X}}_N\left(\mathcal{A}, \mathcal{B}\right)$. As opposed to standard MPC implementations then, it is not possible to use the tail of the solution at time $\bar{t}-1$ to demonstrate part (ii) of Criterion 3.2. Nevertheless, there are at least two ways of verifying it. Indeed, a simple approach would be to characterize the set $\bar{\mathcal{X}}_N\left(\mathcal{A}, \mathcal{B}\right)$ and test for inclusion, however the computational complexity of computing the N-step stabilizability set grows rapidly with the size of the problem (a more detailed discussion about this is provided in Section 4.3.1.2).

Alternatively, note that Criterion 3.2 does not demand for a solution to (3.23) at time $\bar{t}$ nor for the computation of the set $\bar{\mathcal{X}}_N\left(\mathcal{A}, \mathcal{B}\right)$, but merely for verifying if (3.23) is feasible at state $\bar{x}(\bar{t})$. This can be easily verified by solving the linear program

$$
\begin{aligned}
& \max_{\bar{\boldsymbol{u}}, \bar{\beta}} \bar{\beta} \\
\text{s.t. (for } k = 0, \ldots, N-1) \quad & \\
& \bar{x}_0 = \bar{\beta}\bar{x}(\bar{t}) \\
& \bar{x}_{k+1} = \mathcal{A}\bar{x}_k + \mathcal{B}\bar{u}_k \\
& \bar{x}_k \in \bar{\mathbb{X}} \\
& \bar{u}_k \in \bar{\mathbb{U}} \\
& \bar{x}_N \in \bar{\mathbb{X}}_f \subseteq \bar{\mathbb{X}},
\end{aligned}
\tag{3.44}
$$

where $\boldsymbol{u}$ is the optimization variable associated to (3.23) and the optimum is denoted by $\left(\boldsymbol{u}^\star, \bar{\beta}^\star\right)$. Given the constraints employed, the linear program (3.44) produces an optimum such that the MPC optimization is feasible at $\bar{x}_0 = \bar{\beta}^\star \bar{x}(\bar{t})$; that is, there exists a sequence of N control inputs that drives said state to the terminal set while respecting tightened state and input constraints. Furthermore, given that the original constraint sets are $\mathcal{PC}$-sets and the disturbance set is assumed a $\mathcal{C}$-set, the tightening set $\mathbb{S}$ can be designed to be a $\mathcal{PC}$-set as well. Moreover, if $\bar{\mathbb{X}}_f$ is designed as a $\mathcal{PC}$-set, it follows that $\bar{\mathcal{X}}_N(\mathcal{A}, \mathcal{B})$ is a $\mathcal{C}$-set, thus $\bar{x}(\bar{t}) \in \bar{\mathcal{X}}_N(\mathcal{A}, \mathcal{B})$ if and only if $\bar{\beta}^\star \geq 1$.

3.5.1.3 Stability

Finally, Theorem 3.3 guarantees that the set $\mathcal{R} = \mathbb{S} \times \{\mathbf{0}_n\}$ is exponentially stable for the composite closed-loop by ensuring that the optimal cost function is a Lyapunov function for the nominal closed-loop. The upper and lower bounds in (3.24a) do not depend on the prediction model and thus still hold at any time $t \geq \bar{t}$, however the upper bound on the optimal cost variation in (3.24b) does depend on the prediction model. Analogously to the discussion in Section 3.5.1.2 then, in order to guarantee exponential stability after a prediction model update it is necessary to verify the appropriate cost decrease at the update time $\bar{t}$ and at any time in the future.

For all future time instances this can be done by checking whether the matrix inequality of Theorem 3.3–(d) holds for the candidate prediction model. Provided that this happens, it holds that $\Delta V_N(\bar{x}(t)) \leq -\ell\left(\bar{x}_0^\star(\bar{x}(t)), \bar{u}_0^\star(\bar{x}(t))\right)$ for all $t > \bar{t}$. At the time of update, however, this is not necessarily the case because the compared cost functions depend on two different prediction models and the tail of a previously feasible solution is not feasible anymore (as seen in Section 3.5.1.2). Lypaunov stability through and after an update then can be guaranteed if the following criterion is verified.

Criterion 3.3. The current candidate $(\mathcal{A}, \mathcal{B})$ and model update time $\bar{t}$ are such that (i) $\mathcal{A}_K^\top P \mathcal{A}_K + Q + K^\top R K \leq P$ holds with $\mathcal{A}_K = \mathcal{A} + \mathcal{B}K$, and (ii) $V_N(\mathcal{A}, \mathcal{B}, \bar{x}(\bar{t})) - V_N\left(\bar{A}, \bar{B}, \bar{x}(\bar{t}-1)\right) \leq -\ell\left(\bar{x}_0^\star(\bar{x}(\bar{t}-1)), \bar{u}_0^\star(\bar{x}(\bar{t}-1))\right)$.

The cost decrease at the update time is explicitly required by part (ii) of Criterion 3.3, rather than implicitly attained through properties of the terminal cost function. Furthermore, the same upper bound for the cost variation is requested. This is to guarantee the same rate of convergence, however any cost decrease would be enough to guarantee stability throughout the update instant. Part (i) of Criterion 3.3 is required to guarantee that the cost decrease is continued to be met after a change in prediction model.

3.5.1.4 Admissibility of a model update

The main goal of this first approach to prediction model update is to not modify the controller. The properties of the control law $\kappa_N(\cdot)$, however, depend on elements computed for the specific pair $\left(\bar{A}, \bar{B}\right)$, and if they are to remain fixed, it is necessary to verify whether they retain their properties for the new candidate $(\mathcal{A}, \mathcal{B})$. Criteria 3.1, 3.2 and 3.3 provide sufficient conditions for this to happen.

Proposition 3.4. Suppose that at time $\bar{t}$ the estimates have converged to $(\mathcal{A}(\bar{t}), \mathcal{B}(\bar{t})) \in \mathcal{M}$. If the candidate for model update $(\mathcal{A}(\bar{t}), \mathcal{B}(\bar{t}))$ fulfils Criteria 3.1, 3.2 and 3.3, then the prediction model used in the MPC optimization (3.23d) can be replaced by

$$\bar{x}_{k+1} = \mathcal{A}(\bar{t})\bar{x}_k + \mathcal{B}(\bar{t})\bar{u}_k.$$

while guaranteeing constraint satisfaction and stability as depicted in Theorem 3.3 for all times $t \geq \bar{t}$.

Proof. The proof follows from the discussion around Criteria 3.1, 3.2 and 3.3, and the proof of Theorem 3.3. ■

Proposition 3.4 presents a series of comprehensive tests that need to be performed at any time instant at which the estimates are to be employed as a new prediction model. However, the order in which the different criteria is verified does matter. Indeed, albeit the standing assumption of a slowly varying system, parts (ii) of Criteria 3.2 and 3.3 not only depend on the candidate for prediction model, but also on the current time instant. Thus if a model update is to take place at time instant $\bar{t}$, both conditions need to be verified in the time between measuring the state and solving the MPC optimization. Clearly then, once a candidate $(\mathcal{A}(\bar{t}), \mathcal{B}(\bar{t}))$ is identified, it is convenient to verify all the other criteria first, in order to avoid spending unnecessary resources in the on-line testing of parts (ii) of Criteria 3.2 and 3.3.

It is important to note that there exists a finite time instant $\tilde{t} \geq 0$ such that part (ii) of Criterion 3.2 is met; this is independent of the current prediction model $\left(\bar{A}, \bar{B}\right)$ and the candidate for update $(\mathcal{A}, \mathcal{B})$. Indeed, from the properties of the constraint sets, and an appropriate design of the invariant sets, $\bar{\mathcal{X}}_N$ can be guaranteed to be a $\mathcal{PC}$-set. It follows then that $\bar{\mathcal{X}}_N\left(\bar{A},\bar{B}\right)\cap\bar{\mathcal{X}}_N\left(\mathcal{A},\mathcal{B}\right)$ has a non-empty interior. Given this and the exponential stability of the nominal trajectories, it follows that there exists a finite time $\hat{t}\geq 0$ such that $\bar{x}(t)$ is inside $\bar{\mathcal{X}}_N\left(\mathcal{A},\mathcal{B}\right)$ for all $t\geq\hat{t}$ (further discussion on this is provided in Chapter 4).

Notwithstanding the above fact, it is easy to show that for a given set $\mathcal{M}$ there maybe only a limited subset of models $\bar{\mathcal{M}}\subset\mathcal{M}$ for which Criterion 3.1 is met, that is, for which the set $\mathbb{S}$ remains RPI in view of the new set of disturbances and the updated model dynamics. The same happens for the invariance of the terminal set. Ultimately this means that, although a given $(\mathcal{A}(\bar{t}),\mathcal{B}(\bar{t}))\in\mathcal{M}$ may represent the true plant dynamics for all $t\geq\bar{t}$, such candidate may not be allowed to update the prediction model without incurring some modifications to the controller.

3.5.1.5 Criterion relaxation

Another approach to classifying the conditions listed in Criteria 3.1, 3.2 and 3.3 is according to what they depend on. Following this rationale, Criterion 3.1, part (i) of Criterion 3.2 and part (i) of Criterion 3.3 depend solely on the estimates that are a candidate for prediction model updating, thus if they are not met, the former cannot be employed unless the controller is re-designed. On the other hand, parts (ii) of Criteria 3.2 and 3.3 depend not only on the prediction model but also on the current nominal state value $\bar{x}(\bar{t})$, which is fictitious. Indeed, the nominal trajectories are initialized through optimization but left to evolve independently thereafter, following the discussion in Section 2.3.2. However if $\bar{x}(t)$ is regarded as an optimization variable at time $\bar{t}$ the chances of meeting part (ii) of Criteria 3.2 and 3.3 may increase. To see this define the following auxiliary optimization problem

$$
\begin{aligned}
&\mathbb{Q}_N(x(t))\colon && \min_{\bar{\boldsymbol{u}},\bar{x}_0} J_N\left(\bar{\boldsymbol{u}},\bar{x}_0\right)\\
&\text{s.t. (for } k=0,\ldots,N-1) && \\
& && x(t)-\bar{x}_0\in\mathbb{S}\\
& && \bar{x}_{k+1}=\mathcal{A}\bar{x}_k+\mathcal{B}\bar{u}_k\\
& && \bar{x}_k\in\bar{\mathbb{X}}\\
& && \bar{u}_k\in\bar{\mathbb{U}}\\
& && \bar{x}_N\in\bar{\mathbb{X}}_f\subseteq\bar{\mathbb{X}},
\end{aligned}
$$

with associated optimum and optimal value of the cost function respectively defined by

$$\begin{aligned}\big(\bar{\boldsymbol{u}}^{\circ}(x(t)), \bar{x}_0^{\circ}(x(t))\big) &= \arg \mathbb{Q}_N(x(t)) \\ V_N^{\circ}(x(t)) &= J_N(\bar{\boldsymbol{u}}^{\circ}(x(t)), \bar{x}_0^{\circ}(x(t))).\end{aligned}$$

The following result holds.

Proposition 3.5. If $\mathbb{P}_N(\mathcal{A}, \mathcal{B}, x(\bar{t}))$ is feasible, then $\mathbb{Q}_N(x(\bar{t}))$ is feasible and $V_N^{\circ}(x(\bar{t})) \leq V_N(\mathcal{A}, \mathcal{B}, \bar{x}(\bar{t}))$. Furthermore, $\mathbb{Q}_N(x(\bar{t}))$ may be feasible even if $\mathbb{P}_N(\mathcal{A}, \mathcal{B}, x(\bar{t}))$ is not.

Proof. The proof is trivial and follows form the fact that $\mathbb{P}_N(\mathcal{A}, \mathcal{B}, x(\bar{t}))$ is equal to $\mathbb{Q}_N(x(\bar{t}))$ but the latter has an additional optimization variable. For the first part suppose $\mathbb{P}_N(\mathcal{A}, \mathcal{B}, x(\bar{t}))$ is feasible with solution $\bar{\boldsymbol{u}}^{\star}(\bar{x}(\bar{t}))$, it follows from the invariance of $\mathbb{S}$ that the pair $(\bar{\boldsymbol{u}}^{\star}(\bar{x}(\bar{t})), \bar{x}(\bar{t}))$ is a feasible, yet not necessarily optimal solution for $\mathbb{Q}_N(x(\bar{t}))$. It then holds that $V_N^{\circ}(x(\bar{t})) \leq J_N(\bar{\boldsymbol{u}}^{\star}(\bar{x}(\bar{t})), \bar{x}(\bar{t})) = V_N(\mathcal{A}, \mathcal{B}, \bar{x}(\bar{t}))$.

To verify the second part note that to ensure feasibility of $\mathbb{P}_N(\mathcal{A}, \mathcal{B}, x(\bar{t}))$ it is necessary and sufficient that $\bar{x}(\bar{t}) \in \bar{\mathcal{X}}_N(\mathcal{A}, \mathcal{B})$, which then implies $x(\bar{t}) \in \{\bar{x}(\bar{t})\} \oplus \mathbb{S}$. On the other hand, to ensure feasibility of $\mathbb{Q}_N(x(\bar{t}))$, it is necessary and sufficient that $x(\bar{t}) \in \bar{\mathcal{X}}_N(\mathcal{A}, \mathcal{B}) \oplus \mathbb{S}$, resulting in a larger feasibility region at time $\bar{t}$ for $\mathbb{Q}_N(x(\bar{t}))$. ■

It follows from Proposition 3.5 that by solving $\mathbb{Q}_N(\cdot)$ (instead of instead of $\mathbb{P}_N(\cdot)$) at any time instant in which a prediction model change is attempted, part (ii) of Criteria 3.2 and 3.3 are more likely to be met. In fact, more can be said about the likelihood of meeting part (ii) of Criterion 3.3 if $\mathbb{Q}_N(\cdot)$ is solved at an intended update time. This is summarized in the following corollary to Proposition 3.5.

Corollary 3.3. Assume that $\hat{t}$ is such that $x(t) \in \bar{\mathcal{X}}_N(\mathcal{A}, \mathcal{B}) \oplus \mathbb{S}$ for all $t \geq \hat{t}$ and so feasibility of $\mathbb{Q}_N(x(t))$ is guaranteed for all $t \geq \hat{t}$. If for any $t \geq \hat{t}$ it holds that $x(t) \in \mathbb{S}$, then part (ii) of Criterion 3.3 is met immediately at said time instant and for any future one.

Proof. The proof follows from the optimality of $\mathbb{Q}_N(x(t))$. Indeed, it is easy to show that for any $x(t) \in \mathbb{S}$ the optimal nominal state is $\bar{x}(t) = \bar{x}_0^{\star}(x(t)) = \mathbf{0}$ with associated optimal input sequence $\bar{\boldsymbol{u}}^{\star}(x(t)) = \mathbf{0}$. This results in $V_N^{\circ}(x(t)) = 0$, reducing the the cost decrease requirement to

$$-V_N\left(\bar{A}, \bar{B}, \bar{x}(\bar{t}-1)\right) \leq -\ell\left(\bar{x}_0^\star\left(\bar{x}(\bar{t}-1)\right), \bar{u}_0^\star\left(\bar{x}(\bar{t}-1)\right)\right),$$

which is always met. Furthermore, given the invariance of $\mathbb{S}$ and the tube MPC composite control law $\kappa_N(\cdot)$, the same is valid for all future time instances. ■

Finally, note that this approach to criterion relaxation would combine both variants of the tube MPC described in Section 2.3 throughout the operation time of the plant. Nevertheless, Corollary 2.2 ensures that the set $\mathbb{S}$ remains exponentially stable for the true dynamics, as long as the cost decrease condition in Criterion 3.3 is met.

3.5.2 Robust design

It was shown that the likelihood of meeting parts (ii) of Criteria 3.2 and 3.3 can be increased by solving an auxiliary OCP at each time instant an update is attempted. However, this is not enough to guarantee that updating the prediction model will be possible. This is because the requirements relating to set invariance might not be met by any converged estimates provided by the RLS algorithm. It could be argued that, once an accurate set of estimates has been obtained, robustness to parametric uncertainty ceases to be necessary. Thereby certain demands for prediction model update could be lowered, such as the robust invariance of $\mathbb{S}$. However the plant is assumed to be time-varying within $\mathcal{M}$, thus Criterion 3.1 needs to be verified, alongside parts (i) of Criteria 3.2 and 3.3, to allow for a prediction model update.

In order to increase the chance of the aforementioned conditions being met, but without resorting to the on-line re-designing of the controller, the initial design process could be robustified with respect to the parametric uncertainty, or at least a subset of it, say $\tilde{\mathcal{M}} \subset \mathcal{M}$. Several approaches could be proposed to achieve so, however one of the main features of the proposed dual controller is its simplicity of design, and the aim of the following proposals is to maintain it. The following definition is now required.

Definition 3.5. Given a scalar $\lambda \in (0,1)$, the set $\mathbb{S}$ is called a robust λ-contractive set for the closed loop $\bar{A}_K$ and disturbance set $\mathbb{W}$ if for every $x \in \mathbb{S}$, $A_K x + w \in \lambda\mathbb{S}$ for all $w \in \mathbb{W}$; equivalently if $A_K\mathbb{S} \oplus \mathbb{W} \subseteq \lambda\mathbb{S}$. If $\mathbb{W} = \{0\}$ the adjective *robust* is dropped from the definition.

Note that every robust λ-contractive set is RPI, and every λ-contractive set is PI. In what follows, these sets are employed to provide a constructive proposal for robustifying the initial design of the tube MPC parameters. Note also that every λ-contractive set is RPI for some level of disturbance. Indeed, suppose $\mathbb{T}$ is a $\mathcal{PC}$-set and a λ-contractive set for $\bar{A}_K$. Since $\lambda \in (0,1)$ it follows that $\bar{A}_k\mathbb{T} \subseteq \lambda\mathbb{T} \subset \mathbb{T}$. It follows that there exists $r > 0$ such that $\lambda\mathbb{T} \oplus \mathcal{B}_r \subseteq \mathbb{T}$, hence $\bar{A}_k\mathbb{T} \oplus \mathcal{B}_r \subseteq \mathbb{T}$.

3.5.2.1 Terminal constraint

Parts (i) of Criteria 3.2 and 3.3 refer to the verification of the terminal conditions which are usually employed in MPC implementations to guarantee recursive feasibility of the optimization and stability of the control target. Particularly, the terminal constraint set is computed such that $\bar{A}_K\bar{\mathbb{X}}_f \subseteq \bar{\mathbb{X}}_f$ where the linear feedback K is stabilizing for $\left(\bar{A}, \bar{B}\right)$, but also for every other model in $\mathcal{M}$ given an admissible RPI set according to Proposition 3.2. Suppose now that the terminal constraint set is computed not as a mere PI set but as a λ-contractive set for the closed-loop dynamics, then it holds that $\bar{A}_K\bar{\mathbb{X}}_f \subseteq \lambda\bar{\mathbb{X}}_f$. In view of this, for all $\bar{x} \in \bar{\mathbb{X}}_f$ and for any $(\mathcal{A}, \mathcal{B}) \in \mathcal{M}$ it follows that

$$\mathcal{A}_K\bar{x} = \mathcal{A}_K\bar{x} + \left(\bar{A}_K - \bar{A}_K\right)\bar{x} \tag{3.45a}$$

$$= \bar{A}_K\bar{x} + \left(\mathcal{A}_K - \bar{A}_K\right)\bar{x} \tag{3.45b}$$

$$\implies \mathcal{A}_K\bar{x} \in \lambda\bar{\mathbb{X}}_f \oplus \left(\mathcal{A}_K - \bar{A}_K\right)\{\bar{x}\} \subset \lambda\bar{\mathbb{X}}_f \oplus \left(\mathcal{A}_K - \bar{A}_K\right)\bar{\mathbb{X}}_f. \tag{3.45c}$$

Assume now that

$$\left(\mathcal{A}_K - \bar{A}_K\right)\bar{\mathbb{X}}_f \subseteq (1-\lambda)\,\bar{\mathbb{X}}_f, \tag{3.46}$$

it follows that $\mathcal{A}_K\bar{\mathbb{X}}_f \subseteq \bar{\mathbb{X}}_f$ and hence the terminal set remains invariant for the new candidate for prediction model. According to (3.45) then, if the terminal constraint set is constructed as a λ-contractive set for $\left(\bar{A}, \bar{B}\right)$, there exists a non-empty neighbourhood of $\left(\bar{A}, \bar{B}\right)$, say $\tilde{\mathcal{M}}(\lambda) \subseteq \mathcal{M}$, such that the terminal set is PI for all $(\mathcal{A}, \mathcal{B}) \in \tilde{\mathcal{M}}(\lambda)$ when in close-loop with K. As previously discussed, this is because every λ-contractive set is also an RPI set, in this case robust to disturbances stemming from the parametric uncertainty within the terminal set and the associated time-invariant feedback law K.

The drawback of computing the terminal constraint as a λ-contractive set is that the latter is a PI set by definition, thus it is contained within the corresponding maximal PI set [98], resulting in a generally smaller RoA for the controller designed for $\left(\bar{A}, \bar{B}\right)$. Nevertheless, this proposal allows for a-priori certainty about a set of models $\bar{\mathcal{M}}(\lambda)$ for which the terminal set remains invariant. Furthermore, computing the maximal λ-contractive set contained within a certain constraint set, say $\bar{\mathbb{X}}$, is as computationally intensive as computing the associated maximal PI sets [99].

An additional disadvantage is that the λ-contractive set $\bar{\mathbb{X}}_f$ depends indeed on λ, hence verifying (3.46) can only be done after $\bar{\mathbb{X}}_f$ has been computed for a given contraction factor. It follows that $\tilde{\mathcal{M}}(\lambda)$ can only be computed after $\bar{\mathbb{X}}_f$ is known. If the goal is to achieve guaranteed invariance for an arbitrarily defined subset $\tilde{\mathcal{M}}$, then an iterative design process is required. Consider a given target subset $\tilde{\mathcal{M}}$ for which guaranteed invariance is required and define $\mathbb{W}_{\tilde{\mathcal{M}}}$ as a $\mathcal{C}$-set that contains the terminal parametric uncertainty induced by a model update within $\tilde{\mathcal{M}}$, that is $\left(\mathcal{A}_K - \bar{A}_K\right)\bar{\mathbb{X}}_f \subseteq \mathbb{W}_{\tilde{\mathcal{M}}}$ for all $(\mathcal{A}, \mathcal{B}) \in \tilde{\mathcal{M}}$. It follows that given a target subset $\tilde{\mathcal{M}}$ and a tightening constant λ the sets $\bar{\mathbb{X}}_f$ and $\mathbb{W}_{\tilde{\mathcal{M}}}$ can be computed; if $\mathbb{W}_{\tilde{\mathcal{M}}} \nsubseteq (1-\lambda)\,\bar{\mathbb{X}}_f$, recompute with a smaller λ.

Termination of such iterative algorithm is highly model dependent, and can only be guaranteed if λ is allowed to be in $[0, 1)$, however $\lambda = 0$ yields the most conservative result with a terminal equality constraint, leading to the smallest possible RoA. However, if the objective is merely to increase the chances of part (i) of Criterion 3.2 to be met, it is only necessary to design $\bar{\mathbb{X}}_f$ as a λ-contractive set for an arbitrarily chosen λ.

There exists, however, a non-iterative another approach for computing the terminal set in a way that remains invariant for a subset of the uncertainty $\tilde{\mathcal{M}}$. Indeed, the terminal set is not used for tightening, as the RPI set, hence the drawbacks discussed in Section 3.3.1 are not relevant. The terminal set could then be computed as a PI set for the pLDI discussed in Section 3.3.1 following the approach in [93], which guarantees maximum volume and invariance to all possible candidates $(\mathcal{A}, \mathcal{B}) \in \tilde{\mathcal{M}}$. Taking a pLDI approach to compute a comprehensive PI set would, in theory, yield the least conservative terminal set, yet possibly intractable to compute. Indeed, algorithms such as the one in [93] do not compute the absolute maximal PI set for a pLDI, but the maximum volume PI set given an arbitrarily polytopic complexity for its representation. It follows then that, although less conservative, this approach could be intractable when employed to outperform the previous ones.

Independently of the approach, a tighter terminal constraint set is expected when attempting to guarantee its invariance for a subset $\tilde{\mathcal{M}}$ of the model uncertainty. Ultimately, this represents the trade-off between size of the initial RoA and the ability to meet part (i) of Criterion 3.2.

3.5.2.2 Terminal cost

The weight matrix that defines the terminal cost in (3.23) is designed according to the matrix inequality in hypothesis (d) of Theorem 3.3. This is done to guarantee that the terminal cost inequality (3.30) is met at all times, which in turn allows to show that the MPC optimal cost function is a Lyapunov function for the closed-loop nominal state trajectories. Note however, that the cost decrease on which the Lyapunov property depends does not rely on said inequality at the update time $\bar{t}$. Indeed, the comparison of costs at the update time involves different models, hence a decrease is explicitly requested by part (ii) of Criterion 3.3, and not implicitly verified by hypothesis (d) of Theorem 3.3.

If the model is indeed updated at time $\bar{t}$, the cost decrease for all $t > \bar{t}$ does not depend on quantities associated to the previous model. It follows that an entirely new terminal cost matrix for the candidate model could be computed in order to guarantee a cost decrease after the update. In other words, the verification step in part (i) of Criterion 3.3 can be replaced by a computation step to define a new terminal weight matrix $\mathcal{P}$ such that

$$\mathcal{A}_K^\top \mathcal{P} \mathcal{A}_K + Q + K^\top R K \leq \mathcal{P} \tag{3.47}$$

holds. This, of course, has an impact on part (ii) of Criterion 3.3 however not a necessarily detrimental one, since the latter does not explicitly depends on how the terminal cost are defined for times $\bar{t}$ and $\bar{t}-1$.

Replacing the verification in part(i) of Criterion 3.3 by a computation step implies an on-line modification of the controller, however one that is not necessarily prohibitively complex. Indeed, for a given stable closed-loop $\mathcal{A}_K$, and positive definite weight matrices Q and R, there exists a unique and positive definite solution to the discrete Lyapunov equation $\mathcal{A}_K^\top \mathcal{P} \mathcal{A}_K + Q + K^\top R K - \mathcal{P} = 0$. Furthermore, computing a matrix $\mathcal{P}$ such that (3.47) holds with strict inequality is also simple. Indeed, for any $\gamma > 1$ the discrete Lyapunov equation $\mathcal{A}_K^\top \mathcal{P} \mathcal{A}_K + \gamma \left(Q + K^\top R K\right) - \mathcal{P} = 0$ also has a unique solution since $\gamma \left(Q + K^\top R K\right) = \tilde{Q} + K^\top \tilde{R} K$ with $\tilde{Q} = \gamma Q > 0$ and $\tilde{R} = \gamma R > 0$. If follows then that

$$\begin{aligned}
& \mathcal{A}_K^\top \mathcal{P} \mathcal{A}_K + \gamma \left(Q + K^\top R K\right) - \mathcal{P} && = 0 \\
\implies & \mathcal{A}_K^\top \mathcal{P} \mathcal{A}_K - \mathcal{P} && = -\gamma \left(Q + K^\top R K\right) \\
\implies & \mathcal{A}_K^\top \mathcal{P} \mathcal{A}_K + Q + K^\top R K - \mathcal{P} && = (1-\gamma) \left(Q + K^\top R K\right)
\end{aligned}$$

$$\implies \mathcal{A}_K^\top \mathcal{P} \mathcal{A}_K + Q + K^\top R K - \mathcal{P} \qquad < 0.$$

Thus computing a terminal cost that fulfils (3.47) with strict inequality amounts to solving one instance of the discrete Lyapunov equation with Q and R scaled by a particular $\gamma > 1$.

3.5.2.3 RPI set for tightening

The likelihood of meeting Criterion 3.1 can be increased through arguments similar to those employed for the case of the terminal constraint set; that is, designing the RPI set $\mathbb{S}$ to account for possible changes in the plant and its corresponding effects on its invariance. Suppose that $\mathbb{S}$ is computed as an admissible RPI set for the nominal closed-loop $\bar{A}_K$ under disturbances arising from the composite set $\mathbb{W}$ defined in (3.21). For any $(\mathcal{A}, \mathcal{B}) \in \mathcal{M}$ that fulfils Assumption 3.3 and for all $e \in \mathbb{S}$, $\tilde{w} \in \mathcal{B}\hat{\mathbb{W}} \oplus \mathcal{W}_p$ it follows that

$$\begin{aligned} \mathcal{A}_K e + \tilde{w} &= \mathcal{A}_K e + \tilde{w} + \left(\bar{A}_K - \bar{A}_K\right) e \\ &= \bar{A}_K e + \left(\tilde{w} + \left(\mathcal{A}_K - \bar{A}_K\right) e\right) \\ \implies \mathcal{A}_K e + \tilde{w} &\in \bar{A}_K \mathbb{S} \oplus \left(\mathcal{B}\hat{\mathbb{W}} \oplus \mathcal{W}_p \oplus \left(\mathcal{A}_K - \bar{A}_K\right) \mathbb{S}\right). \end{aligned}$$

Clearly, if

$$\left(\mathcal{B}\hat{\mathbb{W}} \oplus \mathcal{W}_p \oplus \left(\mathcal{A}_K - \bar{A}_K\right) \mathbb{S}\right) \subseteq \mathbb{W} \tag{3.48}$$

then $\mathbb{S}$ remains as an RPI set for the candidate $(\mathcal{A}, \mathcal{B})$ and its corresponding disturbance set $\mathcal{W}_p \oplus \mathcal{B}\hat{\mathbb{W}}$.

Similarly to the terminal constraint case, it is possible to use (3.48) to guarantee robust invariance of the set $\mathbb{S}$ for a certain subset of models $\tilde{\mathcal{M}}$. A first approach to do so would be to artificially increase the size of the disturbance set, say to $\tilde{\mathbb{W}} \supset \mathbb{W}$. Note however that, similarly to (3.46), the left hand side of the inclusion in (3.48) depends on the RPI set $\mathbb{S}$, which in turn depends on the enlarged set $\tilde{\mathbb{W}}$. Thereby the uncertainty subset $\tilde{\mathcal{M}}$ for which robust invariance of $\mathbb{S}$ is guaranteed, (thanks to the enlarging of the disturbance set) could only be computed after the RPI set is known. Again then, if the goal is to guarantee robust invariance of $\mathbb{S}$ for an arbitrarily defined $\tilde{\mathcal{M}}$, an iterative design procedure is required.

However, arbitrarily increasing the size of the disturbance set $\mathbb{W}$ is not necessarily meaningful or conducing to a tight verification of (3.48). A more constructive approach would be to compute the RPI set $\mathbb{S}$ as a robust λ-contractive set [99]. This is similar to the approach taken for the terminal constraint however the effect that these two sets have over the nominal RoA is opposite, hence it might be desirable to compute, if exists, the minimal robust λ-contractive set associated to the disturbance set and nominal closed-loop $\bar{A}_K$. The latter can be computed following an iteration similar to the one used to compute the non-contractive minimal RPI set. Indeed, define the sequence

$$\mathbb{F}_{t+1} = \mathbb{F}_t \oplus \frac{\bar{A}_K^t \mathbb{W}}{\lambda^{t+1}} \tag{3.49}$$

with $\mathbb{F}_0 = \{0\}$, then the following results holds.

Theorem 3.6. Assume that A_K is strictly stable and $\mathbb{W}$ is a $\mathcal{C}$-set. If the contraction factor λ is such that $\bar{A}_K/\lambda$ is strictly stable, then (i) $\mathbb{F} = \lim_{t\to\infty} \mathbb{F}_t$ exists, (ii) is a $\mathcal{C}$-set, and (iii) is the minimal λ-contractive set for $\bar{A}_K$ under disturbances bounded in $\mathbb{W}$.

Proof. The proof follows directly form the proof of Theorem 4.1 in [96], and hinges on the fact that (3.49) is a Cauchy sequence. ■

Even though the minimal λ-contractive set exists and is known to be the limit of (3.49), it might be computationally intractable to compute. If the hypotheses of Theorem 3.6 hold, then robust λ-contractive outer approximations of $\mathbb{F}$ can be computed in a finite number of iterations following the results in [100]. Nevertheless, the Minkowski sum scales poorly with the size of the plant, possibly rendering even such approximations intractable to compute.

An alternative is to employ the algorithm proposed in [99] to compute the maximal robust λ-contractive set that is contained within an arbitrary compact set. Suppose that the minimal RPI set associated to $\bar{A}_K$ and $\mathbb{W}$ is admissible and refer to it by $\mathbb{S}_m$. The following result holds

Proposition 3.6. If $\mathcal{X} \subset \mathbb{S}_m \subseteq \mathbb{X}$, the maximal robust λ-contractive set inside $\mathcal{X}$, say $\mathbb{S}_M^\lambda(\mathcal{X})$, is empty.

Proof. Any robust λ-contractive set associated to $\bar{A}_K$ and $\mathbb{W}$ is, by definition, also RPI. Since the minimal RPI is contained within every other RPI set, it must happen that $\mathbb{S}_m \subseteq \mathbb{S}_M^\lambda \subseteq \mathcal{X} \subset \mathbb{S}_m$ which is a contradiction. ■

Following Proposition 3.6, a small (although not necessarily minimal) robust λ-contractive set can be computed as the maximal robust λ-contractive set associated to a small subset of the state constraints that contains $\mathbb{S}_m$. Nevertheless, this could still lead to an empty set $\mathbb{S}_M^\lambda(\mathcal{X})$ depending on the value of λ and the size of $\mathcal{X}$.

Independent of the approach, assume that a certain non-empty and admissible robust λ-contractive set, say $\mathbb{S}^\lambda$, exists. The following holds for all $e \in \mathbb{S}^\lambda$, $w \in \mathbb{W}$ and $\tilde{w} \in \mathcal{W}_p \oplus \mathcal{B}\hat{\mathbb{W}}$

$$\begin{aligned}
\mathcal{A}_K e + \tilde{w} \quad &= \mathcal{A}_K e + \tilde{w} + \left(\bar{A}_K - \bar{A}_K\right) e + (w - w) \\
&= \bar{A}_K e + w + \left(\tilde{w} - w + \left(\mathcal{A}_K - \bar{A}_K\right) e\right) \\
&= \bar{A}_K e + w + \left(\left(\mathcal{B} - \bar{B}\right) \hat{w} + \underbrace{\tilde{w}_p - w_p}_{\bar{w}_p} + \left(\mathcal{A}_K - \bar{A}_K\right) e\right) \\
\Longrightarrow \mathcal{A}_K e + \tilde{w} \quad &\in \lambda \mathbb{S}^\lambda \oplus \left(\left(\mathcal{B} - \bar{B}\right) \hat{\mathbb{W}} \oplus \bar{\mathcal{W}}_p \oplus \left(\mathcal{A}_K - \bar{A}_K\right) \mathbb{S}^\lambda\right).
\end{aligned}$$

It follows that if

$$\left(\mathcal{B} - \bar{B}\right) \hat{\mathbb{W}} \oplus \bar{\mathcal{W}}_p \oplus \left(\mathcal{A}_K - \bar{A}_K\right) \mathbb{S}^\lambda \subseteq (1 - \lambda) \mathbb{S}^\lambda$$

then $\mathbb{S}^\lambda$ remains as an RPI set for the candidate $(\mathcal{A}, \mathcal{B})$ and its corresponding disturbance set $\mathcal{W}_p \oplus \mathcal{B}\hat{\mathbb{W}}$. Ultimately this implies that there exists a non-empty set $\tilde{\mathcal{M}} \subseteq \mathcal{M}$ of candidate models for which $\mathbb{S}^\lambda$ remains robust invariant with respect to their own uncertainties. Again, however, the size of this set can only be estimated once $\mathbb{S}^\lambda$ is known and hence an iterative procedure is needed to compute a λ-contractive RPI set that retains invariance for a given arbitrary subset $\tilde{\mathcal{M}}$.

Note that the disturbance $\bar{w}_p$ is defined as the difference between the parametric uncertainty arising from each prediction model, hence $\bar{w}_p = \left(\mathcal{A} - \bar{A}\right) x + \left(\mathcal{B} - \bar{B}\right) u$. It follows that the set $\bar{\mathcal{W}}_p$, which contains said difference, is contained in the interior of $\mathbb{W}_p$.

The main drawback of using the maximal robust λ-contractive set approach is that the existence and size of $\mathbb{S}^\lambda_M(\mathcal{X})$ depend on two arbitrarily chosen variables, that is λ and $\mathcal{X}$, which may result in a lengthy design process. Also note that, independent of the approach (either increasing the size of $\mathbb{W}$ or computing a robust λ-contractive set), the resulting set for tightening will be larger, yielding a smaller nominal RoA $\bar{\mathcal{X}}_N$. The RoA of the true controller, however, is $\mathcal{X}_N = \bar{\mathcal{X}}_N \oplus \mathbb{S}$, thus a smaller $\bar{\mathcal{X}}_N$ does not necessarily yield a correspondingly smaller $\mathcal{X}_N$ since $\mathbb{S}$ would be larger. Nevertheless, a smaller $\bar{\mathcal{X}}_N$ reduces the control authority of the nominal MPC control law $\bar{\kappa}_N(\cdot)$ and hands it over to the linear tube gain K, possibly resulting in a decrease of performance.

3.5.3 Controller re-design

Even under the relaxations discussed in Section 3.5.1.5, it could happen that the current converged estimates do not meet all the necessary assumptions to update the prediction model. Particularly, as shown by Proposition 3.5, the critical properties are the invariance of the terminal constraint set and the robust invariance of the tightening set. Although the design could be robustified to account for a subset of models $\tilde{\mathcal{M}}$ as discussed in Section 3.5.2, any $(\mathcal{A}, \mathcal{B}) \notin \tilde{\mathcal{M}}$ may still not meet the necessary conditions to become the new prediction model.

If the estimates converge to $(\mathcal{A}, \mathcal{B}) \notin \tilde{\mathcal{M}}$ and are expected to remain there for a considerable amount of time, updating the prediction model would require a re-design of the controller parameters. It is likely that only the invariant sets need to be re-computed, however this would result in a new controller with a completely different RoA and performance. In order to account for such an event define $\mathcal{E} = (\mathbb{W}_p, K, \mathbb{S}, \mathbb{X}, P)$ and consider the following result.

Proposition 3.7. Assume that, for a certain set of converged estimates $(\mathcal{A}, \mathcal{B})$, a set of parameters $\tilde{\mathcal{E}} = \left(\mathcal{W}_p, \tilde{K}, \tilde{\mathbb{S}}, \tilde{\mathbb{X}}_f, \mathcal{P}\right)$ that fulfils the assumptions of Theorem 3.3 exists. Furthermore, assume that the prediction model is to be updated by $(\mathcal{A}, \mathcal{B})$ at time $\bar{t}$. If $\mathbb{Q}_N\left(\tilde{\mathcal{E}}, x(\bar{t})\right)$ is feasible and $V_N^\circ\left(\tilde{\mathcal{E}}, \bar{x}(t)\right) - V_N\left(\mathcal{E}, \bar{x}(\bar{t}-1)\right) < -\ell\left(\bar{x}_0^\star\left(\bar{x}(\bar{t}-1)\right), \bar{u}_0^\star\left(\bar{x}(\bar{t}-1)\right)\right)$, then for all $t \geq \bar{t}$ (i) the optimization problem $\mathbb{P}_N\left(\tilde{\mathcal{E}}, x(t)\right)$ is feasible, (ii) state and input constraints are met despite the disturbances, (iii) the set $\tilde{\mathbb{S}}$ is exponentially stable with a region of attraction $\bar{\mathcal{X}}_N\left(\tilde{\mathcal{E}}\right) \oplus \tilde{\mathbb{S}}$ for the constrained closed-loop system

$$x(t+1) = \mathcal{A}(t) + \mathcal{B}\kappa_N\left(x(t)\right) + w(t),$$

for all $w(t) \in \mathcal{W}_p + \mathcal{B}\hat{\mathbb{W}}$ and (iv) the origin is exponentially stable with a region of attraction $\bar{\mathcal{X}}_N\left(\tilde{\mathcal{E}}\right)$ for the nominal closed-loop system

$$\bar{x}(t+1) = \mathcal{A}\bar{x}(t) + \mathcal{B}\bar{\kappa}_N\left(x(t)\right).$$

Proof. It follows from the proof of Theorem 3.3 and Proposition 3.5. ■

Note that the origin remains exponentially stable for the nominal closed-loop. Provided then that the cost decrease is met, the nominal trajectories remain exponentially converging towards the origin during the transition. The latter implies that the true trajectories also continue to converge exponentially fast towards the RPI set $\tilde{\mathbb{S}}$, albeit $\tilde{\mathbb{S}} \neq \mathbb{S}$ is likely. In fact, note that if $\bar{x}(\bar{t}-1)$

is already inside $\mathbb{S}$, the cost decrease is met only if $\bar{x}(\bar{t}) \in \tilde{\mathbb{S}}$, hence resulting in $V_N^\circ\left(\tilde{\mathcal{E}}, \bar{x}(t)\right) = 0$.

Moreover, following Proposition 3.5 it is straightforward to show that provided $x(t) \in \tilde{\mathbb{S}}$ for some t, the controller transition is possible for any $(\mathcal{A}, \mathcal{B}) \in \mathcal{M}$ for which a corresponding $\tilde{\mathcal{E}}$ exists. The only obstacle in finding an appropriate $\tilde{\mathcal{E}}$ for any $(\mathcal{A}, \mathcal{B}) \in \mathcal{M}$ is the admissibility of the robust invariant set $\tilde{\mathbb{S}}$, which size depends on $(\mathcal{A}, \mathcal{B})$ and $\mathcal{M}$, and thus could not be met given the constraint sets $\mathbb{X}$ and $\mathbb{U}$.

A drawback of resorting to the re-design of the controller is that the computational cost of computing invariant sets grows rapidly with the size of the state vector, thus making it difficult to compute them in most applications (even off-line). Nevertheless, it is important to note that although the plant may be accurately represented by $(\mathcal{A}, \mathcal{B})$ instead of $\left(\bar{A}, \bar{B}\right)$, the robustness of the tube MPC controller guarantees that constraints are met at all times, independent of when the controller transition takes place. Therefore, once the estimates have converged to an accurate representation of the plant, the computation of $\tilde{\mathcal{E}}$ can be performed during multiple sampling periods. Furthermore there exists efficient methods to compute invariant approximations to the type of set usually employed in MPC implementations, such as the minimal RPI set (see for example [100, 101]). Ultimately, this implies that the conditions for controller transition are not necessarily too demanding, and more importantly, that the transition can be performed without demanding the plant to be shut down.

Finally, as mentioned before, the three approaches proposed here are not mutually exclusive. Particularly, a robustified design increases the changes of the verification procedure to yield a positive outcome. If that is not the case, then a full controller re-design may be launched. Moreover, if the estimates have settled and are expected to remain there for a considerable amount of time, even if re-design is not necessary for prediction model update, it may be convenient due to performance requirements.

3.6 Illustrative example

The regulation and estimation capabilities of the propose adaptive MPC controller are now demonstrated through a numerical example. To this end, a point-mass spring-damper system is considered as the controlled plant (see Figure 3.1). The controlling input is an horizontal force applied directly onto the point mass, and the control objective is to steer the mass (which represents a truck) to an horizontal equilibrium arbitrarily defined.

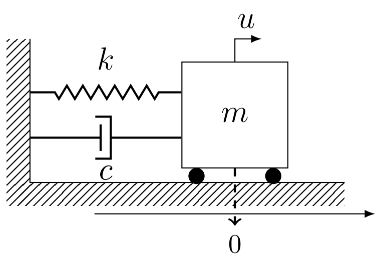

Figure 3.1: True plant.

The continuous time second order differential equation that represents the dynamics of such plant is

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -k/m & -c/m \end{bmatrix} x + \begin{bmatrix} 0 \\ 100/m \end{bmatrix} u. \tag{3.50}$$

The first element of the state vector, say x_1, represents the horizontal position of the truck (with respect to an arbitrary equilibrium point) while the second, x_2, represent the truck's velocity. A sampling time $T_s = 0.1[s]$ is used to discretise (3.50) and obtain a state space model of the form of (2.1a). Finally, the truck is supposed to be subject to state and input box constraints defined by

$$\mathbb{X} = \left\{x \in \mathbb{R}^2 \| ||x||_\infty \leq 15\right\}$$
$$\mathbb{U} = \{u \in \mathbb{R} \mid |u| \leq 5\}.$$

3.6.1 A-priori knowledge

At initialization the truck is loaded with a fixed and known mass, and the spring and damper constants are known factory values. Nevertheless, during operation the truck might receive up to a 25% additional cargo, and changes in the surrounding environmental temperature may result in the spring loosing up to 25% of its stiffness. This a-priori insight results in 4 limit scenarios, summarized in Table 3.1.

The Nominal–Nominal case ((I) in Table 3.1) is used to define the initial prediction model $\left(\bar{A}, \bar{B}\right)$, since that is the true model at initialization and there is no certainty on whether any of the other cases will effectively take place. In order to account for all possible changes within the cases presented in Table 3.1,

Table 3.1: Plant limit conditions.

	Cargo–Stiffness	m[kg]	k[N/m]	c[N/ms]	(A, B)
(I)	Nominal–Nominal	2	10	30	(A_I, B_I)
(II)	Nominal–Decreased	2	7.5	30	(A_{II}, B_{II})
(III)	Increased–Nominal	2.5	10	30	(A_{III}, B_{III})
(IV)	Increased–Decreased	2.5	7.5	30	(A_{IV}, B_{IV})

and given the linearity of the system, the set $\mathcal{M}$ is defined as

$$\mathcal{M} = \mathrm{co}\,\{A_j, B_j\}_{j=II,\ldots,IV}\,,$$

which results in the parametric uncertainty set being defined by

$$\mathbb{W}_p = \mathrm{co}\left(\bigcup_{j=II,III,IV} \mathbb{W}_p^j\right),$$

with

$$\mathbb{W}_p^j = (A_j - A_I)\,\mathbb{X} \oplus (B_j - B_I)\,\mathbb{U}.$$

This guarantees that $\mathbb{W}_p$ is a $\mathcal{PC}$-set.

The corresponding parametric uncertainty set is shown in Figure 3.2, alongside the state constraint set. Although the scenarios considered in Table 3.1 represent changes of up to one fourth in some of the plant parameters, the uncertainty set is contained in the interior of the state constraint set, fulfilling some of the necessary conditions for an admissible RPI set to exist.

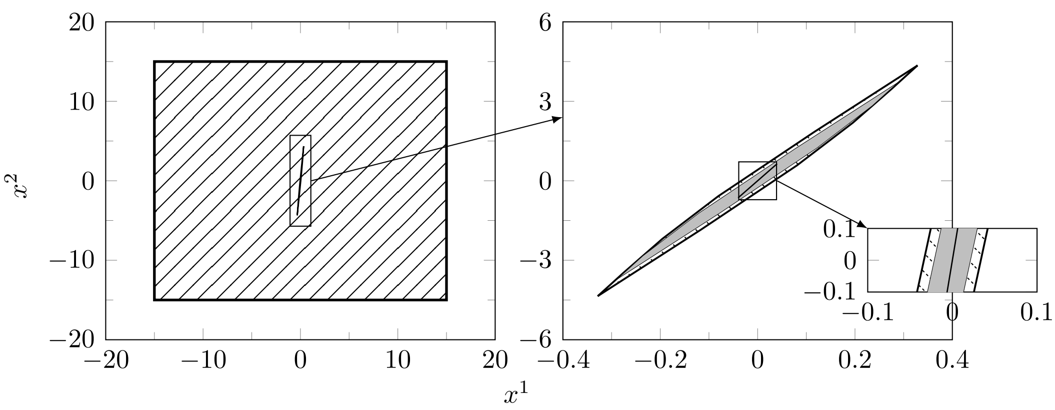

Figure 3.2: Disturbance sets: $\mathbb{X}$, $\mathbb{W}$, $\mathbb{W}_p$, —— $\bar{B}\hat{\mathbb{W}}$.

3.6.2 PE and RLS design parameters

First note that, provided that m is not zero, the (continuous time) system (3.50) is state reachable for any combination of (k, c, m), and so Assumption 3.1 is met independent the current condition of the plant. This allows for the analysis in Section 3.2 to be valid. The selection of the input partition parameter α is also closely related to the selection of the PE parameters l and ρ_0. Indeed, a larger α might be desirable to increase the size of the controller's RoA, however this will also result in a tighter constraint for the exciting part of the input. The latter means that a smaller ρ_0 or a larger l will be required to meet the PE constraint. Also recall that, according to Theorem 3.2, the exciting part of the input is required to be PE of order $2n + m + 1$ in order to guarantee that the regressor is PE of order 1. In view of this $h = 6$ and the values of l and ρ_0 are set to fulfil the discussion in Section 3.2.3.3.

Table 3.2 summarizes the values selected for the parameteres associated to the PE requirements. These fulfil all the conditions discussed in Section 3.2. The value of α is set small, so that 95% of the actuator's capacity is handed

Table 3.2: Parameters for the persistently exciting signal.

Parameter	Value
α	0.95
h	6
l	6
ρ_0	0.02

over to the controller, allowing for a large RoA. The corresponding disturbance set $\bar{B}\hat{\mathbb{W}}$ and the lumped disturbance set $\mathbb{W}$ are shown in Figure 3.2. The latter is also contained in the interior of the state constraint set, which is a necessary condition for an admissible RPI set to exist.

In order to initialize the exciting part it is necessary to compute a buffer sequence that fulfils Assumption 3.2. This is done taking l random elements within $\hat{\mathbb{W}}$ and then repeating them periodically to form the buffer sequence. Figure 3.3 shows the resulting sequence that fulfils all requirements of Assumption 3.2 and the constraint bounds. Finally, the forgetting factor for the RLS algorithm is set to $\lambda = 0.75$, in order to quickly forget the data associated to past realization of the plant after a change takes place.

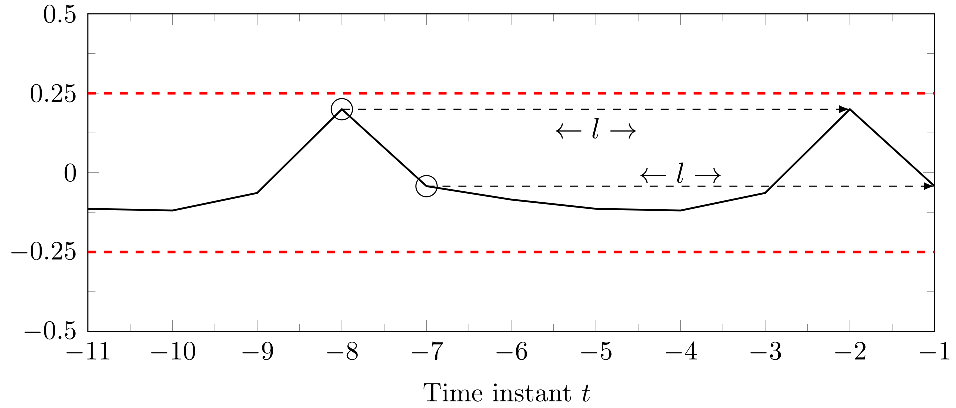

Figure 3.3: Buffer sequence: —— $\hat{w}_b$, - - - $\hat{\mathbb{W}}$.

3.6.3 Tube-MPC design parameters

For the tube MPC optimization the horizon is set to $N = 3$. A larger horizon would increase the size of the nominal RoA, however the main source of conservatism of the propose approach lies in the size of the RPI set associated to the uncertainty set $\mathbb{W}$, which does not depend on the horizon. The cost matrices are set to $R = 1$ and

$$Q = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix}$$

with $K = K_{LQR}\left(\bar{A}, \bar{B}, Q, R\right) = [-2.3441 \;\; -0.1709]$.

Owing to the discussion in Section 3.5, two different approaches are taken to design the remaining parameters. First, the different elements are computed following standard requirements, according the discussion in Chapter 2. After, said elements are computed in a robustified manner, according the discussion in Section 3.5.2, in order to increase the likelihood of a set of converged estimates being a valid candidate for prediction model update. Nevertheless, the particular set $\tilde{\mathcal{M}}$, that contains all the model for which certain criteria are guaranteed to be met, is not comprehensively computed. Tables 3.3 and 3.4 summarize how the remaining elements are designed in each approach. The additional subindices $_s$ and $_r$ will be used wherever explicit differentiation is required.

Figure 3.4 provides a direct comparison of the tightening sets and terminal constraint sets obtained for both approaches. As expected, any robust λ-contractive set is also RPI and hence $\mathbb{S}_s \subset \mathbb{S}_r$, while any λ-contractive set is also PI and hence $\bar{\mathbb{X}}_{f,r} \subset \bar{\mathbb{X}}_{f,s}$. This results, as shown in Figure 3.5, in smaller

RoAs (true and nominal) for the robustified design approach.

Table 3.3: Tube MPC parameters, standard design approach.

Parameter	Standard
$\mathbb{S}_s$	Minimal RPI set (admissible)
$\bar{\mathbb{X}}_{f,s}$	Maximal PI set (admissible)
P_s	Solution of $\bar{A}_K^\top P \bar{A}_K - P + Q + K^\top R K = 0$

Table 3.4: Tube MPC parameters, robustified design approach.

Parameter	Robustified	
$\mathbb{S}_r$	Minimal robust λ-contractive set (admissible)	$\lambda = 0.98$
$\bar{\mathbb{X}}_{f,r}$	Maximal λ-contractive set (admissible)	$\lambda = 0.99$
P_r	Solution of $\bar{A}_K^\top P \bar{A}_K - P + \gamma \left(Q + K^\top R K\right) = 0$	$\gamma = 1.15$

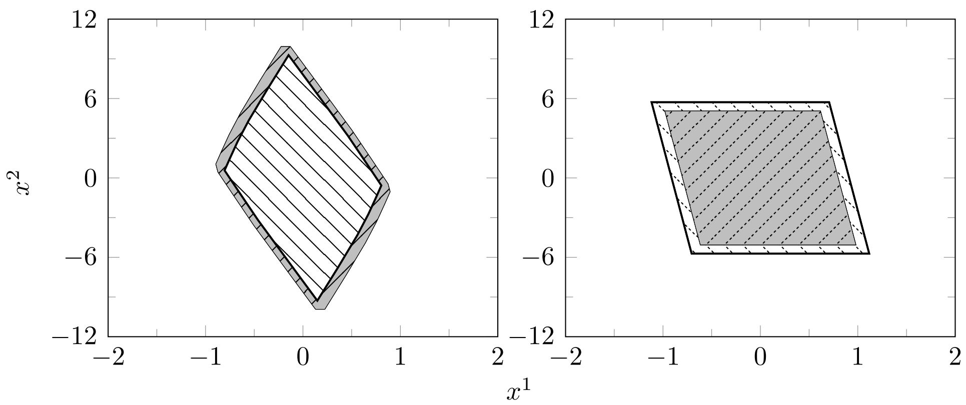

Figure 3.4: Invariant sets: (Left) $\mathbb{S}_s$, $\mathbb{S}_r$, (Right) $\bar{\mathbb{X}}_{f,s}$, $\bar{\mathbb{X}}_{f,r}$.

Figure 3.6 compares the tightening sets and RoAs, of both approaches, with the full state constraint set. The tightening set is, in both cases, large in the second dimension of the state, resulting in a nominal region of attraction $\bar{\mathcal{X}}_N$ that is small compared to the state constraint set $\mathbb{X}$. The true region of attraction $\mathcal{X}_N = \bar{\mathcal{X}}_N \oplus \mathbb{S}$ is considerably larger, however, still only represents up to 20% of the allowable state space. In particular, the proposed controller cannot guarantee constraint satisfaction for initial conditions with velocities larger than around $4[\mathrm{m}/\mathrm{s}]$. This is due to the state cost matrix Q and the dynamics of the system, which ultimately define the shape of the linear gain K and the shape of the tightening set $\mathbb{S}$.

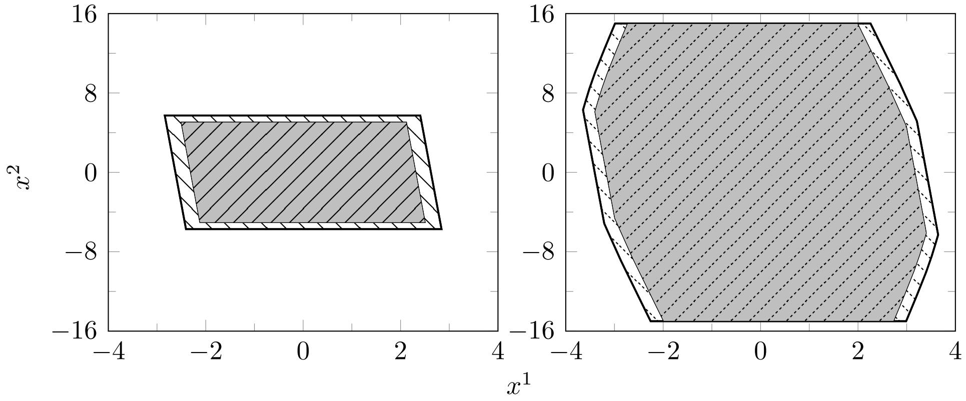

Figure 3.5: RoA sets: (Left) $\bar{\mathcal{X}}_{N,s}$, $\bar{\mathcal{X}}_{N,r}$, (Right) $\mathcal{X}_{N,s}$, $\mathcal{X}_{N,r}$.

