Anticipative model predictive control for linear parameter-varying systems

Hanema, J.

Anticipative model predictive control for linear parameter-varying systems

PROEFSCHRIFT

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de rector magnificus prof.dr.ir. F.P.T. Baaijens, voor een commissie aangewezen door het College voor Promoties, in het openbaar te verdedigen op

dinsdag 30 oktober 2018 om 11:00 uur

door

Jurre Hanema

geboren te Heerenveen

Dit proefschrift is goedgekeurd door de promotoren en de samenstelling van de promotiecommissie is als volgt:

voorzitter:	prof.dr.ing. A.J.M. Pemen
1e promotor:	prof.dr. S. Weiland
copromotoren:	dr.ir. R. Tóth
	dr. M. Lazar
leden:	dr.ir. M. Cannon (University of Oxford)
	prof.dr.ir. A. Bemporad (IMT School for Advanced Studies Lucca)
	prof.dr.ir. W.P.M.H. Heemels
adviseur:	dr.ir. M.M.J. van de Wal (ASML)

Het onderzoek dat in dit proefschrift wordt beschreven is uitgevoerd in overeenstemming met de TU/e Gedragscode Wetenschapsbeoefening.

Anticipative model predictive control for linear parameter-varying systems

Jurre Hanema

The research reported in this thesis is part of the research program of the Dutch Institute of Systems and Control (DISC). The author has successfully completed the educational program of the DISC Graduate School.

The research reported in this thesis was supported by the Impulse 1-program of Eindhoven University of Technology and ASML.

Anticipative model predictive control for linear parameter-varying systems by Jurre Hanema.
Eindhoven: Technische Universiteit Eindhoven, 2018. Proefschrift.

ISBN: 978-94-6375-140-7
NUR: 992

Printed by Ridderprint BV.

The scientific man does not aim at an immediate result. He does not expect that his advanced ideas will be readily taken up. His work is like that of the planter—for the future. His duty is to lay the foundation for those who are to come, and point the way. He lives and labors and hopes.

Nikola Tesla

Contents

List of abbreviations v

Summary vii

1 Introduction 1
- 1.1 Control, constraints, and models . . . 1
- 1.2 The linear parameter-varying paradigm . . . 8
  - 1.2.1 History and overview . . . 8
  - 1.2.2 Scheduling dynamics . . . 11
  - 1.2.3 Relation to other linear system classes . . . 13
- 1.3 Predictions and control under uncertainty . . . 15
  - 1.3.1 Concept of constrained LPV control . . . 15
  - 1.3.2 Desirable properties . . . 17
  - 1.3.3 Classification of strategies . . . 18
  - 1.3.4 Dynamic programming and min-max formulations . . . 19
  - 1.3.5 Tractable feedback strategies . . . 20
- 1.4 Research questions . . . 25
- 1.5 Outline and contributions of the thesis . . . 28

2 Preliminaries 31
- 2.1 Mathematical notation and definitions . . . 31
  - 2.1.1 General notation . . . 31
  - 2.1.2 Sets . . . 32
  - 2.1.3 Functions . . . 34
- 2.2 Linear parameter-varying systems . . . 36
  - 2.2.1 Formal definition . . . 36
  - 2.2.2 LPV state-space representations . . . 37
  - 2.2.3 Constrained LPV representations . . . 38
  - 2.2.4 Stability of (L)PV-SS representations . . . 39
  - 2.2.5 Reachable and contractive sets . . . 40
- 2.3 Embeddings of non-linear systems . . . 43
  - 2.3.1 The embedding principle . . . 44
  - 2.3.2 Construction of embeddings . . . 45
- 2.4 Model predictive control . . . 47

3 Anticipative tube-based MPC for LPV systems 51
3.1 Introduction . . . 51
3.2 Anticipative control . . . 52
3.3 Tubes: motivation and background . . . 55
3.3.1 Why tubes? . . . 55
3.3.2 A survey of tube-based MPC . . . 56
3.4 Foundations of LPV tube MPC algorithms . . . 59
3.4.1 Tube synthesis . . . 60
3.4.2 Receding-horizon tube MPC . . . 62
3.4.3 Desirable properties revisited . . . 63
3.4.4 Domain of attraction . . . 64
3.4.5 The class of cost functions . . . 65
3.5 Homothetic tubes in anticipative LPV MPC . . . 67
3.6 Concluding remarks . . . 68

4 Guaranteeing feasibility and stability 69
4.1 Introduction . . . 69
4.1.1 Problem setting . . . 71
4.2 Finite-step contraction . . . 71
4.3 The finite-step TMPC algorithm . . . 72
4.3.1 Periodically parameterized tubes . . . 72
4.3.2 Terminal cost construction . . . 74
4.3.3 Main result . . . 77
4.4 Implementation details . . . 79
4.4.1 Periodically homothetic parameterization . . . 79
4.4.2 Linear programming formulation . . . 80
4.4.3 Computation of finite-step contractive sets . . . 83
4.5 Numerical examples . . . 84
4.5.1 Second-order academic example . . . 84
4.5.2 Two-cart system . . . 85
4.6 Concluding remarks . . . 93
4.6.1 Parameter-varying input matrix . . . 93
4.6.2 Summary . . . 96

5 Heterogeneously parameterized tube MPC 97
5.1 Introduction . . . 97
5.1.1 Problem setting . . . 99
5.2 Heterogeneously parameterized tubes . . . 99
5.3 The heterogeneous TMPC algorithm . . . 100
5.3.1 Parameterization conditions . . . 100
5.3.2 Main result . . . 103
5.4 Parameterization construction . . . 105
5.4.1 The HpT cost function . . . 106
5.4.2 The HpT-SF parameterization . . . 107

5.5 Numerical examples . . . 110
5.5.1 Parameter-varying double integrator . . . 111
5.5.2 Parameter-varying third-order system . . . 113
5.6 Concluding remarks . . . 117

6 Reduced-complexity tube parameterizations 119
6.1 Introduction . . . 119
6.1.1 Problem setting . . . 121
6.2 Preliminaries and reweighted ℓ_1-optimization . . . 122
6.3 The reduced-complexity TMPC algorithm . . . 123
6.3.1 Sparsely parameterized tubes . . . 124
6.3.2 Dense parameterization and LP formulation . . . 125
6.3.3 Off-line parameterization design . . . 128
6.3.4 On-line parameterization construction . . . 130
6.3.5 Main result . . . 131
6.4 Numerical examples . . . 134
6.4.1 Numerical example 1: second-order system . . . 135
6.4.2 Numerical example 2: third-order system . . . 138
6.5 Concluding remarks . . . 141

7 Towards constrained LPV reference tracking 143
7.1 Introduction . . . 143
7.1.1 Notation . . . 144
7.1.2 Problem setting . . . 145
7.2 Bounded-error invariant set . . . 145
7.3 The tracking TMPC algorithm . . . 147
7.4 Numerical examples . . . 150
7.4.1 Minimal ϵ for fixed r . . . 150
7.4.2 All ϵ-achievable references . . . 151
7.5 Concluding remarks . . . 153

8 Non-linear MPC through LPV embeddings 155
8.1 Introduction . . . 155
8.1.1 Problem setting . . . 158
8.2 The non-linear TMPC algorithm . . . 159
8.2.1 Tubes for LPV embeddings . . . 159
8.2.2 Center-based cost function and terminal set . . . 161
8.2.3 Main result . . . 163
8.3 Initializing the state constraints . . . 165
8.3.1 Bounded rate of variation . . . 165
8.3.2 Initial feasible trajectory . . . 165
8.4 Bounding the scheduling tube . . . 166
8.4.1 Affine scheduling map . . . 166
8.4.2 General scheduling map . . . 167

8.5 Numerical examples . . . 170
8.5.1 Controlled Van der Pol oscillator . . . 170
8.5.2 Electrically driven inverted pendulum . . . 171
8.5.3 Two-tank system . . . 174
8.6 Concluding remarks . . . 178
8.6.1 Extension to input non-linearities . . . 178
8.6.2 Summary . . . 181

9 Anticipative control of a thermal system **183**
9.1 Introduction . . . 183
9.2 Thermal modeling . . . 185
9.3 Experiment description . . . 190
9.4 Simulation results . . . 194
9.4.1 Endpoint actuation case . . . 194
9.4.2 Full actuation case . . . 197
9.4.3 Computational load and LTV control . . . 197
9.4.4 Effect of the IH heat transfer coefficient . . . 204
9.5 Concluding remarks . . . 204

10 Conclusion **207**
10.1 Overview . . . 207
10.2 Main contributions and future work . . . 209

A Synthesizing finite-step contractive sets with PSO **213**
A.1 Problem setting . . . 213
A.2 The algorithm . . . 214
A.2.1 Set parameterization . . . 214
A.2.2 Verifying finite-step contraction . . . 215
A.2.3 Main result . . . 216
A.2.4 Note on volume computation . . . 217

B Estimating domains of attraction **219**
B.1 Problem setting . . . 219
B.2 Approach 1: gridding and testing . . . 220
B.3 Approach 2: scaling . . . 220

Bibliography **223**

List of publications **241**

Acknowledgments **243**

Curriculum vitae **245**

List of abbreviations

$\mathcal{CH}_1$	continous and positively homogeneous of degree one.
BES	bubble extraction system.
CSA	cross-sectional area.
DHC	decreasing-horizon control.
DOA	domain of attraction.
DOF	degree of freedom.
DP	dynamic programming.
HpT	heterogeneously parameterized tube.
HpTMPC	heterogeneously parameterized TMPC.
IC	integrated circuit.
IH	immersion hood.
LDI	linear difference inclusion.
LFR	linear-fractional representation.
LMI	linear matrix inequality.
LP	linear program.
LPU	linear parametrically uncertain.
LPV	linear parameter-varying.
LPV-LFR	linear parameter-varying linear-fractional representation.
LPV-PSS	polytopic LPV state-space.
LPV-SS	LPV state-space.
LTI	linear time-invariant.
LTI+	additively disturbed LTI.
LTV	linear time-varying.

MB	move blocking.
MPC	model predictive control.
MPLP	multi-parametric linear program.
mRPI	minimal robustly positively invariant.
NMPC	non-linear model predictive control.
PCA	principal component analysis.
PpT	periodically parameterized tube.
PSO	particle swarm optimization.
PV	parameter-varying.
PV-SS	PV state-space.
PWA	piecewise affine.
QP	quadratic program.
RHC	receding-horizon control.
ROV	rate-of-variation.
SDP	semi-definite program.
SpT	sparsily parameterized tube.
SS	state-space.
TMPC	tube model predictive control.
WC-DOA	worst-case domain of attraction.

Summary

Anticipative model predictive control for linear parameter-varying systems

This thesis presents several contributions to tube-based model predictive control (MPC) for linear parameter-varying (LPV) systems. A common feature of the presented approaches is their *anticipative* capability: that is, the capability of exploiting available information on the possible future trajectories of the scheduling variable to improve control performance.

The LPV system concept can be used to model physical systems whose dynamics depend on an external parameter or on the current operating point. In an LPV system, the dynamical mapping between the control input and the output is linear, but this mapping depends on a so-called scheduling variable. The dynamics of an aircraft, for instance, can vary with its airspeed and altitude: these two quantities can therefore considered to be scheduling variables. It is also possible to "embed" a non-linear system in an LPV representation. In that case, the scheduling variable is used to capture the non-linear behavior.

In engineering, systems are often subject to constraints. These constraints can include hard limitations on the control actuation inputs that can be applied, and limitations on the allowed resulting behavior of the system. MPC is a control design approach that can ensure that the constraints will not be violated. A central feature of MPC is its explicit use of a mathematical model to predict the future response of the system. At each sampling instant, this prediction model is used in an optimization procedure to decide which control input should be applied.

In the LPV setting that is adopted in this thesis, the current value of the scheduling variable can be measured, but its future evolution is considered uncertain. Therefore, at each sampling instant, an LPV MPC must compute a control action that guarantees closed-loop stability with respect to *all possible* future realizations of the scheduling variable. In some applications, however, future scheduling trajectories are not completely uncertain, but some knowledge about their possible future behavior is available. In this thesis, a thermal control problem from semiconductor lithography is considered in which the scheduling variable corresponds to an exposure trajectory that is approximately known in advance. Because the temperature which is to be controlled depends on the scheduling variable, to achieve the best possible performance, it is necessary to use this information in predicting the future response of the system and deciding which control input should be applied.

Unfortunately, it is observed that existing LPV MPC approaches are limited in their capabilities for exploiting available information on possible future scheduling trajectories. Therefore, to improve upon this situation, an "anticipative" control concept is proposed in this thesis. In anticipative control, the controller can use information about the possible future scheduling trajectories that might become available on-line while the system is operational. In this way, the future response of the system can be predicted with less uncertainty, and better control performance can be attained.

It is shown in this thesis that a tube-based approach can be used to construct anticipative LPV model predictive controllers. The practical implementation of these controllers requires the solution of a linear program (LP) with a number of variables and constraints that scales linearly in the prediction horizon. LPs can be solved efficiently, and therefore the controllers developed in this thesis can be considered to be computationally tractable. Within the proposed framework for anticipative control, several distinct contributions to the literature on tube-based MPC for LPV systems are subsequently made. These are summarized next.

To ensure recursive feasibility and closed-loop stability in MPC, typically an invariant terminal set is employed. The first main contribution of this thesis is to show that this condition can be relaxed into the requirement that the terminal set must be finite-step contractive instead. Finite-step contractive sets have been employed before in nominal MPC and in certain robust MPC approaches, but not yet in tube-based MPC. To achieve this, a new periodic tube parameterization is proposed which yields recursive feasibility. Closed-loop stability is ensured through the construction of an associated terminal cost based on a periodic Lyapunov-like function that is decreasing along set-valued closed-loop trajectories.

Next, a so-called heterogeneous tube parameterization structure is proposed. Instead of parameterizing all tube cross-sections in the same way as is usual in currently available approaches, this setup allows to use different parameterizations for each cross-section. Conditions under which this combination of parameterizations leads to a recursively feasible algorithm, are presented. The combination of different parameterizations into one tube has similarities to the well-known concept of “move blocking” in nominal MPC. A specific parameterization that satisfies the developed conditions and corresponds to a combination of scenario MPC and homothetic tube MPC, is also proposed. The heterogeneous tube has the potential to improve the trade-off between the complexity of the parameterization, and achievable control performance as measured by the size of the corresponding domain of attraction.

Subsequently, an alternative method for optimizing the aforementioned complexity/performance trade-off is proposed. Given a pre-specified domain of attraction, an off-line optimization procedure determines which degrees of freedom can be eliminated from the tube parameterization without affecting feasibility. The procedure relies on reweighted one-norm optimization to construct a parameterization that is as “sparse” as possible. The degrees of freedom that correspond to zeros in this sparse parameterization can be removed as decision variables from the on-line tube synthesis problem. In this way, the number of decision variables can be reduced. A significant reduction is obtained especially when the sets specifying the shape of the tube cross-sections are described in a vertex representation.

The next topic addressed is the tracking of constant, but non-zero, references. In an LPV system, variations in the scheduling signal can lead to unavoidable tracking errors. Existing tracking algorithms ensure offset-free tracking when the scheduling signal converges to a constant value, but do not give guarantees on the error when the scheduling signal keeps varying. In this thesis, an alternative is proposed through the construction of so-called bounded-error invariant sets. By using these sets in the tube-based framework, an a-priori bound on the

tracking error induced by the scheduling variations can be guaranteed. This development can be seen as a first step towards an approach for the bounded-error tracking of a larger class of reference signals in constrained LPV systems.

As the last main contribution of the thesis, it is shown how the previously developed tube MPC solutions can be applied to non-linear systems described by an LPV embedding. As part of the initialization of the controller, sequences of state constraints are constructed to bound the predicted state trajectories. Corresponding scheduling sets are computed based on the relation between the state and scheduling variables that exists in an LPV embedding. This can be seen as a particular way of obtaining a description of possible future scheduling realizations, that are subsequently used in the anticipative control framework. The construction guarantees that the assumed possible evolutions of the scheduling variable are consistent with the realized closed-loop state trajectories. Therefore, the controller is recursively feasible and stabilizes the original non-linear system.

Finally, the developed methods are applied to the control of a simple one-dimensional thermal system. The considered example represents a simplified version of a thermal control problem from semiconductor lithography. In this system, the scheduling signal corresponds to an exposure trajectory that is approximately known in advance and this future information can be included in the framework of anticipative control. The effect of including the available knowledge on approximate future trajectories is investigated for different system configurations. For the considered system, it is found that there are situations in which anticipative control can give a performance improvement as measured in terms of the simulated closed-loop cost.

Chapter 1

Introduction

1.1 Control, constraints, and models

In the theory of systems and control, a *system* is often thought of as a “box” which takes a control input and produces an output in response (Figure 1.1). This mapping between inputs and outputs can be *dynamical*, which means that the output not only depends on the current value of the input, but also on its past trajectory. The car depicted in Figure 1.2 can be considered as a simple illustrative example of a system. The depression of the accelerator pedal is an input, and the resulting forward speed of the vehicle is an output. Clearly, the speed depends not only on the instantaneous position of the pedal, but also on the length of time for which it was depressed. Thus, the car is a dynamical system.

Control can be defined as “the adjustment of the available degrees of freedom (manipulated variables) to assist in achieving acceptable operation of a system (process, plant).” (Skogestad and Postlethwaite 2005). This adjustment can be performed by a *controller*, which is a device that is connected to the input of the system whose behavior shall be influenced. The control design problem is to determine, for a given system, a suitable controller such that “acceptable” operation results.

The meaning of “acceptable” can be defined by specifying a so-called *reference*. The reference corresponds to a desired value or trajectory of the output of the system. Typically one of two objectives is of interest: either *regulation* or *tracking*. In a regulation problem, the reference is a constant signal, and the purpose of the controller is to attenuate the effect of disturbances so as to keep the output of the system near this constant value. In a tracking problem, the reference is a time-varying signal, and the controller must make the output follow (“track”) this reference as closely as possible.

A controller can be connected to the system in the *feedforward* or *open-loop* manner of Figure 1.3. For a given reference input, the controller uses a mathematical model of the system

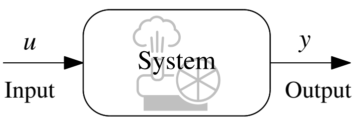


Figure 1.1: A system.

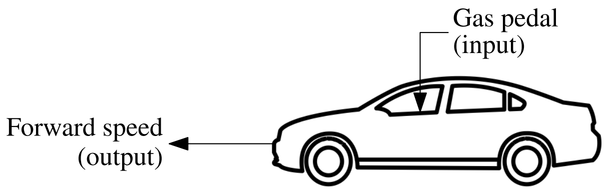


Figure 1.2: A motorized vehicle is a simple example of a system.

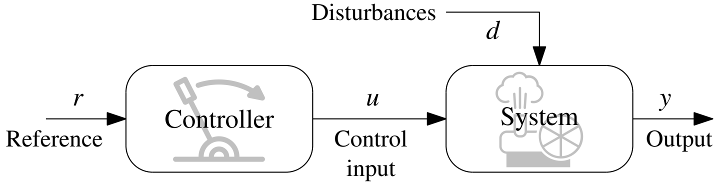


Figure 1.3: The principle of feedforward control. The controller does not make use of the output measurement.

to select an appropriate control input that should lead to acceptable behavior. In this setting, the controller is not aware of the true value of the output of the system. Therefore, if significant disturbances are acting on the system or if the used mathematical model is not accurate, the controller will not be able to achieve its tracking- or regulation-objective.

To overcome this problem, the inclusion of *feedback* is necessary. In feedback control, a measurement of the output is fed back into the controller, so that this measurement can be used in determining the control input. The feedback control concept is illustrated in Figure 1.4. Because a loop is "closed" between the output of the system and the controller, the overall interconnection is called a *closed-loop system*. Due to the inclusion of feedback, the controller can attenuate the effect of disturbances acting on the system, and can compensate to a certain degree for modeling uncertainties.

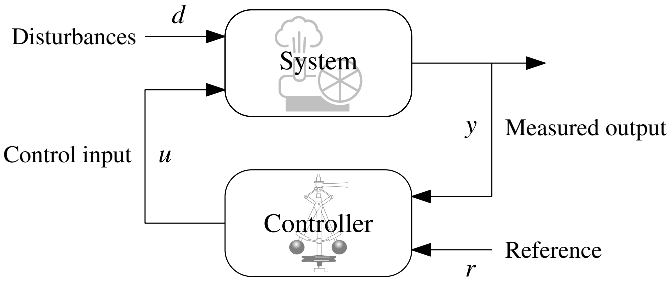


Figure 1.4: The principle of feedback control. The usage of the output measurement allows the controller to compensate the effects of unknown exogenous disturbances.

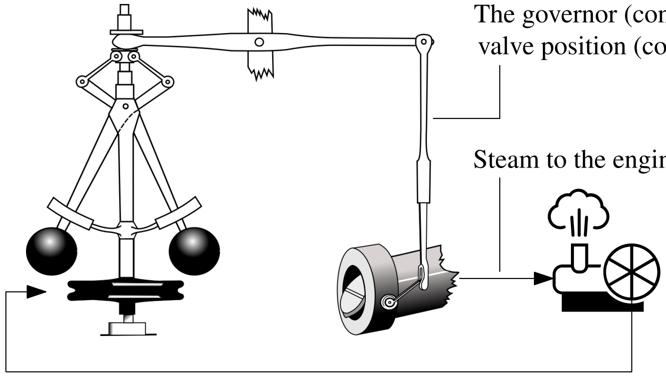


Figure 1.5: Watt's centrifugal governor, patented in 1788. An increase in rotational speed (the measured output) pushes the steel balls outwards. This motion acts on a lever which reduces the opening of the steam valve (the control input), decreasing the flow of steam into the engine and lowering its speed. If the governor is well-designed, an equilibrium will eventually be reached and the engine will run at a constant (reference) pace.

Besides this obvious advantage, a difficulty in feedback design is that the controller can also have a destabilizing effect if it is not designed properly. Roughly speaking, the closed-loop is called *stable* if, in the feedback interconnection of Figure 1.4, the output is guaranteed to converge to—or remain in a small neighborhood of—the reference value. If the closed-loop is unstable, the output continually diverges and reaches an arbitrarily large amplitude, which is clearly undesirable and can have disastrous effects.

An important early industrial feedback control system was the so-called *governor* (Kang 2016). Governors were mechanical devices used to regulate the running speed of steam engines (Figure 1.5). Before the invention of the governor, an engine would slow down or speed up whenever its attached mechanical load was changed. By removing this unwanted effect, the governor greatly improved the steam engine's usefulness for industrial applications. However, sometimes governors caused the speed to oscillate or to continually increase. In search of a solution to this problem, the workings of the governor were analyzed mathematically by J.C. Maxwell in the 19th century (Maxwell 1868). His work provided a theoretical answer to how a governor should be designed such that the speed of the engine is guaranteed to reach the desired constant value (i.e., such that the closed-loop system displays a stable behavior). It can be seen as one of the early foundations of the modern field of systems and control.

The analysis of Maxwell showed that when designing a controller, it is essential to be aware of the relevant characteristics of the system that is to be controlled. This concerned knowledge of the dynamic response—i.e., the dynamical relationship between input- and output variables—of the governor and of the machine that it was attached to, which can be described

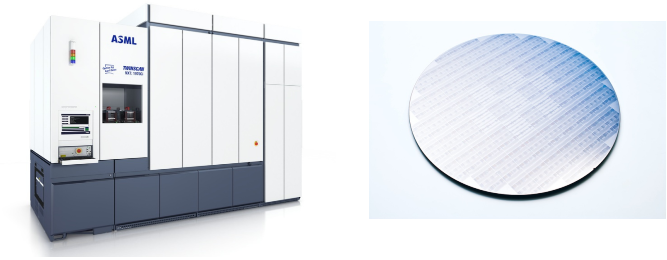


Figure 1.6: Semiconductor lithography machine (left) and silicon wafer (right). The wafer has a diameter of 30 cm.

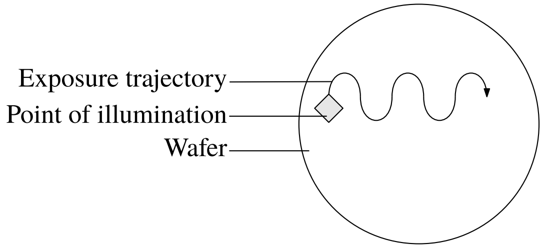


Figure 1.7: Basic concept of semiconductor lithography.

mathematically in terms of differential equations.

For advanced systems that are encountered in engineering today, being able to describe their dynamics mathematically in terms of a model is still equally important to achieve successful control design. Another feature that frequently occurs and must be accounted for, are constraints. Constraints represent hard limitations on the possible or allowed behavior of a system that may not be violated at any cost. Clearly, a controller must be designed with such limitations in mind.

One type of constraints are the so-called *input constraints*. These result from physical actuator limitations, causing restrictions on the control inputs that can be applied to the plant. For instance, a steam valve can not be used to increase the flow rate if it is already fully opened, and an electric motor used to drive a car can not provide more than its rated torque.

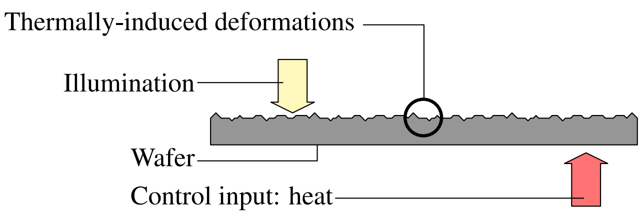


Figure 1.8: Example of a constrained control problem: wafer temperature control.

A second type of constraints corresponds to limits on the behavior of the system that results *after* applying the controls. This type of constraints are known as *output constraints*. Examples of this type are the maximal allowed acceleration of a vehicle, the maximum temperature in a chemical reactor, or obstacles restraining the range of motion of a mechanical system.

A relevant example of a constrained system can be found in semiconductor immersion lithography, which is a central step in the production of integrated circuits (ICs) (Lawson and Robinson 2016; Moreau 1988; Singh et al. 2013) (Figure 1.6). In the lithography step, a pattern specific to the IC that is being produced is projected onto a silicon wafer that is coated by a photosensitive material. In this process, the exposure location is moving with respect to the wafer according to a prescribed trajectory (Figure 1.7). It is required that this projection is extremely accurate, with a margin of error within the sub-nanometer range. To achieve this, it is necessary to eliminate all possible factors that can contribute to imaging errors. One such factor are deformations at the wafer surface that are caused by a non-uniform temperature distribution in the wafer (Figure 1.8). This can be avoided by regulating the temperature of the wafer, and for this purpose the system is equipped with heaters.

The heaters in this system can supply positive heat to the wafer, but can not actively cool it down. Furthermore they can supply a certain maximum amount of heating power. Thus, the input constraint in this system is that only non-negative control inputs can be physically applied. In addition, the magnitude of these inputs must be limited to a given maximum value. A controller must be aware of these constraints: otherwise it might, for instance, attempt to meet its performance objective by requesting a negative heat input— causing poor performance and potentially even destabilizing the system.

Another constraint comes from the fact that the controller is not allowed to heat the wafer beyond a given maximum temperature, in order to prevent excessive deformation of the material. This corresponds to an output constraint.

One way of designing a controller that avoids constraint violations such as requesting negative heat, is by making it "cautious", i.e., to have it operating sufficiently far away from the constraints under all expected operating conditions. However, this limits the performance that can be achieved.

MPC is an advanced control methodology which can explicitly take into account the presence of constraints (Maciejowski 2002; Rawlings and Mayne 2009). A model predictive controller can therefore operate a constrained system safely and within its physical limitations, without being overly cautious. A central feature of MPC is that it relies on a mathematical model of the system to be controlled. This prediction model is used to explicitly predict the response of the system to changes to its manipulated variables, and these predictions are used as part of a decision-making process which decides what is the "best" control action that should be applied to the plant. This process often corresponds to a mathematical optimization problem whose numerical solution must be computed many times per second while the system is in operation ("on-line"). Thus, increased computational resources are required for the implementation of MPC in comparison to many other control approaches. This is a price that has to be paid to benefit from its constraint handling capabilities.

Furthermore, because MPC relies on a mathematical model to predict how the system will respond to changes in its control inputs, it is important that this model is an accurate representation of reality. If there is a large discrepancy between the predicted and realized responses, the closed loop may behave poorly, and constraints may be violated. Arguably the most popular approach in control theory is to describe the system using a linear time-invariant (LTI) model (Dorf and Bishop 2011; Franklin et al. 2014; Schetzen 2002). In such a model, the dynamical relationship between input and output variables is linear and this relationship does not change with time or with any other parameter. Due to its relative simplicity, it is possible to formulate efficient control design approaches working on this class of models. This includes MPC, as the optimization problem that must be solved as part of the MPC algorithm has a relatively simple structure in the linear case (Rawlings and Mayne 2009). The assumption that a system can be well-described by an LTI model is often reasonable in practice, because even a strongly non-linear system behaves approximately linear in a sufficiently small region around a fixed operating point (Dorf and Bishop 2011, Chapter 2).

Problems arise when a non-linear system is to be controlled over a larger operating regime, or when the dynamical response of the system depends on an external parameter that can not be influenced by the controller. These two situations are illustrated in Figure 1.9.

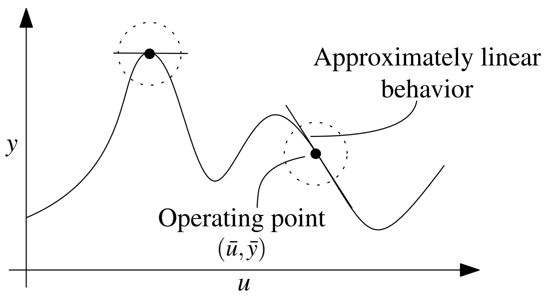
The steady-state output y of some non-linear system in response to an applied input u is depicted in Figure 1.9.(a). In a small neighborhood of a single operating point (corresponding to a pair of values $(\bar{u}, \bar{y})$), the non-linear response can be well approximated by a single linear function. However, if the non-linear behavior is to be represented accurately over a larger set of operating conditions, this is no longer sufficient. Indeed, for the non-linear function from the figure, the linear approximation changes significantly depending on the operating point.

The thermal control problem introduced previously in Figure 1.8 represents a system whose dynamical behavior is dependent on a set of external parameters that can not be influenced by the controller. In immersion lithography, before hitting the wafer, the light first passes a layer of water that is held in between the wafer and the so-called immersion hood (IH). This water acts as a heat sink, and thereby influences the temperature distribution in the wafer and the resulting deformations at the surface. Because the IH is moving with respect to the wafer according to the exposure trajectory, this effect is position-dependent.

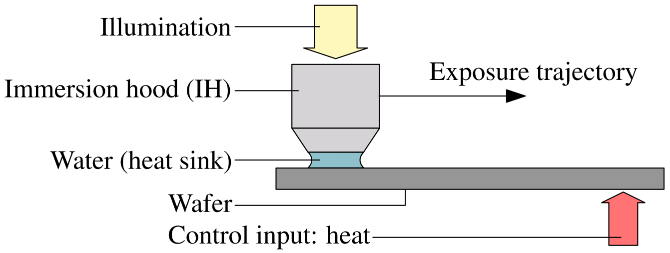
In the two cases illustrated above, the LTI assumption can not offer an accurate description of reality, and if high control performance and guaranteed constraint satisfaction is to be achieved, a more elaborate mathematical model structure is needed. The paradigm of LPV systems offers such a structure (Shamma 2012). In an LPV system, the relationship between control inputs and outputs is still linear, but this relationship is now allowed to depend on an external *scheduling variable*. This scheduling variable can, for instance, represent the operating point of a non-linear process, or any other parameter that influences the dynamical behavior of the system (Rugh and Shamma 2000; Shamma 2012). The class of systems that can be accurately described by LPV models is thus significantly larger than the class of systems that can be accurately described by LTI models.

Because the LPV system class retains the property of linearity between inputs and outputs, it is still possible to formulate computationally efficient design algorithms for LPV systems. Also, many results on LPV systems can be viewed as generalizations of results from the extensively studied LTI case (Tóth 2010).

The remainder of this introductory chapter is structured as follows. First, in Section 1.2, an overview of the LPV system concept is provided. The principles of MPC for LPV systems are discussed in Section 1.3 and a survey of the relevant literature is given. In Section 1.4, it



(a)



(b)

Figure 1.9: (a) The response of a non-linear system can be considered approximately linear in a small region around an operating point. If the system is operated around multiple operating points, a single linear model can no longer describe its behavior accurately. (b) In immersion lithography, the thermal distribution in the wafer is affected by the exposure location, because the water that is in contact with the wafer surface acts as a heat sink.

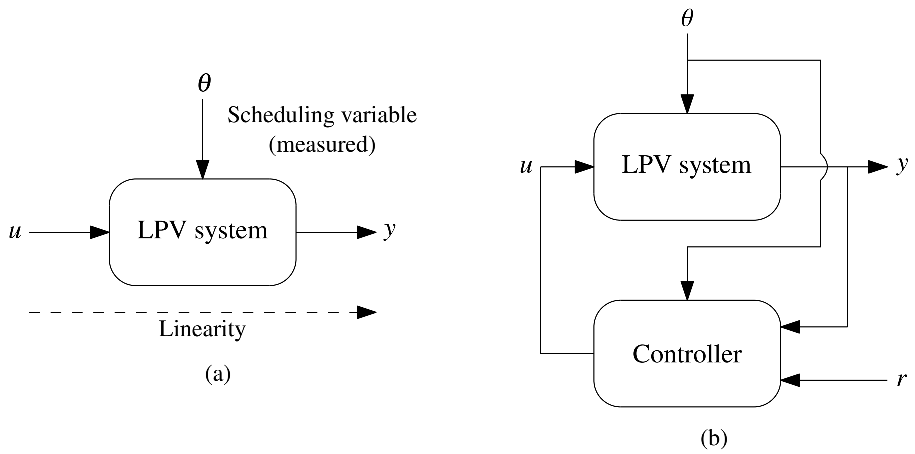


Figure 1.10: LPV system concept (a) and LPV feedback control (b).

is pointed out that there exist certain systems for which the existing LPV MPC approaches are not fully adequate. Based on this observation, a number of research questions that will be addressed in this thesis are stated. Finally, in Section 1.5, the chapter concludes with an outline of the rest of the thesis.

1.2 The linear parameter-varying paradigm

This section gives a more detailed overview of the LPV system concept, which was briefly introduced in the previous section. An overview, including some history, is provided in Section 1.2.1. The issue of uncertainty in future scheduling trajectories is discussed in Section 1.2.2. To conclude, Section 1.2.3 gives an overview of the principal differences between LPV systems and other types of linear systems.

1.2.1 History and overview

Linear parameter-varying systems are a special class of dynamical systems. In an LPV system the dynamical mapping between the control input u and the output y is linear. However, this linear relationship depends on an external and possibly time-varying scheduling variable θ. In the LPV setting that is considered in this thesis, it is assumed that the current value of the scheduling variable can be measured. Conceptually, it is also assumed that θ is an exogenous variable that is independent of u and y. Because a measurement of θ is available, a controller can—and preferably should—make use of this information. These concepts are illustrated in Figure 1.10.

The LPV system concept was first introduced to enable formal analysis of the industrial practice of *gain scheduling* (Shamma 2012). Gain scheduling can be viewed as a “divide

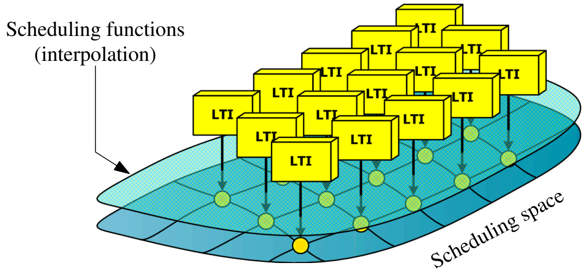


Figure 1.11: The principle of gain scheduling. A set of local models is obtained by linearizing a non-linear system around a number of operating points in the *scheduling space*, and an LTI controller is designed for each of these points. A global controller is then constructed by interpolating between these local controllers as a function of the measured operating point (i.e., the scheduling variable).

and conquer" design approach, in which a complex non-linear control problem is divided into a number of linear subproblems (Leith and Leithead 2000b). Specifically, a collection of operating points of the non-linear system is selected, and for each operating point a single LTI controller is designed which works well in a neighborhood around its corresponding point. While the system is in operation, an interpolation between these individual controllers takes place based on the current measured operating point (Figure 1.11). This can be done according to different possible interpolation schemes, e.g., input-, output-, and coefficient interpolation (Bachnas et al. 2014).

Perhaps the earliest successful applications of this concept can be found in military flight control systems that were being developed after World War II (Rugh and Shamma 2000). The dynamical behavior of an aircraft changes significantly over its flight envelope, which is characterized by parameters such as the airspeed, angle of attack, and altitude. Therefore, a single time-invariant controller could not provide satisfactory performance over the full range of flight conditions, and it became necessary to vary (i.e., "schedule") the controller based on the current flight conditions.

This type of "classical" gain scheduling enables the use of relatively simple linear design tools to address complex non-linear problems. Its primary disadvantage is that it is difficult to obtain theoretical guarantees on the resulting behavior (e.g., stability) of the closed-loop system (Leith and Leithead 2000a; Shamma and Athans 1990, 1991, 1992). There are difficulties associated with the selection of the operating points, and guaranteeing stability is typically only possible for "sufficiently slow" variations in the scheduling signal—where the meaning of "sufficiently" is not straightforward to quantify.