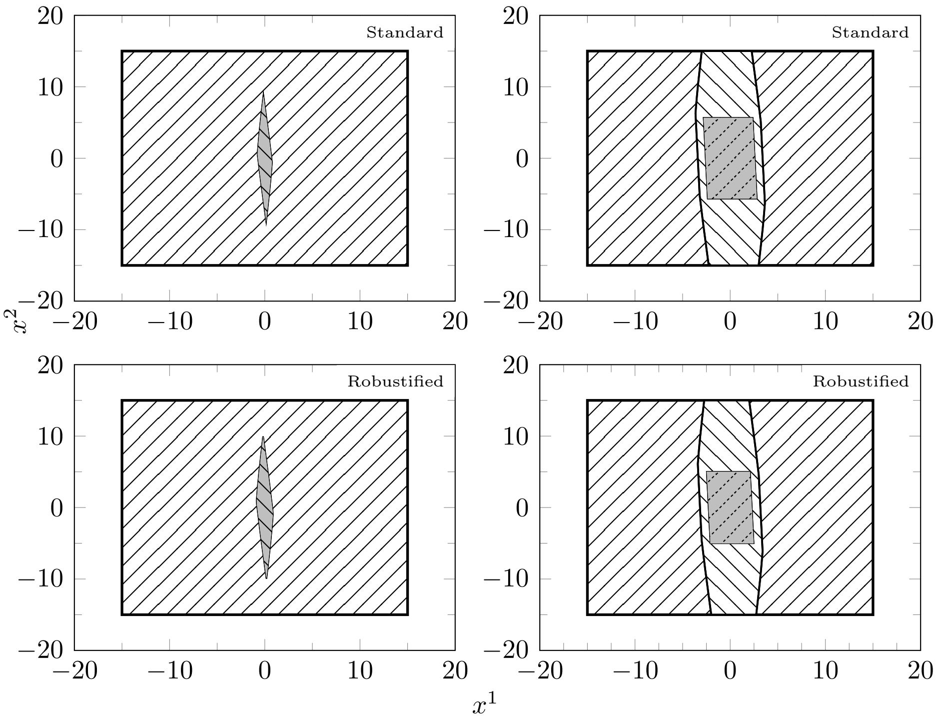

Figure 3.6: Tightening and RoA sets for standard (Top) and robustified (Bottom) designs: $\mathbb{X}$, $\mathbb{S}$, $\mathcal{X}_N$, $\bar{\mathcal{X}}_N$.

3.6.4 Dual controller

In order to assess the performance of the proposed dual controller the plant (3.50) is assumed to be slowly changing in accordance to the a-priori insight described by Table 3.1. Table 3.5 describes the changing profile that the plant is assumed to follow throughout a single realization of operation. Said pattern respects the a-priori knowledge of how the plant may change, and results in a plant that is realized by several interior points of $\mathcal{M}$, rather than just leaping between the vertices. Figure 3.7 shows how the bottom row values of the state

Table 3.5: Time variation of the plant parameters.

Time interval		Plant dynamics
(a)	$t \in [0\ \ 39)$	Nominal-Nominal
(b)	$t \in [40\ \ 59)$	Linear decrease of the spring stiffness up to -25%
(c)	$t \in [60\ \ 79)$	Nominal-Decreased
(d)	$t \in [80\ \ 119)$	Instantaneous increase of 25% in cargo (Increased-Decreased)
(e)	$t \in [120\ \ 139)$	Linear increase of the spring stiffness up to Nominal
(f)	$t \in [140\ \ 159)$	Increased-Nominal
(g)	$t \in [160\ \ 179)$	Linear removal of the additional cargo
(h)	$t \geq 180$	Nominal-Nominal

matrix A (in discrete-time) change throughout time. The results that follow are obtained by initializing the plant at $x(0) = [-1.6\ \ 15]^\top \in \mathcal{X}_{N,r} \subset \mathcal{X}_{N,s}$, and running the simulation for a total of $T = 220$ time instances.

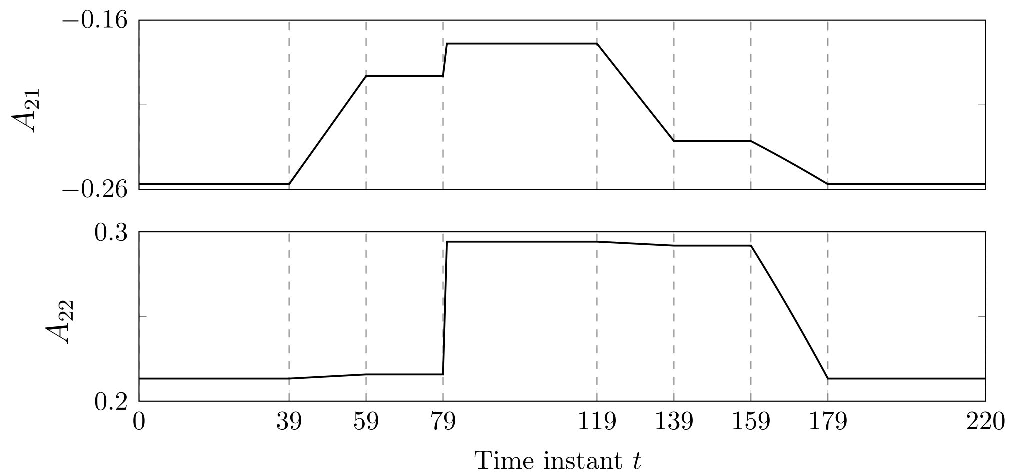

Figure 3.7: Time variation of the plant parameters.

3.6.4.1 Parameter estimation performance

The implementation of the dual MPC controller guarantees that the regressor is PE of order 1, and hence the RLS estimates must converge to the true plant parameters. Note however, that given the time-varying nature of the plant, the speed of convergence is tightly related to the forgetting factor used in the RLS algorithm. In this particular case said factor has been set to $\lambda = 0.75$, in order to quickly forget past information once a change has taken place.

Figures 3.8, 3.9 and 3.10 show the evolution of the true plant parameters and the estimates values through time. In both design approaches the RLS estimates converge to the true plant parameters once the latter have settled. The convergence is exact and occurs in finite time, due to the lack of measurement noise, however the number of time steps needed for convergence depends on the forgetting factor. The estimates of A_{11} and A_{21} present a high level of oscillation during the time intervals associated to a change in the cargo of the truck. This is due to the dynamics of the truck, and the outdated information carried by the RLS algorithm. However, this is not an issue for the MPC controller since the estimates are only considered as a candidate for updating the prediction model once they have converged.

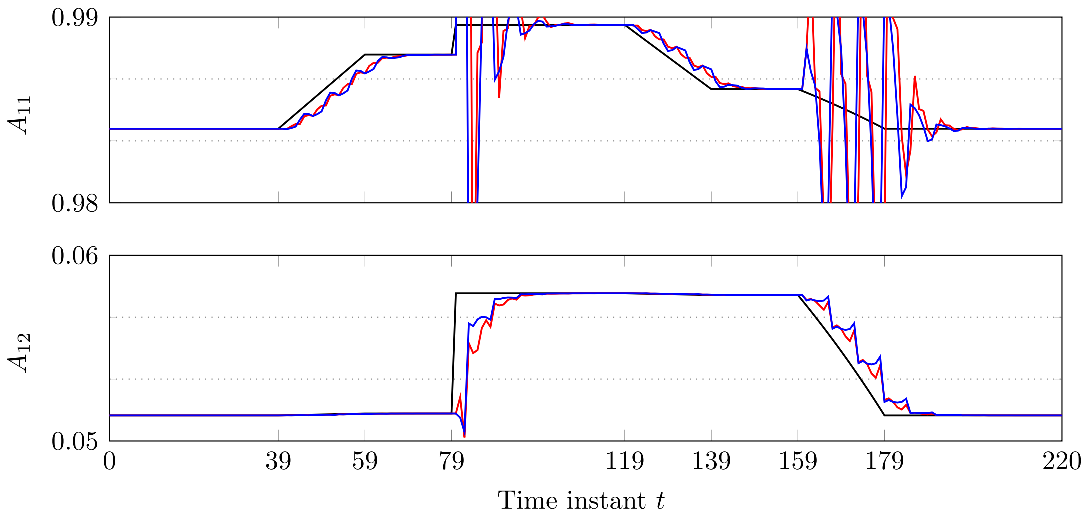

Figure 3.8: Parameter estimation results for A: —— true value, —— standard design estimate, —— robustified design estimate.

Finally, figure 3.11 shows the estimation error $E_{id}(t)$ and the value of the logical variable $f(t)$ that represents whether a particular set of estimates is a

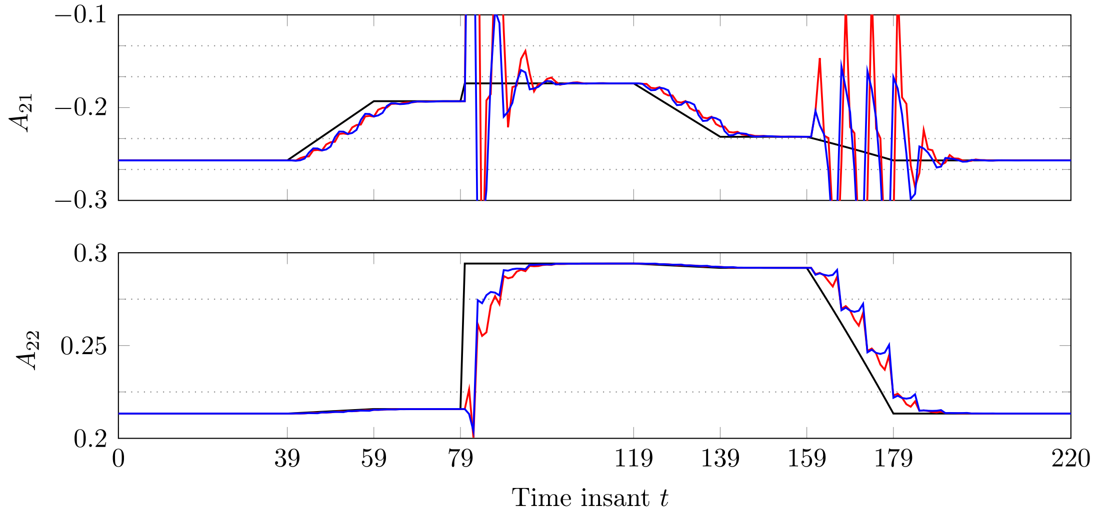

Figure 3.9: Parameter estimation results for A: —— true value, —— standard design estimate, —— robustified design estimate.

valid candidate for prediction model update or not. The error is computed as

$$E_{id}(t) = \max \left| \frac{\hat{\theta}(t) - \theta(t)}{\theta(t)} \right|,$$

where the division and the maximisation are performed element-wise. The flag is set to $f(t) = 1$ if the estimates have converged ($E_{id}(t) \approx 0$) and fulfil all conditions of Proposition 3.4, and $f(t) = 0$ otherwise. As expected, since the nominal prediction model used at initialization matches exactly the true plant parameters, the $E_{id}(t) = 0$ and $f(t) = 1$ for all $t < t_1$, however $f(t) = 0$ for all $t \geq t_1$ for the standard design. In other words, albeit robust and stabilizing, a naive approach to the design of the MPC controller may render impossible to use any other model for prediction, even if it better represents the current state of the plant.

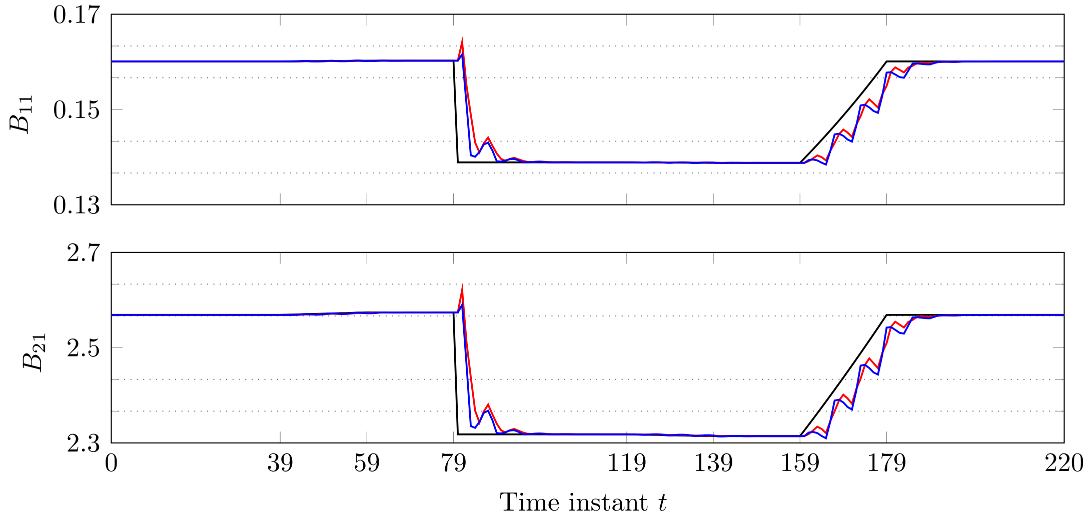

Figure 3.10: Parameter estimation results for B: —— true value, —— standard design estimate, —— robustified design estimate.

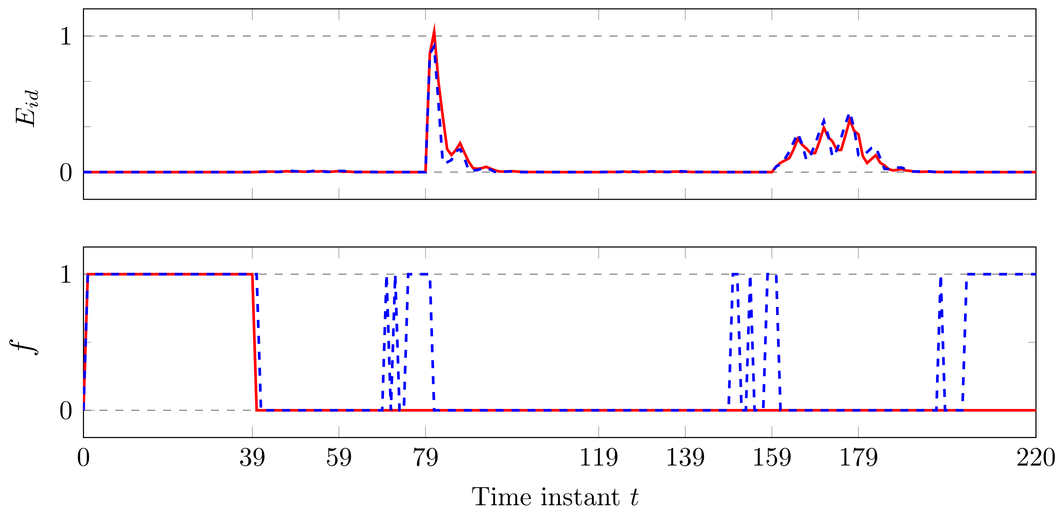

Figure 3.11: Validity analysis of the estimates: —— standard design, - - - robust design.

On the other hand, the robustified design allows for several periods of model update. Indeed, the prediction model is updated at three different times to match (A_{II}, B_{II}), (A_{III}, B_{III}) and to return to (A_I, B_I) during the last time interval. This leaves (A_{IV}, B_{IV}) as the only plant realization that cannot be used as a prediction model for the MPC optimization. For this particular case, it is the robust invariance of $\mathbb{S}$ that cannot be guaranteed for (A_{IV}, B_{IV}).

3.6.4.2 Control performance

The control performance is analysed next. Given that the standard design approach completely fails the task of prediction model update, only the results obtained through the robust design approach are presented. Figure 3.12 shows a phase plot of the nominal $\bar{x}(t)$ and true $x(t)$ state trajectories obtained through the implementation of the proposed dual controller. Recall that the true initial state is set to $x(0) = [-1.6 \;\; 15]^\top$, but at time $t = 0$ the nominal state is defined by the optimizer to guarantee $e(0) \in \mathbb{S}$ and that the nominal constraints are met. In this particular case this results in $\bar{x}(0) = [-1.3753 \;\; 5.0689]^\top$, which lies at the border of the $\mathcal{PC}$-set $\bar{\mathcal{X}}_N$.

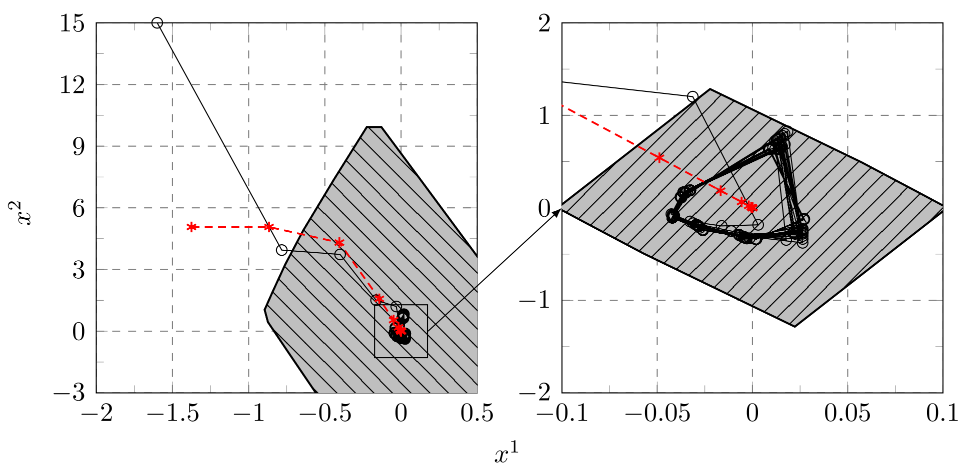

Figure 3.12: Phase plot of the state trajectory from $x(0) = [-1.6 \;\; 15]^\top$: —○— $x(t)$, - -∗- - $\bar{x}(t)$, ▧ $\mathbb{S}$, ▨ $\bar{\mathbb{S}}$.

The result in Theorem 3.3 guarantees exponential convergence of $x(t)$ to $\mathbb{S}$ and $\bar{x}(t)$ to the origin, however, as shown in Figure 3.12, the former converges to a much smaller neighbourhood of the origin. This is due to two reasons. First and foremost, the robust design approach allows for several instances of model update, which results in $w_p(t) = 0$ for several time steps. Ultimately this implies that the only disturbance affecting the system is that induced by the exciting part $\hat{w}(t)$, which is bounded in a set considerably smaller than the one used to compute $\mathbb{S}$ (see Figure 3.2). This has a secondary effect which is to keep the state trajectories close to the origin up until a new change takes place, which ultimately reduces the size of the corresponding uncertainty since not the whole state and input spaces are being used (similar to the ideas of reduced uncertainty sets discussed in [30]).

Secondly, note that the tube control law can be recast as

$$\hat{u}(t) = Kx(t) + \bar{u}_0^\star(t) - K\bar{x}(t)$$

where the last two terms converge exponentially fast to the origin. In the limit then, the true closed-loop simplifies to

$$x(t+1) = (A + BK)\, x(t) + B\hat{w}(t) \tag{3.51}$$

with $(A + BK)$ stable for all $(A, B) \in \mathcal{M}$ according to Proposition 3.1 and $x(t) \in \mathbb{S}$ and $B\hat{w}(t) \in B\hat{\mathbb{W}} \subset \mathbb{W}$. Hence the true state converges to a subset of $\mathbb{S}$. In Figure 3.12 said set is depicted by $\bar{\mathbb{S}}$, computed as the minimal RPI set for (A_I, B_I) (the true representation of the plant for all $t \geq t_7$) in presence of disturbances bounded in $B_I\hat{\mathbb{W}}$.

Figures 3.13 and 3.14 show the true, nominal and exciting part of the input that drive the state trajectories shown in Figure 3.12. As expected, the nominal input converges fairly fast, which results in $u(t) = Kx(t) + B\hat{w}(t)$ for all $t \geq 14$. The exciting part of the input guarantees that the true input remains persistently exciting of order $n + m$, and ultimate that the regressor is persistently exciting (of order 1). The exciting sequence, as shown in Figure 3.14, changes only slightly throughout operation, and is otherwise dominated by the buffer sequence, which is defined off-line and is not necessarily optimal for the current plant conditions and state. This is due to the non-convexity of the PE optimization used to define the exciting sequence, and the approach to deal with its feasibility, which forces l-periodicity on the optimal solution at each time instant.

Nevertheless, the optimizer is able to intervene and modify the exciting sequence at very specific times through operation. For example, after the first $h+l-1$ time steps have passed, the buffer sequence runs out and the optimizer takes charge of defining the exciting sequence in a way that is optimal for the current state and prediction model. Another example is found at $t \geq 150$, after the estimates have converged to (A_{III}, B_{III}). Since the latter is indeed employed to update the prediction model, the optimizer proceeds to update certain elements of the l-periodic exciting sequence to account for the new plant realisation.

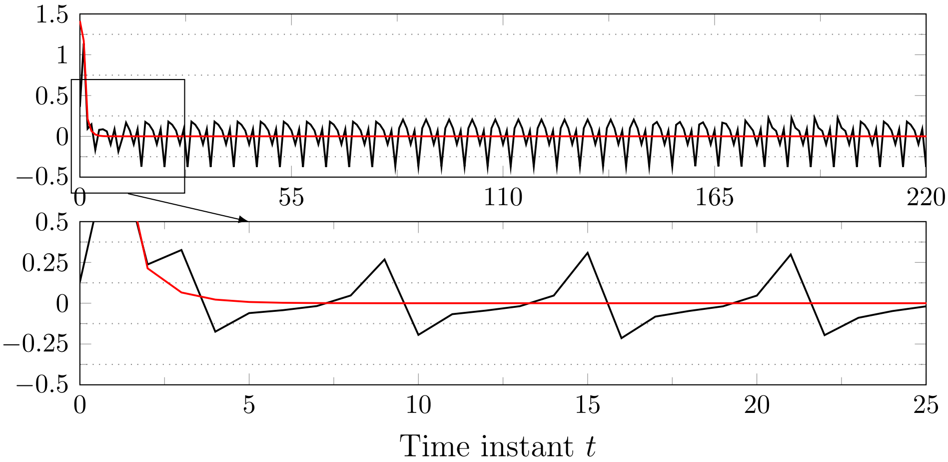

Figure 3.13: Input trajectory: —— $u(t) = \kappa_N(x(t))$, —— $\bar{u}(t) = \bar{\kappa}_N(x(t))$.

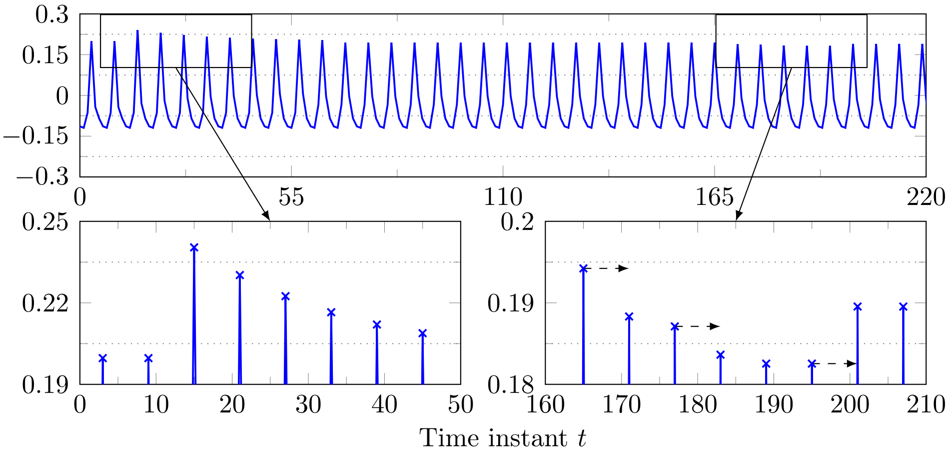

Figure 3.14: Excitation input trajectory: $w_h(t)$.

3.6.4.3 Robust versus adaptive approaches

The proposed adaptive MPC controller suffers from the fact that the excitation is always present, and hence perfect regulation is not possible. In view of Theorem 3.3 and Proposition 3.1, it could be argued that robust control would be enough to achieve stability and constraint satisfaction for such a time-varying plant. This is indeed, the case. Figure 3.15 shows the state trajectories obtained when using a simple tube-based MPC controller to regulate the plant (that is, no input partition). Indeed, given Proposition 3.1, and the lack of excitation, the state converges to the origin fairly fast.

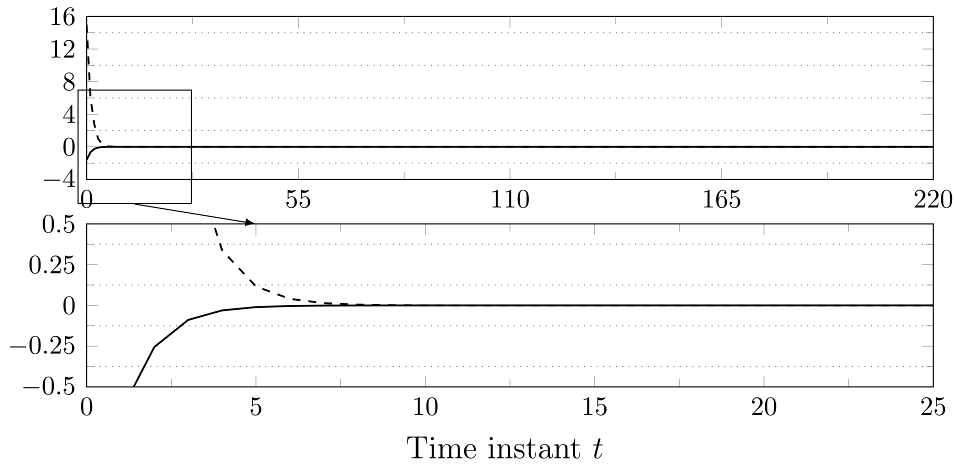

Figure 3.15: State trajectory under standard tube MPC from $x(0) = [-1.6\ \ 15]^\top$: —— $x_1(t)$, - - - $x_2(t)$.

Nevertheless, the objective of PETMPC dual controller is not just to regulate the plant, but also to obtain a more accurate representation of it during its operation. Figure 3.16 shows a selection of the estimates obtained by employing the same RLS algorithm, but fed with the closed-loop data generated by closing the loop with a standard tube-based MPC controller (that is, without guaranteed persistence of excitation). It is clear that, although perfect regulation is achieved, the estimates of the plant's parameters do not converge to the true values.

Finally, Table 3.6 presents the nominal and true aggregated cost of the three discussed approaches: PETMPC with standard design, PETMPC with robustified design and standard TMPC. The nominal aggregated cost is computed as

$$\bar{C} = \sum_{t=0}^{T} \bar{x}^\top(t) Q \bar{x}(t) + \bar{u}^\top(t) R \bar{u}(t)$$

and the true aggregated cost as

$$C = \sum_{t=0}^{T} x^\top(t) Q x(t) + u^\top(t) R u(t).$$

Note that, due to the exciting part of the input, the true states and inputs do not converge to the origin when the loop is closed with the PETMPC controller, as opposed to what happens when standard TMPC is implemented.

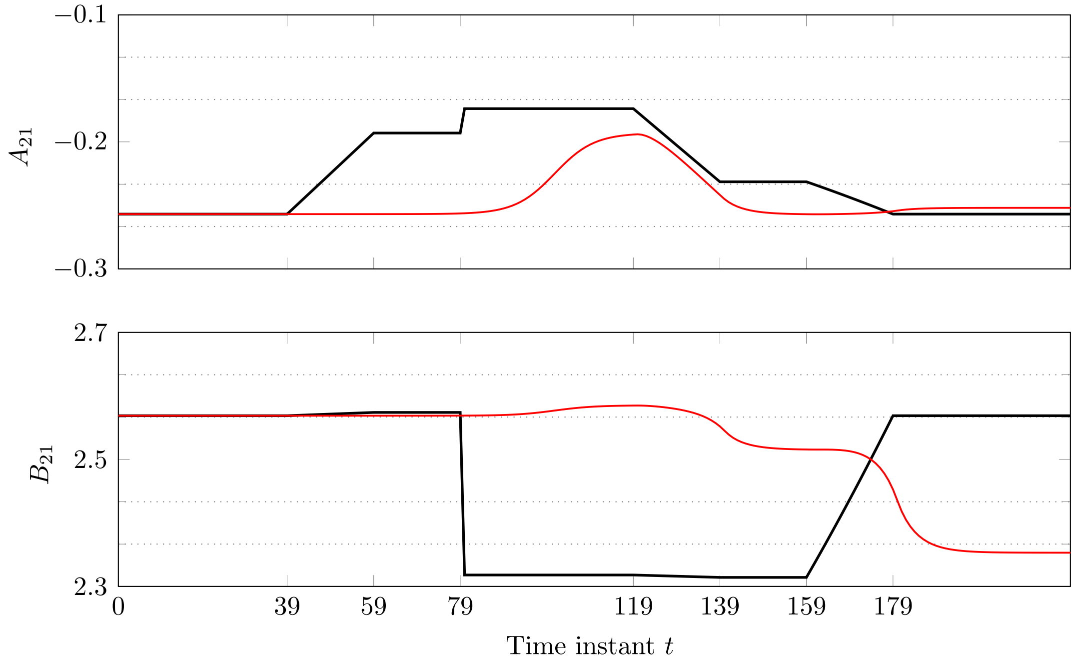

Figure 3.16: Parameter estimation results for standard tube MPC: —— true value, —— Tube MPC design estimate.

This results in a monotonically increasing C, which is why the true aggregated cost of the PETMPC variants is much larger when compared to the TMPC one. Nevertheless, the robustified approach to PETMPC outperforms the standard one in both measures, and it even outperforms a standard TMPC implementation in the nominal aggregated cost. This is notwithstanding the fact that the nominal control law $\bar{\kappa}_N(\cdot)$ has a larger control input authority in the standard TMPC given the smaller tightening set associated to it, stressing the importance of having an accurate prediction model.

Table 3.6: PE associated parameters.

Implementation	$\bar{C}$	C
PETMPC–standard	294.7099	650.5544
PETMPC–robustified	274.0070	644.4342
TMPC–standard	279.4956	577.1174

In view of the true aggregated cost results depicted by Table 3.6 it could be argued that standard TMPC implementations outperform the proposed adaptive controller. However, it is important to note that this apparent improved performance can only be observed in individual experiments, i.e., a single closed-loop trajectory from a single initial state as analysed here. In a more realistic operation setting, a controller has more to deal with than just a single initial value problem; changes to set-points and disturbances will repeatedly move the plant away from equilibrium. In any of these cases, having access to a more accurate representation of the plant will result in better predictions for the nominal MPC controller, and thus in an improved performance.

3.7 Summary

This chapter discussed the current approaches to adaptive model predictive control within the MPC framework, and proposed a new robust-based solution to the dual control problem. The proposed approach hinges on the partition of the control input into a regulatory part and an exciting part, which ultimately allows to tackle both problems independently, and guarantee robust constraint satisfaction, robust stabilizability, and convergence of the estimates of a standard RLS algorithm. In particular, the latter is achieved by forcing the exciting part of the input to be persistently exciting of appropriate order, which provided certain assumptions hold, guarantees the RLS regressor to be persistently exciting. The exciting sequence could be designed off-line to be PE of any required order, however in this chapter it is designed on-line, driven by the solution of a receding horizon MPC-like optimization problem. The objective of the latter is to introduce feedback in the computation of such a sequence, in order to guarantee that no unnecessary excitation is introduced into the system.

The main drawback of the proposed approach is that, although guaranteed to converge to the current true plant, the parameter estimates cannot be readily used to update the MPC's prediction model. This is because several of the MPC's elements are specifically computed for a single prediction model, and their properties may not hold for any other. This is overcome by proposing several simple steps of robustification for the design of such elements, which although not necessarily in full, may guarantee the adequacy of some future estimates for updating the prediction model.

The numerical example shows that all the guarantees claimed in Theorem 3.3 and Theorem 3.2 are indeed attained, however they also present several avenues for improvement. For example, Theorem 3.3 claims robust stability of the set $\mathbb{S}$ for the true dynamics, however the true state trajectory converges to a much smaller neighbourhood of the origin (see Figure 3.12). This is thanks to Proposition 3.1, and the updating of the prediction model. If at any time instant $x(t) \in \mathbb{S}$, then $\bar{x}(t) = \bar{u}(t) = \mathbf{0}$ can be guaranteed by solving the variant of tube MPC that optimizes trajectories, hence guaranteeing that the closed-loop reduces to (3.51) for all subsequent time steps. Stability of a smaller neighbourhood of the origin can then be guaranteed, at the expense of an additional verification step to test whether $x(t) \in \mathbb{S}$.

There is also room for improvement in the approach to guarantee recursive feasibility of the PE related optimization. Indeed, although the optimizer is able to modify the buffer sequence at key time instances (when the model is updated for example), the exciting sequence is mostly dominated by the buffer sequence (see Figure 3.3). As previously discussed, this is due to the non-convexity of the PE constraint, and the recursive feasibility guarantee that relies on the l-periodic application of previously feasible values. In particular, the non-convexity of the PE constraint could be explicitly tackled by convexification techniques [102], possibly leading to less conservative outcomes.

Finally, although the approach is deemed adaptive, it is only the prediction model which is sought to be actively updated, leaving all the other parameters of the MPC controller fixed (or changed when needed according to Section 3.5.3). There are, however, several elements that, if actively updated, may help improve performance throughout the plant's operation. An obvious example are the cost matrices, which may need to be updated to account for changes in the prediction model. Another example is the input partition parameter α. Indeed, the numerical example is set such that the plant continues to undergo changes throughout the simulation time but this might not be the case. If the plant parameters have settled, the control performance could be improved by returning full authority of the control input to the MPC optimization, allowing the excitation to return only when a new change is detected.

Chapter 4

Robust MPC for switching systems: minimum dwell-time for feasible and stabilizing switching

4.1 Introduction

Certainly, as established in Chapter 3, the prediction model of an MPC controller plays a critical role in the overall control performance. In Chapter 3 an adaptive MPC controller was proposed to address the case in which the system's dynamics are slowly changing in an uncertain manner. Robust stability and constraint satisfaction are guaranteed alongside with convergence of parameter estimates due to the probing effects of a part of the input which is specifically designed to excite the system. A key disadvantage of the PETMPC approach is that instantaneous updating of the prediction model might not be possible, thereby limiting the improvement in performance that is sought by implementing adaptive algorithms. This is circumvented by noting that the approach is robust to the entire parametric uncertainty, hence a new controller can be designed on-line and the switch can feasibly take place as soon as the true state enters the new RPI set.

The computation of the several elements related to the new controller, however, may take the time spanning a large number of time steps. Therefore, the assumption on slowly varying systems is key to allow for a prediction model update before the current estimates become obsolete. However, if the system experiments step-wise changes, every set of converged estimates may require a full controller redesign. This is disadvantageous, but not necessarily a drawback of the approach proposed in Chapter 3, but of the control and modelling paradigms employed for controlling a particular physical system.

Step-wise changes in the dynamics of a plant usually obey a modification of the inherent structure of the system. Sources of these changes are many and can range from system faults [103] to parts of the model becoming active/inactive depending on the current values of the state [104, 105]. Given their structural source, however, it is reasonable to expect these changes to be among a finite number of elements, opposed to the compact set $\mathcal{M}$ employed in Chapter 3 which is uncountable. Systems that undergo step-wise changes within a finite amount of different dynamical representations are often referred to as switching systems [106]. In the following discussion the focus is placed around switching control strategies based on MPC and related controllers, such as optimal control [107] and set theoretic methods [108].

Switching systems are comprehensively represented by a finite set of dynamical models and a collection of rules that defines which model is active at every time instant. These rules, albeit time-dependent, are not necessarily defined by explicit functions of time, and may even be completely arbitrary. A typical example of the former arises when several different models are used to represent the non-linear dynamics of a plant over the entire state space [106]. The area of the state space that each model covers depends on the non-linearities of the plant, and the switching takes place at the boundaries of these partitions [18, 104, 105, 109–112]. The switching signal depends on the current state, which in turn depends on the initial state, the control input, the (non-linear) dynamics of the plant and the time that has passed since initialization. The state trajectories, however, are completely defined by the sequence of control actions applied to the plant, hence a certain trajectory of switches can be forced by an appropriate controller design. In fact, in some cases the switching sequence is explicitly addressed as a design variable [110].

Arbitrary switching sequences, on the other hand, do not obey any particular rule such as those defined by partitioning the state space, thus a switch can take place at any given time [113]. Examples of arbitrary switching can be found in systems that are subject to externally controlled actuators, such as upstream valves allowing inflow of certain reactants in a tank system [109]. In the most broad definition of arbitrary switching sequences, future values of the sequence are entirely random and hence cannot be predicted; this means that an MPC controller lacks the information to directly account for a change in the plant's dynamics throughout its prediction horizon [114]. Set-based robust approaches, such as tube MPC, could be implemented to account for all possible changes [105], albeit resorting to high degrees of conservatism.

Conversely, and assuming that each element of the set of describing models fulfils the assumptions of Chapter 2, the design of locally stabilizing and constraint enforcing MPC controllers, with good individual performance, poses no major challenge. It is natural then to inquire whether a collection of stable and constraint admissible closed-loop systems remains stable and constraint admissible through arbitrary switching between them; the answer is no. There are certain cases in which a set of independently stabilized (or autonomous) LTI modes can be shown to be stable despite arbitrary (and arbitrarily fast) switching [107, 115–119]. However, most of these approaches rely on explicit characterization of the closed-loop system – which is not trivial in MPC – and the existence of a common Lyapunov function [120, 121], hence placing strong restrictions on the heterogeneity of the different modes. It follows then, that it is generally not trivial to guarantee stability of a switching system that switches arbitrarily fast between modes that have been stabilized by independent controllers [106, 113] (particularly non-linear controllers such as MPC). Moreover, the problem of constraint admissibility becomes particularly complex when the different dynamical models, also called modes from now on, are allowed to be subject to heterogeneous constraints.

Several authors have addressed the problem of guaranteeing stable and admissible closed-loop trajectories under arbitrary switching between independently closed closed-loops (within the MPC context and related). A tool that is repeatedly employed in this endeavour is the dwell-time, and most of the proposed approaches resort to either coupling the design of the individual MPC controllers, or centralizing the control on a single MPC [108, 114–117, 122–128]. Generally speaking, a dwell-time is the guaranteed number of time steps (or length of switching interval in continuous time implementations) that the switching system remains in a particular dynamical mode. This concept introduces certain conservatism in the notion of arbitrary switching since it effectively imposes a constraint on how fast a switch can happen.

Nevertheless, many physical systems subject to arbitrary switching on their dynamics do exhibit this type of behaviour, as opposed to impulsive changes back and forth. Furthermore, there exists many examples of systems that can only experience directed switching; consider for example an unmanned aerial vehicle that is remotely directed to drop its payload. The mass of the controlled system, and hence its dynamics, experience a step-wise change at an arbitrary time, however a change back is generally possible only when operation has finished. Another frequent concept is the mode-dependent dwell-time, or MDT. MDTs are employed to endow some flexibility to the controller by acknowledging that different modes might be constrained to remain active for different periods of time.

4.1.1 Switching with prescribed dwell-time

Depending on the application, it may happen that the dwell-time between the different dynamical modes is inherently fixed by the system's functioning, hence not open for design. Consider again a tank system with an externally controlled inflow valve. If the source of the inflow is subject to minimum volume constraints then the minimum amount of time the valve is open (prescribed dwell-time of such a mode) is set by an external demand.

In [122], for example, the MDTs are not only assumed prescribed, but also periodically returning; that is, the system is assumed to remain fixed within a certain mode during its corresponding MDT but in between these fixed events arbitrarily fast switching is allowed to take place during a maximum period of time T. This is a more general approach to the dwell-time problem, but still poses constraints on how arbitrary the switching can be. A collection of unconstrained linear modes subject to homogeneous disturbances is studied and the control strategy can be summarized as a cascade of robust controllers. The first layer is a tube-based approach designed to guarantee convergence of a neighbourhood of the origin in presence of the additive disturbance. Standard tube approaches use robust invariant sets as a cross-section for the tube, but in this case a mode-dependent persistent RPI set is introduced, to account for the prescribed periodical MDTs. The second layer of robust control is used to stabilize the undisturbed switching system with the robust LTV approach proposed in [26]. The two layers of robustness, however, may lead to small regions of attraction in the presence of (input) constraints. Furthermore, the iterations required for the computation of the generalized RPI set proposed in [122] may grow prohibitively complex.

A set theory based alternative that accounts for constraints is presented in [123, 129]. A set of command governors (CG) units are designed for the admissible stabilization of each mode of the system, which in this case are considered linear and subject to homogeneous disturbances and constraints. Each CG unit has a particular feasibility region, within the constraints, hence at the moment of switching it may happen that the current state is not feasible for the CG designed for the mode becoming active. If this is the case, the concept of 1-step ahead robust controllable sets is employed to guarantee finite time convergence to the intersection of the neighbouring feasibility regions. In simple terms, a set-based controller is employed to recover feasibility of the corresponding CG unit during the transition. The dwell-time in this approach is not seen as a minimum time a mode is active, but as a maximum time allowed for completing the transition. The drawback of this approach is twofold and pertaining exclusively to the transition controller. First, stability guarantees (or tracking capabilities) are completely lost during the transition, because the goal shifts from stabilization of a point to minimum time convergence of a set. Secondly, although feasibility is achieved thanks to the homogeneity of the constraints (which results in all the CG feasibility regions contained within a common set), there is no guarantee that the transition controller will be able to meet the required transition time.

To guarantee constraint satisfaction without violating the prescribed dwell-times [124, 130] employs the concept of dwell-time invariant/contractive sets first proposed in [131]. These sets are not invariant (or contractive) in the more standard definition [99], but better described as returnable sets under admissible switching sequences (similar to the generalized RPI sets in [122]). In simple terms, a state trajectory that starts within this set is guaranteed to be inside the set after any number of exact or surpassed dwell-times, but may be outside for times that fall short of an exact dwell-time. In [131] the existence of these sets is shown to be necessary and sufficient to guarantee admissible and asymptotically stable closed-loop trajectories of autonomous (or independently controlled) undisturbed switching linear systems subject to state constraints and a prescribed dwell-time.

In [124] these sets are employed as a terminal set for a centralized min-max MPC implementation that maximises over all possible admissible switching sequences, given the horizon length, and current and previously active modes. These sets, however, are returnable only for admissible switching sequences, which are defined following specific dwell times for each mode. On the other hand, the MPC horizon is fixed and recedes only by one time step each instance, possibly resulting in non-admissible sequences within the prediction time. This issue is solved by considering only a subset of the returnable set, which is shown to be returnable for curtailed sequences. The min-max optimization is discretised to comprehensively account for all possible switching sequences; this finite characterization means that the min-max optimization is tractable, however the design required to guarantee recursive feasibility of the optimization and exponential stability of the origin becomes considerably more complex than in standard (robust) MPC implementations. Recursive feasibility is guaranteed by including consistency constraints in the MPC optimization problem, to force the control actions to be feasible throughout the maximisation over switching sequences that have common initial modes. Exponential stability of the origin is guaranteed by extending the length of the horizon over which the mode-2 controller (the terminal controller) is active, to ensure the standard quadratic terminal cost decreases throughout admissible switching sequences. The latter interacts with the consistency constraints resulting in, possibly, small regions of attraction for the overall switching controller.

In order to avoid a min-max optimization [125] assumes that the switching sequence is known over the MPC prediction horizon. This is clearly more demanding than knowledge of the instantaneous value of the active mode, however it also allows for a considerably simpler design procedure. Undisturbed linear systems subject to homogeneous state and input constraints are analysed and the concept of backwards reachability is employed to guarantee constraint satisfaction despite the switching given a prescribe dwell-time d (same for all modes). A collection of d backwards reachability sets is designed for each mode such that the inner-most one is a subset of the outer-most one of the neighbouring modes. Given that this collection is backwards reachable, initialization in the outer-most guarantees the existence a sequence of control actions that drives the state into the inner-most one by the end of the corresponding dwell-time. This behaviour is enforced by actively changing the terminal constraint of the receding horizon optimization problem at each time instant, to align with the current left-over span of the dwell-time. Once the dwell-time has expired, and the inner-most set is guaranteed to be reachable within the horizon, safe transition to any neighbouring mode is ensured given the coupled design. Asymptotic stability of the closed-loop is shown across the switching intervals instead of the switching instances, although for a long enough dwell-time rather than a particularly prescribed one. The algorithm used to compute the backwards inter-reachable sets is initialized at the maximal PI set for a particular stabilizing linear feedback and mode, and it considers reachability under this same feedback rather than any control action. The latter is done neglecting the input constraints, hence there is no guarantee the algorithm will converge to a constraint abiding collection of inter-reachable sets.

A similar approach is presented by the same authors in [126]. In this case the switching sequence is not assumed to be known over the prediction horizon of the MPC controller, however stability guarantees are not devised. Again, undisturbed linear systems are analysed but they are assumed to be subject to heterogeneous constraints. A collection of d-step inter-reachable control invariant sets is computed. Each set is invariant under a single mode dynamics and any state inside it can reach the set of a corresponding neighbouring mode in d steps, under the dynamics and constraints of the neighbouring mode. Again, recursive feasibility is guaranteed by changing the terminal constraint of the receding horizon optimization problem at each time instant. Once a dwell-time has passed the upcoming state is guaranteed to be inside this control invariant set, and hence guaranteed to be able to reach a neighbouring one while respecting the neighbour's constraints.

4.1.2 Computation of minimum dwell-time

Oppositely, it could be the case that the different modes the plant transition through are heavily ruled by the expected performance and the objectives of the process. In this case, the dwell-time of each mode may obey internal demands instead of accommodating for external challenges, thus allowing for the definition of the different dwell-times. Recall the example of a tank system with an externally controlled outflow valve. If the destination of the outflow is not subject to minimum volume requirements, the amount of time the valve needs to remain open or closed (dwell-time of each mode) can be determined by the controllability requirements of the reaction in the tank, and subsequently informed to avoid violation. It is intuitive to expect that long dwell-times will allow for some form of stability to be achieved, however it is generally not as strong as for individual modes (e.g. Lyapunov stability). Furthermore, even slow switching could result in state constraint violation if constraints are not explicitly accounted for.

A minimum dwell-time required to guarantee stable switching is computed in [116] for a collection of linear autonomous discrete-time systems. Each system is assumed independently stable and hence local (quadratic) Lyapunov functions are guaranteed to exist. Stability is studied by analysing the piece-wise continuous function formed by the concatenation of the different individual Lyapunov functions through the switches. Asymptotic stability is guaranteed across the switching intervals instead of the switching instances, which means that although the piece-wise continuous Lyapunov function may grow at any given switching instance, the increase is smaller than the reduction experienced due to the most recent dwell-time spent in a single stable mode. It follows from the results in [116] that if a common quadratic Lyapunov function exists across modes, then a unitary dwell-time guarantees stability (this is a special case of the result in [121]).

Similar results are reported in [107, 115, 132], where the optimal control technique known as model reference adaptive control is implemented for output tracking of continuous-time linear [107] and non-linear [115, 132] systems. In [115, 132] the switching sequence is not arbitrary, but associated to the state space given the linearisation of the non-linear systems at different operation points. However, the stability of the linear switching reference model is analysed from an arbitrary switching perspective. In all these approaches the reference model is defined as a linear switching system with a Hurwitz assumption on the transition matrix of each reference mode. The latter implies that stability of the switching reference model can be guaranteed through stability of its homogeneous part, which reduces to finding quadratic Lyapunov functions for each mode and a sufficiently long dwell-time to guarantee exponential convergence over the switching intervals.

Quadratic Lyapunov functions, however, may be unnecessarily conservative given their fixed structure. In [117] an alternative is proposed in a continuous-time framework. First quadratic Lyapunov functions are employed to obtain an analogous result to that of [116], but in continuous time. The existence of such functions is shown to be merely a sufficient condition for stability and hence the more versatile polynomial Lyapunov functions are proposed as a replacement. The latter yields a necessary and sufficient result for the exponential stability of the continuous-time autonomous switching system, with associated minimum MDTs that are upper bounded by those associated to quadratic Lyapunov functions.

The stability-inducing MDTs proposed in [115–117] can be effectively computed by solving a tractable LMIs problem, however none of these approaches considers constraints. Input constraints, even when they are homogeneous throughout the modes, can result in instability due to saturated actuators hence revoking the stability guarantees. In [108] a set-theoretic method is proposed to compute stabilizing MDTs for undisturbed autonomous non-linear systems subject to state constraints and a Lipschitz continuity assumption. The central idea of this approach is to employ constraint admissible invariant sets to bound the state trajectories of each mode. The minimum dwell-time required to guarantee admissible switching is obtained by computing the amount of time each mode takes to move from any point inside its own constraint admissible invariant set, to the intersection of them all. Clearly, once inside the intersection the state is feasible (and recursively feasible) for all modes, hence a switch into any other mode can safely take place. Once feasible switching is ensured, asymptotic stability throughout the switching intervals (such as in [116]) follows if the invariant sets are computed as proper sub-level sets of corresponding Lyapunov functions.

In the MPC context, a key contribution to the computation of admissible and stabilizing dwell-times is found in [114, 133]. Linear modes subject to individual constraints are analysed both in the nominal and disturbed cases; the latter however only considers homogeneous disturbance levels. Opposed to other approaches such as [124, 125] that rely on a single MPC with changing prediction model and preview of the switching sequence, in [114] individual MPCs are designed for each mode. Given this structure, a switch between modes does not prompt a change in the prediction model, but a switch in the active controller, which ultimately allows to relax the assumptions on the switching sequence to instantly known but a-priori uncertain. For the undisturbed case, standard MPC controllers from the literature [1] are initially proposed. Recursive feasibility through a switch is characterized similarly to [108, 123], by means of the corresponding feasibility regions of each MPC controller. Several results are provided owing to the possibility of knowing a-priori the switching trajectory or at least the initially active mode, however if there is no a-priori knowledge on the sequence, their approach is comparable to the one proposed in [108]. The difference is that [108] considers autonomous systems, while in [114] MPC controlled systems are studied. The latter means that to compute forward reachability sets (to test for inclusion in an intersection of different feasibility sets) the explicit characterization of the MPC control law is required. This is certainly the biggest drawback of [114], since the MPC control law is non-linear and its explicit representation requires the implementation of multi-parametric programming tools [134].

Asymptotic stability of the origin is shown in [114] by enlarging the feasibility MDTs to guarantee a contraction in the forward reachability sets, when initialized in the intersection of the feasibility regions. The latter implies attractivity of the origin while stability is guaranteed by means of returnable sets. The approach in [131] is employed to compute a contractive returnable set that lies inside the intersection of the several terminal regions. Attractivity then implies that the states reach this returnable set in finite time, after which convergence is guaranteed provided the attractivity inducing MDTs are overridden (if necessary) by those associated to the returnable set. The algorithm in [131], however, is valid for explicitly characterized linear closed-loops. This is achieved in [114] by a proper design of the different MPCs terminal conditions, which result in time invariant linear feedback laws inside the terminal regions.

The robust case is tackled in fairly similar way. The generalized RPI sets described in [122] (constructed from the returnable RPI sets presented in [135]) are employed as the cross section of a standard tube MPC implementation [1,2]. Recursive feasibility and asymptotic stability through the switches is guaranteed by setting the MDTs to the largest between those required for the existence of a non-empty generalized RPI set and those previously defined for the convergence of the nominal system, albeit under tightened constraints. The version of tube MPC employed forces the initial state of the nominal dynamics to match the true initial state, reducing the overall region of attraction to that of the nominal system under tightened constraints. In summary then, although the MPC controllers (standard or tube-based) are initially designed independently, the use of the coupled returnable sets proposed in [131, 135] results in that the different controllers are indeed coupled.

Of all the different approaches previously discussed, clear differences can be observed in the proposed stability results, and this is owing to the different constraint considerations. In [115–117] (exponential) stability in the sense of Lyapunov is achieved through the computation of common Lyapunov functions, however this is possible because linear autonomous systems are analysed and constraints are not considered. MPC, on the other hand, usually results in a non-linear implicit closed-loop, furthermore constraints complicate the search for Lyapunov functions. Asymptotic stability, in the sense of Lyapunov, is achieved in [108] for constrained non-linear modes, albeit assumed autonomous and thus with explicit closed-loop representations. In [124, 125] asymptotic Lyapunov stability is guaranteed even in the presence of constraints, but assuming preview of the switching sequence over the prediction horizon. The individually designed, and posteriorly coupled, MPCs in [114] are shown to render the origin asymptotically stable only after a particular returnable set is reached.

A conceptually different approach is presented in [127], where the switching between homogeneously constrained non-linear modes controlled by independently designed MPC controllers is studied. An MDT that guarantees stabilizing switching between modes is computed by directly comparing the optimal cost function (usually employed as a Lyapunov function in MPC implementations) of each controller at the moment of switching. An important contribution is the computation of a multiplicative difference to compare the different Lyapunov functions within the MPC context, after which asymptotic stability follows under similar arguments to those in [116]. In order to guarantee recursive feasibility, an intersection of feasible sub-level sets of the optimal value function of each mode is computed. This set is not invariant but returnable within a single dwell-time under the dynamics of all modes, hence guarantees constraint satisfaction by construction (similar to the ideas in [114, 124, 135]). The same approach is extended to *MPC without terminal conditions* in [128], however constraints are not considered.

4.1.3 Dwell-time computation based on exponential stability

In this chapter the concept of MDTs is further explored and a new approach for their computation is proposed. The object of study is switching systems composed by LTI modes that are subject to heterogeneous constraints and disturbances and that are controlled by individual MPC controllers. The main tool employed to compute the corresponding MDTs is the exponential stability result thoroughly established in the literature for linear MPC [1, 2, 4].

Although each model follows the structure outlined in Section 2.2, the perturbed switching dynamics are formally introduced in Section 4.2, followed by a recast of the tube-based MPC optimization problem in order to properly introduce the mode-dependent notation. In Section 4.3, external disturbances are neglected and the nominal case is studied. Completely independent (off-the-shelf) MPC controllers are designed for each mode and minimum MDTs that guarantee feasible and stable switching between them are computed. This is opposed to other more complex formulations such as the min-max approach in [124] or the LTV approach in [122]. Feasible switching follows standard conditions of inclusion within the intersection of the feasibility regions of neighbouring modes, such as in [114, 127]. However, instead of the explicit characterization of reachable sets [114] or sub-level sets of implicit Lyapunov functions [127], the exponential decay of the independent MPC closed-loops is employed to characterize simple p-norm balls that bound the state trajectories at all times. The latter results in a design procedure that is considerably less computationally complex, which encourages the use of this approach, over other proposals, for systems of higher dimensions.

Stability inducing MDTs, on the other hand, are computed following a similar idea to that in [127], however an additive difference is found to provide a tighter bound (at least in the linear case). Ultimately, the origin is shown to be exponential stable over the switching intervals in the Lyapunov sense, achieving a result similar to that in [116, 117] but considering constraints, and also improving on the asymptotic stability result of [114, 122, 125]. Furthermore, the MPC controllers for each mode are designed independently from each other, as opposed to the solutions proposed in [114, 122, 125, 126] which require the computation of coupled returnable (invariant) sets.

In the perturbed case, and opposed to [114, 122, 135], heterogeneous disturbance levels across the different modes are considered. First, in Section 4.4, independent robust MPC controllers are designed for each individual mode. The tube MPC technique presented in Section 2.3.1 is employed since the optimization of the nominal trajectories allows for a less demanding inclusion test for guaranteeing feasible switching. The independently designed tube MPC controllers yield a collection of RPI sets that are robustly invariant only for a single mode and its corresponding disturbance levels, which ultimately results in that a stability guarantee similar to that of the undisturbed case is not achievable. Nevertheless, finite time convergence to a neighbourhood of the origin is guaranteed, after which a second set of feasibility related MDTs guarantees that such neighbourhood is attractive and stable. In order to improve on the previous results, Section 4.5 employs the concept of invariant multi-sets [136] instead of standard invariant sets for the definition of the corresponding tube cross sections. Invariant multi-sets are computed in a coupled manner and hence the overall design process is now coupled, however this allows for an exponential stability guarantee for a neighbourhood of the origin in a similar fashion than for non-switched tube-based MPC.

The biggest shortcoming of the proposed approach, as shown through numerical examples in Section 4.7, is that the resulting MDTs are excessively conservative owing to the loose analytical convergence rate available for MPC controllers. Nevertheless, the applicability of the proposed approach is demonstrated by recomputing the MDTs with tighter rates obtained numerically.

4.2 Switching linear systems

Consider a switching system composed by M LTI modes subject to, possibly, heterogeneous constraints and levels of disturbance. Each mode is modelled in a general state space representation following the structure presented in Section 2.2, thus the overall switching dynamics can be cast as follows

$$x(t+1) = A_{\sigma(t)}x(t) + B_{\sigma(t)}u(t) + w(t) \tag{4.1a}$$

$$x(t) \in \mathbb{X}_{\sigma(t)} \subset \mathbb{R}^{n_x} \tag{4.1b}$$

$$u(t) \in \mathbb{U}_{\sigma(t)} \subset \mathbb{R}^{n_u} \tag{4.1c}$$

$$w(t) \in \mathbb{W}_{\sigma(t)} \subset \mathbb{R}^{n_x}. \tag{4.1d}$$

At any particular time t, the mode that drives the system, alongside with the constraints that bound it and the disturbances that affect it, are entirely defined by the value of the switching signal $\sigma(\cdot)$. The latter is a assumed to be a piecewise constant function that, at each sampling time, takes values in the finite set $\mathcal{M} = \{1, \ldots, M\}$. Following the discussion in Chapter 2, it is assumed that, for all $m \in \mathcal{M}$, $\mathbb{X}_m$ and $\mathbb{U}_m$ are $\mathcal{PC}$-sets and $\mathbb{W}_m$ is a $\mathcal{C}$-set, but furthermore, that all three constraint sets are polyhedrons. The latter assumption will become relevant for the verification of the different inclusion conditions related to the computation of MDTs for feasible switching. Finally, it is also required that Assumption 2.1 holds for every pair (A_m, B_m) with $m \in \mathcal{M}$.

The switching instances are $\{t_0, t_1, \ldots, t_k, \ldots\}$ with $t_0 = 0$ and $t_k \geq t_{k-1}+1$, which results in $\sigma(t)$ being constant in the interval $[t_{k-1}\ ,\ t_k)$ for all $k \geq 1$. This structure implies that the dwell-times have to be at least of length 1, and that the switches take place exactly at the sampling instances. The former is necessary otherwise several modes could become active at the same time instance, while the latter is crucial since (4.1a) is usually a discretised version of a continuous-time process, hence switching that does not match the sampling instances would effectively result in prediction errors throughout the sampling intervals. Finally, it is assumed that the values of the switching signal are known instantly at each time t. This is a standard assumption in the analysis of switching systems [114,122,124–126] and it introduces less conservatism than assuming a-priori knowledge of the switching sequence (such as in [125]).

The concept of mode-dependent dwell-time, as it will be considered in this chapter, is now defined.

Definition 4.1. The mode-dependent dwell-time (MDT) associated to mode $m \in \mathcal{M}$, say τ_m, is the minimum amount of time during which the dynamics of the switching system (4.1) remain fixed at mode m before leaping into another allowable mode. It follows that if mode m became active at time t_k, that is

$\sigma(t_k) = m$ but $\sigma(t_k - 1) \neq m$, then $t_{k+1} - t_k \geq \tau_m$.

In what follows, minimum values for τ_m that guarantee feasible and stable switching between the different modes of (4.1) will be computed. These lower bounds effectively constrain how arbitrary the switching sequence can be, however there are other types of restrictions that can be placed on the switching sequence. A particular instance of the former is the case in which switching between certain modes is not allowed. In such cases $\sigma(\cdot)$ is referred to as a constrained switching signal (CSS), which can be precisely represented by a directed graph $\mathcal{G}\left(\mathcal{M}, \mathcal{E}\right)$, where $\mathcal{M}$ is the set of nodes, and $\mathcal{E} = \{(s, d)\,|s, d \in \mathcal{M}\}$ the set of edges that link the nodes together. Each edge represents an allowed switch and for each $(s, d) \in \mathcal{E}$, s represents the source node and d the destination node (s and d will also be referred to as neighbouring modes). Note that sources and destinations are not interchangeable, hence for any pair $m, l \in \mathcal{M}$, $(m, l) \in \mathcal{E}$ does not imply $(l, m) \in \mathcal{E}$. Notice also that for all $m \in \mathcal{M}$ it is assumed that $(m, m) \in \mathcal{E}$, otherwise the MDT for mode m would be fixed at $\tau_m = 1$. It follows that at each time instant t

$$\sigma(t) \in \mathcal{M}_{\sigma(t-1)} = \{d \in \mathcal{M}|\,(\sigma(t-1), d) \in \mathcal{E}\} \subseteq \mathcal{M}.$$

The focus is placed on the regulation problem, i.e. the design of a control law $u(t) = \kappa(x(t))$ that admissibly stabilizes the origin (or a neighbourhood of it) for the switching system (4.1) and a driving CSS. Initially, standard stabilizing and admissible (robust) MPC controllers are deployed independently for each mode. As discussed in Section 4.1, arbitrary switching among independently stabilizing controllers can result in an unstable switching closed-loop. Furthermore, the heterogeneity of the constraints may result in constraint violation at the moment of switching. To avoid this issues MDTs that allow for feasible and stabilizing switching between the independent MPCs are computed. Standard MPC controllers are designed for the disturbance free case, while tube MPC is employed for the disturbed case. The standard MPC implementation used to control undisturbed LTI systems with constraints can be seen as a special case of the tube-based MPC controller presented in Chapter 2. In view of this the tube-based approach is now briefly recast to account for the change in notation owing to the different modes.

4.2.1 Single mode tube-based MPC

At each time instant, the optimal control problem solved by the m-TMPC controller is

$$\mathbb{P}_{N_m}(x(t)): \qquad \min_{\bar{\boldsymbol{u}},\bar{x}_0} J_{N_m}(\bar{\boldsymbol{u}},\bar{x}_0) \tag{4.2a}$$

s.t. (for $k = 0, \ldots, N_m - 1$)

$$x(t) - \bar{x}_0 \in \mathbb{S}_m \tag{4.2b}$$

$$\bar{x}_{k+1} = A_m\bar{x}_k + B_m\bar{u}_k \tag{4.2c}$$

$$\bar{x}_k \in \bar{\mathbb{X}}_m \subseteq \mathbb{X}_m \ominus \mathbb{S}_m \tag{4.2d}$$

$$\bar{u}_k \in \bar{\mathbb{U}}_m \subseteq \mathbb{U}_m \ominus K_m\mathbb{S}_m \tag{4.2e}$$

$$\bar{x}_{N_m} \in \bar{\mathbb{X}}_{f,m} \subseteq \bar{\mathbb{X}}_m, \tag{4.2f}$$

where again $(\bar{x}_k, \bar{u}_k)$ are the nominal predictions, updated at each time instant to account for the newly measured true state, N_m is the prediction horizon employed by mode m, and $\bar{\boldsymbol{u}} = \{\bar{u}_0, \ldots, \bar{u}_{N_m-1}\}$ is the input sequence to be optimized. The sets $\mathbb{S}_m$ and $\bar{\mathbb{X}}_{f,m}$ are respectively an RPI and a PI set for the uncertain and nominal dynamics (4.2c) of mode m for a given stabilizing K_m according to Definition 2.2.

The cost function is, again, designed to approximate the infinite horizon LQR cost

$$J_{N_m}(\bar{\boldsymbol{u}},\bar{x}_0) = \sum_{k=0}^{N_m-1}\left(||\bar{x}_k||^2_{Q_m} + ||\bar{u}_k||^2_{R_m}\right) + ||\bar{x}_N||^2_{P_m},$$

with $Q_m, R_m > 0$ and $\bar{A}_m^\top P_m \bar{A}_m + Q_m + K_m^\top R_m K_m - P_m = 0$, where $\bar{A}_m = (A_m + B_m K_m)$. Note that the matrix inequality in Proposition 2.1–(c) is now replaced by an equality, hence the unconstrained infinite horizon LQR cost is not only approximated but exactly met. This is done to guarantee that the MPC control law is linear and time invariant when inside the terminal set $\bar{\mathbb{X}}_{f,m}$, a feature that is needed for subsequent developments. Define, as in (2.9),

$$\begin{aligned}\left(\bar{\boldsymbol{u}}^*(x(t)), \bar{x}_0^*(x(t))\right) &= \arg \mathbb{P}_{N_m}(x(t)) \\ V_{N_m}(x(t)) &= J_{N_m}\left(\bar{\boldsymbol{u}}^*(x(t)), \bar{x}_0^*(x(t))\right),\end{aligned}$$

set the nominal input to the associated receding horizon control law $\bar{u}(t) = \bar{\kappa}_m(x(t)) = \bar{u}_0^\star(x(t))$ and let the nominal trajectories be updated with $\bar{x}(t) = \bar{x}_0^\star(x(t))$. Furthermore, let $\bar{\mathcal{X}}_{N_m}$ be the set of all the states for which $\mathbb{P}_{N_m}(x)$ is feasible when constraint (4.2b) is replaced by $\bar{x}_0 = x(t)$, then Proposition 2.1 can be recast as follows.

Proposition 4.1. If (a) Assumption 2.1 holds with a certain K_m, (b) the sets $\mathbb{S}_m$ and $\bar{\mathbb{X}}_{f,m}$ are, correspondingly, admissible RPI and PI sets for $\bar{A}_m$ with respect to constraints (4.1b) and (4.1c), disturbance set $\mathbb{W}_m$ and tightened constraint (4.2d), (c) the sets $\mathbb{S}_m$ and $\bar{\mathbb{X}}_{f,m}$ are $\mathcal{PC}$-polyhedrons, (d) the loop is closed with $u(t) = \kappa_m(x(t)) = \bar{\kappa}_m(x(t)) + K_m\left(x(t) - \bar{x}_0^*(x(t))\right)$, then (1) the optimization problem (4.2) is recursively feasible with feasibility region $\mathcal{X}_{N_m} = \mathbb{S}_m \oplus \bar{\mathcal{X}}_{N_m}$, (2) the sets $\bar{\mathcal{X}}_{N_m}$ and $\bar{\mathcal{X}}_{N_m-1}$ are $\mathcal{PC}$-polyhedrons and invariant under $\bar{u}_0^*(x(t))$, (3) state and input constraints are met at all times despite the disturbances, and (4) there exist constant scalars $b_m, d_m, f_m > 0$ such that for all $x \in \mathcal{X}_{N_m}$ and $w \in \mathbb{W}_m$ it holds that:

$$b_m|\bar{x}_0^*(x)|_2^2 \leq V_{N_m}(x) \leq d_m|\bar{x}_0^*(x)|_2^2 \tag{4.3a}$$

$$V_{N_m}\left(A_m x + B_m\kappa_m(x) + w\right) - V_{N_m}(x) \leq -f_m|\bar{x}_0^*(x)|_2^2. \tag{4.3b}$$

Analogously, a corollary is provided to explicitly establish the exponential stability result arising from Proposition 4.1.

Corollary 4.1. The system of inequalities (4.3) implies that there exist constant scalars $c_m > 0$ and $\lambda_m \in (0,1)$ such that for all $x(0) \in \mathbb{S}_m \oplus \bar{\mathcal{X}}_{N_m}$, it holds that

$$|\bar{x}(t)|_2 \leq c_m \lambda_m^t |\bar{x}(0)|_2. \tag{4.4}$$

Therefore the origin is exponentially stable for the nominal trajectories of mode m when in closed-loop with $\bar{u}_0^*(x(t))$.

Proofs for Proposition 4.1 and Corollary 4.1 are again omitted and can be found in [1,2]. A proof for the additional claim about the polytopic shape of the feasibility regions can be found in [99,137]. In the subsequent developments specific values of the exponential stability constants are considered, in particular $c_m = \sqrt{d_m/b_m}$ and $\lambda_m = \sqrt{1 - f_m/d_m}$. Furthermore, although not stated, the linear gain employed to compute the terminal cost matrix P_m needs not to be equal to the tube gain employed in the composite control law $\kappa_m(\cdot)$. Both are identified as K_m to simplify notation, however they could be designed independently to pursue different objectives.

In what follows MDTs are computed such that the switching control law

$$\kappa(x(t)) = \kappa_{\sigma(t)}\left(x(t)\right) \tag{4.5}$$

(robustly) stabilizes the origin for the constrained switching system (4.1). This selection of control law implies that at any given time instant, the control action is entirely governed by the MPC designed for the currently active mode, instead of a governing MPC that spans all modes such as in [122, 124].

4.3 Disturbance-free switching linear systems

The disturbance-free case is analysed first, thus $\mathbb{W}_m = \{0\}$ for all $m \in \mathcal{M}$. The standard MPC optimization problem for undisturbed systems can be seen as a special case of (4.2) with constraint (4.2b) reduced to $\bar{x}_0 = x(t)$, given $\mathbb{S}_m = \{0\}$. In the following, and in order to ensure the results presented here remain notation-wise valid for the perturbed case, the true state of the undisturbed plant is referred to as $\bar{x}(t)$. The latter owns to the definitions in Section 4.2.1 that, provided $\mathbb{S}_m = \{0\}$, result in $\bar{x}(t) = \bar{x}_0^*(x(t)) = x(t)$. The problem of guaranteeing admissible switching between the independently designed MPCs is addressed first, since the overall stability of the switching closed-loop depends heavily on the feasibility of the optimization problems.

4.3.1 MDTs for admissible switching

To determine if a certain state $\bar{x}(t)$ is a feasible initial state for an MPC controller it is necessary and sufficient to test whether said state belongs to the MPC's feasibility region. The latter, however, might not be available during the design process owing to computational tractability reasons that are further discussed subsequently. In what follows the problem of admissible switching is analysed in both cases, and dwell-times that avoid constraint violation are computed even if the feasibility regions are unknown.

4.3.1.1 Known feasibility regions

Consider any pair of neighbouring modes $m, l \in \mathcal{M}$ and assume the feasibility regions of each are known. A switch from mode m to mode l is feasible at time t_k if and only if $\bar{x}(t_k) \in \bar{\mathcal{X}}_{N_l}$. However, the different modes of the switching system (4.1) are allowed to be highly heterogeneous, thus it is likely that $\bar{\mathcal{X}}_{N_m} \neq \bar{\mathcal{X}}_{N_l}$. It is also non-trivial to guarantee $\bar{\mathcal{X}}_{N_m} \subseteq \bar{\mathcal{X}}_{N_l}$ simultaneously for all pairs of neighbouring modes without incurring in stringent design conditions that ultimately couple the design process. Therefore it can be safely assumed that $\bar{x}(t_k - 1) \in \bar{\mathcal{X}}_{N_m}$ does not guarantee $\bar{x}(t_k) \in \bar{\mathcal{X}}_{N_l}$, hence a switch at time

t_k might render the control law (4.5) infeasible.

However, provided there is no switching, the individual MPC controllers do guarantee closed-loops that converge exponentially fast to the origin. In fact, it follows from (4.3b) that for any $\bar{x}(0) \in \bar{\mathcal{X}}_{N_m}$, the closed-loop trajectories are continuously contained inside sub-level sets of the value function that follow a backwards inclusion condition [108, 127] (that is, the sub-level set that contains $\bar{x}(t)$ is a strict superset of the sub-level set that contains $\bar{x}(t+1)$). It follows then that, if mode m is active for enough time steps, the sub-level set that contains the state will be a subset of any neighbouring feasibility region, thus allowing for an admissible switch.

In [114] an even more precise account of where the closed-loop trajectories are is employed by exactly characterising the one-step ahead reachability sets. This provides the exact number of time steps required for the inclusion to be met, however it also demands for the explicit characterization of the non-linear MPC control law, which in turn requires the implementation of multi-parametric programming tools [134]. The computational complexity of the latter can be prohibitively large, especially for systems of large dimension. In order to keep the computational complexity low, the exponential stability result of the m-MPC controller is hereafter leveraged to compute a time-varying set, characterised by a decreasing Chebyshev radius, that contains the m closed-loop at all time instances and for any initial state within $\bar{\mathcal{X}}_{N_m}$. The proposed set is a $\mathcal{PC}$-set, it can be computed off-line, and its computation does not require the explicit knowledge of the MPC control law. The MDTs for admissible switching then follow in a straightforward manner: whenever this set is inside the feasibility region of a neighbouring mode, a switch is admissible.

The exponential convergence result in Corollary 4.1 bounds the 2-norm of the nominal state trajectories. Hence, it is trivial to define a time-dependent 2-norm ball with exponentially decreasing Chebyshev radius that is guaranteed to contain the m closed-loop at any given time. However, within the collection of p-norms the 1 and ∞ norms produce balls that are not only $\mathcal{PC}$-sets, but also polyhedrons, hence completely characterised by a finite number of defining half spaces. This results in the set operations needed in the subsequent steps of this approach (for example the intersection of two convex sets) being greatly simplified. Furthermore, since for equal radii the 1-norm ball is contained inside the 2-norm ball and the ∞-norm ball contains both, the 1-norm ball is considered as a less conservative choice. The exponential stability result in

(4.4) is now recast to account for the 1-norm.

$$|\bar{x}(t)|_1 \leq \sqrt{n_x} c_m \lambda_m^t |\bar{x}(0)|_2 \tag{4.6}$$

A drawback of employing a 1-norm bounding ball is that its associated radius is larger than that of the corresponding 2-norm ball for any system with more than two states (since $\sqrt{n_x} > 1$ for all $n_x > 1$). This adds conservativeness as it implies that the admissibility inclusion may require a larger number of time steps to be verified. However, a 1-norm ball of radius r does not necessarily contain a 2-norm ball of radius $r/\sqrt{n_x}$. Furthermore, the difference between both radii decreases exponentially fast following the exponential convergence.

A shortcoming of the bound offered by (4.6) is that it depends on a specific value of initial state, hence a 1-norm ball with decreasing radius defined by the right hand side of (4.6) can only guarantee admissible switching if the system is initialized at that particular state. In order to generalize for any feasible state compute

$$\alpha_m = \max_{x \in \bar{\mathcal{X}}_{N_m - 1}} |x|_2. \tag{4.7}$$

and define the time-varying set

$$\mathbb{B}_{r_m(\tau)} = \bar{\mathcal{X}}_{N_m-1} \cap \mathcal{B}_{r_m(\tau)}. \tag{4.8}$$

with $r_m(\tau) = \sqrt{n_x} c_m \lambda_m^\tau \alpha_m$. Then, the following result holds for any $k \in \mathbb{N}_0$.

Proposition 4.2. If mode m became active at the last switching instant t_{k-1} (feasibly), and the loop is closed with the m-MPC control law $\bar{\kappa}_m(\cdot)$, then the nominal state trajectory of the switching system fulfils $\bar{x}(t) \in \mathbb{B}_{r_m(t-t_{k-1}-1)}$ for all $t \geq t_{k-1} + 1$.

Proof. If $\bar{x}(t_{k-1}) \in \bar{\mathcal{X}}_{N_m}$ and the loop is closed with $\bar{\kappa}_m(\cdot)$, then for all $t \geq t_{k-1}+1$ it holds that $\bar{x}(t) \in \bar{\mathcal{X}}_{N_m-1}$ and

$$\begin{aligned}
&|\bar{x}(t)|_1 \leq \sqrt{n_x} c_m \lambda_m^{t-t_{k-1}} |\bar{x}(t_{k-1})|_2 \\
\implies &|\bar{x}(t)|_1 \leq \sqrt{n_x} c_m \lambda_m^{t-t_{k-1}-1} |\bar{x}(t_{k-1}+1)|_2 \\
\implies &|\bar{x}(t)|_1 \leq \sqrt{n_x} c_m \lambda_m^{t-t_{k-1}-1} \alpha_m \\
\implies &\bar{x}(t) \in \mathcal{B}_{r_m(t-t_{k-1}-1)},
\end{aligned}$$

where the second inequality follows from (4.4) and (4.6), the third inequality follows from (4.7) and the last one from the definition of a 1-norm ball. Hence, for all $t \geq t_{k-1}+1$, it holds that $\bar{x}(t) \in \bar{\mathcal{X}}_{N_m-1}$ and $\bar{x}(t) \in \mathcal{B}_{r_m(t-t_k-1)}$, which given (4.8) completes the proof. ■

A couple remarks are in order. First note that the maximum in (4.7) is taken over $\mathcal{X}_{N_m-1}$ rather than $\mathcal{X}_{N_m}$; this is done to reduce conservativeness. Indeed, it follows by the invariance of the feasibility regions that if $\bar{x}(\bar{t}) \in \bar{\mathcal{X}}_{N_m}$ then $\bar{x}(t) \in \bar{\mathcal{X}}_{N_m-1}$ for all $t \geq \bar{t}+1$. Since $\mathcal{X}_{N_m-1} \subseteq \mathcal{X}_{N_m}$ and the dwell-times are at least 1 time instance long, taking the maximum over $\mathcal{X}_{N_m-1}$ reduces the overall admissibility MDTs. Similarly, the 1-norm ball provided by the bound in (4.6) is intersected with $\mathcal{X}_{N_m-1}$ in (4.8) since by the latter's invariance, $\bar{x}(\bar{t})$ must be inside it for all $t \geq \bar{t}+1$. Proposition 4.2 still holds if this intersection is omitted, however note that given a $c_m > 1$ it could happen that $\mathcal{B}_{r_m(0)} \supseteq \mathcal{X}_{N_l}$, but $\mathcal{X}_{N_m-1} \subseteq \mathcal{X}_{N_l}$, which may result in unnecessarily long waits for the admissibility inclusion to be verified.

Certainly, $r_m(t-t_{k-1}-1)$ is a conservative radius for the ball that contains the state at time t because it is computed with α_m instead of using the norm of the current state $|\bar{x}(t_{k-1}+1)|_2$. However, as previously discussed, this allows a Proposition 4.2 that is entirely independent of the initial state, but more importantly, of the specific switching times. Such dependency is allowed in [114] yielding sequences of feasibility MDTs that depend on the stage of switching (i.e., on the number of switches that have taken place previous to the current one). The same results are attainable with the approach presented here, but are not pursued given that employing such MDTs would require knowledge of the switching trajectory.

Proposition 4.2 ensures that the time varying set $\mathbb{B}_{r_m(\tau)}$, with decreasing Chebyshev radius, contains the state trajectories of the m closed-loop at all time instances. The MDT that allows for feasible switching among neighbouring closed-loops follows then in a straightforward manner.

Theorem 4.1. Consider any pair $m, l \in \mathcal{M}$ with $m \neq l$, $\sigma(t_{k-1}) = m$ and $l \in \mathcal{M}_m$. Furthermore, assume that mode m became active (feasibly) at the previous switching instance t_{k-1}. If $\tau^f_{m,l} \geq 0$ is such that $\mathbb{B}_{r_m(\tau^f_{m,l})} \subseteq \bar{\mathcal{X}}_{N_l}$ but $\mathbb{B}_{r_m(\tau^f_{m,l}-1)} \nsubseteq \bar{\mathcal{X}}_{N_l}$, then a switch to mode l is feasible at any time t_k that fulfils $t_k - t_{k-1} - 1 \geq \tau^f_{m,l}$.

Proof. According to Proposition 4.2, if mode m became active feasibly at time t_{k-1} then $\bar{x}(t_k) \in \mathbb{B}_{r_m(t_k-t_{k-1}-1)}$ for all $t_k \geq t_{k-1}+1$. It follows from (4.8) that if $t_k - t_{k-1} - 1 \geq \tau^f_{m,l}$ then $\mathbb{B}_{r_m(t_k-t_{k-1}-1)} \subseteq \mathbb{B}_{r_m(\tau^f_{m,l})}$, hence $\bar{x}(t_k) \in \mathbb{B}_{r_m(\tau^f_{m,l})}$. Then, by assumption, $\bar{x}(t_k) \subseteq \bar{\mathcal{X}}_{N_l}$ which guarantees feasibility of $\mathbb{P}_{N_l}(\bar{x}(t_k))$.

Finally note that $\tau^f_{m,l}$ is such that $\mathbb{B}_{r_m(\tau^f_{m,l}-1)} \nsubseteq \bar{\mathcal{X}}_{N_l}$, hence feasibility cannot be guaranteed for $t_k < \tau^f_{m,l} + t_{k-1} + 1$. ■

Theorem 4.1 establishes the minimum amount of time that mode m needs to be active before a switch into mode l is feasible. The following corollary to Theorem 4.1 properly defines the MDTs given a set of modes $\mathcal{M}$ and a CSS.

Corollary 4.2. Assume $\sigma(\cdot)$ is a CSS. If for all $m \in \mathcal{M}$ the MDTs are set to τ^f_m defined by

$$\tau^f_m = 1 + \max_{l \in \mathcal{M}_m} \tau^f_{m,l},$$

then the switching control law (4.5) guarantees constraint satisfaction for the switching linear system (4.1).

Proposition 4.2, Theorem 4.1 and Corollary 4.2 provide a simple, yet comprehensive approach to the computation of MDTs that guarantee admissible switching among independently designed MPC controllers for heterogeneous modes. Nevertheless, 1-norm balls are a conservative bound for the state trajectory when compared to sub-level sets of the value function [127] or exact MPC-reachable sets [114], therefore longer (more conservative) feasibility MDTs are expected. This, however, represents a trade-off between accuracy and complexity, since the computation of sub-level sets of the MPC value function usually requires a numerically exhaustive approach [1], while the characterization of exact MPC-reachable sets demands the knowledge of the implicit and non-linear MPC control law. Theorem 4.1, on the other hand, only requires:

- Computation of the exponential stability constants in (4.7)
- Computation of the feasibility regions $\bar{\mathcal{X}}_{N_m}$ and $\bar{\mathcal{X}}_{N_m-1}$
- Intersection operations and inclusion tests among $\mathcal{C}$-polyhedrons

Computing the exponential stability constants is simple [1], however the bounds obtained for the value function are not necessarily tight. The impact that this has on the MDTs is discussed in more depth in Section 4.7. More important is to note that the exponential stability result, and thus the decay in (4.4), can only be guaranteed when the MPC optimization problem is solved to optimality. Suboptimal MPC can also shown to be stable, albeit only asymptotically. Nevertheless, the proposed cost function, polytopic constraints and polytopic invariant sets result in the corresponding optimization being a convex QP problem, for which efficient algorithms exist.

4.3.1.2 Unknown feasibility regions

Computing the feasibility sets, on the other hand, can be challenging. The feasibility region of mode m can also be referred to as the N_m-step stabilizable set to $\bar{\mathbb{X}}_{f,m}$, as it contains all the states that can be driven into the terminal region in N_m steps or less. This set can be computed by the recursive application of the backwards reachability operation [97, 137]. In the case of LTI systems, if the constraint sets and the terminal set $\mathbb{X}_{f,m}$ are convex polyhedrons with known H-representation, computing the i_m-step stabilizable set requires the iteration of several simple operations. These include the Pontryagin difference and intersections between polyhedrons, with fairly low complexity even for large N_m (see for example Algorithm 2.1 and the corresponding discussion in Section 3.3 of [97], and also the seminal results in [96, 98]).

It is still the case, however, that high-dimensional plants with complex constraint sets (i.e. with a large number of defining half-spaces) may result in that the corresponding feasibility sets are prohibitively complex to compute. Given the invariance of the terminal set, it can be shown [137] that the i_m-step stabilizable sets to $\bar{\mathbb{X}}_{f,m}$ are consecutively inclusive, hence any $\bar{\mathcal{X}}_{i_m}$ with $i_m \in [1, N_m)$ represents a set of feasible states for $\mathbb{P}_{N_m}(\cdot)$, albeit not necessarily invariant under the m-MPC control law. Furthermore, if $\bar{\mathcal{X}}_{N_m}$ is not tractable, then the computation of $\bar{\mathcal{X}}_{i_m}$ may also not be.

Nevertheless, MDTs that guarantee admissible switching can still be computed even if the corresponding feasibility regions are unknown. In order to do so first note that $\mathbb{P}_{N_m}(\bar{x}(t))$ is feasible for any $\bar{x}(t) \in \Theta_m$ if $\Theta_m \subseteq \bar{\mathcal{X}}_{N_m}$. The set Θ_m needs no other particular consideration, although if Θ_m is not invariant under the m-MPC control law, then for any $\bar{x}(t) \in \Theta_m$ the subsequent state is only guaranteed to be inside the unknown $\bar{\mathcal{X}}_{N_m}$. In order to overcome this issue, and ensure admissible switching, define

$$\bar{\alpha}_m = \max_{x \in \Theta_m} |x|_2, \tag{4.9}$$

and the corresponding time-varying set

$$\bar{\mathbb{B}}_{\bar{r}_m(\tau)} = \bar{\mathbb{X}}_m \cap \mathcal{B}_{\bar{r}_m(\tau)}. \tag{4.10}$$

with $\bar{r}_m(\tau) = \sqrt{n_x} c_m \lambda_m^\tau \bar{\alpha}_m$. Then, the following result holds for any $k \in \mathbb{N}_0$ if $\Theta_m \subseteq \bar{\mathcal{X}}_{N_m}$.

Proposition 4.3. If mode m became active at the last switching instant

t_{k-1} with $\bar{x}(t_{k-1}) \in \Theta_m$, and the loop is closed with the m-MPC control law $\bar{\kappa}_m(\cdot)$, then the nominal state trajectory of the switching system fulfils $\bar{x}(t) \in \bar{\mathbb{B}}_{\bar{r}_m(t-t_{k-1})}$ for all $t \geq t_{k-1}$.

Proof. The proof is analogous to the proof of Proposition 4.2. Indeed, note that Θ_m is not invariant like $\bar{\mathcal{X}}_{N_m-1}$, however by construction the m-MPC controller guarantees constraint satisfaction and so the state trajectories must remain in $\bar{\mathbb{X}}_m$. If $\bar{x}(t_{k-1}) \in \Theta_m \subseteq \bar{\mathcal{X}}_{N_m}$ and the loop is closed with $\bar{\kappa}_m(\cdot)$, then $\bar{x}(t) \in \bar{\mathbb{X}}_m$ for all $t \geq t_{k-1}$. For any $t \geq t_{k-1}$, it follows from (4.6) that

$$\begin{aligned} &|\bar{x}(t)|_1 \leq \sqrt{n_x} c_m \lambda_m^{t-t_{k-1}} |\bar{x}(t_{k-1})|_2 \\ \implies &|\bar{x}(t)|_1 \leq \sqrt{n_x} c_m \lambda_m^{t-t_{k-1}} \bar{\alpha}_m \\ \implies &\bar{x}(t) \in \mathcal{B}_{\bar{r}_m(t-t_{k-1}-1)}, \end{aligned}$$

where the second inequality follows from (4.9) and the last one from the definition of a 1-norm ball. Hence, for all $t \geq t_{k-1}$, it holds that $\bar{x}(t) \in \bar{\mathbb{X}}_m$ and $\bar{x}(t) \in \mathcal{B}_{\bar{r}_m(t-t_k-1)}$, which given (4.10) completes the proof. ■

Analogously to Proposition 4.2, Proposition 4.3 ensures that the time varying set $\bar{\mathbb{B}}_{\bar{r}_m(\tau)}$, with decreasing Chebyshev radius, contains the state trajectories of the m closed-loop at all time instances; in this case, even if the corresponding feasibility regions are not known. The MDT that allows for feasible switching among neighbouring closed-loops is again defined in a straightforward manner.

Theorem 4.2. Consider any pair $m, l \in \mathcal{M}$ with $m \neq l$, $\sigma(t_{k-1}) = m$ and $l \in \mathcal{M}_m$. Furthermore, assume that mode m became active (feasibly) at the previous switching instance t_{k-1} with $\bar{x}(t_{k-1}) \in \Theta_m \subseteq \bar{\mathcal{X}}_{N_m}$. If $\bar{\tau}^f_{m,l} > 0$ is such that $\bar{\mathbb{B}}_{\bar{r}_m(\tau^f_{m,l})} \subseteq \Theta_l \subseteq \bar{\mathcal{X}}_{N_l}$ but $\bar{\mathbb{B}}_{\bar{r}_m(\tau^f_{m,l}-1)} \nsubseteq \Theta_l$, then a switch to mode l is feasible at any time t_k that fulfils $t_k - t_{k-1} \geq \bar{\tau}^f_{m,l}$.

Proof. The proof is identical to the proof of Theorem 4.1. ■

Similarly, given a set of modes $\mathcal{M}$ and a CSS, the following corollary establishes the MDTs for feasible switching.

Corollary 4.3. Assume $\sigma(\cdot)$ is a CSS. If for all $m \in \mathcal{M}$ the MDTs are set to τ^f_m defined by

$$\bar{\tau}^f_m = \max_{l \in \mathcal{M}_m} \bar{\tau}^f_{m,l},$$

then the switching control law (4.5) guarantees constraint satisfaction for the switching linear system (4.1).

Note that Theorem 4.2 provides the minimum required time to reach the set Θ_l, which is a subset of the true feasibility region of the l-MPC optimization. Nevertheless, this does not necessarily lead to longer feasibility MDTs because the starting point of this reachability problem is Θ_m, also a subset of $\bar{\mathcal{X}}_{N_m}$. The collection of sets Θ_m is not invariant, but returnable and contained inside the collection of feasibility sets, hence the overall region that contains the state at any given time remains as $\bar{\mathcal{X}} = \bigcup_{m\in\mathcal{M}} \bar{\mathcal{X}}_{N_m}$. However, this approach can only guarantee feasible switching when the plant is initialized inside $\Theta_{\sigma(0)}$, hence reducing the size of the allowable initialization region to $\bigcup_{m\in\mathcal{M}} \Theta_m \subseteq \bar{\mathcal{X}}$. The same trade-off is observed in the set-based approach presented in [108], except that the sets Θ_m are computed invariant.

The full region of attraction of the initial mode, however, can be recovered if the initial state is known before initializing the plant. Indeed, even if $\bar{x}(0)$ is not contained inside $\Theta_{\sigma(0)}$, whether it is contained inside $\bar{\mathcal{X}}_{N_{\sigma(0)}}$ can be easily determined by solving a single LP. Assume then $\bar{x}(0) \notin \Theta_{\sigma(0)}$ but $\bar{x}(0) \in \bar{\mathcal{X}}_{N_{\sigma(0)}}$ and define $r_{m,0}(\tau) = c_m\lambda_m^{\tau}|\bar{x}(0)|_2$ with an associated time-varying set

$$\bar{\mathbb{B}}_{r_{m,0}(\tau)} = \bar{\mathbb{X}}_m \cap \mathcal{B}_{r_{m,0}(\tau)}.$$

If $\tau_{m,0}$ is the smallest positive scalar such that $\bar{\mathbb{B}}_{r_{m,0}(\tau_{m,0})} \subseteq \Theta_m$, then $\tau_{m,0}$ is the minimum MDT required to guarantee $\bar{x}(\tau_{m,0}) \in \Theta_m$. Thereafter, the feasibility MDTs computed by Theorem 4.2 and Corollary 4.3 guarantee admissible switching, thus practically recovering the full size of the region of attraction. If the initial mode is unknown, then a generalized initialization MDT can be computed by taking the maximum over the set of modes, that is

$$\tau_0 = \max_{m\in\mathcal{M}} \tau_{m,0}.$$

It is left to discuss how the collection of sets Θ_m can be computed. Any subset of the corresponding RoA is a feasible choice, however the latter are unknown. A simple candidate would be the collection of terminal sets $\bar{\mathbb{X}}_{f,m}$, however this is a trivial choice because $\bar{\mathbb{X}}_{f,m}$ is nothing more than the 0-step stabilizable set to $\bar{\mathbb{X}}_{f,m}$. Furthermore, this choice would confine the switching to possibly small subset of the $\bar{\mathcal{X}}$, yielding a larger τ_0.

In order to obtain a larger collection of sets Θ_m note again that $\bar{x}(t) \in \bar{\mathcal{X}}_m$ if and only if $\mathbb{P}_m\left(\bar{x}(t)\right)$ is feasible, which gives way for the following result.

Proposition 4.4. Define the vertices of $\bar{\mathbb{X}}_m$ by $\{v_m^i\}$ for $i = 1, \ldots, n_m$. For all $i = 1, \ldots, n_m$ there exist $\bar{\beta}_m^i \in (0, 1]$ such that $\mathbb{P}_{N_m}\left(\beta_m^i v_m^i\right)$, with constraint

(2.1a) replaced by $\bar{x}_0 = \beta^i_m v^i_m$, is feasible for $\beta^i_m \in \left(0, \bar{\beta}^i_m\right]$ but infeasible for $\beta^i_m > \bar{\beta}^i_m$. Furthermore, $\Theta_m\left(\bar{\mathbb{X}}_m\right) = \operatorname{conv}\left\{\bar{\beta}^i_m v^i_m\right\} \subseteq \bar{\mathcal{X}}_{N_m}$.

Proof. First note that according to Proposition 4.1 the set $\bar{\mathcal{X}}_{N_m}$ has a non-empty interior, hence there exists $r > 0$ such that $\mathcal{B}_r \subseteq \bar{\mathcal{X}}_{N_m}$. It then follows from the compactness of $\bar{\mathbb{X}}_m$ that there exists $\beta > 0$ such that $\beta\bar{\mathbb{X}}_m \subseteq \mathcal{B}_r$. Finally, since $\bar{\mathcal{X}}_{N_m} \subseteq \bar{\mathbb{X}}_m$, then $\beta \leq 1$, which completes the proof. ■

According to Proposition 4.4 then, a feasible subset of $\bar{\mathcal{X}}_{N_m}$ can be computed from the vertices of the true state constraint set. The purpose of employing such a subset is to avoid the computational complexity involved in computing the exact feasibility regions, and indeed computing $\bar{\beta}^i_m$ is tractable even for high-dimension plants. To make this clear first note that to decide whether $\bar{\beta}^i{}_m v^i_m$ belongs to $\bar{\mathcal{X}}_{N_m}$ it is not necessary to solve $\mathbb{P}_{N_m}\left(\bar{\beta}^i{}_m v^i_m\right)$ to optimality, but only test whether it has a feasible solution. The exact values of $\bar{\beta}^i_m$ in Proposition 4.4 can then be easily found by solving, for each vertex, the linear program

$$\bar{\beta}^i_m = \arg\max_{\bar{\boldsymbol{u}}, \beta^i_m} \beta^i_m$$

subject to constraints (2.2)–(2.4) and $\bar{x}_0 = \beta^i_m v^i_m$, which can be done in polynomial time (see Section 3.5). Furthermore, if the vertices of $\bar{\mathbb{X}}_m$ are not available, any convex polyhedron in $\mathbb{R}^{n_x}$ can be used take its place in Proposition 4.4, while maintaining the validity of Theorem 4.2 and Corollary 4.3.

The level of conservativeness of the collection of sets Θ_m produced by Proposition 4.4, i.e. how smaller is $\bigcup_{m\in\mathcal{M}} \Theta_m$ when compared to $\bar{\mathcal{X}}$, depends solely on the amount of vertices of the starting polyhedrons. Indeed, define $\bar{\mathbb{X}}^{p_m}_m$ as a representation of $\bar{\mathbb{X}}_m$ with $p_m \geq 0$ redundant vertices placed along the original facets of $\bar{\mathbb{X}}_m$. It is easy to show that $\Theta_m\left(\bar{\mathbb{X}}^{p_m}_m\right) \subseteq \Theta_m\left(\bar{\mathbb{X}}^{q_m}_m\right)$ for any $0 \leq p_m \leq q_m$, thus better approximations of the true RoA of each mode can be obtained by increasing the number of redundant vertices in the representation of the chosen initial polyhedron.

4.3.2 MDTs for stabilizing switching

An MDT greater than or equal to τ^f_m (or $\bar{\tau}^f_m$) ensures that the optimization associated to the MPC controller of the destination mode is feasible, thereby ensuring recursive constraint satisfaction of the overall-switching system under the control law (4.5). In [114] an additional contraction demand is imposed over the dwell-time computation, which ultimately guarantees convergence of the state trajectories to an MDT-contractive set [131], and hence to the origin. This is not the case of the MDTs proposed by Theorems 4.1 and 4.2, hence although feasible, the state trajectories could be oscillating close to the boundaries of the feasibility regions, and never approach the origin. In order to ensure stability of the switched closed loop explicit stability demands must be placed on the MDTs.

The exponential stability result provided by Corollary 4.1 relies, primarily, on the value function $V_{N_m}(\cdot)$ being a Lyapunov function for the closed loop trajectories (4.3). However, the proposed control law (4.5) results in a different controller becoming active at the time of a switch. Although $\bar{x}(t_k)$ is the result of the continuous application of the $\bar{\kappa}_{\sigma(t_{k-1})}$ control law, the control action at t_k is defined by $\bar{\kappa}_{\sigma(t_k)}$ hence $V_{\sigma(t_{k-1})}(\bar{x}(t_k))$ is never evaluated. At the time of a switch then, different value functions must be compared and hence the upper bound on the rate (4.3a) does not necessarily hold. Nevertheless, exponential stability in the sense of Lyapunov can still be guaranteed throughout the switching intervals, rather than the specific switching times (similar to the approaches in [116, 117, 125]). In what follows it is shown that, provided sufficiently long MDTs are guaranteed, the function $V(\bar{x}(t)) = V_{N_{\sigma(t)}}(\bar{x}(t))$ is a Lyapunov-like function for the closed-loop trajectories and its existence guarantees exponential stability of the origin for the switching closed-loop.

Proposition 4.5. Consider any pair $m, l \in \mathcal{M}$ with $m \neq l$ and $l \in \mathcal{M}_m$. For any two switching instances $(t_k, \sigma(t_k) = m)$ and $(t_{k+1}, \sigma(t_{k+1}) = l)$ that fulfil the associated feasibility MDT, if there exists scalars $\bar{b}_{m,l}, \bar{d}_{m,l}, \bar{f}_{m,l} > 0$ such that for $V(\bar{x}(t)) = V_{N_{\sigma(t)}}(\bar{x}(t))$ fulfils

$$\bar{b}_{m,l}|\bar{x}(t)|_2^2 \leq V(\bar{x}(t)) \leq \bar{d}_{m,l}|\bar{x}(t)|_2^2 \quad \forall \bar{x}(t) \in \bar{\mathcal{X}}_{N_m} \cup \bar{\mathcal{X}}_{N_l} \tag{4.11a}$$

$$V(\bar{x}(t_{k+1})) - V(\bar{x}(t_k)) \leq -\bar{f}_{m,l}|\bar{x}(t_k)|_2^2 \quad \forall \bar{x}(t_k) \in \bar{\mathcal{X}}_{N_m} \;\; \bar{x}(t_{k+1}) \in \bar{\mathcal{X}}_{N_m} \cap \bar{\mathcal{X}}_{N_l}. \tag{4.11b}$$

for all $t \in [t_k, t_{k+1}]$, then it holds that

$$|\bar{x}(t_{k+1})|_2 \leq \bar{c}_{m,l}\bar{\lambda}_{m,l}|\bar{x}(t_k)|_2$$

for all $k \in \mathbb{N}_0$ with $\bar{c}_{m,l} = \sqrt{\bar{d}_{m,l}/\bar{b}_{m,l}} > 0$ and $\bar{\lambda}_{m,l} = \sqrt{(1 - \bar{f}_{m,l}/\bar{d}_{m,l})} \in (0, 1)$.

Proof. The proof is identical to the proof of exponential stability for standard MPC implementations (see [1, 2]). ■

In view of Proposition 4.5. exponential stability of the origin across the switching intervals follows.

Theorem 4.3. If Proposition 4.5 holds for all $(m,l) \in \mathcal{E}$ with $m \neq l$, then there exist scalars $\bar{c} > 0$ and $\bar{\lambda} \in (0,1)$ such that

$$|\bar{x}(t_{\bar{k}})|_2 \leq \bar{c}\bar{\lambda}^{t_{\bar{k}}-t_k}|\bar{x}(t_k)|_2$$

for any pair of switching instances $\bar{k}, k \in \mathbb{N}_0$ such that $t_{\bar{k}} > t_k$.

Proof. First note that for any three switching instances $(t_k, \sigma(t_k) = m)$, $(t_{k+1}, \sigma(t_{k+1}) = l)$ and $(t_{k+2}, \sigma(t_{k+2}) = n)$ with $l \in \mathcal{M}_m$, $n \in \mathcal{M}_l$ and such that the admissibility MDTs are fulfilled, it follows directly from Proposition 4.5 that

$$|\bar{x}(t_{k+2})|_2 \leq \bar{c}_{m,n}\bar{\lambda}_{m,n}|\bar{x}(t_k)|_2$$

with $\bar{c}_{m,n} = \sqrt{\bar{d}_{m,l}/\bar{b}_{l,n}} > 0$ and $\bar{\lambda}_{m,n} = \bar{\lambda}_{m,l}\bar{\lambda}_{l,n} \in (0,1)$. It follows then by induction that for any $\bar{k}, k \in \mathbb{N}_0$ with $\bar{k} > k$ it holds that

$$|\bar{x}(t_{\bar{k}})|_2 \leq \bar{c}\bar{\lambda}^{t_{\bar{k}}-t_k}|\bar{x}(t_k)|_2$$

with

$$\bar{c} = \sqrt{\frac{\max\limits_{(m,l)\in\mathcal{E}} \bar{d}_{m,l}}{\min\limits_{(m,l)\in\mathcal{E}} \bar{b}_{m,l}}}$$
$$\bar{\lambda} = \max_{(m,l)\in\mathcal{E}} \bar{\lambda}_{m,l}.$$

■

Theorem 4.3 guarantees a single, unified, convergence rate by choosing the slowest one amongst the mode-dependent rates provided by Proposition 4.5. The true convergence rate, however, remains dependent on the switching sequence and is expected to be considerably faster. Furthermore, convergence and boundedness throughout the switching interval, that is for all $t \in [t_k, t_{k+1})$, is readily guaranteed by the exponential stability result available for each independent MPC controller.

It has not yet been discussed how to compute the positive scalars that define the bounding functions in (4.11). The upper and lower bounds in (4.11a) are trivially fulfilled by setting $\bar{b}_{m,l} = \min_{j\in\{m,l\}} b_j$ and $\bar{d}_{m,l} = \max_{j\in\{m,l\}} d_j$. To compute a suitable $\bar{f}_{m,l}$, however, it is first necessary to establish a relation between the value functions of modes connected by an edge. For any pair of neighbouring modes $m, l \in \mathcal{M}$ then, define $\eta_{l,m} \geq {}^{d_l}/_{b_m} > 0$, where d_l and b_m are those in (4.3). It follows that

$$V_{N_l}(\bar{x}) \leq \eta_{l,m} V_{N_m}(\bar{x}) \quad \forall \bar{x} \in \bar{\mathcal{X}}_{N_m} \cap \bar{\mathcal{X}}_{N_l}. \tag{4.12}$$

Depending on the cost matrices employed, it may happen that $\eta_{l,m} > 1$, however it will always be finite, hence there exists a finite bound on how much larger the value function of a destination mode can be when compared to that of its source mode. The multiplicative bound in (4.12) is the discrete-time, linear analogous to the bound proposed in [127], and although valid for the subsequent developments, a tighter one exists. Indeed, note that (4.12) implies $V_{N_l}(\bar{x}) - V_{N_m}(\bar{x}) \leq (\eta_{l,m} - 1)\, d_m |\bar{x}|^2$, however it also follows from (4.3) that

$$V_{N_l}(\bar{x}) - V_{N_m}(\bar{x}) \leq \mu_{l,m} |\bar{x}|_2^2 \quad \forall \bar{x} \in \bar{\mathcal{X}}_{N_m} \cap \bar{\mathcal{X}}_{N_l}. \tag{4.13}$$

with $\mu_{l,m} \geq d_l - b_m$. Since $\mu_{l,m} \leq (\eta_{l,m} - 1)\, d_m$, (4.13) represents a tighter bound on the difference of neighbouring value functions. In view of (4.13) then, the following result holds for all $k \in \mathbb{N}_0$.

Theorem 4.4. Consider any pair $m, l \in \mathcal{M}$ with $m \neq l$ and $l \in \mathcal{M}_m$. For any two switching instances $(t_k, \sigma(t_k) = m)$ and $(t_{k+1}, \sigma(t_{k+1}) = l)$ that fulfil the associated feasibility MDT, if $t_{k+1} - t_k \geq \tau^s_{m,l}$ with $\tau^s_{m,l} > 0$ such that

$$\frac{\mu_{l,m} c_m^2 \lambda_m^{2\tau^s_{m,l}}}{d_m} < 1 - \lambda_m^{2\tau^s_{m,l}}, \tag{4.14}$$

then there exists a scalar $\bar{f}_{m,l} > 0$ such that

$$V\left(\bar{x}(t_{k+1})\right) - V(\bar{x}(t_k)) \leq -\bar{f}_{m,l} |\bar{x}(t_k)|_2^2 \tag{4.15}$$

with $V\left(\bar{x}(t)\right) = V_{N_{\sigma(t)}}\left(\bar{x}(t)\right)$.

Proof. First note that, since $V_{N_m}(\cdot)$ is a Lyapunov function for all $m \in \mathcal{M}$, it must happen that $f_m < d_m$, otherwise the rate inequality in (4.3b) does not hold. It follows then that ${}^{f_m}/_{d_m} \in (0, 1)$ and so the right hand side of (4.14) is positive, monotonically increasing on $\tau^s_{m,l}$ and bounded above by 1. If $\mu_{l,m} < 0$, the left hand side of (4.14) is negative and fulfilling (4.15) is trivial. If $\mu_{l,m} > 0$, the right hand side of (4.14) is positive but monotonically decreasing on $\tau^s_{m,l}$ and bounded below by 0. This implies that there exists a finite $\tau^s_{m,l}$ such that

(4.14) holds, and that

$$\frac{\mu_{l,m} c_m^2 \lambda_m^{2(t_{k+1}-t_k)}}{d_m} < 1 - \lambda_m^{2(t_{k+1}-t_k)}, \tag{4.16}$$

holds since $t_{k+1} - t_k \geq \tau_{m,l}^s$ by assumption. Secondly note that (4.3b) and the upper bound in (4.3a) result in

$$V_{N_m}(\bar{x}(t_{k+1})) \leq \lambda_m^{2(t_{k+1}-t_k)} V_{N_m}(\bar{x}(t_k)).$$

It follows then, from the upper bound in (4.3a), that

$$V_{N_m}(\bar{x}(t_{k+1})) - V_{N_m}(\bar{x}(t_k)) \leq -\left(1 - \lambda_m^{2(t_{k+1}-t_k)}\right) d_m |\bar{x}(t_k)|_2^2 \tag{4.17}$$

From (4.17), and the additive bound (4.13) evaluated at time t_{k+1}, it follows that

$$\begin{aligned} V_{N_l}(\bar{x}(t_{k+1})) - V_{N_m}(\bar{x}(t_k)) \leq & -\left(1 - \lambda_m^{2(t_{k+1}-t_k)}\right) d_m |\bar{x}(t_k)|_2^2 \\ & + \mu_{l,m} c_m^2 \lambda_m^{2(t_{k+1}-t_k)} |\bar{x}(t_k)|_2^2. \end{aligned} \tag{4.18}$$

From (4.16) it follows that the left hand side of (4.18) is negative and so (4.15) is met with

$$\bar{f}_{m,l} = \left(1 - \lambda_m^{2\tau_{m,l}^s}\right) d_m - \mu_{l,m} c_m^2 \lambda_m^{2\tau_{m,l}^s}.$$

■

The central part of Theorem 4.4 is the inequality (4.14), which is formed by scalar functions of $\tau_{m,l}^s$. It follows that a $\tau_{m,l}^s$ that meets (4.14) can be easily found by iterating over the set of integers. Furthermore, if $\tau_{m,l}^s$ is chosen such that (4.14) does not hold for $\tau_{m,l}^s - 1$, then $\tau_{m,l}^s$ represents the minimum MDT that guarantees a positive $\bar{f}_{m,l}$ and so stable switching between modes m and l. In what follows it is assumed that $\tau_{m,l}^s$ is computed to be minimal. Given Theorem 4.4, it is straightforward to compute the required $\bar{f}_m$ for each mode by taking the minimum over all allowable switches.

Corollary 4.4. Assume $\sigma(\cdot)$ is a CSS. If for all $m \in \mathcal{M}$ the MDTs are set to τ_m^s defined by

$$\tau_m^s = \max_{l \in \mathcal{M}_m} \tau_{m,l}^s,$$

then the switching control law (4.5) guarantees (4.11b) is met with $V(\bar{x}(t)) = V_{N_{\sigma(t)}}(\bar{x}(t))$ and

$$\bar{f}_m = \min_{l \in \mathcal{M}_m} \bar{f}_{m,l},$$

hence Proposition 4.5 and Theorem 4.3 ensure that the origin is exponential stable for the switching closed-loop under control law (4.5).

4.3.2.1 Dynamically adjacent value functions

The implementation of the switching control law (4.5) implies that at any given switching instance t_k there is only one MPC controller active, and so $u(t_k-1) = \bar{\kappa}_{\sigma(t_{k-1})}(\bar{x}(t_k-1))$ but $u(t_k) = \bar{\kappa}_{\sigma(t_k)}(\bar{x}(t_k))$. Furthermore, setting $V(\bar{x}(t)) = V_{N_{\sigma(t)}}(\bar{x}(t))$ results in that $V_{N_{\sigma(t_{k-1})}}(x(t_k))$ is never really part of the overall Lyapunov equation. Ultimately, this implies that although valid, the additive bound in (4.13) may result in unnecessarily conservative MDTs since two value functions are being compared at the same state however only one is ever evaluated. Figure 4.1 presents a diagram that clarifies this situation.

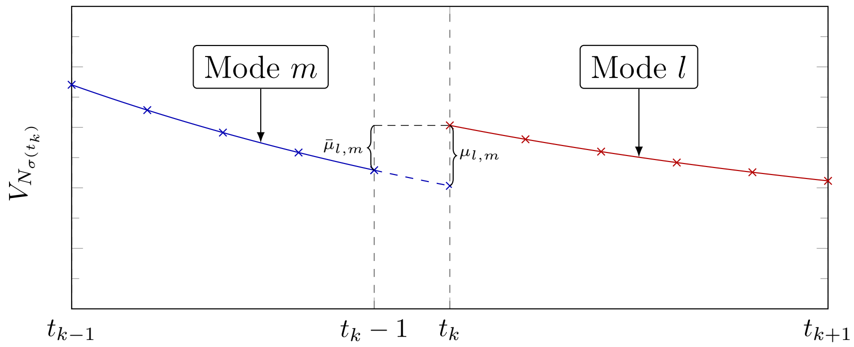

Figure 4.1: Bound on the increase of the cost function during a switch: $V_m(x(t))$, $V_l(x(t))$.

An alternative, then, is to compare the neighbouring value functions at the dynamically adjacent states $\bar{x}(t_k-1)$ and $\bar{x}(t_k)$. For any pair of neighbouring modes $m, l \in \mathcal{M}$ with $l \in \mathcal{M}_m$ define then $\bar{\mu}_{l,m} \geq \mu_{l,m} - f_l$. It follows that if $\bar{x} \in \bar{\mathcal{X}}_{N_m}$, then

$$V_{N_l}(A_m\bar{x} + B_m\bar{\kappa}_m(\bar{x})) - V_{N_m}(\bar{x}) \leq \bar{\mu}_{l,m}|\bar{x}|^2 \tag{4.19}$$

for all $(A_m\bar{x} + B_m\bar{\kappa}_m(\bar{x})) \in \bar{\mathcal{X}}_{N_m} \cap \bar{\mathcal{X}}_{N_l}$. Inequality (4.19) provides an additive bound on the change of the optimal value functions at dynamically adjacent states, and allows for an analogous result to Theorem 4.4.

Proposition 4.6. Consider any pair $m, l \in \mathcal{M}$ with $m \neq l$ and $l \in \mathcal{M}_m$. For any two switching instances $(t_k, \sigma(t_k) = m)$ and $(t_{k+1}, \sigma(t_{k+1}) = l)$ that fulfil the associated feasibility MDT, if $t_{k+1} - t_k \geq \bar{\tau}^s_{m,l}$ with $\bar{\tau}^s_{m,l} > 0$ such that

$$\frac{\bar{\mu}_{l,m} c_m^2 \lambda_m^{2\left(\bar{\tau}^s_{m,l}-1\right)}}{d_m} < 1 - \lambda_m^{2\left(\bar{\tau}^s_{m,l}-1\right)}, \tag{4.20}$$

then the origin is exponentially stable for the switched closed-loop, with respect to the switching instants.

Proof. Follows directly from Theorems 4.3 and 4.4. ■

The same arguments presented in the proof of Theorem 4.4 can be used to guarantee the existence of a finite $\bar{\tau}^s_{m,l}$ such that (4.20) is met, and similarly $\bar{\tau}^s_{m,l}$ can be chosen minimal. Corollary 4.4 then, applies without changes to compute the analogous $\bar{\tau}^s_m$. It is not trivial, however, to guarantee $\bar{\tau}^s_{l,m} \leq \tau^s_{l,m}$ (or vice-versa). Indeed, a preliminary inspection shows that the right hand side of (4.20) is upper bounded by the right hand side of (4.14), but $\bar{\mu}_{l,m}$ is strictly smaller than $\mu_{l,m}$. Which stability inducing MDT is less demanding depends then on the particular values of the different bounding constants. In view of the latter is that it is proposed to compute both and choose the smaller one.

4.3.3 MDTs for admissible and stabilizing switching

A key requirement that is not explicitly stated in Corollary 4.4 (albeit clear from Theorem 4.4 and Proposition 4.6) is that (4.11b) and (4.13) (or (4.19)) are only valid in the intersection of the corresponding feasibility regions, hence stable switching cannot be guaranteed independently of admissible switching. The following corollary then brings both results together to guarantee stable and admissible switching under the proposed control law (4.5).

Corollary 4.5. If the feasibility regions $\bar{\mathcal{X}}_{N_m}$ have been computed and the minimum MDT for each mode is set to τ_m defined by

$$\tau_m = \max\left\{\tau^f_m, \min\left\{\tau^s_m, \bar{\tau}^s_m\right\}\right\},$$

the control law (4.5) results in constraint admissible closed-loop trajectories that converge exponentially fast to the origin (at different rates for each active mode). Alternatively, if the feasibility regions are not available, the minimum required MDTs are

$$\tau_m = \max\left\{\bar{\tau}^f_m, \min\left\{\tau^s_m, \bar{\tau}^s_m\right\}\right\},$$

and the full region of attraction ca be recovered by enforcing the initialization MDT τ_0.

4.4 Disturbed switching linear systems: independent design

The standard tube-based MPC implementation [2] provides robust control under minimal additional design requirements with respect to standard non-robust MPC implementations. The only additional design parameters are a stabilizing linear feedback and a corresponding admissible RPI set $\mathbb{S}_m$ to work as the cross section of the tube. However, the MDT results obtained for the disturbance-free case, in the previous section, cannot be cast for the robust case without some additional considerations.

4.4.1 MDTs for robustly admissible switching

The main consequence of accounting for additive disturbances such as $w(t)$ in (4.1) is that the origin cannot be rendered (exponentially) stable, even in the non-switching case. The tube-based MPC described in Chapter 2 and recast in Section 4.2.1 can only guarantee exponential stability of $\mathbb{S}_m$ for the true state (provided mode m remains indefinitely active). Since $\mathbb{S}_m$ is an RPI set for the error dynamics of mode m, once $x(t) \in \mathbb{S}_m$ it remains inside for all future time instances as long as mode m continues to be active. It follows that $\mathbb{S}_m$ is the smallest neighbourhood of the origin that can be guaranteed to contain the state at any given time instance, as opposed to the time-varying set $\mathbb{B}_r$ defined in (4.8) that converges, in the limit, to the origin. Ultimately, this implies that feasible switching between neighbouring modes, say m and l, can only be guaranteed if $\mathbb{S}_m \subseteq \mathcal{X}_{N_l}$, thus the following assumption is required.

Assumption 4.1. For all $(l, m) \in \mathcal{E}$ it holds that $\mathbb{S}_m \subset \mathcal{X}_{N_l} = \mathbb{S}_l \oplus \bar{\mathcal{X}}_{N_l}$.