Instead of gain scheduling, it is also possible to apply a global modeling approach called *embedding* (Abbas et al. 2014; Kwiatkowski et al. 2006; Scherer et al. 1997; Tóth et al. 2011). In gain scheduling, a global model or controller is constructed by interpolating between several distinct local models. In contrast, in the embedding case, a *single* LPV model is constructed in such a way that variations in the scheduling variable can represent the non-linearities in the

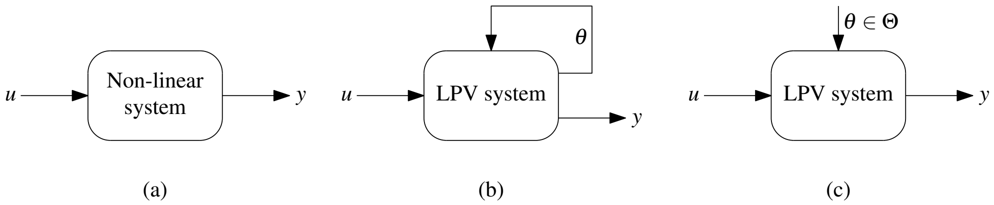


Figure 1.12: The principle of constructing an LPV embedding.

original model.

A non-linear system, as depicted in Figure 1.12.(a), defines a non-linear dynamical relationship between its input and output. The process of constructing an LPV embedding proceeds roughly as follows. A scheduling variable θ corresponding to some internal signals of the system is introduced that allows to equivalently represent this relation in terms of a linear dynamical map $u \mapsto y$ that depends on θ (Figure 1.12.(b)). The signal θ is subsequently converted into an exogenous signal by "cutting the loop" and by confining the values of θ to some suitable set Θ. This gives an LPV embedding of the non-linear system (Figure 1.12.(c)). Later, in Section 2.3, a more detailed explanation of embeddings and their construction is provided.

A fundamental property of an LPV embedding is that its associated set of admissible trajectories (i.e., the set of input and output signals that are compatible with the dynamics of the system) is a superset of the set of trajectories of the original non-linear system. This property ensures that a controller which stabilizes the embedding is guaranteed to stabilize the original non-linear system as well, something that can not be guaranteed when the local gain-scheduling approach is used.

As sketched above, the original development of the LPV paradigm was motivated to a large extent by the desire of being able to apply linear design methods to difficult non-linear problems. Besides capturing the effect of non-linearities, the scheduling variable can also be employed to model the effect of varying exogenous parameters of the system. This was the case in the lithography example introduced previously in Section 1.1, where the scheduling variable corresponds to the position of the IH. In summary, it can therefore be said that the scheduling variable can fulfill one or both of the following roles:

Modeling non-linearities The system has non-linear dynamics, and a scheduling variable is being introduced as an abstraction so that the non-linear system can be (over-)approximated by an LPV model. This is done in the gain scheduling- and embedding approaches described above.

Modeling the effect of external parameters The system has linear dynamics, but these dynamics depend on one or more external, measurable, parameters. For instance, this is the case with the wafer of Figure 1.9.(b) as its dynamical behavior is dependent on the exposure location. This position can therefore be regarded as a scheduling variable.

Nowadays, LPV modeling and control has expanded into many different application domains. A few examples of applications where LPV techniques have been considered are listed below:

- Aerospace applications, such as flight control (Hjartarson et al. 2014; Lhachemi et al. 2015; Papageorgiou et al. 2000; Scherer et al. 1997), jet engines (Falugi et al. 2001; Richter 2012), and missiles (Gahinet et al. 1995);
- Automotive applications, e.g., engine control (Castillo et al. 2015; Majecki et al. 2015; Wei et al. 2008) and active suspension systems (Do et al. 2013; Tudón-Martínez et al. 2015);
- Applications in power systems, such as the control of wind turbines (Bianchi et al. 2007; Shirazi et al. 2012);
- Electromechanical applications, e.g., a high-precision positioning stage (Groot Wassink et al. 2005), a control-moment gyroscope (Abbas et al. 2013; Theis et al. 2014), and a CD player (Dettori and Scherer 2001);
- Chemical process control, such as the modeling of a distillation column (Bachnas et al. 2014) and a polymerization reactor (Rahme et al. 2016);
- Biomedical applications in, e.g., artificial pancreases (Colmegna et al. 2018; Kovács 2017) and blood pressure regulation (Luspay and Grigoriadis 2015).

More applications of LPV modeling and control can be found, e.g., in the book (Mohammadpour and Scherer 2012) and in the survey paper (Hoffmann et al. 2014). It is also worth mentioning that LPV models of physical systems can be obtained through the use of data-driven identification approaches (Cox and Tóth 2016; Gunes et al. 2017; Rizvi et al. 2018; Tóth et al. 2012; Wingerden et al. 2009), or through conversion of first-principle non-linear models (Donida et al. 2009; Kwiatkowski et al. 2006; Rugh and Shamma 2000).

1.2.2 Scheduling dynamics

As stated before, in the considered LPV framework, it is assumed that at every time instant the scheduling value $\theta(k)$ is known. In MPC, it is necessary to be able to predict the response of the LPV system for some time steps ahead into the future. Although the current scheduling value is known, the future scheduling values are usually not known exactly. Therefore, some assumption on the future scheduling trajectories is necessary. Although the scheduling signal is assumed to be exogenous, some assumption on its possible behavior can typically be made, e.g., based on physical insight. This section discusses some types of knowledge on the possible future scheduling trajectories that might be available.

The first and obvious type of information consists of bounds on the possible scheduling values. The position of the IH in Figure 1.9.(b) for example must be somewhere on the wafer. Also, in an LPV embedding, the scheduling variable that is introduced to represent the non-linearities is assumed to belong to a bounded set.

A common type of further knowledge concerns bounds on the rate-of-variation (ROV) of the scheduling. This means that the scheduling variable can only change at a limited bounded rate, which follows from the physical properties of the process that is being modeled. The IH in the

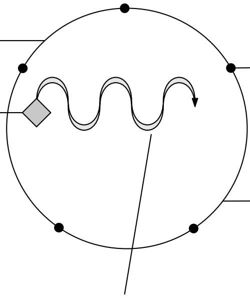


Figure 1.13: In the wafer temperature control problem, the future evolution of the scheduling variable can be predicted with high confidence.

lithography machine for instance can not jump instantaneously from one end of the wafer to the other, but instead its movement is bounded by a certain maximum velocity.

In the lithography application, as explained before using Figure 1.9.(b), the response of the wafer temperature to an input from one of the heaters depends on the current point of illumination. The position of the IH can therefore be considered as a scheduling variable. When manufacturing an IC, the point of illumination will follow approximately a pre-defined reference trajectory. Based on this nominal path and starting from the current IH position (the scheduling variable), all possible future scheduling trajectories are known to belong to a set around a nominal path. This situation is depicted in Figure 1.13. Hence, in this application, much more detailed information on possible future scheduling trajectories is available that can not be described by assuming just a bounded ROV.

A comparable situation occurs when controlling a non-linear system that is described in terms of an LPV embedding. For the purpose of control design, in an embedding, the scheduling variable is considered independent from other variables in the system. This is a necessary assumption to ensure linearity of the system representation and to enable the use of linear control design approaches. However, as can be seen from the sketch of Figure 1.12, the scheduling variable is actually constructed based on signals from the original non-linear system. Therefore, when the system is operating in closed loop with the designed controller, the evolution of the scheduling variable is not free: instead, it is inextricably linked to the input-, output-, and state trajectories of the real non-linear system.

This implies that also in the embedding case, more detailed information about possible future scheduling trajectories is available than just global value bounds or bounds on the ROV. Suppose, for instance, that the scheduling variable in the embedding depends on the system output and that the output has to satisfy a constraint such as being close to a reference signal. Then, this information can be used directly to construct more refined sets describing all possible future trajectories of the scheduling variable[1].

[1] Further details on this approach are provided in this thesis in Chapter 8.

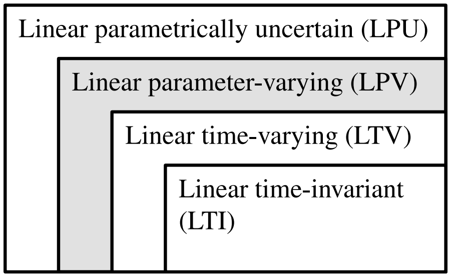


Figure 1.14: Different classes of linear systems.

A key distinction can be made between the types of knowledge on the scheduling behavior that were introduced above. The global bound on the allowable scheduling values, for instance, is a property of the system that does not change after it has been designed. In the lithography machine, the minimum- and maximum values of the scheduling variable are fixed by the physical dimensions of the wafer. On the other hand, the approximate future exposure trajectories are not fixed a-priori, but are dependent on the specific microchip design that is being manufactured. Likewise, in an embedding, the future evolution of the scheduling variable is connected to the future input and output trajectories that are being predicted as part of the MPC optimization process. Preferably, a controller should be flexible enough to be able to exploit this type of future information that becomes available while the system is in operation.

1.2.3 Relation to other linear system classes

Besides LPV systems, there exist several other classes of linear systems (Figure 1.14). All systems in this class share the property of linearity between inputs and outputs, which in turn leads to similarities between the control design approaches that can be applied. Nonetheless, there are also several differences that are relevant for the design of predictive controllers. Before proceeding to the literature overview on LPV MPC approaches, it is important to be aware of some of these differences.

To highlight the differences between the different system classes, it is convenient to represent an LPV system mathematically by a state-space (SS) representation of the form

$$\begin{aligned} x(k+1) &= A(\theta(k))x(k) + B(\theta(k))u(k), \\ y(k) &= C(\theta(k))x(k) + D(\theta(k))u(k), \end{aligned} \tag{1.1}$$

where $k \in \mathbb{N}$ denotes the discrete time instant, $u(\cdot)$ is the control input, $y(\cdot)$ is the system output, $x(\cdot)$ is an internal state variable, and $\theta(\cdot)$ is the scheduling variable. The matrices $\{A(\cdot), B(\cdot), C(\cdot), D(\cdot)\}$ are functions of the scheduling variable, and are of conformable dimensions. A more rigorous exposition of LPV state-space (LPV-SS) representations will be provided later in Section 2.2.

Relation to linear time-invariant (LTI) systems

An LTI system admits a state-space realization of the form

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k), \\ y(k) &= Cx(k) + Du(k). \end{aligned}$$

It is easy to see that this is a special case of (1.1) where the matrices $\{A, B, C, D\}$ are constant instead of functions of a scheduling variable. Equivalently, an LPV system in which the scheduling variable takes a constant value $\theta(k) = \bar{\theta}$ for all time k corresponds to an LTI system.

Relation to linear time-varying (LTV) systems

A linear time-varying (LTV) system is a system that has an SS representation of the form

$$\begin{aligned} x(k+1) &= A(k)x(k) + B(k)u(k), \\ y(k) &= C(k)x(k) + D(k)u(k), \end{aligned}$$

and can be interpreted as a special case of an LPV system where the parameter θ is equal to time, i.e., where $\theta(k) = k$ (Shamma 2012, Section 1.1.2). A major implication of this difference is that in the LTV case, there is no uncertainty in the future "scheduling" trajectories as there was in Section 1.2.2. As will be discussed in Section 1.3.1, this makes the MPC design problem for LPV systems fundamentally more difficult than that for LTV systems.

Relation to linear parametrically uncertain (LPU) systems

In the LPV framework as introduced previously, it is a fundamental assumption that the current value $\theta(k)$ of the scheduling variable is known or can be measured. If this is not the case, then the system is not called an LPV system but a linear parametrically uncertain (LPU) system. Such a system can be represented by an SS representation of the form (1.1); the only difference with the LPV case being that the value of $\theta(k)$ can not be measured.

These uncertain linear systems are also said to be subject to "parametric" or "multiplicative" uncertainty[2]. Because the assumption that $\theta(k)$ is measurable can be seen as an additional restriction on the systems that can be described in the LPV framework, LPV systems are considered as a subset of LPU systems, as depicted in Figure 1.14.

Relation to additively disturbed LTI (LTI+) systems

In the MPC literature, another case that is frequently encountered besides LPV systems, is that of LTI systems subject to so-called additive uncertainties. These can be described by an SS representation of the form

$$\begin{aligned} x(k+1) &= Ax(k) + B_1u(k) + B_2w(k), \\ y(k) &= Cx(k) + D_1u(k) + D_2w(k), \end{aligned}$$

[2]In the convention adopted in this thesis, "parametric" and "multiplicative" uncertainty are used synonymously. A parametric uncertainty can, therefore, also be time-varying according to a function $\theta(k)$.

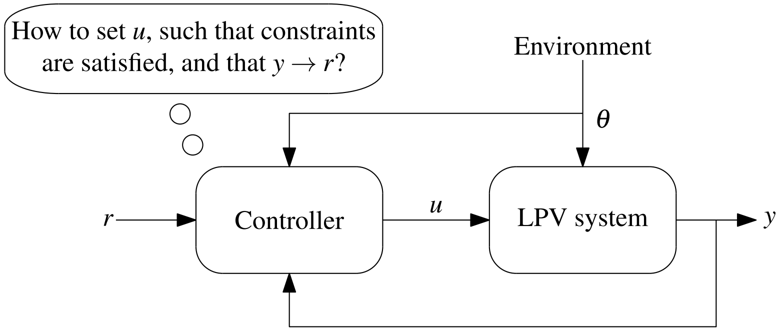


Figure 1.15: The interaction between the controller, the LPV system, and its environment.

where $w(k)$ is a disturbance term, also referred to as an "additive uncertainty". Strictly speaking, these systems are not different from "normal" LTI systems, because the disturbance is just an additional input. Nonetheless, for purposes of MPC, there is a major difference because the presence of the disturbances leads to uncertainty in the predicted response of the system to an applied control input. In other words, uncertainty now does not arise due to the unknown future evolution of a scheduling variable θ, but instead due to the unknown current- and future values of a disturbance term w. Because the two classes share a similar linear structure, there exist close relationships between MPC design approaches for these system types, as will be seen later in the overview of Section 1.3.5. An additive disturbance is usually not considered to be measurable.

1.3 Predictions and control under uncertainty

This section reviews the constrained control of LPV systems. In Section 1.3.1 the fundamental concepts are illustrated. Next, Section 1.3.2 summarizes the desirable properties of constrained LPV control solutions, namely "stability" and "tractability". The strategies used to obtain constrained control solutions can be classified in open/closed-loop and robust/scheduling strategies. These important distinctions are introduced in Section 1.3.3. An overview of relevant literature on the topic is subsequently given in Sections 1.3.4–1.3.5.

1.3.1 Concept of constrained LPV control

The output trajectory of an LPV system is not only dependent on the applied control inputs, but also on the evolution of the scheduling variable. As discussed in the preceding sections, the current value of the scheduling variable can be measured exactly, but its future evolution is usually uncertain. Thus, the response of the system to a given control input can only be predicted up to a certain degree of uncertainty. A controller then has to decide which control input to apply now—at the current time instant—based on these uncertain predictions (Rawlings and Mayne 2009).

An understanding of how this process works can be obtained by studying the interaction between the controller, the system, and its environment, as depicted in Figure 1.15. In this setting, which in some sense resembles a strategy game, the objective of the controller is to make the system behave in a certain desired way. First and foremost this includes guaranteeing the satisfaction of constraints on the input and output variables. Second, this includes stability and the optimization of performance, e.g., to drive the output of the system to a desired constant value and ensuring that it stays there (regulation), or to make the output follow closely a time-varying reference trajectory (tracking). To achieve these goals, the controller can manipulate the control inputs of the system.

The value of the scheduling variable can not be influenced by the controller, but it is an exogenous signal that is determined by an external process. This process can be considered to be part of the "environment" of the system. As discussed in Section 1.2.2, some knowledge that describes the possible future behavior of the scheduling variable can be available: in the lithography system, for instance, this is a set of possible future IH trajectories that are centered around a "nominal" reference trajectory.

Besides the scheduling variable, the system can also be affected by other (unmeasurable) disturbances. Such additional disturbances are omitted from this sketch for simplicity.

The controller can observe the current output y of the system, as well as the current value θ. Based on these observations the controller has to decide which "move" to make, i.e., to decide how to set u. To make this decision, it is necessary to plan ahead and to consider the effect of the present move, as well as that of possible future moves, on the behavior of y. However, the controller does not yet know precisely how the scheduling variable is going to evolve in the future. Therefore, it is necessary to come up with a plan that contains a strategy of how to respond to all possible future scheduling trajectories. Suppose that at each sampling time instant, the controller is able to plan N moves ahead. Then, this plan consists of a sequence

$$\text{Plan} = \{u_0, K_1, K_2, \ldots, K_{N-1}\} \tag{1.2}$$

where u_0 is the move that the controller will execute at the current time instant, and where $K_1, \ldots, K_{N-1}$ are functions mapping $(y, \theta) \mapsto u$. In other words, these functions contain a strategy ("policy") that describes how the controller plans to respond to future scheduling trajectories and the resulting behavior of y. In (1.2), the natural number N is called the *prediction horizon* of the controller.

After implementing the move u_0, a new measurement of y and θ becomes available at the next sampling time instant. Based on the new observations, the controller formulates a new plan, and again executes its move. This process is then repeated for as long as the system is in operation.

The difficulty in making a plan (1.2) lies mainly in the determination of the policies $K_1, \ldots, K_{N-1}$. These are functions that can depend on (y, θ) in a way that is a-priori unknown and possibly highly complex. If there is no uncertainty in the future evolution of θ and the response of the system to a given control input can therefore be predicted with absolute certainty, this difficulty is resolved. In that case, it is sufficient to optimize over sequences of control actions, i.e., the plan (1.2) reduces to

$$\text{Plan}_{\text{open-loop}} = \{u_0, u_1, u_2, \ldots, u_{N-1}\}\,. \tag{1.3}$$

Due to the much simpler structure, optimizing over the sequence of control actions in (1.3) is easier than optimizing over the functions K_i in (1.2). This is the fundamental reason why, as mentioned in Section 1.2.3, applying MPC to an LTI or LTV system is much simpler than applying it to an LPV system.

In the LPV case, it is theoretically always possible to either find a "winning" strategy (i.e., one that steers the system from its initial condition to the desired reference while avoiding constraint violation), or to conclude that such a strategy does not exist. This can be done following the principles of dynamic programming (DP) (Bertsekas 2005). Although theoretically possible, doing this is unfortunately computationally intractable for all but the simplest cases. Hence, there is a need for approaches which are conservative[3] with respect to DP, but which can be implemented at a more reasonable computational expense.

1.3.2 Desirable properties

This section introduces the properties that an MPC strategy should possess in order to be considered for practical implementation. The two principal desirable properties are (i) guaranteed stability and (ii) computational tractability.

The first notion—stability—is related to obtaining guarantees on the behavior of the controlled system once it is connected in a feedback loop with the designed controller. Consider, for instance, again the wafer thermal control problem of Figure 1.8. If the initial temperature of the wafer is lower than the reference value, a naively designed controller could decide to apply full heating power to reach the target as quickly as possible. However, due to the way that the heat distributes itself throughout the wafer, applying too much heat too quickly can lead to violation of the maximum temperature constraint. Because it is not possible to cool down the wafer actively, recovering from this situation can be impossible. A stabilizing controller, on the other hand, is guaranteed to bring the temperature to its desired level without overheating.

In MPC, stability is related to so-called *recursive feasibility*. A recursively feasible controller is guaranteed to produce a well-defined control input at each time instant, i.e., an input that is such that constraints will not be violated now or in the future. Recursive feasibility is therefore a necessary condition implied in the definition of stability: a controller is called stable if it avoids constraint violation, and makes the output of the system converge to a desired reference or to a neighborhood thereof. Observe that the converse need not be true: a controller can be recursively feasible (i.e., avoid constraint violation at all times) but not stable (i.e., the output of the system does not converge to the reference).

The second notion—tractability—is related to how easy or difficult it is to make a plan (1.2). Consider again the wafer temperature control problem. Once the process of exposing the wafer has started, it is necessary to monitor the present wafer temperature and to adjust the control input provided by the heaters perhaps several times within each second. As explained before, in MPC, the control input to apply is decided through a decision-making process which can be formulated in terms of an optimization problem. When such a decision has to be made many times each second, it is of paramount importance that this optimization problem can be solved

[3] A solution method for a given set of problems is called "conservative" if it only considers solutions of a certain class, which may not include all possible solutions.

quickly enough. It is perhaps not surprising that some classes of optimization problems are much harder to solve than others. An extremely important class of problems that can be solved efficiently, are the convex optimization problems[4] (Boyd and Vandenberghe 2004). Therefore, the first part of the definition of tractability is the ability to formulate the decision-making process in terms of such a convex problem.

Besides convexity, the second aspect of tractability is scalability. Intuitively, predicting the near future is easier than predicting far ahead. Suppose that a predictive controller has been designed which plans ahead for five time steps (e.g., seconds), and suppose that the decision-making process has been formulated as a convex optimization that can be solved fast enough. The natural question to ask, then, is: can this convex problem still be solved reasonably fast if we increase the planning horizon to, e.g., ten seconds? Or to twenty? Preferably, the answer to this question is "yes", and scalability is therefore an essential part of the notion of tractability.

Summarizing, an MPC methodology is called "tractable", if its implementation involves the solution of a convex optimization problem whose size scales well (e.g., polynomially) in the problem dimensions.

Precise definitions of the properties that were informally introduced in this subsection are provided later in Section 3.4. Note that it is usually also desired that a controller can optimize performance: this could mean, for instance, that it can realize its objective in the most energy-efficient manner. However, before control performance can be considered in a meaningful way, stability must be ensured. Furthermore, a controller that can not be implemented because it is computationally intractable is useless even if in theory it would be able to achieve high performance. Thus, stability and tractability are considered to be the two most important properties that must always be satisfied first.

1.3.3 Classification of strategies

Before proceeding to the literature overview of Sections 1.3.4–1.3.5, it is useful to be aware of some of the distinctions that can be made between different types of approaches for determining the strategy (1.2). Two important ones, that are introduced here, are the distinction between robust- and scheduling strategies, and the distinction between open-loop and closed-loop strategies.

Robust- and scheduling strategies

The first distinction concerns the difference between robust- and scheduling strategies. In a "scheduling" strategy, the functions K_i in (1.2) are dependent on (y, θ): this requires the assumption that the current values of θ can be measured. A "robust" strategy does not require this assumption by making the functions K_i dependent on y only. Obviously, the usage of additional measured information from the system has the potential of improving control performance. The structure of the functions K_i may however become more complex if the dependence on θ is included, leading to a higher computational demand. Because in an LPV

[4]Linear programs, quadratic programs and semidefinite programs all belong to the class of convex optimization problems.

system, the scheduling variable is assumed to be measurable at each time instant, it makes sense to use scheduling strategies whenever this is computationally feasible.

Open- and closed-loop strategies

A second distinction is made between open-loop, and closed-loop (or "feedback") strategies. In an open-loop strategy, the functions K_i in the plan (1.2) are not really functions, but instead they degenerate into a sequence of control actions (1.3). Open-loop strategies can be used in the uncertain case, as long as it is verified that the single predicted control sequence leads to constraint satisfaction and stability for all admissible uncertainty realizations. This was done, for instance, in (Limón Marruedo et al. 2002; Pin et al. 2009) for the robust predictive control of non-linear systems. The computation of open-loop strategies is much easier than that of closed-loop strategies, but a major disadvantage is their inherent conservatism when uncertainty is present. This conservatism arises because *one* control sequence has to satisfy the constraints for *all* possible uncertainty scenarios. A closed-loop strategy does not have this inherent disadvantage, because the feedback functions K_i can generate different control inputs for different realizations of the uncertainty. Therefore, feedback strategies are chosen to be the focus of this thesis.

1.3.4 Dynamic programming and min-max formulations

In the min-max setting, the controller determines a strategy (1.2) that guarantees constraint satisfaction with respect to all possible future scheduling trajectories. The min-max formulation for linear systems subject to parametric uncertainty is discussed in (J. H. Lee and Z. Yu 1997) together with its solution through dynamic programming. This formulation is non-conservative in the sense that no restrictive structure is imposed on the policies in (1.2). Therefore, if a feasible strategy (i.e., one that is able to steer the system towards a desired reference while respecting constraints) exists, it will be found.

The computational cost of DP is however too high to be applied in practice for all but the simplest systems. Therefore, (J. H. Lee and Z. Yu 1997) also proposes a less computationally intensive, but sub-optimal, algorithm applicable to moving-average systems with an integrator.

It is possible to formulate a single convex optimization problem that is equivalent to the original min-max problem, thereby avoiding the need to compute dynamic programming solutions on-line. This was done for additively disturbed LTI systems in (Kerrigan and Maciejowski 2004; Scokaert and Mayne 1998). The central idea is to construct a "scenario tree" containing every possible extreme disturbance realization over the prediction horizon N. Then, by convexity, the existence of an admissible control sequence associated with every extreme realization implies the existence of a control sequence for all possible future disturbance realizations. Although this leads to convex optimization problems to be solved on-line, the number of variables scales exponentially with the prediction horizon. Therefore, these approaches are not to be called tractable if scalability is taken into account.

The work (Muñoz de la Peña et al. 2005) presents an approach based on the same type of scenario tree, but extends it to the class of general uncertain linear systems (i.e., with parametric uncertainties and additive disturbances). Instead of solving the problem directly, a more efficient algorithm is proposed using Bender's decomposition which can avoid the exponential increase in computation time.

Another alternative is to apply an explicit solution to the min-max problem. In explicit MPC, the solution of the optimization problem is pre-computed off-line as a piecewise affine function of the state. This is achieved through multi-parametric optimization (Alessio and Bemporad 2009; Pistikopoulos et al. 2011; Pistikopoulos 2012). On-line, all that has to be done is to search for the solution corresponding to the current measured state—and, in the LPV case, scheduling value—in a look-up table, and apply it to the system. Such solutions for uncertain linear systems were developed in (Bemporad et al. 2003; Diehl and Björnberg 2004), while (Besselmann et al. 2012) gives an explicit DP approach for the LPV case. For relatively simple systems, this is extremely efficient, because the major computational burden is shifted off-line and the required on-line computation reduces to the evaluation of a single piecewise affine function. However, as the system complexity increases, the complexity of the explicit solution grows rapidly: even in the much simpler LTI case, explicit MPC is usually limited to systems with up to two control inputs, less than ten states, and to short control horizons (Alessio and Bemporad 2009).

Due to the inherent complexity of solving the min-max problem without any approximations (through dynamic programming, or otherwise), there is a consensus in the literature that it is necessary to employ simplifying approximations. Such approximations, called "tractable" feedback strategies, are the topic of the next subsection.

1.3.5 Tractable feedback strategies

Introduction

The practical intractability of dynamic programming has inspired a tremendous amount of research aiming to find control solutions that strike a more favorable balance between computational complexity and achievable performance (Bemporad and Morari 1999; Mayne 2014, 2016; Raković 2015; Saltık et al. 2018). The min-max formulation described in Section 1.3.4 can be seen as one end of a spectrum (high complexity, highest possible performance), whereas the open-loop strategies mentioned in Sections 1.3.1 and 1.3.3 are at the other end (low complexity, low performance). The tractable feedback strategies discussed in the present section are all positioned somewhere in between these two extremes, i.e., they aim to achieve a trade-off between computational complexity and achievable control performance.

When referring to achievable control performance of an MPC design approach, what is meant is its degree of conservatism when compared to DP. Recall that a non-conservative controller can always find a feasible strategy if one exist. A conservative controller only looks for solutions within a restricted class, and may therefore fail to find a feasible strategy even if there is one. Furthermore, by restricting the class of solutions, such a controller may not be optimal in the sense that it does not achieve its goal in the "best" way possible (e.g., by using more energy-consuming control inputs than strictly necessary).

The aim of this section is not to present an exhaustive overview of all the available literature, but rather to sketch the main lines in the development on tractable feedback policies in the deterministic setting. More exhaustive surveys can be found in, e.g., (Kouvaritakis and Cannon

2016; Mayne 2014; Saltık et al. 2018). The overarching theme in the summarized research is the quest for control strategies that are computationally simple, yet achieve acceptable levels of performance. In light of the preceding discussions, special attention is given to

- The (possible) applicability of the methods to LPV systems;
- Their computational properties;
- Their potential for including information about the future scheduling variable evolution.

In this short survey, developments that share similar characteristics are grouped together. For this purpose, the following categorization has been made:

- Methods that optimize over linear state feedback policies;
- Methods based on interpolation;
- Methods that employ prediction dynamics;
- Methods based on tubes;
- Methods that optimize over disturbance-feedback policies.

Some of the mentioned approaches are not directly applicable to LPV systems, but are developed for different system classes such as additively disturbed LTI (LTI+) or LPU systems. These are included because it might be possible to adapt them to the LPV case. The survey ends with some concluding remarks commenting on the problems of reference tracking and the control of LPV embeddings.

Optimization over linear feedback policies

One of the earliest works in robust MPC that was highly influential is (Kothare et al. 1996). Therein, the principal idea is to replace the general strategy (1.2) with a policy consisting of a single robust, quadratically stabilizing, linear state feedback. This feedback is computed on-line as the solution of a set of linear matrix inequalities (LMIs). Constraint satisfaction is guaranteed by requiring that the state of the system is inside of a suitable level set of a single quadratic Lyapunov function corresponding to the optimized state feedback. The approach is applicable to parametrically uncertain systems, with the uncertainties described either in a polytopic or in a norm-bounded representation.

The ideas of (Kothare et al. 1996) have been modified and extended in various ways. The quasi-min-max approach of (Lu and Arkun 2000b) is an extension to the polytopic LPV case, where a measurement of the scheduling variable is available. This measurement is exploited by extending the policy of (Kothare et al. 1996) with one free control move and by allowing optimization over a parameter-dependent state feedback. A further generalization, which adds yet another free control move, is developed in (Lu and Arkun 2000a) which considers LPV systems subject to bounded rates of parameter variation.

The above mentioned methodologies require the on-line solution of a single semi-definite optimization problem. The prediction horizon is fixed and equal to zero in (Kothare et al. 1996), to one in (Lu and Arkun 2000b), and to two in (Lu and Arkun 2000a). This is computationally simple, but implies a considerable structural restriction on the strategy (1.2). Also, these short and fixed horizons offer little potential for including knowledge about possible future scheduling trajectories.

The robust approaches (Casavola et al. 2000; Schuurmans and Rossiter 2000) generalize (Kothare et al. 1996) to arbitrary prediction horizons, by superimposing N free control moves onto a linear state feedback[5]. In these approaches, similarly to (Kothare et al. 1996), the terminal state feedback is re-optimized at each sample. This terminal controller is subsequently used to update the sequence of feedbacks used during the N prediction steps at the next sample. An extension to the LPV case was made in (Casavola et al. 2002)[6].

The aforementioned approaches all require that the system is quadratically stabilizable, i.e., that there exists a linear state feedback with an associated single quadratic Lyapunov function. This requirement was relaxed in (Casavola et al. 2006, 2007), which considers invariant sets associated with parameter-dependent Lyapunov functions. Other approaches that employ parameter-dependent Lyapunov functions or less conservative LMI conditions, are, e.g., (Wada et al. 2006; S. Yu et al. 2012).

The mentioned formulations of (Casavola et al. 2002, 2007) support arbitrary prediction horizons N, and could therefore offer some possibilities for exploiting information about future scheduling trajectories. However, these algorithms suffer from a computational complexity that grows exponentially with N.

It was shown in (Casavola et al. 2004) for the parametrically uncertain case and in (Casavola et al. 2008) for the LPV case, that this exponential growth can be avoided if a norm-bounded description of the uncertainty or scheduling variable is employed. Although the used norm-bounded representation is equivalent to a certain polytopic representation, it nonetheless allows the use of an alternative description of uncertain predicted trajectories that does not result in exponential complexity.

Interpolation

General interpolation in robust MPC for LPU systems is considered in (Bacic et al. 2003). Several pairs of linear controllers and corresponding ellipsoidal invariant sets are computed off-line. This contrasts with the approaches discussed previously under "Optimization over linear feedback policies", where a terminal set- and corresponding controller are computed on-line as part of the optimization process. Typically, one of these controllers achieves high performance but yields a small invariant set, whereas the other controllers achieve lower performance but have larger corresponding invariant sets. On-line, based on the current measured state, the control input is computed as a suitable linear combination of the control inputs produced by the pre-designed local controllers. A similar approach, but using polyhedral

[5](Casavola et al. 2000) contains some mistakes in its stability result, which are corrected in (Pluymers 2006, Section 3.5).

[6](Casavola et al. 2002) also contains a mistake concerning the initialization of the controller, see (Ding and Huang 2007) for a solution.

sets, is proposed in (Pluymers et al. 2005a). In principle, the interpolation-based approaches can be extended from the robust to the LPV case by adding the assumption that there is a scheduling variable that can be measured (Rossiter et al. 2007).

In these approaches, the "prediction" horizon is equal to zero as it was in (Kothare et al. 1996). Thus, there is little potential for including expectations on the future evolution of the scheduling signal.

A robust MPC methodology for LPU systems based on interpolation that does allow for arbitrary prediction horizons N is presented in (Pluymers et al. 2005b; Wan et al. 2006) [7]. Similarly to (Bacic et al. 2003), two controller/set-pairs are computed off-line. The terminal set used on-line is constructed as a continuous interpolation between these two sets, preferring the smallest set whenever this is feasible, but resorting to a larger set otherwise. The state predictions over the horizon are based on a scenario tree, hence the computational complexity is exponential in N.

The use of prediction dynamics

An alternative, computationally highly efficient, robust MPC formulation for polytopic parametrically uncertain linear systems was proposed in (Kouvaritakis et al. 2000). Therein, the control policy consists of a series of control actions upon a pre-designed linear state feedback. A "lifted" prediction system, which includes both the state of the original system and N free control actions as part of its state vector, is defined. Constraint satisfaction is achieved by requiring that the full lifted state of this prediction system is contained inside of a suitable ellipsoidal invariant set, which is designed off-line. The computational load of this approach is similar to that of normal LTI MPC, but the requirement that the lifted state must be included in a pre-designed ellipsoid can be conservative.

An off-line design procedure, aimed at "optimizing" the prediction dynamics of (Kouvaritakis et al. 2000), is proposed in (Cannon and Kouvaritakis 2005). In this way, it is possible to maximize the volume of the ellipsoidal invariant set, which leads to a less conservative MPC. An alternative strategy of optimizing the prediction dynamics, aimed at reducing the sensitivity of the realized closed-loop cost to model uncertainty, is developed in (Cheng et al. 2013) by means of a Youla-type parameterization.

In (Pluymers 2006, Section 4.3), the approach of (Kouvaritakis et al. 2000) is modified to use polyhedral invariant sets instead of ellipsoidal ones. Because the state dimension of the lifted system grows linearly with N, the representation complexity of the polyhedral invariant sets can grow exponentially with N in the worst case. In (Pluymers 2006, Chapter 5) it is shown that it is possible—under some conditions—to compute reduced-complexity polyhedral invariant sets for the lifted system. The complexity of these sets grows only linearly in N, leading to a scalable algorithm.

Because the full lifted state vector—which includes the control degrees of freedom—must be inside of an invariant set that is computed off-line as part of the controller design, there is no obvious way in which these algorithms can incorporate knowledge on possible future scheduling trajectories that becomes available while the system is running.

[7] The original idea appeared in (Wan and Kothare 2003), but that paper contained an error in its recursive feasibility proof, which was corrected later in (Pluymers et al. 2005b; Wan et al. 2006).

Tube-based approaches

Yet another paradigm devised to reduce complexity with respect to the dynamic programming solution is the so-called tube model predictive control (TMPC). In MPC, tubes were employed at first in the robust control of LTI systems subject to additive uncertainties. Influential works in this area include (Langson et al. 2004; Mayne et al. 2005; Raković et al. 2012a). Although the exact form of employed control policy differs, these works share the same common idea of using fixed-complexity over-approximations of predicted uncertain trajectories. By doing so, the representation complexity of the predicted trajectories does not grow exponentially with the horizon as is the case for a scenario tree.

The same principle can also be applied to uncertain linear systems (Fleming et al. 2015; Muñoz-Carpintero et al. 2015). In these works, uncertain predicted trajectories are over-approximated by scaled and translated versions of pre-designed polytopic sets. Because the complexity of these sets stays constant along the prediction horizon, the size of the overall optimization problem that has to be solved increases linearly with N.

Due to its favorable linear scaling behavior, the tube-based framework offers good possibilities for including future information on the possible scheduling trajectories into the formulation. However, this case has not yet been treated in literature, as the mentioned approaches consider robust control with respect to unmeasurable additive disturbances (the LTI+ case) or to unmeasurable parametric uncertainties (the LPU case).

Disturbance-feedback policies

Another approach that can be applied for the robust control of LTI systems subject to additive disturbances, is to optimize over feedback policies that are functions of the past disturbances. (Although the current value of the disturbance is not directly measurable, its previous value can be inferred as the difference between the predicted and measured system state).

Such an approach, where the considered feedbacks are affine functions of the past disturbances, is presented, e.g., in (Löfberg 2003). The disturbance-feedback concept is in some sense similar to the idea of “recourse” in robust optimization theory (Ben-Tal et al. 2004).

In (Goulart et al. 2006) it is shown that the optimization of policies that are affine functions of past disturbances is, in fact, equivalent to the optimization over linear dynamic state feedback policies. A remarkable result is that the optimization over dynamic linear state feedbacks is a non-convex problem, whereas the equivalent problem in terms of disturbance feedbacks is convex. The number of variables and constraints in the convex problem grows quadratically with N, so the approach can be considered tractable. Furthermore, it is possible to re-formulate the optimization problem in an equivalent form with a sparse structure that can be exploited to significantly increase computational efficiency (Goulart et al. 2008). In (Raković et al. 2012b), the approach of (Goulart et al. 2006) is generalized further by dropping the restriction that the disturbance feedback is affine.

The optimization over “disturbance”-feedback policies has only been considered for LTI systems subject to additive disturbances. It is a question whether or not a convex re-formulation of the state feedback synthesis problem can be obtained in the LPV case.