If Assumption 4.1 is met, the approach used to compute admissibility MDTs for the disturbance-free case can be easily extended to the perturbed case. Indeed, (4.4) guarantees that the origin remains exponentially stable for the optimized nominal state trajectories of each independent closed-loop, despite the perturbation introduced by $w(t)$. This, in turn, implies that the bounding set provided by Proposition 4.2 remains a valid bound for the nominal state trajectories in the perturbed case. Intuitively then, Theorem 4.1 could be considered for the definition of MDTs in the robust case, however note that for any given pair of neighbouring modes m and l, it does not necessarily hold that $\mathbb{S}_m \subseteq \mathbb{S}_l$, hence even if the nominal closed-loop of mode m reaches $\bar{\mathcal{X}}_{N_l}$, the true state is only guaranteed to be in $\mathbb{S}_m \oplus \bar{\mathcal{X}}_{N_l}$ and not necessarily in $\mathcal{X}_{N_l}$. This is why the TMPC approach that optimizes nominal trajectories is selected over the one that lets them evolve independently. The former allows to easily overcome this issue by modifying the set inclusion that needs to be tested in Theorem 4.1.

Theorem 4.5. Consider any pair $m, l \in \mathcal{M}$ with $m \neq l$, $\sigma(t_{k-1}) = m$ and $l \in \mathcal{M}_m$. Furthermore, assume that mode m became active (feasibly) at the previous switching instance t_{k-1}. If Assumption 4.1 holds and $\tau^f_{m,l} \geq 0$ is such that $\mathbb{S}_m \oplus \mathbb{B}_{r_m(\tau^f_{m,l})} \subseteq \mathbb{S}_l \oplus \bar{\mathcal{X}}_{N_l}$ but $\mathbb{S}_m \oplus \mathbb{B}_{r_m(\tau^f_{m,l}-1)} \not\subseteq \mathbb{S}_l \oplus \bar{\mathcal{X}}_{N_l}$, then a switch to mode l is feasible at any time t_k that fulfils $t_k - t_{k-1} - 1 \geq \tau^f_{m,l}$.

Proof. The proof is identical to the proof of Theorem 4.1. Existence of a finite $\tau^f_{m,l}$ is guaranteed by the strict inclusion demanded in Assumption 4.1 and the decreasing Chebyshev radius of $\mathbb{B}_{r_m(\tau^f_{m,l})}$ as defined in (4.8). ■

In view of Theorem 4.5 then, minimum MDTs that guarantee feasible switching among independently designed TMPC controllers can be defined.

Corollary 4.6. Assume $\sigma(\cdot)$ is a CSS. If for all $m \in \mathcal{M}$ the MDTs are set to τ^f_m defined by

$$\tau^f_m = 1 + \max_{l \in \mathcal{M}_m} \tau^f_{m,l},$$

then the switching control law (4.5) guarantees robust constraint satisfaction for the switching linear system (4.1).

Theorem 4.5 and Corollary 4.6 parallel the disturbance-free results and provide an approach to compute MDTs that guarantee admissible switching, albeit demanding the knowledge of the nominal feasibility regions. Nevertheless, the results provided in Section 4.3.1.2 for the disturbance-free case are also valid for the perturbed case with the appropriate modifications for the set inclusion that needs to be verified (as in Theorem 4.5).

4.4.2 MDTs for stabilizing switching

In a non-switching scenario an individual TMPC controller can guarantee exponential stability of the RPI set $\mathbb{S}_m$ for the true state trajectories as a consequence of the origin being exponentially stable for the nominal optimized trajectories. It would be expected then that in a switching scenario the set

$$\mathcal{O}_s = \bigcup_{m \in \mathcal{M}} \mathbb{S}_m \tag{4.21}$$

can be shown to be (exponentially) stable by readily extending the stability inducing MDTs computed in Section 4.3.2, yet this is not trivial to achieve. The main reason is that the RPI sets of neighbouring modes are designed in an independent fashion hence $(m,l) \in \mathcal{E}$ does not imply $\mathbb{S}_m \subseteq \mathbb{S}_l$. In view of this, even if $x(t_{k-1}) \in \mathbb{S}_m$, it is not possible to guarantee that at the switching instance $x(t_k) \in \mathbb{S}_l$, which ultimately results in an increase of the value function across switching intervals that cannot be countered by any finite dwell-time.

In order to see this first define $\bar{x}_m(t) = \bar{x}^*_{0,m}(x(t))$ as the optimal value of the predicted nominal state obtained by solving $\mathbb{P}_{N_m}(x(t))$. For any pair of neighbouring modes $m, l \in \mathcal{M}$, heterogeneity of modes and cost functions will, almost surely, yield $\bar{x}^*_{0,m}(x(t)) \neq \bar{x}^*_{0,l}(x(t))$ for all $x(t) \notin \mathbb{S}_m \cap \mathbb{S}_l$. It follows then that a bound like (4.13) depends on both optimized variables,

$$V_{N_l}(x(t)) - V_{N_m}(x(t)) \leq d_l |\bar{x}_l(t)|_2^2 - b_m |\bar{x}_m(t)|_2^2 \quad \forall x \in \mathcal{X}_{N_m} \cap \mathcal{X}_{N_l}. \tag{4.22}$$

The same chain of arguments employed in Theorem 4.4 would then yield

$$V_{N_l}(x(t_{k+1})) - V_{N_m}(x(t_k)) \leq d_l |\bar{x}_l(t_{k+1})|_2^2 - d_m |\bar{x}_m(t_k)|_2^2. \tag{4.23}$$

If $x(t_k) \in \mathbb{S}_m$, then $\bar{x}_m(t_k) = 0$, and so the rate of change is upper bounded by a positive function, and a decrease cannot be guaranteed. Furthermore, the rate in (4.23) depends not only on the nominal state at the previous switching instance, but also at the current one. The value of $|\bar{x}_l(t_{k+1})|$ can be related to that of $|\bar{x}_m(t_k)|$, however through the explicit characterization of the control law $\kappa_m(\cdot)$ (or $\kappa_l(\cdot)$), thus (4.22) does not provide an explicit bound on the rate of change of $V(x(t)) = V_{N_{\sigma(t)}}(x(t))$ as opposed to (4.18).

Despite these issues, it is still possible to establish stability of a neighbourhood of the origin for the switching closed-loop under control law (4.5), albeit not by directly comparing value functions, but by employing the robust invariance properties of $\mathbb{S}_m$. Indeed, if the admissibility inducing MDTs computed in Section 4.4.1 are enforced, it holds that for $x(0) \in \mathcal{O}$ with

$$\mathcal{O} = \bigcup_{m \in \mathbb{M}} \left(\mathbb{S}_m \oplus \bar{\mathcal{X}}_{N_m}\right) \tag{4.24}$$

the closed-loop remains inside $\mathcal{O}$ for all future time instances, or in other words, $\mathcal{O}$ is a robust control invariant set for the switching closed-loop. In a similar way, smaller neighbourhoods of the origin can be rendered invariant provided long enough MDTs are satisfied. To do so first suppose that a collection of sets $\{\Omega_m\}_{m\in\mathcal{M}}$ that meets the following assumption is available.

Assumption 4.2. For all $m \in \mathcal{M}$ the set Ω_m is a such that

$$\bar{x}^*_{0,m}(x(t)) \in \Omega_m \implies \bar{x}^*_{0,m}(x(t+1)) \in \Omega_m \tag{4.25a}$$

$$\Omega_m \subseteq \bar{\mathcal{X}}_{N_m} \tag{4.25b}$$

$$\mathbb{S}_m \subset \mathbb{S}_l \oplus \Omega_l, \quad \forall l \in \mathcal{M}_m. \tag{4.25c}$$

It can be shown that if such a collection of sets exists, and given a particular collection of MDTs, the set

$$\mathcal{O}_g = \bigcup_{m\in\mathcal{M}} (\mathbb{S}_m \oplus \Omega_m) \tag{4.26}$$

is robust invariant for the switching closed-loop, yet finding such a group of sets is not an easy task. The property outlined by (4.25a) implies that Ω_m is an invariant set for the optimized nominal state trajectories, but this is not the same as Ω_m being PI for the nominal dynamics. The latter requires that

$$\bar{x}^*_{0,m}(x(t)) \in \Omega_m \implies A_m\bar{x}^*_{0,m}(x(t)) + B_m\bar{\kappa}_m(\bar{x}^*_{0,m}(x(t))) \in \Omega_m,$$

yet $\bar{x}^*_{0,m}(x(t+1))$ is defined by the optimization at time $t+1$ and hence is not necessarily equal to $A_m\bar{x}^*_{0,m}(x(t))+B_m\bar{\kappa}_m(\bar{x}^*_{0,m}(x(t)))$. In general any sub-level set of the corresponding value function, $V_{N_m}(\cdot)$ guarantees that (4.25a) is met, and there exists an infinite number of sub-level sets that meet (4.25b). The inclusion in (4.25c), however, is not necessarily fulfilled by any sub-level set and it would be up to the design process to verify this. Additionally, computing sub-level sets of the constrained m-TMPC value function is not a trivial task. For unconstrained linear systems stabilized by a linear control law, these sets are characterized by simple ellipsoids (given the quadratic cost). The MPC control law, on the other hand, is rendered non-linear due to the state constraints and implicit due to the optimization, hence its sub-level sets need to be obtained numerically [1].

There are, however, two simple sets that can serve as candidates to meet Assumption 4.2. The first option is the trivial choice $\Omega_m = \bar{\mathcal{X}}_{N_m}$. By Assumption 4.1 and the recursive feasibility property of the individual TMPC controllers, it is easy to show that $\bar{\mathcal{X}}_{N_m}$ fulfils all the requirements of Assumption 4.2. The drawback of such selection is that nothing is gained from the regulation perspective as $\mathcal{O}_g = \mathcal{O}$ is the largest neighbourhood of the origin that can be rendered attractive. A second option is $\Omega_m = \bar{\mathbb{X}}_{f,m}$, which by definition fulfils (4.25b). In general, $\bar{\mathbb{X}}_{f,m}$ does not fulfil (4.25a), however a specific design choice can guarantee it does.

In Section 4.2.1 the terminal cost is designed to match exactly the infinite horizon unconstrained LQR cost. If additionally the terminal controller K_m is chosen to equate the optimal LQR gain associated to the cost matrices Q_m and R_m, it follows by optimality that $\bar{x}_m(t) = \bar{x}^*_{0,m}(x(t)) \in \bar{\mathbb{X}}_{f,m}$ and $\bar{u}_m(t) = \bar{u}^*_{0,m}(x(t)) = K_m \bar{x}^*_{0,m}(x(t))$ for all $x(t) \in \mathbb{S}_m \oplus \bar{\mathbb{X}}_{f,m}$. Furthermore, the composite control law $\kappa_m(\cdot)$ results in

$$
\begin{aligned}
x(t+1) \quad &= Ax(t) + B\kappa(x(t)) + w(t) \\
&= Ax(t) + B\left(\bar{u}_m(t) + \bar{K}_m\left(x(t) - \bar{x}_m(t)\right)\right) + w(t) \\
&= Ax(t) + B\bar{u}_m(t) + B\bar{K}_m e(t) + w(t) \\
&= Ax(t) + BK_m\bar{x}_m(t) + B\bar{K}_m e(t) + w(t) + \left(A\bar{x}_m(t) - A\bar{x}_m(t)\right) \\
&= \left(A + B\bar{K}_m\right)e(t) + w(t) + \left(A + BK_m\right)\bar{x}_m(t) \\
\implies x(t+1) \quad &\in \mathbb{S}_m \oplus \bar{\mathbb{X}}_{f,m}
\end{aligned}
\tag{4.27}
$$

and so $\bar{x}_m(t+1) = \bar{x}^*_{0,m}(x(t+1))$ is also contained in $\in \bar{\mathbb{X}}_{f,m}$; thereby $\bar{\mathbb{X}}_{f,m}$ also fulfils (4.25a). Note that in (4.27) $\bar{K}_m$ and K_m are employed to refer to the tube and the terminal gains correspondingly. This is done to emphasize that these gains need not to be the same. The terminal gain K_m can then be set to the LQR gain associated to the m-TMPC cost to guarantee that $\bar{\mathbb{X}}_{f,m}$ fulfils (4.25a) while the tube gain $\bar{K}_m$ can be designed to guarantee that $\bar{\mathbb{X}}_{f,m}$ meets the inclusion condition (4.25c). Finally, note that provided the design conditions discussed above are met, any $\Omega_m = \varepsilon\bar{\mathbb{X}}_{f,m}$ with $\varepsilon \in [0, 1]$ fulfils (4.25a) and (4.25b), so if $\bar{\mathbb{X}}_{f,m}$ meets (4.25c) with strict inclusion, an even smaller neighbourhood of the origin can be rendered (exponentially) stable by appropriate scaling of the terminal set. Nevertheless, if $\mathbb{S}_m \not\subseteq \mathbb{S}_l$, then Ω_l must have a non-zero Chebyshev radius.

4.4.2.1 Feasibility implies stability

Independent of how the collection $\{\Omega_m\}_{m\in\mathcal{M}}$ is computed, its existence guarantees that $\mathcal{O}_g$ is robust positive invariant for the switching closed-loop, provided certain MDTs are enforced. To prove this first define the time-varying set

$$\hat{\mathbb{B}}_{\hat{r}_m(t)} = \Omega_m \cap \mathcal{B}_{\hat{r}_m(\tau)} \tag{4.28}$$

with $\hat{r}_m(t) = \sqrt{n_x} c_m \lambda_m^\tau \max_{x\in\Omega_m} |x|_2$. Since Ω_m is assumed invariant for the optimized nominal dynamics, the same arguments used in the proof of Proposition 4.2 guarantee that, if mode m became feasibly active at time t_k then $x(t) \in \mathbb{S}_m \oplus \hat{\mathbb{B}}_{\hat{r}_m(t-t_k)} \subseteq \mathbb{S}_m \oplus \Omega_m$ for all $t > t_k$ as long as mode m remains active. In view of this the robust invariance of $\mathcal{O}_g$ follows.

Proposition 4.7. If Assumption 4.2 holds, there exists $\tau_{m,l}^g \geq 1$ such that

$$\mathbb{S}_m \oplus \hat{\mathbb{B}}_{\hat{r}_m(\tau_{m,l}^g)} \subseteq \mathbb{S}_l \oplus \Omega_l, \tag{4.29}$$

for all pairs $(m,l) \in \mathcal{E}$. Furthermore, if the feasibility MDTs for each mode are set to $\tau_m^g = \max_{l\in\mathcal{M}} \tau_{m,l}^g$, the set $\mathcal{O}_g$ is RPI for the switched closed-loop dynamics under the control law (4.14).

Proof. The existence of a finite $\tau_{m,l}^g$ such that the inclusion (4.29) holds follows from the strict inclusion requirement in (4.25c) and the fact that the Chebyshev radius of $\hat{\mathbb{B}}_{\hat{r}_m(t)}$, as defined in (4.28) is exponentially decreasing. Robust invariance of $\mathcal{O}_g$ follows from the invariance of the collection $\{\Omega_m\}_{m\in\mathcal{M}}$ and the definition of the required minimum MDT to the maximum over all allowable switches for each mode. ■

Similar to the result observed in Section 4.3.1.2 for unknown feasibility regions, the MDTs that guarantee invariance of $\mathcal{O}_g$ are not necessarily longer than those that guarantee invariance of $\mathcal{O}$. Nevertheless, since there is no contraction condition imposed in the computation of the feasibility MDTs that render $\mathcal{O}$ invariant, there is no immediate guarantee of convergence to $\mathcal{O}_g$. Indeed, $\mathcal{O}_g$ is robustly invariant but the states might never reach it. A simple, yet arguably conservative way of ensuring that $\mathcal{O}_g$ is robustly exponentially stable for the switching closed-loop is to impose a long enough MDT over a single mode, say $\bar{m}$, such that $\mathbb{S}_{\bar{m}} \oplus \Omega_{\bar{m}}$ is reached before another switch. The following theorem establishes this result formally.

Theorem 4.6. Suppose a collection of sets $\{\Omega_m\}_{m\in\mathcal{M}}$ that fulfils Assumption 4.2 is known and that τ_m^g are the MDTs computed for such a collection of sets following Proposition 4.3. If (a) the MDTs for each mode are set to $\hat{\tau}_m^f = \max\left\{\tau_m^f, \tau_m^g\right\}$ for all $m \in \mathcal{M}$, and (b) for at least one $\bar{m} \in \mathcal{M}$ the MDT is further extended to $\tau_{\bar{m}} = \max\left\{\hat{\tau}^f_{\bar{m}}, \tau^s_{\bar{m}}\right\}$ with $\tau^s_{\bar{m}}$ such that $\mathbb{B}_{r_{\bar{m}}(\tau^s_{\bar{m}})} \subseteq \hat{\mathbb{B}}_{\hat{r}_{\bar{m}}(\tau^g_{\bar{m}})}$, then (1) the overall feasibility region of the switching control law (4.5) is $\mathcal{O}$, (2) as soon as $\sigma(t_k) = \bar{m}$ the true state enters $\mathcal{O}_g$ in finite time posterior to the switch into mode $\bar{m}$ and (3) it remains therein for all future time instances.

Proof. First, by the definition of $\hat{\tau}^f_m$ it holds that $\hat{\tau}^f_m \geq \tau^f_m$ and so robust invariance of $\mathcal{O}$ is guaranteed. Second, since $r_{\bar{m}}(\tau^s_{\bar{m}})$ is such that $\mathbb{B}_{r(t)} \subseteq \hat{\mathbb{B}}_{\hat{r}_{\bar{m}}(\tau^g_{\bar{m}})}$ for all $t \geq \tau^s_{\bar{m}}$, if mode $\bar{m}$ became feasible active at time t_k, then $x(t) \in \mathbb{S}_{\bar{m}} \oplus \Omega_{\bar{m}}$ for all $t \geq t_k + \tau^s_{\bar{m}}$. Furthermore, since (4.29) holds, then $x(t) \in \mathbb{S}_l \oplus \Omega_l$ for all $l \in \mathcal{M}$, hence a switch is feasible, and the state remains inside $\mathcal{O}_g$ for all future time instances and switches due to the invariance of $\mathcal{O}_g$ given $\hat{\tau}^f_m$. ∎

Theorem 4.6 proposes a way to compute MDTs that ensure, not only that $\mathcal{O}$ is the RoA of the proposed switching control law (4.5), but also to guarantee convergence (in finite time) to a neighbourhood of the origin defined by $\mathcal{O}_g$. The latter is guaranteed to be a superset of the set in (4.21) and will most likely be a strict superset. This is a shortcoming when compared to non-switching systems, but an acceptable trade-off given the heterogeneity of the modes and the independent design approach that results in the several obstacles previously discussed.

The finite time convergence guarantee provided by Theorem 4.6 requires that a single mode is fixed for $\tau^s_{\bar{m}}$ time steps until $\mathbb{S}_{\bar{m}} \oplus \Omega_{\bar{m}}$ has been reached. Since the objective is to render stable the smallest possible neighbourhood of the origin, $\tau^s_{\bar{m}}$ may be large, and hence translate into a conservative dwell-time requirement. Nevertheless, conservatism can be reduced by resorting to sub-optimal solutions. Indeed, for any $x(t) \in \mathcal{X}_{N_m}$ the solution to $\mathbb{P}_{N_m}(x(t))$ is a sequence of N_m control actions that, if applied unchanged, guarantee $x(t + N_m) \in \mathbb{S}_m \oplus \bar{\mathbb{X}}_{f,m}$ independent of the disturbances. Hence if $\Omega_{\bar{m}} = \bar{\mathbb{X}}_{f,\bar{m}}$ and $\tau^s_{\bar{m}} \gg N_{\bar{m}}$, the sub-optimal sequence of inputs can be employed to reduce the time required to reach $\mathcal{O}_g$.

4.5 Disturbed switching linear systems: coupled design

The MDTs computed in Section 4.4 guarantee that the control law (4.5) results in a constraint admissible and robustly stable switching closed-loop independent of the arbitrary switching and the action of the disturbances. The MDTs that guarantee admissible switching are computed with a straightforward extension of the approach proposed for the disturbance-free case, however the same is not possible for the stability inducing MDTs. The latter is, mainly, due to the nominal state trajectories being an optimization variable of the proposed robust controllers, hence allowing for the possibility of a cost increase. This can be overcome by employing the version of TMPC that allows for independently evolving nominal trajectories (described in Section 2.3.2), however additional considerations are required to guarantee admissible switching under this architecture.

In what follows all reference to TMPC controllers or control laws is made assuming the version that does not optimize the nominal trajectories has been implemented for each mode. If this is the case, the nominal trajectories still converge exponentially fast to the origin. Thereby Proposition 4.2 and Theorem 4.1 can again be used to find minimum MDTs that guarantees switches that are nominally feasible, yet not necessarily robustly feasible.

Indeed, if mode m became feasible active at time t_{k-1}, it ought to be that $\bar{x}(t_{k-1}) \in \bar{\mathcal{X}}_{N_m}$ and $e(t_{k-1}) \in \mathbb{S}_m$. If the true open-loop is closed with $\kappa_m(\cdot)$ and the nominal one with $\bar{\kappa}_m(\cdot)$, then there exist a finite time $\bar{t} > t_{k-1}$ such that $\bar{x}(\bar{t}) \in \mathbb{B}_{r_m(\bar{t}-t_{k-1}-1)} \subseteq \bar{\mathcal{X}}_{N_l}$ where l is any neighbouring mode of m. If a switch takes place, $\mathbb{P}_{N_l}\left(\bar{x}(\bar{t})\right)$ is feasible and can be solved, resulting in $\bar{u}(\bar{t}) = \bar{\kappa}_l(\bar{x}(\bar{t})) \in \bar{\mathbb{U}}_l$, however the robust invariance property of the corresponding RPI sets can only guarantee $e(\bar{t}) \in \mathbb{S}_m$, hence the composite control law $\kappa_l(\cdot)$ results in

$$u(\bar{t}) = \bar{u}(\bar{t}) + K_l e(\bar{t}) \in \bar{\mathbb{U}}_l \oplus K_l \mathbb{S}_m. \tag{4.30}$$

By definition $\bar{\mathbb{U}}_l \oplus K_l \mathbb{S}_l \subseteq \mathbb{U}_l$, but $\bar{\mathbb{U}}_l \oplus K_l \mathbb{S}_m$ is not necessarily a subset of $\mathbb{U}_l$. The control action that results from the switching control law (4.5) could then violate the input constraints. Moreover, even if input constraints are respected, it follows that

$$e(\bar{t}+1) = x(\bar{t}+1) - \bar{x}(\bar{t}+1) \in \left(A_l + B_l K_l\right) \mathbb{S}_m \oplus \mathbb{W}_l, \tag{4.31}$$

which, again, is not necessarily a subset of $\mathbb{S}_l$. Hence even if the nominal state at time $\bar{t}$ is inside the nominal RoA of the l-TMPC controller, the dynamics and disturbances associated to the newly active mode could result in the violation of the true (non-tightened) state constraints at time $\bar{t}+1$.

In summary then, employing the version of TMPC that does not optimize trajectories gives rise to two specific constraint related issues. These are not present in the approach detailed in Section 4.4.1 because the individual TMPC controllers optimize the nominal trajectories to guarantee $e(\bar{t}) \in \mathbb{S}_l$; yet this feature is also the main obstacle in computing simple stabilizability MDTs using the standard exponential stability result of the independent controllers. In what follows, solutions are proposed to tackle both constraint challenges, in an attempt to guarantee feasible switching between TMPC controllers that do not optimize trajectories, and thereby allow for a simple extension of the disturbance-free approach to computing stability MDTs.

4.5.1 Invariant multi-sets

In order to tackle the state constraint satisfaction issue that stems from the fact that the RPI sets are computed independently (4.31), a coupled design approach is proposed. Particularly, the concept of multi-set invariance, introduced in [136], is employed. Multi-set invariance has some similarities to the idea of invariant families of sets [138], however tailored for switching systems rather than distributed systems. In what follows, the multi-sets are reported as $\bar{\mathbb{S}}_m$, in order to avoid confusion with the standard RPI sets, reported as $\mathbb{S}_m$.

Definition 4.2. Consider an autonomous switching linear system

$$x(t+1) = A_{\sigma(t)}x(t) + w(t)$$

with $w(t) \in \mathbb{W}_{\sigma(t)}$, subject to constraints $x(t) \in \mathbb{X}_{\sigma(t)}$ and with $\sigma(t)$ a CSS taking values in the finite set $\mathcal{M}$ and represented by a directed graph $\mathcal{G}(\mathcal{M}, \mathcal{E})$. A collection of sets $\{\bar{\mathbb{S}}_m\}_{m\in\mathcal{M}}$ is called an invariant multi-set if

$$A_l\bar{\mathbb{S}}_m \oplus \mathbb{W}_l \subset \bar{\mathbb{S}}_l$$

for all $(m, l) \in \mathcal{E}$.

Suppose again that independent TMPC controllers are designed for each mode. Given the switching control law (4.5), as long as the different TMPC controllers are feasible at the switching instances, it follows that

$$\begin{aligned}
x(t_k+1) &= A_{\sigma(t_k)}x(t_k) + B_{\sigma(t_k)}\kappa_{\sigma(_kt)}(x(t_k)) + w(t_k) \\
\bar{x}(t_k+1) &= A_{\sigma(t_k)}\bar{x}(t_k) + B_{\sigma(t_k)}\bar{\kappa}_{\sigma(t_k)}(\bar{x}(t_k)) \\
\implies e(t_k+1) &= \left(A_{\sigma(t_k)} + B_{\sigma(t_k)}K_{\sigma(t_k)}\right)e(t_k) + w(t_k) \\
&= \bar{A}_{\sigma(t_k)}e(t_k) + w(t_k) \\
\implies e(t_k+1) &\in \bar{A}_{\sigma(t_k)}\mathbb{S}_{\sigma(t_{k-1})} \oplus \mathbb{W}_{\sigma(t_k)} \subseteq \mathbb{S}_{\sigma(t_k)}.
\end{aligned} \tag{4.32}$$

By definition then, the elements of an invariant multi-set provide a solution to the state constraint problem depicted in (4.31). Furthermore, since $(m, m) \in \mathcal{E}$ for all $m \in \mathcal{M}$, every element $\bar{\mathbb{S}}_m$ of an invariant multi-set computed for the switching error dynamics (4.32) is also an RPI set for the m error mode dynamics. Thereby if $\bar{\mathbb{S}}_m$ is used to tighten the m mode constraints, the constraint satisfaction guarantees provided by the independent m-TMPC controller during the period mode m is active remain valid. In view of this is that, in order to achieve feasible switching between TMPC controllers that do not optimize nominal trajectories, an invariant multi-set will now be employed for constraint tightening.

There are two drawbacks with using invariant multi-sets. The first one has to do with the design process which will now be coupled among all modes. Indeed, even if the graph that defines the CSS is not strongly connected, if each mode is allowed to switch into at least one other, the computation of all the elements of $\{\bar{\mathbb{S}}_m\}_{m\in\mathcal{M}}$ is coupled. An exception exists if one mode, say $\bar{m}$, has no destination nodes. In this case $\bar{\mathbb{S}}_{\bar{m}}$ can be computed independently as an RPI set for the $\bar{m}$ dynamics, yet the elements of the invariant multi-set associated to nodes that are source to $\bar{m}$ still depend on $\bar{\mathbb{S}}_{\bar{m}}$.

The second drawback has to do with the constraint tightening procedure that is key in tube MPC implementations. Generally the RPI sets used in tube MPC are designed to be small. This is done to bring the constraint tightening to a minimum and allow more freedom to the optimisation-driven part of the composite control law (the nominal input). If the RPI set is large, most of the control authority is seized by the linear control law associated to the RPI set, reducing the overall control capabilities. In [136] a procedure to compute the minimal invariant multi-set is devised. This is similar in complexity to the computation of the minimal RPI set for a linear system and approximations equivalent to those devised in [100] are also provided. Nevertheless, each element of an invariant multi-set is RPI for its corresponding dynamics, yet not every RPI set is part of an invariant multi-set, hence it is expected that $\mathbb{S}_m \subseteq \bar{\mathbb{S}}_m$, leading to a larger tightening.

4.5.2 MDTs for admissible switching

Suppose now that TMPC controllers are designed for each mode, however not in an independent fashion but using the corresponding elements of an invariant multi-set computed for the error dynamics as their tightening set. Consider again that mode m became feasible active at time t_{k-1}. Proposition 4.2 guarantees the existence of a finite time $\bar{t} > t_{k-1}$ such that $\bar{x}(t_{k-1}) \in \mathbb{B}_{r_m(\bar{t}-t_{k-1}-1)} \subseteq \bar{\mathcal{X}}_{N_l}$ where l is any neighbouring mode of m. It follows that $\mathbb{P}_{N_l}(\bar{x}(\bar{t}))$ is feasible and can be solved, and hence a switch could tentatively take place. The multi-set invariance of the collection of sets $\{\bar{\mathbb{S}}_m\}_{m\in\mathcal{M}}$ takes care of the possible state constraint violation discussed in (4.31), however the error switching dynamics only take the form in (4.31) if $u(\bar{t}) = \kappa_l(x(\bar{t})) \in \mathbb{U}_l$ which is not necessarily the case according to (4.30).

To guarantee that the input constraint associated to a destination mode is met when the switch takes place, an auxiliary controller is now proposed. In principle, the latter can be designed following any technique, but tube-based MPC is chosen for consistency. The objective of this auxiliary, or transition, controller is to guarantee input constraint satisfaction during the transition step, after which the destination TMPC controller becomes feasible due to the multi-set invariance. It follows then that the transition controller is only used during one time step, the switching instance. Given the heterogeneity of the modes, an auxiliary controller is required for each allowable switch, hence for any pair $(m, l) \in \mathcal{E}$ an m, l-transition tube-based MPC controller is defined. Consider the following optimization problem

$$\tilde{\mathbb{P}}^m_{N_l}(\bar{x}(t))\colon \qquad \min_{\bar{\boldsymbol{u}}} J_{N_l}(\bar{\boldsymbol{u}}, \bar{x}_0) \tag{4.33a}$$

s.t. (for $k = 0, \dots, N_l - 1$)

$$\bar{x}_0 = \bar{x}(t) \tag{4.33b}$$

$$\bar{u}_0 \in \tilde{\mathbb{U}}^m_l \subseteq \mathbb{U}_l \ominus K_l \bar{\mathbb{S}}_m \tag{4.33c}$$

$$\bar{x}_{k+1} = A_l \bar{x}_k + B_l \bar{u}_k \tag{4.33d}$$

$$\bar{x}_k \in \bar{\mathbb{X}}_l \subseteq \mathbb{X}_l \ominus \bar{\mathbb{S}}_l \tag{4.33e}$$

$$\bar{u}_k \in \bar{\mathbb{U}}_l \subseteq \mathbb{U}_l \ominus K_l \bar{\mathbb{S}}_l \tag{4.33f}$$

$$\bar{x}_{N_l} \in \bar{\mathbb{X}}_{f,l} \subseteq \bar{\mathbb{X}}_l, \tag{4.33g}$$

with

$$\bar{\boldsymbol{u}}^\circ(\bar{x}(t)) = \arg \tilde{\mathbb{P}}^m_{N_l}(\bar{x}(t))$$
$$V^m_{N_l}(\bar{x}(t)) = J_{N_l}\big(\bar{\boldsymbol{u}}^\circ(x(t)), \bar{x}(t)\big),$$

and define $\tilde{\mathcal{X}}^m_{N_l}$ as the set of all the states for which $\tilde{\mathbb{P}}^m_{N_l}(x)$ is feasible. Finally, the nominal and composite transition control laws are correspondingly

$$\tilde{\kappa}_{m,l}(\bar{x}(t)) = \bar{u}^\circ_0(\bar{x}(t)) \tag{4.34a}$$

$$\kappa_{m,l}(\bar{x}(t)) = \tilde{\kappa}_{m,l}(\bar{x}(t)) + K_l\left(x(t) - \bar{x}(t)\right). \tag{4.34b}$$

It follows that the optimization problem associated to the transition controller $\tilde{\mathbb{P}}^m_{N_l}(\cdot)$ and the one associated to the destination mode controller $\mathbb{P}_{N_l}(\cdot)$ differ solely in that the former has an additional input constraint (4.33c) over the first element of the optimized input sequence.

Following Proposition 4.2 and Theorem 4.1 it is easy to compute the minimum MDT, say $\tilde{\tau}^f_{m,l}$, that guarantees $\bar{x}(t) \in \tilde{\mathcal{X}}^m_{N_l}$ for all $t \geq t_{k-1} + \tilde{\tau}^f_{m,l} + 1$ given $\bar{x}(t_{k-1}) \in \bar{\mathcal{X}}_{N_m}$. The following result then holds for all $k \in \mathbb{N}_0$.

Proposition 4.8. Consider any pair $m, l \in \mathcal{M}$ with $m \neq l$ and $l \in \mathcal{M}_m$. For any two switching instances $(t_{k-1}, \sigma(t_{k-1}) = m)$ and $(t_k, \sigma(t_k) = l)$ that fulfil the feasibility MDT $\tilde{\tau}^f_{m,l}$ associated to the m,l-transition controller, if the control law (4.34) is used to define the nominal and true inputs at time t_k, then (1) $u(t_k) \in \mathbb{U}_l$, (2) $\mathbb{P}_{N_l}(\bar{x}(t_k+1))$ is feasible and (3) $x(t_k+1) \in \mathbb{X}_l$.

Proof. For (1) first note that if mode m became feasible active at time t_{k-1} it follows that $e(t_{k-1}) \in \mathbb{S}_m$ and $\bar{x}(t_{k-1}) \in \bar{\mathcal{X}}_{N_m}$. Furthermore if the feasibility MDT associated to the m,l-transition controller is met, it holds that $\bar{x}(t_k) \in \tilde{\mathcal{X}}^m_{N_l}$, hence $\tilde{\mathbb{P}}^m_{N_l}(\bar{x}(t_k))$ is feasible and solving it yields

$$\bar{u}(t_k) = \tilde{\kappa}_{m,l}(\bar{x}(t_k)) = \bar{u}^\circ_0(\bar{x}(t)) \in \tilde{\mathbb{U}}^m_l$$

which then implies

$$u(t_k) = \kappa_{m,l}(\bar{x}(t_k)) = \bar{u}^\circ_0(\bar{x}(t)) + K_l e(t_k) \in \tilde{\mathbb{U}}^m_l \oplus K_l \bar{\mathbb{S}}_m \subseteq \mathbb{U}_l$$

where the inclusion holds by the definition of $\tilde{\mathbb{U}}^m_l$ in (4.33c).

For (2) note that the nominal state is not subject to disturbances thus the nominal control law (4.34) results in that $\bar{x}(t_k+1)$ matches the prediction made at the previous time instant $\bar{x}_1(\bar{x}(t))$. It follows that $\bar{x}(t_k+1) \in \bar{\mathbb{X}}_l$, but more importantly, that there exists a sequence of $N_l - 1$ control actions, namely $\left\{\bar{u}^\circ_1(\bar{x}(t)), \ldots, \bar{u}^\circ_{N_l-1}(\bar{x}(t))\right\}$, that drive the nominal state to the terminal region while satisfying the l-mode constraints. The latter implies $\bar{x}(t_k+1) \in \bar{\mathcal{X}}_{N_l-1} \subseteq \bar{\mathcal{X}}_{N_l}$ which guarantees feasibility of $\mathbb{P}_{N_l}(\bar{x}(t_k+1))$.

For (3) define $\bar{A}_l = (A_l + B_l K_l)$ and note that

$$\begin{aligned} x(t_k+1) &= A_l x(t_k) + B_l u(t_k) + w(t_k) \\ \bar{x}(t_k+1) &= A_l \bar{x}(t_k) + B_l \bar{u}(t_k) \end{aligned}$$

$$\implies e(t_k+1) = (A_l + B_l K_l)\, e(t_k) + w(t_k) = \bar{A}_l e(t_k) + w(t_k)$$
$$\implies e(t_k+1) \in \bar{A}_l \bar{\mathbb{S}}_m \oplus \mathbb{W}_l \subseteq \bar{\mathbb{S}}_l$$
$$\implies x(t_k+1) = e(t_k+1) + \bar{x}(t_k+1) \in \bar{\mathbb{S}}_l \oplus \bar{\mathcal{X}}_{N_l} \subseteq \bar{\mathbb{S}}_l \oplus \bar{\mathbb{X}}_l \subseteq \mathbb{X}_l,$$

where the first implication follows from the definition of the error dynamics, the second one from the definition of the tightening sets as elements of an invariant multi-set and the last one from the feasibility of $\mathbb{P}_{N_l}(\bar{x}(t_k+1))$. ■

In view of Proposition 4.8, it is easy to establish minimum MDTs that guarantee admissible switching between independently designed TMPC controllers that do not optimize trajectories, provided transition controllers have been defined.

Theorem 4.7. Assume $\sigma(\cdot)$ is a CSS. If for all $m \in \mathcal{M}$ the MDTs are set to $\tilde{\tau}_m^f$ defined by

$$\tilde{\tau}_m^f = 1 + \max_{l \in \mathcal{M}_m} \tilde{\tau}_{m,l}^f,$$

then for all $k \in \mathbb{N}_0$ the switching control law

$$\kappa(x(t)) = \begin{cases} \kappa_{\sigma(t)}(x(t)) & t = 0 \\ \kappa_{\sigma(t)}(x(t)) & t \in (t_{k-1}, t_k) \\ \tilde{\kappa}_{\sigma(t_{k-1}),\sigma(t_k)}(x(t)) & t = t_k, \end{cases} \tag{4.35}$$

guarantees constraint satisfaction for the switching linear system (4.1).

Proof. Follows directly from Proposition 4.8. ■

There are two important things to remark. First note that such an approach is only applicable because the switching sequence is assumed to be instantly known, hence at a switching time instance t_k the mode that became active is available and the appropriate transition controller can be employed. Secondly, note that the minimum MDTs computed in Section 4.4.1 guarantee $x(t_k) \in \mathcal{X}_{N_l} \subseteq \mathbb{X}_l$. That is the state constraint of the destination mode is met at the switching instant t_k. The approach depicted in this section, however, only guarantees $x(t_k) \in \bar{\mathbb{S}}_m \oplus \tilde{\mathcal{X}}_{N_l}^m \subseteq \bar{\mathbb{S}}_m \oplus \bar{\mathbb{X}}_l$, which is not necessarily a subset of the true state constraint set $\mathbb{X}_l$. This is not necessarily seen as a drawback for two reasons. First, $x(t_k)$ is a result of the m-TMPC controller, thus demanding $x(t_k) \in \mathbb{X}_l$ for all $l \in \mathcal{M}_l$ might be unnecessarily conservative for an a-priori unknown switching sequence. Secondly, $x(t_k+1) \in \mathbb{X}_l$ is guaranteed. Furthermore, $x(t_k) \in \mathbb{X}_l$ can be easily enforced by extending the feasibility MDTs $\tilde{\tau}^f_{m,l}$ so that it not only guarantees $\mathbb{B}_{r_m(\tilde{\tau}^f_{m,l})} \subseteq \tilde{\mathcal{X}}^m_{N_l}$ but also $\bar{\mathbb{S}}_m \oplus \mathbb{B}_{r_m(\tilde{\tau}^f_{m,l})} \subseteq \mathbb{X}_l$, which is guaranteed to happen at a finite $\tilde{\tau}^f_{m,l}$ given Assumption 4.1. Finally, note that if the RoA of the different controllers are intractable to compute, the approach devised in Section 4.3.1.2 is readily applicable for this case.

4.5.3 MDTs for stable switching

The primary objective of employing the TMPC variant that does not optimize trajectories is to achieve an stability result similar to that developed for the disturbance-free case. The latter hinges on comparing value functions of the neighbouring controllers at the time of a switch. In this case, however, three controllers come into play at every switching instance.

There are certain cases in which the transition controller can be replaced by the destination controller, and hence the stability analysis depicted in Section 4.3.2 can be applied without changes for the nominal trajectories. Indeed, for any pair $(m, l) \in \mathcal{E}$ such that $\bar{\mathbb{U}}_l \subseteq \tilde{\mathbb{U}}^m_l$, it holds that

$$\begin{aligned} & \bar{\mathbb{U}}_l \oplus K_l \mathbb{S}_m \subseteq \tilde{\mathbb{U}}^m_l \oplus K_l \mathbb{S}_m \\ \Longrightarrow\ & \bar{\mathbb{U}}_l \oplus K_l \mathbb{S}_m \subseteq (\mathbb{U}_l \ominus K_l \mathbb{S}_m) \oplus K_l \mathbb{S}_m \\ \Longrightarrow\ & \bar{\mathbb{U}}_l \oplus K_l \mathbb{S}_m \subseteq \mathbb{U}_l \end{aligned}$$

where the first implication follows from (4.33c) and the second one from the Pontryagin difference properties. It holds then that by solving $\mathbb{P}_{N_l}(\bar{x}(t_k))$ the input constraints are met and the invariant multi-set guarantees the state constraint satisfaction

Nevertheless, verifying such inclusion for all pairs of neighbouring modes is not trivial and requires a design process considerably more complex than simply resorting to invariant multi-sets. However, it is still possible to employ the approach proposed in Section 4.3.2 to guarantee exponential stability of the origin for the nominal trajectories of TMPC controllers. In order to demonstrate so, consider the following result.

Proposition 4.9. For all $\bar{x}(t) \in \tilde{\mathcal{X}}^m_{N_l}$ it holds that

$$V_{N_l}(\bar{x}(t+1)) - V^m_{N_l}(\bar{x}(t)) \leq -f_l |x(t)|^2_2,$$

with $f_l > 0$ from (4.3b).

Proof. The proof follows from the concept "feasibility implies stability" usually employed in establishing stability of MPC controllers. First note that if $\bar{x}(t) \in \tilde{\mathcal{X}}^m_{N_l}$, then $\tilde{\mathbb{P}}^m_{N_l}$ is feasible with solution

$$\bar{\boldsymbol{u}}^\circ(\bar{x}(t)) = \left\{\bar{u}^\circ_0(\bar{x}(t)), \dots, \bar{u}^\circ_{N_l-1}(\bar{x}(t))\right\}$$

with $\bar{u}^\circ_j(\bar{x}(t)) \in \bar{\mathbb{U}}_l$ for all $j \in [0, N_l - 1]$, and associated predicted nominal state trajectory

$$\bar{\boldsymbol{x}}^\circ(\bar{x}(t)) = \left\{\bar{x}^\circ_0(\bar{x}(t)) = \bar{x}(t), \dots, \bar{x}^\circ_{N_l}(\bar{x}(t))\right\}$$

with $\bar{x}^\circ_j(\bar{x}(t)) \in \bar{\mathbb{X}}_l$ for all $j \in [0, N_l - 1]$ and $\bar{x}^\circ_{N_l}(\bar{x}(t)) \in \bar{\mathbb{X}}_{f,l}$. Since the nominal system is not subject to disturbances it follows that $\bar{x}(t+1) = \bar{x}^\circ_1(\bar{x}(t))$ and so

$$\bar{\boldsymbol{u}}^\triangle = \left\{\bar{u}^\circ_1(\bar{x}(t)), \dots, \bar{u}^\circ_{N_l-1}(\bar{x}(t)), K_l\bar{x}^\circ_{N_l}(\bar{x}(t))\right\}$$

is a feasible solution to $\mathbb{P}_{N_l}(\bar{x}(t+1))$. Since the cost function of both optimization problems is the same, it follows that

$$V_{N_l}(\bar{x}(t+1)) - V^m_{N_l}(\bar{x}(t)) \leq -\left(||\bar{x}^\circ_0(\bar{x}(t))||^2_{Q_m} + ||\bar{u}^\circ_0(\bar{x}(t))||^2_{R_m}\right)$$

which is the same bound found for $V_{N_l}(\bar{x}(t+1)) - V_{N_l}(\bar{x}(t))$ during the proof of Proposition 4.1 (see [1]). ■

It follows from Proposition 4.9 that, given the proposed auxiliary controller, the decrease rate of the corresponding value functions when leaping from the m, l-transition controller to the l-TMPC is the same as if the latter had been active at the switching instant t_k. It follows then that the exponential stability result available for the independent TMPC controllers is valid for the entirety of the interval $[t_k, t_{k+1})$ and so the same arguments employed in Section 4.3.2 also guarantee exponential stability of the origin for the switching closed-loop trajectories given the individual TMPCs considered in this section and the control law (4.35). A direct consequence of this is that the set $\bar{\mathcal{O}}_s$, defined analogously to (4.21) as

$$\bar{\mathcal{O}}_s = \bigcup_{m \in \mathcal{M}} \bar{\mathbb{S}}_m, \tag{4.36}$$

is exponentially stable for the true switching closed-loop, a result that follows from Corollary 2.2 and Proposition 2.2.

Table 4.1: Summary of proposed MDTs

Dwell-time		Feasible switching		Stable Switching
Disturbance	Design	Known RoA	Unknown RoA	
No	I	$\left(\tau_m^f\right)$ § 4.3.1.1	$\left(\bar{\tau}_m^f\right)$ § 4.3.1.2	$\left(\tau_m^s\right)$ § 4.3.2
Yes	I	$\left(\tau_m^f\right)$ § 4.4.1	$\left(\bar{\tau}_m^f\right)$ § 4.3.1.2	$\left(\tau_m^g, \tau_{\bar{m}}^s\right)$ § 4.4.2
Yes	C	$\left(\tilde{\tau}_m^f\right)$ § 4.5.2	$\left(\bar{\tilde{\tau}}_m^f\right)$ § 4.3.1.2	$\left(\tau_m^s\right)$ § 4.5.3

4.6 Summary of MDTs

Throughout this chapter several approaches to computing MDTs have been proposed. The common objective in all of them is to compute minimum MDTs such that the control law (4.5) (or (4.35)) is able to stabilize the switching system (4.1) while respecting state and input constraints independent of external perturbations. The applicability of each MDT depends on whether disturbances are considered, the feasibility region of the independent controllers is known, and/or a coupled designed of the different tube-based controllers is allowed. Table 4.1 summarizes all the case for which MDTs were proposed (I stands for independent and C for coupled).

4.7 Illustrative example

The capabilities of the proposed approach to compute MDTs are now demonstrated via a numerical example. A system of order two is considered in order to be able to compute the corresponding feasibility regions $\bar{\mathcal{X}}_{N_m}$, and have access to all the results discussed in this chapter. The analysed system has $M = 5$ different modes and an associated CSS with constraints as shown by the graph in Fig. 4.2. The dynamics of each mode are reported in Table 4.2, and the constraints sets in Table 4.3, where the linear map T is defined as $T = [1.5\ 0;\ 0\ 1]$. Although of low order, the proposed example incorporates a high degree of heterogeneity, with no two modes being defined by the same dynamics, disturbances and/or constraint sets.

The results presented in this chapter do not rely on the heterogeneity of the modes, nor of the associated cost functions. Nevertheless, for simplicity of exposition the cost matrices are set to $Q_1 = 10\mathcal{I}_2$, $Q_{2:5} = \mathcal{I}_2$, $R_{1:5} = 1$, and the MPC horizons to $N_{1:5} = 5$. The local tube and terminal gains K_m are set to the corresponding LQR gains, also reported in Table 4.2. Furthermore, recall that the multi-sets are reported as $\bar{\mathbb{S}}_m$, in order to avoid confusion with the

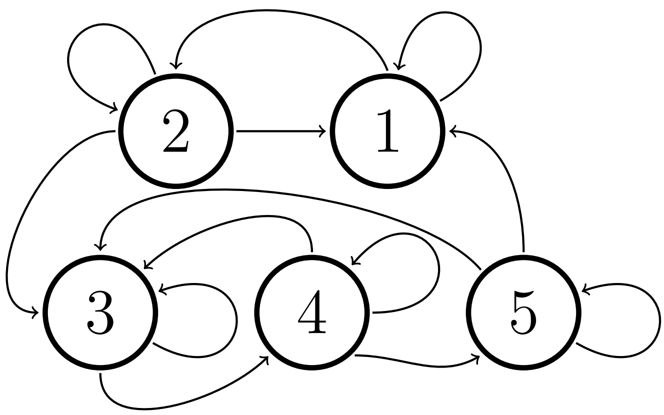

Figure 4.2: Graph representing the switching constraints of $\sigma(\cdot)$ for the illustrative example.

Table 4.2: Dynamics for all modes of the switching system.

Mode	1	2	3	4	5
A_m	$\begin{bmatrix}1.5 & 0\\1.5 & 1\end{bmatrix}$	$\begin{bmatrix}1 & 1.5\\0 & 1.5\end{bmatrix}$	$\begin{bmatrix}0.7 & 0.1\\0.2 & 0.4\end{bmatrix}$	$\begin{bmatrix}0.8 & 0.3\\0.4 & 0.1\end{bmatrix}$	$\begin{bmatrix}0.2 & 0.1\\0.2 & 0.6\end{bmatrix}$
B_m	$\begin{bmatrix}1\\0.8\end{bmatrix}$	$\begin{bmatrix}1\\0.8\end{bmatrix}$	$\begin{bmatrix}1\\0.5\end{bmatrix}$	$\begin{bmatrix}0.7\\0.8\end{bmatrix}$	$\begin{bmatrix}1.3\\0.6\end{bmatrix}$
$-K_m^\top$	$\begin{bmatrix}1.6219\\0.5669\end{bmatrix}$	$\begin{bmatrix}0.4100\\1.4061\end{bmatrix}$	$\begin{bmatrix}0.3959\\0.1446\end{bmatrix}$	$\begin{bmatrix}0.5253\\0.1782\end{bmatrix}$	$\begin{bmatrix}0.1387\\0.2088\end{bmatrix}$

standard RPI sets, reported as $\mathbb{S}_m$.

4.7.1 Set related results

4.7.1.1 Nominal MPC

Figure 4.3 shows, for mode 1, the true RoA $\bar{\mathcal{X}}_{N_m}$ and the auxiliary set Θ_m used for feasibility purposes in case the former is not available. The latter is computed following the discussion in Section 4.3.1.2, but without adding redundant vertices in the description of $\mathbb{X}_m$. The sets for the other modes are not reported because $\bar{\mathcal{X}}_{N_m} = \mathbb{X}_m$ for $m = 2, \ldots, 5$, hence the approach discussed in Section 4.3.1.2 yields $\Theta_m = \mathbb{X}_m = \bar{\mathcal{X}}_{N_m}$. For mode 1, however, the auxiliary set is such that $\Theta_1 \subset \bar{\mathcal{X}}_{N_1}$, resulting in a smaller overall RoA for the switching controller when the feasibility regions cannot be exactly computed.

Table 4.3: Constraints for all modes of the switching system.

Mode	1	2	3	4	5
$\mathbb{X}_m$	$\{\lvert x\rvert_\infty \leq 2\}$	$^1\!/_2\mathbb{X}_1$	$\mathbb{X}_1$	$T\mathbb{X}_1$	$T\mathbb{X}_1$
$\mathbb{U}_m$	$^3\!/_2\mathbb{U}_2$	$\{\lvert u\rvert_\infty \leq 2\}$	$2\mathbb{U}_2$	$^3\!/_2\mathbb{U}_2$	$^3\!/_8\mathbb{U}_2$
$\mathbb{W}_m$	$^1\!/_{10}\mathbb{W}_3$	$^1\!/_{10}\mathbb{W}_3$	$\{\lvert w\rvert_\infty \leq 1\}$	$^1\!/_2\mathbb{W}_3$	$^7\!/_{10}\mathbb{W}_3$

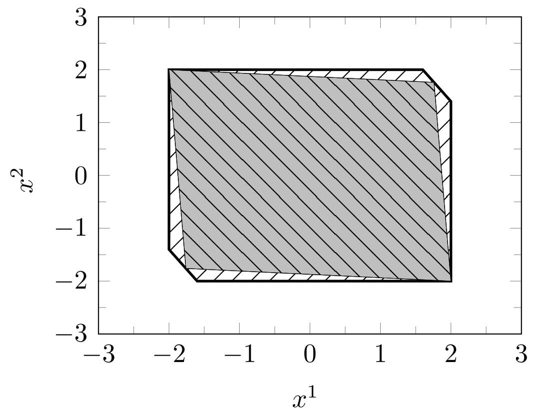

Figure 4.3: Feasibility regions used to compute the feasibility MDT for mode 1 in the undisturbed case: ▨ $\bar{\mathcal{X}}_{N_1}$ and ▧ Θ_1.

4.7.1.2 Tube MPC: independent design

The sets Ω_m, required to characterize the robustly stable set for the switching closed-loop dynamics, are computed following the discussion in Section 4.4.2. In this particular case, said sets are defined by $\Omega_m = \varepsilon_m \bar{\mathbb{X}}_{f,m}$ with the values of ε_m reported in Table 4.4. It follows that $\Omega_m = \{\mathbf{0}\}$ for $m = 2, 3$, but has a non-zero volume for modes 1, 4 and 5.

Figure 4.4 compares the size of the tightening set $\mathbb{S}_m$, which is robustly stable for each mode in a non-switching scenario, to that of the augmented set $\mathbb{S}_m \oplus \Omega_m$ required to guarantee stability given the switching sequence. The largest difference is observed for mode 1, given that a the CSS allows a switch from mode 5 to 1 and so $\mathbb{S}_5 \subset \mathbb{S}_1 \oplus \Omega_1$ is required, but $\mathbb{S}_1$ is much smaller than $\mathbb{S}_5$. Figure 4.5 compares the exponentially stable region of the switching system $\mathcal{O}_g$ (see (4.26)) with the union of robustly stable sets in a non-switching scenario $\mathcal{O}_s$ (see (4.21)). For this particular example $\mathbb{S}_3 \supseteq \mathbb{S}_m$ and $\mathbb{S}_4 \oplus \Omega_4 \supseteq \mathbb{S}_m \oplus \Omega_m$ for all $m \in \mathcal{M}$, so $\mathcal{O}_s = \mathbb{S}_3$ and $\mathcal{O}_g = \mathbb{S}_4 \oplus \Omega_4$.

As expected, given that $\Omega_m \neq \{\mathbf{0}\}$ for some modes, $\mathcal{O}_g$ is larger than $\mathcal{O}_s$ (by a 55% in volume). This represents the trade-off between complexity and control performance when designing controllers for switching systems. Indeed,

$\mathcal{O}_g$ is considerably larger than $\mathcal{O}_s$, however it is guaranteed to be robustly stable for the switching system when in closed-loop with simple, off-the-shelf, tube MPC controllers designed for each mode.

Table 4.4: Value of the scaling factor ε_m used to compute $\Omega_m = \varepsilon_m \bar{\mathbb{X}}_{f,m}$.

Mode	1	2	3	4	5
ε_m	0.69	0	0	0.66	0.11

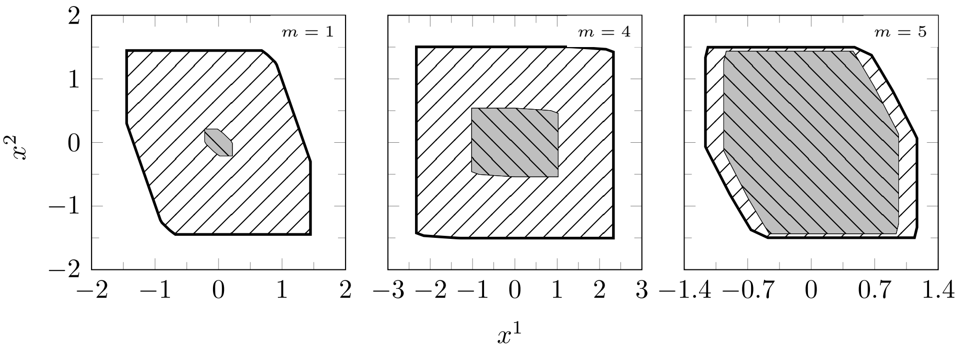

Figure 4.4: Convergence regions for tube MPC with independent design: ▨ $\mathbb{S}_m \oplus \Omega_m$ and ▧ $\mathbb{S}_m$.

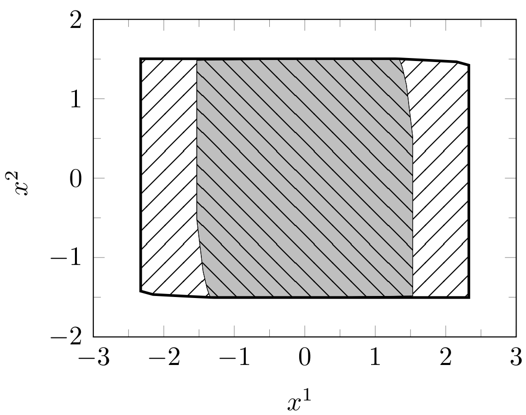

Figure 4.5: Convergence regions for tube MPC with independent design: ▨ $\mathcal{O}_g$ and ▧ $\mathcal{O}_s$.

4.7.1.3 Tube MPC: coupled design

Figure 4.6 compares the minimal RPI set $\mathbb{S}_m$ and the minimal multi-set $\bar{\mathbb{S}}_m$ for all modes except 3, since for the latter both sets are equal. As expected, the minimal multi-set, being more demanding in its properties, can be larger that the minimal RPI set (Figure 4.6). In turn, this has an adverse effect in the size of the independent feasibility regions of each controller, resulting in smaller RoAs for modes 1,2 and 4 (see Figure 4.7).

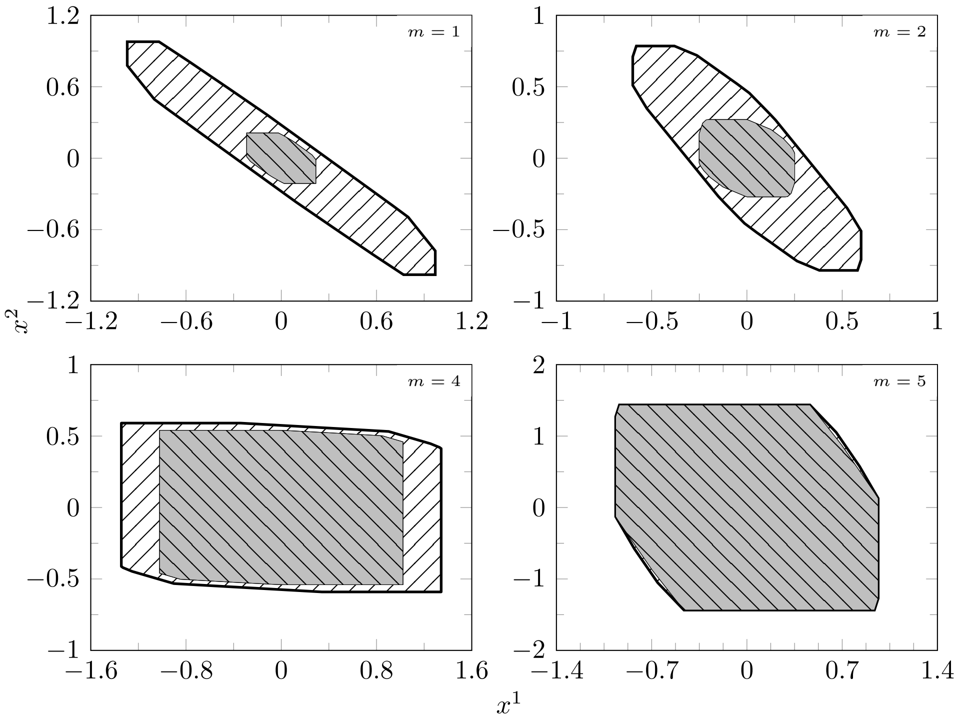

Figure 4.6: Convergence regions for tube MPC with coupled and independent design: minimal RPI set $\mathbb{S}_m$ and minimal multi-set $\bar{\mathbb{S}}_m$.