Concluding remarks: tracking and embeddings

All the MPC methods reviewed so far focused on stabilization: that is, the regulation of the state of the controlled system towards the origin. As was mentioned at the beginning of this chapter, another important control problem besides regulation is that of tracking a reference. Furthermore, in the reviewed approaches, the control of non-linear systems represented by an LPV embedding (see Section 1.2.1) has not yet been considered. Later, these more specific developments will be reviewed in more detail in Chapter 7 and Chapter 8: for completeness, a few approaches are mentioned here.

Robust tracking MPC approaches for LPU systems were developed in, e.g., (Pannocchia 2004; Y. J. Wang and Rawlings 2004). It is shown that offset-free tracking is achieved if the uncertainty (corresponding to θ in an LPV system) is time-invariant. The papers (Alvarado et al. 2007; Limon et al. 2010) consider the tracking of piecewise constant references for LTI+ systems. Using a tube-based approach, asymptotic convergence of the output towards a bounded region around the reference is established.

In the MPC scheme for embeddings of (Lu and Arkun 2002), which is based on (Lu and Arkun 2000b), it is assumed that the non-linear system can be represented by an LPV model obtained as a family of linearized models around various operating points. It is shown that if the optimization remains feasible, the closed loop is asymptotically stable. In the method of (Chisci et al. 2003), the non-linear dynamics are embedded in an LPV representation with a discrete scheduling set, effectively representing a collection of uncertain linear models. ROV bounds on the state variable are imposed in a way reminiscent of classical gain scheduling (Rugh and Shamma 2000). The work (Casavola et al. 2003) presents an MPC algorithm applicable to LPV embeddings based on (Casavola et al. 2002). A known and bounded ROV on the scheduling variable is assumed.

In (Cisneros et al. 2016), it is proposed to control non-linear systems embedded in an LPV representation using an “iterative” approach. Based on an initial guess of the future scheduling trajectory, a simple *linear time-varying* (LTV) MPC problem is solved. The resulting optimal state- and input trajectories are used to generated a new scheduling trajectory, and the procedure is iterated until the predicted trajectories converge.

1.4 Research questions

Throughout this chapter, a thermal control problem from semiconductor lithography was used as a running example to demonstrate several concepts. As introduced previously, the main characteristics of this application are

- The presence of constraints (the wafer can be heated up, but not actively cooled down);
- The presence of linear parameter-varying dynamics (the response of the thermal distribution in the wafer to heat inputs depends on the position of the IH);
- The availability of detailed—but not exact—knowledge on possible future scheduling trajectories, that can not be captured by merely considering bounds on the scheduling

ROV (the IH is controlled to follow approximately a prescribed reference trajectory that is known in advance).

As the system is constrained and exhibits LPV dynamics, the use of an LPV MPC method appears to be a natural solution for control. Preferably, the employed control method should satisfy the desirable properties outlined in Section 1.3.2. Of the approaches surveyed in Section 1.3.2, the property of "stability" is shared by all, whereas the property of "tractability" is satisfied to varying degrees.

What is notable, however, is that none of the reviewed approaches has a built-in capability to take into account all types of scheduling dynamics as described in Section 1.2.2. Only a-priori specified bounds, and in some cases bounds on the rate of variation of, the scheduling variable are typically assumed. Thus, the discussed approaches do not provide a structured way to deal with more detailed information on the future scheduling variations that becomes available on-line, such as the well-predictable position dependence in the considered thermal control problem or the possible scheduling trajectories in an LPV embedding.

Note that such a feature is not only restricted to this specific thermal problem, but can be observed in several other applications. For instance, in an airliner the future altitude profile is often pre-planned making it possible to predict expected changes in dynamic pressure (a scheduling variable). In robotics, path planning is used to design feasible motion profiles around which a controller has to suppress disturbances; in chemical process control operating points and transients are planned in advance by a higher-level optimization layer.

Therefore, there exists a need to develop a tractable method for the constrained control of LPV systems that can exploit all available knowledge on the possible behavior of the scheduling variable. To this purpose, the following main research question is posed.

> Q_0. How to design a stable and tractable LPV MPC strategy for regulation and tracking that can exploit knowledge about the possible future evolution of the scheduling variable?

To be able to answer this main question, it is necessary to first define how this "knowledge" on the future evolution of the scheduling variable can be described mathematically.

> Q_1. How to construct a practically useful description of the uncertain future evolution of the scheduling variable?

Question Q_1 will be addressed in Chapter 3, and leads to the introduction of the notion of "anticipative" control. Answering this question is a necessary condition to answer the main research question Q_0. The next step is then to discover how the obtained type of model can actually be integrated into an overall control strategy which achieves the desirable properties discussed before.

> Q_2. How to use the model developed as the answer to Question Q_1 in an LPV MPC strategy for stabilization? In particular: how should this strategy be designed, such that the resulting controller regulates the measured output towards the origin?

In Chapter 3, it is shown that a tube-based MPC approach using a homothetic tube parameterization can provide an answer to Q_2. A more general answer is subsequently provided in Chapter 4, which introduces a tube parameterization based on the concept of finite-step contractive sets (Gilbert and Kolmanovsky 2002).

As it will turn out, the parameterization of the tube in terms of (periodically) homothetic sets is the crucial design choice that allows the formulation of an MPC algorithm that is computationally tractable. This leads to a tractable feedback strategy that fits in the large family of approaches that were reviewed in Section 1.3.5. Just like those approaches, the developed tube-based LPV MPC solution with homothetic tubes achieves its own particular trade-off between computational complexity and achievable control performance. However, it must be realized that the selected homothetic parameterization is just one out of a multitude of possibilities. A natural follow-up question is then if there exist different tube parameterizations that can possibly lead to algorithms that achieve different trade-offs.

> Q_3. Is it possible to introduce an extra degree-of-freedom in the design of the tube parameterization, that can be used to achieve different trade-offs between computational complexity and achievable performance?

Question Q_3 will be addressed in two different ways in Chapters 5 and 6. The measure of control performance that will be adopted in these chapters is the size of the domain of attraction, i.e., the set of initial system conditions for which a feasible control input can be determined.

In Question Q_2, the objective of the controller was restricted to the regulation of the output of the system towards the origin. It is therefore questioned if the tracking objective that was part of Question Q_0 can be included in the developed formulations as well:

> Q_4. How can the developed strategies for "stabilization" be adapted to the "tracking" case?

A first step towards a full answer to Q_4 is provided in Chapter 7, where the problem of tracking a constant (but non-zero) reference signal is considered.

The preceding research questions were all formulated in a strict LPV setting, that is a setting in which the scheduling variable is considered to be a free signal being fully independent of the input- and output variables of the system. However, as outlined in Section 1.2.2, the scheduling variable can also be used to obtain an LPV approximation of a non-linear system— in fact, this was one of the main reasons why the development of the LPV framework was started in the first place. It is therefore relevant to see if the solutions obtained as answers to the previous questions can be applied in this case as well, or if they should be modified—and if so, in what sense. This motivates the next and final research question, which is addressed in Chapter 8:

> Q_5. How to apply the developed LPV MPC approaches to the control of LPV embeddings, such that the desirable properties of stability and tractability are retained when the controller is connected in closed-loop with the real non-linear system?

1.5 Outline and contributions of the thesis

This thesis is composed of several chapters beyond this introductory chapter. It is advised that Chapters 1-3 be read first and in order, after which all the remaining chapters can be read independently. Some chapters are partially based on papers which, at the time of writing of this thesis, were already published or were still in preparation. A list of these publications is included on page 241.

Chapter 2 Chapter 2 contains the preliminary material which is necessary for the understanding of the rest of this thesis. Fundamental mathematical notations and concepts related to sets and functions are presented. Linear parameter-varying (LPV) systems are formally introduced, and the related important concepts such as LPV state-space representations, stability, and invariant sets are discussed. The embedding of non-linear systems in LPV representations is also covered. Finally, some of the fundamental ideas of model predictive control (MPC) are reviewed.

Chapter 3 Chapter 3 proposes the general framework of anticipative MPC for LPV systems based on the construction of so-called tubes. The problem setting that is adopted throughout most of the thesis is defined and, as a first contribution, the notion of "anticipative" control is introduced as a preliminary answer to Research Question Q_1. Based on this, it is argued that a tube-based MPC paradigm has the potential to provide a fitting solution to Question Q_2. As a second contribution, an anticipative LPV MPC algorithm based on the construction of homothetic tubes is presented. This approach can be seen as a preliminary solution to the main research question.

Chapter 4 This chapter addresses the problem of guaranteeing recursive feasibility and closed-loop stability in a receding-horizon LPV tube MPC algorithm. Stability can be guaranteed through the construction of an appropriate terminal set and terminal cost. The main contribution of this chapter is to provide a new terminal set- and cost construction, and an associated periodic tube parameterization, based on the concept of finite-step λ-contraction. This is a more relaxed variant of the usual notion of λ-contraction, which was used in Chapter 3. This chapter provides an answer to Research Question Q_2, and the stability of the basic homothetic tube-based algorithm from Chapter 3 follows as a special case from this result.

Chapter 5 In this chapter, as one answer to Question Q_3, a so-called heterogeneous tube parameterization structure is proposed. Instead of parameterizing all tube cross-sections in the same way, this setup allows to use different parameterizations for each cross-section. Conditions under which such a combination of parameterizations leads to a recursively feasible algorithm, are presented. The combination of various parameterizations into one tube has certain similarities to the well-known concept of "move blocking" in nominal MPC. It is shown that the proposal has the potential to improve computational properties—in terms of the complexity/performance trade-off—of tube-based LPV MPC.

Chapter 6 In Chapter 6 an alternative answer to Research Question Q_3 is proposed, i.e., an alternative way of designing a tube parameterization. The starting point is a given set of initial conditions that represents an inner-approximation of the desired domain of attraction of the controller. Then, an off-line procedure based on sparsity-inducing ℓ_1-optimization is used to determine a collection of tube parameterizations with a reduced number of degree of freedom. On-line, on the basis of the measured state- and scheduling values, a complete tube parameterization is constructed as the combination of a suitable selection from the parameterizations that were computed off-line. It is shown that this construction of a reduced-complexity parameterization leads to a domain of attraction that is at least as large as the set of initial states that was assigned at the beginning.

Chapter 7 This chapter addresses Research Question Q_4, i.e., the application of the previously developed control methods to the tracking problem. A preliminary answer is provided by considering the constrained tracking of constant, but non-zero, references. In an LPV system, variations in the scheduling signal causes unavoidable tracking errors. It is proposed to bound these errors through the construction of so-called bounded-error invariant sets. By incorporating these sets into the tube-based MPC framework, it becomes possible to track non-zero references with an a-priori guaranteed bound on the error induced by the scheduling variations.

Chapter 8 Chapter 8 addresses Question Q_5, and shows how the previously developed tube MPC solutions can be applied for the control of non-linear systems described in terms of an LPV embedding. In principle, this is already possible without modifications, but might be conservative because the underlying relationship between state- and scheduling variables is ignored. It is therefore proposed to employ sequences of state constraints to bound the predicted state trajectories. These bounds can, in turn, be used to construct corresponding sets of scheduling variables. These will be smaller than the sets that would be obtained if the state/scheduling-relation were not considered. In contrast to several existing approaches from the literature that rely on local linearizations, the resulting algorithm is guaranteed to be recursively feasible and stabilizing for the original non-linear system that was described in terms of an LPV embedding.

Chapter 9 The purpose of this chapter is to analyze the performance of the developed algorithms on a simulation example inspired by the wafer temperature control problem.

Particular attention is given to the effect of including the “anticipative” knowledge into the control problem.

Chapter 10 Lastly, Chapter 10 summarizes the research described in the thesis. Several suggestions for further research are proposed.

Chapter 2

Preliminaries

THIS CHAPTER presents the preliminary material on which the results in the upcoming chapters are based. In Section 2.1, fundamental mathematical notations and concepts related to sets and functions are presented. Linear parameter-varying (LPV) systems are formally introduced in Section 2.2. Related concepts such as linear parameter-varying (LPV) state-space representations, stability, and invariant sets are discussed. The embedding of non-linear systems in LPV representations is covered in Section 2.3. Finally, Section 2.4 reviews the fundamental ideas of model predictive control (MPC).

2.1 Mathematical notation and definitions

This section gives the basic mathematical notations, and the definitions of important fundamental concepts that will be used throughout this thesis.

2.1.1 General notation

Some possibly non-standard mathematical notation and conventions are specified below.

- The set of real numbers is denoted by $\mathbb{R}$, and the set of non-negative real numbers by $\mathbb{R}_+$.
- Closed and open intervals on $\mathbb{R}$ are denoted by $[a, b]$ and (a, b), respectively.
- The symbol $\mathbb{N}$ is used to denote the set of non-negative integers (i.e., the integers including zero).
- Closed and open index sets on $\mathbb{N}$ are defined as $[a..b] = \{i \in \mathbb{N} \mid a \le i \le b\}$ and $[a..b) = \{i \in \mathbb{N} \mid a \le i < b\}$, respectively.
- Inequalities between vector and scalar quantities are interpreted element-wise. That is, if $v \in \mathbb{R}^n$ and $\alpha \in \mathbb{R}$, then $v \le \alpha$ means that every element of v is smaller than or equal to α.
- The i-th element of a vector $v \in \mathbb{R}^n$ is denoted by v_i. If some ambiguities could arise due to a double subscript, the notation $[v]_i$ is used.

- A vector norm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}_+$ is a function that satisfies for all $x, y \in \mathbb{R}^n$ and all $\alpha \in \mathbb{R}$ the properties (i) $\|x + y\| \leq \|x\| + \|y\|$, (ii) $\|\alpha x\| = |\alpha|\|x\|$, and (iii) $\|x\| = 0$ if and only if $x = 0$. In particular, for a vector $x \in \mathbb{R}^n$, the 1-norm is defined as $\|x\|_1 = \sum_{i=1}^n |x_i|$, the 2-norm as $\|x\|_2 = \sqrt{\sum_{i=1}^n x_i^2}$, and the ∞-norm as $\|x\|_\infty = \max_{i \in [1..n]} |x_i|$.
- Sequences are denoted compactly as $\{X_i\}_{i=a}^b = \{X_a, X_{a+1}, \ldots, X_b\}$.

2.1.2 Sets

Basic set-related notation

Some useful basic notions related to sets are the following:

- The scalar multiple of a set $\mathcal{A} \subseteq \mathbb{R}^n$ is $\lambda\mathcal{A} = \{\lambda a \mid a \in \mathcal{A}\}$ for real numbers λ.
- Let $x, y \in \mathcal{A}$. The set $\mathcal{A}$ is convex if and only if $\lambda x + (1 - \lambda)y \in \mathcal{A}$ for all $\lambda \in [0, 1]$.
- The convex hull of a set $\mathcal{A}$ is the intersection of all convex sets that contain $\mathcal{A}$, and it is denoted by $\mathrm{convh}\{\mathcal{A}\}$.
- The notation $\mathcal{B} \subseteq \mathcal{A}$ means that $\mathcal{B}$ is a subset of, and can be equal to, $\mathcal{A}$. If $\mathcal{B}$ is a proper subset, i.e., if there is at least one element in $\mathcal{A}$ that is not in $\mathcal{B}$, this is denoted as $\mathcal{B} \subset \mathcal{A}$.
- A set is called a proper C-set, or PC-set for short, if it is convex, compact (i.e., closed and bounded), and has a non-empty interior that includes the origin.
- The boundary of a set $\mathcal{A} \subseteq \mathbb{R}^n$ is denoted by $\partial\mathcal{A}$. Informally, the boundary of $\mathcal{A}$ is the set of all points $a \in \mathcal{A}$ such that any neighborhood of a contains at least one element in $\mathcal{A}$ and at least one element not in $\mathcal{A}$.
- The power set of $\mathcal{A}$ is the set of all subsets of $\mathcal{A}$ (including the empty set and $\mathcal{A}$ itself), and is denoted by $2^{\mathcal{A}}$.
- For some $a \in \mathbb{R}^n$ and $b \in \mathbb{R}$, the set $\{x \in \mathbb{R}^n \mid a^\top x = b\}$ is the *hyperplane* corresponding to the pair (a, b). This hyperplane bounds a *half-space*, which is the set $\{x \in \mathbb{R}^n \mid a^\top x \leq b\}$.

Minkowski addition

The Minkowski sum of two sets $\mathcal{A} \subseteq \mathbb{R}^n$ and $\mathcal{B} \subseteq \mathbb{R}^n$ is

$$\mathcal{A} \oplus \mathcal{B} = \{a + b \mid a \in \mathcal{A}, b \in \mathcal{B}\} .$$

If $a \in \mathbb{R}^n$ is a vector, with a slight abuse of notation, define

$$a \oplus \mathcal{B} = \{a\} \oplus \mathcal{B} = \{a + b \mid b \in \mathcal{B}\} .$$

The Minkowski sum of convex sets is convex. If $\mathcal{A}$ is convex, then $\lambda\mathcal{A} \oplus \mu\mathcal{A} = (\lambda + \mu)\mathcal{A}$. For more properties, see (Schneider 2013b, Chapter 3).

Hausdorff distance

The Hausdorff distance between two sets $\mathcal{A} \subseteq \mathbb{R}^n$ and $\mathcal{B} \subseteq \mathbb{R}^n$ is defined as

$$d_H(\mathcal{A}, \mathcal{B}) = \max \left\{ \sup_{a \in \mathcal{A}} \inf_{b \in \mathcal{B}} \|a - b\|, \sup_{b \in \mathcal{B}} \inf_{a \in \mathcal{A}} \|a - b\| \right\}$$

where $\| \cdot \|$ can be any vector norm on $\mathbb{R}^n$. The Hausdorff distance between a set $\mathcal{A} \subseteq \mathbb{R}^n$ and the origin is therefore

$$d_H^0(\mathcal{A}) = d_H(\mathcal{A}, \{0\}) = \sup_{a \in \mathcal{A}} \|a\|.$$

When for actual computations of the distance a particular norm is used, this will be made clear in the context.

Polyhedra and polytopes

A polyhedron is a convex set $\mathcal{P} \subseteq \mathbb{R}^n$ that can be represented as the intersection of finitely many half-spaces, i.e., admits a representation of the form

$$\mathcal{P} = \{x \in \mathbb{R}^n \mid Hx \leq h\} \text{ for some } H \in \mathbb{R}^{r \times n},\ h \in \mathbb{R}^r \tag{2.1}$$

where $r \in \mathbb{N}$ is the number of hyperplanes defining the representation. A polytope is a compact polyhedron and can, in addition to (2.1), equivalently be described as the convex hull of finitely many vertices as

$$\mathcal{P} = \text{convh}\{\bar{p}^1, \ldots, \bar{p}^q\} \text{ for some vectors } \bar{p}^i \in \mathbb{R}^n, \tag{2.2}$$

where $q \in \mathbb{N}$ is the number of vertices in the representation. If a set $\mathcal{P}$ is a polytope, then representation (2.1) is referred to as its H-representation (from “half-space”), and representation (2.2) is called its V-representation (from “vertex”). It is always possible to convert the V-representation of a given polytope into its equivalent H-representation (and vice-versa), but in dimensions greater than two, the complexity of the two representations may be significantly different[1]. An H- or V-representation of a polytope can be redundant, which means that the representation is described by more hyperplanes or vertices than necessary. It is possible to eliminate these redundant hyperplanes or vertices to obtain an equivalent representation of minimal size.

An accessible introduction to the important problems in polyhedral computation, which includes the conversion between V- and H-representations and redundancy removal, can be found in (Fukuda 2004). More general background on polytopes may be found in, e.g., (Grünbaum 2003; Ziegler 1995).

Because $\mathcal{P}$ in (2.2) is a convex set, any point $p \in \mathcal{P}$ can be represented as a convex combination of the extreme points (i.e., vertices) of $\mathcal{P}$. For future reference, this is summarized in the following definition of convex multipliers.

[1] An n-rhombus in $\mathbb{R}^n$ for instance admits a V-representation described by $2n$ vertices, but its equivalent H-representation consists of the intersection of 2^n half-spaces. A general bound on the number of hyperplanes required to represent a polytope given in vertex representation (and vice versa) is provided by McMullen’s upper bound theorem, see, e.g., (Ziegler 1995, Section 8.4).

Definition 2.1. *Let* $\mathcal{P} = \operatorname{convh}\left\{\bar{p}^1, \ldots, \bar{p}^{q_\mathrm{p}}\right\} \subset \mathbb{R}^n$. *Then* $\operatorname{Convm}(\cdot \mid \mathcal{P}) : \mathcal{P} \to 2^{\mathbb{R}_+^{q_\mathrm{p}}}$ *is the set-valued function*

$$\operatorname{Convm}(p \mid \mathcal{P}) = \left\{ \eta \in \mathbb{R}_+^{q_\mathrm{p}} \,\middle|\, \sum_{i=1}^{q_\mathrm{p}} \eta_i \bar{p}^i = p, \sum_{i=1}^{q_\mathrm{p}} \eta_i = 1 \right\}.$$

Further, define the function $\operatorname{convm}(\cdot|\mathcal{P}) : \mathcal{P} \to \mathbb{R}_+^{q_\mathrm{p}}$ *as*

$$\operatorname{convm}(p \mid \mathcal{P}) = \arg\inf_{\eta} \|\eta\| \text{ subject to } \eta \in \operatorname{Convm}(p \mid \mathcal{P}).$$

In general the multipliers η are non-unique. In Definition 2.1, all possible multipliers are described in terms of the set Convm$(\cdot|\cdot)$, whereas in the definition of convm$(\cdot|\cdot)$ a unique choice is made by minimizing a norm of η.

2.1.3 Functions

Comparison functions

The function classes $\mathcal{K}$, $\mathcal{K}_\infty$, $\mathcal{L}$ and $\mathcal{KL}$ of so-called comparison functions are defined as follows:

- A function $f : \mathbb{R}_+ \to \mathbb{R}_+$ is of class $\mathcal{K}$ if it is continuous, strictly increasing, and $f(0) = 0$.
- It is in class $\mathcal{K}_\infty$ if, in addition to being in class $\mathcal{K}$, $\lim_{\xi\to\infty} f(\xi) = \infty$.
- A function $g : \mathbb{R}_+ \to \mathbb{R}_+$ is of class $\mathcal{L}$ if it is continuous, strictly decreasing, and $\lim_{\xi\to 0} g(\xi) = 0$.
- A function $h : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}_+$ is said to be in class $\mathcal{KL}$ if it is class-$\mathcal{K}$ in its first argument and class-$\mathcal{L}$ in its second argument.

Comparison functions are useful tools in stability analysis. An overview of their properties can be found in (Kellett 2014).

Parameterized functions

A function $f : \mathcal{A} \to \mathcal{B}$ maps elements $a \in \mathcal{A}$ into elements $b \in \mathcal{B}$. Sometimes this mapping can depend on one or more parameters. This is denoted by $f(\cdot|p) : \mathcal{A} \to \mathcal{B}$, where p is a parameter.

Example 2.1. *A simple example of a parameterized function is* $f(\cdot|c) : \mathbb{R} \to \mathbb{R}$, $f(x|c) = cx$, *with the parameter being the multiplication factor* c. *Another example is the function* $g(\cdot|d) : \mathbb{R}^d \to \mathbb{R}$, $g(x|d) = x^\top x$, *where the parameter* d *defines the dimension of its first argument.*

Second-order functions

In this thesis, to describe parameterized control policies, use will be made of so-called second-order functions. A function $f : \mathcal{A} \to \mathcal{B}$ is called *first-order* if both $\mathcal{A}$ and $\mathcal{B}$ are subsets of real vector spaces. The function f is called *second-order* if it returns another first-order function, i.e., if $\mathcal{A}$ is a subset of a real vector space but $\mathcal{B}$ is a subset of all first-order functions $g : \mathcal{C} \to \mathcal{D}$ (i.e., with $\mathcal{C}, \mathcal{D}$ being subsets of real vector spaces). The concept of second-order (or, indeed, higher-order) functions is widely used in computer science as a useful way of abstraction (Abelson et al. 1996, Chapter 1.3).

Example 2.2. *A simple second-order function is* $g : \mathbb{R} \to (\mathbb{R}^n \to \mathbb{R}^n)$, $g(c) = (x \mapsto cx)$. *It is then possible to say, e.g.,* $h(x) = g(2)$ *meaning that* h *is the function* $h : \mathbb{R}^n \to \mathbb{R}^n$, $h(x) = 2x$.

Gauge functions

The gauge- or Minkowski-function corresponding to a PC-set $\mathcal{A} \subset \mathbb{R}^n$ is $\psi_{\mathcal{A}}(x) = \inf\{\gamma \geq 0 \,|\, x \in \gamma\mathcal{A}\}$. A generalized "set"-gauge function can be defined as follows.

Definition 2.2. *The set-gauge function* $\Psi_{\mathcal{A}} : 2^{\mathbb{R}^n} \to \mathbb{R}_+$ *corresponding to a PC-set* $\mathcal{A} \subset \mathbb{R}^n$ *is*

$$\Psi_{\mathcal{A}}(X) = \sup_{x \in X} \psi_{\mathcal{A}}(x) = \inf\left\{\gamma \geq 0 \mid X \subseteq \gamma\mathcal{A}\right\}.$$

The gauge- and set-gauge functions can be bounded by $\mathcal{K}_\infty$-functions, as shown in the following lemma:

Lemma 2.1. *Let* $S \subset \mathbb{R}^n$ *be a PC-set. Then, the following properties hold:*

(i) There exist $\mathcal{K}_\infty$*-functions* $\underline{\psi}, \overline{\psi}$ *such that* $\forall x \in \mathbb{R}^n : \underline{\psi}(\|x\|) \leq \psi_S(x) \leq \overline{\psi}(\|x\|)$,

(ii) There exist $\mathcal{K}_\infty$*-functions* $\underline{\Psi}, \overline{\Psi} \in \mathcal{K}_\infty$ *such that for all compact and convex sets* $X \subset \mathbb{R}^n$: $\underline{\Psi}\left(d_H^0(X)\right) \leq \Psi_S(X) \leq \overline{\Psi}\left(d_H^0(X)\right)$.

Proof. Let $\|\cdot\|$ be an arbitrary vector norm and denote $\mathcal{B} = \{x \mid \|x\| \leq 1\}$. Note that $\psi_{\mathcal{B}}(x) = \|x\|$. For sets $S_1, S_2 \subset \mathbb{R}^n$ with $S_1 \subseteq S_2$, it holds $\psi_{S_1}(x) \geq \psi_{S_2}(x)$ for all $x \in \mathbb{R}^n$ (Raković and Lazar 2012, Lemma 1). Because S is a PC-set, $\exists a, b \in \mathbb{R}_+$ such that $a\mathcal{B} \subseteq S \subseteq b\mathcal{B}$. Thus, $\forall x \in \mathbb{R}^n : b^{-1}\psi_{\mathcal{B}}(x) \leq \psi_S(x) \leq a^{-1}\psi_{\mathcal{B}}(x)$, i.e., statement (i) holds with $\underline{\psi}(\xi) = b^{-1}\xi$ and $\overline{\psi}(\xi) = a^{-1}\xi$. Next, observe that $\Psi_{\mathcal{B}}(X) = \sup_{x \in X} \|x\| = d_H^0(X)$. For sets $S_1, S_2 \subset \mathbb{R}^n$ with $S_1 \subseteq S_2$, it similarly holds $\Psi_{S_1}(X) \geq \Psi_{S_2}(X)$ for all compact and convex $X \subset \mathbb{R}^n$. Hence, for all compact and convex $X \subset \mathbb{R}^n : b^{-1}\Psi_{\mathcal{B}}(X) \leq \Psi_S(X) \leq a^{-1}\Psi_{\mathcal{B}}(X)$, i.e., statement (ii) follows with $\underline{\Psi}(\xi) = b^{-1}\xi$ and $\overline{\Psi}(\xi) = a^{-1}\xi$. □

Complexity and big-O notation

To quantify the complexity of algorithms, the big-O notation will be employed. For a function $f : \mathbb{R} \to \mathbb{R}$, writing

$$f(n) = O\left(g(n)\right)$$

means that as $n \to \infty$, the absolute value of $f(n)$ is bounded from above by the absolute value of $g(n)$ multiplied by some scalar factor. Formally, $f(n) = O(g(n))$ if and only if there exist constants $C > 0$, $n_0 \in \mathbb{R}$ such that

$$\forall n \geq n_0 : \ |f(n)| \leq C\,|g(n)|\,.$$

Example 2.3. *Let* $f : \mathbb{N} \to \mathbb{N}$, $f(n) = 1 + 10n + n^2$. *Then,* $f(n) = O(n^2)$ *because for all* $n \geq 11$, $f(n) \leq 2n^2$.

In the context of algorithms, the function $f : \mathbb{R} \to \mathbb{R}$ can represent a measure of complexity of the algorithm. Such a measure can be, for instance, the time complexity (number of required arithmetic operations) or a higher-order measure such as the number of variables and constraints in an optimization problem that must be solved.

Definition 2.3. *A function* $f : \mathbb{R} \to \mathbb{R}$ *is said to be polynomially (linearly) bounded if there exists a constant* $c \geq 1$ ($c = 1$) *such that* $f(n) = O(n^c)$. *If* f *represents a complexity measure of some algorithm, this algorithm is said to have polynomial (linear) complexity in n.*

Definition 2.4. *A function* $f : \mathbb{R} \to \mathbb{R}$ *is said to be exponentially bounded if there exists a constant* $a > 0$ *such that* $f(n) = O(a^n)$. *If* f *represents a complexity measure of some algorithm, this algorithm is said to have exponential complexity in n.*

Typically, algorithms whose complexities are polynomially bounded are considered scalable, whereas exponentially bounded complexity is highly undesirable.

2.2 Linear parameter-varying systems

This section introduces the concept of LPV systems. The formal definition of an LPV system is given in Section 2.2.1, whereas the common LPV state-space (LPV-SS) representation is presented in Section 2.2.2. Constrained LPV representations, which are the main focus of MPC approaches developed in this thesis, are introduced in Section 2.2.3. Finally, Section 2.2.4 and Section 2.2.5 respectively present important stability concepts and introduce the notion of contractive sets, which are both crucial ingredients for the development for LPV MPC algorithms.

2.2.1 Formal definition

Formally, a parameter-varying (PV) dynamical system can be defined as a quadruple (Tóth 2010)

$$\Sigma = (\mathbb{T}, \Theta, \mathbb{W}, \mathscr{B}) \tag{2.3}$$

where $\mathbb{T}$ is the time-axis, Θ is the scheduling set which is a bounded subset of an n_θ-dimensional space, $\mathbb{W}$ is the n_w-dimensional signal space, and $\mathscr{B} = (\mathbb{W} \times \Theta)^\mathbb{T}$ is the so-called behavior, i.e., the set of all possible maps $\mathbb{T} \to \mathbb{W} \times \Theta$.

If Σ is a system described in continuous time, then $\mathbb{T} = \mathbb{R}$. The case considered in this thesis, however, is that of discrete-time dynamical systems in which case $\mathbb{T} = \mathbb{N}$.

In (2.3), there is no immediate distinction as to which signals in $\mathbb{W}$ should be considered "inputs" or "outputs". If causal relationships between the signals in $\mathbb{W}$ exist, then these are specified in terms of the behavior $\mathscr{B}$.

This thesis is concerned with *linear* parameter-varying (LPV) systems:

Definition 2.5 (Tóth 2010, Definition 3.3)**.** *The parameter-varying system* Σ (2.3) *is a linear parameter-varying system, if the following conditions are satisfied:*

- $\mathbb{W}$ *is a vector space and for all possible signals* $\theta : \mathbb{T} \to \Theta$*, the set* $\mathscr{B}_\theta = \{w \in \mathbb{W}^\mathbb{T} \mid (w, \theta) \in \mathscr{B}\}$ *is a linear subspace of* $\mathbb{W}^\mathbb{T}$*;*
- $\mathbb{T}$ *is closed under addition;*
- *For all* $(w, \theta) \in \mathscr{B}$ *and for all* $\tau \in \mathbb{T}$*, there holds* $(t \mapsto (w(t+\tau), \theta(t+\tau))) \in \mathscr{B}$*.*

The following assumption is fundamental to LPV control.

Assumption 2.1. *The value* $\theta(k)$ *of the scheduling signal* $\theta : \mathbb{N} \to \Theta$ *can be measured for all time instants* $k \in \mathbb{N}$*.*

Definition 2.5 is rather abstract, and indeed does not provide much insight into how to actually *represent* an LPV system (2.3). The next section will therefore introduce the LPV-SS representation.

2.2.2 LPV state-space representations

A state-space representation of a discrete-time LPV system Σ, with a given input/state/output partitioning of $\mathbb{W}$, is described by the equations

$$\begin{aligned} x(k+1) &= A(\theta(k))x(k) + B(\theta(k))u(k); \;\; x(0) = x_0, \\ y(k) &= C(\theta(k))x(k) + D(\theta(k))u(k), \end{aligned} \tag{2.4}$$

where $x : \mathbb{N} \to \mathbb{R}^{n_\mathrm{x}}$ is the state vector, $u : \mathbb{N} \to \mathbb{R}^{n_\mathrm{u}}$ is the control input, and $y : \mathbb{N} \to \mathbb{R}^{n_\mathrm{y}}$ is the measured output. The signal $\theta : \mathbb{N} \to \Theta \subset \mathbb{R}^{n_\theta}$ is called the scheduling signal. The scheduling set Θ is assumed to be bounded, and the functions

$$(A, B, C, D) : \Theta \times \Theta \times \Theta \times \Theta \to \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{x}} \times \mathbb{R}^{n_\mathrm{x} \times n_\mathrm{u}} \times \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{x}} \times \mathbb{R}^{n_\mathrm{y} \times n_\mathrm{u}}$$

are required to be bounded on Θ. In general, (A, B, C, D) can be dependent on past values $\{\theta(k-1), \theta(k-2), \dots\}$ as in addition to just the current value $\theta(k)$. For simplicity, only LPV systems (2.3) that admit a state-space representation (2.4) with dependency on $\theta(k)$ alone are considered here. This case is called *static parameter dependence* in the LPV literature.

When compared to the general parameter-varying system (2.3), the state-space representation (2.4) has introduced a partitioning of the signal space $\mathbb{W}$ into a set of input, output, and state variables. Consequently, the behavior corresponding to an LPV system represented by (2.4) becomes

$$\mathscr{B}_\mathrm{SS} = \left\{ (u, y, x, \theta) \in (\mathbb{W} \times \Theta)^\mathbb{T} = (\mathbb{R}^{n_\mathrm{u}} \times \mathbb{R}^{n_\mathrm{y}} \times \mathbb{R}^{n_\mathrm{x}} \times \Theta)^\mathbb{T} \,\middle|\, (2.4) \text{ holds} \right\}. \tag{2.5}$$

A special, but very important case of (2.4) is the class of polytopic LPV state-space (LPV-PSS) representations. In fact, this is the type of representation that will be used throughout the majority of this thesis. An LPV-SS representation is called polytopic if its associated scheduling set is an n_θ-dimensional polytope, i.e.,

$$\Theta = \text{convh}\left\{\bar{\theta}^1, \bar{\theta}^2, \ldots, \bar{\theta}^{q_\theta}\right\} \subset \mathbb{R}^{n_\theta} \tag{2.6}$$

given in V-representation as the convex hull of q_θ vertices, and if the functions (A, B, C, D) take the affine form

$$\begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{bmatrix} = \begin{bmatrix} A_0 & B_0 \\ C_0 & D_0 \end{bmatrix} + \sum_{i=1}^{n_\theta} \begin{bmatrix} A_i & B_i \\ C_i & D_i \end{bmatrix} \theta_i \tag{2.7}$$

where $\{A_0, \ldots, D_{n_\theta}\}$ are matrices of conformable dimensions. For future reference, this concept is summarized in the following definition.

Definition 2.6. *An LPV-SS representation* (2.4) *is a polytopic LPV state-space (LPV-PSS) representation, if its associated scheduling set* Θ *is a polytope of the form* (2.6) *and if the functions* (A, B, C, D) *are of the affine form* (2.7).

An important property of LPV-PSS representations is the fact that the set of matrices

$$\mathcal{M} = \left\{ \begin{bmatrix} A(\theta) & B(\theta) \\ C(\theta) & D(\theta) \end{bmatrix} \middle| \theta \in \Theta \right\} \subset \mathbb{R}^{(n_\text{x}+n_\text{y})\times(n_\text{x}+n_\text{u})}$$

is itself a polytope with q_θ vertices (see (2.6)). This property makes it possible to formulate efficient computational procedures working on this class of LPV-SS representations.

Throughout the thesis, the following assumption is made.

Assumption 2.2. *The values of the state vector* $x(k)$ *can be measured for all time instants* $k \in \mathbb{N}$.

This assumption of full state feedback may appear restrictive. In practice, often only a measurement of the output $y(k)$ is available. However, the development of control methods utilizing full state feedback serves as an important stepping stone towards tackling the more difficult case of output feedback: compare, e.g., the separate design of state feedback controllers and observers in classical linear quadratic control theory (Anderson and Moore 1989).

In this thesis, when the tracking of an output reference is considered in Chapter 7, it is assumed that this can be handled by tracking a suitable reference for the state variable instead.

2.2.3 Constrained LPV representations

An LPV-SS representation (2.4) is called constrained, if for all time $k \in \mathbb{N}$, the state vector $x(k)$ and control input $u(k)$ are elements of the sets $\mathbb{X} \subset \mathbb{R}^{n_\text{x}}$ and $\mathbb{U} \subset \mathbb{R}^{n_\text{u}}$ respectively, as opposed to the full spaces $\mathbb{R}^{n_\text{x}}$ and $\mathbb{R}^{n_\text{u}}$. In other words, the state variable in (2.4) is now a signal $x : \mathbb{N} \to \mathbb{X}$ and the control input is a signal $u : \mathbb{N} \to \mathbb{U}$. For future reference, this is summarized in the following definition.