The tightening sets, when computed as multi-sets, are at least as large as when computed as simple RPI sets for the local dynamics. However, this does not necessarily imply that the neighbourhood of the origin shown to be robustly stable is also larger than the one obtained with independently designed tube-based MPC controllers. Indeed, the latter can only guarantee robust stability of $\mathcal{O}_g$ defined as in (4.26), however the multi-set approach guarantees robust stability of $\bar{\mathcal{O}}_s$ defined by (4.36). For this particular example, it holds that $\bar{\mathbb{S}}_3 \supseteq \bar{\mathbb{S}}_m$ for all $m \in \mathcal{M}$ and so $\bar{\mathcal{O}}_s = \bar{\mathbb{S}}_3$. Furthermore, it also holds that $\bar{\mathbb{S}}_3 = \mathbb{S}_3$, hence $\bar{\mathcal{O}}_s = \mathcal{O}_s$, and so robust stability of the collection of independently computed RPI sets is achieved. This is, however, not the norm, and the relation between $\mathcal{O}_s$, $\mathcal{O}_g$ and $\bar{\mathcal{O}}_s$ is heavily dependent on the characteristics of the problem. In this particular example, $\bar{\mathcal{O}}_s = \mathcal{O}_s$ follows from the fact that mode 3 is subject to the largest disturbances, resulting in a large minimal RPI set $\mathbb{S}_3$.

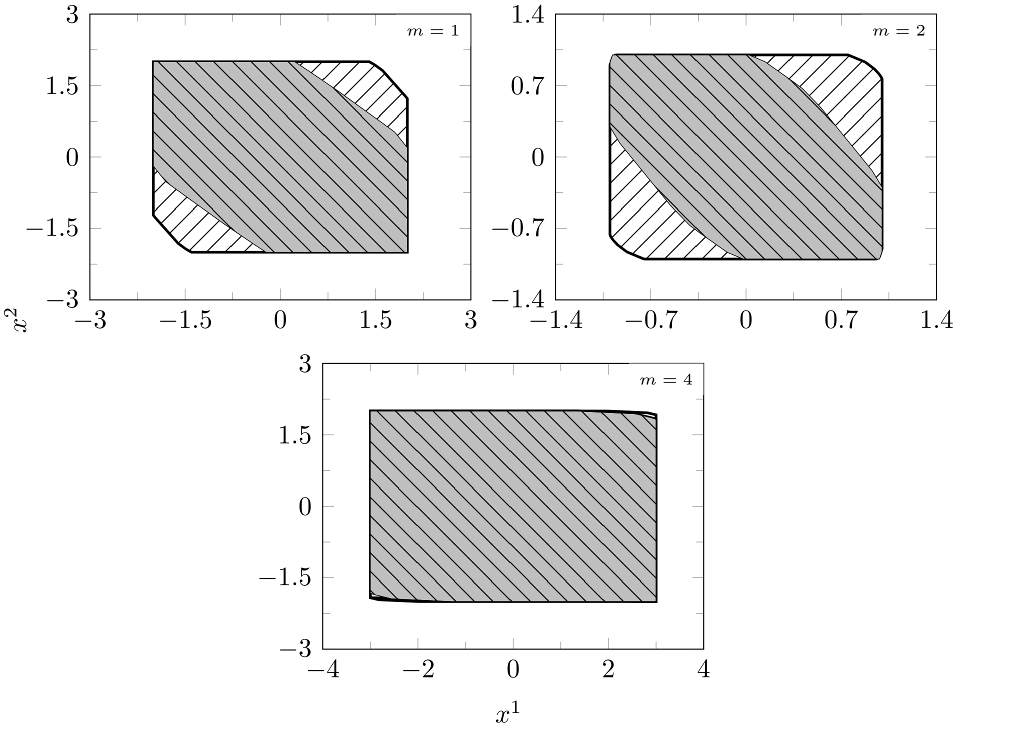

Figure 4.7: RoA of tube MPC with coupled and independent design $\mathcal{X}_{N_m}$: $\mathcal{X}_{N_m}(\mathbb{S}_m)$ and minimal multi-set $\mathcal{X}_{N_m}(\bar{\mathbb{S}}_m)$.

4.7.2 Minimum mode-dependent dwell times

To compute the MDTs that guarantee feasible and stable switching, it is first necessary to compute the exponential stability constant c_m and λ_m associated to each MPC controller. These two constant are, as discussed in Section 4.2.1, a function of the Lyapunov bounds b_m, d_m and f_m. It is shown in the proof of Theorem 3.3 that $b_m = f_m = \xi_m(Q_m)$ fulfils the required inequalities, however computing an appropriate d_m is not trivial. In [1, Section 2.4] it is shown that, if $\bar{\mathbb{X}}_{f,m}$ is compact, then there exists $d_m > \xi_M(P_m)$ such that the corresponding inequality holds, and moreover, a procedure is provided to compute it. Said approach relies on the size of the nominal feasibility regions $\bar{\mathcal{X}}_{N_m}$, and hence yields a different result depending on whether disturbances are considered, and on which type of tightening set is used. Table 4.5 show the values of the exponential stability constants for all three cases, when d_m is computed following the guidelines in [1, Section 2.4]. In all cases c_m is greater than 2.5 and $\lambda_m \approx 1$, resulting in slow convergence rates an hence a slow shrinkage rate of the set $\mathbb{B}_{r_m(\tau)}$ in (4.8) (or its corresponding parallels).

Table 4.6 contains the feasibility MDTs computed following the different

Table 4.5: Convergence constants computed according to [1].

	Nominal MPC		Tube MPC		Multi-set	
Mode	c_m	λ_m	c_m	λ_m	c_m	λ_m
1	4.0354	0.9688	4.0929	0.9697	3.3640	0.9548
2	3.3090	0.9532	3.3050	0.9531	3.2051	0.9501
3	2.5648	0.9209	2.5635	0.9208	2.5635	0.9208
4	2.6716	0.9273	2.6698	0.9272	2.6667	0.9270
5	4.3179	0.9728	7.6055	0.9913	7.6055	0.9913

Table 4.6: Feasibility MDTs (with (c_m, λ_m) computed according to [1]).

	Nominal MPC		Tube MPC		Multi-set	
Mode	τ_m^f	$\bar{\tau}_m^f$	τ_m^f	$\bar{\tau}_m^f$	$\tilde{\tau}_m^f$	$\bar{\tilde{\tau}}_m^f$
1	89	88	97	96	77	75
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	27	26	31	30	39	38
5	89	90	392	459	447	445

approaches devised in this chapter, and employing the exponential stability constants reported in Table 4.5. As expected, given the slow convergence rate guaranteed by the constants in Table 4.5, most of the minimum dwell-times required to guarantee feasibility are unnecessarily long. Consider, for example, the tube MPC case with known feasibility regions. If mode 5 became feasibly active at a certain time instant, it must remain active for at least 392 future time steps before a switch into modes 1 or 3 is guaranteed to be feasible.

The MDTs in Table 4.6, although feasibility inducing, are not practical. This, however, is not attributable to the particular approach proposed here to compute them, but to the conservativeness with which the upper bound for the Lyapunov functions d_m is computed. Indeed, the approach in [1] is mostly concerned with guaranteeing existence of such a bounding constant, rather than finding a tight one. In order to demonstrate the practicality of the approach proposed in this chapter to compute MDTs, a tighter bound needs to be obtained. The latter is not a trivial task, and is out of the scope of this chapter. In what follows, a tighter bound is estimated through Monte Carlo simulations. In order to obtain said estimate, the corresponding optimization problem (depending on the case) is solved for 1000 randomly selected, albeit feasible, values of the state. A less conservative upper bounding scalar d_m can then be obtained by comparing $V_{N_m}(x(t))$ and $d_m|\bar{x}_0^*(x)|_2^2$ at the randomly

Table 4.7: Convergence constants computed with numerically obtained d_m.

	Nominal MPC		Tube MPC		Multi-set	
Mode	c_m	λ_m	c_m	λ_m	c_m	λ_m
1	1.4094	0.7047	1.4092	0.7046	1.4087	0.7043
2	1.9011	0.8505	1.9019	0.8506	1.9013	0.8505
3	1.1402	0.4805	1.1408	0.4813	1.1408	0.4813
4	1.2873	0.6298	1.2879	0.6302	1.2879	0.6302
5	1.1863	0.5380	1.1863	0.5380	1.1863	0.5380

Table 4.8: Feasibility MDTs (with (c_m, λ_m) computed numerically).

	Nominal MPC		Tube MPC		Multi-set	
Mode	τ_m^f	$\bar{\tau}_m^f$	τ_m^f	$\bar{\tau}_m^f$	$\tilde{\tau}_m^f$	$\bar{\tilde{\tau}}_m^f$
1	6	5	7	6	9	8
2	1	1	1	1	1	1
3	1	1	1	1	1	1
4	4	3	5	4	6	5
5	3	2	5	4	5	4

selected point.

Table 4.7 shows the convergence constants resulting from these numerically obtained bounds and Table 4.8 presents the feasibility MDTs that result from using these tighter bounds. It can be observed that c_m is closer to 1 and $\lambda_m \ll 1$ in all cases, which result in a faster convergence rate for the radius of the set $\mathbb{B}_{r_m(\tau)}$. This, in turn, yields considerably shorter feasibility inducing MDTs. Consider the same example than before; a switch out of mode 5 into modes 1 or 3 is guaranteed to be feasible only after 5 time steps, which represents less than 2% of the MDT computed with the analytically obtained bounds. In view of these results it is possible to conclude that the approach proposed in this chapter can produce practical feasibility inducing dwell-times, given tight bounds on the optimal value function.

Finally note that, although $\Theta_m = \bar{\mathcal{X}}_{N_m} = \mathbb{X}_m$ for all $m = 2, \ldots, 5$, the feasibility MDTs do not fulfil $\tau_m^f = \bar{\tau}_m^f$ for $m = 4, 5$. This is particular to this example in which $\bar{\mathcal{X}}_{N_m} = \bar{\mathcal{X}}_{N_m - 1}$ for all $m \in \mathcal{M}$.

Finally, Table 4.9 presents the stability MDTs obtained using the numerically estimated bounds. Recall that in the TMPC case the stability guarantee relies on feasibility (Theorem 4.6), therefore the MDTs are generally larger when compared to the Multi-set case. The value τ_m^s represents the time it takes to reach the extended neighbourhood Ω_m from $\bar{\mathcal{X}}_{N_m}$, and the value $\bar{\tau}_m^s$ is the

Table 4.9: Stability MDTs (with (c_m, λ_m) computed numerically).

	MPC	TMPC		Multi-set
Mode	τ_m^s	τ_m^g	$\tau_m^s(\bar{\tau}_m^s)$	τ_m^s
1	1	15	19(18)	1
2	14	1	95(94)	13
3	2	1	21(20)	2
4	3	16	19(18)	4
5	4	6	12(11)	4

analogue for when $\bar{\mathcal{X}}_{N_m}$ is unknown and replaced by Θ_m. In the multi-set or nominal MPC cases, on the other hand, the stability MDT depends on the behaviour of neighbouring value functions. In these two cases the stability MDT of mode 2 is larger than for other modes. This can be explained by the cost functions; indeed, mode 2 is allowed to switch into mode 1 (see Fig. 4.2) however $Q_1 = 10Q_2$, therefore a longer time is needed in mode 2 to guarantee a cost decrease before switching to mode 1.

4.8 Summary

This chapter examined the current algorithms for switching MPC and proposed a novel approach to guarantee feasible and stable switching based on the computation of minimum dwell-times. Several avenues where explored, including nominal and disturbed dynamic and independent or coupled designed of the MPC controllers for each mode. The technique proposed to compute the minimum dwell-times relies on the exponential stability result available for MPC when the cost is quadratic, the constraints are linear, and the optimization problem is solved to optimality at each time instant. The latter is, evidently, a drawback since suboptimal MPC (due to early termination of the optimization algorithm) can only guarantee asymptotic stability. Nevertheless, the proposed set-up results in the optimization being a convex QP problem, which can be efficiently solved to an arbitrary accuracy.

An illustrative example was put forward, consisting of a switching system with highly heterogeneous modes (both in dynamics and constraints). The results showed that the quality of the obtained minimum dwell-times is highly dependent on the tightness with which the Lyapunov bounds are computed. Nevertheless, the proposed approach requires only off-the-shelf MPC controllers, being the computation of multi-sets the most complex design step. This is opposed to other approaches such as [114,122,124,130,133], which require several different design steps such as the implementation of multi-parametric programming tools to characterize the MPC feedback-law and the implementation of several additional consistency constraints.

One of the main assumptions of the proposed approach is that the switch, although a-priori unknown, is detected immediately. This allows for the corresponding MPC controller to be solved, and hence secures feasibility and stability provided the appropriate dwell-time has taken place. If the switch is not instantaneously detected, the control input will be set by an MPC controller fitted with an inaccurate prediction model and constraints, most likely leading to constraint violations. An avenue for future work is then the inclusion of neighbour to neighbour robustness, as in [127], in order to allow for a delayed detection of the switch, possibly through analysis of the input-output data.

Another direction for future work is, of course, the computation of a guaranteed tighter upper bound d_m. This parameter has, possibly, the highest impact in the quality of the MDTs that the proposed approach provides, hence obtaining a guaranteed value, rather than a numerically estimated one, is necessary.

Chapter 5

Distributed MPC for dynamically coupled systems: a chain of tubes

5.1 Introduction

The past decade has seen a rapid increase in the demand for more efficient, reliable and safer providing of services [139], particularly in fields such as power and transportation, where the lack of coordination can cause large economic losses [140]. This need has been met with considerable improvements in the fields of computational capabilities, data acquisition and wireless communication [141–143], which have allowed the implementation of several concepts of process control and systems engineering in order to cope with an ever increasing demand on performance and coordination of such large-scale systems.

A particularly important challenge, in the pursue of efficiency and reliability, is the size of these systems, both physical and digital (data). Power networks, for example, are composed by several generating, distributing and consuming agents that are spread over a wide physical area (possibly continents) [144]. The control and coordination required to meet the power demand then requires a large amount of information to be transmitted over large distances. Communicating all this data to a central hub may be prohibitively expensive, slow, and prompt to data loss [145], which is why a centralized coordinator (controller) may not be a feasible solution in terms of of safety and reliability.

Some of these issues are particularly aggravating in the context of control via MPC. Although MPC is implemented in a receding horizon fashion and hence could account for some uncertainty introduced by data miscommunication, large data losses would render the predictions unreliable. Furthermore MPC needs to solve an optimization problem at each time instant a new control action is to be applied, and although the MPC optimization can be posed as a QP problem, the latter may not be solvable fast enough if a large amount of actuators and constraints are considered (which would be the case of large service-providing networks). Finally, and possibly more importantly, invariant sets are considerably complex to compute for high-dimensional plants, resulting in that many of the theoretical guarantees associated to (robust) MPC controllers might not be available when controlling a large-scale system.

Nevertheless, "*the primary challenge to implementing centralized MPC is not computational, but organizational*" [1]. A large-scale network might already have several local MPC controllers in place that are able to cope with the size of their local tasks, hence to improve performance it is not necessary (or feasible) to design a centralized controller, but to properly coordinate the existing agents. In this context, another paradigm in which centralized control may not be a feasible solution is the coordination of a network composed by several independent subsystems with clearly defined physical boundaries and concurrent objectives. Consider for example the problem of coordinating a platoon of vehicles within a transport network [146]. Platoons may form at any location in the network and comprise any number of concurring vehicles, hence a centralized coordinator may need to be unnecessarily complex to account for all the different possibilities. Another example can be found in the coordination of a swarm of robots. Such architectures usually comprise a high number of independent agents yet the interactions only happen between a few of them at a time. Furthermore, the task assigned to a robot may be different to that of its neighbour, hence resulting in a possibly complex centralized controller.

A natural approach to overcome some of these challenges is to take into account the pre-existing partition of the network or proceed to partition it in a number of smaller subsystems, and then to design a controller for each one of the resulting elements. How to perform the partition is not a trivial task, yet depending on the system a pre-existent partition might already be in place (in a swarm of robots, for example, each robot would become a subsystem). Nevertheless, regardless of the partition, the subsystems are still part of a larger network, hence their local behaviour will influence and be influenced by the behaviour of its corresponding neighbouring subsystems. Depending on the type of plant/process being studied, and on how the splitting is performed, the subsystems may interact through their dynamics (dynamic coupling) or through their constraints (constraint coupling) [139]. Consider for example a transport network in which the different agents are independent freight hubs or destinations [147]; the outgoing cargo of the different subsystems becomes the incoming cargo for its neighbours, hence there exists a coupling in the storage dynamics. Alternatively, in the case of fleets of unmanned vehicles all subsystems are dynamically independent, yet to avoid collisions it is necessary to impose constraints on trajectories depending on the trajectories of the neighbour [148, 149].

If disregarded, these interactions may act as unexpected disturbances affecting the local dynamics of each subsystem, and so they must be taken into account if theoretical guarantees are to be provided. Furthermore, it is most likely that communication between the different agents needs to take place, in order to properly coordinate and achieve network-wide efficient performances. If the controller for each individual subsystem is designed as an MPC, the overall network controller can be filed under the label of non-centralized MPC. Non-centralized MPC architectures can be classified under several subcategories depending on whether the coupling is acknowledged or not, the type of coupling that is considered, the level of communication established between the different controllers (also called agents from now on), and whether these cooperate to achieve a network-wide goal or strive for their own local objectives in a selfish manner. A comprehensive description of these categories and the several algorithms that have been proposed to date can be found in [139, 143, 150].

If the interaction is entirely neglected, the overall architecture is usually referred to as decentralized. The latter suffer from a number of disadvantages that originate precisely from neglecting the interaction between sub-systems. As an example, a brief analysis about the effect that closing the loop of one sub-systems has on the rest is made in [151]. Results show that zeros of the transfer function of the other subsystems may be moved into the right hand plane, causing instability. Preliminary results on the implementation of a decentralized MPC controller are shown in [152]. A large unconstrained linear system is divided into three smaller sub-systems in such a way that there is a residuary state coupling. This interaction, however, is entirely neglected and a standard MPC is implemented for each one of the agents. The results are meant to be just illustrative, yet the applicability of such a scheme is demonstrated.

In what follows some of the most relevant non-centralized MPC proposals are discussed in order to contextualize the approach presented in this chapter. The focus is placed on Distributed MPC (DMPC) architectures for dynamically coupled networks, which is the type of network addressed in this chapter. The latter acknowledge the possible interaction between neighbouring subsystems and therefore its coordination requires a certain level of on-line communication between the agents. Within DMPC two subcategories are observed:

- Non-cooperative DMPC: Each local controller tries to optimise its own selfish objective, and does not account for the effect this may have on its neighbours' performance.
- Cooperative DMPC: Each local controller tries to optimise a wider performance measure that takes into consideration its neighbours' objectives.

Generally speaking, the network performance achieved by non-cooperative implementations is expected to be lower than that of cooperative ones given that the former are characterized by the optimization of local selfish objectives [150]. Nevertheless, cooperative architectures may still result in lower performance than centralized control implementations due to lack of information or time to converge [150]. This is the trade-off that exists when distributing the problem.

Cooperative DMPC implementations strive for an improved plant-wide performance, and so are often posed as a distributed implementation of a centralized MPC controller requiring distributed optimizations. Furthermore, since each agent accounts for the impact of its own actions on its neighbours, each agent in a cooperative DMPC architecture usually requires knowledge of its neighbours' entire dynamics in order to make the necessary predictions. Non-cooperative DMPC algorithms, on the other hand, usually need knowledge of the interacting dynamics only, since each agent only cares about how these may affect local objectives. Nevertheless, the design of most DMPC algorithms requires the coordination of and communication between groups of neighbours, at least in order to share constraint and coupling information at the design stage.

In view of the above, an important feature of the various DMPC algorithms is the number of steps involved in obtaining a feasible and stabilizing control action for each subsystem at each time instant. Cooperative and non-cooperative DMPC architectures usually employ some form of online communication so that agents can inform neighbours of their plans. Each agent then takes this information into account to solve its own optimization (cooperative or not), yet if this is done simultaneously by many agents the information previously shared becomes instantaneously inaccurate. Several options exist to account for this, but most DMPC architectures employ robustness, converging iterations, or a combination of both.

The idea of allowing different MPC agents to communicate during operation was first introduced in [153]. Initially the problem is presented in a very general set-up allowing for different types of interaction between the subsystems (which are assumed linear and constrained). Each local agent minimizes a local cost that may depend on the state and inputs of other subsystems (due to dynamical coupling) hence neighbouring agents need to communicate. It is shown in [153] that, provided the overall objective (cost function) of the large-scale system can be exactly decomposed and associated to the different subsystems resulting from the partition, then an iterative approach to solving the local optimization problems results in a solution to the global problem. The iteration follows a standard iterative communication protocol:

1. Agent i computes a solution to its local control problem by assuming fixed values for its neighbours' variables.
2. The solution is broadcast to all the agents whose dynamics are influenced by agent i. Analogously, agent i receives updated values for the variables that were first assumed fixed.
3. The optimisation problem is solved iteratively until convergence is reached.

The drawback of the first proposal in [153] is the necessity of iteration until convergence. Note that this iteration takes place within a single sampling time, and hence the time required to achieve convergence may be prohibitively high albeit each independent optimization is of reduced size due to the partition. A tentative solution is put forward in [153] that allows to stop iterations at any time and still guarantee stability of the network-wide closed-loop. This proposal is based on the inclusion of a Lyapunov type constraint at the beginning of the prediction horizon [154], however stability is only attainable for unconstrained linear systems with loose dynamical coupling.

A similar approach is taken in [155], where an MPC algorithm that relies on a contractive constraint [156] is employed for the design of each agent in a network composed by non-linear subsystems subject to input constraints and state coupling. A key difference, however, is that the interaction between subsystems is considered as bounded decaying disturbances. This is an implicit way of addressing the interaction since it requires no on-line communication between agents, just the knowledge of an initial bound on the interacting effects and an expected decay rate which is also computed off-line. This type of disturbance is readily handled by the MPC approach in [156], which allows to guarantee asymptotic stability of the origin given a possibly open-loop unstable plant.

If the dynamical interactions are not assumed decaying, it is not possible to guarantee stability of the origin. This issue is addressed in [157], where state constraints are included in the analysis. The dynamical interaction is now seen as a bounded disturbance and input-to-state stability (ISS) is guaranteed for each system provided that the interaction does respect these bounds and a terminal invariant set exists for the local dynamics of each subsystem. Recursive feasibility of each optimization is ensured by tightening the state constraint of each subsystem (similar to the tube approach but without a properly defined tube or gain), after which ISS of the whole network follows from ISS of each subsystem.

5.1.1 Cooperative DMPC

A cooperative approach based on the truncated infinite horizon implementation depicted in [8] is developed in [158, 159] for a network of undisturbed LTI systems coupled through their inputs. First, an iterative non-cooperative implementation in which each agent minimizes an individual cost is shown to result in a Nash equilibrium [160–162]. A cooperative approach, in which each agent minimizes a plant-wide cost, is then proposed in order to achieve a Pareto optimal solution; that is a plant-wide optimum. Plant-wide optimality is sufficient to guarantee stability (given the lack of state constraints), yet it can only be obtained if the distributed optimization is allowed to converge at each time instant. If the iterations are terminated early, the resulting controller is not necessarily stabilizing. This issue is solved by including a convex combination step in the iteration. At each step the local agents optimize their input sequences given past values of their neighbours plans, yet this new optimal trajectory is not directly informed to their neighbours. Instead, it is averaged with the previous broadcast solution, and the resulting sequence is communicated. Provided an adequate initialization, this additional step ensures that early termination of the distributed algorithm results in stable closed-loop for the network.

A quasi-cooperative DMPC architecture is presented in [163, 164] for controlling a pair of constrained linear subsystems coupled through their inputs. The overall scheme only requires two steps of communication and two steps of optimization at each sampling instant, however system-wide optimality is not guaranteed. Each agent is fitted with its own selfish objective, yet the prediction throughout the horizon is done with explicit account of its neighbours' plan; the algorithm can be summarized in four steps. First each agent optimizes its local input with respect to its own objective and assuming its neighbours' plans fixed. Then each agent optimizes its neighbours' input with respect to its own objective again, while fixing its local input to the recently obtained optimal. After these values are computed they are shared and each agent evaluates its own cost with all possible sequence combinations. These values are then shared after which each agent chooses the combination that minimizes the plant-wide cost (defined as the sum of each individual one).

An evident drawback of this implementation is the exponential increase on complexity with the addition of new subsystems. It is not clear what the strategy would be, yet at least a total of M^M optimisations must be performed and $(M+1)^{(M+1)}$ combinations evaluated, where M is the number of agents. In [164] state and input constraints are included and stability is guaranteed by usual terminal conditions. Given the distribution of the problem, however, the latter cannot be computed using standard approaches. Instead a jointly robust collection of terminal controllers and constraints are computed through a set of LMI optimization problems, which results in an overall centralized design process, as opposed to the decentralized approach achieved in [158] allowed by the lack of state constraints.

The same authors proposed another approach in [165] to provide a solution that is less computationally demanding in the case of more than two subsystems. The distributed control algorithm is different to [164] in that only one agent is allowed to optimize at a time (or several non-concurring ones). Furthermore, the optimization is done simultaneously over a subset of input trajectories that affect its dynamics (not sequentially as in [164]). These tentative new solutions are then broadcast and evaluated by the agents of all other affected subsystems. The resulting difference in cost (with respect to continuing with a previously feasible solution) is then communicated back to the optimizing agent, which makes a decision to update only if the overall cost has decreased (infeasibility is assigned an infinite cost increase). The second contribution of this proposal is in the stability result. In [164] a decentralized linear controller is sought to fulfil the standard terminal conditions in MPC implementations. This choice is simple but it can yield conservative results, hence it is replaced in [165] by a distributed linear terminal controller (i.e. one that is not block-diagonal). The corresponding gains and terminal sets are again computed as a jointly robust collection through a larger LMI problem.

A finite horizon MPC is proposed in [166] to control each subsystem within a network of input coupled LTI subsystems subject to input constraints. The overall communication and iteration procedure is analogous to that in [158], including the convex combination step; the main difference between both approaches lies in the optimization associated to each MPC agent. The stability induced by considering infinite horizons in [158] is achieved in the finite horizon framework considered by [166] through the inclusion of a Lyapunov type constraint in the optimization. The same approach is extended to non-linear subsystems possibly coupled through their states in [167], yet no state constraint is considered. Similarly to [158] then, the design procedure is entirely distributed and simpler than that of [164, 165], yet this is only possible due to the lack fo state constraints.

A generally different framework is tackled in [168], where each subsystem is considered to have an independent input but they all share the exact same state vector. In this context each agent optimizes a single input, yet it requires knowledge of the overall network dynamics and planned inputs to make predictions (as opposed to [158, 164–166]). Two architectures of Lyapunov based MPC controllers [17] are proposed. First a sequential one, in which each agent optimizes its trajectories in a cascade, and so the currently optimizing one only knows what the previously optimized ones will do. The not yet optimized plans are supposed to follow a Lyapunov-based control law defined off-line. The latter is assumed to be capable to stabilize the non-linear plant in a decentralized fashion (as in [164]). The second architecture allows simultaneous optimization, and hence it resorts to iteration in order to eliminate (or reduce) the uncertainty introduced. State constraints are not considered, and ultimate boundedness, in the presence of disturbances, is guaranteed for both architectures.

An approach similar to that in [166] is presented in [169], but in this case the subsystems are assumed to be coupled through their states, and state constraints are accounted for. Local terminal conditions are employed to guarantee asymptotic stability of the origin for the network, and a similar approach to that in [164] is proposed for structuring the terminal feedback gain and its corresponding terminal cost function. However, the requirements are relaxed to allow for a local cost increase in benefit of a global cost decrease. The terminal constraint sets, then, are allowed to be time-varying in order to account for a possible local increase in the terminal cost. An important contribution of [169] is that sufficient conditions are found to meet the design requirements in a distributed fashion, by finding a distributed upper bound to the globally coupling LMI that appears in the process of computing the terminal gains (same constraint that appears in [165]).

5.1.2 Non-cooperative DMPC

A non-cooperative framework for non-linear systems with state coupling and subject to input constraints is presented in [170]. Each subsystem is linearised around the origin in order to compute a decentralized linear gain that stabilizes each local dynamics and the overall network dynamics as well. The latter implies that a weak coupling in the linearised network is required. This decentralized gain (valid only around the origin) is employed to compute an invariant set for the network, from which local terminal constraints are derived and included in the optimization problem of each agent. As opposed to standard MPC implementations, this terminal controller is not used to prove stability over the whole RoA of the local controller, but to actually control the network once the closed-loop has reached the terminal region (mode-2 controller). Convergence to the terminal region is guaranteed by the inclusion of a contractive constraint which forces the current state predictions to be smaller (in some sense) than those from the previous step.

Coordination between the different agents is achieved in [170] by a single step of on-line communication (at each sampling instant) in which predicted state trajectories from the previous time-step are shared. Since all agents are allowed to optimize simultaneously, the shared information is outdated. The ensuing prediction mismatch is dealt with by including a *consistency* constraint in each optimization, which forces the current predicted trajectories to remain close to those previously informed to neighbours. This is done to ensure a bounded prediction mismatch, and so the existence of a feasible solution at each time instant (indeed, that the tail of the previous solution concatenated with the terminal controller is feasible). This approach is comparable to the tightening of the state constraints in [157], yet in a more comprehensive manner since the tightening is done around a previously feasible and converging optimal trajectory.

A network of LTI subsystems coupled through constraints and subjected to independent persistent bounded disturbances are studied in [171]. Each local controller is designed following the tube approach in order to guarantee constraint satisfaction and robust stability despite of the local disturbances acting on the subsystems (the coupled constraints are also tightened). The coupling is explicitly tackled by augmenting the standard tube-based MPC optimization of each agent with the set of coupled constraints in which the agent takes part. To guarantee recursive satisfaction of such constraints only a single agent is allowed to optimize its control sequence during a time step, as opposed to [170] where all agents optimize simultaneously. The remaining agents define their current control action by falling back to a previously feasible sequence. Each time an agent is allowed to update its optimal control sequence it is also required to broadcast it among its neighbours, hence the optimizing agent knows exactly what its neighbours will do and constraints can be met accordingly.

A plant-wide feasible and stabilising control plan is necessary for initialising the distributed sequential algorithm in [171], however this is an assumption found in several DMPC algorithms. Recursive feasibility of the optimization follows from standard terminal controller arguments and the fact that the neighbours plans are supposed to be exactly known. Robust stability then follows by the same arguments than in standard centralized tube MPC [2]. In [31] an extension is proposed to allow for the optimization of multiple subsystems with concurrent constraints. The general idea is to further tighten the coupled constraints in order to account for the uncertainty arising from simultaneous optimization. The second tightening is done in a time dependent fashion, similar to [23], in order to reduce the conservativeness introduced.

The sequential non-cooperative scheme depicted in [171] is extended to a single-iteration cooperative framework in [172]. To promote cooperation each local controller minimises a cost function that depends not only on its local objective, but also on a weighted sum of the objectives of neighbouring subsystems. Said agent optimises w.r.t its own input and that of this subset of neighbours. To this effect the MPC optimization problem is augmented with the (tightened and coupled) constraints corresponding to the considered neighbours. The ultimate purpose of optimizing neighbours' plans is to account for what they may do, and hence improve plant-wide performance, but not to tell other agents what to do. In view of this the resulting local optimal sequence is used to update previous local plans, while the optimized plans of neighbours' inputs are discarded given that only one agent is allowed to update at any single time. Since several objectives are optimized simultaneously, an increase in local cost could take place at any given time instant, hence an additional cost decrease constraint is employed to guarantee stability with the usual arguments.

The tube-approach to robust MPC has also been employed in the context of non-cooperative DMPC for networks of dynamically coupled subsystems. Indeed, as shown in [30, 33–35, 173], exact knowledge of the dynamical coupling between neighbouring subsystems allows to treat the whole interaction as an external perturbation affecting the local dynamics. This disturbance can be bounded on the basis of constraint satisfaction assumptions, after which standard tube MPC implementations can be readily implemented to guarantee robust constraint satisfaction and stabilizability (for the local and global closed-loop systems). The resulting controller does not need on-line communication with its neighbours, yet at the expense of a high degree of conservatism (large disturbance sets resulting in small RoAs). In order to decrease this conservatism, several additional control elements and communication steps have been proposed.

In [33] the tube approach is implemented twice, in series, for each subsystem in the network. It is assumed that each agent receives, at the beginning of a time step, information of what its neighbours plan to do, hence the first instance of tube MPC aims to introduce robustness against the uncertainty on such plans. The latter exists because all agents optimize at the same time, hence the informed plans are not necessarily followed. The second instance of tube MPC is employed to introduce robustness against the entirety of the neighbours' plans (not just their uncertainty), in order to eliminate external inputs from the local dynamics, and simplify the task of controlling each subsystem. The second tube also serves the purpose of an explicit bound on the allowable change between previous and current planned trajectories. The overall goal of this series of tubes is to reduce the conservativeness and improve performance by using the information provided by neighbours to specialize the robust control action, yet several drawbacks appear. For example, it is not clear whether this architecture results in an overall larger RoA when compared to the decentralized tube approach. Furthermore, the several tightening tubes depend on the tightened constraint sets in a circular way, which results in an iterative and centralized design procedure.

A similar tube-based implementation is devised in [34] to deal with constraint and dynamic coupling in a network of LTI subsystems. It is assumed that each agent will try to follow a reference trajectory defined off-line (both in the state and the input). These references are accompanied by a-priori defined bounds which gives agents some extra freedom to improve performance by locally modifying their predefined trajectories. At each time instant agents optimize local trajectories around the original ones. The optimized ones are not used to replaced the entire original reference as in [33], but only to add to the tail of it in order to guarantee feasibility (similarly to the rationale employed in [65]). Even so, the true state trajectories will be different from the planned ones due to the simultaneous updating of agents. To deal with this uncertainty tube MPC is employed and a corresponding RPI set is used to tighten the local constraints. The overall disturbance is computed by employing the off-line defined bounds for performance improvement and overall feasibility is achieved by an additional constraint on the change between original plans and local optimized plans (similar to the consistency constraints in [170]).

Similarly to [33], only one step of communication is required at each time-step, however the amount of information is decreased from a whole planned trajectory (over the horizon), to a single additional element. Stability of the origin is guaranteed for the network through the usual terminal conditions based on the same type of weak linear coupling assumption used in [33,170,171], and the approach is extended for output tracking in [174] and continuous time systems in [175]. Albeit [34] implements only one step of robust control, a reduction in the overall conservativeness of the approach is not immediate. The size of the disturbances affecting each subsystem is readily defined by the off-line selection of the allowable innovation on its neighbours. If these sets are small, the overall disturbance is small, leading to small tightening, yet this also reduces the freedom of the controller to deviate from the predefine trajectory, possibly reducing the overall space of the RoA used (even if it is larger), and putting a bound on performance. Furthermore, this architecture introduces several additional design parameters such as the allowed response variability size and the planned trajectories for initialization, both having a direct impact on the size of the feasibility region of the algorithm.

An approach similar to [34], but for non-linear subsystems, is presented in [176]. Each agent is designed as an MPC controller for a constrained set-difference equation, in order to embed robustness against the possible change in the planned trajectory of neighbouring subsystems. This variability is called a contract and is shared at every time step among neighbouring agents after predictions have been made. Contracts are computed as the k-step ahead reachability sets by considering the contracts that were informed by neighbours in the previous step as external inputs. This is, possibly, the biggest drawback of this proposal since contracts need to be computed at each time instant and reachability sets of non-linear closed-loops are not trivial to compute. Furthermore, this architecture requires a centralized initialization in order to obtain contracts that are jointly feasible. Finally, as in [33,34,170] a consistency constraint is required to guarantee that no agent will violate the contract that communicated to its neighbours the previous step.

A quasi-distributed approach is presented in [173] for output-feedback and to account for both constraint and dynamical coupling. It is assumed that each subsystem measures its outputs locally with some error, and a Luenberger observer is employed for a decentralized state estimation. The latter induces a prediction error, which is lumped with the prediction mismatch arising from ignoring the dynamical coupling and a doubly robust tube-based controller is proposed. The latter guarantees constraint satisfaction of every constraint except the coupled ones, since these are evaluated with outdated information provided by neighbouring subsystems which are allowed to optimize new trajectories simultaneously. In [31] this is dealt with a further tightening of the constraints, yet the inclusion of state estimation error already results in reduced RoAs, hence in [173] another solution is explored. After each (coupled) agent has optimized, the planned trajectories are shared and the coupledconstraints evaluated. If violation takes place, the new optimized plans are averaged (in a convex combination) with the previously feasible ones, which results in feasibility.

In [35] a succession of MPC controllers is employed in series to guarantee stabilizability and constraint satisfaction in a network of LTI subsystems that are dynamically coupled. The approach is similar to [33] in that the resulting control law is formed by three different terms acting on different representations of the plant, yet the overall architecture is considerably more complex. The first step is a standard tube MPC implementation that is robust against the entirety of the dynamical interaction. The second step is again a tube-inspired MPC implementation used to replace the linear control law usually employed to reject disturbances. The later gives way to an implicit non-linear control law that regulates the error between real and predicted trajectories taking into account previously obtained plans of what the neighbours may do. As opposed to standard tubes, constraint satisfaction cannot be readily guaranteed because all agents are allowed to optimize at the same time, and so the plans accounted for are not accurate.

In [34, 170, 176] the issue of inaccurate information of what the neighbours may do is dealt with by explicitly including consistency constraints in one of the optimization problems. In [35] these are implicitly introduced by the third term in the control law, which is defined to be invariant inducing given the uncertainty on the neighbours plans. Furthermore, [35] proposes a tightening of the constraint sets in the different optimization problems based on scaling factors and not on explicitly computed invariant sets, which is an advantage given the complexity involved in obtaining the latter. Nevertheless, a drawback of this nested approach, when compared to [31, 33, 34, 173], is that each agent needs to solve an additional optimization problem at each time instant.

A conceptually different approach is proposed in [30, 177] to reduce the conservativeness introduced by decentralized tube-based MPC. The driving argument is that MPC-controlled systems do not necessarily use their whole constraint space, particularly the state constraint and specially when the state is already close to the origin. This notion is formalized by parametrizing polytopic constraint sets by scalars on the right hand side of the defining half spaces. These scalars are then included in the MPC optimization, so as to force their minimization, and so the overall space that state and input trajectories use. Ultimately this means that the overall disturbance neighbours need to account for is decreasing over time, hence allowing for looser a tightening and an overall larger RoA for each tube MPC. In order to take advantage of this, one step of communication is required, as well as the on-line computation of robust invariant sets. The latter might render the whole approach inapplicable, yet several solutions exist that allow to obtain (coarse) invariant sets with low computational demand [100, 101]. Recursive feasibility under continuous modification of constraints is guaranteed by forcing the parametrizing scalars to be non-increasing, resulting in a standard proof of exponential stability of the origin for the overall network.

5.1.3 DMPC using a chain of tubes

In this chapter a new non-cooperative DMPC approach is presented. The object of study is a network of LTI subsystems subject to state and input constraints and dynamically coupled through their states and inputs. The general modelling structure of each subsystem is as outlined in Section 2.2, nevertheless the interacting network dynamics are introduced in Section 5.2 to provide the necessary notation for the chapter. The overall approach to distribution is based on robustness, as in most of the non-cooperative DMPC algorithms presented up to date, and it only requires one step of communication and two steps of optimization solving at each sampling time (i.e. a non-iterative approach). The controlling agent for each subsystem is designed as a two-step robust controller, in which each step is driven by a tube-based MPC controller as described in Section 2.3.1. The tube technique is chosen given its negligible added complexity, when compared to standard MPC, which allows to focus the design efforts in the additional elements required for coordination.

The inner robust control step, presented in Section 5.3.1, considers the entire interaction with its neighbours as a disturbance. The latter is bounded in view of network-wide constraint satisfaction assumptions and knowledge of the interacting models, which then allows for an off-the-shelf tube MPC controller to be designed (similar to [157]). This step represents, possibly, the main drawback of the proposed approach, since large interactions may render impossible to compute admissible RPI sets for tightening. This issue can be dealt with by pre-tightening the allowable constraint space of certain elements of the network, yet this will result in a reduced RoA of the plant when compared to a hypothetical iterative scheme. Feasibility provided, the results from the inner step of robust control are a set of predicted optimal (nominal) trajectories for both state and input and a nominal input that can be used in the standard disturbance rejection policy of tube MPC to feasibly control each subsystem.

In the proposed algorithm, however, the latter is disregarded, and the predicted state and input trajectories are broadcast in full between neighbouring subsystems. The receiving agent sees this information as the intended plans of its neighbours, that is, what they would do if they were not being disturbed, and a second (outer) step of robust control is employed to obtain a refined optimal input trajectory given this new information. At this stage the interacting models are used to include the effect of the neighbours' plans as uncontrolled external inputs in the prediction. This second step also needs to be robust because, in principle, all agents are allowed to run the inner step simultaneously at each time instant, hence nominal reference trajectories informed between neighbours become instantaneously inaccurate (as in [31, 33, 35, 170, 173]). In order to define the amount of robustness needed for the outer controller, the optimization associated to the latter is fitted with an additional constraint which allows for a comprehensive design and a guarantee on recursive feasibility of the overall scheme.

The outer controller is presented in Section 5.3.2. Structurally, this step is equivalent to the robust controller proposed in [34], however there are several design and implementation differences. First of all, the reference trajectories proposed in [34] need to be computed off-line and are only updated one element at a time by adding a step of the terminal control law at the end. This approach results in that the reference trajectories become critical tuning parameters that have a direct impact in the overall RoA of the controller. Furthermore, the updating scheme implicitly induces suboptimality by forcing the predicted trajectories to be close to the terminal set after a number of time steps equal to the prediction horizon. Alternatively, the algorithm proposed here not only computes the reference trajectories during operation, but does it in an optimal way at each time instant, favouring a reduction in structural conservatism and an increase in performance at the cost of an additional optimization step.

The inner-outer architecture resembles that of the approach in [33], but there are several differences. First, and more evidently, the approach in [33] considers only state coupling, while in this chapter coupling through the inputs is also analysed. Secondly, the tube-based approach proposed in [33] corresponds to the one described in Section 2.3.2, while here the initial nominal state is optimized at each time instant.

More specific differences are that the robust steps of control in [33] are applied in series, that is the overall control law of each agent is composed by an independent (nominal) term and the rejection policies of both tubes. In the approach proposed here the steps of robust control are applied in parallel. The control law of each agent then depends on an independent (nominal) term and a single tube rejection policy (although the independent term depends implicitly on the results of the inner step of robust control since the predicted trajectories associated to it are used in the predictions of the outer step). It is expected that this favours an increase in performance, since the tube policy arising from the more conservative tightening is entirely disregarded, allowing for more freedom in the control law definition. It is the case, however, that if the gains of both tubes are chosen to be the same, the control laws of [33] and that proposed here equate in structure, yet the steps of robust control are applied in an inverse manner and so the true control action will ultimately be different.

The additional constraint incorporated to the outer optimization fulfils, in essence, the same role as the consistency constraints in [170], the allowable innovation constraints in [33, 34] and the second step of tightening in [31, 176]. The main difference with [34] is that the constraints proposed here are not arbitrarily defined, but constructed following invariance requirements to ensure the overall feasibility of the two-step controller. The latter is similar to [33], but does not require the computation of global RPI sets for the entire network, which is usually computationally intractable for large-scale plants.

Most of the DMPC algorithms presented up to date enjoy strong theoretical guarantees on stability and constraint satisfaction, and the trade-off is present in the overall performance achieved and the computational capabilities required at each time step to compute the corresponding control action. Nevertheless, several DMPC algorithms may not be easily implementable due to the complexity of their design procedures, particularly when there is a need for computing centralized elements as in [33]. The features of most algorithms proposed to date rely, either explicitly or implicitly, on the existence of a linear stabilizing feedback for the overall network with a particular structure associated to the interaction pattern of the plant. It is seldom the case, however, that guidelines for computing such feedback are provided, or even its existence shown [30, 33–35, 170, 176]. In [164, 165, 169] this issue is explicitly tackled and a suitable gain is shown to be obtainable from the solution of a set of LMIs, yet only in [169] this procedure is non-centralized.

The stability features of the non-cooperative DMPC algorithm proposed in this chapter also rely on such an assumption. The latter cannot be easily relaxed, however sufficient conditions that guarantee the existence of an appropriate linear gain are provided; furthermore, one such feedback is presented. The arguments that allow for such sufficient conditions follow the analysis of a fundamental relationship that exists between this type of stabilizability assumption and the various concepts of invariance used in tube-based DMPC approaches. The design of each agent is discussed in Section 5.4, placing particular attention in how to guarantee recursive feasibility of the overall DMPC algorithm, while Section 5.5 discusses how to meet the corresponding stabilizability assumption made over the network.

5.2 Network of dynamically coupled linear systems

First consider the problem of regulating a large-scale system for which a discrete time state space representation is available. Consider such a model to be

$$\boldsymbol{x}(t+1) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t)$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the state vector, $\boldsymbol{u} \in \mathbb{R}^m$ is the input vector and $(\boldsymbol{A}, \boldsymbol{B})$ are the state and input matrices of corresponding dimension. Now, assume that this system is composed by M dynamically coupled previously partitioned subsystems, such that their driving states and inputs are not overlapping (as opposed to [168], for example, where all subsystems share the same state vector).

The network's state and input vectors can then be rewritten as

$$\begin{aligned}\boldsymbol{x} &= \left(x_1^\top, x_2^\top, \ldots, x_M^\top\right)^\top \\ \boldsymbol{u} &= \left(u_1^\top, u_2^\top, \ldots, u_M^\top\right)^\top\end{aligned}$$

where $(x_i, u_i) \in \mathbb{R}^{n_i} \times \mathbb{R}^{m_i}$ are the state and input pair associated to subsystem i and the non-overlapping requirement implies

$$(n, m) = \left(\sum_{i=1}^{M} n_i, \sum_{i=1}^{M} m_i\right).$$

The dynamics of each subsystem can then be cast as

$$x_i(t+1) = A_{ii}x_i(t) + B_{ii}u_i(t) + \sum_{j\in\mathcal{M}/\{i\}} (A_{ij}x_j(t) + B_{ij}u_j(t)) \tag{5.1}$$

where $\mathcal{M} = \{1, \ldots, M\}$, and for all $i, j \in \mathcal{M}$ the matrices (A_{ij}, B_{ij}) are the corresponding block elements of $(\boldsymbol{A}, \boldsymbol{B})$ of appropriate dimension. The summation term in (5.1) represents the coupled dynamics that arise from partitioning the system and lumps the effect that all other subsystems in the network have over subsystem i. However, it is usually the case that the direct interactions within a network can be represented by a strongly connected graph, rather than a complete one. Thereby each subsystem i will only be coupled with a subset of the entire network. To properly represent these interactions three sets are now introduced

$$\begin{aligned}\mathcal{N}_i^u &= \{j \in \mathcal{M} \setminus \{i\} \mid [A_{ij}\ B_{ij}] \neq \mathbf{0}\} \\ \mathcal{N}_{i+}^u &= \mathcal{N}_i^u \cup \{i\} \\ \mathcal{N}_i^d &= \{j \in \mathcal{M} \setminus \{i\} \mid [A_{ji}\ B_{ji}] \neq \mathbf{0}\}.\end{aligned}$$

$\mathcal{N}_i^u$ is the set of upstream neighbours of subsystem i and contains the indices of all the subsystems in the network whose dynamics affect those of subsystem i. The set $\mathcal{N}_{i+}^u$ is the same as the former, but including the sub-index i. On the other hand, $\mathcal{N}_i^d$ is the set of downstream neighbours of subsystem i and contains the indices of all subsystems whose dynamics are affected by subsystem i. Note that for any pair $i, j, \in \mathcal{M}$, $j \in \mathcal{N}_i^u$ is equivalent to $i \in \mathcal{N}_j^d$, but does not imply $j \in \mathcal{N}_i^d$. In fact, for any $i \in \mathcal{M}$ it could happen that $\mathcal{N}_i^u \cap \mathcal{N}_i^d = \emptyset$.

In view of this the local state space representation can be simplified to

$$x_i(t+1) = A_{ii}x_i(t) + B_{ii}u_i(t) + \sum_{j \in \mathcal{N}_i^u} \left(A_{ij}x_j(t) + B_{ij}u_j(t)\right) \tag{5.2}$$

In what follows it is assumed that each subsystem is subject to local constraints of the form

$$x_i(t) \in \mathbb{X}_i \subset \mathbb{R}^{n_i}$$
$$u_i(t) \in \mathbb{U}_i \subset \mathbb{R}^{m_i}$$

where $\mathbb{X}_i$ and $\mathbb{U}_i$ are assumed to be $\mathcal{PC}$-sets, as in Chapter 2. At this point an assumption on the decentralized stabilizability of the system is made.

Assumption 5.1. The pairs $(\boldsymbol{A}, \boldsymbol{B})$ and (A_{ii}, B_{ii}) are stabilizable. Furthermore, there exists a collection of linear gains K_i such that $\boldsymbol{F} = \boldsymbol{A} + \boldsymbol{BK}$ and $F_{ii} = A_{ii} + B_{ii}K_i$ are Schur for all $i \in \mathcal{M}$ with $\boldsymbol{K} = diag\,(K_1, \ldots, K_M)$.

Assumption 5.1 is an specialization of Assumption 2.1 for the case of a block-diagonal feedback gain. Such an assumption, or a similar one, appears in several DMPC algorithms proposed to date [30,33,34,165,169,173], and its necessity will be made clear in subsequent sections. Section 5.5 presents sufficient conditions to guarantee that a particular feedback meets Assumption 5.1.

5.3 Distributed MPC using a chain of tubes

5.3.1 Inner step of robust control

The first step of robust control is performed considering the entire dynamical interaction in the network as an unknown disturbance affecting the local dynamics of each subsystem. This step of decentralized control is performed not to robustly control each subsystem, but to define a reference trajectory that serves the purpose of informing neighbours of local planned actions. Nevertheless, this is not the true trajectory that each subsystem will follow since it will be modified in the second step of robust control to account for the information received. For simplicity, and unless otherwise stated, the following analysis and developments are performed for a single subsystem/agent i but are valid for all subsystems/agents in the network.

Consider the local dynamics in (5.2) and define

$$w_i(t) = \sum_{j \in \mathcal{N}_i^u} \left(A_{ij} x_j(t) + B_{ij} u_j(t) \right).$$

It follows that if constraints are satisfied by all subsystems $j \in \mathcal{N}_i^u$ at time t, then $w_i(t) \in \mathbb{W}_i$ with

$$\mathbb{W}_i = \bigoplus_{j \in \mathcal{N}_i} \left(A_{ij} \mathbb{X}_j \oplus B_{ij} \mathbb{U}_j \right). \tag{5.3}$$

The local dynamics of each subsystem can then be simplified to a constrained linear state space model subject to additive bounded disturbances

$$\begin{aligned} x_i(t+1) &= A_{ii} x_i(t) + B_{ii} u_i(t) + w_i(t) && \text{(5.4a)} \\ x_i(t) &\in \mathbb{X}_i && \text{(5.4b)} \\ u_i(t) &\in \mathbb{U}_i && \text{(5.4c)} \\ w_i(t) &\in \mathbb{W}_i. && \text{(5.4d)} \end{aligned}$$

In view of (5.3) and (5.4) then, tube MPC as described in Section 2.3.1 is readily applicable to robustly control each subsystem independently. Define then an undisturbed nominal model

$$\hat{x}_i(t+1) = A_{ii} \hat{x}_i(t) + B_{ii} \hat{u}_i(t)$$

with associated trajectory error $z_i(t) = x_i(t) - \hat{x}_i(t)$ and an admissible RPI set $\mathbb{Z}_i$ for the error dynamics with associated linear gain $\hat{K}_i$ resulting in nominal tightened constraints

$$\begin{aligned} \hat{x}_i(t) &\in \hat{\mathbb{X}}_i \subseteq \mathbb{X}_i \ominus \mathbb{Z}_i \\ \hat{u}_i(t) &\in \hat{\mathbb{U}}_i \subseteq \mathbb{U}_i \ominus \hat{K}_i \mathbb{Z}_i. \end{aligned}$$

The optimal control problem (OCP) associated to the inner step of robust control is

$$\mathbb{P}_{N,i}^1(x_i(t)): \qquad \min_{\hat{\boldsymbol{u}}_i, \hat{x}_{i,0}} J_{N,i}^1 \left(\hat{\boldsymbol{u}}_i, \hat{x}_{i,0} \right) \tag{5.5a}$$

s.t. (for $k = 0, \ldots, N-1$)

$$x_i(t) - \hat{x}_{i,0} \in \mathbb{Z}_i \tag{5.5b}$$

$$\hat{x}_{i,k+1} = A_{ii} \hat{x}_{i,k} + B_{ii} \hat{u}_{i,k} \tag{5.5c}$$

$$\hat{x}_{i,k} \in \hat{\mathbb{X}}_i \tag{5.5d}$$

$$\hat{u}_{i,k} \in \hat{\mathbb{U}}_i \tag{5.5e}$$

$$\hat{x}_{i,N} \in \hat{\mathbb{X}}_{f,i} \subseteq \hat{\mathbb{X}}_i, \tag{5.5f}$$

with $(\hat{x}_{i,k}, \hat{u}_{i,k})$ representing the nominal predictions at prediction time k for subsystem i.

If only this step of robust control is implemented, standard arguments can be employed to guarantee feasibility and stability of the local and global dynamics. Nevertheless, a second step of robust control is to be defined, hence proper guarantees are provided in the subsequent sections, once the overall distributed controller has been introduced.

Nevertheless, several elements are defined in a similar way to the non-distributed case presented in Section 2.3, such as the admissible RPI set $\mathbb{Z}_i$ and the terminal constraint set $\hat{\mathbb{X}}_{f,i}$. The latter, in particular, is assumed to be PI for the local undisturbed dynamics (A_{ii}, B_{ii}) when in closed-loop with a linear stabilizing gain $\hat{K}_{f,i}$ and admissible with respect to the tightened constraints $\hat{\mathbb{X}}_i$ and $\hat{\mathbb{U}}_i$. Furthermore, the feasibility region of the OCP (5.5) is defined as $\mathcal{X}^1_{N,i} = \hat{\mathcal{X}}_{N,i} \oplus \mathbb{Z}_i$ where $\hat{\mathcal{X}}_{N,i}$ is the feasibility region associated to constraints (5.5c)–(5.5f) (as in (2.8)). Finally, the optimum and optimal value of the cost function associated to (5.5) are defined by

$$\begin{aligned} \left(\hat{\boldsymbol{u}}^\star_i(x_i(t)), \hat{x}^\star_{i,0}(x_i(t))\right) &= \arg \mathbb{P}^1_{N,i}(x_i(t)) \\ V_{N,i}(x_i(t)) &= J^1_{N,i}(\hat{\boldsymbol{u}}^\star_i(x_i(t)), \hat{x}^\star_{i0}(x_i(t))) \end{aligned}$$

with $\hat{\boldsymbol{u}}^\star_i(x_i(t)) = \left\{\hat{u}^\star_{i,0}(x_i(t)), \ldots, \hat{u}^\star_{i,N-1}(x_i(t))\right\}$ and associated optimal state trajectory $\hat{\boldsymbol{x}}^\star_i(x_i(t)) = \left\{\hat{x}^\star_{i,0}(x_i(t)), \ldots, \hat{x}^\star_{i,N}(x_i(t))\right\}$.

The objective of this first step of robust control is not to generate a control action that will be fed to the plant, but to obtain reference trajectories that will be communicated among neighbours. In general terms, this information is nothing more than the optimal input and state trajectories that form a solution to the inner OCP, but in what follows the shared information will be depicted in a simplified manner as

$$\hat{\boldsymbol{u}}^s_i(x_i(t)) = \left\{\hat{u}_{i,0/t}, \ldots, \hat{u}_{i,N-1/t}\right\} \tag{5.6a}$$

$$\hat{\boldsymbol{x}}^s_i(x_i(t)) = \left\{\hat{x}_{i,0/t}, \ldots, \hat{x}_{i,N/t}\right\}. \tag{5.6b}$$

The super-index ^{s} is used to replace the optimality super-index of the sequences in order to allow the inner step of robust control to be solved at a lower frequency than the outer step. The goal of this is to allow for a reduced computational complexity of the overall scheme. When the inner OCP is not solved, the informed trajectories are constructed to remain feasible but not necessarily optimal (hence the replacement of the super-index). Furthermore, all super-indices are dropped from the elements of each sequence and the explicit dependency of each element on the current state is replaced by a sub-index time dependency $_{/t}$ in order to reduce notational complexity. Also note that the horizon is set to N for all agents, however the cost functions are allowed to be different. This obeys the outset of the problem which is a non-cooperative implementation in which each agent tries to improve its own measure of performance.