Definition 2.7. *An LPV-SS representation is* constrained*, if for all $k \in \mathbb{N}$, $x(k) \in \mathbb{X} \subset \mathbb{R}^{n_\mathrm{x}}$ and $u(k) \in \mathbb{U} \subset \mathbb{R}^{n_\mathrm{u}}$ where $(\mathbb{X}, \mathbb{U})$ are called the constraint sets. These constraint sets are PC-sets (i.e., they are convex, compact, and have a non-empty interior that contains the origin).*

Naturally, it is possible to speak of constrained LPV-PSS representations as well: indeed, this is the class of representations that will see the most use throughout this thesis.

Whenever numerical implementation is concerned, it is additionally useful to assume that the sets $(\mathbb{X}, \mathbb{U})$ are polytopes. In those cases, it is assumed that the constraint sets can be described by an H-representation:

$$\begin{aligned} \mathbb{X} &= \{x \in \mathbb{R}^{n_\mathrm{x}} \mid H_\mathrm{x} x \leq 1\} \text{ for some } H_\mathrm{x} \in \mathbb{R}^{r_\mathrm{x} \times n_\mathrm{x}}, \\ \mathbb{U} &= \{u \in \mathbb{R}^{n_\mathrm{u}} \mid H_\mathrm{u} u \leq 1\} \text{ for some } H_\mathrm{u} \in \mathbb{R}^{r_\mathrm{u} \times n_\mathrm{u}}. \end{aligned} \tag{2.8}$$

An advantage of using the H-representation for constraint sets is the fact that this allows the efficient representation of "box"-shaped constraints

$$\begin{aligned} \mathbb{X} &= \left\{x \in \mathbb{R}^{n_\mathrm{x}} \,\middle|\, \underline{x} \leq x \leq \overline{x}\right\}, \\ \mathbb{U} &= \left\{u \in \mathbb{R}^{n_\mathrm{u}} \,\middle|\, \underline{u} \leq u \leq \overline{u}\right\}, \end{aligned}$$

where $(\underline{x}, \underline{u})$ and $(\overline{x}, \overline{u})$ are vectors of appropriate dimensions containing respectively lower and upper bounds on the state vector and control input.

Sometimes it is useful to impose constraints on the output $y \in \mathbb{R}^{n_\mathrm{y}}$ as well in terms of an output constraint set $y \in \mathbb{Y}$. As in the state and input constraint cases, it is assumed that $\mathbb{Y}$ is a PC-set. Because all results in this thesis consider full state feedback (i.e., the full state vector x can be measured and used for feedback), it is then assumed that such output constraints can be translated to a suitable state constraint set as

$$\mathbb{X} = \{x \in \mathbb{R}^{n_\mathrm{x}} \mid \forall (\theta, u) \in \Theta \times \mathbb{U} : C(\theta)x + D(\theta)u \in \mathbb{Y}\}.$$

2.2.4 Stability of (L)PV-SS representations

This section reviews the stability of the flow characterized by an autonomous constrained PV state-space (PV-SS) representation

$$x(k+1) = \Phi\left(x(k), \theta(k), k\right); \; x(0) = x_0, \tag{2.9}$$

where $\Phi : \mathbb{X} \times \Theta \times \mathbb{N} \to \mathbb{X}$ is a—possibly non-linear and time-dependent—state transition map. The reason for considering non-linear and time-dependent representations in this section is that an MPC typically corresponds to a non-linear control law. Thus, an LPV-PSS representation that is connected in closed-loop with an MPC can be represented in the general form (2.9), and the results from this section can then be used to assess the stability of the closed-loop. The only assumption that is taken on $\Phi(\cdot, \cdot, \cdot)$ is the following.

Assumption 2.3. *For all $(\theta, k) \in \Theta \times \mathbb{N}$, the representation* (2.9) *satisfies $\Phi(0, \theta, k) = 0$. This implies that the origin is an equilibrium of* (2.9).

If the origin is an asymptotically stable equilibrium of (2.9), then $x(k) \to 0$ as $k \to \infty$ for all possible signals $\theta : \mathbb{N} \to \Theta$. Because the representation (2.9) is subject to constraints, this convergence can typically not be attained for all possible initial conditions $x_0 \in \mathbb{X}$. Therefore, the following notion of *regional* asymptotic stability is considered:

Definition 2.8. *Let $\mathbf{x}(k|\theta, x_0)$ denote the solution $x(k)$ of* (2.9)*, for a given scheduling signal $\theta : \mathbb{N} \to \Theta$ and for the initial state $x(0) = x_0$. The origin is said to be a regionally asymptotically stable equilibrium of* (2.9)*, if there exists a $\mathcal{KL}$-function β and a PC-set $\mathcal{X} \subseteq \mathbb{X} \subset \mathbb{R}^{n_x}$ such that*

$$\|\mathbf{x}(k|\theta, x_0)\| \leq \beta(\|x_0\|, k)$$

for all possible scheduling signals $\theta : \mathbb{N} \to \Theta$, for all $x_0 \in \mathcal{X}$, and for all $k \in \mathbb{N}$.

A similar notion of regional input-to-state stability has been considered in (Magni et al. 2006). The origin is called a globally asymptotically stable equilibrium if Definition 2.8 holds for $\mathcal{X} = \mathbb{R}^{n_x}$. However, if the state $x(k)$ is constrained to belong to a compact set $\mathbb{X}$, asymptotic stability can only be obtained in a regional sense. Definition 2.8 is usually not verified directly. Instead, use is made of regional Lyapunov functions which are defined as follows.

Definition 2.9. *A function $V : \mathbb{R}^{n_x} \times \mathbb{N} \to \mathbb{R}$ is a (regional, time-varying) Lyapunov function on an invariant PC-set $\mathcal{X} \subseteq \mathbb{X} \subset \mathbb{R}^{n_x}$ for* (2.9) *if*

(i) There exist $\mathcal{K}_\infty$-functions $\underline{v}, \overline{v}$ such that for all $(x, k) \in \mathcal{X} \times \mathbb{N}$: $\underline{v}(\|x\|) \leq V(x, k) \leq \overline{v}(\|x\|)$;

(ii) There exists a $\mathcal{K}$-function δ such that for all $(x, \theta, k) \in \mathcal{X} \times \Theta \times \mathbb{N}$: $V(\Phi(x, \theta, k), k+1) \leq V(x, k) - \delta(\|x\|)$.

With Definition 2.9, to verify if the origin is a regionally asymptotically stable equilibrium of (2.9), the following lemma can be used:

Lemma 2.2 ((Aeyels and Peuteman 1998; Jiang and Y. Wang 2002))**.** *If there exists a regional time-varying Lyapunov function satisfying Definition 2.9, then the origin is a regionally asymptotically stable equilibrium of* (2.9) *in the sense of Definition 2.8.*

Definition 2.9 and Lemma 2.2 are based on (Aeyels and Peuteman 1998) which presents results concerning the stability of non-linear time-varying difference equations, of which representation (2.9) can be considered to be a special case. Similar results for general non-linear systems with disturbances are found in (Jiang and Y. Wang 2002). Throughout this thesis, Lemma 2.2 will be employed to establish closed-loop stability of the developed MPC approaches. A more detailed overview of stability concepts for discrete-time non-linear systems can also be found in (Grüne and Pannek 2011, Chapter 2.3): with respect to that book, the notions in this section have been modified to include the scheduling variable θ.

2.2.5 Reachable and contractive sets

This section introduces the basic concepts of reachable and contractive sets, which are crucial tools in the construction of stabilizing MPC algorithms for LPV systems.

The one-step backward reachable set with respect to a state set $X \subseteq \mathbb{X}$ and to the dynamics (2.4), is the set of states that can be steered to X in one step for all possible values of the scheduling variable. It is defined formally as follows.

Definition 2.10. *For a constrained LPV-SS representation, the backward reachable set is the map* $\mathcal{Q}(\cdot,\cdot) : 2^{\mathbb{X}} \times 2^{\Theta} \to 2^{\mathbb{X}}$ *defined as*

$$\mathcal{Q}(X,\Theta) = \{x \in \mathbb{X} \,|\, \forall\theta \in \Theta : \exists u(x,\theta) \in \mathbb{U} : A(\theta)x + B(\theta)u(x,\theta) \in X\}\,. \tag{2.10}$$

The determination of a backwards reachable set requires computing a projection (2.10). With the general definition of (2.10), this is challenging. To enable efficient computation, at least one of the following two conditions must be satisfied:

- The input matrix of the involved LPV-PSS representation is constant, i.e., $B(\theta) = B$;
- The control input does not depend on θ, i.e., (2.10) is replaced by

$$\mathcal{Q}(X,\Theta) = \{x \in \mathbb{X} \,|\, \exists u(x) \in \mathbb{U} : \forall\theta \in \Theta : A(\theta)x + B(\theta)u(x) \in X\}\,. \tag{2.11}$$

If one of the above conditions is satisfied and if the target set X is a polytope, then the projection can be computed and the set $\mathcal{Q}(X,\Theta)$ is a polytope as well. One algorithm that can be used for this is Fourier-Motzkin elimination (Dantzig and Eaves 1973).

Conversely, the one-step forward reachable set—or *image*—of a set X for the dynamics (2.4) under a given controller, and for a corresponding scheduling set, is defined as follows.

Definition 2.11. *The controlled image of a set for a constrained LPV-SS representation with a given controller* $K : X \times \Theta \to \mathbb{U}$ *is the map* $\mathcal{I}(\cdot,\cdot|K) : 2^X \times 2^{\Theta} \to 2^{\mathbb{R}^{n_x}}$ *defined by*

$$\mathcal{I}(X,\Theta|K) = \big\{A(\theta)x + B(\theta)K(x,\theta) \mid x \in X, \theta \in \Theta\big\}.$$

In contrast to the backward reachable set, the image $\mathcal{I}(X,\Theta|K)$ for a controlled LPV-PSS representation is in general non-convex even in the simplest possible case that $X \subseteq \mathbb{X}$ is a polytope and $K(\cdot,\cdot)$ is linear (Blanchini and Miani 2015, Section 6.1.2). Nonetheless, the convex hull of forward reachable sets can be propagated recursively (Blanchini and Miani 2015, Proposition 6.5). The image satisfies the following "subset inclusion" property:

Lemma 2.3. *Let* $K : X \times \Theta \to \mathbb{U}$*. For any subset* $X' \times \Theta' \subseteq X \times \Theta$*, it holds* $\mathcal{I}(X',\Theta'|K) \subseteq \mathcal{I}(X,\Theta|K)$*.*

Now, the concepts of invariant and contractive sets are introduced. For this purpose, it is useful to consider control functions $K(\cdot,\cdot)$ which satisfy some additional properties. These properties are here summarized under the name of continous and positively homogeneous of degree one ($\mathcal{CH}_1$):

Definition 2.12. *A controller* $K : X \times \Theta \to \mathbb{U}$ *is called* $\mathcal{CH}_1$ *if it is (i) continuous, and (ii) positively homogeneous of degree one in the sense that* $\forall\alpha \in \mathbb{R}_+ :\ K(\alpha x,\theta) = \alpha K(x,\theta)$*.*

The next two definitions formally introduce the concepts of (controlled) invariant and contractive sets.

Definition 2.13. *A PC-set $X \subseteq \mathbb{X}$ is called controlled λ-contractive (respectively, controlled invariant) for an LPV-SS representation* (2.4)*, if there exists a local $\mathcal{CH}_1$ controller K : $X \times \Theta \to \mathbb{U}$ such that $\lambda = \inf\{\mu \geq 0 \,|\, \mathcal{I}(X, \Theta|K) \subseteq \mu X\} < 1$ (respectively, $= 1$).*

Definition 2.14. *A PC-set $X \subseteq \mathbb{X}$ is called λ-contractive (resp., invariant) with respect to a $\mathcal{CH}_1$ controller $K : X \times \Theta \to \mathbb{U}$, if it satisfies Definition 2.13 for the given (as opposed to "some") $K(\cdot, \cdot)$.*

Controlled invariant and contractive sets are used as tools to guarantee the stability of predictive controllers. It should be understood that the limitation to $\mathcal{CH}_1$-controllers in the above definitions is not restrictive. Suppose that for a set $X \subset \mathbb{X}$, an initial state $x \in \mathbb{X}$, and a scheduling value $\theta \in \Theta$, there exists a control action $u \in \mathbb{U}$ such that

$$A(\theta)x + B(\theta)u \in X.$$

Then it is easy to see that, for the scaled initial state αx, the control action u can be scaled by the same factor to obtain

$$A(\theta)\alpha x + B(\theta)\alpha u = \alpha \left(A(\theta)x + B(\theta)u\right) \in \alpha X.$$

Furthermore, if $\alpha \in [0, 1]$, then $\alpha u \in \mathbb{U}$ due to the PC-property of $\mathbb{U}$ (Definition 2.7).

Verifying if a given PC-set $X \subseteq \mathbb{X}$ is controlled λ-contractive in the sense of Definition 2.13 can be done by checking if there exists an associated $\mathcal{CH}_1$ controller such that $\forall(x, \theta) \in X \times \Theta :$ $A(\theta)x + B(\theta)K(x, \theta) \in \lambda X$. In the case of LPV-PSS representations, the existence of such a controller on a polytopic set is equivalent to the existence of individual control actions on the vertices of that set, as summarized in the following lemma.

Lemma 2.4. *Let $X = \operatorname{convh}\left\{\bar{x}^1, \ldots, \bar{x}^{q_x}\right\} \subseteq \mathbb{X}$ be a PC-set. The set X is controlled λ-contractive for a given constrained LPV-PSS representation with parameter-independent input matrix $B(\theta) = B$, if and only if $\forall(i, j) \in [1..q_x] \times [1..q_\theta] :\ \exists u \in \mathbb{U} :\ A(\bar{\theta}^j)\bar{x}^i + Bu \in \lambda X$.*

Proof. This fact is well-known in the literature, but for easy further reference, a proof based on some simple convexity arguments is provided here. It is sufficient to consider two pairs of arbitrary vertices $(\bar{x}^a, \bar{x}^b)$ and $(\bar{\theta}^c, \bar{\theta}^d)$. Suppose that $\forall(i, j) \in \{a, b\} \times \{c, d\} :\ \exists \bar{u}^{(i,j)}$: $A(\bar{\theta}^j)\bar{x}^i + Bu \in \lambda X$. Let $x = \alpha\bar{x}^a + (1 - \alpha)\bar{x}^b$ and $\theta = \beta\bar{\theta}^c + (1 - \beta)\bar{\theta}^d$ where $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ are arbitrary convex multipliers. Then, the convex combination of control inputs $u = \alpha(\beta\bar{u}^{(a,c)} + (1 - \beta)\bar{u}^{(a,d)}) + (1 - \alpha)(\beta\bar{u}^{(b,c)} + (1 - \beta)\bar{u}^{(b,d)})$ is such that $A(\theta)x + Bu \in \lambda X$:

$$\begin{aligned}
A(\theta)x + Bu = &\left(\beta A(\bar{\theta}^c) + (1 - \beta)A(\bar{\theta}^d)\right)\left(\alpha\bar{x}^a + (1 - \alpha)\bar{x}^b\right) \\
&+ B\left(\alpha\left(\beta\bar{u}^{(a,c)} + (1 - \beta)\bar{u}^{(a,d)}\right) + (1 - \alpha)\left(\beta\bar{u}^{(b,c)} + (1 - \beta)\bar{u}^{(b,d)}\right)\right) \\
= &\ \alpha\left(\beta\left(A(\bar{\theta}^c)\bar{x}^a + B\bar{u}^{(a,c)}\right) + (1 - \beta)\left(A(\bar{\theta}^d)\bar{x}^a + B\bar{u}^{(a,d)}\right)\right) \\
&+ (1 - \alpha)\left(\beta\left(A(\bar{\theta}^c)\bar{x}^b + B\bar{u}^{(b,c)}\right) + (1 - \beta)\left(A(\bar{\theta}^d)\bar{x}^b + B\bar{u}^{(b,d)}\right)\right) \\
\in &\ \lambda X,
\end{aligned}$$

where the last inclusion follows because the set X is convex and because it was known that $A(\bar{\theta}^j)\bar{x}^i + B\bar{u}^{(i,j)} \in \lambda X$, $(i, j) \in \{a, b\} \times \{c, d\}$. □

Lemma 2.4 establishes that determining if a given PC-set $X \subset \mathbb{X}$ is λ-contractive is equivalent to solving a set of finitely many linear inequalities. This can be done in a computationally efficient manner using linear programming.

Some variations on Lemma 2.4 with similar proofs are possible. For instance, when the considered LPV-PSS representation does not have a constant B-matrix, then the set X is controlled λ-contractive if for all $i \in [1..q_\mathrm{x}]$ there exists a $u \in \mathbb{U}$ such that for all $j \in [1..q_\theta]$ there holds $A(\bar{\theta}^j)\bar{x}^i + B(\bar{\theta}^j)u \in \lambda X$.

One standard approach for computing (controlled) contractive sets for LPV-PSS representations, is the backwards iterative algorithm of (Blanchini and Miani 2015, Chapter 5)[2]. This approach is summarized for future reference in Algorithm 2.1.

Algorithm 2.1 Backwards iterative computation of controlled λ-contractive sets.

Require: Desired contraction $\lambda \in [0, 1]$, tolerance $\epsilon > 0$ such that $\lambda + \epsilon \leq 1$, iteration limit $k_{\max} > 0$

1: $k \leftarrow 0$
2: Choose arbitrary PC-set $X_0 \subseteq \mathbb{X}$
3: **loop**
4: $\quad X_{k+1} = X_k \cap \mathcal{Q}(\lambda X_k, \Theta)$ ▷ $\mathcal{Q}(\cdot, \cdot)$ as in Definition 2.10
5: $\quad$ **if** $X_k \subseteq (\lambda + \epsilon)X_{k+1}$ **then**
6: $\quad\quad$ **stop** successfully
7: $\quad$ **else if** $X_k = \emptyset$ **then**
8: $\quad\quad$ **stop** unsuccessfully
9: $\quad$ **else if** $k \geq k_{\max}$ **then**
10: $\quad\quad$ **stop** indeterminately
11: $\quad$ **end if**
12: $\quad k \leftarrow k + 1$
13: **end loop**

This iterative approach is implemented in freely available software such as (Kvasnica et al. 2015; Miani and Savorgnan 2005). Because the computation of backwards reachable sets is a key step in the iterative algorithm, the computational comments below Definition 2.10 apply to Line 4 of Algorithm 2.1 as well.

2.3 Embeddings of non-linear systems

This section reviews the principles of LPV embeddings of non-linear systems. Roughly speaking, when constructing an embedding, non-linearities are "covered" by introducing

[2](Blanchini and Miani 2015) considers a robust pre-image map of the form (2.11), but the same algorithm works with the "scheduling" pre-image map (2.10) provided that the B-matrix of the LPV-PSS representation is constant, see (Miani and Savorgnan 2005).

scheduling variables. In this way, it becomes possible to use computationally efficient LPV control design approaches to find controllers for non-linear systems.

2.3.1 The embedding principle

Consider a general system represented in terms of a state-space (SS) representation

$$x(k+1) = f\left(x(k), u(k)\right);\ x(0) = x_0, \tag{2.12}$$

where $x : \mathbb{N} \to \mathbb{R}^{n_\mathrm{x}}$ and $u : \mathbb{N} \to \mathbb{R}^{n_\mathrm{u}}$. In the specific context of SS representations, the concept of an LPV embedding can be defined as follows.

Definition 2.15. *Let $X \subseteq \mathbb{R}^{n_\mathrm{x}}$ be an open, bounded, and invariant set corresponding to an operating region of the non-linear system* (2.12). *Define $U = \{u \in \mathbb{R}^{n_\mathrm{u}} \mid \exists x \in X : f(x,u) \in X\}$. An LPV-SS representation* (2.4) *is an embedding of* (2.12) *on X, if for all $(x,u) \in X \times U$ there exists a value $\theta \in \Theta$ such that $f(x,u) = A(\theta)x + B(\theta)u$ with $\Theta \subseteq \mathbb{R}^{n_\theta}$ being a suitably defined scheduling set.*

From the above definition, it can be concluded that state trajectories—within the operating domain X—of the non-linear system (2.12) are equal to state trajectories of the embedding for *one particular* trajectory of the scheduling signal. When a controller is designed for an LPV system, it has to guarantee closed-loop stability with respect to *all possible* scheduling trajectories inside of the scheduling set. Therefore, if a stabilizing controller is found for an LPV embedding of a non-linear system, then the same controller stabilizes the original non-linear system. Constrained and polytopic LPV embeddings are defined analogously to Definitions 2.7 and 2.6, respectively.

In terms of behaviors, the embedding principle for state-space representations can be formalized as follows. Corresponding to (2.12), a signal space $W = U \times X$ can be defined according to the input/state partition $w = (u, x)$, with (U, X) as in Definition 2.15. Let an LPV-SS representation (2.4) be given, and let

$$\mathcal{B}_{\mathrm{LPV}} = \left\{(u, x, \theta) \in (W \times \Theta)^{\mathbb{T}} = (U \times X \times \Theta)^{\mathbb{T}} \,\middle|\, (2.4) \text{ holds}\right\}.$$

be its corresponding behavior on W (note that now, the output y is considered equal to x for simplicity). Define a projected behavior as

$$\mathcal{B}_{\mathrm{LPV}}^{\perp} = \left\{w \in W^{\mathrm{T}} \,\middle|\, \exists \theta \in \Theta^{\mathbb{T}} :\ (w, \theta) \in \mathcal{B}_{\mathrm{LPV}}\right\},$$

and consider the non-linear behavior

$$\mathcal{B}_{\mathrm{NL}} = \left\{w \in W^{\mathbb{T}} \,\middle|\, (2.12) \text{ holds}\right\}.$$

Then, the LPV-SS representation is an embedding of the non-linear system (2.12) if the behavioral inclusion

$$\mathcal{B}_{\mathrm{NL}} \subseteq \mathcal{B}_{\mathrm{LPV}}^{\perp} \tag{2.13}$$

is satisfied.

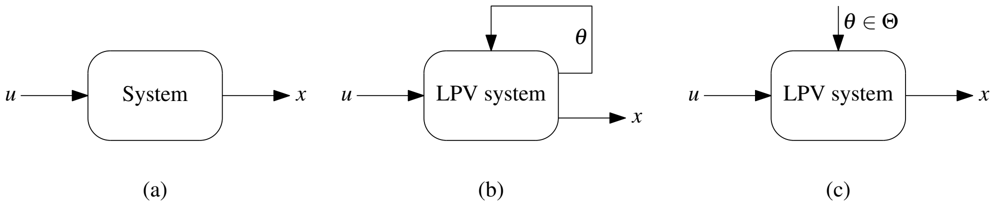


Figure 2.1: The construction of an LPV embedding. (a) The original non-linear system. (b) The non-linear system equivalently represented. (c) The scheduling variable is "disconnected" and becomes a free signal: the LPV embedding is obtained. The behaviors of these three systems are related as $\mathcal{B}_{\mathrm{NL}} = \mathcal{B}_{\mathrm{a}} = \mathcal{B}_{\mathrm{b}}^{\perp} \subseteq \mathcal{B}_{\mathrm{c}}^{\perp} = \mathcal{B}_{\mathrm{LPV}}^{\perp}$.

The idea of an LPV embedding is reminiscent of that of a linear difference inclusion (LDI). Such inclusions (as well as their continuous-time counterpart, the linear differential inclusion) are well-known tools in the stability of analysis of non-linear systems or systems with time delay (Boyd et al. 1994; Gielen et al. 2010). The principal difference between an LPV embedding and a difference inclusion, is the fact that in an embedding the scheduling variable is assumed to be measurable. This measurement can be exploited by a controller, as was done, e.g., in (Scherer et al. 1997) for flight control design.

In this approach, there is a classical trade-off between (computational) simplicity and achievable control performance. On the one hand, using an LPV embedding enables the use of well-developed efficient linear design methods to obtain controllers for non-linear systems. On the other hand, the requirement that this linear controller has to be stabilizing with respect to all possible scheduling trajectories can introduce conservatism.

2.3.2 Construction of embeddings

The general process of constructing an LPV embedding of a non-linear system is depicted in Figure 2.1. Starting with the original non-linear system, scheduling variables $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ are introduced such that the non-linear system is equivalently represented in an LPV form as

$$f(x,u) = A(\theta)x + B(\theta)u \tag{2.14}$$

where the dependence of the matrix functions $(A, B) : \Theta \times \Theta \to \mathbb{R}^{n_{\mathrm{x}} \times n_{\mathrm{x}}} \times \mathbb{R}^{n_{\mathrm{x}} \times n_{\mathrm{u}}}$ can be non-linear. Typically, the introduced scheduling variables are related to the state and input signals, i.e., there exists a map $(x,u) \mapsto \theta$ (Figure 2.1.(b)). Because an embedding is constructed on a bounded operating region $X \subseteq \mathbb{R}^{n_{\mathrm{x}}}$, it is possible to bound the values of θ inside of a bounded scheduling set $\Theta \subset \mathbb{R}^{n_\theta}$. Then, to construct the embedding, the map $(x,u) \mapsto \theta$ is discarded and only the inclusion $\theta(k) \in \Theta$ is assumed (Figure 2.1.(c)).

Example 2.4. *Consider the simple non-linear system $f(x,u) = (1+x)x+u$ on the open interval $X = (-1,1)$. Introduce a scheduling variable $\theta = x$, leading to the equivalent representation $f(x,u) = (1+\theta)x+u$. Now, if the relation $\theta = x$ is discarded and it is only assumed that $\theta \in \Theta$, an LPV embedding is obtained. In this example, a suitable choice for the scheduling set Θ is $\Theta = X = (-1,1)$.*

If it is necessary to obtain a polytopic LPV embedding, the scheduling variable θ must be selected such that the dependence of the matrix functions (A, B) in (2.14) on θ is affine and the set Θ must be designed to be a polytope. This can usually be achieved by an appropriate change of variables, or by introducing new auxiliary scheduling signals.

In the literature, a number of methods are available to generate embeddings for (certain classes of) non-linear systems. These can be classified into three important categories: automated transformation approaches, substitution-based approaches, and local approaches. A selection of methods from these categories is now reviewed.

In automated transformation approaches, the idea is to algorithmically explore the different ways in which the scheduling variables can be selected so as to obtain an equivalent representation of the non-linear system. In (Kwiatkowski et al. 2006) this is accomplished by factorizing the non-linear equations in a standard form. Then, a selection is made of the factors that will correspond to scheduling variables in the embedding. A similar strategy, based on a decision tree that collects all possible factorizations, is presented in (Tóth 2010, Section 7.4). Out of all the possibilities described in terms of this tree, it is then possible to select the "best" embedding, for instance the one with the least number of scheduling variables. In (Donida et al. 2009), a method is developed to transform system models described in the Modelica modeling language into a linear parameter-varying linear-fractional representation (LPV-LFR).

A framework for modeling non-linear mechanical systems in terms of an LPV-LFR is presented in (Hoffmann and Werner 2014). The work (Hoffmann and Werner 2015a) proposes an automated procedure, based on principal component analysis (PCA), to select an efficient parameterization of the LFR. In (Hoffmann and Werner 2015b), the method is applied to the LPV-LFR modeling of a control moment gyroscope.

Substitution-based approaches involve the elimination of non-linear terms through the introduction of "virtual" scheduling signals to cover the non-linearities. Often, this is a manual process. It was applied in Example 2.4 where the substitution $\theta = x$ was made. In (Shamma and Cloutier 1993), an LPV embedding is obtained by applying a state transformation. The state vector of the non-linear system has to be decomposed into a part that becomes the scheduling variable and a part that becomes the state vector of the embedding. A more systematic substitution-approach based on feedback linearization of control-affine non-linear systems is presented in (Abbas et al. 2014). The method therein yields LPV-SS embeddings in the observability canonical form, where the scheduling variables correspond to the time derivatives of the inputs and outputs of the system. The construction of embeddings trough velocity-based linearization (Leith and Leithead 1998) can also be interpreted as a substitution approach, because this introduces a virtual scheduling variable that corresponds to the state and input variables of the system (Tóth 2010, Section 7.3.4.3).

In (Petersson and Löfberg 2009), a method is presented to construct an LPV model of a system that is described by multiple local models. These local models can, for instance, represent the linearizations of a non-linear system around a collection of operating points. The principal idea is to minimize the H_2-norm of the error systems defined as the difference between the LPV models evaluated at frozen scheduling values and the local models. A similar approach, but based on minimization of the H_∞-norm, is given in (Vízer and Mercère 2014). This approach was applied in (Vízer et al. 2015) for the LPV modeling of a surgical robot based on local identification experiments. It should be emphasized that local methods are not a-priori guaranteed to produce a “valid” embedding that satisfies (2.13).

Summarizing, a number of approaches for the construction for embeddings is available in the literature. However, it is difficult to say which method yields the “best” embedding for a given non-linear system and for a given control design purpose. Therefore, constructing an embedding of a given non-linear system remains to a large extent a manual process involving various choices and trade-offs.

2.4 Model predictive control

This section introduces the main ideas of model predictive control (MPC) in the nominal case. The term “nominal” means that there is no uncertainty in the model of the system that is to be controlled. An understanding of these concepts is useful before proceeding to the more involved case of LPV MPC. In particular, the (well-known) nominal stability result presented here can be seen as a precursor to the LPV MPC stability results that are developed as part of this thesis.

Consider a constrained system of the form

$$x(k+1) = f\left(x(k), u(k)\right); \; x(0) = x_0, \tag{2.15}$$

where $x : \mathbb{N} \to \mathbb{X} \subseteq \mathbb{R}^{n_\mathrm{x}}$ is the state variable, and $u : \mathbb{N} \to \mathbb{U} \subseteq \mathbb{R}^{n_\mathrm{u}}$ is the control input. In line with Definition 2.7, it is assumed that the constraint sets $(\mathbb{X}, \mathbb{U})$ are PC. The problem of interest is to design a controller $K_\mathrm{mpc} : \mathbb{X} \to \mathbb{U}$ such that the origin is a regionally asymptotically stable equilibrium (Definition 2.8) of the closed-loop system

$$x(k+1) = f\left(x(k), K_\mathrm{mpc}(x(k))\right); \; x(0) = x_0. \tag{2.16}$$

Let

$$\mathbf{u}_{N-1} = \{u_0, \ldots, u_{N-1}\}, \; \mathbf{x}_N = \{x_0, \ldots, x_N\}$$

denote sequences of control inputs and state variables, respectively. In MPC, at each sampling time instant, the control input $u(k) = K_\mathrm{mpc}(x(k))$ is determined by solving the optimization problem

$$\begin{aligned} V_N(x) = \min_{\mathbf{u}_{N-1}, \mathbf{x}_N} \quad & \sum_{i=0}^{N-1} \ell\left(x_i, u_i\right) + F\left(x_N\right) \\ \text{subject to} \quad & x_0 = x, \\ & \forall i \in [0..N-1] : \; x_{i+1} = f(x_i, u_i), \\ & \forall i \in [0..N-1] : \; u_i \in \mathbb{U}, \\ & \forall i \in [0..N-1] : \; x_i \in \mathbb{X}, \\ & x_N \in X_\mathrm{f}, \end{aligned} \tag{2.17}$$

where

- $N \in [1..\infty)$ is the desired prediction horizon;

- $\ell : \mathbb{X} \times \mathbb{U} \to \mathbb{R}_+$ is the stage cost, which specifies a user-defined performance objective;
- $X_\mathrm{f} \subseteq \mathbb{X}$ is the terminal set;
- $F : X_\mathrm{f} \to \mathbb{R}_+$ is the terminal cost, which together with the terminal set has to be designed appropriately in order to achieve closed-loop stability.

The function $V_N(\cdot) : \mathbb{X} \to \mathbb{R}_+$ in (2.17) is called the *value function*. Now, the controller $K_\mathrm{mpc}(\cdot)$ can be defined as

$$K_\mathrm{mpc}(x) = u_0^\star, \text{ where } \mathbf{u}_{N-1}^\star = \left\{u_0^\star, \ldots, u_{N-1}^\star\right\} \text{ minimizes (2.17) for the initial state } x. \quad (2.18)$$

It is possible to design the terminal set and cost such that the origin is a regionally asymptotically stable equilibrium of (2.16). The following theorem summarizes the (sufficient) conditions that X_f and $F(\cdot)$ must satisfy.

Theorem 2.1 (See, e.g., (Rawlings and Mayne 2009))**.** *Let $K_\mathrm{mpc}(\cdot)$ be defined as in* (2.18)*. The origin is a regionally asymptotically stable equilibrium of* (2.16)*, if the following three conditions are satisfied:*

(i) The terminal set X_f is contained inside of the constraints ($X_\mathrm{f} \subseteq \mathbb{X}$), is closed, and contains the origin ($0 \in X_\mathrm{f}$).

(ii) There exists a local controller $K_\mathrm{f} : X_\mathrm{f} \to \mathbb{U}$ such that X_f is invariant with respect to K_f, i.e., $\forall x \in X_\mathrm{f} : f(x, K_\mathrm{f}(x)) \in X_\mathrm{f}$.

(iii) The terminal cost $F : X_\mathrm{f} \to \mathbb{R}_+$ is a regional Lyapunov function which satisfies $\forall x \in X_\mathrm{f} : F(f(x, K_\mathrm{f}(x))) - F(x) \leq -\ell(x, K_\mathrm{f}(x))$.

Proof. The proof sketch given here is meant to illustrate the main ideas. The motivation for this is that the concepts introduced in this proof for the nominal case, will be used later in a modified form to prove feasibility and stability for the LPV case as well. Full details on the stability properties of nominal MPC can be found in, e.g., (Grüne and Pannek 2011; Mayne et al. 2000; Rawlings and Mayne 2009).

Suppose that at an arbitrary initial time instant k, the measured initial state is x, and $(\mathbf{u}_{N-1}, \mathbf{x}_N)$ is a solution to (2.17). Recall that $\mathbf{u}_{N-1} = \{u_0, u_1, \ldots, u_{N-1}\}$ and $\mathbf{x}_N = \{x_0, x_1, \ldots, x_N\}$, where $x_0 = x$. By construction of the constraints, $x_N \in X_\mathrm{f}$. After applying the control input u_0 to the system, at the next time instant $k+1$ the state of the system is $x^+ = f(x, u_0) = x_1$. A feasible sequence of control inputs $\mathbf{u}_{N-2}^+$ that steers the state from x_1 to $x_N \in X_\mathrm{f}$ can be constructed by shifting and truncating the sequence $\mathbf{u}_{N-1}$ that was found at time k, i.e., $\mathbf{u}_{N-2}^+ = \{u_1, \ldots, u_{N-1}\}$. Furthermore, by Property (ii) there exists a control input u_{N-1}^+ that is such that $f(x_N, u_\mathrm{f}^+) \in X_\mathrm{f}$, namely $u_{N-1}^+ = K_\mathrm{f}(x_N)$. Thus at least one feasible input sequence that gives a solution to (2.17) at time $k+1$ exists and it is explicitly given as $\mathbf{u}_{N-1}^+ = \{u_1, \ldots, u_{N-1}, K_\mathrm{f}(x_N)\}$; the corresponding state sequence is $\mathbf{x}_N^+ = \{x_1, \ldots, x_N, f(x_N, K_\mathrm{f}(x_N))\}$. This property is called recursive feasibility.

Define $J_N(\mathbf{u}, \mathbf{x}) = \sum_{i=0}^{N-1} \ell(x_i, u_i) + F(x_N)$. Because $(\mathbf{u}_{N-1}, \mathbf{x}_N)$ was a solution of (2.17) for the initial state x, $J_N(\mathbf{u}_{N-1}, \mathbf{x}_N) = V(x)$. As the explicitly constructed sequences

$(\mathbf{u}_{N-1}^+, \mathbf{x}_N^+)$ are feasible, but not necessarily optimal at time $k+1$, it is known that $V(x^+) = V(x_1) \leq J_N(\mathbf{u}_{N-1}^+, \mathbf{x}_N^+)$. With some algebra, it is then possible to show that $V(x^+) - V(x) \leq J_N(\mathbf{u}_{N-1}^+, \mathbf{x}_N^+) - J_N(\mathbf{u}_{N-1}, \mathbf{x}_N) \leq -\ell(x, u_0)$. Provided that some technical conditions—namely, the existence of $\mathcal{K}_\infty$-bounds on $V(\cdot)$ and $\ell(\cdot,\cdot)$—are satisfied, this relation implies that $V(\cdot)$ is a regional Lyapunov function. The set of states on which this regional Lyapunov condition holds is equal to the set of states for which the problem (2.17) admits a feasible solution. Therefore, the origin is a regionally asymptotically stable equilibrium of the closed-loop system (2.16). □

In case the system (2.15) is linear time-invariant (LTI)—i.e., $x(k+1) = Ax(k) + Bu(k)$—the stage cost is typically selected to be quadratic as

$$\ell(x, u) = x^\top Q x + u^\top R u$$

where $Q \geq 0$ and $R \geq 0$ are the tuning parameters. Then, the local controller required in Theorem 2.1 can be chosen to be linear, i.e., $K_\mathrm{f}(x) = Kx$ such that there exists a matrix $P \succ 0$ satisfying

$$(A+BK)^\top P (A+BK) - P + Q + K^\top R K \preceq 0. \tag{2.19}$$

It can then be verified that $F(x) = x^\top P x$ is a terminal cost satisfying the conditions of Theorem 2.1. Note that (2.19) is satisfied when K is designed to be LQ-optimal with respect to the weights (Q, R). Finally, the terminal set $X_\mathrm{f} \subseteq \mathbb{X}$ can be computed as an invariant set for the closed-loop system $x(k+1) = (A+BK)x(k)$. Such a set can be computed, e.g., by using Algorithm 2.1.

Chapter 3

Anticipative tube-based MPC for LPV systems

The purpose of this chapter is to introduce a uniform framework for the control approaches that are described in this thesis. This framework includes the problem setting that is adopted throughout most of the thesis and a flexible description of future uncertain scheduling trajectories. The latter leads to the notion of “anticipative” control and is an answer to Research Question Q_1. Based on this, it is concluded that a tube-based model predictive control (MPC) approach has the potential to provide a fitting solution to the remaining research questions. As the next part of the framework, the foundations of tube-based control for linear parameter-varying (LPV) systems are presented. A class of cost functions employed by the MPC algorithms developed in the subsequent chapters is proposed, and $\mathcal{K}_\infty$-boundedness of the value function is proven under some assumptions. This result is instrumental for proving stability of the algorithms in the following chapters. Next, an anticipative LPV MPC algorithm based on the construction of homothetic tubes is presented. This approach can be seen as a preliminary solution to Research Question Q_2.

3.1 Introduction

This section introduces the main problem setting for this chapter and for the majority of this thesis. Consider a constrained polytopic LPV state-space (LPV-PSS) representation, connected in closed loop with a yet-to-be designed controller $K_{\mathrm{mpc}} : \mathbb{X} \times \Theta \times \mathbb{N} \to \mathbb{U}$

$$x(k+1) = A\left(\theta(k)\right) x(k) + B\left(\theta(k)\right) K_{\mathrm{mpc}}\left(x(k), \theta(k), k\right). \tag{3.1}$$

As formulated in Question Q_2, the aim of $K_{\mathrm{mpc}}\left(\cdot,\cdot,\cdot\right)$ is to stabilize the origin of the dynamics described by the given constrained LPV-PSS representation. This problem can be stated more precisely as follows.

Problem 3.1. *Given a constrained LPV-PSS representation, find a controller* $K_{\mathrm{mpc}} : \mathbb{X} \times \Theta \times \mathbb{N} \to \mathbb{U}$ *such that the origin of the state space is a regionally asymptotically stable equilibrium of the closed-loop system* (3.1) *in the sense of Definition 2.8.*

As has become clear from the literature overview of Section 1.3.5, a significant number of approaches addressing this very problem already exist. Nonetheless, some particularities in the research questions posed in Section 1.4 motivate the search for yet another, complementary, solution. Specifically, in reference to Research Questions Q_0–Q_2, any designed controller $K_{\text{mpc}}(\cdot,\cdot,\cdot)$ must:

- Be able to make use of any available knowledge about the possible trajectories of the scheduling variable;
- Be "stable" and "tractable".

In this chapter, the first steps towards answering Questions Q_0–Q_2 are undertaken. The remainder of this chapter is structured as follows. In Section 3.2, a description of possible future scheduling trajectories is developed in terms of sequences of sets as an answer to Research Question Q_1. A controller that can make use of such a description, is called "anticipative". Next, in Section 3.3, it is argued that a tube-based approach can be suitable to obtain an anticipative MPC for LPV systems. An overview of the literature on tube-based MPC is provided. Section 3.4 subsequently presents the basic concepts of LPV tube MPC as they will be employed in this thesis. This includes a proof of $\mathcal{K}_\infty$-boundedness of the MPC value function. A preliminary anticipative LPV MPC approach based on homothetic tubes is presented in Section 3.5. This provides a first answer to Research Question Q_2. Finally, Section 3.6 summarizes the chapter.

3.2 Anticipative control

As stated in the primary Research Question Q_0, the objective of the research described in this thesis is to find LPV MPC strategies that can exploit knowledge about the possible behaviors of the scheduling variable. It is thus necessary to define first what is meant by this "knowledge" and how it can be represented in terms of a mathematical description, i.e., to find an answer to Question Q_1.

It is a fundamental part of the LPV concept that at each time instant k, the value $\theta(k)$ of the scheduling variable can be measured (see Assumption 2.1). Another fundamental assumption is that $\theta : \mathbb{N} \to \Theta$ is a "free" signal in the sense that it does not depend on the input- or state variables. Therefore, in principle, for future time instants $k+i$, $i \in [1..\infty)$ it is only known that

$$\theta(k+i) \in \Theta \tag{3.2}$$

where $\Theta \subseteq \mathbb{R}^{n_\theta}$ is the scheduling set. As discussed previously in Section 1.2.2, this assumption can be too pessimistic. In the lithography application considered therein it was, for instance, known that the scheduling variable (the position of the immersion hood (IH)) can not jump instantaneously from one end of its range to the other. Instead, the rate-of-variation (ROV) of the scheduling variable was restricted by the maximum velocity of the IH. In general, the existence of a bound on the scheduling ROV means that there exists a constant $\delta\theta$ such that for all $k \in \mathbb{N}$,

$$|\theta(k+1) - \theta(k)| \leq \delta\theta. \tag{3.3}$$

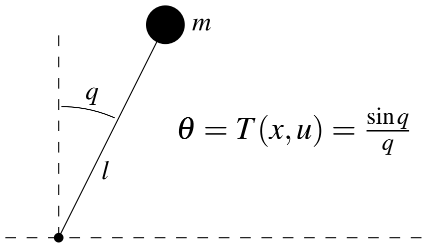


Figure 3.1: The non-linear dynamics of a pendulum can be embedded in an LPV representation by introducing a scheduling variable θ that depends on the angle q.

Thus, if the current value $\theta(k)$ is measured, then the future values $\theta(k+i)$, $i \in [1..\infty)$ are known to belong to a "cone" expanding outwards from the current point $\theta(k)$.

Also, in an LPV embedding of a non-linear system, a ROV on the scheduling variable can often be assumed because there is a dependence between the scheduling and state or input variables, i.e., $\theta = T(x,u)$. For instance, the non-linear dynamics of an inverted pendulum can be embedded in an LPV representation by introducing a scheduling variable θ that depends on the angle of the pendulum[1], as depicted in Figure 3.1. Because the pendulum obeys the laws of physics and can therefore not spin infinitely fast, a bound $\delta\theta$ is known to exist.

In the lithography application, a bound on the scheduling ROV is not the only information available. Because the scheduling variable corresponds to the position of the IH that is controlled to follow a prescribed reference trajectory, future scheduling trajectories can be predicted with high confidence. This situation can be described by defining a "nominal", fully known, scheduling signal $\bar{\theta} : \mathbb{N} \to \Theta$ and an "uncertainty set" $\Delta \subseteq \Theta$ such that it is known that

$$\forall i \in [0..\infty) : \ \theta(k+i) \in \left(\bar{\theta}(k+i) \oplus \Delta\right) \cap \Theta. \tag{3.4}$$

In an embedding, it can be the case that the system is controlled to follow a reference trajectory. Then, it can be assumed that at each future time instant $k+i$, $i \in [1..\infty)$, the state $x(k+i)$ belongs to a set $\mathbb{X}_i \subseteq \mathbb{X} \subset \mathbb{R}^{n_\mathrm{x}}$ around this reference. Likewise, it could be the case that future control inputs $u(k+i)$ are known to belong to sets $\mathbb{U}_i \subseteq \mathbb{U} \subset \mathbb{R}^{n_\mathrm{u}}$. By using the known relationship $\theta = T(x,u)$ between the scheduling and state variables, corresponding scheduling sets can be calculated. This leads to a description of possible future scheduling trajectories of the form

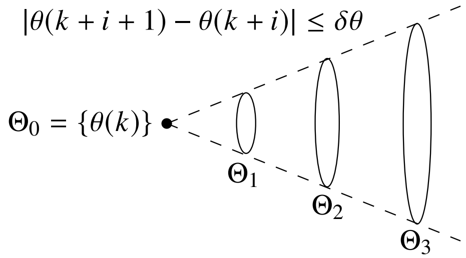
$$\forall i \in [0..\infty) : \ \theta(k+i) \in \{T(x,u) \mid x \in \mathbb{X}_i, u \in \mathbb{U}_i\}. \tag{3.5}$$

The four situations of (3.2)–(3.5) represent particular instances of knowledge on possible future trajectories of θ. Towards the end of providing a general framework in which these and other cases can be described, the notion of "scheduling tube" is introduced.

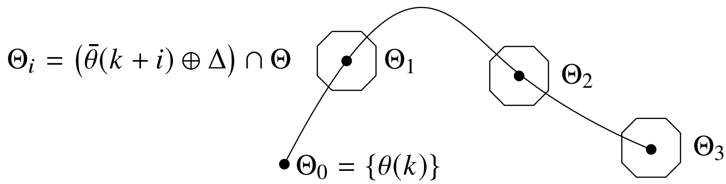
Definition 3.1. *A scheduling tube* $\boldsymbol{\Theta}$ *of length* N *is a sequence of sets*

$$\boldsymbol{\Theta} = \{\Theta_0, \ldots, \Theta_{N-1}\} = \left\{\Theta_i\right\}_{i=0}^{N-1},$$

[1] A concrete example where this is done is given later in Section 8.5.2



(a) Bounded ROV (Equation 3.3).



(b) Uncertainty around nominal trajectory (Equation 3.4).

Figure 3.2: Example of two different scheduling tubes. The availability of the measurement $\theta(k)$ is exploited by setting $\Theta_0 = \{\theta(k)\}$.

where $\forall i \in [0..N-1] : \Theta_i \subseteq \Theta$, *or equivalently,* $\boldsymbol{\Theta} \subseteq \Theta^N$.

In an anticipative predictive controller, at each sampling instant k, a new scheduling tube is constructed such that it contains the expected future variation of the scheduling variable. Due to the availability of the measurement $\theta(k)$, it is always constructed such that

$$\Theta_0 = \{\theta(k)\}.$$

Then it is assumed that, at each instant $k+i$ with $i \in [0..N-1]$ it holds that

$$\theta(k+i) \in \Theta_i.$$

The sets Θ_i, $i \in [1..N-1]$ can be generated according to any one of (3.2)–(3.5), or in any other way that fits the application at hand. Two examples of scheduling tubes, corresponding to (3.3) and (3.4), are provided in Figure 3.2.

A "scheduling model" can now be defined to be any function that—given suitable measurements of the state, scheduling variable, and possibly exogenous inputs—produces a scheduling tube of the form of Definition 3.1. As can be inferred from the previous discussion, the precise form of this model is application-specific and depends on what knowledge is available on its scheduling dynamics. However, any scheduling model must satisfy some general conditions. In particular, the scheduling tube computed at a time instant $k+1$ must in some sense be "contained" inside of the tube that was computed at time instant k. This requirement can be conveniently formulated in terms of a certain ordering of scheduling tubes, which is defined next.

Definition 3.2. *Let* $\mathbf{\Theta} = \{\Theta_i\}_{i=0}^{N-1} \subseteq \Theta^N$ *and* $\mathbf{\Theta}' = \{\Theta'_i\}_{i=0}^{N-1} \subseteq \Theta^N$ *be two scheduling tubes of length N. The relation* $\mathbf{\Theta}' \sqsubseteq \mathbf{\Theta}$ *is satisfied if and only if* $\forall i \in [0..N-2]: \ \Theta'_i \subseteq \Theta_{i+1}$.

In the chapters that follow, to guarantee recursive feasibility of the developed MPC schemes, it will always be required that the employed scheduling model is such that for two sequentially generated scheduling tubes it holds that $\mathbf{\Theta}_{k+1} \sqsubseteq \mathbf{\Theta}_k$.

Furthermore, if a scheduling tube $\mathbf{\Theta} \subseteq \Theta^N$ is given for an LPV-PSS representation, then it is implicitly required that all sets in $\mathbf{\Theta}$ are polytopes (otherwise the state-space (SS) representation would no longer be "polytopic"). In the rest of this thesis, for notational simplicity it is assumed that these polytopes are represented as the convex hulls of equally many vertices. This is summarized in the following assumption.

Assumption 3.1. *Let* $\mathbf{\Theta} \subseteq \Theta^N$ *be a scheduling tube according to Definition 3.1 corresponding to a given LPV-PSS representation. Then, every set* Θ_i *is a polytope described as the convex hull of* q_θ *vertices, i.e.,* $\forall i \in [1..N-1] : \Theta_i = \mathrm{convh}\{\bar{\theta}_i^1, \ldots, \bar{\theta}_i^{q_\theta}\}$.

To summarize, this section has introduced the notion of a "scheduling tube" to enable the description of the available information on possible future scheduling trajectories, thereby answering Research Question Q_1. Given this concept, it has become possible to define precisely what is meant by "anticipative" control:

Definition 3.3. *A predictive controller* $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ *for an LPV system is called* anticipative *if it is able to exploit knowledge about possible future scheduling trajectories, as described in terms of a scheduling tube* $\mathbf{\Theta}_k$ *of the form of Definition 3.1, to potentially achieve increased control performance (as measured by, e.g, the size of the domain of attraction or the realized closed-loop cost). This scheduling tube may be generated in any way that is compatible with the application at hand, as long as at every two sequential time instances the condition* $\mathbf{\Theta}_{k+1} \sqsubseteq \mathbf{\Theta}_k$ *is satisfied.*

3.3 Tubes: motivation and background

In this section, first, the motivation for developing a tube-based LPV control approach is outlined in Section 3.3.1. Next, in Section 3.3.2, a literature overview of currently available tube MPC solutions is provided.

3.3.1 Why tubes?

The previous section introduced the idea of anticipative predictive control. In the considered setting, at each time instant, an estimated behavior of the scheduling variable over N samples into the future is assumed to be available. To make full use of this information, an MPC should use a prediction horizon that is close to this N. If the prediction horizon is shorter, then potentially useful information is neglected. Thus, a methodology for designing a stable and tractable anticipative predictive controller must:

- Allow for arbitrary prediction horizons N so that the full scheduling tube $\mathbf{\Theta}^N$ can be used in predictions;

- Therefore, have a computational complexity that scales well in N.

As summarized in the literature survey on tractable feedback strategies in Section 1.3.5, the tube-based framework satisfies these criteria. Hence, the principles of the tube-based approach will be adopted for formulating the anticipative model predictive controllers described in this thesis.

3.3.2 A survey of tube-based MPC

This section expands on the literature review of Section 1.3.5 by providing a more detailed overview focused on various methods of tube-based MPC. Tube MPC is a paradigm devised to reduce complexity with respect to the dynamic programming solution. This complexity reduction is achieved by employing an approximate description of uncertain state trajectories. Due to the uncertainty in the future behavior of the scheduling variable, the N-step ahead predicted closed-loop response of an LPV system to an optimized sequence of control laws will be set-valued. The exact description of these set-valued trajectories has a representation whose complexity grows exponentially with the prediction horizon. In tube model predictive control (TMPC), this exponential growth is avoided by over-approximating the exact sets by so-called cross sections, which are sets with a shape that is parameterized in some simpler way. Therefore, the representation complexity of the sets does not grow with the horizon. Naturally, the price that has to be paid for this is an increase in conservatism. Tubes were employed first in reachability analysis (Bertsekas and Rhodes 1971), and these ideas were later adopted into MPC.

Although the focus of this thesis is on the control of LPV systems, tube-based MPC approaches were originally proposed to control constrained linear systems subject to *additive* disturbances, and therefore these approaches are reviewed first. In this type of TMPC, the purpose of constructing a tube is to keep the closed-loop trajectories of the disturbed system close to a "nominal" trajectory. The work (Mayne et al. 2005) uses *rigid* tubes, consisting of a sequence of translated copies of a pre-designed robustly positively invariant basic shape set. The center of the tube follows a trajectory which corresponds to the response of the undisturbed system to an optimized sequence of control actions. A policy consisting of these control actions superimposed upon a linear state feedback law is then guaranteed to keep the realized (disturbed) trajectories inside of the tube around this nominal trajectory. The cross sections of this tube are equal to the minimal robustly positively invariant (mRPI) set of the system under the provided linear feedback. The word rigid refers to the fact that the size (scaling) of the cross sections is not optimized on-line. Recursive feasibility is ensured by tightening the constraints: if the nominal state and control trajectory is contained inside of a reduced (tightened) constraint set, then all possible closed-loop trajectories contained inside of the tube around this nominal trajectory are guaranteed to be inside of the original constraint set. The rigid size of the cross sections induces some conservatism, which can be alleviated somewhat by making the initial state of the model a variable in the on-line optimization as well. The computational complexity of this method is virtually equal to that of LTI MPC, as only the initial model state and nominal trajectory have to be optimized on-line while disturbance rejection is ensured by the pre-computed linear feedback law.

In (Langson et al. 2004; Raković et al. 2012a), the concept of so-called *homothetic* tubes is introduced: these tubes consist of a sequence of translated and scaled copies of a polytopic basic shape set[2]. The scalings are optimized on-line, and the center of the tube does not necessarily follow a "nominal" trajectory. Because the shape of the cross sections is fixed and they are only scaled by a scalar factor, the optimization problem to be solved on-line is convex. Due to the increased flexibility obtained by optimizing the scalings on-line, this approach is less conservative than the rigid tube-approach at the price of an increased computational cost. In (Langson et al. 2004), a vertex control policy is used, whereas (Raković et al. 2012a) employs a policy consisting of perturbations upon a pre-determined homogeneous (not necessarily linear) state feedback controller. On-line, (Langson et al. 2004) involves the solution of a single quadratic program (QP) whereas (Raković et al. 2012a) requires the solution of a single linear program (LP). In both cases, the number of variables and constraints in the optimization problem is a linear function of the prediction horizon.

A much more flexible, but more expensive, parameterization is given in (Raković et al. 2012b), which drops the restriction that the basic shape sets are designed off-line. In fact, (Raković et al. 2012b) also generalizes the disturbance-affine policies of (Goulart et al. 2006) by allowing non-affine feedback laws. The on-line computation still consists of solving a single LP, and the number of variables and constraints is a quadratic function of the prediction horizon.

In homothetic TMPC, each cross section is formed by scaling a basic shape set by a scalar factor. The *elastic* tubes of (Raković et al. 2016b) improve upon the flexibility of homothetic tubes, by scaling each hyperplane of the polytopic basic shape set individually. This elasticity renders the tube synthesis problem that has to be solved on-line non-convex. To counter this issue, a conservative convex approximation of the non-convex problem is solved instead. With respect to the homothetic case of (Raković et al. 2012a), the number of decision variables is increased further, but it still grows linearly in the horizon.

In all the TMPC approaches mentioned so far, persistent additive disturbances are assumed to be present, and hence asymptotic stability of the origin can not be established. Instead, convergence to a limit set—typically, the minimal robustly invariant set—centered around the origin is attained. In contrast, for an LPV system, it should be possible to asymptotically stabilize the origin because the uncertainty enters *multiplicatively* in the state transition matrix. Thus, the notion of robust stability required for LPV systems is different from the one employed in the aforementioned TMPC approaches for additively disturbed systems.

A framework for the construction of "stabilizing" tubes, with application to the predictive control of linear systems on assigned initial condition sets, was presented in (Brunner et al. 2013). All system trajectories emanating from the initial condition set are restricted to be inside a tube, which terminates in a controlled λ-contractive terminal set. The specific form of tube used in (Brunner et al. 2013) is homothetic to the terminal set, and in that sense similar to the tubes from (Langson et al. 2004; Raković et al. 2012a). However, the purpose of the tube (stabilizing the origin, versus a robustly invariant set *around* the origin) is different. The theoretical conditions in (Brunner et al. 2013) can be extended beyond the homothetic case, but to do so, new terminal cost- and set-constructions are necessary.

[2]Two sets $A, B \subseteq \mathbb{R}^n$ are homothetic to each other if there exists a vector $z \in \mathbb{R}^n$ and a scalar $\alpha \in \mathbb{R}_+$ such that $A = z \oplus \alpha B$.

Rigid tubes can be applied for controlling LPV systems as well (Rawlings and Mayne 2009, Section 3.5). This works by overbounding the effect of the unknown future parameter variations with an additive disturbance term, which gradually vanishes as the initial state of the system approaches the origin. The true parametric nature of the uncertainty is, however, ignored during predictions.

The papers (Gonzalez et al. 2011; Richards 2005) extend the rigid-tube approach to linear time-varying (LTV) systems. Recall that an LTV system is equivalent to an LPV system for a single, exactly known, choice of scheduling signal (see Section 1.2.3). Both approaches are similar because the local feedbacks are computed on-line based on the known future system realizations, instead of using a single pre-determined local linear time-invariant (LTI) feedback as in (Mayne et al. 2005). The tube cross sections are also computed on-line as a sequence of reachable sets of the disturbed time-varying system under the computed local feedbacks. Like (Mayne et al. 2005), robust feasibility is ensured by constraint tightening. A difference between (Richards 2005) and (Gonzalez et al. 2011) is that (Richards 2005) uses a terminal endpoint constraint, whereas (Gonzalez et al. 2011) uses a terminal set constraint with a terminal cost.

The LTV MPC approach of (Richards 2005) was modified to the LPV case in (Broomhead et al. 2014). An LTI prediction model is employed, and the true future system realization is no longer assumed to be known. The constraint tightening is now done with respect to the uncertainty arising from the unknown future scheduling trajectories as well as to the uncertainty induced by additive disturbances. Furthermore, the terminal endpoint constraint is replaced by a set. The work (Satzger et al. 2017) improves on (Broomhead et al. 2014) by allowing for an LPV prediction model with a rate-bounded scheduling variable, and for an LPV local controller.

Essentially, in (Broomhead et al. 2014; Satzger et al. 2017), the shape and size of the tube cross sections is computed first, and the center trajectory is optimized independently subject to tightened constraints. Provided that the required Pontryagin differences[3] can be computed efficiently, this leads to optimization problems with a complexity comparable to that of standard LTI MPC. However, this also means that the size of the tube is considered to be independent of the optimized center trajectory, which can be conservative. This is because in an LPV system, the parameter uncertainty enters multiplicatively in the state transition matrix, and so the choice of center trajectory has a definite influence on the size of the tube that can be created around it.

In homothetic TMPC on the other hand, the scalings of the tube cross sections are optimized together with the center trajectory. In principle, this allows for a more direct and elegant handling of multiplicative uncertainties. The authors of (Langson et al. 2004) already discuss a possible adaptation of "additive" TMPC to parametrically uncertain systems. Closed-loop stability in a receding-horizon setting is not considered, however. A similar observation was made in (Brunner et al. 2013).

Specialized TMPC approaches for multiplicatively uncertain systems are (Fleming et al. 2015; Muñoz-Carpintero et al. 2015). The shape of the tubes is elastic like those in (Raković et al. 2016a), in the sense that each hyperplane of the basic shape set can be scaled individually. Similarly to (Raković et al. 2016b), this renders the tube synthesis problem non-convex in principle, but this is circumvented by conservatively verifying the set inclusions through a use

[3]The Pontryagin difference between two sets $A, B \subseteq \mathbb{R}^n$ is $A \ominus B = \{x \in \mathbb{R}^n \,|\, \forall b \in B : x + b \in A\}$.

of Farkas' Lemma (Mangasarian 1994). The considered feedback policy consists of control actions superimposed upon a pre-determined linear state feedback, and the tube center has to satisfy a "nominal" trajectory. Recall that a "multiplicatively uncertain" system is considered different from an "LPV" system, in the sense that no measurement of the current value of the scheduling variable is assumed to be available.

All of the aforementioned tube MPC approaches were based on polyhedral tubes, but ellipsoidal tubes have been employed as well. The work (Smith 2004) uses such tubes for the predictive control of systems subject to norm-bounded parametric uncertainties. In (Suzuki and Sugie 2006) an ellipsoidal tube MPC algorithm is given for LPV-PSS representations subject to rate-bounded parameter variations. Note that when using ellipsoids, the on-line optimization problem is still convex, but becomes a semi-definite program (SDP) instead of an LP or QP.

Finally, it must be mentioned that there exist tube MPC approaches for LPV systems subject to additive disturbances, in which the tube is used only to handle the effect of the additive disturbance. The uncertainty due to variations in the scheduling variable is then dealt with by other means. This is different from the approach taken in this thesis, where a tube is constructed to handle multiplicative uncertainties directly, but nevertheless a few of these approaches are now reviewed for completeness.

The work (Bumroongsri 2015) considers LPV systems subject to additive disturbances and in effect combines (Kothare et al. 1996) with (Mayne et al. 2005). The LMI-based LPV MPC approach (Kothare et al. 1996) is used to stabilize the "LPV" part of the dynamics, and on top of that the tube of (Mayne et al. 2005) is employed to ensure constraint satisfaction despite the presence of additive disturbances.

Similarly, (Su et al. 2012) addresses output feedback of LPV systems that are subject to additive disturbances. To control the "nominal" LPV dynamics, the quasi-min-max approach (Lu and Arkun 2000b) is used. It is augmented with a tube that is used to handle the effects of the additive disturbance as well as state estimation errors.

A short comparison of the mentioned tube MPC approaches that are most relevant to the developments of this thesis, is given in Table 3.1.

3.4 Foundations of LPV tube MPC algorithms

In this section, the basic framework that will be employed throughout this thesis for the construction of LPV predictive controllers is introduced. First, the fundamental ideas behind the synthesis of tubes are presented. It is then outlined how a receding-horizon predictive controller can be built on the basis of these ideas. The desirable properties of such a controller, which were already informally introduced in Section 1.3.2, are defined. Next, the notion of *domain of attraction* is explained, and the class of cost functions that will be used for the algorithms developed in the remainder of this thesis is given. For these cost functions, some properties are proven which will be essential later to establish stability of the developed algorithms.

Approach	System	Cross-sections	Control laws
(Langson et al. 2004)	LTI+	Homothetic polytope	Vertex control
(Mayne et al. 2005)	LTI+	Rigid polytope computed off-line	$u = c + Ke$
(Suzuki and Sugie 2006)	LPV	Homothetic ellips., nominal center	$u = c + Kx$
(Raković et al. 2012a)	LTI+	Homothetic polytope	$u = c + \kappa(e)$
(Brunner et al. 2013)	LTI	Homothetic polytope	Vertex control
(Raković et al. 2016a)	LTI+	Elastic polytope	$u = c + K(\alpha)e$
(Fleming et al. 2015)	LPU	Elastic polytope, nominal center	$u = c + Kx$
(Satzger et al. 2017)	LPV	Rigid polytopes computed on-line	$u = c + K(\theta)e$
This thesis, Chapter 4	LPV	Periodically homothetic polytope	Vertex control
This thesis, Chapter 5	LPV	Heterogeneous	Heterogeneous
This thesis, Chapter 6	LPV	Sparsely parameterized polytopes	Sparsely parameterized

Table 3.1: Comparison of some representative tube-based MPC approaches. Besides LPV and LTI, the system classes mentioned in the table are additively disturbed LTI (LTI+) and linear parametrically uncertain (LPU).

3.4.1 Tube synthesis

This section presents a general framework for the construction of tubes, the essential component of the tube MPC algorithms that will be employed throughout this thesis. This framework is inspired by (Brunner et al. 2013) with suitable modifications made to accommodate the anticipative LPV setting. A tube, in essence, is a sequence of sets in the state space (i.e., $\mathbb{R}^{n_x}$) with an associated sequence of control laws. The following definition formalizes this concept.

Definition 3.4. *Let $\mathbf{\Theta} \subseteq \Theta^N$ be given according to Definition 3.1. A tube of length N is a pair*

$$\mathbf{T} = (\mathbf{X}, \mathbf{K}) = \left(\{X_i\}_{i=0}^{N}, \{K_i\}_{i=0}^{N-1}\right)$$

where $X_i \subseteq \mathbb{R}^{n_x}$ are sets and where $K_i : X_i \times \Theta_i \to \mathbb{U}$ are $\mathcal{CH}_1$ control laws such that for all $i \in [0..N-1]$, the condition $\mathcal{I}(X_i, \Theta_i | K_i) \subseteq X_{i+1} \cap \mathbb{X}$ holds[4]. Each set X_i is called a cross section.

The length of the tube N in Definition 3.4, is called the *prediction horizon*. A subtle, but important, feature of the tubes in Definition 3.4 is the fact that the cross sections X_i are not necessarily subsets of the state constraints. Only the actual possible trajectories of the system, as generated by the image mapping $\mathcal{I}(X_i, \Theta_i | K_i)$, are required to be inside of the constraints.

Remark 3.1. *When reading this thesis, it is important to be aware of the difference between a "tube" (Definition 3.4) and a "scheduling tube" (Definition 3.1).*

In the sequel, occasionally the scalar multiple of a tube will be used, which can be defined as

$$\alpha\mathbf{T} = \left(\{\alpha X_i\}_{i=0}^{N}, \{K_i\}_{i=0}^{N-1}\right). \tag{3.6}$$

[4]See Definition 2.11 for the definition of the image map $\mathcal{I}(\cdot, \cdot|\cdot)$.

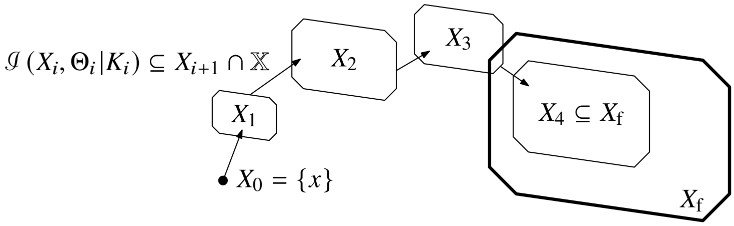


Figure 3.3: A "feasible" tube $\mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta})$ in a two-dimensional state space with $N = 4$. Recall that $\mathbf{\Theta} = \{\Theta_i\}_{i=0}^{N-1}$.

Note that in (3.6) it is not necessary to explicitly scale the sequence $\{K_i\}_{i=0}^{N-1}$, because the assumed homogeneity of the controllers implies that

$$\{K_i(x,\theta) \,|\, x \in \alpha X_i, \theta \in \Theta_i\} = \{K_i(\alpha x,\theta) \,|\, x \in X_i, \theta \in \Theta_i\} = \alpha \{K_i(x,\theta) \,|\, x \in X_i, \theta \in \Theta_i\}.$$

The scaled tube satisfies the state and input constraints for all $\alpha \in [0,1]$, due to the PC-property of $\mathbb{X}$ and $\mathbb{U}$.

This section will now review in general terms the process of constructing tubes in the context of MPC. When particular tube constructions are presented in the course of the following chapters, modifications to these general definitions will be made where necessary. The purpose of this section is therefore to give an outline of the general approach which is the foundation for the next chapters.

In what follows, a tube is called *feasible* if it satisfies an initial condition constraint, and if it satisfies a terminal constraint. The set of such feasible tubes $\mathcal{T}_N(\cdot,\cdot)$ can be defined as

$$\mathcal{T}_N(x, \mathbf{\Theta}) = \{\mathbf{T} \,|\, \mathbf{T} \text{ satisfies Def. 3.4 with } X_0 = \{x\} \text{ and } X_N \subseteq X_\mathrm{f}\} \tag{3.7}$$

where $X_\mathrm{f} \subseteq \mathbb{X}$ is a terminal set. An example of a feasible tube $\mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta})$ is depicted in Figure 3.3.

It is useful to choose from this set of feasible tubes a single tube that optimizes a given performance criterion. This is done by solving the *tube synthesis* problem

$$V(x, \mathbf{\Theta}) = \min_{\mathbf{T}} J_N(\mathbf{T}, \mathbf{\Theta}) \text{ subject to } \mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta}) \tag{3.8}$$

where

$$J_N(\mathbf{T}, \mathbf{\Theta}) = \sum_{i=0}^{N-1} \ell(X_i, K_i, \Theta_i) + F(X_N) \tag{3.9}$$

is a finite-horizon cost function with $\ell(\cdot,\cdot,\cdot)$ being the stage cost and $F(\cdot)$ being the terminal cost. The stage cost is selected to optimize a certain performance criterion: this could, for instance, be to drive the state to the origin as fast as possible or to minimize the energy of the control inputs. A type of norm-based stage cost that will be used in a large part of this thesis will be presented later in Section 3.4.5. The terminal cost $F(\cdot)$ must be computed in a special way to guarantee stability of the closed loop (see Section 2.4). Note that the cost functions take set-valued arguments.

The optimization (3.8) involves arbitrary sets and control laws as decision variables, so it is actually intractable to solve. Therefore, the optimization is always carried out with respect to *parameterized* sets and controllers. The selected parameterization determines the computational complexity of (3.8), as well as the level of achievable performance of the resulting controller. Typically, it is desired to have a parameterization that leads to a favorable balance between computational complexity and obtained control performance. The differences between the algorithms summarized in Table 3.1 mainly come from different choices of the parameterization within the general framework of Definition 3.4.

The illustration of Figure 3.3 already gives an example of a parameterized tube, because each cross section corresponds to a scaled and translated copy of the terminal set X_f, i.e,

$$X_i = z_i \oplus \alpha_i X_\mathrm{f} \text{ for some } (z_i, \alpha_i) \in \mathbb{R}^{n_\mathrm{x}} \times \mathbb{R}_+.$$

Then, the centers z_i and the scalings α_i are parameters which correspond to decision variables in the tube synthesis (3.8). If the cross sections are parameterized in this way it is said that they are *homothetic* to X_f, and so this parameterization is called a homothetic parameterization. Parameterizations of this type have been extensively used in the literature (see Section 3.3.2), and several variations on it will be considered throughout the course of this thesis.

The function $V(\cdot,\cdot)$ in (3.8) is called the *value function*, and an optimizer of (3.8) is denoted as

$$\mathbf{T}^\star = \left(\left\{X_i^\star\right\}_{i=0}^{N}, \left\{K_i^\star\right\}_{i=0}^{N-1}\right), \tag{3.10}$$

where by definition, $V(x,\mathbf{\Theta}) = J_N(\mathbf{T}^\star, \mathbf{\Theta})$.

3.4.2 Receding-horizon tube MPC

The receding-horizon methodology is the standard approach in which the tube synthesis as presented in the previous subsection can be used to obtain an MPC algorithm. It is summarized in Algorithm 3.1.

If at a certain time instant $k \in \mathbb{N}$ no solution to the problem (3.8) exists, it is not possible to determine a valid control input $u(k)$ that can be applied to the plant. This can mean, for instance, that it is not possible to determine an input $u(k)$ that satisfies the constraints and that is such that the state constraints will not be violated. Thus, the execution of the controller has to be aborted, as is done in Step 10 of Algorithm 3.1. If Algorithm 3.1 is recursively feasible, and if there are no unmodeled disturbances or modeling errors present, then this abort can only occur at time $k = 0$. This is because in a recursively feasible controller, the existence of a solution at $k = 0$ implies the existence of a solution at all future $k > 0$. In practice it may sometimes be possible to recover from an "aborted" state, for instance by dropping some constraints $(\mathbb{X}, \mathbb{U})$ and attempting to re-solve (3.8).

In a receding-horizon scheme, establishing recursive feasibility and closed-loop stability is not trivial. Recursive feasibility has to be ensured through a special design of the terminal set X_f and the tube parameterization. These two design elements are often mutually dependent. Closed-loop stability has to be ensured by appropriately choosing the terminal cost $F(\cdot)$.

Algorithm 3.1 Prototype receding-horizon TMPC algorithm.

Require: $N \in [1..\infty)$

$\boldsymbol{\Theta}_{-1} \leftarrow \Theta^N$

$k \leftarrow 0$

loop

  Construct $\boldsymbol{\Theta}_k = \left\{\{\theta(k)\}, \left\{\Theta_{i|k}\right\}_{i=1}^{N-1}\right\} \subseteq \Theta^N$ such that $\boldsymbol{\Theta}_k \sqsubseteq \boldsymbol{\Theta}_{k-1}$

  if $\mathcal{T}_N\left(x(k), \boldsymbol{\Theta}_k\right) \neq \emptyset$ **then**

    Solve (3.8) to obtain $\mathbf{T}^\star \in \mathcal{T}_N\left(x(k), \boldsymbol{\Theta}_k\right)$

    Apply $u(k) = K_0^\star\left(x(k), \theta(k)\right) = u_0^\star$ to the system (2.4)

    $k \leftarrow k + 1$

  else

    abort

  end if

end loop

Particular constructions of the terminal set, tube parameterization, and terminal cost will be presented in the following chapters.

3.4.3 Desirable properties revisited

In Section 1.3.2, the desirable properties that an MPC algorithm must satisfy have been informally introduced. The purpose of this section is to give a more precise definition for the introduced concepts of recursive feasibility, stability, and tractability.

Assume that a controller $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ has been designed leading to a closed-loop system (3.1). At time instant k, suppose the controller solves (3.8) and produces a control input $u(k) = K_{\mathrm{mpc}}(x(k), \theta(k), k)$. After applying this input to the system, a new measurement $(x(k+1), \theta(k+1))$ becomes available. The question is if the existence of a solution to (3.8) can be guaranteed at time $k+1$. More specifically, the question to be answered is whether or not the existence of a solution at time k implies the existence of a solution at the next time instant $k+1$. If the answer is "yes", then the controller $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ is called recursively feasible. Clearly, this property is highly desirable. If no solution $u(k+1)$ exists, the behavior of the closed-loop system becomes essentially undefined, and therefore no stability guarantee can be given. Based on this discussion, it is now possible to give a thorough definition of a "stable" strategy, which was an integral part of Research Question Q_0 in Section 1.4.

Definition 3.5. *A tube MPC strategy, leading to a controller* $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$*, is called* stable *if*

(i) $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ *is recursively feasible;*

(ii) The origin is a regionally asymptotically stable equilibrium of the closed-loop system (3.1) *in the sense of Definition 2.8.*

Note that it is not possible to speak of closed-loop stability if $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ is not recursively feasible, because the closed-loop behavior of (3.1) is undefined when (3.8) becomes infeasible. Hence, recursive feasibility is a necessary condition for asymptotic stability of the origin.

The optimization problem (3.8) is the core of any tube-based MPC strategy. Thus, when speaking about the tractability of an MPC algorithm, what is really meant is the tractability of the underlying optimization problem (3.8). As was already discussed in the introduction, convexity and scalability are the two important parts that together form the notion of tractability. This is now formalized in the following definition.