5.3.2 Outer step of robust control

After the information in (5.6) has been shared, agent i knows something about what its neighbours' plans are for the next N time steps. Agent i could then compute a better plan for itself by explicitly taking into consideration what its neighbours may do. Indeed, after the inner reference trajectories of neighbours have been received, agent i can use them to predict a part of the disturbances affecting subsystem i for all $k = 0, \ldots, N-1$ since

$$\begin{aligned} w^i_{k/t} = & \underbrace{\sum_{j \in \mathcal{N}^u_i} \left(A_{ij} \hat{x}_{j,k/t} + B_{ij} \hat{u}_{j,k/t} \right)}_{d_{i,k/t}} + \\ & + \underbrace{\sum_{j \in \mathcal{N}^u_i} \left(A_{ij} \left(x_j(t+k) - \hat{x}_{j,k/t} \right) + B_{ij} \left(u_j(t+k) - \hat{u}_{j,k/t} \right) \right)}_{v_{i,k/t}} . \end{aligned} \tag{5.7}$$

However, $x_j(t+k)$ and $u_j(t+k)$ are unknown future values expected to not match the reference trajectories exactly, given that neighbours will also make amends to their original plans. It follows then that $v_{i,k/t}$ remains uncertain. Nevertheless, the goal of sharing the reference trajectories among neighbouring agents is to gather some information about neighbours' plans and so reduce the uncertainty associated to considering the whole interaction as a disturbance. An assumption is made accordingly.

Assumption 5.2. After information has been shared among neighbouring subsystems, the remaining uncertainty on neighbours plans decreases. More specifically, there exists $\mathbb{V}_i \subset \operatorname{int}(\mathbb{W}_i)$ for all $i \in \mathcal{M}$ such that $v_{i,k/t} \in \mathbb{V}_i$ for all $k \in [0, N-1]$ and $t \geq 0$.

Assumption 5.2 is key in achieving the goal of the second step of robust control which is to improve performance by reducing conservativeness, yet it is not easy to achieve. In [34] this set is arbitrarily defined off-line. In this chapter, however, the collection of sets $\mathbb{V}_i$ is constructively defined to meet invariance conditions required to guarantee recursive feasibility of the optimization problems associated with the DMPC algorithm.

If Assumption 5.2 is met, and each agent assumes that the information received accurately corresponds to their neighbours plans for the following N steps, the prediction model (5.4) can be further specialized to

$$\begin{aligned} x_i(t+1) &= A_{ii}x_i(t) + B_{ii}u_i(t) + d_i(t) + v_i(t) \\ d_i(t) &= d_{i,t-\tilde{t}/\tilde{t}} \\ x_i(t) &\in \mathbb{X}_i \\ u_i(t) &\in \mathbb{U}_i \\ v_i(t) &\in \mathbb{V}_i. \end{aligned}$$

where $\tilde{t}$ is the current time instant (during which communication has taken place) and $t \geq \tilde{t}$. The outer step of robust control acts over this model and the argument is that, although uncontrolled, $d_i(t)$ is known and $v_i(t)$ is bounded, hence a tube-based approach can be implemented again. As before then, a nominal *undisturbed* model can be constructed

$$\hat{\hat{x}}_i(t+1) = A_{ii}\hat{\hat{x}}_i(t) + B_{ii}\hat{\hat{u}}_i(t) + d_i(t)$$

with associated trajectory error $s_i(t) = x_i(t) - \hat{\hat{x}}_i(t)$. Having computed an admissible RPI set $\mathbb{S}_i$ for the $s(t)$ error dynamics with associated linear gain $\hat{\hat{K}}_i$ results in nominal tightened constraints defined by

$$\begin{aligned} \hat{\hat{x}}_i(t) &\in \hat{\hat{\mathbb{X}}}_i = \mathbb{X}_i \ominus \mathbb{S}_i \\ \hat{\hat{u}}_i(t) &\in \hat{\hat{\mathbb{U}}}_i = \mathbb{U}_i \ominus \hat{\hat{K}}_i\mathbb{S}_i. \end{aligned}$$

The OCP associated to the outer step of robust control is

$$\mathbb{P}^2_{N,i}(x_i(t), \boldsymbol{f}_i(t)) : \qquad \min_{\hat{\hat{\boldsymbol{u}}}_i, \hat{\hat{x}}_{i,0}} J^2_{N,i}\left(\hat{\hat{\boldsymbol{u}}}_i, \hat{\hat{x}}_{i,0}, \boldsymbol{f}_i(t)\right) \tag{5.8a}$$

s.t. (for $k = 0, \ldots, N-1$)

$$x_i(t) - \hat{\hat{x}}_{i,0} \in \mathbb{S}_i \tag{5.8b}$$

$$\hat{\hat{x}}_{i,k+1} = A_{ii}\hat{\hat{x}}_{i,k} + B_{ii}\hat{\hat{u}}_{i,k} + d_{i,k/t} \tag{5.8c}$$

$$\hat{\hat{x}}_{i,k} \in \hat{\hat{\mathbb{X}}}_i \tag{5.8d}$$

$$\hat{\hat{u}}_{i,k} \in \hat{\hat{\mathbb{U}}}_i \tag{5.8e}$$

$$\hat{\hat{x}}_{i,N} \in \hat{\hat{\mathbb{X}}}_{f,i} \subseteq \hat{\hat{\mathbb{X}}}_i, \tag{5.8f}$$

$$\hat{\hat{x}}_{i,k} - \hat{x}_{i,k/t} \in \mathbb{H}_i, \tag{5.8g}$$

where $\boldsymbol{f}_i(t)$ represents the information received at time t and is formed by the pairs $\big(\hat{\boldsymbol{x}}_j^s(x_j(t)), \hat{\boldsymbol{u}}_j^s(u_j(t))\big)$ for all $j \in \mathcal{N}_{i+}^u$. The associated optimum and optimal value of the cost function are defined by

$$\begin{aligned}\big(\hat{\hat{\boldsymbol{u}}}_i^\star(x_i(t)), \hat{\hat{x}}_{i,0}^\star(x_i(t))\big) &= \arg \mathbb{P}_{N,i}^2(x_i(t), \boldsymbol{f}_i(t))\\ V_{N,i}^2(x_i(t), \boldsymbol{f}_i(t)) &= J_{N,i}^2(\hat{\hat{\boldsymbol{u}}}_i^\star(x_i(t)), \hat{\hat{x}}_{i0}^\star(x_i(t)), \boldsymbol{f}_i(t))\end{aligned}$$

with $\hat{\hat{\boldsymbol{u}}}_i^\star(x_i(t)) = \Big\{\hat{\hat{u}}_{i,0}^\star(x_i(t)), \ldots, \hat{\hat{u}}_{i,N-1}^\star(x_i(t))\Big\}$ and associated optimal state trajectory $\hat{\hat{\boldsymbol{x}}}_i^\star(x_i(t)) = \Big\{\hat{\hat{x}}_{i,0}^\star(x_i(t)), \ldots, \hat{\hat{x}}_{i,N}^\star(x_i(t))\Big\}$.

Again, the architecture of the controller makes it so the arguments employed in standard tube MPC implementations are not valid to guarantee stability of the closed-loop and recursive feasibility of the optimization. In particular, $\mathbb{P}_{N,i}^2(x_i(t), \boldsymbol{f}_i(t))$ is parametrized by the information received $\boldsymbol{f}_i(t)$ and is subject to constraint (5.8g) which is non-standard in tube-MPC and serves, essentially, the role of the reference tracking constraints in [34, 170].

Analogously to the inner OCP, the horizon is set to N for all agents but the cost for each agent is allowed to be different from that of its neighbours. Furthermore, the costs of the inner and outer OCPs for the same agent need not to be the same, however this might not be consistent since both OCPs are designed to control the same subsystem, for which a particular performance is sought.

5.3.3 DMPC algorithm architecture

A schematic of the inner-outer architecture is presented in Figure 5.1. The dashed lines represent communication between controllers and the solid lines communication between the plant and the agents (which involves measurement of local states and information of current optimal input to be applied). At each time instant each agent produces a reference trajectory by means of the inner

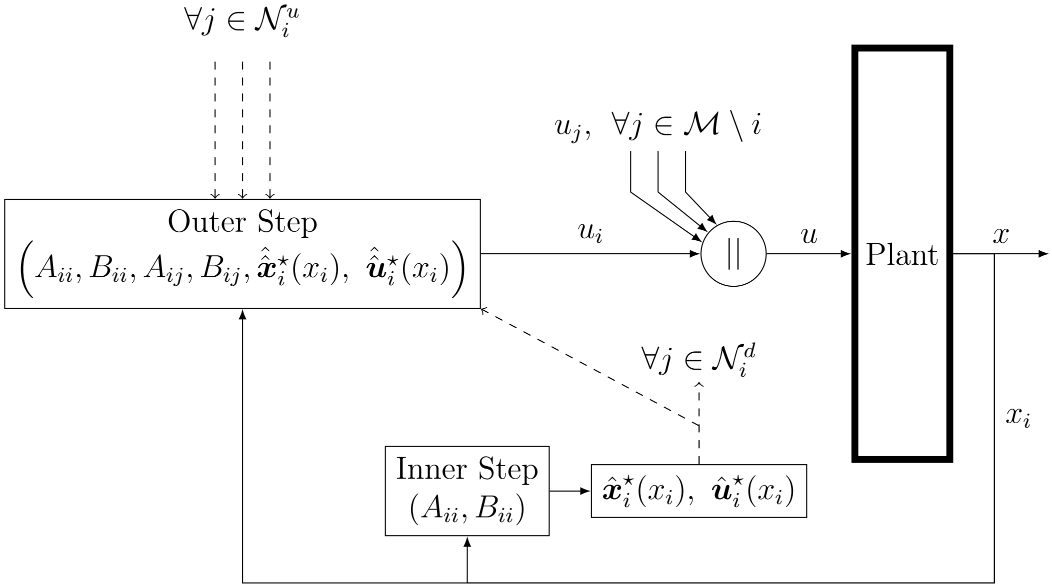

Figure 5.1: Architecture of the proposed DMPC algorithm (the symbol || stands for concatenation).

step of robust control. Subsequently, a single step of communication between agents takes place. Agent i informs all agents $j \in \mathcal{N}_i^d$ of its current nominal plan and receives information from all agents $j \in \mathcal{N}_i^u$ of their current nominal plans. Then, the outer step of robust control computes the true local control action to be fed to the plant taking into account the plans of neighbours.

The overall distributed control algorithm is now presented and in the subsequent sections it is shown how to design its several elements (including the sets $\mathbb{H}_i$) in order to guarantee a stable closed-loop and recursive feasibility of the related optimization problems.

Algorithm 1

1: $\lambda = 0$
2: Measure $\boldsymbol{x}_t$
3: **for** $i \in \mathcal{M}$ **do**
4: **if** $t = \lambda T$ **then**
5: Solve (5.5) and set $\left(\hat{\boldsymbol{u}}_i^s(x_i(t)), \hat{\boldsymbol{x}}_i^s(x_i(t))\right) = \left(\hat{\boldsymbol{u}}_i^\star(x_i(t)), \hat{\boldsymbol{x}}_i^\star(x_i(t))\right)$
6: $\lambda = \lambda + 1$
7: **else**
8: Set $\hat{\boldsymbol{x}}_i^s(x_i(t)) = \left\{\hat{x}_{i,1/t-1}, \dots, \hat{x}_{i,N/t-1}, \left(A_{ii} + B_{ii}\hat{K}_{f,i}\right)\hat{x}_{i,N/t-1}\right\}$ and $\hat{\boldsymbol{u}}_i^s(x_i(t)) = \left\{\hat{u}_{i,1/t-1}, \dots, \hat{u}_{i,N-1/t-1}, \hat{K}_{f,i}\hat{x}_{i,N/t-1}\right\}$
9: **end if**
10: **end for**
11: Broadcast $\left(\hat{\boldsymbol{u}}_i^s(x_i(t)), \hat{\boldsymbol{x}}_i^s(x_i(t))\right)$ among neighbours
12: **for** $i \in \mathcal{M}$ **do**
13: Solve (5.8)
14: Set $\hat{x}_i(t) = \hat{x}_{i,0/t}$ and $\hat{\hat{x}}_i(t) = \hat{\hat{x}}_{i,0}^\star(x_i(t))$
15: Set $u_i(t) = \hat{\hat{u}}_{i,0}^\star(x_i(t)) + \hat{\hat{K}}_i\left(x_i(t) - \hat{\hat{x}}_{i,0}^\star(x_i(t))\right)$ and apply to true plant
16: **end for**
17: Set $t = t + 1$ and go to step 2.

Steps 4–8 of Algorithm 1 define how the reference trajectories are updated. Each T time instants, the optimization (5.5) is solved and the trajectories are updated in its whole. At any other time instant, the reference trajectories are updated making use of the local feedback gain $\hat{K}_{f,i}$, which renders $\hat{\mathbb{X}}_{f,i}$ PI. If $T = 0$ the reference trajectories are updated only at initialization, resembling the update scheme in [34]. After the references have been computed and broadcast among neighbours, each agent solves (5.8) and computes the true control action to be fed to the plant following the standard disturbance rejection control law associated to the outer tube-based controller. Step 14 updates inner and outer nominal state trajectories for informative purposes. Note that the outer nominal state trajectories are updated with the optimized value obtained by solving the outer OCP (Step 14), as is standard in tube-based MPC controllers with optimizing trajectories. The inner state trajectories, on the other hand, are updated following the shared information, which will only match the optimized value for $t = \lambda T$ since at said times the inner OCP is actually solved. Otherwise the trajectories will be extended following a previously feasible sequence.

5.4 Design for feasibility and stability

A detailed account on the properties needed for several elements related to the outer step of robust control is now provided. The goal in the design of these elements is to to guarantee that Algorithm 1 is recursively feasible and that the control law defined in Step 15 results in a stable closed-loop. The focus is placed in the consistency constraint $\mathbb{H}_i$ in (5.8g) and in the terminal constraint set $\hat{\hat{\mathbb{X}}}_{f,i}$ in (5.8f). In what follows, a feasible but not necessarily optimal solution to the inner OCP (5.5) at time t is depicted as $\left(\hat{\boldsymbol{u}}_i^f(x_i(t)), \hat{\boldsymbol{x}}_i^f(x_i(t))\right)$ with

$$\hat{\boldsymbol{u}}_i^f(x_i(t)) = \left\{\hat{u}_{i,0}^f(x_i(t)), \ldots, \hat{u}_{i,N-1}^f(x_i(t))\right\} \tag{5.9a}$$

$$\hat{\boldsymbol{x}}_i^f(x_i(t)) = \left\{\hat{x}_{i,0}^f(x_i(t)), \ldots, \hat{x}_{i,N}^f(x_i(t))\right\}. \tag{5.9b}$$

Similarly, a feasible but not necessarily optimal solution to the outer OCP (5.8) at time t is $\left(\hat{\hat{\boldsymbol{u}}}_i^f(x_i(t)), \hat{\hat{\boldsymbol{x}}}_i^f(x_i(t))\right)$ with

$$\hat{\hat{\boldsymbol{u}}}_i^f(x_i(t)) = \left\{\hat{\hat{u}}_{i,0}^f(x_i(t)), \ldots, \hat{\hat{u}}_{i,N-1}^f(x_i(t))\right\} \tag{5.10a}$$

$$\hat{\hat{\boldsymbol{x}}}_i^f(x_i(t)) = \left\{\hat{\hat{x}}_{i,0}^f(x_i(t)), \ldots, \hat{\hat{x}}_{i,N}^f(x_i(t))\right\}. \tag{5.10b}$$

Finally, for any $t \neq \lambda T$ the information $\boldsymbol{f}_i(t)$ received by agent i is nothing more than an extension of $\boldsymbol{f}_i(t-1)$ following Step 8, hence such information will also be referred to as $\boldsymbol{f}_i^+(t-1)$.

5.4.1 Feasibility of the inner OCP

The frequency with which the inner OCP (5.5) is solved depends on the design parameter T. If $T = 1$ then (5.5) is solved at each time instance and hence it must be shown to be recursively feasible. On the other hand, if $T > 1$, the inner OCP will be solved only every T steps and hence its T-step feasibility must be ensured. The latter depends on the information extension procedure in Step 8, since this step effectively replaces the execution of the optimization and drives the inner nominal trajectories following Step 14. Nevertheless, recursive feasibility for any $T > 0$ can be guaranteed given recursive feasibility for $T = 1$ and Steps 8 and 14, which enforce a very particular update for the inner nominal dynamics.

Suppose then that at time $t = \lambda T$ the inner OCP is feasible with solution as in (5.9). Given the invariance of the terminal set $\hat{\mathbb{X}}_{f,i}$, it is easy to show that the candidate pair $\left(\hat{\boldsymbol{u}}_i^{\triangle}, \hat{x}_{i,0}^{\triangle}\right)$ defined by

$$\hat{\boldsymbol{u}}_i^{\triangle} = \left\{\hat{u}_{i,1}^f(x_i(t)), \ldots, \hat{u}_{i,N-1}^f(x_i(t)), \hat{K}_{f,i}\hat{x}_{i,N}^f(x_i(t))\right\} \tag{5.11a}$$

$$\hat{x}_{i,0}^{\triangle} = \hat{x}_{i,1}^f(x_i(t)) \tag{5.11b}$$

satisfies constraints (5.5c)–(5.5f) at time $t+1$. The RPI constraint (5.5b), however, is not necessarily met. This is because said constraint depends on the true state measured at time $t+1$, and the true plant is controlled following the disturbance rejection policy associated with outer OCP. Indeed, if (5.5) was feasible at time t then $z_i(t) = x_i(t) - \hat{x}_i(t) = x_i(t) - \hat{x}_{i,0}^f(x_i(t)) \in \mathbb{Z}_i$, but considering Step 15 of Algorithm 1 and the candidate solution (5.11), the future error is

$$\begin{aligned} z_i(t+1) &= A_{ii}z_i(t) + B_{ii}\left(u_i(t) - \hat{u}_{i,0}^f(x_i(t))\right) + w_i(t), \\ &= A_{ii}z_i(t) + B_{ii}\left(\hat{\hat{K}}_i s_i(t) + \left(\hat{u}_i(t) - \hat{u}_{i,0}^f(x_i(t))\right)\right) + w_i(t), \end{aligned}$$

which is not guaranteed to be inside $\mathbb{Z}_i$ (note $s_i(t) = x_i(t) - \hat{\hat{x}}_{i,0}^f(x_i(t))$). This is not a guarantee on infeasibility of the inner OCP, but it certainly means that the tail of a previously feasible solution (5.11) cannot be used as a candidate to provide a proof for recursive feasibility due to the RPI constraint (5.5b) not being met.

This issue could be overcome by simply adding a verification step to Algorithm 1 to allow the execution of Step 4 only when (5.5) is feasible, but there would be no guarantee this would ever happen after a feasible initialization. Alternatively, the consistency constraint (5.8g) can be employed to guarantee recursive feasibility of the inner OCP.

Assumption 5.3. The consistency constraint set $\mathbb{H}_i$ is such that $\mathbb{H}_i \subseteq \mathbb{Z}_i \ominus \mathbb{S}_i$

Proposition 5.1. Assume the outer OCP is feasible at time t and that Assumption 5.3 holds. If the inner OCP (5.5) is feasible at time t, then it is also feasible at time $t+1$.

Proof. Indeed, assume that the inner and outer OCPs are feasible at time t with solutions as in (5.9) and (5.10) respectively and so the system is controlled at time t following Steps 15 and 14 of Algorithm 1. It follows that if (5.11) is used as a candidate solution for the inner OCP at time $t+1$, then

$$z_i(t+1) = x_i(t+1) - \hat{x}_i(t+1) = x_i(t+1) - \hat{x}_{i,1}^f(x_i(t)) \tag{5.12a}$$

$$= x_i(t+1) + \left(\hat{\hat{x}}^f_{i,1}(x_i(t)) - \hat{\hat{x}}^f_{i,1}(x_i(t))\right) - \hat{x}^f_{i,1}(x_i(t)) \tag{5.12b}$$

$$= \left(x_i(t+1) - \hat{\hat{x}}^f_{i,1}(x_i(t))\right) + \left(\hat{\hat{x}}^f_{i,1}(x_i(t)) - \hat{x}^f_{i,1}(x_i(t))\right) \tag{5.12c}$$

$$\in \mathbb{S}_i \oplus \mathbb{H}_i \tag{5.12d}$$

where the first equality follows from the definition of the inner trajectory error and the candidate solution (5.11). The set inclusion in (5.12d) then follows from the robust invariance of $\mathbb{S}_i$, the implementation of the associated control law in Step 15 of Algorithm 1, and the feasibility of constraint (5.8g) given a feasible outer OCP at time t. It follows then from Assumption 5.3 that $z_i(t+1) \in \mathbb{Z}_i$, while feasibility of the remaining constraints follows from the standard proof of recursive feasibility for tube-based MPC implementations [1, 2], and so the candidate (5.11) is a feasible solution for the inner OCP at time $t+1$. ■

The following corollary to Proposition 5.1 establishes recursive feasibility.

Corollary 5.1. Suppose that Assumption 5.3 holds. If the inner OCP is feasible at time $t = 0$ and the outer OCP is feasible at time $t-1$, then the inner OPC is feasible at time t.

Proof. At times $t \neq \lambda T$ the shared information is constructed by extending the information shared at the last time instant with the terminal controller of the inner OCP (Step 8 of Algorithm 1). Proposition 5.1 shows that, given a purposely defined consistency constraint of the outer OCP, such an extension results in a feasible solution for the inner OCP. Step 14 of Algorithm 1 updates the inner nominal trajectories with the shared information for $t \neq \lambda T$. It follows then that feasibility of the inner OCP at time $t = \lambda T$ implies feasibility at time $t = (\lambda+1)T$ provided the outer OCP was feasible throughout. It follows then that feasibility of the inner OCP at time $t = 0$ implies feasibility at all $t > 0$, again, provided the outer OCP was feasible until $t-1$. ■

A few remarks are in order. First note that solving the outer OCP at time $t+1$ could result in that $\hat{\hat{x}}^\star_{i,0}(x_i(t+1)) \neq \hat{\hat{x}}^f_{i,1}(x_i(t))$. This is not an issue, however, since (5.12) does not depend on the optimal trajectories matching at subsequent time instances, but only on the existence of an element $\hat{\hat{x}} \in \hat{\hat{\mathbb{X}}}_i$ such that $x_i(t+1) - \hat{\hat{x}}_i \in \mathbb{S}_i$ and $\hat{\hat{x}}_i - \hat{x}^\star_{i,1}(x_i(t)) \in \mathbb{H}_i$; in this case such element is $\hat{\hat{x}}^f_{i,1}(x_i(t))$.

It is also worth noting that Assumption 5.3 can only be verified if $\mathbb{S}_i \subseteq \text{int}\,(\mathbb{Z})_i$, however this is not a stringent demand. Indeed, the RPI sets computed in the context of tube-based MPC controllers are usually (approximations of the) minimal RPI set. Given then that Assumption 5.2 requires the disturbance levels affecting the outer OCP to be smaller when compared to those of the inner OCP, it is expected that $\mathbb{S}_i \subseteq \mathbb{Z}_i$. In fact, a suitable $\mathbb{S}_i$ could be computed from $\mathbb{Z}_i$, by scaling, or using the method in [101].

Finally, note that Proposition 5.1 guarantees feasibility of the inner OCP at time $t+1$ given feasibility of the outer OCP at time t, but it will be shown later that the latter can be established from feasibility of the inner OCP at time t, which is an assumption of Proposition 5.1 as well.

5.4.2 Feasibility of the outer OCP

The outer OCP is solved at each time instant independently of the frequency at which the inner OCP is solved, hence it is necessary to show its feasibility at every time instant given the communication approach proposed in Algorithm 1. Indeed, the feasibility region of the outer OCP depends on the consistency constraint (5.8g) and the prediction model (5.8c) and hence is parametrized by the information received from neighbours. It makes a difference then, whether this is an entirely new set of predictions as in Step 5 of Algorithm 1, or an extended version of previously received information, as in Step 8.

5.4.2.1 Recursive feasibility for $t+1 \neq \lambda T$

The case in which $\boldsymbol{f}_i(t+1) = \boldsymbol{f}_i^+(t)$ is analysed first. Suppose then that at time t the outer controller receives some information about its neighbours plans (whether this is completely new or an extension of that received at time $t+1$) and that the outer OCP is feasible with solution as in (5.10). It is natural then to attempt to show that the candidate pair $\left(\hat{\hat{\boldsymbol{u}}}_i^{\triangle}, \hat{\hat{x}}_{i,0}^{\triangle}\right)$ defined by

$$\hat{\hat{\boldsymbol{u}}}_i^{\triangle} = \left\{\hat{\hat{u}}_{i,1}^f(x_i(t)), \dots, \hat{\hat{u}}_{i,N-1}^f(x_i(t)), \hat{K}_{f,i}\hat{\hat{x}}_{i,N}^f(x_i(t))\right\} \tag{5.13a}$$

$$\hat{\hat{x}}_{i,0}^{\triangle} = \hat{\hat{x}}_{i,1}^f(x_i(t)) \tag{5.13b}$$

satisfies all constraints at time $t+1$, as it was done with the inner OCP, yet this is not necessary the case. The main obstacles are in the satisfaction of the terminal constraint and of the consistency constraint at the last prediction time, which is why the properties of both sets have been left unstated until now. Define the error $e_i(t) = \hat{\hat{x}}_i(t) - \hat{x}_i(t)$, the set

$$\mathbb{D}_{f,i} = \bigoplus_{j \in \mathcal{N}_i^u} \left(A_{ij} + B_{ij}\hat{K}_{f,j}\right)\hat{\mathbb{X}}_{f,j}, \tag{5.14}$$

and consider the following assumptions.

Assumption 5.4. The terminal constraint set of the outer OCP is an RPI set for the closed-loop dynamics $\left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)$ with respect to disturbances bounded in $\mathbb{D}_{f,i}$ and admissible with respect to tightened constraint sets $\hat{\hat{\mathbb{X}}}_i$ and $\hat{\hat{\mathbb{U}}}_i$.

Assumption 5.5. There exists a set $\mathbb{H}_{f,i} \subseteq \mathbb{H}_i$ which is RPI for $\left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)$ with respect to disturbances bounded in $\hat{\mathbb{D}}_{f,i} = B_{ii}\left(\hat{\hat{K}}_{f,i} - \hat{K}_{f,i}\right)\hat{\mathbb{X}}_{f,i} \oplus \mathbb{D}_{f,i}$.

In view of such design conditions, the following can be stated about the feasibility of the outer OCP at time $t+1 \neq \lambda T$.

Proposition 5.2. Suppose that Assumptions 5.4 and 5.5 hold and that

$$e_{i,N} = \hat{\hat{x}}_{i,N} - \hat{x}_{i,N/t} \in \mathbb{H}_{f,i} \subseteq \mathbb{H}_i \tag{5.15}$$

is added as a constraint to the outer OCP (5.8). If the modified outer OCP is feasible at time t with solution as in (5.10) and $\boldsymbol{f}_i(t+1) = \boldsymbol{f}_i^+(t)$, then (5.13) is a feasible solution for the outer OCP at time $t+1$.

Proof. For the RPI constraint (5.8b) note that if (5.8) was feasible at time t then $s_i(t) = x_i(t) - \hat{\hat{x}}_i(t) = x_i(t) - \hat{\hat{x}}_{i,0}^f(x_i(t)) \in \mathbb{S}_i$, and considering Step 15 of Algorithm 1 and the candidate solution (5.11), the future error is

$$s_i(t+1) = A_{ii}s_i(t) + B_{ii}\hat{\hat{K}}_i s_i(t) + v_i(t),$$

which is guaranteed to be inside $\mathbb{S}_i$. It is then straightforward to show that $\hat{\hat{x}}_{i,k}^{\triangle} = \hat{\hat{x}}_{i,k+1}^f(x_i(t)) \in \hat{\hat{\mathbb{X}}}_i$ for all $k \in [0, N-1]$ and that $\hat{\hat{u}}_{i,k}^{\triangle} = \hat{\hat{u}}_{i,k+1}^f(x_i(t)) \in \hat{\hat{\mathbb{U}}}_i$ for all $k \in [0, N-2]$. Furthermore, since $\boldsymbol{f}_i(t+1) = \boldsymbol{f}_i^+(t)$ it also holds that $\hat{\hat{x}}_{i,k}^{\triangle} - \hat{x}_{i,k/t+1} \in \mathbb{H}_i$ for all $k \in [0, N-1]$, so it is only left to show that

$$\begin{aligned} \hat{\hat{u}}_{i,N-1}^{\triangle} &\in \hat{\hat{\mathbb{U}}}_i \\ \hat{\hat{x}}_{i,N}^{\triangle} &\in \hat{\hat{\mathbb{X}}}_{f,i} \\ \hat{\hat{x}}_{i,N}^{\triangle} - \hat{x}_{i,N/t+1} &\in \mathbb{H}_{f,i}. \end{aligned}$$

The candidate terminal control input in (5.13) is $\hat{\hat{u}}_{i,N-1}^{\triangle} = \hat{\hat{K}}_{f,i}\hat{\hat{x}}_{i,N}^f(x_i(t))$ which by Assumption 5.4 and feasibility of the outer OCP at time t guarantees $\hat{\hat{u}}_{i,N-1}^{\triangle} \in \hat{\hat{K}}_{f,i}\hat{\hat{\mathbb{X}}}_{f,i} \subseteq \hat{\hat{\mathbb{U}}}_i$. Following the dynamics in (5.8c), and the fact that

$\boldsymbol{f}_i(t+1) = \boldsymbol{f}_i^+(t)$, it is easy to show that

$$\hat{\hat{x}}_{i,N}^{\triangle} = \left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\hat{\hat{x}}_{i,N-1}^{\triangle} + d_{i,N-1/t+1} = \left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\hat{\hat{x}}_{i,N}^{f} + d_{i,N/t},$$

with

$$d_{i,N/t} = \sum_{j\in\mathcal{N}_i^u}\left(A_{ij} + B_{ij}\hat{K}_{f,j}\right)\hat{x}_{j,N/t} \in \mathbb{D}_{f,i}, \tag{5.16}$$

which again results in $\hat{\hat{x}}_{i,N}^{\triangle} \in \hat{\hat{\mathbb{X}}}_{f,i} \subseteq \hat{\hat{\mathbb{X}}}_i$, given Assumption 5.4 and feasibility of the outer OCP at time t. The latter also guarantees $e_{i,N-1}^{\triangle} = \hat{\hat{x}}_{i,N-1}^{\triangle} - \hat{x}_{i,N-1/t+1} = \hat{\hat{x}}_{i,N}^{f}(x_i(t)) - \hat{x}_{i,N/t} \in \mathbb{H}_{f,i}$ and since

$$e_{i,N}^{\triangle} = \left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\hat{\hat{x}}_{i,N}^{f}(x_i(t)) + d_{i,N/t} - \left(A_{ii} + B_{ii}\hat{K}_{f,i}\right)\hat{x}_{i,N/t} \tag{5.17a}$$

$$= \left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)e_{i,N-1}^{\triangle} + B_{ii}\left(\hat{\hat{K}}_{f,i} - \hat{K}_{f,i}\right)\hat{x}_{i,N/t} + d_{i,N/t} \tag{5.17b}$$

then by Assumption 5.5 it holds that $e_{i,N}^{\triangle} \in \mathbb{H}_{f,i}$ and all constraints of the outer OCP are met at time $t+1$ by the candidate solution (5.13). ■

Note that the issue of guaranteeing feasibility of the consistency constraint is addressed in Assumption 5.5 by including an additional terminal constraint on the outer OCP. This makes sense given that the original terminal constraint is there to guarantee that state constraints are recursively met and that the consistency constraint is nothing more than another state constraint. Furthermore, the additional requirement on robustness over the the terminal constraint set $\hat{\hat{\mathbb{X}}}_{f,i}$ is similar to the requirement in [34] that demands the collection of terminal constraint sets to form a PI set for the overall network. Indeed, it can be shown that each element of such a collection is in fact an RPI set for the local dynamics (Section 5.5 provides a detailed discussion on this matter).

In summary then, feasibility of the outer OCP at time $t+1$ can be guaranteed given a feasible solution at time t exists and that the information received at time $t+1$ is only an extension of that received at the last time instant. The following corollary to Proposition 5.2 ensues.

Corollary 5.2. Suppose that Assumptions 5.4 and 5.5 hold and that (5.15) is added as a constraint to the outer OCP (5.8). If the inner and outer OCPs are feasible at time $t = \lambda T$, then Algorithm 1 is feasible for all $\tilde{t} \in [t, (\lambda+1)T - 1]$.

Proof. Follows straightforwardly from Proposition 5.2 and Corollary 5.1. ■

5.4.2.2 Recursive feasibility for $t = \lambda T$

As soon as $t = \lambda T$, however, $\boldsymbol{f}_i(t)$ is not necessarily equal to $\boldsymbol{f}_i^+(t-1)$, and hence feasibility of the outer OCP cannot be guaranteed through Proposition 5.2. A simple workaround would be to implement a feasibility check step in Algorithm 1 in order to verify whether the new information allows for feasibility of the outer OCP. If not, $\boldsymbol{f}_i(t)$ could be discarded and both inner and outer OCPs would fall back to $\boldsymbol{f}_i^+(t-1)$, i.e. Step 8 in Algorithm 1. However, there is no a-priori guarantee on whether the outer OCP would ever be feasible under a complete revamp of the nominal reference trajectories, hence $\boldsymbol{f}_i(\lambda T)$ might be rejected for all $\lambda > 0$ and the initially optimized nominal sequence would remain as a reference for all time steps (similar to the approach in [34], albeit without an arbitrary initialization).

Instead, it is proposed here to modify the RPI constraint of the inner OCP (5.5b) in order to guarantee that a feasible solution will exist for the outer OCP at every time instant the former is solved. This shows the symbiosis that exists between both steps of robust control: the consistency constraint of the outer OCP was designed to guarantee backwards recursive feasibility while now the RPI constraint of the inner OCP will be modified to guarantee forward recursive feasibility. Before, however, a result found in [178] is necessary.

Lemma 5.1. Consider three non-empty convex and closed sets $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{C}$ in $\mathbb{R}^n$. It holds that $\mathcal{A} \oplus (\mathcal{B} \ominus \mathcal{C}) \subseteq (\mathcal{A} \oplus \mathcal{B}) \ominus \mathcal{C}$.

Proposition 5.3. Assume the RPI constraint of the inner OCP (5.5b) is modified to

$$x_i(t) - \hat{x}_{i,0} \in \mathbb{S}_i \oplus \mathbb{H}_i \tag{5.18}$$

with $\mathbb{H}_i$ satisfying Assumption 5.3. If at time t the inner OCP is feasible with solution as in (5.9), then there exists $s_i \in \mathbb{S}_i$ and $e_i \in \mathbb{H}_i$ such that $z_i(t) = x_i(t) - \hat{x}_{i,0}^f(x_i(t)) = s_i + e_i$. Furthermore, there exists $\hat{\hat{x}}_i \in \hat{\hat{\mathbb{X}}}_i$ such that

$$e_i = \hat{\hat{x}}_i - \hat{x}_{i,0}^f(x_i(t)) \in \mathbb{H}_i \tag{5.19a}$$

$$s_i = x_i(t) - \hat{\hat{x}}_i \in \mathbb{S}_i. \tag{5.19b}$$

Proof. For the first part note that since the inner OCP with modified RPI constraint is feasible at time t, the error $z_i(t)$ is contained inside $\mathbb{S}_i \oplus \mathbb{H}_i$. The error vector can then be partitioned where it intersects the first of both sets to obtain scalars $\alpha, \beta > 0$ such that $s_i = \alpha z_i \in \mathbb{S}_i$ and $e_i = \beta z_i \in \mathbb{H}_i$ with $\alpha + \beta = 1$. It follows then that $s_i + e_i = z_i(t)$. For the second part define

$\hat{\hat{x}}_i^e = \beta x_i(t) + (1-\beta)\,\hat{x}_{i,0}^f(x_i(t))$ and $\hat{\hat{x}}_i^s = (1-\alpha)\,x_i(t) + \alpha\hat{x}_{i,0}^f(x_i(t))$. It follows that $e_i = \hat{\hat{x}}_i^e - \hat{x}_{i,0}^f(x_i(t))$ and that $s_i = x_i(t) - \hat{\hat{x}}_i^s$. Furthermore, it holds that $\hat{\hat{x}}_i^e = \hat{\hat{x}}_i^s = \hat{\hat{x}}_i$ and so (5.19) holds. Finally, from (5.19a) it holds that

$$\hat{\hat{x}}_i \in \left\{\hat{x}_{i,0}^f(x_i(t))\right\} \oplus \mathbb{H}_i \subseteq \hat{\mathbb{X}}_i \oplus \mathbb{H}_i \tag{5.20a}$$

$$\implies \hat{\hat{x}}_i \in \hat{\mathbb{X}}_i \oplus (\mathbb{Z}_i \ominus \mathbb{S}_i) \tag{5.20b}$$

$$\implies \hat{\hat{x}}_i \in \left(\hat{\mathbb{X}}_i \oplus \mathbb{Z}_i\right) \ominus \mathbb{S}_i \tag{5.20c}$$

$$\implies \hat{\hat{x}}_i \in \mathbb{X}_i \ominus \mathbb{S}_i \tag{5.20d}$$

$$\implies \hat{\hat{x}}_i \in \hat{\hat{\mathbb{X}}}_i \tag{5.20e}$$

where (5.20b) follows by Assumption 5.3, (5.20c) from Lemma 5.1, (5.20d) from the tightening of the inner OCP and (5.20e) from the definition of the tightened state constraint set $\hat{\hat{\mathbb{X}}}_i$. ■

Note that $\mathbb{S}_i \oplus \mathbb{H}_i$ is not necessarily RPI for $\left(A_{ii} + B_{ii}\hat{K}_i\right)$ as $\mathbb{Z}_i$ is, yet Proposition 5.1 shows that the candidate solution (5.11) results in $z_i(t+1) \in \mathbb{S}_i \oplus \mathbb{H}_i$. It follows then that modifying the RPI constraints of the inner OCP to (5.18) does not break the feasibility guarantee provided by Proposition 5.1. In view of this, and the following assumption, feasibility of the outer OCP at time $t = \lambda T$ can be guaranteed.

Assumption 5.6. The set $\mathbb{H}_i$ and the gain $\bar{K}_i$ (not necessarily equal to $\hat{K}_i$ or $\hat{\hat{K}}_i$) are designed such that

$$\mathbb{H}_i \subseteq \mathbb{Z}_i \ominus \mathbb{S}_i \tag{5.21a}$$

$$\left(A_{ii} + B_{ii}\bar{K}_i\right)\mathbb{H}_i \oplus \mathbb{D}_i \subseteq \mathbb{H}_i \tag{5.21b}$$

$$\hat{\mathbb{U}}_i \oplus \bar{K}\mathbb{H}_i \subseteq \hat{\hat{\mathbb{U}}}_i \tag{5.21c}$$

$$\mathbb{H}_i \oplus \hat{\mathbb{X}}_{f,i} \subseteq \hat{\hat{\mathbb{X}}}_{f,i} \tag{5.21d}$$

with

$$\mathbb{D}_i = \bigoplus_{j \in \mathcal{N}_i^u} \left(A_{ij}\hat{\mathbb{X}}_j \oplus B_{ij}\hat{\mathbb{U}}_j\right).$$

Proposition 5.4. Assume that the RPI constraint of the inner OCP (5.5b) is modified to (5.18) and that Assumption 5.6 holds. If the inner OCP is feasible at time t with solution as in (5.9), then the pair $\left(\hat{\hat{\boldsymbol{u}}}_i^\triangle, \hat{\hat{x}}_{i,0}^\triangle\right)$ defined by

$$\hat{\hat{x}}_{i,0}^\triangle = (1-\alpha)x_i(t) + \alpha\hat{x}_{i,0}^f(x_i(t)) \tag{5.22a}$$

$$\hat{\hat{\boldsymbol{u}}}_i^{\triangle} = \left\{\hat{\hat{u}}_{i,0}^{\triangle}, \ldots, \hat{\hat{u}}_{i,N-1}^{\triangle}\right\} \tag{5.22b}$$

$$\hat{\hat{u}}_{i,k}^{\triangle} = \hat{u}_{i,k}^{f}(x_i(t)) + \bar{K}_i\left(\hat{\hat{x}}_{i,k} - \hat{x}_{i,k}^{f}(x_i(t))\right), \tag{5.22c}$$

is a feasible solution to the outer OCP at time $t+1$.

Proof. In view of (5.21a) and a feasible inner OCP with modified RPI constraint, Proposition 5.3 guarantees that $\hat{\hat{x}}_{i,0}^{\triangle}$ in (5.22a) is contained in $\hat{\hat{\mathbb{X}}}_i$ and is such that

$$e_{i,0} = \hat{\hat{x}}_{i,0}^{\triangle} - \hat{x}_{i,0}^{f}(x_i(t)) \in \mathbb{H}_i \tag{5.23a}$$

$$s_{i,0} = x_i(t) - \hat{\hat{x}}_{i,0}^{\triangle} \in \mathbb{S}_i. \tag{5.23b}$$

The latter guarantees that the RPI constraint (5.8b) is met, while the former ensures that the initial error between both nominal trajectories is contained inside $\mathbb{H}_i$. By employing the proposed control sequence (5.22) it holds that

$$e_{i,k+1} = \left(A_{ii} + B_{ii}\bar{K}_i\right) e_{i,k} + d_{i,k/t}, \tag{5.24}$$

for all $k \in [0, N-1]$, and so the consistency constraint (5.8g) is met throughout the horizon by (5.21b) and (5.23a). From (5.24) it follows that $\hat{\hat{x}}_{i,k/t} \in \hat{\hat{\mathbb{X}}}_i \oplus \mathbb{H}_i$ for all $k \in [0, N-1]$ and so from Lemma 5.1 the tightened state constraint (5.8d) is also met throughout the horizon. From (5.22) it follows that $\hat{\hat{u}}_{i,k/t}^{\triangle} \in \hat{\hat{\mathbb{U}}}_i \oplus \bar{K}_i\mathbb{H}_i$ which implies the tightened input constraints are also met given (5.21c). Finally, (5.24) implies that $\hat{\hat{x}}_{i,N/t} \in \mathbb{H}_i \oplus \hat{\hat{\mathbb{X}}}_{f,i}$ which by (5.21d) in Assumption 5.6 guarantees that the terminal constraint of the outer OCP (5.8f) is also met, and hence (5.22) represents a feasible solution for the outer OCP at time $t+1$. ∎

Proposition 5.4 guarantees feasibility of the outer OCP at time t independent of whether the information shared is entirely new or an extension of past data, albeit by resorting to a feasible solution of a tightened inner OCP at time t, rather than a solution to the outer OCP at time $t-1$ as is usual in MPC implementations. In view of this the standard approach to guarantee that the optimal cost function is a Lyapunov function for the closed-loop trajectories will not be applicable. Proposition 5.2 does allow for such an approach but it is valid only for $t \neq \lambda T$. In order to combine both approaches into a single design procedure, and to pave the way for an stability guarantee, consider the following assumption.

Assumption 5.7. Assume $\bar{K} = \hat{\hat{K}}_{f,i}$ with $\hat{\hat{K}}_{f,i}\hat{\hat{\mathbb{X}}}_{f,i} \subseteq \hat{\hat{\mathbb{U}}}_i$, and that $\hat{\mathbb{D}}_{f,i}$ from Assumption 5.5 is a subset of $\mathbb{D}_i$.

The following corollary establishes the link between both recursive feasibility propositions.

Corollary 5.3. Assume that the RPI constraint of the inner OCP (5.5b) is modified to (5.18) and that Assumptions 5.6 and 5.7 hold. If the outer OCP is feasible at time t with solution as in (5.10) and $\boldsymbol{f}_i(t+1) = \boldsymbol{f}_i^+(t)$, then (5.13) is a feasible solution for the outer OCP at time $t+1$.

Proof. The proof follows from the proof of Proposition 5.2 and by noting that Assumptions 5.6 and 5.7 supersede Assumption 5.5 and hence render unnecessary the additional terminal constraint (5.15). Indeed, the first part of the proof to Proposition 5.2 holds unchanged so it is left to show that

$$\begin{aligned} \hat{\hat{u}}^{\triangle}_{i,N-1} &\in \hat{\hat{\mathbb{U}}}_i \\ \hat{\hat{x}}^{\triangle}_{i,N} &\in \hat{\hat{\mathbb{X}}}_{f,i} \\ \hat{\hat{x}}^{\triangle}_{i,N} - \hat{x}_{i,N/t+1} &\in \mathbb{H}_i. \end{aligned}$$

holds. Again, the candidate terminal control input in (5.13) is $\hat{\hat{u}}^{\triangle}_{i,N-1} = \hat{\hat{K}}_{f,i}\hat{\hat{x}}^{f}_{i,N}(x_i(t))$ which by Assumption 5.7 and feasibility of the outer OCP at time t guarantees $\hat{\hat{u}}^{\triangle}_{i,N-1} \in \hat{\hat{K}}_{f,i}\hat{\hat{\mathbb{X}}}_{f,i} \subseteq \hat{\hat{\mathbb{U}}}_i$. Feasibility of the outer OCP at time t also guarantees $e^{\triangle}_{i,N-1} = \hat{\hat{x}}^{\triangle}_{i,N-1} - \hat{x}_{i,N-1/t+1} = \hat{\hat{x}}^{f}_{i,N}(x_i(t)) - \hat{x}_{i,N/t} \in \mathbb{H}_i$ and since (5.17) still holds it follows that

$$\begin{aligned} e^{\triangle}_{i,N} \quad &\in \left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\mathbb{H}_i \oplus B_{ii}\left(\hat{\hat{K}}_{f,i} - \hat{K}_{f,i}\right)\hat{\hat{\mathbb{X}}}_{f,i} \oplus \mathbb{D}_{f,i} \\ &\in \left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\mathbb{H}_i \oplus \hat{\mathbb{D}}_{f,i} \\ \Longrightarrow e^{\triangle}_{i,N} \quad &\in \left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\mathbb{H}_i \oplus \mathbb{D}_i \end{aligned}$$

where the implication follows from Assumption 5.7, and so $e^{\triangle}_{i,N} \in \mathbb{H}_i$ by Assumption 5.6, rendering the additional terminal constraint unnecessary. Finally, note that $\hat{\hat{x}}^{\triangle}_{i,N} = e^{\triangle}_{i,N} + \hat{x}^{f}_{i,N/t}$ and so $\hat{\hat{x}}^{\triangle}_{i,N} \in \mathbb{H}_i \oplus \hat{\mathbb{X}}_{f,i}$ which implies $\hat{\hat{x}}^{\triangle}_{i,N} \in \hat{\hat{\mathbb{X}}}_{f,i}$ by Assumption 5.6. ■

In summary, the design conditions in Assumptions 5.6 and 5.7 guarantee that the outer OCP is feasible at any time instant given a feasible inner OCP at time $t = 0$ and the information sharing procedure defined in Algorithm 1. The trade-off, when compared to the conditions in Assumption 5.5, is a considerably more complex design of the controller parameters (particularly of the set $\mathbb{H}_i$, which not only has to fulfil the inclusion (5.21a) but also conditions (5.21b)–(5.21d). This is opposed to the requirements of Proposition 5.2 which allow $\mathbb{H}_i$ to be fairly arbitrary, provided the inclusion of an additional terminal constraint defined by (5.15). Oppositely, Assumptions 5.6 and 5.7 place no explicit invariance requirements over the terminal constraint set $\hat{\hat{\mathbb{X}}}_{f,i}$, however (5.21d) places an implicit one, by forcing it to be equal or larger than a sum of two invariant sets.

Note also that, although Assumption 5.6 requires $\mathbb{H}_i$ to be an RPI set for a given gain $\bar{K}_i$, it does not necessarily force $\hat{\hat{u}}_{i,k}$ to follow (5.22), since there exists infinitely many other invariant inducing control actions for a given $e_{i,k} \in \mathbb{H}_i$. The law in (5.22) provides nothing more than a feasible control action, but the optimizer is free to chose a more efficient one. It is also interesting to note that, if (5.22) is employed, the overall control action is

$$u_i(t) = \hat{u}_i(t) + \bar{K}\left(x_i(t) - \hat{x}_i(t)\right) + \hat{\bar{K}}\left(x_i(t) - \hat{\hat{x}}_i(t)\right),$$

which equates the structure of the control law in [33] given a particular choice of gains. Ultimately this implies that the control law proposed in [33] can be seen as a feasible, yet not necessarily optimal solution to the distributed controller proposed here.

Nevertheless, the design conditions required to meet Assumptions 5.6 and 5.7 could be difficult to meet. In this case, as previously discussed, a verification step could be included in the algorithm such that $\boldsymbol{f}_i(t)$ is discarded every time it renders the outer OCP infeasible. This would result in that Proposition 5.2 is enough to guarantee recursive feasibility of Algorithm 1 and the design requirements over the set $\mathbb{H}_i$ are greatly simplified.

5.4.2.3 Computation of the outer uncertainty set

The recursive feasibility guarantees for the outer OCP rely on Assumption 5.6 which can only be met if $\mathbb{S}_i \subset \operatorname{int}(\mathbb{Z})_i$. The latter, as previously discussed, is not necessarily too demanding given Assumption 5.2, however the computation of an appropriate set $\mathbb{V}_i$ such that it is contained in the interior of $\mathbb{W}_i$ has not yet been discussed. The reason for this is that the set $\mathbb{V}_i$ can be computed comprehensively (as opposed to arbitrarily as in [33, 34]) only after the recursive feasibility of Algorithm 1, which depends on Assumption 5.2, has been established.

Indeed, from (5.7) it follows that

$$v_{i,k/t} = \sum_{j\in\mathcal{N}_i^u} \left(A_{ij}\left(x_j(t+k) - \hat{x}_{j,k/t}\right) + B_{ij}\left(u_j(t+k) - \hat{u}_{j,k/t}\right)\right),$$

so in order to find $\mathbb{V}_i$ it is necessary to find sets that bound $\left(x_j(t+k) - \hat{x}_{j,k/t}\right)$ and $\left(u_j(t+k) - \hat{u}_{j,k/t}\right)$. If Assumptions 5.2, 5.6 and 5.7 hold, recursive feasibility of the inner OCP can be guaranteed, and hence it holds that $x_j(t) - \hat{x}_j(t) \in \mathbb{S}_j \oplus \mathbb{H}_j$ for all $t \geq 0$. Bounding the difference between the control actions, however, is not that simple. Again, recursive feasibility of the RPI constraint of the inner OCP must mean that $u_j(t) - \hat{u}_j(t)$ is invariant inducing for the set $\mathbb{S}_j \oplus \mathbb{H}_j$, but not necessarily through the linear feedback gain $\hat{K}_j$.

In fact, the true control action is driven by the tube law associated to the outer OCP and so

$$u_j(t) - \hat{u}_j(t) = \hat{\hat{u}}_j(t) - \hat{u}_j(t) + \hat{\hat{K}}_i s_j(t). \tag{5.25}$$

Again, given recursive feasibility of outer OCP, and in particular of the consistency constraint that bounds the error between both nominal trajectories to lie inside $\mathbb{H}_j$, the difference $\hat{\hat{u}}_j(t) - \hat{u}_j(t)$ has to be invariant inducing for said set, yet not necessarily following (5.22c).

Define the set of all (admissible) invariant inducing control actions for the set $\mathbb{H}_j$ as $\mathbb{U}_j(\mathbb{H}_j \oplus \mathbb{S}_j)$. It is, in general, not a trivial task to characterize such a set, nevertheless it follows from (5.25) that $\mathbb{U}_j(\mathbb{H}_j \oplus \mathbb{S}_j) \subseteq \tilde{\mathbb{U}}_j$ with

$$\tilde{\mathbb{U}}_j = \hat{\hat{\mathbb{U}}}_j \oplus \hat{\hat{K}}_j \mathbb{S}_j \oplus \hat{\mathbb{U}}_j^-,$$

where $\hat{\mathbb{U}}_i^-$ is the reflection of $\hat{\mathbb{U}}_i$ through the origin. The sets $\mathbb{V}_i$ can then be outer approximated by each agent using the sets $\mathbb{S}_j$, $\mathbb{H}_j$, $\hat{\hat{\mathbb{U}}}_j$ and $\tilde{\mathbb{U}}_j$. However, the first three sets actually depend on $\mathbb{V}_i$. In order to overcome this without resorting to an iterative design procedure as in [33], note that by Assumption 5.6 $\mathbb{S}_j \oplus \mathbb{H}_j \subseteq \mathbb{Z}_i$ and by definition of the tightening $\hat{\hat{\mathbb{U}}}_j \oplus \hat{\hat{K}}_j \mathbb{S}_j \subseteq \mathbb{U}_j$. It follows that if

$$\mathbb{V}_i = \bigoplus_{j\in\mathcal{N}_i^u} \left(A_{ij}\mathbb{Z}_j \oplus B_{ij}\left(\mathbb{U}_j \oplus \hat{\mathbb{U}}_j^-\right)\right) \tag{5.26}$$

then $v_{i,k/t} \in \mathbb{V}_i$ for all $k, t > 0$.

It is expected, however, that $u_j(t) - \hat{u}_j(t)$ is contained in the interior of $\mathbb{U}_j \oplus \hat{\mathbb{U}}_j^-$, making (5.26) a conservative estimate of $\mathbb{V}_i$. If $\mathbb{V}_i$ computed as in (5.26) does not meet Assumption 5.2 it is up to the designer to modify the tube gain $\hat{K}_j$ in order to obtain a smaller set $\mathbb{Z}_j$ or $\hat{\mathbb{U}}_j^-$, depending on which is stronger, the input or state coupling. Nevertheless, there might not exist a linear gain that results in Assumption 5.2 being met given that the bound on the input deviation $\mathbb{U}_j \oplus \hat{\mathbb{U}}_j^-$ is conservative. Therefore, although simple, the definition of $\mathbb{V}_i$ as in (5.26) is mostly valid for networks with negligible (or null) input coupling.

5.4.3 Stability

Similarly to Section 2.3.1, stability of the origin for the closed-loop can be achieved given a proper design of the cost function, and particularly of the terminal cost. In the following, as discussed previously, it is assumed that the cost functions of the inner and outer OCPs can be different, although consistency might demand them to be equal. Furthermore, it is assumed that:

- The collection of gains $\hat{K}_i$ and $\hat{\hat{K}}_i$ fulfil Assumption 5.1.
- The inner OCP is (5.5) with RPI constraint (5.5b) modified to (5.18).
- The outer OCP is (5.8).
- Assumptions 5.2, 5.6 and 5.7 hold.

In view of the above Algorithm 1, which drives the execution of the inner and outer OCPs, is guaranteed to be recursively feasible. Furthermore, the state trajectories are guaranteed to be constraint admissible when the loop of each subsystem is closed with $u_i(t) = \hat{\hat{u}}^\star_{i,0}(x_i(t)) + \hat{\hat{K}}_i\left(x_i(t) - \hat{\hat{x}}^\star_{i,0}(x_i(t))\right)$.

5.4.3.1 Stability of the inner nominal trajectories

Setting aside the fact that the RPI constraint of the inner OCP is tightened to (5.18) in order to guarantee feasibility of the outer OCP, the inner OCP is entirely independent of the outer step of robust control. Furthermore, the inner OCP is nothing more than a standard nominal MPC controller in a tube-based approach. In view of this, its cost function can be constructed following the design approach depicted in Section 2.3.1 in order to guarantee stability of the origin for the inner nominal state trajectories. Suppose then that the cost function of the inner OCP is set to

$$\begin{aligned} J^1_{N,i}\left(\hat{\boldsymbol{u}}_i, \hat{x}_{i,0}\right) &= \sum_{k=0}^{N-1} \hat{\ell}_i\left(\hat{x}_{i,k}, \hat{u}_{i,k}\right) + \hat{V}_{f,i}\left(\hat{x}_{i,N}\right) \\ &= \sum_{k=0}^{N-1} \left(||\hat{x}_{i,k}||^2_{\hat{Q}_i} + ||\hat{u}_{i,k}||^2_{\hat{R}_i}\right) + ||\hat{x}_{i,N}||^2_{\hat{P}_i}. \end{aligned}$$

and define $\hat{F}_{ii,f} = \left(A_{ii} + B_{ii}\hat{K}_{f,i}\right)$. The following result holds.

Proposition 5.5. If (a) $\hat{Q}_i, \hat{R}_i > 0$, and $\hat{P}_i$ fulfils $\hat{F}^\top_{ii,f}\hat{P}_i\hat{F}_{ii,f} + \hat{Q}_i + \hat{K}^\top_{f,i}\hat{R}_i\hat{K}_{f,i} - \hat{P}_i \leq 0$ and (b) the nominal state trajectories $\hat{x}_i(t)$ are updated following Step 14 of Algorithm 1, then (i) the origin is exponentially stable for the nominal state trajectories $\hat{x}_i(t)$ for $T \in [0, 1]$ and (ii) the origin is asymptotically stable for the state trajectories $\hat{x}_i(t)$ for $T > 1$.

Proof. (i) follows from the proof of Proposition 2.1 and (ii) from the proof of stability of suboptimal MPC in [1]. ∎

Proposition 5.5 makes a difference with respect to T to account for the fact that the communication procedure results in a feasible yet not necessarily optimal solution to the inner OCP at each time instant $t \neq \lambda T$. It is shown in [1] that the same type of cost decrease is achievable for the case of such suboptimal solutions but the suboptimal cost cannot play the role of a Lyapunov function because it is not uniquely determined for a given state $x_i(t)$. Ultimately, this results in that only asymptotic stability of the origin is achievable. Nevertheless, Proposition 5.5 guarantees that $\hat{x}_i(t) \to \mathbf{0}$ as $t \to \infty$ and so $\hat{u}_i(t) \to \mathbf{0}$ as well.

5.4.3.2 Stability of the outer nominal trajectories

The same, however, cannot be readily guaranteed for the outer nominal trajectories due to the solution of the outer OCP being parametrized by the information received from neighbours. The latter, however, is nothing more than planned inner nominal trajectories, which are bounded and guaranteed to converge given Proposition 5.5. In view of this, the concept of input-to-state stability (ISS) is now employed to provide a guarantee on the stability of the origin for the overall closed-loop controller depicted by Algorithm 1.

First note that, albeit the outer OCP is parametrized by the solution of the inner OCP, Proposition 5.2 guarantees feasibility of the former given feasibility of the latter. Furthermore, Algorithm 1 requires the inner OCP to be feasible at initialization, hence the overall RoA of the two step controller is

$\mathcal{X}_{N,i} = \hat{\mathcal{X}}_{N,i} \oplus \mathbb{H}_i \oplus \mathbb{S}_i$ with $\hat{\mathcal{X}}_{N,i}$ as defined in Section 5.3.1. Given the recursive feasibility of the inner OCP as in Proposition 5.1 and Corollary 5.1, it follows that

$$(\hat{\boldsymbol{x}}_i^s(x_i(t)), \hat{\boldsymbol{u}}_i^s(u_i(t))) \in \prod_{k=0}^{N} \hat{\mathcal{X}}_{N,i} \times \prod_{k=0}^{N-1} \hat{\mathcal{U}}_{N,i}$$

where $\hat{\mathcal{U}}_{N,i}$ is the set of control sequences of length $N-1$ that fulfil the inner OCP constraints for at least one $\hat{x} \in \hat{\mathcal{X}}_{N,i}$. It follows then that $\boldsymbol{f}_i(t) \in \mathcal{F}_i$ with

$$\mathcal{F}_i = \prod_{j \in \mathcal{N}_{i+}^u} \left(\prod_{k=0}^{N} \hat{\mathcal{X}}_{N,j} \times \prod_{k=0}^{N-1} \hat{\mathcal{U}}_{N,j} \right).$$

As before, and owing to the possible necessity of consistency, define the cost function for the outer OCP as

$$\begin{aligned} J_{N,i}^2 \left(\hat{\hat{\boldsymbol{u}}}_i, \hat{\hat{x}}_{i,0}, \boldsymbol{f}_i(t) \right) &= \sum_{k=0}^{N-1} \hat{\hat{\ell}}_i \left(\hat{\hat{x}}_{i,k}, \hat{\hat{u}}_{i,k} \right) + \hat{\hat{V}}_{f,i} \left(\hat{\hat{x}}_{i,N} \right) \\ &= \sum_{k=0}^{N-1} \left(||\hat{\hat{x}}_{i,k}||_{\hat{\hat{Q}}_i}^2 + ||\hat{\hat{u}}_{i,k}||_{\hat{\hat{R}}_i}^2 \right) + ||\hat{\hat{x}}_{i,N}||_{\hat{\hat{P}}_i}^2 \end{aligned} \tag{5.27}$$

and consider the following continuity assumptions.

Assumption 5.8. The optimal cost function of the outer OCP $V_{N,i}^2 (x_i, \boldsymbol{f}_i) : \mathcal{X}_{N,i} \times \mathcal{F}_i \to \mathbb{R}_+$ is Lipschitz continuous in the variable $\boldsymbol{f}_i$. That is, for any pair $\boldsymbol{f}_i, \tilde{\boldsymbol{f}}_i \in \mathcal{F}_i$ and $x_i \in \mathcal{X}_{N,i}$ there exists $L_i > 0$ such that $V_{N,i}^2 (x_i, \boldsymbol{f}_i) - V_{N,i}^2 \left(x_i, \tilde{\boldsymbol{f}}_i \right) \leq L_i |\boldsymbol{f}_i - \tilde{\boldsymbol{f}}_i|_2$.

Assumption 5.9. The terminal cost function of the outer OCP $\hat{\hat{V}}_{f,i} \left(\hat{\hat{x}}_i \right) : \hat{\hat{\mathbb{X}}}_{f,i} \to \mathbb{R}_+$ is Lipschitz continuous. That is, for all pairs $x_i, \tilde{x}_i \in \hat{\hat{\mathbb{X}}}_{f,i}$ there exists $L_{f,i} > 0$ such that $\hat{\hat{V}}_{f,i} (x_i) - \hat{\hat{V}}_{f,i} (\tilde{x}_i) \leq L_{f,i} |x_i - \tilde{x}_i|_2$.

Assumptions 5.8 and 5.9 may seem demanding however it is easy to show that any quadratic form is Lipschitz continuous in a bounded domain and it is shown in [1] that the optimal cost function, as defined in (5.27), also is. Define now $\hat{\hat{F}}_{ii} = \left(A_{ii} + B_{ii} \hat{\hat{K}}_{f,i} \right)$. In view of Assumptions 5.8 and 5.9 ISS of the origin can be established for the outer nominal trajectories.

Theorem 5.1. If (a) Assumptions 5.6 - 5.9 hold and (b) $\hat{\hat{Q}}_i, \hat{\hat{R}}_i > 0$, and $\hat{\hat{P}}_i$ fulfils $\hat{\hat{F}}_{ii}^\top \hat{\hat{P}}_i \hat{\hat{F}}_{ii} + \hat{\hat{Q}}_i + \hat{\hat{K}}_{f,i}^\top \hat{\hat{R}}_i \hat{\hat{K}}_{f,i} - \hat{\hat{P}}_i \leq 0$, then the origin is ISS for the outer nominal trajectories when updated following Step 14 of Algorithm 1.

Proof. Suppose that at time t the state measured by agent i is $x_i(t)$ and the information received by the outer OCP is $\boldsymbol{f}_i(t)$. Suppose also that the optimal solution is obtained and given by $\left(\hat{\boldsymbol{u}}_i^\star(x_i(t)), \hat{\hat{x}}_{i,0}^\star(x_i(t))\right)$ with $\hat{\boldsymbol{u}}_i^\star(x_i(t)) = \left\{\hat{u}_{i,0}^\star(x_i(t)), \dots, \hat{u}_{i,N-1}^\star(x_i(t))\right\}$, associated optimal state trajectory $\hat{\boldsymbol{x}}_i^\star(x_i(t)) = \left\{\hat{\hat{x}}_{i,0}^\star(x_i(t)), \dots, \hat{\hat{x}}_{i,N}^\star(x_i(t))\right\}$ and optimal cost $V_{N,i}^2(x_i(t), \boldsymbol{f}_i(t))$.

Consider first the case in which $t+1 \neq \lambda T$. It follows from Corollary 5.3 that Assumptions 5.6 and 5.7 guarantee that

$$\hat{\boldsymbol{u}}_i^\triangle = \left\{\hat{u}_{i,1}^\star(x_i(t)), \dots, \hat{u}_{i,N-1}^\star(x_i(t)), \hat{K}_{f,i}\hat{\hat{x}}_{i,N}^\star(x_i(t))\right\} \tag{5.28a}$$

$$\hat{\hat{x}}_{i,0}^\triangle = \hat{\hat{x}}_{i,1}^\star(x_i(t)) \tag{5.28b}$$

is a feasible solution to the outer OCP at time $t+1$. Define the cost associated to such a feasible solution as $J_{N,i}^2\left(\hat{\boldsymbol{u}}_i^\triangle, \hat{\hat{x}}_{i,0}^\triangle, \boldsymbol{f}_i(t+1)\right)$, which simplifies to $J_{N,i}^2\left(\hat{\boldsymbol{u}}_i^\triangle, \hat{\hat{x}}_{i,0}^\triangle, \boldsymbol{f}_i^+(t)\right)$ since $t+1 \neq \lambda T$. It is easy to show that

$$\begin{aligned} J_{N,i}^2\left(\hat{\boldsymbol{u}}_i^\triangle, \hat{\hat{x}}_{i,0}^\triangle, \boldsymbol{f}_i^+(t)\right) - V_{N,i}^2(x_i(t), \boldsymbol{f}_i(t)) &= -\hat{\hat{\ell}}_i\left(\hat{\hat{x}}_{i,0}^\star(x_i(t)), \hat{u}_{i,0}^\star(x_i(t))\right) \\ &- \hat{\hat{V}}_{f,i}\left(\hat{\hat{x}}_{i,N}^\star(x_i(t))\right) + \hat{\hat{\ell}}_i\left(\hat{\hat{x}}_{i,N}^\star(x_i(t)), \hat{K}_{f,i}\hat{\hat{x}}_{i,N}^\star(x_i(t))\right) \\ &+ \hat{\hat{V}}_{f,i}\left(\left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\hat{\hat{x}}_{i,N}^\star(x_i(t)) + d_{i,N/t}\right) \end{aligned} \tag{5.29}$$

with $d_{i,N/t}$ defined as in (5.16). By Assumption 5.9 it holds that

$$\begin{aligned} \hat{\hat{V}}_{f,i}\left(\left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\hat{\hat{x}}_{i,N}^\star(x_i(t)) + d_{i,N/t}\right) - \hat{\hat{V}}_{f,i}\left(\left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\hat{\hat{x}}_{i,N}^\star(x_i(t))\right) \\ \leq L_{f,i}|d_{i,N/t}|_2. \end{aligned}$$

In view of this, and by adding and subtracting $\hat{\hat{V}}_{f,i}\left(\left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\hat{\hat{x}}_{i,N}^\star(x_i(t))\right)$ from the right hand side of (5.29) it follows that

$$\begin{aligned} J_{N,i}^2\left(\hat{\boldsymbol{u}}_i^\triangle, \hat{\hat{x}}_{i,0}^\triangle, \boldsymbol{f}_i^+(t)\right) - V_{N,i}^2(x_i(t), \boldsymbol{f}_i(t)) &\leq -\hat{\hat{\ell}}_i\left(\hat{\hat{x}}_{i,0}^\star(x_i(t)), \hat{u}_{i,0}^\star(x_i(t))\right) \\ &- \hat{\hat{V}}_{f,i}\left(\hat{\hat{x}}_{i,N}^\star(x_i(t))\right) + \hat{\hat{\ell}}_i\left(\hat{\hat{x}}_{i,N}^\star(x_i(t)), \hat{K}_{f,i}\hat{\hat{x}}_{i,N}^\star(x_i(t))\right) \\ &+ \hat{\hat{V}}_{f,i}\left(\left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\hat{\hat{x}}_{i,N}^\star(x_i(t))\right) + L_{f,i}|d_{i,N/t}|_2. \end{aligned}$$

Furthermore, by assumption the terminal cost function fulfils

$$\hat{\hat{V}}_{f,i}\left(\left(A_{ii} + B_{ii}\hat{\hat{K}}_{f,i}\right)\hat{\hat{x}}_{i,N}^\star(x_i(t))\right) - \hat{\hat{V}}_{f,i}\left(\hat{\hat{x}}_{i,N}^\star(x_i(t))\right) \leq$$

$$- \hat{\hat{\ell}}_i \left(\hat{\hat{x}}^{\star}_{i,N}(x_i(t)), \hat{K}_{f,i} \hat{\hat{x}}^{\star}_{i,N}(x_i(t)) \right)$$

for all $\hat{\hat{x}}^{\star}_{i,N}(x_i(t) \in \hat{\hat{\mathbb{X}}}_{f,i}$ and so

$$J^2_{N,i}\left(\hat{\hat{\boldsymbol{u}}}^{\triangle}_i, \hat{\hat{x}}^{\triangle}_{i,0}, \boldsymbol{f}^+_i(t) \right) - V^2_{N,i}(x_i(t), \boldsymbol{f}_i(t)) \leq - \hat{\hat{\ell}}_i \left(\hat{\hat{x}}^{\star}_{i,0}(x_i(t)), \hat{\hat{u}}^{\star}_{i,0}(x_i(t)) \right) + L_{f,i}|d_{i,N/t}|_2.$$

Finally, note that by assumption the pair $\left(\hat{\hat{\boldsymbol{u}}}^{\triangle}_i, \hat{\hat{x}}^{\triangle}_{i,0} \right)$ is feasible yet not necessarily optimal and so

$$V^2_{N,i}(x_i(t+1), \boldsymbol{f}^+_i(t)) - V^2_{N,i}(x_i(t), \boldsymbol{f}_i(t)) \leq - \hat{\hat{\ell}}_i \left(\hat{\hat{x}}^{\star}_{i,0}(x_i(t)), \hat{\hat{u}}^{\star}_{i,0}(x_i(t)) \right) + L_{f,i}|d_{i,N/t}|_2.$$

If $t+1 = \lambda T$ it follows that $\boldsymbol{f}_i(t+1)$ is not necessarily equal to $\boldsymbol{f}^+_i(t)$ and so (5.28) is not necessarily feasible. Nevertheless, in view of Assumption 5.8 it follows that

$$V^2_{N,i}(x_i(t+1), \boldsymbol{f}_i(t+1)) - V^2_{N,i}(x_i(t+1), \boldsymbol{f}^+_i(t)) \leq L_i|\boldsymbol{f}_i(t+1) - \boldsymbol{f}^+_i(t)|_2,$$

and so

$$V^2_{N,i}(x_i(t+1), \boldsymbol{f}_i(t+1)) - V^2_{N,i}(x_i(t), \boldsymbol{f}_i(t)) \leq - \hat{\hat{\ell}}_i \left(\hat{\hat{x}}^{\star}_{i,0}(x_i(t)), \hat{\hat{u}}^{\star}_{i,0}(x_i(t)) \right) + L_{f,i}|d_{i,N/t}|_2 + L_i|\boldsymbol{f}_i(t+1) - \boldsymbol{f}^+_i(t)|_2. \tag{5.30}$$

Since $V^2_{N,i}(x_i(t), \boldsymbol{f}_i(t))$ is defined as (5.27) with bounded domain $\mathcal{X}_{N,i} \times \mathcal{F}_i$, it can be upper and lower bounded by appropriate functions such as in Section 2.3.1. It follows then, from (5.30), that $V^2_{N,i}(x_i(t), \boldsymbol{f}_i(t))$ is an ISS-Lyapunov function for the outer nominal trajectories when updated following Step 14 of Algorithm 1, and so the origin is ISS. ■

The ISS approach regards the information in $d_{i,N/t}$ and $\boldsymbol{f}_i(t)$ as bounded disturbances. Therefore, when in closed-loop with Algorithm 1 the only guarantee available is that $\hat{\mathcal{X}}_{N,i}$ is invariant for the inner nominal trajectories for subsystem i and so $\mathcal{X}_N = \prod_{i \in \mathcal{N}} \mathcal{X}_{N,i}$ is invariant for the state trajectories of the network. However, if Proposition 5.5 holds it follows that $d_{i,N/t} \rightarrow \mathbf{0}$ and $\left(\boldsymbol{f}_i(t+1) - \boldsymbol{f}^+_i(t)\right) \rightarrow \mathbf{0}$ as $t \rightarrow \infty$. In the limit then, the optimal cost function of the outer OCP becomes a Lyapunov function for the outer nominal state trajectories $\hat{\hat{x}}_i(t)$, and the origin becomes exponentially stable for $\hat{\hat{x}}_i(t)$ (in the limit). In fact, if at time instant $\tilde{t}$ it happens that $x(\tilde{t}) \in \mathbb{S} \oplus \mathbb{H}$, then it holds that $d_{i,N/t} = \mathbf{0}$ and $\left(\boldsymbol{f}_i(t+1) - \boldsymbol{f}_i^+(t)\right) = \mathbf{0}$ for all $t \geq \tilde{t}+1$. It follows then that the optimal cost function of the outer OCP becomes a Lyapunov function for the outer nominal state trajectories for any $t \geq \tilde{t}+1$.

In all the numerical simulations carried out it was observed that the optimal cost variation in (5.30) becomes bounded by a negative value during the first few time steps of operation of Algorithm 1, rather than at the limit (see Section 5.6). Consider then the overall network dynamics, when each subsystem is in closed-loop with Algorithm 1

$$\begin{aligned}\boldsymbol{x}(t+1) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\left(\hat{\hat{\boldsymbol{u}}}(t) + \hat{\hat{\boldsymbol{K}}}_i\left(\boldsymbol{x}(t) - \hat{\hat{\boldsymbol{x}}}(t)\right)\right) \\ &= \left(\boldsymbol{A} + \boldsymbol{B}\hat{\hat{\boldsymbol{K}}}_i\right)\boldsymbol{x}(t) + \left(\boldsymbol{B}\hat{\hat{\boldsymbol{u}}}(t) - \hat{\hat{\boldsymbol{K}}}_i\hat{\hat{\boldsymbol{x}}}(t)\right),\end{aligned} \tag{5.31}$$

where $\hat{\hat{\boldsymbol{x}}} = \left(\hat{\hat{x}}_1^\top, \hat{\hat{x}}_2^\top, \ldots, \hat{\hat{x}}_M^\top\right)^\top$ and $\hat{\hat{\boldsymbol{u}}} = \left(\hat{\hat{u}}_1^\top, \hat{\hat{u}}_2^\top, \ldots, \hat{\hat{u}}_M^\top\right)^\top$. In practice, Theorem 5.1 results in $\hat{\hat{x}}_i(t) \to \mathbf{0}$ as $t \to \infty$ for all $i \in \mathcal{M}$ and so $\hat{\hat{\boldsymbol{x}}}(t) \to \mathbf{0}$ and $\hat{\hat{\boldsymbol{u}}}(t) \to \mathbf{0}$ as well. This implies that the rightmost summand in (5.31) vanishes as $t \to \infty$.

The overall network dynamics then reduce to $\boldsymbol{x}(t) = \left(\boldsymbol{A} + \boldsymbol{B}\hat{\hat{\boldsymbol{K}}}_i\right)\boldsymbol{x}(t)$ at the limit, which highlights the importance of Assumption 5.1. In particular, stability of the origin for the overall network can only be attained (and in practice will be attained) if the collection of gains $\hat{\hat{K}}_i$ satisfies Assumption 5.1.

5.5 Stabilizability assumption

From the discussion in Section 5.4.3.2 it follows that a necessary condition for stabilizability of the origin is that the collection of gains $\hat{\hat{K}}_i$, associated to the outer tubes, satisfies Assumption 5.1. This is because convergence of the outer nominal trajectories results in a closed-loop network where the loop of each subsystem is closed independently by $\hat{\hat{K}}_i$, which is nothing more than locally stabilizing. Generally speaking, this may result in the overall network to be unstable [151] (if Assumption 5.1 does not hold). Indeed, consider for example a pair of coupled integrators with $A_{11} = A_{22} = 1$, $A_{12} = A_{21} = 0$, $B_{11} = B_{22} = 1$ and $B_{12} = B_{21} = 0.5$. The local linear feedbacks $K_1 = K_2 = -1.5$ result in stable local closed-loops with $A_{ii} + B_{ii}K_i = -0.5$, but in a closed-loop network $\boldsymbol{A} + \boldsymbol{B}\boldsymbol{K}$ with an spectral radius of 1.25. It follows then that the selection of $\hat{\hat{K}}_i$, if done naively, could result in an unstable network once the outer nominal trajectories have converged.

The gains $\hat{\hat{K}}_i$, however, are not only required to be locally stabilizing, but also to render admissible RPI sets $\mathbb{S}_i$. In what follows it is shown that the latter guarantees that Assumption 5.1 is met by the collection of gains $\hat{\hat{K}}_i$. There are, however, other ways in which Assumption 5.1 can be met in the context of designing an stabilizing DMPC controller. Some of the techniques used to compute these feedbacks also exploit notions of invariance, and hence are related to the more fundamental result shown in this section. A brief review of such approaches is presented first, and the following definition is required for simplicity.

Definition 5.1. The set $\mathbb{Y} \subseteq \mathbb{R}^{n_a}$ is a referred to as a square or separable set if $\mathbb{Y} = \prod_{i=1}^{b} \mathcal{Y}_i$ with $\mathcal{Y}_i \in \mathbb{R}^{n_i}$ and $n_a = \sum_{i=1}^{b} n_i$.

The design of DMPC controllers (not necessarily tube-based) is generally not trivial and usually requires some form of centralized computation. Algorithm 1, for example, not only needs a collection of gains that fulfils Assumption 5.1, but also the computation of the sets $\mathbb{V}_i$, which means agents must share the information about their sets $\mathbb{Z}_i$, $\mathbb{U}_i$ and $\hat{\mathbb{U}}_i$ with their neighbours. With respect to the construction of a linear feedback with a particular structure (such as block-diagonal), it is often the case that the several characteristics that this gain needs to fulfil (both at local and global level) can be posed in the form of a set of (coupled) LMIs. In [165] for example, subsystems that are coupled through the input are studied and a global linear feedback with binary structure equating that of the input matrix is sought to play the role of terminal controller often used in MPC implementations [4]. It follows then that some form of Lyapunov condition must be imposed over the linear gain, both at the local and global levels. These requirements are translated into a set of M local LMIs alongside with a single global LMI of dimension $4n \times 4n$ (where n is the overall dimension of the plant); if such LMIs are verified, the structured gain meets all requirements and can be partitioned to be used as a terminal controller for the local optimization problems.

A similar approach is presented in [169] but for subsystems coupled through the state. Stability of the proposed cooperative DMPC scheme hinges on the terminal cost of the centralized problem being a Lyapunov function for the network dynamics when in closed-loop with a linear terminal controller. The MPC optimization problem, however, is solved in a distributed fashion, hence a collection of local terminal cost function that result in a network-wide Lyapunov function is sought. Local feedbacks are associated to each local terminal cost and the Lyapunov requirements are again presented as a set of M local LMIs in presence of a system-wide coupled LMI. The difference with respect to [165] is that a distributed optimization is proposed to find a collection of local terminal controllers that fulfils the LMIs, allowing for distributed design even in the presence of a global LMI.

In the context of tube-based DMPC schemes a similar approach is found in [34]. Analogously to [169], stability of the origin depends on the terminal cost of the aggregated problem being a Lyapunov function for the closed-loop network. The main difference is that the approach in [34] is non-cooperative, as opposed to the cooperative scheme in [169]. Nevertheless, the properties of the aggregated terminal cost are again guaranteed by a collection of independent local terminal costs and gains designed to respect a set of coupled LMIs that represent the cost decrease requirements required for Lyapunov stability.

An interesting by-product of the block-diagonal structure imposed over the stabilizing terminal gain in [34] is that the local terminal constraint sets must not only be PI for the local dynamics when in closed loop with the local gains, but its Cartesian product be a PI set for the global dynamics when in closed-loop with the block-diagonal global gain. Such a global PI set is a square set according to Definition 5.1. A similar result is found in [169], except that the local terminal sets are allowed to be time-varying rather than PI, although still resulting in a square global terminal set. It follows then that the synthesis of a square (or separable) PI set also results in obtaining a collection of local linear gains that guarantee Assumption 5.1. The synthesis of such sets is specifically tackled in [179], by posing a set of LMIs that represent the square invariance requirements. The procedure, however, is centralized, and involves LMIs of dimension $2n \times 2n$ (where n is the overall dimension of the plant).

The construction of the time-varying square terminal constraint sets in [169] can be included in the more general notion of positively invariant family of sets (PIFs) introduced in [138], although the latter propose a different parametrization. A PIFs is nothing more than a collection of bounded square sets that contains, at all times, the trajectories of an autonomous system (or system in closed-loop with a collection of local gains). Each element of the family is square and not necessarily invariant on its own, but the family as a whole is. This concept also encompasses the square PI set required by the DMPC algorithm in [34], since the latter is nothing more than a PIFs with a single element.

If the goal is just to find a collection of feedbacks that fulfils Assumption 5.1, as opposed to one with an associated Lyapunov function as in [34, 165], the parametrization in [180] presents certain advantages when compared to that in [169], albeit it resorts to a fully centralized design process. In [180] the authors present what is called a comparison system which has dimension M (the number of subsystems) and whose dynamics depends on the dynamics of the network when in closed-loop with a particularly structured gain. It is shown that a stable comparison system is a sufficient condition for the true system to accept a PIFs, and that the latter implies that the network admits a PI set, hence is stable. It might then prove easier to find a collection of local gains that renders the comparison system stable and, by construction, stabilize the true system.

This notion is the driving idea of the approach proposed in [181], where local LMIs are constructed to find, in a non-centralized fashion, a collection of local gains with its corresponding PIFs. Nevertheless, although a stabilizing decentralized gain may be found by exploiting the parametrization in [180], such a gain will not necessarily have an associated PIFs with a single element, such as is needed in [34]. It follows that the notions in [180] may provide a simpler procedure to the design of a gain that fulfils Assumption 5.1 when compared to solving system-wide coupled LMIs such as in [34, 165, 169], but the characterization of associated square PI sets remains complex. In [164], however, a square PI set for the global dynamics is easily computed employing local RPI sets, rather than PI sets. This is the key aspect of the sufficient conditions shown in the following to guarantee existence of a block-diagonal feedback that meets Assumption 5.1.

5.5.1 Computation of a block-diagonal stabilizing gain

In what follows it will be shown that admissibility of the inner OCP is sufficient to guarantee that the collection of gains $\hat{K}_i$ fulfils Assumption 5.1. The stabilizability result in Section 5.4.3 requires $\hat{\hat{K}}_i$ to meet Assumption 5.1, rather than $\hat{K}_i$, but by showing that the latter does meet the assumption, existence of (at least) one such gain is guaranteed.

First define $\hat{F}_{ii} = A_{ii} + B_{ii}\hat{K}_i$ and consider the following assumption.

Assumption 5.10. The inner OCP is admissible for all $i \in \mathcal{M}$. That is, the tube gain $\hat{K}_i$ and the RPI set associated to the inner OCP $\mathbb{Z}_i$ are such that

$$\begin{aligned} \hat{F}_{ii}\mathbb{Z}_i \oplus \mathbb{W}_i &\subseteq \mathbb{Z}_i \\ \mathbb{Z}_i &\subseteq \mathbb{X}_i \end{aligned}$$

$$\hat{K}_i \mathbb{Z}_i \subseteq \mathbb{U}_i.$$

The following result then holds

Proposition 5.6. Assume that the outer OCP is not in place and that the independent inner OCPs are used to control each subsystem independently with $u_i(t) = \hat{K}_i x_i(t)$. If for all $i \in \mathcal{M}$ Assumption 5.10 holds and $x_i(0) \in \mathbb{Z}_i$, then $x_i(t) \in \mathbb{X}_i$ and $u_i(t) \in \mathbb{U}_i$ for all $i \in \mathcal{M}$ and for all $t \geq 0$.

Proof. If the control law is set to $u_i(t) = \hat{K}_i x_i(t)$, it follows that the closed-loop reduces to $x_i(t+1) = \hat{F}_{ii} x_i(t) + w_i(t)$ with $w_i(t) \in \mathbb{W}_i$. The rest of the proof follows from $x_i(0) \in \mathbb{Z}_i$ and the robust invariance of $\mathbb{Z}_i$ for all $i \in \mathcal{M}$. Indeed, $x_i(t) \in \mathbb{Z}_i \subseteq \mathbb{X}_i$ and $u_i(t) \in \hat{K}_i \mathbb{Z}_i \subseteq \mathbb{U}_i$ for all $t \geq 0$, given Assumption 5.10. ■

Proposition 5.6 provides a guarantee of constraint satisfaction in a decentralized design approach where agents have shared only information about their constraint sets to compute $\mathbb{W}_i$. This is counter-intuitive with respect to the previous discussion since Proposition 5.6 does not require Assumption 5.1 and, as shown, a naive selection of the local gains could result in an unstable closed-loop network which then cannot guarantee constraint satisfaction.

Consider, for example, a network of two subsystems and suppose that Proposition 5.6 holds, but that Assumption 5.1 does not. That is $\hat{F}_{ii}$ is Schur for $i = 1, 2$, but the system-wide closed-loop $\hat{\boldsymbol{F}} = \boldsymbol{A} + \boldsymbol{B}\hat{\boldsymbol{K}}$, with $\hat{\boldsymbol{K}} = diag\left(\hat{K}_1, \hat{K}_2\right)$, is not. Since $\boldsymbol{x}(0) \in \mathbb{Z} = \mathbb{Z}_1 \times \mathbb{Z}_2 \subset \mathbb{X}$, instability of $\hat{\boldsymbol{F}}$ means that there must exist a time instant $t_1 > 0$ such that, $x_1(t_1 - 1) \in \mathbb{Z}_1$ but $x_1(t_1) \notin \mathbb{Z}_1$. At the core, this implies that the robust invariance of the set $\mathbb{Z}_1$ for the closed-loop $\hat{F}_{11}$ has been broken, which can only take place if $w_1(t_1 - 1) \notin \mathbb{W}_1$. This behaviour is summarized in the following result.

Proposition 5.7. Assume that the outer OCP is not in place and that the independent inner OCPs are used to control each subsystem independently with $u_i(t) = \hat{K}_i x_i(t)$ such that $\hat{\boldsymbol{F}}$ is not Schur. If for all $i \in \mathcal{M}$ Assumption 5.10 holds and $x_i(0) \in \mathbb{Z}_i$, then there exists a finite time $\hat{t} > 0$ such that $\boldsymbol{x}_{\hat{t}-1} \in \mathbb{Z}$ but $\boldsymbol{x}_{\hat{t}} \notin \mathbb{X}$ or $\boldsymbol{u}_{\hat{t}} \notin \mathbb{U}$.

Proof. For the proof consider again a network of two subsystems. As before, instability of $\hat{\boldsymbol{F}}$ means there exist a time instant $t_1 > 0$ such that, $x_1(t_1 - 1) \in \mathbb{Z}_1$ but $x_1(t_1) \notin \mathbb{Z}_1$. The violation of the robust invariance of $\mathbb{Z}_1$ implies $w_1(t_1 - 1) \notin \mathbb{W}_1$, which can only happen if $x_2(t_1 - 1) \notin \mathbb{X}_2$ or $u_2(t_1 - 1) \notin \mathbb{U}_2$, where the latter implies the former given the control law considered. Furthermore $x_2(t_1 - 1) \notin \mathbb{X}_2$

implies $x_2(t_1-1) \notin \mathbb{Z}_2$ and so, if $x_2(t_1-2) \in \mathbb{Z}_2$ it must be that $w_2(t_1-2) \notin \mathbb{W}_2$ with the obvious implications on the state constraint satisfaction of subsystem 1 at time $t_1 - 2$. ■

Proposition 5.7 states that, if the local gains are chosen without considering global behaviour, the network-wide state may evolve such that, in a single time step, it will go from inside $\mathbb{Z}$ to outside $\mathbb{X}$. This is irrespective of the particular $\hat{\boldsymbol{K}}$, as long as $\hat{\boldsymbol{F}}$ is not Schur, which is counter-intuitive since for a particular non-stabilizing $\hat{\boldsymbol{K}}$ it may happen that $\hat{\boldsymbol{F}}\mathbb{Z} \subseteq \mathbb{X}$. This inconsistency owes to the fact that the local gains $\hat{K}_i$ are not only chosen to stabilize the local dynamics (A_{ii}, B_{ii}), but to fulfil all conditions in Assumption 5.10. The effect that these additional requirements have on the global properties of $\hat{\boldsymbol{K}}$ is summarized in the following result.

Theorem 5.2. If Assumption 5.10 holds, then the collection of local gains $\hat{K}_i$ fulfils Assumption 5.1.

Proof. First define $\hat{F}_{ij} = \left(A_{ij} + B_{ij}\hat{K}_j\right)$. For all $i, j \in \mathcal{M}$ with $i \neq j$ it holds that

$$\begin{aligned}\tilde{\mathbb{W}}_{ij} = \hat{F}_{ij}\mathbb{Z}_j &\subseteq A_{ij}\mathbb{Z}_j \oplus B_{ij}K_j\mathbb{Z}_j \\ &\subseteq A_{ij}\mathbb{X}_j \oplus B_{ij}\mathbb{U}_j,\end{aligned}$$

where the first inclusion follows from Minkwoski sum properties and the second one by Assumption 5.10. It holds then that

$$\tilde{\mathbb{W}}_i = \bigoplus_{j\in\mathcal{N}_i^u} \tilde{\mathbb{W}}_{ij} \subseteq \bigoplus_{j\in\mathcal{N}_i^u} (A_{ij}\mathbb{X}_j \oplus B_{ij}\mathbb{U}_j) = \mathbb{W}_i,$$

and so $\hat{F}_{ii}\mathbb{Z}_i \oplus \tilde{\mathbb{W}}_i \subseteq \hat{F}_{ii}\mathbb{Z}_i \oplus \mathbb{W}_i \subseteq \mathbb{Z}_i$ for all $i \in \mathcal{M}$. It follows then that

$$\hat{F}_{ii}\mathbb{Z}_i \oplus \bigoplus_{j\in\mathcal{N}_i} \hat{F}_{ij}\mathbb{Z}_j \subset \mathbb{Z}_i, \quad \forall i \in \mathcal{M},$$

and hence, $\mathbb{Z} = \mathbb{Z}_1 \times \cdots \times \mathbb{Z}_M \subseteq \mathbb{R}^n$ is a square PI set for the closed-loop dynamics $\hat{\boldsymbol{F}}$. In view of this, $\hat{\boldsymbol{F}}$ is Schur. ■

Theorem 5.2 establishes that, in a distributed control set-up, admissibility of local tubes designed to counter the interaction is sufficient to guarantee that the associated local gains form a block-diagonal stabilizing feedback for the overall network. In summary then, Assumption 5.1 is redundant in view of the necessity of admissible local tubes, nevertheless, both design demands are usually found together in the literature of tube-based DMPC architectures [30,31,33,34,173,174].

Proposition 5.7 sets off with an assumption on unstable global dynamics, however it has been shown in Theorem 5.2 that this cannot be the case since this would imply that constraints are not met and hence RPI sets cannot be computed given unbounded disturbance sets. The behaviour described in the proof of Proposition 5.7 then is not possible since the corresponding invariant sets would have not been admissible to start with and so the assumptions that give way to Proposition 5.7 cannot hold simultaneously.

Theorem 5.2 provides sufficient conditions to find a globally stabilizing block-diagonal gain that depends on the admissibility of local tubes. However, if the coupling is large or the constraint sets of some neighbours are larger when compared to the local ones, there might not exist an admissible RPI set $\mathbb{Z}_i$. This is not a drawback of Theorem 5.2 in particular, and owes to the fact that dealing with strong coupling between subsystems is complex when a completely decentralized approach is pursued. Even in a distributed set-up, such as in [165,169], there is no guarantee that a collection of gains that fulfils the corresponding LMIs will exist. This should not be a surprise, given that the successful synthesis of a non-centralized controller depends greatly on the size of the interaction between neighbouring subsystems, and how these are dealt with (communication, iterative optimization, etc.).

The DMPC approach proposed here requires admissibility of such local tubes, but if the only goal is to compute a global gain that fulfils Assumption 5.1, then Theorem 5.2 is still a useful tool. Indeed, if a given collection of local gains does not render all tubes admissible, it is possible to reduce size of certain constraint sets to ultimately obtain admissible tubes for all subsystems and hence a guarantee on global stabilizability (as is done in [165]).

Finally, note that a collection of constraint admissible tubes $\mathbb{Z}_i$, such as that required by Assumption 5.10, forms a square PI set, as shown in Theorem 5.2. Depending on the size of the coupling, however, a system may not admit such a PI set (excluding the trivial set containing only the origin). This highlights the conservativeness introduced in DMPC algorithms such as the one presented in here or the one in [34] when compared to that of [169], which allows for time-varying terminal sets that are not PI but PIFs [180]. Indeed, the parametrization of PIFs proposed by [138] results in that a square PI set is guaranteed to exist only if the chosen gains result in a marginally stable comparison system.

5.6 Illustrative example

To test the proposed controller, a slightly modified version of the four truck system from [34] is employed. Recall that the proposed approach is fitted to deal with input coupling, however the computation of a comprehensive bounding set that contains the input trajectory deviation is not trivial. In view of this, and in order to employ the simplified approach described in Section 5.4.2.3, an illustrative example without input coupling is considered. The plant is depicted in Figure 5.2; four trucks represented by point masses m_i are dynamically coupled through springs k_{ij} and dampers c_{ij}. Each truck is connected only to its immediate neighbours: truck 1 is connected to truck 2; truck 2 is connected to trucks 1 and 3, and so on. The control objective is to steer each truck to an arbitrary equilibrium point with null velocity, using the independent control inputs u_i.

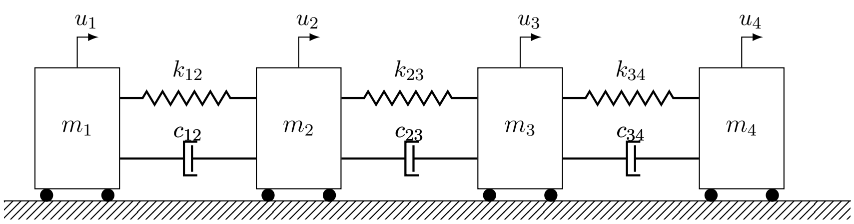

Figure 5.2: Network composed by four subsystems (trucks).

Assuming over damping, the relative displacement between neighbouring trucks will converge to 0 and all velocities will converge to the same value, however not necessarily to 0, as the trucks are not fixed to the ground. Indeed, the system is marginally stable. Table 5.1 reports the values of the different plant parameters. The coupling between trucks 1 and 2 is purposely higher than among the other neighbouring trucks, in order to study the behaviour of the proposed DMPC algorithm under high and low couplings.

The state vector of the plant is $\boldsymbol{x} \in \mathbb{R}^8$ with 4 pairs of elements representing the position and velocity of each truck. Given their clearly defined physical boundaries, and their independent control actions, the overall system is divided into 4 subsystems, each of them representing a single truck. The continuous

Table 5.1: Dynamic parameters of the network.

Spring	Damper	Mass
$k_{12} = 7.50$	$c_{12} = 4.00$	$m_1 = 3$
$k_{23} = 0.75$	$c_{23} = 0.25$	$m_2 = 2$
$k_{34} = 1.00$	$c_{34} = 0.30$	$m_3 = 3$
–	–	$m_4 = 6$

time dynamics of each subsystem are then defined by

$$\dot{x}_i = A^c_{ii} x_i + B^c_{ii} u_i + \sum_{j \in \mathcal{N}_i} A^c_{ij} x_j$$

where the upper index ^{c} indicates that it is a continuous time representation and $\dot{}$ indicates the time derivative. The local input matrices B^c_{ii} are a function of m_i while the matrices A^c_{ij} are a function of the various spring and damper coefficients involved in the dynamics of each truck.

The state and input constraints are homogeneous for all trucks and defined by

$$\mathbb{X}_i = \left\{ x_i \in \mathbb{R}^2 \mid x_i \leq \begin{bmatrix} 2 \\ 8 \end{bmatrix} \right\}$$
$$\mathbb{U}_i = \{ u_i \in \mathbb{R} \mid |u_i| \leq 4 \}.$$

The cost functions are also defined in a homogeneous way, both across subsystems and OCPs, with $R_{1:4} = \hat{R}_{1:4} = \hat{\hat{R}}_{1:4} = 1$ and

$$Q_{1:4} = \hat{Q} = \hat{\hat{Q}}_{1:4} = \begin{bmatrix} 10 & 0 \\ 0 & 1 \end{bmatrix}.$$

A sampling time of $T_s = 0.1[s]$ is used to discretize the system, and in order to simplify the design approach, all the different linear gains required are set to the optimal infinite horizon LQR gain associated to each truck. Table 5.2 reports the gain values. Finally, the terminal costs P_i are set to the solution

Table 5.2: Linear gains for each subsystem $K_i = \hat{K}_i = \hat{\hat{K}}_i = \hat{K}_{f,i} = \hat{\hat{K}}_{f,i} = \bar{K}_i$.

Truck	1	2	3	4
$-K_i^\top$	$\begin{bmatrix} 0.7379 \\ 0.2992 \end{bmatrix}$	$\begin{bmatrix} 0.5057 \\ 0.1990 \end{bmatrix}$	$\begin{bmatrix} 0.7478 \\ 0.3174 \end{bmatrix}$	$\begin{bmatrix} 1.2790 \\ 0.5631 \end{bmatrix}$

of the corresponding Lyapunov equation, that is, to approximate exactly the unconstrained infinite horizon LQR cost.

5.6.1 Set related results

As discussed in Section 5.4.2.3, the recursive feasibility guarantee for Algorithm 1 depends almost completely on Assumption 5.2, however the comprehensive computation of the sets $\mathbb{V}_i$ is not trivial. In this example, there is no input coupling, and hence $\mathbb{V}_i$ is accurately defined by (5.26). Given that by admissibility requirements $\mathbb{Z}_i$ needs to be in the interior of $\mathbb{X}_i$, Assumption 5.2 is straightforwardly met. Figure 5.3 shows the different disturbance sets. As

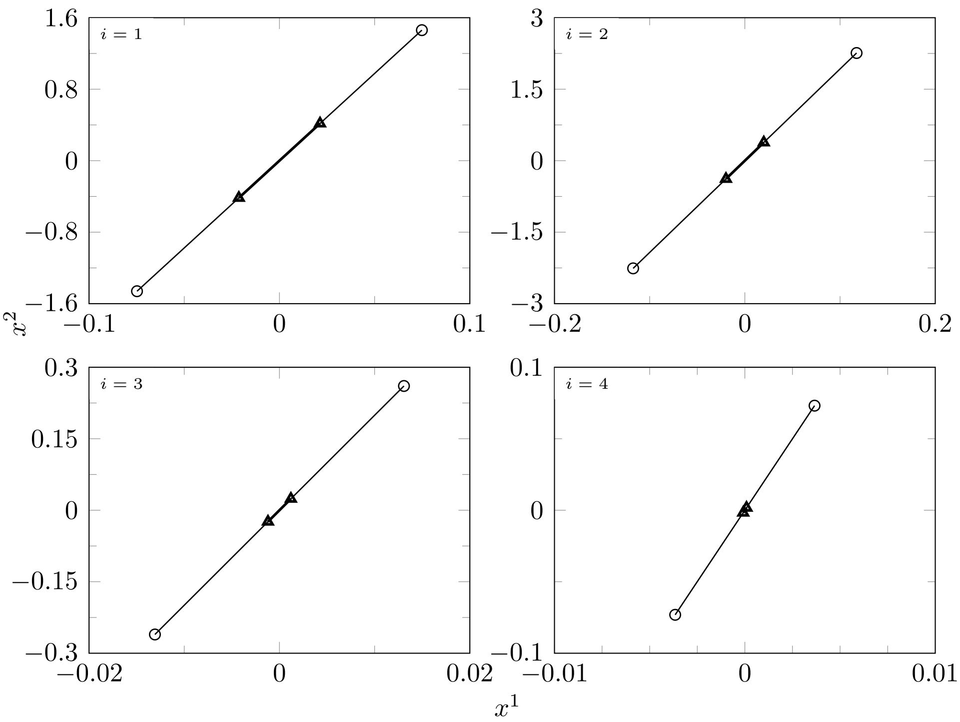

Figure 5.3: Disturbance sets for inner and outer OCPs: ─○─ inner OCP $\mathbb{W}_i$, ─▲─ outer OCP $\mathbb{V}_i$.

expected, the largest disturbance sets are obtained for truck 2, since it has two neighbours, being the coupling between trucks 1 and 2 the strongest one in the whole network. The smallest disturbance sets are those associated to truck 4, which again is expected given its single neighbour and weak coupling.

Figure 5.4 shows the tightening sets for both OCPs, $\mathbb{Z}_i$ and $\mathbb{S}_i$, computed as the minimal RPI sets associated to the respective disturbance sets. Again, given the sizes of the corresponding perturbation sets, the largest tightening sets are obtained for truck 2, and the smallest ones for truck 4. Furthermore, the largest difference between the inner and outer tightening set is also observed in subsystem 4, where $\mathbb{S}_4$ represents only 0.05% of the volume of $\mathbb{Z}_4$. On the other hand, the largest outer tightening set, relative to the size of the inner for the same subsystem, takes place in subsystem 1, where $\mathbb{S}_1$ represents 8.1% of the volume of $\mathbb{Z}_1$. The latter is due to two reasons: firstly, the strong coupling with subsystem 2 and secondly, the fact that subsystem 2 has the largest inner tightening set $\mathbb{Z}_2$, resulting in disturbance set $\mathbb{V}_1$ that is not considerably smaller than $\mathbb{W}_1$.

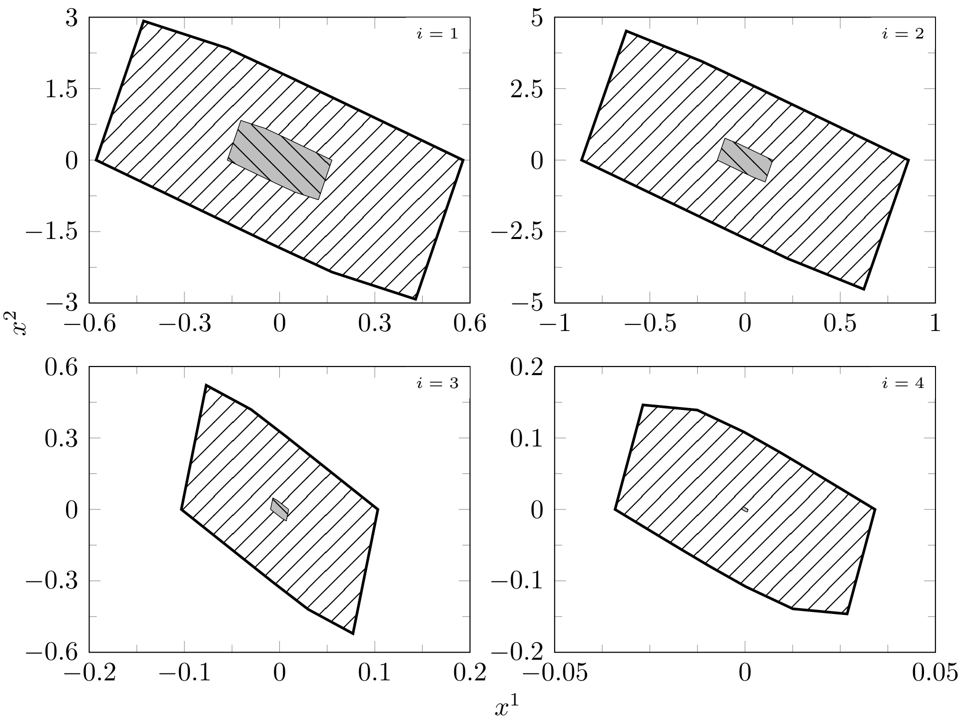

Figure 5.4: Tightening sets for inner and outer OCPs: ▱ inner OCP $\mathbb{Z}_i$, ▧ outer OCP $\mathbb{S}_i$.

However, the RPI set $\mathbb{Z}_i$ is not used as a constraint in the inner OCP. Indeed, in order to guarantee recursive feasibility of the overall scheme the RPI constraint of the inner OCP is replaced by $\mathbb{S}_i \oplus \mathbb{H}_i$ which needs to be a subset of $\mathbb{Z}_i$. This has an impact on the size of the inner OCP's RoA, and hence, on the RoA of the overall distributed controller. It follows that it would be beneficial to compute $\mathbb{H}_i$ as the largest set that fulfils all the requirements of Assumption 5.6. The sets $\mathbb{H}_i$ could be computed as the maximal RPI sets for $\left(A_{ii} + B_{ii}\bar{K}_i\right)$ and disturbance set $\mathbb{D}_i$ contained inside $\mathbb{Z}_i \ominus \mathbb{S}_i$, however this may result in that conditions (5.21c) and (5.21d) are not met.

To guarantee that all requirements are met, the sets $\mathbb{H}_i$ for this example where computed as the minimal robust λ-contractive sets for $\big(A_{ii} + B_{ii}\bar{K}_i\big)$ and disturbance set $\mathbb{D}_i$ (see Definition 3.5). The value of λ for each subsystem was set to the smallest possible such that all requirements of Assumption 5.6 where met. Figure 5.5 compares the sizes of $\mathbb{Z}_i$ and $\mathbb{S}_i \oplus \mathbb{H}_i$. It can be observed that the proposed approach to compute $\mathbb{H}_i$ results in a minimal loss in the size of the RPI constrain, and hence on the feasibility region. The largest reduction is observed for subsystem 1, for the same reasons as before, with a set $\mathbb{S}_1 \oplus \mathbb{H}_1$ that amounts only to a 79% of the volume of $\mathbb{Z}_1$.

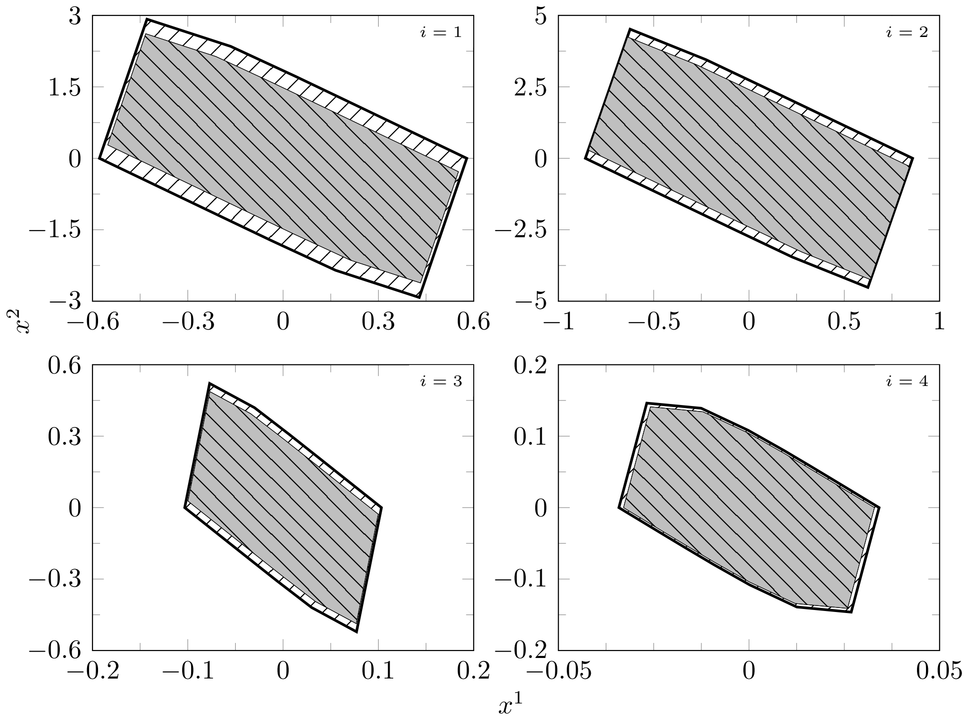

Figure 5.5: RPI constraint for inner OCP: standard $\mathbb{Z}_i$, modified $\mathbb{S}_i \oplus \mathbb{H}_i$.

Finally, the RoAs for each subsystem are shown in Figure 5.6, and compared to the original state constraint set. As observed, and thanks to the approach for computing $\mathbb{H}_i$, the feasibility regions of each subsystem account for a considerable amount of the overall allowable state region. The smallest RoA is that of subsystem 2, owing to its strong coupling with subsystem 1 and additional coupling with subsystem 3, which yields the largest RPI set $\mathbb{Z}_2$ as observed in Figure 5.4.

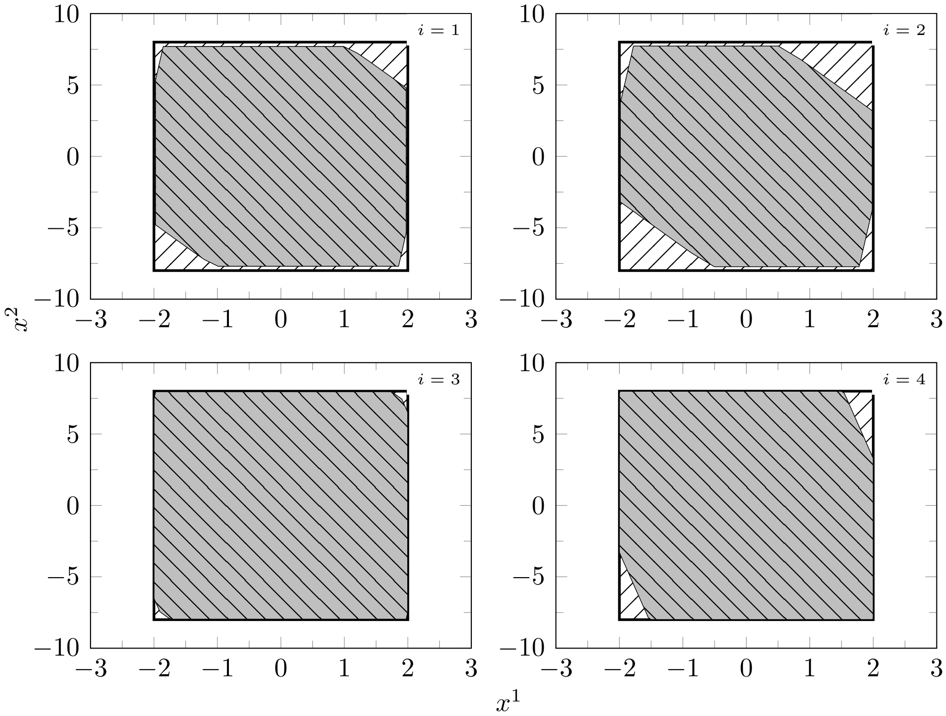

Figure 5.6: Region of attraction: $\mathbb{X}_i$, $\mathcal{X}_i = \hat{\mathcal{X}}_i \oplus \mathbb{S}_i \oplus \mathbb{H}_i$.

5.6.2 Control performance

In order to assess the control performance of the proposed DMPC algorithm, all four trucks where initialized at a random vertex of their corresponding controller's RoA. The total simulated time is $T = 50$ time instances. Table 5.3 report the initial states and Figure 5.7 shows a phase plot of the trajectories of each truck.

Table 5.3: Initial state of each truck $x_i(0)$.

Truck	1	2	3	4
$x_i(0)$	$\begin{bmatrix}-1.5265\\-6.2271\end{bmatrix}$	$\begin{bmatrix}-1.6280\\-4.3328\end{bmatrix}$	$\begin{bmatrix}-1.9055\\-7.4512\end{bmatrix}$	$\begin{bmatrix}1.5431\\7.8579\end{bmatrix}$

As guaranteed by Theorem 5.1, and the discussion that followed it, the optimal cost variation (5.30) becomes upper bounded by a negative value during the first few sampling instants (see Figure 5.8). This results in that the state trajectories of all subsystems converge to the origin. This is due to $x_i(t) \in \mathbb{S}_i \subseteq \mathbb{Z}_i$ for all $i \in \mathcal{M}$ and $t \geq 30$, which results in $\hat{x}_i(t) = \hat{\hat{x}}_i(t) = \mathbf{0}$ for all $i \in \mathcal{M}$ and $t \geq 30$. In turn this implies that the overall network, in closed-loop with the proposed DMPC controller, reduces to $\boldsymbol{A} + \boldsymbol{BK}$, which is

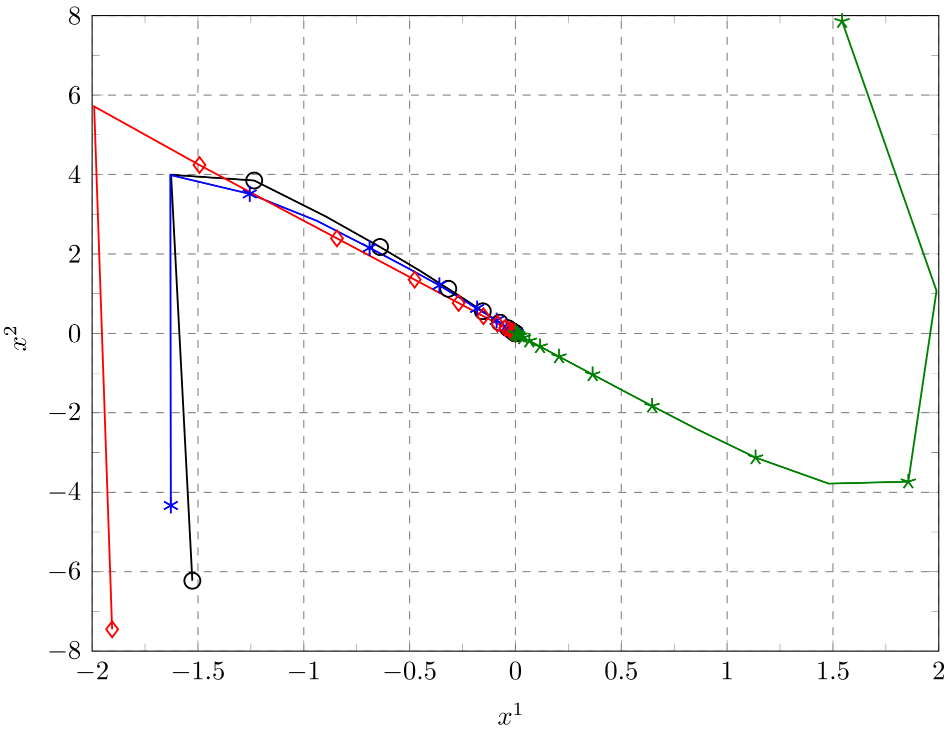

Figure 5.7: Phase plot of the trucks state trajectories: —o— $i = 1$, —∗— $i = 2$, —◇— $i = 3$, —✶— $i = 4$.

guaranteed to be stable. The latter is despite the gains K_i being chosen in a decentralized fashion, given the discussion in Section 5.5 and that the collection of RPI sets $\mathbb{Z}_i$ is admissible.

Figure 5.9 shows the input sequences used to control each truck when in closed-loop with the proposed DMPC algorithm. Again, since the true state enters the outer OCP's tightening set fairly fast, the control law of each subsystem reduces to the linear stabilizing gain $\boldsymbol{K}_i$, resulting in a convergent control input.

The main tool used in achieving recursive feasibility of the proposed approach is the consistency constraint $\mathbb{H}_i$. This, however, reduces the authority of the outer OCP, and hence raises the question of how much can the outer OCP achieve. Figure 5.10 shows the inner and outer optimized input sequences for trucks 2 and 3 at times 1 and 4 and Figure 5.11 shows the optimized state trajectories. The biggest difference between the optimal sequences of both OCPs, in state and input, is observed for truck 2, which is expected given that it is the one with strongest coupling, and hence benefits the most from the additional outer step of control. Nevertheless, as time passes and the states and inputs approach the origin, the differences are less important.