Definition 3.6. *Let $\mathcal{C}(N) : \mathbb{N} \to \mathbb{R}_+$ be proportional to the number of variables and constraints in* (3.8)*. A tube MPC strategy is called* tractable*, if*

(i) The corresponding optimization problem (3.8) *is convex;*

(ii) $\mathcal{C}(\cdot)$ is polynomially bounded in the sense of Definition 2.3.

In Section 3.1, it was said that tube-based algorithms available in the literature often have a linear complexity in N. This will turn out to be true also for the the algorithms presented in the remaining chapters of this thesis.

3.4.4 Domain of attraction

For a given sequence $\mathbf{\Theta} \subseteq \Theta^N$, the domain of attraction (DOA) of the closed-loop system (3.1) under a controller defined by (3.8) is

$$\mathcal{X}_N(\mathbf{\Theta}) = \{x \in \mathbb{X} \mid \mathcal{T}_N(x, \mathbf{\Theta}) \neq \emptyset\}. \tag{3.11}$$

The domain of attraction (DOA) is the set of initial states for which a feasible tube $\mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta})$ exists. In a stable MPC algorithm, if an initial state is in the DOA, it can be guaranteed that the controller drives the state of the system to the origin asymptotically. In contrast, if the initial state is outside of the DOA, it is not possible to determine a control input $u(k)$ that should be applied to the plant (i.e., the value of $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ is undefined). It this therefore usually desirable that the DOA (3.11) corresponding to (3.8) is as large as possible.

For a fixed prediction horizon N, a given target set X_{f}, and a given sequence $\mathbf{\Theta} \subseteq \Theta^N$, the maximum DOA that can be achieved by any possible controller is denoted by $\mathcal{X}_N^{\max}(\mathbf{\Theta})$. This set can be computed recursively in terms of backwards reachable sets as

$$\begin{aligned} \forall i \in [0..N-1] : \mathcal{X}_{i+1}^{\max}(\mathbf{\Theta}) &= \mathcal{Q}\left(\mathcal{X}_i^{\max}, \Theta_{N-i-1}\right), \\ \mathcal{X}_0^{\max}(\mathbf{\Theta}) &= X_{\mathrm{f}}, \end{aligned} \tag{3.12}$$

with $\mathcal{Q}(\cdot,\cdot)$ as in Definition 2.10. Depending on how the tube $\mathbf{T}$ in (3.8) is parameterized, it may happen that $\mathcal{X}_N(\mathbf{\Theta}) = \mathcal{X}_N^{\max}(\mathbf{\Theta})$. Usually however, the parameterization is designed such that (3.8) is conservative, and $\mathcal{X}_N(\mathbf{\Theta}) \subset \mathcal{X}_N^{\max}(\mathbf{\Theta})$. In fact, the meaning of the word "conservative" can be defined in terms of the DOA.

Definition 3.7. *A controller $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ based on the tube synthesis* (3.8) *is said to be conservative, if and only if there exists a $\mathbf{\Theta} \subseteq \Theta^N$ such that the DOA* (3.11) *is a proper subset of the maximal DOA* (3.12)*, i.e., if $\mathcal{X}_N(\mathbf{\Theta}) \subset \mathcal{X}_N^{\max}(\mathbf{\Theta})$. Furthermore, a controller $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ is called "less" conservative than another controller $K'_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ if and only if for all $\mathbf{\Theta} \subseteq \Theta^N$, $\mathcal{X}_N(\mathbf{\Theta}) \subseteq \mathcal{X}'_N(\mathbf{\Theta})$. This is provided that both $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ and $K'_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ use the same prediction horizon N and the same terminal set X_{f}.*

Achieving a large DOA often necessitates the use of a computationally expensive parameterization, and a bit of conservatism is acceptable if this leads to a more tractable optimization problem (3.8). Observe—for example—that an MPC based on dynamic programming (DP) achieves the largest possible DOA, but at an excessive computational cost.

3.4.5 The class of cost functions

In this section, the class of stage cost functions employed throughout this thesis is introduced. Sufficient conditions on the terminal cost $F(\cdot)$ under which the value function $V(\cdot,\cdot)$ can be upper and lower bounded by a pair of $\mathcal{K}_\infty$-functions are provided.

As discussed in the previous sections, the tube parameterization and the corresponding terminal cost will be different in each of the following chapters. However, the stage cost used in (3.9) will be the same in each chapter except Chapter 7. Hence, its definition and its important properties are presented here. The stage cost

$$\ell\left(X, K, \Theta\right) = \max_{(x,\theta)\in X\times\Theta} \left(\|Qx\|^c + \|RK(x,\theta)\|^c\right) \tag{3.13}$$

is proposed where $\|\cdot\|$ can be any vector norm, $c \geq 1$, and $(Q,R) \in \mathbb{R}^{n_\mathrm{q}\times n_\mathrm{x}} \times \mathbb{R}^{n_\mathrm{r}\times n_\mathrm{x}}$ are full column rank matrices corresponding to tuning parameters.

Several properties of (3.13) are important to guarantee stability of the MPC algorithms presented in the next chapters. These are summarized in the following proposition.

Proposition 3.1. *In what follows, let $K : X\times\Theta \to \mathbb{U}$ be a $\mathcal{CH}_1$-controller.*

(i) For all subsets $X'\times\Theta' \subseteq X\times\Theta$, it holds that $\ell\big(X',K,\Theta'\big) \leq \ell\big(X,K,\Theta\big)$.

(ii) There exists a $\mathcal{K}_\infty$-function $\underline{\ell}$ such that $\underline{\ell}\big(d_H^0(X)\big) \leq \ell(X,K,\Theta)$.

(iii) The stage cost is homogeneous of degree c in the sense that for all $\alpha\in\mathbb{R}_+$, $\ell(\alpha X,K,\Theta) = \alpha^c\ell(X,K,\Theta)$.

Proof of (i). This property follows directly from the definition of (3.13) in terms of the maximum over a compact set.

Proof of (ii). From (3.13), $\max_{x\in X}\|Qx\|^c \leq \ell\big(X,K,\Theta\big)$. As Q is assumed to be full column rank, $x\mapsto\|Qx\|$ is a norm (in particular, $\|Qx\| = 0$ if and only if $x = 0$). All norms in finite-dimensional vector spaces are equivalent, hence $\exists\alpha > 0 : \alpha\|x\| \leq \|Qx\|$, implying that $\alpha\max_{x\in X}\|x\| = \alpha d_H^0(X) \leq \max_{x\in X}\|Qx\|$. This directly leads to

$$\left(\alpha d_H^0(X)\right)^c \leq \left(\max_{x\in X}\|Qx\|\right)^c = \max_{x\in X}\|Qx\|^c,$$

proving the statement with $\underline{\ell}\big(\xi\big) = (\alpha\xi)^c$.

Proof of (iii). This property follows directly from the $\mathcal{CH}_1$-property of the controller $K(\cdot,\cdot)$ combined with homogeneity of the vector norm $\|\cdot\|$. □

To prove closed-loop stability of a predictive controller, the usual approach is to show that the value function of (3.8) is a Lyapunov function in the sense of Definition 2.9. An important first step, then, is to show that the value function satisfies Definition 2.9.(i), i.e., to show that it can be upper and lower bounded by a pair of $\mathcal{K}_\infty$-functions. The remainder of this section is devoted to providing, under some assumptions, a proof of the existence of these bounds. This proof exploits homogeneity of the LPV state-space (LPV-SS) representation, of the control laws in the tube, and of the involved cost functions. Consider first the following assumptions on the terminal cost.

Assumption 3.2. *Consider the terminal cost function* $F(\cdot) : 2^{\mathbb{R}^{n_x}} \to \mathbb{R}_+$ *in* (3.9).

(i) Let c have the same value as in (3.13). *The function* $F(\cdot)$ *is homogeneous of degree c in the sense that for all* $\alpha \in \mathbb{R}_+$, $F(\alpha X) = \alpha^c F(X)$.

(ii) There exist $\mathcal{K}_\infty$*-functions* $\underline{F}, \overline{F}$ *such that for all* $X \subseteq X_{\mathrm{f}}$, $\underline{F}\big(d_H^0(X)\big) \leq F\big(X\big) \leq \overline{F}\big(d_H^0(X)\big)$.

Observe that by Proposition 3.1.(iii) and Assumption 3.2.(i), the whole cost function $J_N(\cdot,\cdot)$ is homogeneous of degree c in the sense that $J_N(\alpha \mathbf{T}, \mathbf{\Theta}) = \alpha^c J_N(\mathbf{T}, \mathbf{\Theta})$ for all $\alpha \in \mathbb{R}_+$. The main result of this section is stated next.

Proposition 3.2. *There exist* $\mathcal{K}_\infty$*-functions* $\underline{V}, \overline{V}$ *such that for all* $x \in \mathbb{X}$ *and* $\mathbf{\Theta} \subseteq \Theta^N$ *for which* (3.8) *is feasible, it holds* $\underline{V}\big(\|x\|\big) \leq V\big(x, \mathbf{\Theta}\big) \leq \overline{V}\big(\|x\|\big)$.

Proof. The lower bound on the value function is established trivially as $V(x, \mathbf{\Theta}) \geq \underline{\ell}(\|x\|)$. The feasible set of initial states $\mathcal{X}_N(\mathbf{\Theta}) = \{x \in \mathbb{X} \mid \mathcal{T}_N(x, \mathbf{\Theta}) \neq \emptyset\}$ of (3.8) for a given sequence $\mathbf{\Theta}$ is a PC-set. Let $\phi(x) = \psi_{\mathcal{X}_N(\mathbf{\Theta})}(x)$. For any $x \in \mathcal{X}_N(\mathbf{\Theta})$, it holds that $x \in \partial\phi(x)\mathcal{X}_N(\mathbf{\Theta})$. Hence,

$$V(x, \mathbf{\Theta}) \leq \max_{x \in \phi(x)\partial\mathcal{X}_N(\mathbf{\Theta})} V(x, \mathbf{\Theta}) = \max_{x \in \partial\mathcal{X}_N(\mathbf{\Theta})} V(\phi(x)x, \mathbf{\Theta}). \tag{3.14}$$

The representation (2.4) is homogeneous of degree one in (x, u), and $(X_{\mathrm{f}}, \mathbb{X}, \mathbb{U})$ are PC-sets. Therefore, the existence of a $\mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta})$ implies that for all $\alpha \in [0, 1]$, there exists a $\mathbf{T}^\circ = \alpha\mathbf{T} \in \mathcal{T}_N(\alpha x, \mathbf{\Theta})$. Because the cost function is homogeneous of degree c, it follows that $J_N(\mathbf{T}^\circ, \mathbf{\Theta}) = \alpha^c J_N(\mathbf{T}, \mathbf{\Theta})$. Now observe that, for all $x \in \mathcal{X}_N(\mathbf{\Theta})$, $\phi(x) \in [0, 1]$ holds, and that the solution $\mathbf{T}^\circ$ is feasible but not necessarily optimal for the initial state αx. Combining these facts with (3.14) yields

$$V(x, \mathbf{\Theta}) \leq \max_{x \in \partial\mathcal{X}_N(\mathbf{\Theta})} V(\phi(x)x, \mathbf{\Theta}) \leq \phi^c(x) \max_{x \in \partial\mathcal{X}_N(\mathbf{\Theta})} V(x, \mathbf{\Theta}). \tag{3.15}$$

Because $(X_{\mathrm{f}}, \mathbb{X}, \mathbb{U})$ are compact, it can be assumed that there exists a constant $\hat{V} > 0$ such that

$$\max_{x \in \partial\mathcal{X}_N(\mathbf{\Theta})} V(x, \mathbf{\Theta}) \leq \hat{V}. \tag{3.16}$$

As $\phi(\cdot)$ is the gauge function of a PC-set, there exists a $\mathcal{K}_\infty$-function $\overline{\phi}$ such that for all $x \in \mathbb{R}^{n_x} : \phi(x) \leq \overline{\phi}(\|x\|)$. Combining this with (3.15)-(3.16) gives the final desired result that $\overline{V}(\xi) = (\overline{\phi}(\xi))^c \hat{V}$ is a $\mathcal{K}_\infty$-upper bound on $V(\cdot, \mathbf{\Theta})$. □

3.5 Homothetic tubes in anticipative LPV MPC

This section proposes an initial solution to Research Question Q_2 based on the construction of homothetic tubes. Homothetic tubes were introduced in (Langson et al. 2004; Raković et al. 2012a) in the context of tube-based MPC for LTI systems subject to additive disturbances, and their possible utilization in LPV MPC was pointed out, e.g., in (Brunner et al. 2013; Langson et al. 2004). However, these works did not fully address the construction of a terminal set and cost to ensure recursive feasibility and closed-loop stability in the LPV setting.

The preliminary approach presented in this section will serve as a basis for the extended approaches developed in the following chapters. In this approach, the cross sections of the tube (see Definition 3.4) are parameterized as homothetic to a terminal set, whereas the associated controllers are parameterized as vertex controllers. First, let us define the precise meaning of "vertex control".

Definition 3.8. *Let* $X = \operatorname{convh}\left\{\bar{x}^1, \dots, \bar{x}^{q_\mathrm{x}}\right\} \subset \mathbb{R}^{n_\mathrm{x}}$ *and, likewise, let* $\Theta = \operatorname{convh}\left\{\bar{\theta}^1, \dots, \bar{\theta}^{q_\theta}\right\}$. *Let* $p^\mathrm{k} = \left(u^{(1,1)}, \dots, u^{(q_\mathrm{x},1)}, \dots, u^{(q_\mathrm{x},q_\theta)}\right) \in \mathbb{U}^{q_\theta q_\mathrm{x}}$ *be a discrete set of corresponding control actions. Then* $\operatorname{vertpol}(\cdot|X,\Theta) : \mathbb{U}^{q_\theta q_\mathrm{x}} \to (X \times \Theta \to \mathbb{U})$ *is a second-order function defined such that* $\operatorname{vertpol}\left(p^\mathrm{k}\middle|X,\Theta\right)(x,\theta) = \sum_{i=1}^{q_\mathrm{x}} [\operatorname{convm}(x|X)]_i \sum_{j=1}^{q_\theta} [\operatorname{convm}(\theta|\Theta)]_j \, u^{(i,j)}$, *where* $[v]_i$ *denotes the i-th element of a vector v.*

Next, the notion of a homothetic tube—specialized to the LPV setting—can be defined.

Definition 3.9. *Let* $\mathbf{T}$ *be a tube according to Definition 3.4, and let* $S = \operatorname{convh}\{\bar{s}^1, \dots, \bar{s}^{q_\mathrm{s}}\} \subseteq \mathbb{R}^{n_\mathrm{x}}$. *For all* $i \in \mathbb{N}_{[0,N]}$, *introduce parameter sets* $\mathbb{P}^\mathrm{x} = \mathbb{R}_+ \times \mathbb{R}^{n_\mathrm{x}}$ *and* $\mathbb{P}^\mathrm{k} = \mathbb{R}^{q_\theta q_\mathrm{s} n_\mathrm{u}}$. *A tube* $\mathbf{T}$ *is called a* homothetic tube *if it satisfies the following conditions:*

(i) For all $i \in \mathbb{N}_{[0,N]}$ *there exists a parameter* $p_i^\mathrm{x} = (\alpha_i, z_i) \in \mathbb{P}^\mathrm{x}$ *such that* $X_i = P^\mathrm{x}(p_i^\mathrm{x}) = z_i \oplus \alpha_i S$.

(ii) For all $i \in \mathbb{N}_{[0,N-1]}$ *there exists a parameter* $p_i^\mathrm{k} = \left(u_i^{(1,1)}, \dots, u_i^{(q_\mathrm{s},1)}, \dots, u_i^{(q_\mathrm{s},q_\theta)}\right) \in \mathbb{P}^\mathrm{k}$ *such that* $K_i = P^\mathrm{k}(p_i^\mathrm{k}) = \operatorname{vertpol}(p_i^\mathrm{k}|X_i,\Theta_i)$.

Given Definition 3.9, the set of feasible homothetic tubes corresponding to (3.8) can be given as

$$\mathcal{T}_N(x,\mathbf{\Theta}) = \{\mathbf{T} \,|\, \mathbf{T} \text{ satisfies Def. 3.9 with } X_0 = \{x\} \text{ and } X_N \subseteq X_\mathrm{f}\}. \tag{3.17}$$

Next, as a special case of (3.13), consider the ∞-norm based stage cost

$$\ell(X,K,\Theta) = \max_{(x,\theta)\in X\times\Theta} \left(\|Qx\|_\infty + \|RK(x,\theta)\|_\infty\right) \tag{3.18}$$

and the terminal cost

$$F(X) = \frac{\ell(X_\mathrm{f},K_\mathrm{f},\Theta)}{1-\lambda} \Psi_{X_\mathrm{f}}(X) \tag{3.19}$$

where $\ell(X_\mathrm{f},K_\mathrm{f},\Theta)$ is a constant which is computed off-line by evaluating (3.18) over the terminal set. The terminal cost (3.19) is based on the set-gauge function of the terminal set X_f

(see Definition 2.2). Terminal costs based on a "normal" gauge function were considered before, e.g., in (Raković and Lazar 2012) for LTI MPC with a gauge-based stage cost, in (Gallieri 2016) for ℓ_1-regularized LTI MPC, and in (Besselmann et al. 2012) in the context of explicit dynamic programming for LPV systems. In (Brunner et al. 2013) a set-gauge based terminal cost similar to (3.19) was employed for tube-based MPC of LTI systems subject to uncertain initial conditions.

With the tubes of Definition 3.9 and the cost functions (3.18)-(3.19), a stable and tractable receding-horizon algorithm is obtained. This result is stated in the following theorem.

Theorem 3.1. *Let X_f be a controlled λ-contractive set in the sense of Definition 2.13. In Definition 3.9, set $S = X_\mathrm{f}$. Let $\mathbf{\Theta} \subseteq \Theta^N$, $\mathbf{\Theta}^+ \subseteq \Theta^N$ be two sequences related as $\mathbf{\Theta}^+ \sqsubseteq \mathbf{\Theta}$. Then, Algorithm 3.1 achieves the following properties:*

(i) It is recursively feasible, i.e., if $\exists \mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta})$, then $\exists \mathbf{T}^+ \in \mathcal{T}_N(x^+, \mathbf{\Theta}^+)$ where $x^+ = A(\theta)x + B(\theta)K_0(x, \theta)$.

(ii) The origin is an asymptotically stable equilibrium of the closed-loop system (3.1) *in the sense of Definition 2.8.*

(iii) The tube synthesis problem (3.8) *is a linear program. The number of variables and the amount of constraints are linear functions of the prediction horizon N.*

The properties in this theorem follow by taking $M = 1$ in the more general setting which is the topic of Chapter 4. Therefore, the proof is omitted here.

From statements (i)-(iii) in Theorem 3.1, it follows that the MPC approach presented in this section already gives a possible answer to Research Question Q_2: that is, it is a computationally tractable MPC approach with a stability guarantee that can exploit information about possible future scheduling trajectories.

3.6 Concluding remarks

This chapter has introduced a common framework for the control approaches that are described in this thesis. A flexible description of future uncertain scheduling trajectories was developed, which lead to the notion of "anticipative" control. The idea of tube-based control of LPV systems that will be used throughout this thesis, was presented. An algorithm based on the construction of homothetic tubes was developed. This algorithm is computationally tractable, comes with a closed-loop stability guarantee, and can exploit information about possible future scheduling trajectories: therefore, it provides a first possible answer to Research Question Q_2.

However, this is but a possible answer: a more advanced answer to Question Q_2 will be provided in the upcoming chapter, which focuses on the development of a more general stability condition based on the notion of finite-step contraction.

Chapter 4

Guaranteeing feasibility and stability

The present chapter addresses the problem of guaranteeing recursive feasibility and closed-loop stability in a receding-horizon linear parameter-varying (LPV) tube model predictive control (MPC) algorithm. Feasibility and stability can be guaranteed through the construction of an appropriate terminal set and terminal cost. The construction provided in this chapter is based on the concept of finite-step λ-contraction, which is a more relaxed variant of the usual notion of λ-contraction. A periodic tube parameterization is proposed, which together with a finite-step contractive terminal set leads to a recursively feasible algorithm. The terminal cost is constructed based on a new Lyapunov-type metric on proper C-sets. Recursive feasibility and closed-loop stability of the basic homothetic tube-based algorithm from Section 3.5 follows as a special case from the results of this chapter. It is shown that the tube synthesis problem to be solved on-line is a single linear program (LP). Numerical examples are provided to demonstrate the computational properties of the approach.

4.1 Introduction

A basic homothetic tube-based LPV MPC algorithm was presented previously in Section 3.5. In this algorithm, the set inclusions $\mathcal{I}(X_i, \Theta_i | K_i) \subseteq X_{i+1} \cap \mathbb{X}$ from Definition 3.4 can be verified by checking a set of linear constraints. The number of necessary constraints however depends strongly on the complexity of the representation of the polytope X_f, because each set X_i is homothetic to X_f. Further, as the control laws $K_i : X_i \times \Theta_i \to \mathbb{U}$ were parameterized as vertex controllers, the number of decision variables necessary to parameterize these controllers also directly depends[1] on the number of vertices of X_f.

To guarantee stability, the terminal set X_f in the method of Section 3.5 has to be controlled λ-contractive. In fact, all of the tube-based algorithms reviewed in Section 3.3.2 rely on terminal sets that are controlled contractive or invariant. It is well-known that the computation of polyhedral contractive sets for LPV systems is not easy. An iterative algorithm based on reachable set computations is guaranteed to converge to the maximum-volume contractive set if one exists (Blanchini and Miani 2015; Miani and Savorgnan 2005). However, the representation complexity of these sets can grow exponentially as a function of the state dimension, making

[1] An exact expression for the number of variables and constraints in the resulting LP will be given in the course of this chapter in Section 4.4.

computation difficult for systems with more than a few states. Furthermore, even if a contractive set can be found, the complexity of its representation may still be too great to be used in a computationally viable tube model predictive control (TMPC) strategy.

Some research effort has therefore been spent on developing methods for computing invariant- or contractive sets of reduced complexity with respect to the maximal ones, see, e.g., (Athanasopoulos and Bitsoris 2010; Blanco et al. 2010; Fiacchini and Alamir 2017; Munir et al. 2016; Pluymers 2006; Sheer and Gutman 2016; Tahir and Jaimoukha 2015).

In this chapter, an alternative approach is proposed. The requirement that the terminal set must be controlled λ-contractive is relaxed. Instead of "standard" contractive sets, so-called finite-step controlled λ-contractive sets are considered. Informally, a set is finite-step contractive if state trajectories starting inside of the set are allowed to leave the set temporarily, before returning after a finite number—say, M—of time instances. A useful fact related to such sets is the following.

Proposition 4.1. *(Athanasopoulos et al. 2013, Proposition 1) Let $x^+ = f(x,u)$ be a system subject to constraints of the form of Definition 2.7. Let $S_0 \subset \mathbb{X}$ be a PC-set. Under some technical conditions on $f(\cdot,\cdot)$, the following statements are equivalent:*

(i) The set S_0 is controlled finite-step contractive with respect to the system $x^+ = f(x,u)$ and under the given constraints.

(ii) There exists a stabilizing state feedback $K : \mathbb{X} \to \mathbb{U}$ such that the closed-loop system $x^+ = f(x,K(x))$ is asymptotically stable on S_0, and respects the constraints.

The previous proposition implies an important difference between "normal" (i.e., one-step) and finite-step contractive sets. A one-step contractive set has to be constructed using an algorithmic procedure, and it is therefore difficult to give a-priori bounds on its representational complexity. On the other hand, if the system is stabilizable, it is possible to make *any* PC-set S_0 finite-step contractive through the use of an appropriate controller[2]. Of course, if the initial set S_0 is chosen poorly, the number of time steps required to achieve finite-step contraction may still be too large for practical purposes. Nonetheless, the proposed setting offers a new degree of freedom—namely, the choice of set S_0—in the TMPC design.

In predictive control, finite-step invariant sets have been employed in nominal non-linear MPC in (Lazar and Spinu 2015). The framework of (Gondhalekar and Jones 2011) employs periodically invariant sets in MPC of linear periodic systems. Finite-step invariant *ellipsoids* for LPV systems were introduced in (Y. I. Lee et al. 2005). Therein, a stabilizing open-loop min-max MPC algorithm that uses these ellipsoids as a terminal set is proposed. A practical controller based on the ellipsoids of (Y. I. Lee et al. 2005) was given in (Y. I. Lee and Kouvaritakis 2006), where a single free control action is being optimized on-line.

In TMPC, finite-step invariant- or contractive sets have not been used previously. The main contribution of this chapter is therefore to provide a new terminal set, terminal cost, and tube parameterization that enables the use of sets of this type in constructing stabilizing tube-based LPV predictive controllers.

[2] In this chapter, set-induced periodic $\mathcal{CH}_1$ controllers are considered.

The remainder of this chapter is structured as follows. Section 4.1.1 introduces in more detail the problem setting considered in this chapter. The fundamental concepts of finite-step contractive sets are presented in Section 4.2. In Section 4.3, the TMPC algorithm based on such sets is developed, and its implementation as a single LP is discussed in Section 4.4. Two numerical examples in Section 4.5 demonstrate the properties of the approach. Finally, some concluding remarks are given in Section 4.6.

4.1.1 Problem setting

For notational and computational simplicity, this chapter considers constrained polytopic LPV state-space (LPV-PSS) representations where only the state transition matrix A is parameter-varying, i.e., representations of the form

$$x(k+1) = A(\theta(k))x(k) + Bu(k). \tag{4.1}$$

Other than the simplified system representation (4.1), the problem setting is identical to that of Problem 3.1, i.e., it is desired to design an MPC such that the origin is a regionally asymptotically stable equilibrium for the closed-loop system. In this chapter, such a stabilizing MPC design is proposed based on finite-step contractive sets.

A few words on how the presented method can be extended to systems with a parameter-dependent input matrix are provided in Section 4.6.

4.2 Finite-step contraction

This section introduces the concept of finite-step contractive sets, which is central to the control approach developed in this chapter.

Definition 4.1. *Let $M \geq 1$ be an integer, let $\lambda \in [0,1)$, let $\mathbf{S}_M = \{S_0, \ldots, S_{M-1}\}$ be a sequence of PC-sets, and define $\sigma(k) := k \bmod M$. The PC-set $S_0 \subseteq \mathbb{X}$ is called controlled (M,λ)-contractive, if there exists a periodic $\mathcal{CH}_1$-control law $u(k) = K^{\mathrm{f}}_{\sigma(k)}(x(k),\theta(k))$ with $K^{\mathrm{f}}_i : S_i \times \Theta \to \mathbb{U}$, $i \in [0..M-1]$ such that:*

$$\forall i \in [0..M-2], \forall x \in S_i, \forall \theta \in \Theta : A(\theta)x + BK^{\mathrm{f}}_i(x,\theta) \in S_{i+1}, \tag{4.2a}$$

$$\forall x \in S_{M-1}, \forall \theta \in \Theta : A(\theta)x + BK^{\mathrm{f}}_{M-1}(x,\theta) \in \lambda S_0, \tag{4.2b}$$

$$\forall i \in [0..M-1] : \{0\} \subset S_i \subseteq \mathbb{X}. \tag{4.2c}$$

Observe that (4.2b) means that contraction of S_0 is achieved after M time instances. Thus, if $M = 1$, a controlled (M,λ)-contractive set corresponds to a "normal" controlled λ-contractive set (Definition 2.13). Sets satisfying these properties were used previously in reference governor design (Gilbert and Kolmanovsky 2002). They can also be interpreted as an instance of the positively invariant families from (Artstein and Raković 2011), which were employed in nominal MPC in (Raković et al. 2016a).

If S_0 is a polytope the periodic control laws in Definition 4.1 can always be selected as gain-scheduled vertex controllers, because – by convexity – existence of suitable controls on the vertices of $S_i \times \Theta$ implies existence of suitable controls over the full sets $S_i \times \Theta$, $i \in [0..M-1]$ (compare Lemma 2.4).

Finally, the closed-loop set-valued dynamics of (4.1) under a given local periodic controller $K^{\mathrm{f}}_{\sigma(\cdot)}(\cdot,\cdot)$ are

$$X(k+1) = G\left(k, X(k) \,\middle|\, K^{\mathrm{f}}\right) = \left\{A(\theta)x + BK^{\mathrm{f}}_{\sigma(k)}(x,\theta) \,\middle|\, x \in X(k),\ \theta \in \Theta\right\}. \tag{4.3}$$

The local uncertain closed-loop dynamics (4.3) are fundamentally different from those in nominally stabilizing MPC (Mayne et al. 2000). Hence, the construction of a suitable terminal cost is a challenge that is addressed in this chapter.

4.3 The finite-step TMPC algorithm

In this section, a tube-based LPV MPC algorithm using finite-step stabilizing conditions is developed. First, the concept of a *periodically parameterized tube (PpT)* is introduced in Section 4.3.1. Sufficient assumptions on this periodic parameterization are given, such that employing a finite-step contractive terminal set leads to a recursively feasible controller. In Section 4.3.2 terminal cost construction, based on a Lyapunov-type metric on proper C-sets, is presented and is shown to lead to a stable closed-loop system. In Section 4.3.3, the algorithm is summarized and its main properties are proven.

4.3.1 Periodically parameterized tubes

As a first necessary step, a periodically parameterized variant of the tubes from Definition 3.4 is introduced.

Definition 4.2. *Let* $(\mathbf{T}, k)$ *be a pair, with* $\mathbf{T}$ *being a tube according to Definition 3.4 which was synthesized at the time instant* $k \in \mathbb{N}$*. Introduce periodic parameter sets* $\mathbb{P}(k+i) = \mathbb{P}(\sigma(k+i)) = \mathbb{P}^{\mathrm{x}}(k+i) \times \mathbb{P}^{\mathrm{k}}(k+i)$ *with* $\sigma(\cdot)$ *as in Definition 4.1. A tube* $\mathbf{T}$ *is called a* periodically parameterized tube (PpT) *with period* M *if it satisfies the following conditions:*

(i) There is a function $P^{\mathrm{x}}(\cdot|k) : \mathbb{P}^{\mathrm{x}}(k) \to \mathbb{R}^{n_{\mathrm{x}}}$ *and, for all* $i \in [0..N]$*, there exists a parameter* $p^{\mathrm{x}}_i \in \mathbb{P}^{\mathrm{x}}(k+i)$ *such that* $X_i = P^{\mathrm{x}}\left(p^{\mathrm{x}}_i|k+i\right) = P^{\mathrm{x}}\left(p^{\mathrm{x}}_i|\sigma(k+i)\right)$.

(ii) There is a second-order function $P^{\mathrm{k}}(\cdot|k) : \mathbb{P}^{\mathrm{k}}(k) \to \left(X_i \times \Theta_i \to \mathbb{U}\right)$ *and, for all* $i \in [0..N-1]$*, there exists a parameter* $p^{\mathrm{k}}_i \in \mathbb{P}^{\mathrm{k}}(k+i)$ *such that* $K_i = P^{\mathrm{k}}\left(p^{\mathrm{k}}_i|k+i\right) = P^{\mathrm{k}}\left(p^{\mathrm{k}}_i|\sigma(k+i)\right)$.

Further, define the shorthand $P(p|k) = \left(P^{\mathrm{x}}\left(p^{\mathrm{x}}|k\right), P^{\mathrm{k}}\left(p^{\mathrm{k}}|k\right)\right)$ *where* $p = \left(p^{\mathrm{x}}, p^{\mathrm{k}}\right)$.

Let

$$\mathbf{X}^{\mathrm{f}} = \left\{X^{\mathrm{f}}_0, \ldots, X^{\mathrm{f}}_{M-1}\right\} \tag{4.4}$$

be a sequence of (M, λ)-contractive sets according to Definition 4.1. Based on this, the set $X^{\mathrm{f}}_{\sigma(k+N)}$ will be used as a periodically time-varying terminal set. Due to the periodic time-dependence in the parameterization of Definition 4.2 and of the terminal set, the tube synthesis problem that needs to be solved on-line as part of the control algorithm becomes time-dependent. Thus, in this chapter, time-dependent variants on (3.8) and (3.7) are considered. In this setting, the set of feasible tubes is now given by

$$\mathcal{T}_N\left(x,\mathbf{\Theta}|k\right) = \left\{\mathbf{T}\,\middle|\,\mathbf{T}\text{ satisfies Def. 4.2 with } X_0 = \{x\} \text{ and } X_N \subseteq X^{\mathrm{f}}_{\sigma(k+N)}\right\} \tag{4.5}$$

and the tube synthesis optimization problem becomes

$$V\left(x,\mathbf{\Theta}|k\right) = \min_{\mathbf{T}} J_N\left(\mathbf{T},\mathbf{\Theta}|k\right) \text{ subject to } \mathbf{T} \in \mathcal{T}_N\left(x,\mathbf{\Theta}|k\right). \tag{4.6}$$

In this chapter, the cost function $J_N(\mathbf{T},\mathbf{\Theta}|k)$ in (4.6) is similar to (3.9). The same type of stage cost (3.13) is used, but for the sake of simplicity, attention is focused on the ∞-norm and $c = 1$, i.e.,

$$\ell\left(X,K,\Theta\right) = \max_{(x,\theta)\in X\times\Theta}\left(\|Qx\| + \|RK(x,\theta)\|\right) = \max_{(x,\theta)\in X\times\Theta}\left(\|Qx\|_\infty + \|RK(x,\theta)\|_\infty\right). \tag{4.7}$$

Furthermore, because the terminal set of (4.5) is time-varying, a time-varying terminal cost will be adopted as well. This gives the time-dependent cost function

$$J_N\left(\mathbf{T},\mathbf{\Theta}|k\right) = \sum_{i=0}^{N-1}\ell\left(X_i,K_i,\Theta_i\right) + F_k\left(X_N\right). \tag{4.8}$$

The derivation of the function $F_k(\cdot)$ is the topic of Section 4.3.2. The remainder of the present section is concerned with establishing recursive feasibility of (4.6). To achieve this, the following assumption—an extended variant of (Brunner et al. 2013, Assumption 7)—on the tube parameterization is necessary.

Assumption 4.1. *The terminal set and the associated local controller are "homogeneously parameterizable" in* $\mathbb{P}(k+N)$*, i.e.,* $\forall(k,\gamma)\in\mathbb{N}\times[0,1] : \exists p_{\mathrm{f}} \in \mathbb{P}(k+N)$ *such that* $P\left(p_{\mathrm{f}}|k+N\right) = \gamma\left(X^{\mathrm{f}}_{\sigma(k+N)}, K^{\mathrm{f}}_{\sigma(k+N)}\right)$.

Based on Assumption 4.1, recursive feasibility of the tube synthesis (4.6) can be shown. Later, in Section 4.4, a concrete parameterization that satisfies this assumption is given.

Proposition 4.2. *Let* $\mathbf{S}_M$ *be a sequence of controlled* (M,λ)*-contractive sets for* (4.1) *according to Definition 4.1, and let the associated closed-loop dynamics* $G(\cdot,\cdot|K^{\mathrm{f}})$ *be as in* (4.3)*. Let* $\mathbf{\Theta} \subseteq \Theta^N$, $\mathbf{\Theta}^+ \subseteq \Theta^N$ *be two sequences related as* $\mathbf{\Theta}^+ \sqsubseteq \mathbf{\Theta}$ *and with* $\Theta_0 = \theta$*. Suppose that Assumption 4.1 is satisfied. Then* (4.6) *is recursively feasible in the sense that if* $\exists\mathbf{T} \in \mathcal{T}_N(x,\mathbf{\Theta}|k)$*, then* $\exists\mathbf{T}^+ \in \mathcal{T}_N(x^+,\mathbf{\Theta}^+|k+1)$ *where* $x^+ = A(\theta)x + BK_0(x,\theta)$*.*

Proof. Suppose that (4.6) is feasible at time k and let

$$\mathbf{T} = \left(\{X_0,\ldots,X_N\},\{K_0,\ldots,K_{N-1}\}\right)$$

be a tube corresponding to a solution of (4.6) at time k. By construction, $X_0 = \{x\}$ and $\exists \gamma \in [0,1] : X_N \subseteq \gamma X^{\mathrm{f}}_{\sigma(k+N)}$. Note that $\gamma = 1$ would be sufficient here, but keeping it variable simplifies the subsequent stability proof of Theorem 4.1. After applying K_0 to the system, a feasible tube at time $k+1$ can be explicitly given by the definition of the terminal set and because $\mathbf{\Theta}^+ \sqsubseteq \mathbf{\Theta}$:

$$\mathbf{T}^+ = \left(\left\{X_0^+, X_2, \ldots, X_{N-1}, \gamma X^{\mathrm{f}}_{\sigma(k+N)}, \gamma G\left(k+N, X^{\mathrm{f}}_{\sigma(k+N)}\middle| K^{\mathrm{f}}\right)\right\},\ \left\{K_1, \ldots, K_{N-1}, K^{\mathrm{f}}_{\sigma(k+N)}\right\}\right),$$

where $X_0^+ = \{x^+\} \subset X_1$, which implies feasibility of $K_0^+ = K_1$. Since (4.6) only optimizes over finitely parameterized sets and controllers, there must exist parameters $\left(p_{\mathrm{f}}, p_{\mathrm{f}}^+\right) \in \mathbb{P}(k+N) \times \mathbb{P}(k+N+1)$ such that

$$\begin{aligned} P^{\mathrm{x}}\left(p_{\mathrm{f}}^{\mathrm{x}}\middle| k+N\right) &= \gamma X^{\mathrm{f}}_{\sigma(k+N)}, \\ P^{\mathrm{k}}\left(p_{\mathrm{f}}^{\mathrm{k}}\middle| k+N\right) &= K^{\mathrm{f}}_{\sigma(k+N)}, \\ P^{\mathrm{x}}\left(p_{\mathrm{f}}^{\mathrm{x}+}\middle| k+N+1\right) &= \gamma G\left(k+N, X^{\mathrm{f}}_{\sigma(k+N)}\middle| K^{\mathrm{f}}\right). \end{aligned}$$

Because $\gamma G\left(k+N, X^{\mathrm{f}}_{\sigma(k+N)}\middle| K^{\mathrm{f}}\right) = \gamma X^{\mathrm{f}}_{\sigma(k+N+1)}$ by Definition 4.1, this is all guaranteed through Assumption 4.1. Therefore it follows that (4.6) is feasible at time $k+1$. □

Now that recursive feasibility has been established under appropriate conditions on the tube parameterization, it is possible to proceed with the derivation of conditions under which a controller based on the tube synthesis (4.6) is asymptotically stabilizing. This requires the construction of a terminal cost, which is the topic of the next subsection.