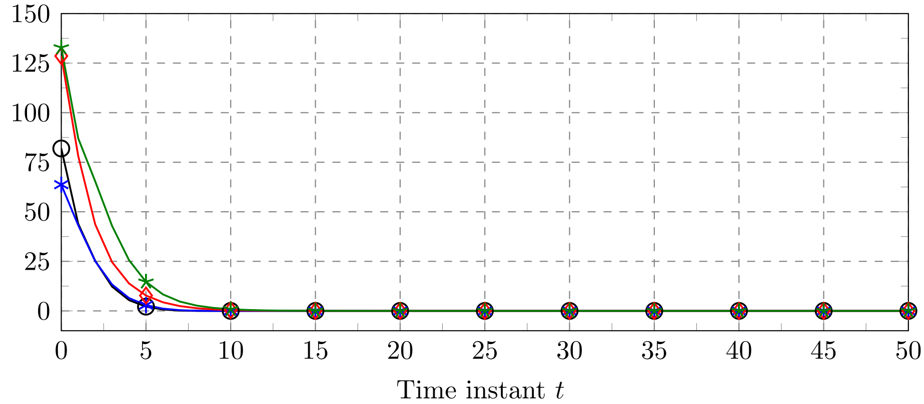

Figure 5.8: Trucks optimal costs $V^2_{N,i}(x_i(t), \boldsymbol{f}_i(t))$: $i = 1$, $i = 2$, $i = 3$, $i = 4$.

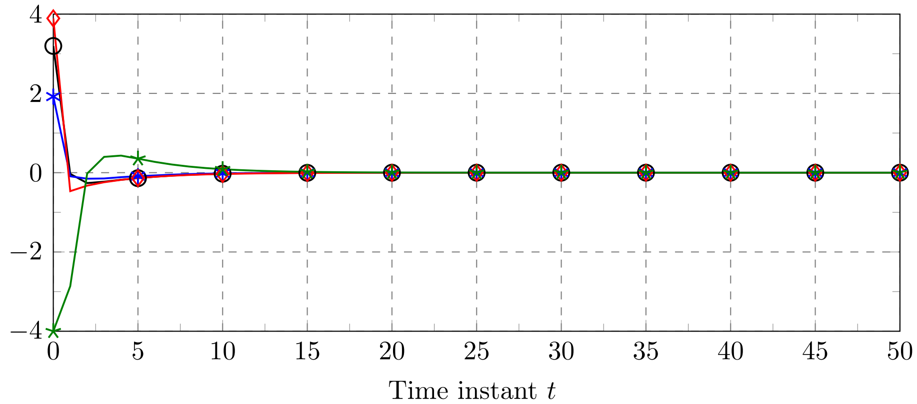

Figure 5.9: Trucks input trajectories: $i = 1$, $i = 2$, $i = 3$, $i = 4$.

Finally, in order to properly assess the performance of the proposed DMPC controller, the same network was simulated in closed-loop with three other controllers: centralized MPC, decentralized MPC and tube MPC. The first is standard centralized MPC and hence there is no need for robustness or communication. The second is decentralized MPC, that is, the loop for each truck is closed with a standard non-robust MPC controller, completely ignoring the interactions between subsystems. The last option is a standard robust approach to distribution, and compared to the DMPC algorithm proposed in this chapter, it amounts to the inner OCP without modifying the RPI constraint.

The controller parameters used in these additional simulations are, whenever possible, the same as those used for the DMPC algorithm proposed in this

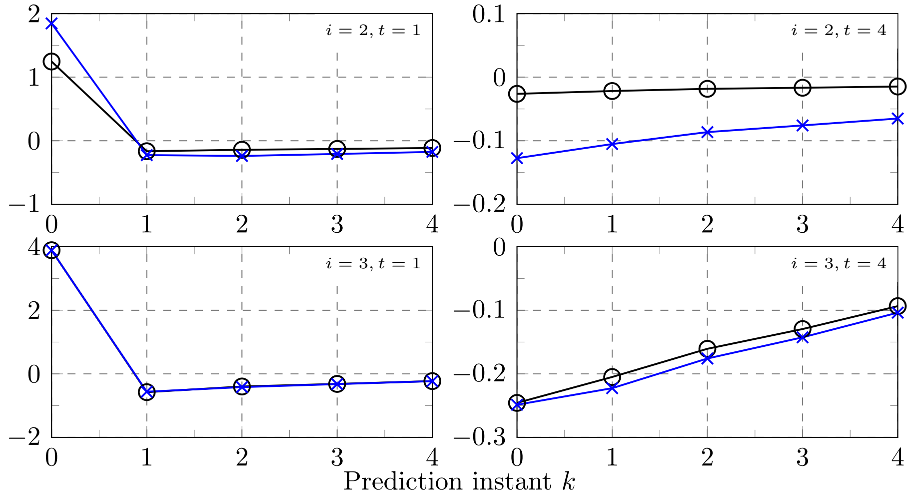

Figure 5.10: Optimized input sequence for inner and outer OCPs: —⊖— $\hat{u}_{i,k}$, —×— $\hat{\hat{u}}_{i,k}$.

chapter (i.e. the values reported in this section). This is done to obtain comparable results, however it is not always possible. In particular, the state and input cost matrices used for the centralized MPC are built as block diagonal matrices formed by the individual elements reported in this section, however the terminal cost and constraint set are computed with respect to the whole network. This is because the collection of terminal cost matrices P_i, obtained individually for each subsystem, does not fulfil the Lyapunov equation in the centralized case.

Figures 5.12 and 5.13 show a phase plot of the state trajectories of trucks 2 and 3 when in closed-loop with the different controllers. As expected, none of them are exactly the same, although the differences are more clear between the centralized MPC and the rest. The different trajectories differ more for truck 2, which is again a expected result given that the latter is the truck that experiences the higher level of coupling. It is also interesting to note that, although the decentralized controller completely neglects the interaction, is able to achieve constraint satisfaction. This is due to the inherent stability of the state associated to the position of each truck, and the nominal robustness that standard MPC enjoys [1].

Table 5.4 shows the cost associated to the closed-loop trajectories for each controller. This was computed as

$$C_i = \sum_{t=1}^{T} \left(x_i^\top(t) Q_i x_i(t) + u_i^\top(t) R_i u_i(t) \right) .$$

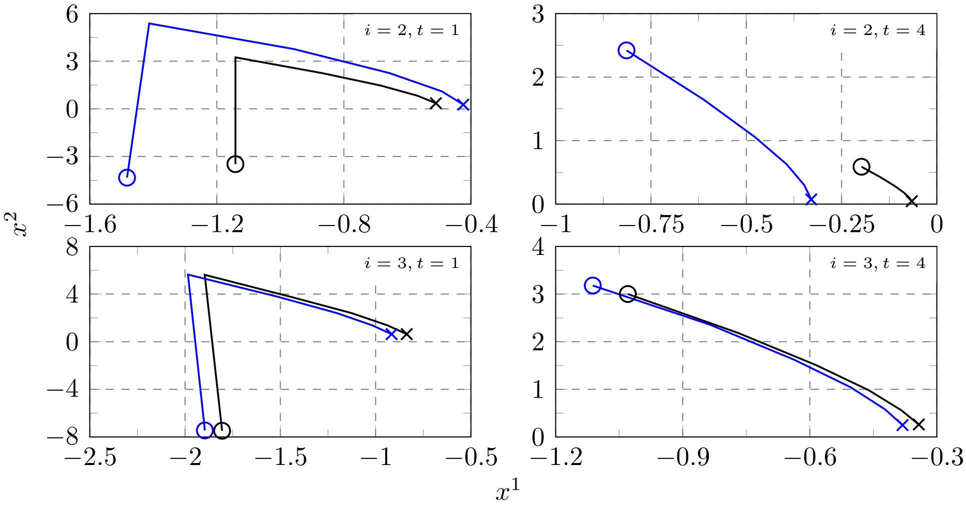

Figure 5.11: Optimized state sequence for inner and outer OCPs: initial optimized states, $\hat{x}_{i,k}$, $\hat{\hat{x}}_{i,k}$, final predicted states.

The centralized MPC is, undoubtedly, the controller that achieves the smallest running cost, due to the accuracy of the information associated to centralizing the problem. Again, the biggest variations are observed for truck 2, with Algorithm 1 outperforming both, the decentralized and tube approaches. This is also the case for truck 1, however the decentralized MPC outperforms Algorithm 1 in trucks 3 and 4. This is due to the weak coupling that those trucks are subject to. Nevertheless, the proposed DMPC algorithm yields a total cost that is smaller compared to the decentralized and tube approaches, being bested only by centralized MPC.

Table 5.4: Cost of the closed-loop trajectories C_i.

Truck	1	2	3	4	$\sum_{i=1}^{M} C_i$
Centralized MPC	179.30	155.68	271.38	287.60	893.96
Decentralized MPC	180.60	162.62	271.91	288.68	903.81
Tube MPC	180.64	161.73	272.12	289.16	903.67
Algorithm 1	179.42	156.56	272.13	288.81	896.92

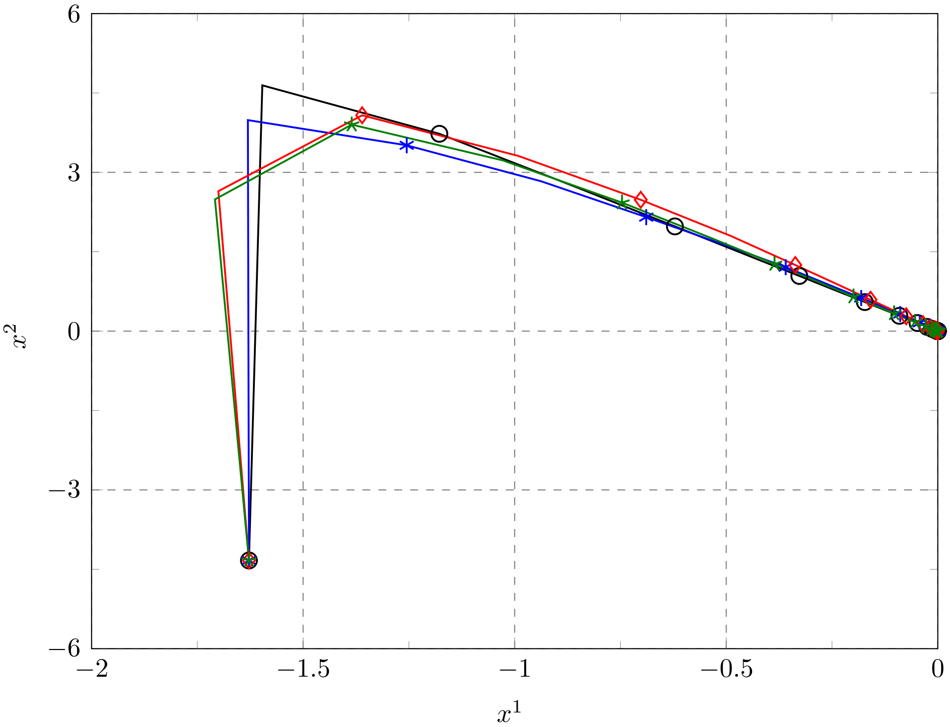

Figure 5.12: Phase plot of the state trajectory of truck $i = 2$: —○— Centralized MPC, —✶— Decentralized MPC, —◇— Tube MPC, —∗— Algorithm 1.

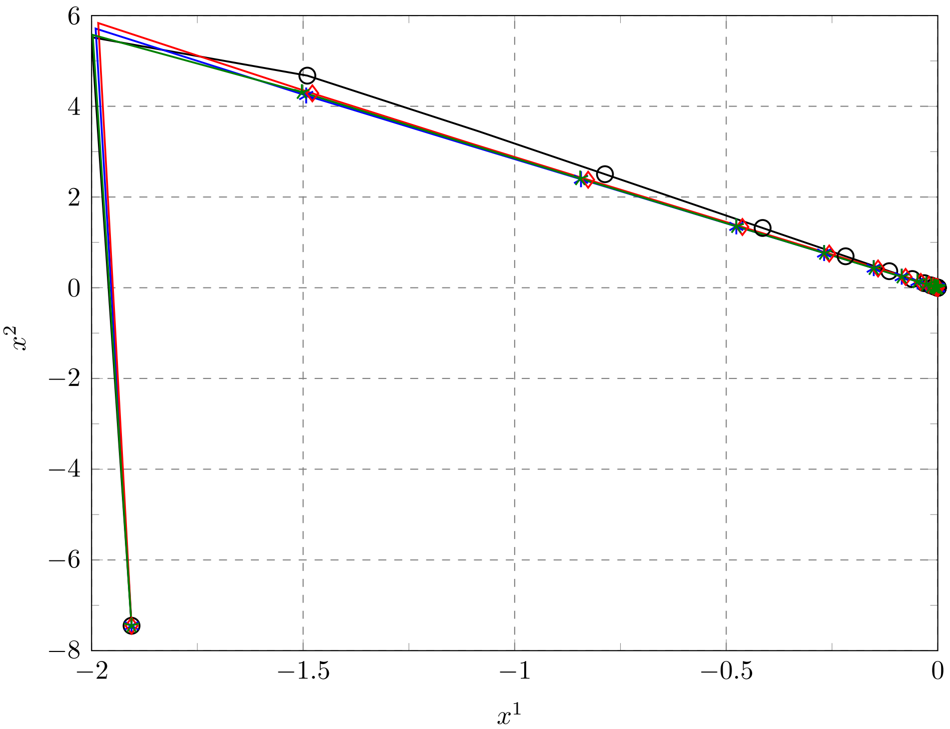

Figure 5.13: Phase plot of the state trajectory of truck $i = 3$: —○— Centralized MPC, —✶— Decentralized MPC, —◇— Tube MPC, —∗— Algorithm 1.

5.7 Summary

In this chapter a new distributed non-cooperative MPC controller was presented. The approach is robust based and its architecture is conceptually similar to other robust based approaches such as [33, 34], but more general and possibly less conservative, particularly in the size of the RoA given the utilization of tube MPC with optimizing trajectories. As expected, the two-step robust approach shows larger margins of improvement in performance as the coupling between neighbouring subsystems is strengthened. The latter however, has a limit, since feasibility of the overall approach requires the existence of an admissible RPI set for the local dynamics in presence of the disturbance set $\mathbb{W}_i$, whose size depends directly on the strength of the coupling.

One of the main sources of conservatism of the proposed DMPC controller is Assumption 5.1, which requires not only the existence of a block diagonal feedback gain that stabilizes the network, but knowledge of it. This assumption is not particular to the implementation proposed here. Indeed, it is at the core of many DMPC algorithms proposed to date, however it has not received much attention within the DMPC related literature. This assumption was investigated in detail in this chapter, concluding that, although necessary, it is a redundant requirement in many DMPC approaches. Indeed, the existence of admissible local tubes is sufficient to guarantee that the associated gains form a network-wide stabilizing feedback, and the former is a common requirement in robust approaches to DMPC.

Another source of conservatism is the need for additional tightening of the RPI constraint in the inner OCP. However, the sets $\mathbb{H}_i$ need not to be computed as minimal in any way, and the example put forward shows that good approximations of $\mathbb{Z}_i$ can be obtained in a fairly simple manner. On the other hand, the reduction in conservatism obtained by implementing the outer OCP is evident from the size of $\mathbb{S}_i$ when compared to $\mathbb{Z}_i$.

Opportunities for future work arise from the main drawback of the proposed DMPC controller, which is the complexity in accurately computing the sets $\mathbb{V}_i$. A simple solution is to include an arbitrarily defined input consistency constraint in the outer OCP. This would allow a straightforward computation of $\mathbb{V}_i$, but possibly increasing the conservativeness of the approach. On the other hand, in order to compute the outer's OCP uncertainty set comprehensively, it is necessary to characterize the set of inducing control actions associated to an arbitrary set. This is, to the author's knowledge, an open problem.

Chapter 6

Concluding Remarks

This chapter briefly summarises the work presented in Chapters 3–5 in order to highlight the main contributions presented in this thesis, and discuss avenues for future work.

6.1 Summary and contributions

The main objective of this thesis was to develop a series of MPC controllers to improve the capabilities of standard MPC in dealing with uncertain, changing and large-scale plants. The main contributions put forward in this thesis are now listed.

Adaptive Model Predictive Control

- A dual MPC controller with control and estimation guarantees was presented to control uncertain linear time varying systems. Under appropriate design, constraint satisfaction, robust stability of the control target and convergence of RLS estimates are guaranteed.

- The dual control problem was tackled by partitioning the input, an approach that, to the author's knowledge, has not been attempted in the MPC framework. This allowed to tackle both problems independently, avoiding the modification of the standard MPC control problem, which enjoys many desirable properties.

- A novel MPC-like receding horizon optimization was proposed to introduce feedback in the computation of the part of the input used to excite the system, and hence take into account current plant states and dynamics.

- The concept of persistence of excitation was thoroughly reviewed within the framework of tube based MPC and linear time varying systems. It was guaranteed that the required excitation properties can be achieved through an appropriate design.

- The problem of characterising a single linear gain that stabilizes a set of models was approached from a robust invariance perspective. It was shown that admissibility of an RPI set (robust to the parametric uncertainty) guarantees that the associated gain stabilizes the entire set of models.

Switching Model Predictive Control

- A new approach to compute dwell-times for switched linear systems, that yields an admissible and stable closed-loop when the loop of each mode is closed by a different MPC controller, was presented. The proposed approach relies on the well-known exponential stability result available for standard MPC controllers, and provides a computationally tractable way to compute the corresponding mode-dependent dwell-times, even for large scale systems.

- Both, undisturbed and perturbed systems where studied. For the latter, two possibilities were explored, independent and coupled design of local tube-based controllers. A coupled design of the tube based controllers yields a disturbed closed-loop with stability guarantees that match, in quality, those available for non-switching systems (i.e., a single mode). This is thanks to the inclusion of the multi-set invariance concept, that allows one to enhance the standard robust invariance ideas used in tube MPC without any major modification to the standard approach.

- The different modes of the switching system are allowed to be highly heterogeneous, with nothing in common between any two neighbouring modes. This poses no obstacle to the proposed approach in the non disturbed case, and in the perturbed case is easily dealt with by enlarging the region that is shown to be robustly stable, or through the inclusion of an auxiliary controller. The latter is another tube MPC controller with a different set of constraints to account for the switching between two-modes subject to non-matching constraint.

Distributed Model Predictive Control

- A new robust-based non-cooperative distributed MPC controller for a network with dynamically coupled subsystems was presented in this chapter. As with many other DMPC approaches, the proposed algorithm requires two steps of optimization and one step of communication at each sampling time, but does not require (centralized) initialization control strategies nor arbitrarily defined state trajectory targets or constraints.

- The proposed DMPC controller is guaranteed to be input-to-state stable with respect to the dynamical interaction, which is treated as a disturbance, but examples show that stabilization of the origin is indeed achieved. This is due to the utilization of the tube MPC variant that optimizes nominal state trajectories in both optimization stages.

- The second step of optimization (outer OCP) is fitted with a consistency constraint designed to guarantee recursive feasibility of the overall control algorithm. The consistency constraint is naturally defined by the feasibility requirements, rather than arbitrarily chosen. Furthermore, the concatenation of both OCPs guarantees that optimality, with respect to the current measurement, is sought at each time instant, as opposed to tracking an arbitrarily defined feasible trajectory.

- This chapter also studied a particular assumption that is often required to guarantee stability of DMPC approaches. This is a decentralized stabilizability assumption, but it can also be seen as a weak coupling assumption. The issue of finding a decentralized gain that stabilizes each subsystem, as well as the whole network, is tackled from the robust invariance perspective. Indeed, it is shown in this chapter that the existence of admissible RPI sets for each subsystem (robust to the entirety of possible dynamical interaction) guarantees that the collection of associated gains stabilizes the network. This is, essentially, the same rationale behind the search for a gain that stabilizes the entirety of models within an uncertain set in Chapter 3.

6.2 Directions for future work

In view of the results presented in this thesis, there exists several avenues for possible future work. These are now outlined.

Adaptive Model Predictive Control

- The proposed AMPC algorithm suffers from the non-convexity of the PE constraint, and thus the non-convexity of the PE optimization. There exists several convexification techniques that could be explored in order to simplify the PE optimization problem to a convex QP problem. This would allow to improve the optimality of the PE sequence with respect to the current plant state and prediction model.

- Another important direction of future work is related to improving the control performance once the parameters have been accurately estimated. If the latter is indeed the case, and the plant is not experiencing any immediate change, the exciting sequence loses its purpose, and prevents perfect regulation to be achieved.

Switching Model Predictive Control

- With respect to the proposed switching systems framework, a direction for future work lies on the assumption of instantaneous detection of the switch. This is a key assumption of the proposed approach to compute MDTs, since the latter can only guarantee admissible and stable switching if the correct MPC controller is active. Future work then needs to focus on the case in which the switch is detected with some delay, what are the effects of this on the closed-loop performance and what can be done to mitigate them (e.g. robustify the design to account for a maximum delay).

- Another clear direction for future work is in the estimation of a tighter upper bound for the MPC Lyapunov function. Indeed, the illustrative example showed that useful MDTs can be computed only with a tight upper bound, however the guaranteed one is loose, resulting in real-world impractical MDTs.

Distributed Model Predictive Control

- When it comes to the design stage of the proposed DMPC controller, the main open issue is the proper characterization of the disturbance set for the outer OCP. The set that contains the state trajectory error is easily defined, but computing the corresponding set for the input trajectory error is not as straightforward. In order to do so, it is necessary to characterise the set of control actions that render an arbitrary set invariant, which is a non-trivial problem.

Bibliography

[1] James B. Rawlings and David Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill, Madison, Wisconsin, electronic edition, 2014.

[2] D. Q. Mayne, M. M. Seron, and S. V. Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2):219–224, 2005.

[3] Lennart Ljung. *System Identification Theory for the user*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1999.

[4] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

[5] Jan M. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, Harlow, 1st edition, 2002.

[6] A. M. Lyapunov. *The General Problem of the Stability of Motion*. Taylor & Francis Ltd., London, 1992.

[7] S. S. Keerthi and E. G. Gilbert. Optimal Infinite-Horizon Feedback Laws for a General Class of Constrained Discrete-Time Systems: Stability and Moving-Horizon Approximations. *Journal of Optimization Theory and Applications*, 57(2):265–293, 1988.

[8] James B. Rawlings and Kenneth R. Muske. The Stability of Constrained Receding Horizon Control. *IEEE Transactions on Automatic Control*, 38(10):1512–1516, 1993.

[9] Pierre O. M. Scokaert and James B. Rawlings. Constrained Linear Quadratic Regulation. *IEEE Transactions on Automatic Control*, 43(8):1163–1169, 1998.

[10] David Q. Mayne and Hannah Michalska. Receding Horizon Control of Nonlinear Systems. *IEEE Transactions on Automatic Control*, 35(7):814–824, jul 1990.

[11] Mario Sznaier and Mark J. Damborg. Suboptimal control of linear systems with state and control inequality constraints. In *Proceedings of the 26th Conference on Decision and Control*, volume 26, pages 761–762, Los Angeles, CA, 1987.

[12] H. Michalska and D. Q. Mayne. Robust Receding Horizon Control of Constrained Nonlinear Systems. *IEEE Transactions on Automatic Control*, 38(11):1623–1633, 1993.

[13] T. Parisini and R. Zoppoli. A Receding-Horizon Regulator for Nonlinear Systems and a Neural Approximation. *Automatica*, 31(10):1443–1451, 1995.

[14] G. De Nicolao, L. Magni, and R. Scattolini. Stabilizing Receding-Horizon Control of Nonlinear Time-Varying Systems. *IEEE Transactions on Automatic Control*, 43(7):1030–1036, 1998.

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

[16] L. Magni, G. De Nicolao, L. Magnani, and R. Scattolini. A stabilizing model-based predictive control algorithm for nonlinear systems. *Automatica*, 37(9):1351–1362, 2001.

[17] Nael H. El-Farra, Prashant Mhaskar, and Panagiotis D. Christofides. Uniting bounded control and MPC for stabilization of constrained linear systems. *Automatica*, 40(1):101–110, jan 2004.

[18] Prashant Mhaskar, Nael H. El-Farra, and Panagiotis D. Christofides. Predictive Control of Switched Nonlinear Systems With Scheduled Mode Transitions. *IEEE Transactions on Automatic Control*, 50(11):1670–1680, nov 2005.

[19] Prashant Mhaskar, Nael H. El-Farra, and Panagiotis D. Christofides. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. *Systems & Control Letters*, 55(8):650–659, aug 2006.

[20] Shuyou Yu, Marcus Reble, Hong Chen, and Frank Allgöwer. Inherent robustness properties of quasi-infinite horizon nonlinear model predictive control. *Automatica*, 50(9):2269–2280, sep 2014.

[21] Eric C. Kerrigan and Jan M. Maciejowski. Robust feasibility in model predictive control: Necessary and sufficient conditions. In *Proccedings of the 40th IEEE Conference on Decision and Control*, pages 728–733, 2001.

[22] P. O. M. Scokaert and D. Q. Mayne. Min-max feedback model predictive control for constrained linear systems. *IEEE Transactions on Automatic Control*, 43(8):1136–1142, 1998.

[23] L. Chisci, J. A. Rossiter, and G. Zappa. Systems with persistent disturbances: predictive control with restricted constraints. *Automatica*, 37:1019–1028, 2001.

[24] B. Kouvaritakis, J. A. Rossiter, and J. Schuurmans. Efficient Robust Predictive Control. *IEEE Transactions on Automatic Control*, 45(8):1545–1549, 2000.

[25] D. Q. Mayne and W. Langson. Robustifying model predictive control of constrained linear systems. *Electronics Letters*, 37(23):1422–1423, 2001.

[26] Mayuresh V. Kothare, Venkataramanan Balakrishnan, and Manfred Morari. Robust Constrained Model Predictive Control Using Linear Matrix Inequalities. *Automatica*, 32(10):1361–1379, 1996.

[27] Paul J. Goulart, Eric C. Kerrigan, and Jan M. Maciejowski. Optimization over state feedback policies for robust control with constraints. *Automatica*, 42(4):523–533, 2006.

[28] L. Imsland, J. A. Rossiter, B. Pluymers, and J. Suykens. Robust triple mode MPC. *International Journal of Control*, 81(4):679–689, apr 2006.

[29] D. Q. Mayne, E. C. Kerrigan, E. J. Van Wyk, and P. Falugi. Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, 21(11):1341–1353, jul 2011.

[30] Paul A. Trodden and J. M. Maestre. Distributed predictive control with minimization of mutual disturbances. *Automatica*, 77:31–43, 2017.

[31] Paul Trodden. Feasible parallel-update distributed MPC for uncertain linear systems sharing convex constraints. *Systems & Control Letters*, 74:98–107, 2014.

[32] Christian Conte, Melanie N. Zeilinger, Manfred Morari, and Colin N. Jones. Robust Distributed Model Predictive Control of Linear Systems. In *Proceedings of the 2013 European Control Conference*, pages 2764–2769, Zürich, 2013. IEEE.

[33] Stefano Riverso and Giancarlo Ferrari-Trecate. Tube-based distributed control of linear constrained systems. *Automatica*, 48(11):2860–2865, nov 2012.

[34] Marcello Farina and Riccardo Scattolini. Distributed predictive control: A non-cooperative algorithm with neighbor-to-neighbor communication for linear systems. *Automatica*, 48(6):1088–1096, jun 2012.

[35] Pablo R. Baldivieso Monasterios, Bernardo Hernandez, and Paul A. Trodden. Nested distributed MPC. *IFAC-PapersOnLine*, 50(1):11822–11828, 2017.

[36] Pablo R. Baldivieso Monasterios and Paul Trodden. Low-Complexity Distributed Predictive Automatic Generation Control with Guaranteed Properties. *IEEE Transactions on Smart Grid*, 8(6):3045–3054, 2017.

[37] Sasa V. Rakovic, William S. Levine, and Behcet Acikmese. Discretely generalized model predictive control. In *Proceedings of the 2016 American Control Conference*, pages 616–621, Boston, MA, 2016. IEEE.

[38] Sasa V. Rakovic, William S. Levine, and Behcet Acikmese. Elastic tube model predictive control. In *Proceedings of the 2016 American Control Conference*, pages 3594–3599, Boston, MA, 2016. IEEE.

[39] Veronica Adetola and Martin Guay. Robust adaptive MPC for constrained uncertain nonlinear systems. *International Journal of Adaptive Control and Signal Processing*, 25(2):155–167, 2011.

[40] Anil Aswani, Humberto Gonzalez, S. Shankar Sastry, and Claire Tomlin. Provably safe and robust learning-based model predictive control. *Automatica*, 49(5):1216–1226, may 2013.

[41] O. J. Dellar and B. Ll. Jones. Dynamically correct formulations of the linearised Navier-Stoke equations. *International Journal for Numerical Methods in Fluids*, 85:601–629, 2008.

[42] Eva Žáčeková, Zdeněk Váňa, and Jiří Cigler. Towards the real-life implementation of MPC for an office building: Identification issues. *Applied Energy*, 135:53–62, 2014.

[43] Christian A. Larsson, Cristian R. Rojas, Xavier Bombois, and Håkan Hjalmarsson. Experimental evaluation of model predictive control with excitation (MPC-X) on an industrial depropanizer. *J. Process. Contr.*, 31:1–16, 2015.

[44] Premkiran Vuthandam and Michael Nikolaou. Constrained MPCI: A Weak Persistent Excitation Approach. *AIChE Journal*, 43(9):2279–2288, 1997.

[45] David Q. Mayne. Model predictive control: Recent developments and future promise. *Automatica*, 50(12):2967–2986, nov 2014.

[46] Mayuresh V. Kothare, Bernard Mettler, Manfred Morari, Pascale Bendotti, and Clément-Marc Falinower. Level control in the steam generator of a nuclear power plant. *IEEE Transactions on Control Systems Technology*, 8(1):55–69, 2000.

[47] Xionglin Luo, Xin Zuo, and Dianlin Du. Varying Model Based Adaptive Predictive Control of Highly Nonlinear Chemical Process. In *Proceedings of the 2005 International Conference on Control and Automation*, pages 537–540, Budapest, 2005. IEEE.

[48] M. Fiacchini, T. Alamo, C. Albea, and E. F. Camacho. Adaptive model predictive control of the hybrid dynamics of a fuel cell system. In *Proceedings of the 16th IEEE International Conference on Control Applications*, pages 1420–1425, Singapore, 2007. IEEE.

[49] Guolian Hou, Jinfang Zhang, Junjun Liu, and Jianhua Zhang. Multiple-model Predictive Control Based on Fuzzy Adaptive Weights and its Application to Main-Steam Temperature in Power Plant. In *Proceedings of the 5th IEEE Conference on Industrial Electronics and Applications*, pages 668–673, Taichung, jun 2010. IEEE.

[50] P. Falcone, F. Borrelli, H. E. Tseng, J. Asgari, and D. Hrovat. Linear time-varying model predictive control and its application to active steering systems: Stability analysis and experimental validation. *International Journal of Robust and Nonlinear Control*, 18(8):862–875, 2008.

[51] Akbar Rahideh, M. Hasan Shaheed, and Henry J. C. Huijberts. Stable adaptive model predictive control for nonlinear systems. In *Proceedings of the 2008 American Control Conference*, pages 1673–1678, Seattle, WA, jun 2008. IEEE.

[52] Vladimir Bobal, Marek Kubalcik, Petr Dostal, and Jakub Matejicek. Adaptive predictive control of time-delay systems. *Computers and Mathematics with Applications*, 66(2):165–176, aug 2013.

[53] Jerry Mamboundou and Nicolas Langlois. Indirect Adaptive Model Predictive Control Supervised by Fuzzy Logic. In *Proceedings of the 2011 International Conference on Fuzzy Systems*, pages 2979–2986, Taipei, jun 2011. IEEE.

[54] N. A. Wahab, R. Katebi, J. Balderud, and M.F. Rahmat. Data-driven adaptive model-based predictive control with application in wastewater systems. *IET Control Theory and Applications*, 5(6):803–812, apr 2011.

[55] Weng Caihao, Zhang Xiaowu, and Sun Jing. Adaptive Model Predictive Control for Hybrid Electric Vehicles Power Management. In *Proceedings of the 32nd Chinese Control Conference*, pages 7756–7761, Xi'an, 2013. IEEE.

[56] Mouhacine Benosman, Stefano Di Cairano, and Avishai Weiss. Extremum Seeking-based Iterative Learning Linear MPC. In *Proceedings of the 2014 IEEE Conference on Control Applications*, pages 1849–1854, Antibes, 2014. IEEE.

[57] Vincent A. Akpan, Ioakeim K. Samaras, and George D. Hassapis. Implementation of Neural Network-Based Nonlinear Adaptive Model Predictive Control over a Service-Oriented Computer Network. In *Proceedings of the 2010 American Control Conference*, pages 5495–5500, Baltimore, MD, 2010. IEEE.

[58] Avishai Weiss and Stefano Di Cairano. Robust Dual Control MPC with Guaranteed Constraint Satisfaction. In *Proceedings of the 53rd Conference on Decision and Control*, pages 6713–6718, Los Angeles, CA, 2014. IEEE.

[59] Hiroaki Fukushima, Tae-Hyoung Kim, and Toshiharu Sugie. Adaptive model predictive control for a class of constrained linear systems based on the comparison model. *Automatica*, 43(2):301–308, feb 2007.

[60] Veronica Adetola, Darryl DeHaan, and Martin Guay. Adaptive model predictive control for constrained nonlinear systems. *Systems & Control Letters*, 58(5):320–326, may 2009.

[61] Hiroaki Fukushima and Robert R. Bitmead. Robust constrained predictive control using comparison model. *Automatica*, 41(1):97–106, 2005.

[62] Guilherme A. A. Gonçalves and Martin Guay. Robust discrete-time set-based adaptive predictive control for nonlinear systems. *Journal of Process Control*, 39:111–122, 2016.

[63] Marko Tanaskovic, Lorenzo Fagiano, Roy Smith, and Manfred Morari. Adaptive receding horizon control for constrained MIMO systems. *Automatica*, 50(12):3019–3029, dec 2014.

[64] Xiaofeng Wang, Yu Sun, and Kun Deng. Adaptive model predictive control of uncertain constrained systems. In *Proceedings of the 2014 American Control Conference*, pages 2857–2862, Portland, OR, jun 2014. IEEE.

[65] Stefano Di Cairano. Indirect Adaptive Model Predictive Control for Linear Systems with Polytopic Uncertainty. In *Proceedings of the 2016 American Control Conference*, pages 3570–3575, 2016.

[66] Donglei Fan and Stefano Di Cairano. Further Results and Properties of Indirect Adaptive Model Predictive Control for Linear Systems with Polytopic Uncertainty. In *Proceedings of the 2016 American Control Conference*, pages 2948–2953, 2016.

[67] Y. Cheng, S. Haghighat, and S. Di Cairano. Robust dual control MPC with application to soft-landing control. In *Proceedings of the 2015 American Control Conference*, pages 3862–3867, Chicago, IL, 2015. IEEE.

[68] D. Limon, J. Calliess, and J. M. Maciejowski. Learning-based Nonlinear Model Predictive Control. *IFAC-PapersOnLine*, 50(1):7769–7776, 2017.

[69] A. H. González, A. Ferramosca, G. A. Bustos, J. L. Marchetti, and D. Odloak. Model predictive control suitable for closed-loop re-identification. *Systems & Control Letters*, 69:23–33, 2014.

[70] A. Ferramosca, D. Limon, A. H. González, D. Odloak, and E. F. Camacho. MPC for tracking zone regions. *Journal of Process Control*, 20(4):506–516, 2010.

[71] Alejandro H. González and Darci Odloak. A stable MPC with zone control. *Journal of Process Control*, 19(1):110–122, 2009.

[72] A. Anderson, A. H. González, A. Ferramosca, A. D'Jorge, and E. Kofman. Robust MPC suitable for closed-loop re-identification, based on probabilistic invariant sets. *Systems & Control Letters*, 118:84–93, 2018.

[73] Matthias Lorenzen, Frank Allgöwer, and Mark Cannon. Adaptive Model Predictive Control with Robust Constraint Satisfaction. *IFAC-PapersOnLine*, 50(1):3313–3318, 2017.

[74] Giancarlo Marafioti, Robert R. Bitmead, and Morten Hovd. Persistently exciting model predictive control. *International Journal of Adaptive Control and Signal Processing*, 28(6):536–552, 2014.

[75] Graham C Goodwin and Kwai Sang Sin. *Adaptive filtering, prediction and control*. Prentice Hall, Englewoods Cliffs, NJ, 1st edition, 1984.

[76] Karl J. Åström and Bjorn Wittenmark. *Adaptive Control*. Dover Publications Inc., Mineola, NY, 2nd edition, 2008.

[77] Hasmet Genceli and Michael Nikolaou. New Approach to Constrained Predictive Control with Simultaneous Model Identification. *AIChE Journal*, 42(10):2857–2868, 1996.

[78] Manoj Shouche, Hasmet Genceli, Premkiran Vuthandam, and Michael Nikolaou. Simultaneous Constrained Model Predictive Control and Identification of DARX Processes. *Automatica*, 34(12):1521–1530, 1998.

[79] Jan Rathouský and Vladimír Havlena. Multiple-step active control with dual properties. In *Proceedings of the 18th IFAC World Congress*, volume 18, pages 1522–1527, Milano, 2011.

[80] Jan Rathouský and Vladimir Havlena. MPC-based approximate dual controller by information matrix maximization. *International Journal of Adaptive Control and Signal Processing*, 27(11):974–999, 2013.

[81] Eva Žáčeková, Samuel Prívara, and Josef Komárek. On dual control for buildings using persistent excitation condition. In *Proceedings of the 51st*

IEEE Conference on Decision and Control, pages 2158–2163, Maui, HI, 2012. IEEE.

[82] Eva Žáčeková, Samuel Prívara, and Matej Pčolka. Persistent excitation condition within the dual control framework. *Journal of Process Control*, 23(9):1270–1280, 2013.

[83] Eva Žačeková, Matej Pčolka, Sergej Čelikovský, and Michael Sebek. Semi-receding Horizon Algorithm for "Sufficiently Exciting" MPC with Adaptive Search Step. In *Proceedings of the 53rd IEEE Conference on Decision and Control*, pages 4142–4147, Los Angeles, CA, 2014. IEEE.

[84] Vinay A. Bavdekar and Ali Mesbah. Stochastic Model Predictive Control with Integrated Experiment Design for Nonlinear Systems. *IFAC-PapersOnLine*, 49(7):49–54, 2016.

[85] Tor Aksel N. Heirung, Bjarne Foss, and B. Erik Ydstie. MPC-based dual control with online experiment design. *Journal of Process Control*, 32:64–76, 2015.

[86] Tor Aksel N. Heirung, B. Erik Ydstie, and Bjarne Foss. Dual adaptive model predictive control. *Automatica*, 80:340–348, 2017.

[87] Jurre Hanema, Mircea Lazar, and Roland Tóth. Stabilizing tube-based model predictive control: Terminal set and cost construction for LPV systems. *Automatica*, 85:137–144, 2017.

[88] D. Q. Mayne, S. V. Raković, R. Findeisen, and F. Allgöwer. Robust output feedback model predictive control of constrained linear systems. *Automatica*, 42(7):1217–1222, 2006.

[89] Michael Green and John B. Moore. Persistence of excitation in linear systems. *Systems & Control Letters*, 7(5):351–360, 1986.

[90] E. W. Bai and S. S. Sastry. Persistency of excitation, sufficient richness and parameter convergence in discrete time adaptive control. *Systems & Control Letters*, 6(3):153–163, 1985.

[91] Michael Green and John B. Moore. Persistence of excitation in linear systems. In *Proceedings of the 1985 American Control Conference*, pages 412–417, Boston, MA, 1985. IEEE.

[92] Jamal Daafouz, Pierre Riedinger, and Claude Iung. Stability analysis and control synthesis for switched systems: a switched Lyapunov function approach. *IEEE Transactions on Automatic Control*, 47(11):1883–1887, 2002.

[93] Toni Barjas Blanco, Mark Cannon, and Bart De Moor. On efficient computation of low-complexity controlled invariant sets for uncertain linear systems. *International Journal of Control*, 83(7):1339–1346, 2010.

[94] K. I. Kouramas, S. V. Raković, E. C. Kerrigan, J. C. Allwright, and D. Q. Mayne. On the minimal robust positively invariant set for linear difference inclusions. *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 2296–2301, 2005.

[95] John J. Martínez. Minimal RPI sets computation for polytopic systems using the Bounded-real lemma and a new shrinking procedure. *IFAC-PapersOnLine*, 48(26):182–187, 2015.

[96] Ilya Kolmanovsky and Elmer G. Gilbert. Theory and Computation of Disturbance Invariant Sets for Discrete-Time Linear Systems. *Mathematical Problems in Engineering*, 4:317 – 367, 1998.

[97] Eric C. Kerrigan. *Robust constraint satisfaction: Invariant sets and predictive control*. PhD thesis, University of Cambridge, 2000.

[98] Elmer G. Gilbert and Kok Tin Tan. Linear Systems with State and Control Constraints : The Theory and Application of Maximal Output Admissible Sets. *IEEE Transactions on Automatic Control*, 36(9):1008–1020, 1991.

[99] Franco Blanchini. Ultimate boundness control for uncertain discretetime systems via set-induced Lyapunov functions. *IEEE Transactions on Automatic Control*, 39(2):428–433, 1994.

[100] S. V. Rakovic, E. C. Kerrigan, K. I. Kouramas, and D. Q. Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005.

[101] Paul Trodden. A One-Step Approach to Computing a Polytopic Robust Positively Invariant Set. *IEEE Transactions on Automatic Control*, 61(12):4100–4105, 2016.

[102] Yuanqi Mao, Daniel Dueri, Michael Szmuk, and Behçet Açıkmeşe. Successive Convexification of Non-Convex Optimal Control Problems with State Constraints. *IFAC-PapersOnLine*, 50(1):4063–4069, 2017.

[103] Ying Jin, Jun Fu, Youmin Zhang, and Yuanwei Jing. Reliable control of a class of switched cascade nonlinear systems with its application to flight control. *Nonlinear Analysis: Hybrid Systems*, 11(1):11–21, 2014.

[104] Iman Nodozi and Mehdi Rahmani. LMI-based model predictive control for switched nonlinear systems. *Journal of Process Control*, 59:49–58, 2017.

[105] K. Hariprasad and Sharad Bhartiya. A computationally efficient robust tube based MPC for linear switched systems. *Nonlinear Analysis: Hybrid Systems*, 19:60–76, 2016.

[106] Panos J. Antsaklis and Xenofon D. Koutsoukos. Hybrid Systems: Review and recent progress. In *Software–Enabled Control: Information Technology for Dynamical Systems*, chapter 14, pages 273–298. IEEE, 1 edition, 2003.

[107] Shuai Yuan, Bart De Schutter, and Simone Baldi. Robust adaptive tracking control of uncertain slowly switched linear systems. *Nonlinear Analysis: Hybrid Systems*, 27:1–12, 2018.

[108] Franco Blanchini, Daniele Casagrande, and Stefano Miani. Modal and transition dwell time computation in switching systems: A set-theoretic approach. *Automatica*, 46(9):1477–1482, 2010.

[109] Nael H. El-Farra, Prashant Mhaskar, and Panagiotis D. Christofides. Output feedback control of switched nonlinear systems using multiple Lyapunov functions. *Systems and Control Letters*, 54(12):1163–1182, 2005.

[110] Patrizio Colaneri and Riccardo Scattolini. *Robust Model Predictive Control of Discrete-Time Switched Systems*, volume 40. IFAC, 2007.

[111] Francesco Borrelli, Mato Baotić, Alberto Bemporad, and Manfred Morari. Dynamic programming for constrained optimal control of discrete-time linear hybrid systems. *Automatica*, 41(10):1709–1721, 2005.

[112] Prashant Mhaskar, Nael H. El-Farra, and Panagiotis D. Christofides. Robust predictive control of switched systems: Satisfying uncertain schedules subject to state and control constraints. *International Journal of Adaptive Control and Signal Processing*, 22(4):161–179, 2008.

[113] Robert Shorten, Fabian Wirth, Oliver Mason, Kai Wulff, and Christopher King. Stability Criteria for Switched and Hybrid Systems. *SIAM Review*, 49(4):545–592, 2007.

[114] Lixian Zhang, Songlin Zhuang, and Richard D. Braatz. Switched model predictive control of switched linear systems: Feasibility, stability and robustness. *Automatica*, 67:8–21, 2016.

[115] Qian Sang and Gang Tao. Adaptive control of piecewise linear systems: The state tracking case. *IEEE Transactions on Automatic Control*, 57(2):522–528, 2012.

[116] J. C. Geromel and P. Colaneri. Stability and stabilization of discrete time switched systems. *International Journal of Control*, 79(7):719–728, 2006.

[117] Graziano Chesi, Patrizio Colaneri, Jose C. Geromel, Richard Middleton, and Robert Shorten. A nonconservative LMI condition for stability of switched systems with guaranteed dwell time. *IEEE Transactions on Automatic Control*, 57(5):1297–1302, 2012.

[118] Franco Blanchini, Stefano Miani, and Carlo Savorgnan. Stability results for linear parameter varying and switching systems. *Automatica*, 43(10):1817–1823, 2007.

[119] Matthew Philippe, Ray Essick, Geir E. Dullerud, and Raphaël M. Jungers. Stability of discrete-time switching systems with constrained switching sequences. *Automatica*, 72:242–250, 2016.

[120] A. P. Molchanov and Ye S. Pyatnitskiy. Criteria of asymptotic stability of differential and difference inclusions encountered in control theory. *Systems & Control Letters*, 13(1):59–64, 1989.

[121] Michael S. Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

[122] Lixian Zhang, Songlin Zhuang, Peng Shi, and Yanzheng Zhu. Uniform tube based stabilization of switched linear systems with mode-dependent persistent dwell-time. *IEEE Transactions on Automatic Control*, 60(11):2994–2999, 2015.

[123] Walter Lucia and Giuseppe Franzè. Stabilization and reference tracking for constrained switching systems: A predictive control approach. *International Journal of Adaptive Control and Signal Processing*, 31(12):1871–1884, 2017.

[124] Chong-Jin Ong, Zheming Wang, and Masood Dehghan. Model Predictive Control for Switching System with Dwell-time Restriction. *IEEE Transactions on Automatic Control*, 61(12):4189–4195, 2016.

[125] Leila Jasmine Bridgeman, Claus Danielson, and Stefano Di Cairano. Stability and feasibility of MPC for switched linear systems with dwell-time constraints. In *Proceedings of the 2016 American Control Conference*, pages 2681–2686, Seattle, WA, 2016. IEEE.

[126] Claus Danielson, Leila Bridgeman, and Stefano Di Cairano. Constraint Satisfaction for Switched Linear Systems with Restricted Dwell-Time. In *Proceedings of the 2017 American Control Conference*, pages 3682–3687, Seattle, WA, 2017. IEEE.

[127] Matthias A. Müller, Pascal Martius, and Frank Allgöwer. Model predictive control of switched nonlinear systems under average dwell-time. *Journal of Process Control*, 22(9):1702–1710, 2012.

[128] Sven Knüfer, Matthias A. Müller, and Frank Allgöwer. Stabilizing Model Predictive Control without Terminal Constraints for Switched Nonlinear Systems. *IFAC-PapersOnLine*, 49(18):65–70, 2016.

[129] Giuseppe Franzè and Walter Lucia. A set-theoretic control architecture for constrained switching systems. In *Proceedings of the 2016 American Control Conference*, pages 685–690, Boston, MA, 2016. IEEE.

[130] Chong-Jin Ong, Zheming Wang, and Masood Dehghan. Characterization of switching sequences on system with dwell-time restriction for Model Predictive Control. In *Proceedings of the 54th IEEE Conference on Decision and Control*, pages 3657–3662, Osaka, 2015.

[131] Masood Dehghan and Chong-Jin Ong. Discrete-time switching linear system with constraints: Characterization and computation of invariant sets under dwell-time consideration. *Automatica*, 48(9):2175–2181, 2012.

[132] Qian Sang and Gang Tao. Adaptive Control of Piecewise Linear Systems : the State Tracking Case. In *Proceedings of the 2010 American Control Conference*, pages 4040–4045, 2010.

[133] Lixian Zhang and Richard D. Braatz. On switched MPC of a class of switched linear systems with modal dwell time. In *Proceedings of the IEEE 52nd Conference on Decision and Control*, pages 91–96. IEEE, dec 2013.

[134] Alberto Bemporad, Francesco Borrelli, and Manfred Morari. Model predictive control based on linear programming - the explicit solution. *IEEE Transactions on Automatic Control*, 47(12):1974–1985, 2002.

[135] Masood Dehghan and Chong-jin Ong. Characterization and computation of disturbance invariant sets for constrained switched linear systems with dwell time restriction. *Automatica*, 48(9):2175–2181, sep 2012.

[136] N. Athanasopoulos, K. Smpoukis, and R. M. Jungers. Invariant Sets Analysis for Constrained Switching Systems. *IEEE Control Systems Letters*, 1(2):256–261, 2017.

[137] D. P. Bertsekas and I. B. Rhodes. On the minimax reachability of target sets and target tubes. *Automatica*, 7(2):233–247, 1971.

[138] Sasa V. Raković, Benjamin Kern, and Rolf Findeisen. Practical Set Invariance for Decentralized Discrete Time Systems. In *Proceedings of the 49th IEEE Conference on Decision and Control*, pages 3283–3288, Atlanta, GA, 2010. IEEE.

[139] Riccardo Scattolini. Architectures for distributed and hierarchical Model Predictive Control – A review. *Journal of Process Control*, 19(5):723–731, may 2009.

[140] S. Bhattacharyya, J. M. A. Myrzik, and W. L. Kling. Consequences of poor power quality - An overview. In *Proceedings of the Universities Power Engineering Conference*, pages 651–656, 2007.

[141] B. Erik Ydstie. New vistas for process control: Integrating physics and communication networks. *AIChE Journal*, 48(3):422–426, 2002.

[142] Panagiotis D. Christofides, James F. Davis, Nael H. El-Farra, Don Clark, Kevin R. D. Harris, and Jerry N. Gipson. Smart plant operations: Vision, progress and challenges. *AIChE Journal*, 53(11):2734–2741, 2007.

[143] Panagiotis D. Christofides, Riccardo Scattolini, David Muñoz de la Peña, and Jinfeng Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers and Chemical Engineering*, 51:21–41, apr 2013.

[144] E. E. Ejegi, J. A. Rossiter, and P. Trodden. A survey of techniques and opportunities in power system automatic generation control. In *Proceedings of the 2014 UKACC International Conference on Control*, pages 537–542, 2014.

[145] John Baillieul and Panos J. Antsaklis. Control and Communication Challenges in Networked Real-Time Systems. *Proceedings of the IEEE*, 95(1):9–28, jan 2007.

[146] Valerio Turri, Bart Besselink, and Karl H. Johansson. Cooperative look-ahead control for fuel-efficient and safe heavy-duty vehicle platooning. *IEEE Transactions on Systems Technology*, 25(1):12–28, 2015.

[147] Le Li, Rudy R. Negenborn, and Bart De Schutter. Distributed model predictive control for cooperative synchromodal freight transport. *Transportation Research Part E*, 105:240–260, 2017.

[148] Yoshiaki Kuwata, Arthur Richards, Tom Schouwenaars, and Jonathan P. How. Distributed robust receding horizon control for multivehicle guidance. *IEEE Transactions on Control Systems Technology*, 15(4):627–641, 2007.

[149] William B. Dunbar and Derek S. Caveney. Distributed receding horizon control of vehicle platoons: Stability and string stability. *IEEE Transactions on Automatic Control*, 57(3):620–633, 2012.

[150] J. M. Maestre and R. R. Negenborn. *Distributed Model Predictive Control Made Easy*. Springer Netherlands, 1 edition, 2014.

[151] Hong Cui and Elling W. Jacobsen. Performance limitations in decentralized control. *Journal of Process Control*, 12(4):485–494, jun 2002.

[152] Levent Acar. Some examples for the decentralized receding horizon control. In *Proceedings of the 31st Conference on Decision and Control*, pages 1356–1359, Tucson, AZ, 1992. IEEE.

[153] Eduardo Camponogara, Dong Jia, Bruce H. Krogh, and Sarosh Talukdar. Distributed model predictive control. *IEEE Control Systems Magazine*, 22(1):44–52, 2002.

[154] Xu Cheng and Bruce H. Krogh. Stability-constrained model predictive control. *IEEE Transactions on Automatic Control*, 46(11):1816–1820, 2001.

[155] L. Magni and R. Scattolini. Stabilizing decentralized model predictive control of nonlinear systems. *Automatica*, 42(7):1231–1236, jul 2006.

[156] Simone Loureiro de Oliveira and Manfred Morari. Contractive model predictive control for constrained nonlinear systems. *IEEE Transactions on Automatic Control*, 45(6):1053–1071, 2000.

[157] D. M. Raimondo, L. Magni, and R. Scattolini. Decentralized MPC of nonlinear systems: An input-to-state stability approach. *International Journal of Robust and Nonlinear Control*, 17(17):1651–1667, nov 2007.

[158] Aswin N. Venkat, James B. Rawlings, and Stephen J. Wright. Stability and optimality of distributed model predictive control. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 6680–6685, Seville, 2005. IEEE.

[159] Aswin N. Venkat, James B. Rawlings, and Stephen J. Wright. Implementable distributed model predictive control with guaranteed performance properties. In *Proceedings of the 2006 American Control Conference*, pages 613–618, Minneapolis, MN, 2006. IEEE.

[160] Reinhard Neck and Engelbert Dockner. Conflict and cooperation in a model of stabilization policies. *Journal of Economic Dynamics and Control*, 11(2):153–158, 1987.

[161] Tamer Başar. Asynchronous algorithms in non-cooperative games. *Journal of Economic Dynamics and Control*, 12(1):167–172, 1988.

[162] Joel E. Cohen. Cooperation and self-interest: Pareto-inefficiency of Nash equilibria in finite random games. *Proceedings of the National Academy of Sciences of the United States of America*, 95(17):9724–9731, 1998.

[163] J. M. Maestre, D. Muñoz De La Peña, and E. F. Camacho. A Distributed MPC scheme with Low Communication Requirements. In *Proceedings of the 2009 American Control Conference*, pages 2797–2802, St. Louis, MO, 2009. IEEE.

[164] J. M. Maestre, David Muñoz de la Peña, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Control Applications and Methods*, 32(2):153–176, 2011.

[165] J. M. Maestre, D. Muñoz de la Peña, E. F. Camacho, and T. Alamo. Distributed model predictive control based on agent negotiation. *Journal of Process Control*, 21(5):685–697, 2011.

[166] Brett T. Stewart, Aswin N. Venkat, James B. Rawlings, Stephen J. Wright, and Gabriele Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460–469, aug 2010.

[167] Brett T. Stewart, Stephen J. Wright, and James B. Rawlings. Cooperative distributed model predictive control for nonlinear systems. *Journal of Process Control*, 21(5):698–704, jun 2011.

[168] Jinfeng Liu, Xianzhong Chen, David Muñoz de la Peña, and Panagiotis D. Christofides. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE Journal*, 56(8):2137–2149, 2010.

[169] Christian Conte, Colin N. Jones, Manfred Morari, and Melanie N. Zeilinger. Distributed synthesis and stability of cooperative distributed model predictive control for linear systems. *Automatica*, 69:117–125, 2016.

[170] William B. Dunbar. Distributed Receding Horizon Control of Dynamically Coupled Nonlinear Systems. *IEEE Transactions on Automatic Control*, 52(7):1249–1263, jul 2007.

[171] Paul Trodden and Arthur Richards. Distributed model predictive control of linear systems with persistent disturbances. *International Journal of Control*, 83(8):1653–1663, 2010.

[172] Paul Trodden and Arthur Richards. Cooperative distributed MPC of linear systems with coupled constraints. *Automatica*, 49(2):479–487, feb 2013.

[173] P. R. Baldivieso Monasterios and P. A. Trodden. Output feedback quasi-distributed MPC for linear systems coupled via dynamics and constraints. In *Proceedings of the 2016 UKACC International Conference on Control*, Belfast, 2016.

[174] Marcello Farina, Giulio Betti, Luca Giulioni, and Riccardo Scattolini. An Approach to Distributed Predictive Control for Tracking–Theory and Applications. *IEEE Transactions on Control Systems Technology*, 22(4):1558–1566, 2014.

[175] Marcello Farina, Giulio Betti, and Riccardo Scattolini. Distributed predictive control of continuous-time systems. *Systems & Control Letters*, 74:32–40, 2014.

[176] Sergio Lucia, Markus Kögel, and Rolf Findeisen. Contract-based Predictive Control of Distributed Systems with Plug and Play Capabilities. In *IFAC-PapersOnLine*, volume 48, pages 205–211. Elsevier B.V., 2015.

[177] Paul Trodden, P. R. Baldivieso Monasterios, and J. M. Maestre. Distributed MPC with Minimization of Mutual Disturbance Sets. In *Proceedings of the 2016 American Control Conference*, pages 5193–5198, Boston, MA, 2016. IEEE.

[178] Rolf Schneider. *Convex Bodies: The Brunn–Minkowski Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2 edition, 2013.

[179] Petter Nilsson and Necmiye Ozay. Synthesis of separable controlled invariant sets for modular local control design. In *Proceedings of the 2016 American Control Conference*, pages 5656–5663, Boston, MA, 2016.

[180] S. V. Raković and R. H. Gielen. Positively Invariant Families of Sets for Interconnected and Time-Delay Discrete-Time Systems. *SIAM Journal on Control and Optimization*, 52(4):2261–2283, 2014.

[181] Benjamin Kern and Rolf Findeisen. Analysis and Constrained Control of Nonlinear Interconnected Systems exploiting Positively Invariant Family of Sets. In *Proceedings of the 52nd IEEE Conference on Decision and Control*, pages 3806–3811, Florence, 2013. IEEE.