4.3.2 Terminal cost construction

To guarantee stability of the MPC scheme, an appropriate terminal cost has to be constructed. The first step towards this end is to find a Lyapunov-type function which is monotonically decreasing along the set-valued trajectories of (4.3). In some sense, this approach is analogous to the construction in the nominal case (Theorem 2.1), where a terminal cost is also constructed based on a certain local Lyapunov function.

To derive the required Lyapunov-like function, we need the following finite-step decrease property of the function $\Psi_{S_i}(\cdot)$. The abbreviated notations $\psi_i(\cdot) := \psi_{S_i}$ and $\Psi_i(\cdot) := \Psi_{S_i}(\cdot)$ are used in the sequel.

Lemma 4.1. *Let $\mathbf{S}_M$ be a sequence of controlled (M, λ)-contractive sets for* (4.1) *in the sense of Definition 4.1. Define the resulting closed-loop dynamics $G(\cdot, \cdot | K^{\mathrm{f}})$ as in* (4.3). *Then, $\Psi_{\sigma(k)}(\cdot)$ satisfies for all $k \in \mathbb{N}$ and for all $X \subseteq S_{\sigma(k)}$ the following relation:*

$$\Psi_{\sigma(k+1)}\left(G\left(k, X\middle| K^{\mathrm{f}}\right)\right) \leq \begin{cases} \Psi_{\sigma(k)}(X), & \sigma(k) \in [0..M-2], \\ \lambda \Psi_{\sigma(k)}(X), & \sigma(k) = M-1. \end{cases}$$

Proof. Let ∂S denote the boundary of a set $S \subset \mathbb{R}^n$. By Definition 4.1, $\forall \bar{x} \in \partial S_{\sigma(k)}$:

$$G\left(k, \{\bar{x}\} \middle| K^{\mathrm{f}}\right) \in \begin{cases} S_{\sigma(k+1)}, & \sigma(k) \in [0..M-2], \\ \lambda S_{\sigma(k+1)}, & \sigma(k) = M-1. \end{cases}$$

Now let $x \in S_{\sigma(k)}$. By definition of the gauge function, it holds that $x \in \psi_{\sigma(k)}(x)\partial S_{\sigma(k)}$ (Schneider 2013a). Thus, $\exists \bar{x} \in \partial S_{\sigma(k)} : x = \psi_{\sigma(k)}(x)\bar{x}$. By homogeneity it follows directly that

$$G\left(k, \{x\} \middle| K^{\mathrm{f}}\right) = G\left(k, \{\psi_{\sigma(k)}(x)\bar{x}\} \middle| K^{\mathrm{f}}\right) = \psi_{\sigma(k)}(x) G\left(k, \{\bar{x}\} \middle| K^{\mathrm{f}}\right)$$

and therefore $\forall x \in S_{\sigma(k)}$:

$$G\left(k, \{x\} \middle| K^{\mathrm{f}}\right) \in \begin{cases} \psi_{\sigma(k)}(x) S_{\sigma(k+1)}, & \sigma(k) \in [0..M-2], \\ \lambda \psi_{\sigma(k)}(x) S_{\sigma(k+1)}, & \sigma(k) = M-1. \end{cases}$$

From the above we get that $\forall X \subseteq S_{\sigma(k)}$:

$$G\left(k, X \middle| K^{\mathrm{f}}\right) \subseteq \begin{cases} \sup_{x \in X} \psi_{\sigma(k)}(x) S_{\sigma(k+1)}, & \sigma(k) \in [0..M-2], \\ \lambda \sup_{x \in X} \psi_{\sigma(k)}(x) S_{\sigma(k+1)}, & \sigma(k) = M-1 \end{cases}$$

and by applying Definition 2.2 the desired property follows. □

The above lemma can be exploited to construct a Lyapunov-type function enabling the computation of a stabilizing terminal cost for (4.6). In (Geiselhart et al. 2014, Theorem 20), it was shown that it is possible to construct a "true" (i.e., monotonically decreasing along trajectories) Lyapunov function based on a function that satisfies a finite-step decrease property like Lemma 4.1. The result therein is, however, not directly applicable to set-valued dynamics. Therefore, the following proposition applies a suitably modified version of the construction of (Geiselhart et al. 2014, Theorem 20) to sequences of sets, yielding the desired Lyapunov-type function.

Proposition 4.3. *Suppose that the conditions from Lemma 4.1 are satisfied. Then, the function*

$$W(k, X) := (M + (\lambda - 1)\sigma(k))\, \Psi_{\sigma(k)}(X)$$

is a Lyapunov-type function for the dynamics (4.3), i.e., it satisfies the following properties:

(i) There exist $\underline{w}, \overline{w} \in \mathcal{K}_\infty$ *such that for all* $k \in \mathbb{N}$ *and for all compact and convex sets* $X \subset \mathbb{R}^n$, $\underline{w}\left(d_H^0(X)\right) \leq W(k, X) \leq \overline{w}\left(d_H^0(X)\right)$ *holds,*

(ii) There exists $\varrho(k) : \mathbb{N} \to [0, 1)$ *such that for all* $k \in \mathbb{N}$ *and for all* $X \subseteq S_{\sigma(k)}$, $W\left(k+1, G\left(k, X \middle| K^{\mathrm{f}}\right)\right) \leq \varrho(k) W(k, X)$,

(iii) There exists $\varrho \in [0, 1)$ *such that for all* $k \in \mathbb{N}$ *and for all* $X \subseteq S_{\sigma(k)}$, $W\left(k+1, G\left(k, X \middle| K^{\mathrm{f}}\right)\right) \leq \varrho W(k, X)$.

Proof. Since $(M + (\lambda - 1)\sigma(k))$ is a positive number for all $k \in \mathbb{N}$, it follows from Lemma 2.1 that $\exists \underline{w}^i, \overline{w}^i \in \mathcal{K}_\infty$ for each $i \in [0..M-1]$ such that for all compact and convex $X \subset \mathbb{R}^n$, $\underline{w}^{\sigma(k)}\left(d_H^0(X)\right) \leq W(k,X) \leq \overline{w}^{\sigma(k)}\left(d_H^0(X)\right)$. As the minimum- and maximum over a finite set of $\mathcal{K}_\infty$-functions is again $\mathcal{K}_\infty$ (Kellett 2014), statement (i) holds with $\underline{w}(\xi) = \min_{i\in[0..M-1]} \underline{w}^i(\xi)$ and $\overline{w}(\xi) = \max_{i\in[0..M-1]} \overline{w}^i(\xi)$. For the proof of (ii), consider first that k is such that $\sigma(k) \in [0..M-2]$. Then by Lemma 4.1, $\Psi_{\sigma(k+1)}\left(G\left(k, X \middle| K^{\mathrm{f}}\right)\right) \leq \Psi_{\sigma(k)}(X)$, and therefore

$$\begin{aligned} W\left(k+1, G\left(k, X \middle| K^{\mathrm{f}}\right)\right) &= (M + (\lambda-1)\,\sigma(k+1))\,\Psi_{\sigma(k+1)}\left(G\left(k, X \middle| K^{\mathrm{f}}\right)\right) \\ &\leq (M + (\lambda-1)\,\sigma(k+1))\,\Psi_{\sigma(k)}(X) \\ &= \frac{(M + (\lambda-1)\,\sigma(k+1))}{(M + (\lambda-1)\,\sigma(k))} W(k,X). \end{aligned}$$

Next, let k be such that $\sigma(k) = M-1$. Again by Lemma 4.1, $\Psi_{\sigma(k+1)}\left(G\left(k, X \middle| K^{\mathrm{f}}\right)\right) \leq \lambda \Psi_{\sigma(k)}(X)$, so

$$\begin{aligned} W\left(k+1, G\left(k, X \middle| K^{\mathrm{f}}\right)\right) &= M\Psi_0\left(G\left(M-1, X \middle| K^{\mathrm{f}}\right)\right) \\ &\leq \lambda M \Psi_{M-1}(X) \\ &= \frac{\lambda M}{\lambda(M-1)+1} W(k,X). \end{aligned}$$

Hence, statement (ii) is satisfied with

$$\varrho(k) = \begin{cases} \frac{(M+(\lambda-1)\sigma(k+1))}{(M+(\lambda-1)\sigma(k))}, & \sigma(k) \in [0..M-2], \\ \frac{\lambda M}{\lambda(M-1)+1}, & \sigma(k) = M-1, \end{cases}$$

and (iii) follows with $\varrho = \max_{k\in\mathbb{N}} \varrho(k) = \varrho(0)$. □

In Proposition 4.3, a suitable Lyapunov-type function has been obtained that can serve as a basis for the construction of a terminal cost. The remaining step is to compute a scaling of $W(\cdot,\cdot)$, such that its rate of decrease can be related to the stage cost function, similar to Condition (iii) of Theorem 2.1. For all $i \in [0..M-1]$, let

$$\bar{\ell}_i = \max_{(x,u)\in S_i \times \mathbb{U}} (\|Qx\| + \|Ru\|) \text{ s.t. } \forall \theta \in \Theta : \begin{cases} A(\theta)x + Bu \in S_{i+1}, & i \in [0..M-2], \\ A(\theta)x + Bu \in \lambda S_0, & i = M-1. \end{cases} \tag{4.9}$$

From Proposition 4.3, the next Corollary follows directly.

Corollary 4.1. *Let* $\bar{\ell}_i$ *be as in* (4.9) *and define* $\bar{\ell} = \max_{i\in[0..M-1]} \bar{\ell}_i$. *Define* $W(k,X)$ *and* ϱ *as in Proposition 4.3. Then the function*

$$\overline{W}(k,X) := \frac{\bar{\ell}}{1-\varrho} W(k,X) \tag{4.10}$$

satisfies $\forall k \in \mathbb{N}$ *and* $\forall X \subseteq S_{\sigma(k)}$ *that*

$$\overline{W}\left(k+1, G\left(k, X \middle| K^{\mathrm{f}}\right)\right) - \overline{W}\left(k, X\right) \leq -\bar{\ell} W\left(k, X\right).$$

Furthermore, $\forall k \in \mathbb{N} : 1 \leq W\left(k+N, X^{\mathrm{f}}_{\sigma(k+N)}\right) \leq M$.

With this result, the desired terminal cost which leads to closed-loop asymptotic stability is constructed. In the next subsection, it is proven that this indeed leads to an MPC algorithm that is asymptotically stabilizing.

4.3.3 Main result

The TMPC algorithm based on finite-step contractive sets is given in Algorithm 4.1. It is a straightforward modification of the receding-horizon control (RHC) approach of Algorithm 3.1, as the only difference is in the introduced time-dependence of the set $\mathcal{T}_N(\cdot, \cdot|k)$.

Algorithm 4.1 Finite-step receding-horizon TMPC algorithm.

Require: $N \in [1..\infty)$
$\mathbf{\Theta}_{-1} \leftarrow \Theta^N$
$k \leftarrow 0$
loop
  Construct $\mathbf{\Theta}_k = \left\{\{\theta(k)\}, \left\{\Theta_{i|k}\right\}_{i=1}^{N-1}\right\} \subseteq \Theta^N$ such that $\mathbf{\Theta}_k \sqsubseteq \mathbf{\Theta}_{k-1}$
  if $\mathcal{T}_N\left(x(k), \mathbf{\Theta}_k|k\right) \neq \emptyset$ **then**
    Solve (4.6) to obtain $\mathbf{T}^\star \in \mathcal{T}_N\left(x(k), \mathbf{\Theta}_k|k\right)$
    Apply $u(k) = K_0^\star\left(x(k), \theta(k)\right) = u_0^\star$ to the system (4.1)
    $k \leftarrow k+1$
  else
    abort
  end if
end loop

Based on the developed results, the main result of this chapter concerning the properties of Algorithm 4.1, can be stated.

Theorem 4.1. *Suppose that the conditions of Proposition 4.2 are satisfied. Let* $F_k(\cdot) := \overline{W}\left(k+N, \cdot\right)$ *according to* (4.10)*. Then the TMPC defined by* (4.6) *asymptotically stabilizes the origin of the closed-loop system.*

Proof. Denote $G_{\mathrm{f}|k}(\cdot) := G\left(k+N, \cdot|K^{\mathrm{f}}\right)$ according to (4.3). Consider the solution $\mathbf{T}$ and the feasible, but not necessarily optimal, solution $\mathbf{T}^+$ constructed in the proof of Proposition 4.2. By definition of $F_k(\cdot)$, it follows that we can take $\gamma = \Psi_{\sigma(k+N)}\left(X_N\right)$. Substitute the solutions $\mathbf{T}$ and $\mathbf{T}^+$ in the cost function of (4.6) and compute the difference between the value functions at time k and time $k+1$ to obtain

$$\begin{aligned}\Delta V_k &= V\left(x^+, \mathbf{\Theta}^+ \middle| k+1\right) - V\left(x, \mathbf{\Theta} | k\right) \\ &\leq \ell\left(X_0^+, K_1\right) + \gamma \ell\left(X^{\mathrm{f}}_{\sigma(k+N)}, K^{\mathrm{f}}_{\sigma(k+N)}\right) + \gamma F_{k+1}\left(G_{\mathrm{f}|k}\left(X^{\mathrm{f}}_{\sigma(k+N)}\right)\right) - F_k\left(X_N\right) \\ &\quad + \sum_{i=2}^{N-1} \ell\left(X_i, K_i\right) - \sum_{i=0}^{N-1} \ell\left(X_i, K_i\right).\end{aligned}$$

Observe that $X_0^+ = \{x^+\} \subset X_1$, so $\ell\left(X_0^+, K_1\right) \leq \ell\left(X_1, K_1\right)$ and therefore

$$\begin{aligned}\Delta V_k &\leq \sum_{i=1}^{N-1} \ell\left(X_i, K_i\right) - \sum_{i=0}^{N-1} \ell\left(X_i, K_i\right) + \gamma \ell\left(X^{\mathrm{f}}_{\sigma(k+N)}, K^{\mathrm{f}}_{\sigma(k+N)}\right) \\ &\quad + \gamma F_{k+1}\left(G_{\mathrm{f}|k}\left(X^{\mathrm{f}}_{\sigma(k+N)}\right)\right) - F_k\left(X_N\right) \\ &= -\ell\left(X_0, K_0\right) + \gamma \ell\left(X^{\mathrm{f}}_{\sigma(k+N)}, K^{\mathrm{f}}_{\sigma(k+N)}\right) + \gamma F_{k+1}\left(G_{\mathrm{f}|k}\left(X^{\mathrm{f}}_{\sigma(k+N)}\right)\right) - F_k\left(X_N\right) \\ &\leq -\ell\left(X_0, K_0\right) + \gamma \bar{\ell} + \gamma F_{k+1}\left(G_{\mathrm{f}|k}\left(X^{\mathrm{f}}_{\sigma(k+N)}\right)\right) - F_k\left(X_N\right)\end{aligned}$$

where the last inequality follows from the definition of $\bar{\ell}$ in Corollary 4.1. Since $X_N \subseteq \gamma X^{\mathrm{f}}_{\sigma(k+N)}$, according to the definition of the terminal cost

$$\begin{aligned}F_k\left(X_N\right) &= \frac{\bar{\ell}}{1-\varrho}\left(M + (\lambda - 1)\sigma(k+N)\right) \Psi_{\sigma(k)}\left(X_N\right) \\ &= \gamma \frac{\bar{\ell}}{1-\varrho}\left(M + (\lambda - 1)\sigma(k+N)\right) \Psi_{\sigma(k)}\left(X^{\mathrm{f}}_{\sigma(k+N)}\right) \\ &= \gamma F_k\left(X^{\mathrm{f}}_{\sigma(k+N)}\right).\end{aligned}$$

Hence,

$$\begin{aligned}\Delta V_k &\leq -\ell\left(X_0, K_0\right) + \gamma\left(\bar{\ell} + F_{k+1}\left(G_{\mathrm{f}|k}\left(X^{\mathrm{f}}_{\sigma(k+N)}\right)\right) - F_k\left(X^{\mathrm{f}}_{\sigma(k+N)}\right)\right) \\ &\leq -\ell\left(X_0, K_0\right) + \gamma\left(\bar{\ell} - \bar{\ell} W\left(k+N, X^{\mathrm{f}}_{\sigma(k+N)}\right)\right) \\ &\leq -\ell\left(X_0, K_0\right) \\ &\leq -\underline{\ell}\left(\|x\|\right)\end{aligned}$$

where the second and third inequalities follow from Corollary 4.1, and the last inequality from Proposition 3.1.(ii). The fact that $V\left(x, \mathbf{\Theta} | k\right)$ is monotonically decreasing with rate $\underline{\ell}\left(\|x\|\right)$ is, in conjunction with the bounds of Proposition 3.2, sufficient to conclude that $V(\cdot, \cdot|\cdot)$ is a (time-varying) Lyapunov function. Hence, asymptotic stability of the controlled system follows (Aeyels and Peuteman 1998, Theorem 2). □

With the completion of Theorem 4.1 it has been shown that it is possible to formulate a stabilizing tube-based MPC algorithm for LPV systems based on a terminal set that is finite-step contractive. Overall, the proof of Theorem 4.1 is quite similar to the standard type of stability proof employed in nominal MPC (Theorem 2.1). To enable the use of this standard proof technique, it was however necessary to derive a new terminal cost function.

In the next section, the practical implementation of the obtained MPC algorithm is discussed.

4.4 Implementation details

In this section, practical implementation of the algorithm presented in Section 4.3 is discussed. In Section 4.4.1, a concrete parameterization—a periodic variant on the homothetic parameterization from Section 3.5—is proposed and is shown to satisfy all the required assumptions. Using this parameterization, as shown in Section 4.4.2, the tube synthesis problem can be formulated as a single linear program with a number of variables and constraints that is a linear function of the prediction horizon N. Finally, in Section 4.4.3, a short discussion on how to compute finite-step contractive sets for LPV-PSS representations is provided.

4.4.1 Periodically homothetic parameterization

In this section, it is shown how the general results presented previously can be used in practice by developing a specific parameterization that satisfies Assumption 4.1. To satisfy Assumption 4.1, consider a "periodic" variant on the homothetic parameterization (Brunner et al. 2013; Langson et al. 2004; Raković et al. 2012a) by parameterizing the tube cross sections as

$$X_i = z_i \oplus \alpha_i X^{\mathrm{f}}_{\sigma(k+i)} \tag{4.11}$$

such that for a tube synthesized at time k, each cross section X_i is homothetic to $X^{\mathrm{f}}_{\sigma(k+i)}$ with center z_i and scaling α_i. The parameters $z_i \in \mathbb{R}^{n_{\mathrm{x}}}$ and $\alpha_i \in \mathbb{R}_+$ are optimized on-line. In the language of Definition 4.2, the parameterization (4.11) corresponds to the choices

$$\begin{aligned} p^{\mathrm{x}}_i &= (\alpha_i, z_i) \in \mathbb{P}^{\mathrm{x}}(k+i), \\ \mathbb{P}^{\mathrm{x}}(k) &= \mathbb{R}_+ \times \mathbb{R}^{n_{\mathrm{x}}}, \\ P^{\mathrm{x}}\left(p^{\mathrm{x}}_i \middle| k+i\right) &= z_i \oplus \alpha_i X^{\mathrm{f}}_{\sigma(k+i)}. \end{aligned} \tag{4.12}$$

The sets X^{f}_i, $i \in [0..M-1]$ are the same as in (4.4) and they are polytopes represented by the convex hull of t_i vertices as

$$\forall i \in [0..M-1] : X^{\mathrm{f}}_i = \mathrm{convh}\left\{\bar{s}^1_i, \ldots, \bar{s}^{t_i}_i\right\}. \tag{4.13}$$

The associated control laws are parameterized as gain-scheduled vertex controllers. In terms of Definition 4.2,

$$\begin{aligned} p^{\mathrm{k}}_i &= \left(u^{(1,1)}_i, \ldots, u^{(t_{\sigma(k+i)},1)}_i, \ldots, u^{(t_{\sigma(k+i)},q_\theta)}_i\right) \in \mathbb{P}^{\mathrm{k}}(k+i), \\ \mathbb{P}^{\mathrm{k}}(k) &= \mathbb{R}^{q_\theta t_{\sigma(k)} n_{\mathrm{u}}}, \\ P^{\mathrm{k}}\left(p^{\mathrm{k}}_i \middle| k+i\right) &= \mathrm{vertpol}\left(p^{\mathrm{k}}_i \middle| X_i, \Theta_i\right). \end{aligned} \tag{4.14}$$

With the parameterization discussed above, the infinity-norm based stage cost (4.7) for a tube synthesized at time k is equivalent to

$$\ell(X_i, K_i, \Theta_i) = \max_{j\in[1..t_{\sigma(k+i)}], l\in[1..q]} \left(\|Q\bar{x}_i^j\| + \|Ru_i^{(j,l)}\|\right) \tag{4.15}$$

where $\bar{x}_i^j = z_i + \alpha_i \bar{s}^j_{\sigma(k+i)}$.

The scaling $\bar{\ell}$ in Corollary 4.1 can be efficiently computed as follows. For all $(i,l) \in [0..M-1] \times [1..q]$ and all corresponding $j \in [1..t_i]$, compute the control actions

$$u_{\mathrm{f}}^{(i,j,l)} = \arg\min_{u\in\mathbb{U}} \left(\|Q\bar{s}_i^j\| + \|Ru\|\right) \text{ s.t. } \begin{cases} A(\bar{\theta}^l)\bar{s}_i^j + Bu \in X_{i+1}^{\mathrm{f}}, & i \in [0..M-2], \\ A(\bar{\theta}^l)\bar{s}_i^j + Bu \in \lambda X_0^{\mathrm{f}}, & i = M-1, \end{cases}$$

to obtain a local periodic vertex control law which is feasible and asymptotically stabilizing on $\mathbf{X}_M^{\mathrm{f}}$. Then, the constants $\bar{\ell}_i$ are directly found by computing

$$\forall i \in [0..M-1] : \bar{\ell}_i = \max_{j\in[1..t_i], l\in[1..q]} \left(\|Q\bar{s}_i^j\| + \|Ru_{\mathrm{f}}^{(i,j,l)}\|\right). \tag{4.16}$$

It can be verified that the periodically homothetic parameterization described in this section satisfies Assumption 4.1, and hence, the main result of this subsection follows.

Corollary 4.2. *The LPV TMPC algorithm with tube parameterization* (4.12) *and* (4.14) *is recursively feasible and asymptotically stabilizes the origin of the closed-loop system (i.e., "stable" in the sense of Definition 3.5).*

4.4.2 Linear programming formulation

In this section, it is shown how the tube synthesis problem (4.6) can be formulated as a single LP. The number of variables and constraints in this LP is a linear function of the prediction horizon N. Hence, the main result of this subsection is to establish that the algorithm of Section 4.3 with the parameterization of Section 4.4.1 is computationally tractable in the sense of Definition 3.6.

As they are polytopic PC-sets, each set X_i^{f} in (4.13) can be equivalently represented in a half-space form with r_i hyperplanes. That is, for all $i \in [0..M-1]$, there exists a matrix $H_i \in \mathbb{R}^{r_i \times n_{\mathrm{x}}}$ such that

$$X_i^{\mathrm{f}} = \mathrm{convh}\left\{\bar{s}_i^1, \dots, \bar{s}_i^{t_i}\right\} = \{x \in \mathbb{R}^{n_{\mathrm{x}}} | H_i x \leq 1\}.$$

Half-space representations of $\mathbb{X}$ and $\mathbb{U}$ are also assumed to be available with r_{x} and r_{u} hyperplanes, respectively (see Section 2.2.3 and specifically Equation 2.8). In agreement with Assumption 2.1 and according to the discussion below Definition 3.1, let

$$q_i = \begin{cases} 1, & i = 0, \\ q, & \text{otherwise.} \end{cases}$$

Furthermore, in what follows, introduce the shorthand notation

$$\varsigma_i = \sigma(k+i)$$

with $\sigma(\cdot)$ as in Definition 4.1. A step-by-step construction of the optimization problem (4.6) and the corresponding number of variables and constraints is now provided.

Cross section parameterization According to (4.12), each tube cross section is characterized by a scaling $\alpha_i \in \mathbb{R}_+$ and a center $z_i \in \mathbb{R}^{n_\mathrm{x}}$. As there are $N+1$ cross sections in the tube (counting X_0 up to and including X_N), this introduces $(N+1)(1+n_\mathrm{x})$ decision variables. Furthermore, the scalings must be non-negative, leading to the $N+1$ inequality constraints

$$\forall i \in [0..N]: \ \alpha_i \geq 0. \tag{4.17}$$

Control parameterization At each prediction time instant $i \in [0..N-1]$, the parameterization (4.14) consists of $t_{\varsigma_i} q_i$ control actions, i.e., $n_\mathrm{u} t_{\varsigma_i} q_i$ decision variables. Thus the control parameterization introduces a total of $n_\mathrm{u} \sum_{i=0}^{N-1} t_{\varsigma_i} q_i$ variables. However, because of the constraint $X_0 = \{x\}$, all control actions for $i = 0$ are equal. Thus the number of introduced decision variables can be slightly reduced to $n_\mathrm{u} + n_\mathrm{u} \sum_{i=1}^{N-1} t_{\varsigma_i} q_i$.

Initial condition constraint The constraint $X_0 = \{x\}$ in (4.5) is implemented by $n_\mathrm{x} + 1$ equality constraints

$$\begin{aligned} z_0 &= x, \\ \alpha_0 &= 0. \end{aligned} \tag{4.18}$$

Tube dynamics constraints The transition constraints $\mathcal{I}(X_i, \Theta_i | K_i) \subseteq X_{i+1}$ from Definition 3.4 are equivalent to the following $\sum_{i=0}^{N-1} r_{\varsigma_{i+1}} t_{\varsigma_i} q_i$ linear inequalities

$$\forall i \in [0..N-1]: \ \forall (j,l) \in [1..t_{\varsigma_i}] \times [1..q_i]: \ H_{\varsigma_{i+1}} \left(A\left(\bar{\theta}_i^l\right) \left(z_i + \alpha_i \bar{s}_{\varsigma_i}^j\right) + B u_i^{(j,l)} \right) \leq 1. \tag{4.19}$$

State constraints The state constraints $\mathcal{I}(X_i, \Theta_i | K_i) \subseteq \mathbb{X}$ from Definition 3.4 are equivalent to the $r_\mathrm{x} \sum_{i=0}^{N-1} t_{\varsigma_i} q_i$ linear inequalities

$$\forall i \in [0..N-1]: \ \forall (j,l) \in [1..t_{\varsigma_i}] \times [1..q_i]: \ H_\mathrm{x} \left(A\left(\bar{\theta}_i^l\right) \left(z_i + \alpha_i \bar{s}_{\varsigma_i}^j\right) + B u_i^{(j,l)} \right) \leq 1. \tag{4.20}$$

Input constraints The input constraints are realized by the following $r_\mathrm{u} \sum_{i=0}^{N-1} t_{\varsigma_i} q_i$ linear inequalities

$$\forall i \in [0..N-1]: \ \forall (j,l) \in [1..t_{\varsigma_i}] \times [1..q_i]: \ H_\mathrm{u} u_i^{(j,l)} \leq 1. \tag{4.21}$$

Terminal constraint The terminal constraint $X_N \subseteq X^{\mathrm{f}}_{\varsigma_N}$ from (4.5) is equivalent to the following $r_{\varsigma_N} t_{\varsigma_N} + 2$ linear inequalities

$$\begin{aligned} \forall j \in [1..t_{\varsigma_N}]:\ H_{\varsigma_N}\left(z_N + \alpha_N \bar{s}^j_{\varsigma_N}\right) &\leq \gamma, \\ 0 \leq \gamma &\leq 1, \end{aligned} \tag{4.22}$$

where $\gamma \in \mathbb{R}_+$ is a decision variable.

Stage cost The stage cost (4.7), which is equivalent to (4.15), can be implemented by introducing non-negative slack variables (μ, ν, ζ). Recall that in this chapter, the ∞-norm is used. Thus, $\ell(X_i, K_i, \Theta_i) \leq \zeta_i$ is equivalent to $\forall i \in [0..N-1]$:

$$\begin{aligned} \forall (j,l) \in [1..t_{\varsigma_i}] \times [1..q_i] :& -\nu_i^{(j,l)} \leq R u_i^{(j,l)} \leq \nu_i^{(j,l)},\ \nu_i^{(j,l)} \geq 0 \\ \forall j \in [1..t_{\varsigma_i}] :& -\mu_i^j \leq Q\left(z_i + \alpha_i \bar{s}^j_{\varsigma_i}\right) \leq \mu_i^j,\ \mu_i^j \geq 0 \\ \forall (j,l) \in [1..t_{\varsigma_i}] \times [1..q_i] :& \nu_i^{(j,l)} + \mu_i^j \leq \zeta_i, \end{aligned} \tag{4.23}$$

which corresponds to a total number of $\sum_{i=0}^{N-1} \left(t_{\varsigma_i} q_i (n_{\mathrm{r}} + 1) + q_i \left(n_{\mathrm{q}} + 1\right) + t_{\varsigma_i} q_i\right)$ inequality constraints and introduces $\sum_{i=0}^{N-1} \left(t_{\varsigma_i} q_i + t_{\varsigma_i} + 1\right)$ decision (slack) variables.

Terminal cost The decision variable $\gamma \in \mathbb{R}_+$ as introduced in (4.22) corresponds to $\Psi_{\varsigma_N}(X_N)$. Therefore,

$$F_k(X_N) = \overline{W}(k+N, X_N) = \frac{\bar{\ell}}{1-\varrho}\left(M + (\lambda - 1)\varsigma_N\right)\gamma. \tag{4.24}$$

The total LP Given the above constructions, the total LP to be solved-online becomes

$$\begin{aligned} V(x, \boldsymbol{\Theta} | k) = \min_{\mathbf{d}} & \frac{\bar{\ell}}{1-\varrho}\left(M + (\lambda - 1)\varsigma_N\right)\gamma + \sum_{i=0} \zeta_i \\ & \text{subject to (4.17)–(4.24)} \end{aligned} \tag{4.25}$$

with the decision variable $\mathbf{d}$ being the collection of all the variables introduced above, i.e.,

$$\begin{aligned} \mathbf{d} = \big(&\alpha_0, \ldots, \alpha_N, z_0, \ldots, z_N, u_0^{(1,1)}, \ldots, u_{N-1}^{t(\varsigma_{N-1}, q_{N-1})}, \\ &\nu_0^{(1,1)}, \ldots, \nu_{N-1}^{t(\varsigma_{N-1}, q_{N-1})}, \mu_0^1, \ldots, \mu_{N-1}^{t(\varsigma_{N-1}}, \\ &\zeta_0, \ldots, \zeta_{N-1}, \gamma\big). \end{aligned}$$

Summing up the number of decision variables n_d, the number of equality constraints n_eq and the number of inequality constraints n_ineq for each sub-part of the problem yields

$$\begin{aligned} n_\mathrm{d} &= (N+1)(1+n_\mathrm{x}) + n_\mathrm{u} + n_\mathrm{u} \sum_{i=1}^{N-1} t_{\varsigma i} q_i + 1 + \sum_{i=0}^{N-1} \left(t_{\varsigma i} q_i + t_{\varsigma_i} + 1 \right) = O(N), \\ n_\mathrm{eq} &= n_\mathrm{x} + 1 = O(1), \\ n_\mathrm{ineq} &= N + 1 + \sum_{i=0}^{N-1} r_{\varsigma i+1} t_{\varsigma i} q_i + r_\mathrm{x} \sum_{i=0}^{N-1} t_{\varsigma i} q_i + r_\mathrm{u} \sum_{i=0}^{N-1} t_{\varsigma i} q_i + r_{\varsigma N} t_{\varsigma N} + 2 \\ &\quad + \sum_{i=0}^{N-1} \left(t_{\varsigma_i} q_i \left(n_\mathrm{r} + 1 \right) + q_i \left(n_\mathrm{q} + 1 \right) + t_{\varsigma_i} q_i \right) = O(N). \end{aligned}$$

Because the numbers of variables and constraints in (4.25) are linearly bounded in N, the problem is tractable:

Corollary 4.3. *The tube synthesis problem* (4.6)*—which is equivalent to* (4.25)*—is computationally tractable in the sense of Definition 3.6.*

Alternative formulations of the LP avoiding the computation of the hyperplane representations of X_i^f, $i \in [0..M-1]$, can be constructed. This involves the use of additional decision variables corresponding to convex multipliers to verify all set inclusions. The number of decision variables and constraints will still be linear in N, but the exact complexity of this approach is not studied in the context of this chapter. An implementation entirely based on vertex representations, but without using finite-step contractive sets, is discussed in Chapter 6.

With regards to the scaling of the number of variables and constraints in other parameters besides N, the following observations can be made:

- $n_\mathrm{d} = O(q_\theta)$ and $n_\mathrm{ineq} = O(q_\theta)$, i.e., the problem size scales linearly in the number of vertices q of the scheduling sets. If there are n_θ scheduling variables and if the scheduling sets are hyperboxes, this leads to $n_\mathrm{d} = O(2^{n_\theta})$ and $n_\mathrm{ineq} = O(2^{n_\theta})$.
- $n_\mathrm{d} = O(\max_i t_{\varsigma_i})$ and $n_\mathrm{ineq} = O(\max_i r_{\varsigma_i} \max_i t_{\varsigma_i})$. That is, the number of decision variables scales linearly in the number of vertices of the (finite-step contractive) shape sets. The number of inequality constraints scales linearly in the product of the number of hyperplanes and vertices. If $n_\mathrm{x} = 2$, this leads to $n_\mathrm{ineq} = O(\max_i t_{\varsigma_i}^2)$. For higher-dimensional systems, a pessimistic complexity bound in terms of $\max_i t_{\varsigma_i}$ could be obtained by means of McMullen's upper bound theorem (Ziegler 1995, Section 8.4).
- Finally, observe that $n_\mathrm{d} = O(n_\mathrm{x})$, $n_\mathrm{eq} = O(n_\mathrm{x})$, and $n_\mathrm{ineq} = O(n_\mathrm{x})$. However, the representation complexity of the shape sets is also a function of n_x, and thus in practice this linear scaling in the state dimension is not attained.

4.4.3 Computation of finite-step contractive sets

The construction of a sequence of finite-step contractive sets $\mathbf{S}_M$ for an LPV system can be done in several ways. One can pick an arbitrary PC-set S_0 and find the smallest M for which a sequence $\mathbf{S}_M$ exists, using a straightforward extension of the algorithm for the LTI case from (Athanasopoulos and Lazar 2013). Due to exponential complexity in M, this method is only practical when contraction can be achieved for small M.

Alternatively, it is possible to first determine any stabilizing controller for (4.1). Then again we can choose an arbitrary PC-set S_0 and propagate this set forwards under the resulting closed-loop dynamics until finite-step contraction is achieved, as proposed in (Lazar and Spinu 2015). The number of vertices of the sets in the resulting sequence $\mathbf{S}_M$ grows exponentially in principle, but often many vertices are redundant and can be eliminated using standard algorithms: a similar technique was employed in (Athanasopoulos and Lazar 2014) for the stability analysis of switched systems.

A method based on particle swarm optimization (PSO) that can be used to compute controlled finite-step contractive sets in vertex representation is presented in Appendix A.

4.5 Numerical examples

In this section, the approach developed in this chapter is demonstrated on two numerical examples: a simple second-order academic example in Section 4.5.1 and a two-cart mechanical system in Section 4.5.2.

4.5.1 Second-order academic example

Consider a second-order LPV system defined in the state-space form of (4.1) with two scheduling variables

$$x(k+1) = \left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0.08 & -0.6 \\ 0.4 & 0.1 \end{bmatrix} \theta_1(k) + \begin{bmatrix} 0.23 & 0 \\ 0 & -0.32 \end{bmatrix} \theta_2(k) \right) x(k) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k)$$

where

$$\begin{aligned} \Theta &= \left\{ \theta \in \mathbb{R}^2 \mid \|\theta\| \leq 1 \right\}, \\ \mathbb{U} &= \{ u \in \mathbb{R} \mid |u| \leq 6 \}, \\ \mathbb{X} &= \left\{ x \in \mathbb{R}^2 \mid |x_1| \leq 4, |x_2| \leq 10 \right\}. \end{aligned}$$

The MPC tuning parameters are $N = 8$, $Q = I$, and $R = 0.25$. This tuning assigns a low weight to the control input, leading to a fast response. For simplicity, the scheduling tubes $\boldsymbol{\Theta} \subseteq \Theta^N$ were constructed such that $\boldsymbol{\Theta}_k = \{\{\theta(k)\}, \Theta, \ldots, \Theta\}$ for all k.

A set X_0^{f} was chosen which leads to a sequence $\mathbf{S}_M$ of (5, 0.95)-contractive sets, as depicted in Figure 4.2. The set X_0^{f} was designed with 4 vertices, and all subsequent sets also have 4 vertices except for X_1^{f}, which has 6. For comparison, the maximal controlled 0.95-contractive set was also calculated using the algorithm from (Blanchini and Miani 2015) and it has 8 vertices.

The relative difference in computational load of the resulting TMPC algorithm, based on an LP implementation where both the vertex- and hyperplane representations of the sets were

(M,λ)	n_{d}	n_{ineq}	Avg. time [ms]	Max. time [ms]
$(1,0.95)$	276	4034	14	20
$(5,0.95)$	[168..176]	[1674..1810]	6	8

Table 4.1: Illustration of complexity: number of decision variables, number of inequality constraints, and solver time per sample.

used, is displayed in Table 4.1. The simulations were carried out on a 3.6 GHz Intel Core i7-4790 with 8 GB RAM, running Arch Linux, and using the Gurobi 7.0.2 LP solver with its default settings. Because the complexity of the terminal set in the $(5,0.95)$-contractive case is time-dependent, the number of variables and constraints varies periodically between the numbers shown. To illustrate their linear growth, the maximum number of variables and constraints for the $(5,0.95)$-contractive case is calculated as a function of N and shown in Figure 4.1.

An example closed-loop output trajectory of the controller with finite-step terminal condition is shown in Figure 4.3. The initial state was $x(0) = \begin{bmatrix}4 & -6\end{bmatrix}$, i.e., taken at the boundary of the state constraint set. As expected, the system's state variables are steered to the origin and input- and state constraints are satisfied. The scheduling trajectory that was used during this simulation is depicted in Figure 4.4.

The achieved domains of attraction of the controller with the finite-step terminal condition are also compared to that of the "homothetic" controller from Section 3.5, which uses the maximal 0.95-contractive terminal set (Figure 4.5).

The feasible set was calculated for a fixed initial value $\theta(0) = \begin{bmatrix}1 & -1\end{bmatrix}^\top$. In the present case, the reduction in computational load due to the lesser complexity of the sets in $\mathbf{S}_M$ is paid for by a marginally smaller domain of attraction (DOA). The DOAs were estimated using the method from Appendix B.

4.5.2 Two-cart system

A two-cart positioning system, similar to that of (Kothare et al. 1996), is depicted in Figure 4.6. Therein, the variables q_1 and q_2 represent the position of the first and second cart, respectively. A force u can be applied to the first cart as a control input. The spring constant θ is time-varying and corresponds to the scheduling variable. It is assumed that the scheduling signal is exogenous, i.e., it is varied independently from the state and input signals. The used physical parameters are given in Table 4.2.

For this system, a state vector can be defined as $x = \begin{bmatrix}q_1 & q_2 & \dot{q}_1 & \dot{q}_2\end{bmatrix}^\top$. Its dynamics are described by the continuous-time state-space (SS) representation

$$\dot{x}(t) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\frac{\theta(t)}{m_1} & \frac{\theta(t)}{m_1} & -\frac{d}{m_1} & \frac{d}{m_1} \\ \frac{\theta(t)}{m_2} & -\frac{\theta(t)}{m_2} & \frac{d}{m_2} & -\frac{d}{m_2} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_1} \\ 0 \end{bmatrix} u(t) \tag{4.26}$$

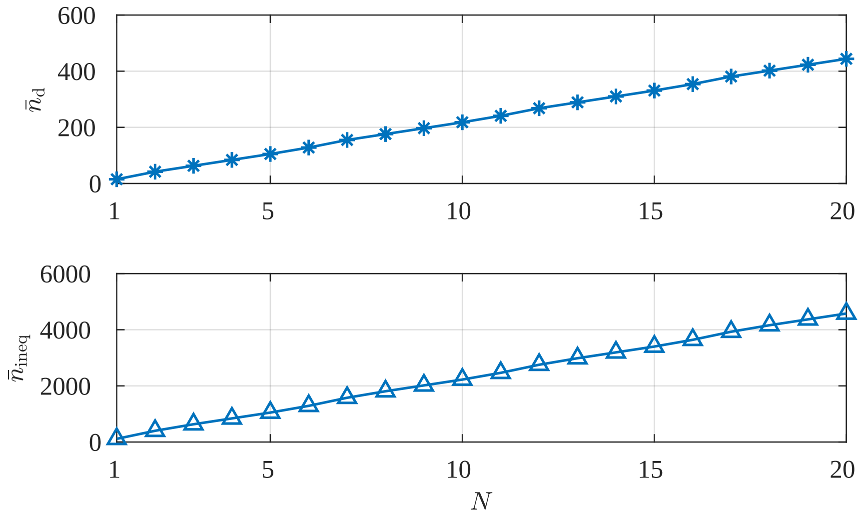


Figure 4.1: Maximum number of variables $\bar{n}_\mathrm{d}(N) = \max_k n_\mathrm{d}(k, N)$ and constraints $\bar{n}_\mathrm{ineq}(N) = \max_k n_\mathrm{ineq}(k, N)$.

Description	Symbol	Value	Unit
Mass of the first cart	m_1	1	kg
Mass of the second cart	m_2	1	kg
Spring constant	θ	$\theta \in [\underline{\theta}, \overline{\theta}] = [1.0, 2.5]$	Nm^{-1}
Damping coefficient	d	0.01	Nsm^{-1}

Table 4.2: Parameters of the two-cart system.

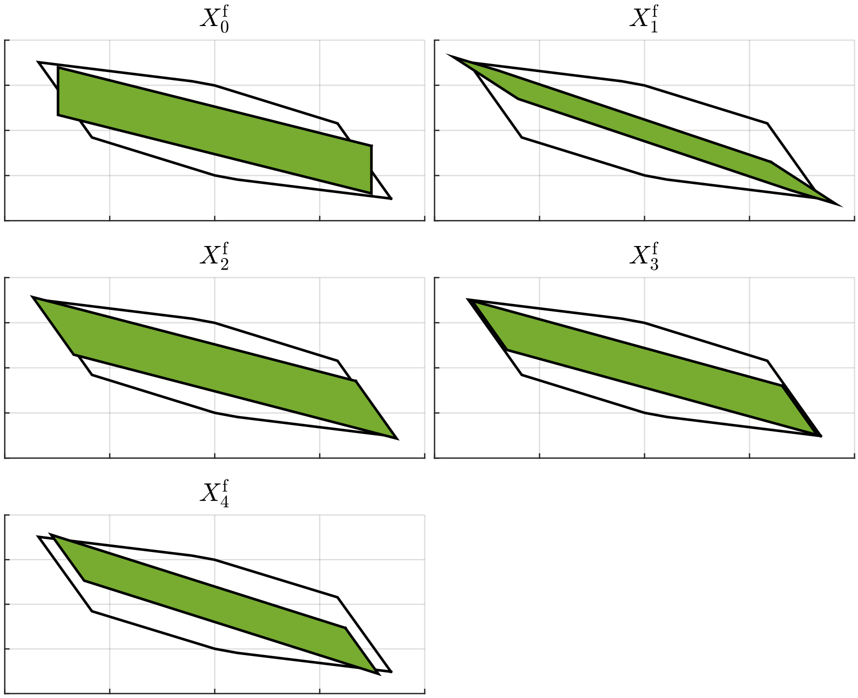


Figure 4.2: Constructed sequence of $(5, 0.95)$-contractive sets (solid, green) compared with the maximal 0.95-contractive set (outline).

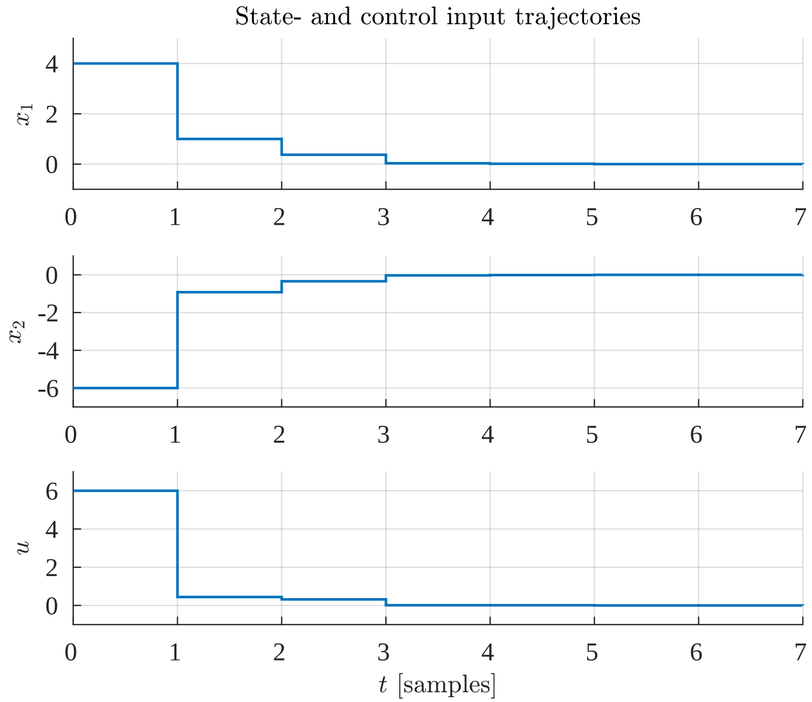


Figure 4.3: Closed-loop state- and input trajectories with finite-step terminal condition.

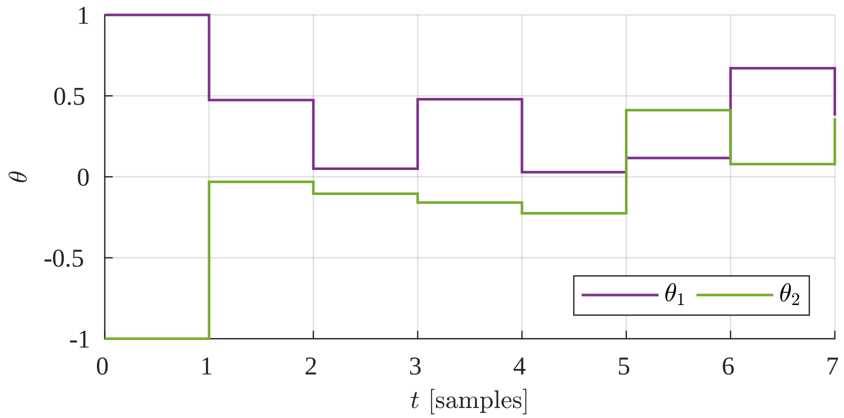


Figure 4.4: Scheduling trajectory.

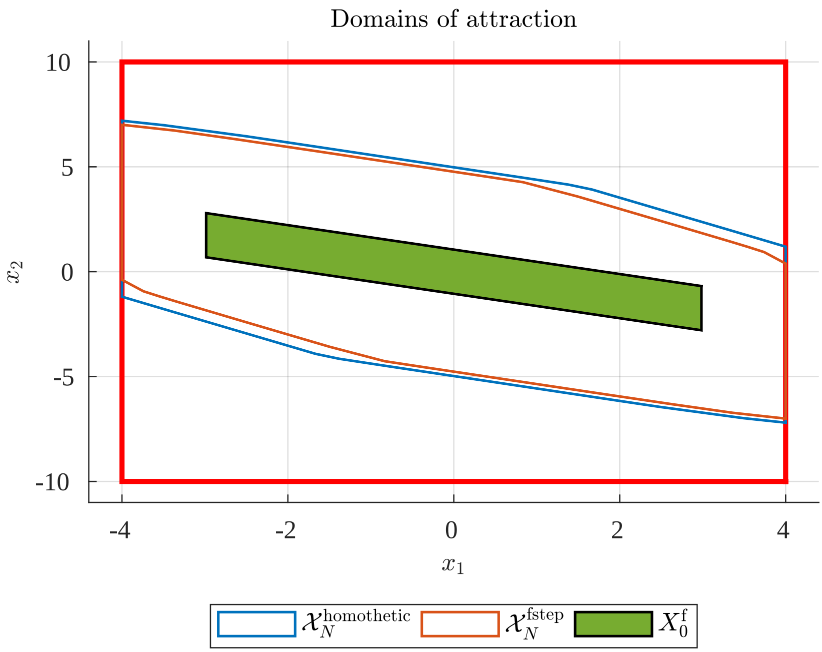


Figure 4.5: Approximate domains of attraction with finite-step terminal condition ($\mathcal{X}_N^{\text{fstep}}$) and with maximal contractive terminal set terminal condition ($\mathcal{X}_N^{\text{homothetic}}$), for $\theta(0) = [1\ -1]^\top$. The innermost set (solid, green) is X_0^{f}, and the outer box represents the state constraints.

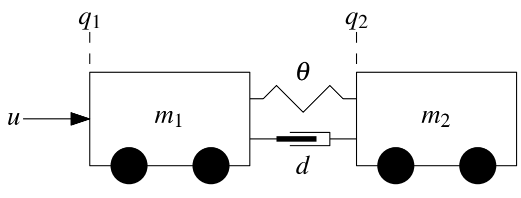


Figure 4.6: The two-cart system.

where $\theta : \mathbb{R}_+ \to \Theta$, $\Theta = [\underline{\theta}, \overline{\theta}]$, represents the time-varying spring constant. The system is subject to the constraints

$$\mathbb{X} = \left\{ x \in \mathbb{R}^4 \,\middle|\, \|x\|_\infty \leq 1 \right\},$$
$$\mathbb{U} = \{ u \in \mathbb{R} \,|\, |u| \leq 1 \}.$$

In (Kothare et al. 1996), an Euler-discretized version of (4.26) is used for the MPC design. However, in this example, it was found that Euler discretization is unsuitable because it leads to an unstable discrete-time model irrespective of the sampling time. Unfortunately, using a more accurate approximation (e.g., polynomial or Tustin) would lead to a non-polytopic discrete-time LPV state-space (LPV-SS) representation. Therefore, to obtain a polytopic discrete-time representation, the following "ad hoc" approach—which nevertheless works well for this example—was used. The exact discretization (see (Tóth 2010, Chapter 6)) was computed on a grid $\Theta_{\mathrm{grid}} = \{\bar{\theta}^1, \ldots, \bar{\theta}^L\} \subset \Theta$. An LPV-PSS representation of the form (4.1) was subsequently fitted on the obtained collection of discrete-time models by solving

$$(A_0, A_1, B) = \arg \min_{A_0, A_1, B} \sum_{i=1}^{L} \left\| \begin{bmatrix} \tilde{A}\left(\bar{\theta}^i\right) \\ \tilde{B}\left(\bar{\theta}^i\right) \end{bmatrix} - \begin{bmatrix} A_0 + A_1 \bar{\theta}^i \\ B \end{bmatrix} \right\|_{\mathrm{F}}$$

where $\|\cdot\|_{\mathrm{F}}$ is the Frobenius norm[3], and where $(\tilde{A}(\cdot), \tilde{B}(\cdot))$ are the matrices corresponding to the exact discretization. It was found that the discrete-time model obtained in this way, although not exact due to the fitting to a polytopic representation, is more accurate than the first-order Euler approximation: in particular, stability of the representation (4.26) is preserved. In the sequel, a sampling time of $\tau = 0.25$ seconds is used. This gives a discrete-time LPV-PSS representation with the numerical values

$$x(k+1) = \left(\begin{bmatrix} 0.999 & 0.001 & 0.250 & 0.000 \\ 0.001 & 0.999 & 0.000 & 0.250 \\ -0.015 & 0.015 & 0.997 & 0.003 \\ 0.015 & -0.015 & 0.003 & 0.997 \end{bmatrix} + \begin{bmatrix} -0.03 & 0.03 & -0.003 & 0.003 \\ 0.03 & -0.03 & 0.003 & -0.003 \\ -0.232 & 0.232 & -0.03 & 0.03 \\ 0.232 & -0.232 & 0.03 & -0.03 \end{bmatrix} \theta(k) \right) x(k) + \begin{bmatrix} 0.031 \\ 0.000 \\ 0.245 \\ 0.005 \end{bmatrix} u(k). \tag{4.27}$$

A controlled $(2, 0.985)$-contractive sequence for (4.27) was computed using the algorithm of Appendix A, and it is depicted in Figure 4.7. Because the sets are four-dimensional, they are visualized by means of projections. The number of vertices of the first set in the sequence S_0 was fixed to be 16, and the second set S_1 has 30 vertices. For comparison, a "normal" 0.985-contractive set corresponding to a robustly stabilizing linear feedback $u = Kx$ was also computed using Algorithm 2.1. This set is represented by 250 vertices and its projections are also shown in Figure 4.7.

[3] For a matrix $M \in \mathbb{R}^{m \times n}$, the Frobenius norm is defined as $\|M\|_{\mathrm{F}} = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |m_{ij}|^2}$, with m_{ij} denoting the (i, j)-th element of M.

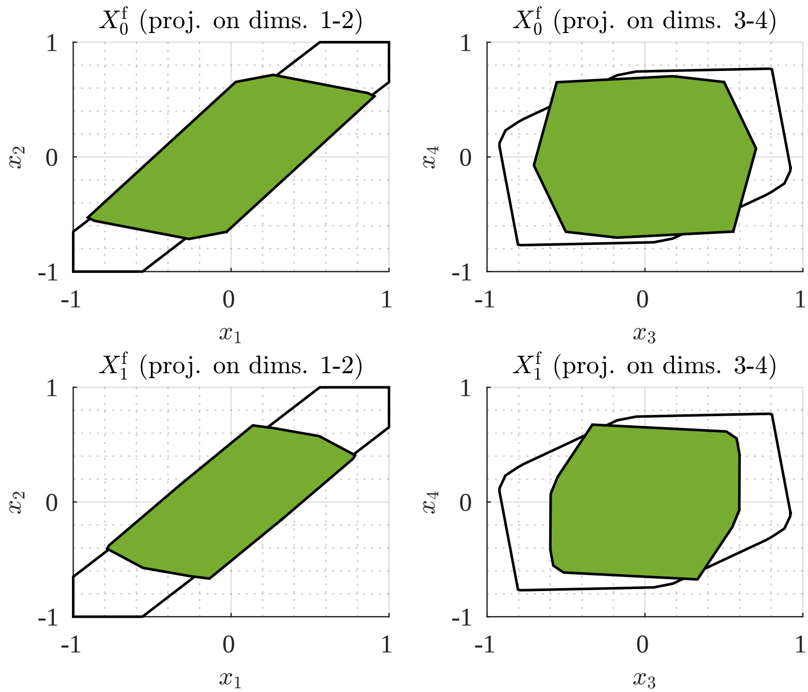


Figure 4.7: The computed $(2, 0.985)$-contractive sequence (solid green), displayed together with a "normal" 0.985-contractive set with a higher number of vertices.

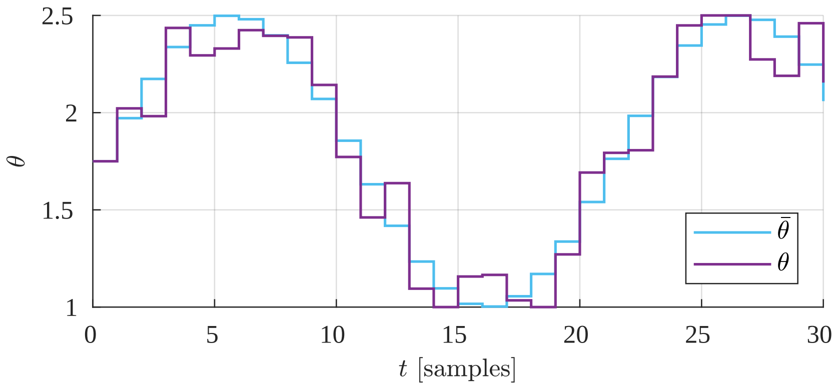


Figure 4.8: Scheduling trajectories for the two-cart example: true realized trajectory $\theta(\cdot)$ and "nominal" trajectory $\bar{\theta}(\cdot)$ used for predicting the possible future scheduling trajectories in an anticipative controller.

It was attempted to use the method in Appendix A to synthesize a standard controlled contractive set with a comparably low number of vertices as well, but this was not successful. Therefore, for this system, it seems that a finite-step terminal set can indeed be easier to compute.

For the simulation, the prediction horizon of the tube-based controller was set to $N = 8$, and its tuning parameters were set as $Q = \operatorname{diag}\{5, 5, 1, 1\}$ and $R = 10$. The representation (4.27) was used in the MPC, but the continuous-time model (4.26) was used for simulation. The scheduling trajectory $\theta(\cdot)$ that was used in the simulation is depicted in Figure 4.8. The following three controllers are compared in closed-loop simulation:

Anticipative TMPC The tube-based controller developed in this chapter with the $(2, 0.985)$-contractive terminal set of Figure 4.7. An "anticipative" description of possible future scheduling trajectories is used similar to the description (3.4). In particular, at every sample instant $k \in \mathbb{N}$, it is assumed that

$$\forall i \in [1..N-1]: \ \theta(k+i) \in \big(\bar{\theta}(k+i) \oplus [-0.25, 0.25]\big) \cap \Theta \tag{4.28}$$

where the "nominal" scheduling trajectory $\bar{\theta}(\cdot)$ is also depicted in Figure 4.8. Note that the actual realized trajectory $\theta(\cdot)$ is consistent with (4.28) in the sense that for all k, $|\theta(k) - \bar{\theta}(k)| \leq 0.25$.

Classical TMPC This controller is the same as the "anticipative TMPC", except that the description (4.28) of possible future scheduling trajectories is not used. Instead, at every sample instant $k \in \mathbb{N}$, it is simply assumed for all $i \in [1..N-1]$ that $\theta(k+i) \in \Theta$ (i.e., it is assumed that θ can vary arbitrarily fast inside of its full range Θ).

Quasi min-max The quasi min-max approach from (Lu and Arkun 2000b).

The obtained closed-loop state trajectories are shown in Figure 4.9, and the corresponding closed-loop control inputs are shown in Figure 4.10. The initial state was $x(0) = \begin{bmatrix} 0 & 0.65 & 0 & 0 \end{bmatrix}^\top$. It is observed that the TMPC controllers apply more aggressive control inputs than the quasi min-max controller, and drive the state of the system to the origin somewhat faster. In turn, the anticipative TMPC is able to exploit the description (4.28) of possible future scheduling trajectories and drives the state to the origin faster than the non-anticipative TMPC. Because the difference between the trajectories in Figure 4.9 can be difficult to distinguish, an enlarged plot showing only the evolution of the first state variable is additionally given in Figure 4.11.

4.6 Concluding remarks

This concluding section starts with Section 4.6.1, which contains a discussion on the possibility of including a parameter-varying input matrix in the representation (4.1). A summary of the chapter and outlook on future developments is presented in Section 4.6.2.

4.6.1 Parameter-varying input matrix

An extension of the theoretical results from this chapter to the case that the B-matrix in (4.1) is also parameter-varying, is straightforward. However, if a gain-scheduled control policy is used, the tube synthesis optimization problem is no longer convex due to the multiplication of the parameter-dependent matrix with parameter-dependent control actions (decision variables). This non-convexity is a problem that arises also in the synthesis of LPV controllers for LPV-PSS representations (Daafouz and Bernussou 2001). Several workarounds to this issue exist:

- It is possible to revert to “robust” control, that is to make the controllers $K_i : X_i \times \Theta_i \to \mathbb{U}$ independent of θ (or at least, independent of those elements in θ that appear in $B(\theta)$). A more detailed characterization of this approach is presented in this thesis in Chapter 5.
- Given an LPV-SS representation $x(k+1) = A(\theta(k))x(k) + B(\theta(k))u(k)$, it is possible to add an input filter in the loop to obtain an augmented system where all parameter dependencies are shifted into the A-matrix. For instance, in the case that the input filter is chosen to be an integrator, the augmented system becomes

  $$\begin{bmatrix} x(k+1) \\ u(k+1) \end{bmatrix} = \begin{bmatrix} A(\theta(k)) & B(\theta(k)) \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ u(k) \end{bmatrix} + \begin{bmatrix} 0 \\ I \end{bmatrix} v(k),$$

  where the state dimension is increased by n_u. Due to the input filter, an input delay of one sample is introduced. This may deteriorate performance and can even render the system unstabilizable (Blanchini et al. 2007).
- It is possible to convexify the optimization problem by over-approximating the non-convex constraints with a larger number of convex constraints. In (Besselmann et al. 2012) this is achieved through (a reverse form of) Pólya’s relaxation. Similarly, (Casavola et al. 2012) gives a convex outer-approximation of the non-convex constraint set.

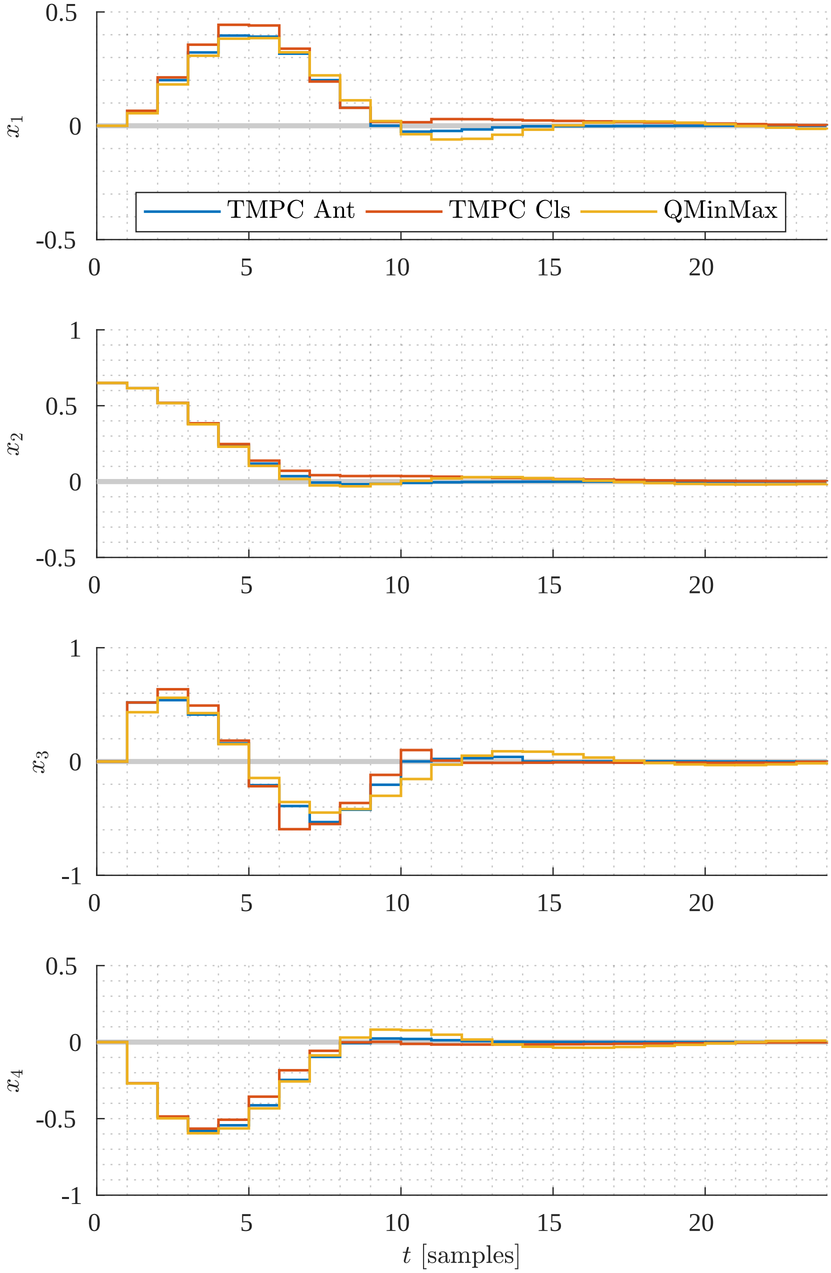


Figure 4.9: Closed-loop state trajectories for the two-cart example.

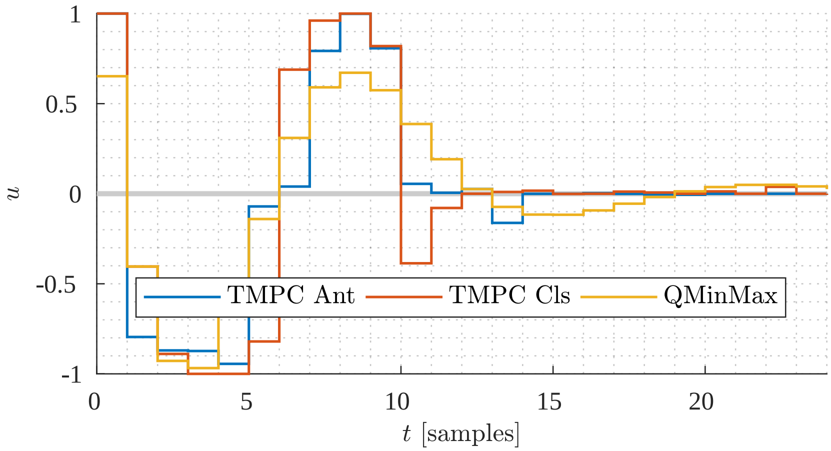


Figure 4.10: Closed-loop input trajectories for the two-cart example.

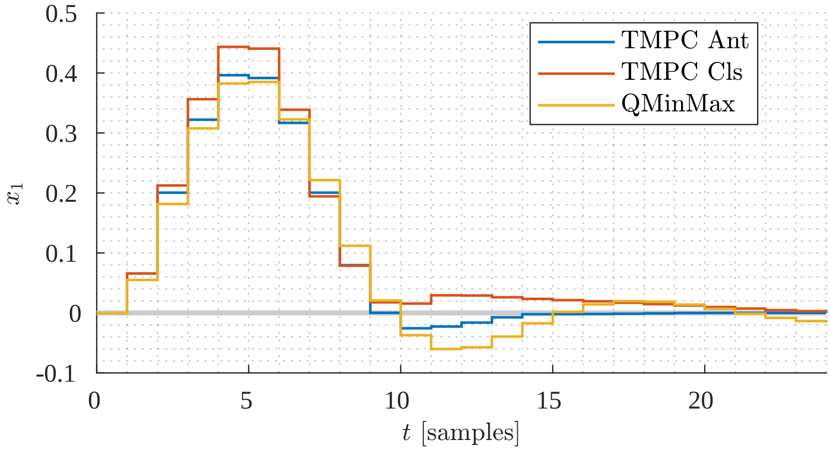


Figure 4.11: Closed-loop trajectory of the first state variable for the two-cart example.

Recently, it was proven that the problem of synthesizing an LPV state feedback for an LPV-PSS representation with a parameter-dependent B-matrix can be formulated as a convex semi-definite program (SDP) without the need for an input filter (Pandey and Oliveira 2017). Whether or not it is possible to formulate the synthesis of vertex controls for such a representation as a single convex problem, is currently an open question.

4.6.2 Summary

The present chapter has introduced finite-step terminal conditions in tube-based MPC for LPV systems. It was shown that, under certain assumptions on the tube parameterization, the method is recursively feasible. A new Lyapunov-like function on periodic sequences of PC-sets was constructed, and it was subsequently used to derive a terminal cost. This construction enabled to prove closed-loop asymptotic stability. The implementation of the proposed control approach in terms of an LP was discussed, and its computational properties were demonstrated on a numerical example. The developments from this chapter have provided an answer to Research Question Q_2: namely, how to design a stabilizing MPC strategy for LPV systems that can exploit knowledge about possible future scheduling trajectories.

In the presented approach, a particular "periodic" tube parameterization is used. This is a more elaborate tube parameterization than the one employed previously in Section 3.5. It can then be questioned if it is possible to consider even more different tube parameterizations, for instance with the goal of improving the trade-off between the computational complexity of the controller and the size of its achieved DOA. This makes Research Question Q_3—is it possible to introduce an extra degree-of-freedom in the design of the tube parameterization?—a natural follow-up question to be addressed. Indeed, this question will be addressed next in Chapter 5.

A possible topic for future research is the development of improved methods for the computation of finite-step contractive sets for LPV systems.

Chapter 5

Heterogeneously parameterized tube MPC

THIS CHAPTER presents a heterogeneously parameterized tube-based model predictive control design applicable to linear parameter-varying systems. In a heterogeneous tube, the parameterization of the tube cross sections and the associated control laws are allowed to vary along the prediction horizon. Two extremes that can be described in this framework are scenario MPC (high complexity, large domain of attraction) and homothetic tube MPC with a simple time-invariant control parameterization (low complexity, small domain of attraction). In the proposed framework these extreme parameterizations, as well as other parameterizations of intermediate complexity, can be combined to form a single tube. By allowing for more flexibility in the parameterization design, one can control the trade-off between computational cost and the size of the domain of attraction of heterogeneously parameterized tube MPC. Non-restrictive sufficient conditions on the parameterization structure are developed under which recursive feasibility and closed-loop stability is guaranteed. A specific parameterization that combines the principles of scenario and homothetic tube MPC is proposed and it is shown to satisfy the required conditions. The computational properties of the approach are demonstrated using a numerical example.

5.1 Introduction

Any tractable linear parameter-varying (LPV) model predictive control (MPC) design necessarily incurs a degree of conservatism with respect to the DP solution in terms of the achievable optimal cost and the resulting domain of attraction (DOA). In the tube-based framework, a key question that should be addressed is how to parameterize the tube to strike a good balance between control performance and computational complexity. However, this question seems to be hardly addressed in the literature. Typically, a single parameterization is first selected for the full prediction horizon, and properties such as the volume of the ensuing DOA are analyzed *a posteriori*. It may then turn out that a computationally efficient controller based on a simple parameterization such as (Fleming et al. 2015) attains an unsatisfactorily small DOA. On the other hand, a controller utilizing a full vertex control parameterization (Chapter 4) or even a scenario parameterization may achieve the required DOA, but at a disproportionate computational cost. There is currently no way to “interpolate” between these cases: one has to use either the simple or the complex parameterization over the full prediction horizon.

Based on the observation that a simple controller usually works well in a small region around the terminal set, whereas a complex controller is only necessary if the domain of attraction is to be increased further, one can wonder if the same reasoning applies to the predictions themselves. If so, it could be worthwhile to vary the complexity of the parameterization along the prediction horizon. Then, it becomes possible to use the richest prediction structure where it is needed most (far from the terminal set and close to the constraints), without needlessly considering many degrees of freedom (DOFs) when the predictions are getting close to the terminal set.

In this chapter, this intuitive idea is formalized by introducing the notion of a heterogeneously parameterized tube (HpT). In a heterogeneously parameterized tube, the parameterization of the cross sections and associated controllers is allowed to be time-varying along the prediction horizon. The HpT concept can be used to describe a number of tube parameterizations that have appeared in the literature in a unified fashion, including but not limited to scenario-, homothetic-, and elastic tubes. Broadly speaking, this allows the use of a detailed but complex prediction structure near the beginning of the horizon while gradually transitioning to a simpler but more conservative structure as the predictions tend towards a neighborhood of the terminal set. This framework provides a new degree of freedom in the parameterization design, which can be exploited by the user to obtain new and improved performance/complexity trade-offs.

Based on the introduced HpT concept, an MPC algorithm based on repetitive on-line construction of an HpT is presented. Sufficient conditions on the underlying heterogeneous parameterization, under which the controller is guaranteed to be recursively feasible and asymptotically stabilizing, are given. A specific heterogeneous parameterization, called HpT-SF, is proposed. This HpT-SF parameterization combines the principles of scenario and homothetic tube MPC, and it is shown to satisfy the required conditions.

The proposed HpT idea shares a conceptual similarity with the concept of move blocking (MB) in nominal MPC (Cagienard et al. 2007). In MB, the number of control DOFs is reduced by fixing the predicted control inputs to be constant over a part of the horizon. Although the mathematical formulation and conditions are quite different, the MB and HpT approaches both allow one to parameterize the predicted control inputs along a part of the prediction horizon in a simpler way, thereby reducing the number of decision variables. Move blocking for robust MPC based on constraint tightening for additively disturbed LTI systems was considered in (Shekhar and Maciejowski 2012, 2013).

The remainder of this chapter is structured as follows. Section 5.1.1 introduces in more detail the problem setting considered in this chapter. The concept of heterogeneously parameterized tubes is presented in Section 5.2, and the tube model predictive control (TMPC) algorithm based on these tubes is developed in Section 5.3. Conditions such that the algorithm is recursively feasible and stabilizing are given. Subsequently, in Section 5.4, a cost function and an implementable heterogeneous parameterization are provided that satisfy the required assumptions. Numerical examples to demonstrate the properties of the method are provided in Section 5.5, and the chapter ends with some concluding remarks in Section 5.6.

5.1.1 Problem setting

The control problem considered in this chapter is equal to Problem 3.1, i.e., it is desired to design a controller $K_{\text{mpc}}(\cdot,\cdot,\cdot)$ that renders the origin of a given constrained polytopic LPV state-space (LPV-PSS) representation regionally asymptotically stable.

In contrast to the situation in Chapter 4, all theoretical results in this chapter are derived with respect to the general case that the input matrix $B(\theta)$ can be parameter-varying. The necessary assumptions that enable the implementation of the proposed algorithms in terms of finite-dimensional convex optimization problems are given in Section 5.4.

The control problem will, in this chapter, be addressed through an MPC algorithm based on the construction of heterogeneously parameterized tubes.

5.2 Heterogeneously parameterized tubes

In this section, the concept of a heterogeneously parameterized tube (HpT) is introduced. The resulting HpT tube synthesis problem, which will be at the core of the MPC strategy proposed in this chapter, is specified.

Formally, an HpT is defined as follows.

Definition 5.1. *Let* $\mathbf{T}$ *be a tube according to Definition 3.4. For all* $i \in \mathbb{N}_{[0,N]}$*, introduce parameter sets* $\mathbb{P}(i) = \mathbb{P}^{\text{x}}(i) \times \mathbb{P}^{\text{k}}(i)$*. A tube* $\mathbf{T}$ *is called a* heterogeneously parameterized tube (HpT) *if it satisfies the following conditions:*

(i) For all $i \in \mathbb{N}_{[0,N]}$*, there is a function* $P^{\text{x}}(\cdot|i) : \mathbb{P}^{\text{x}}(i) \to \mathbb{R}^{n_{\text{x}}}$ *and there exists a parameter* $p_i^{\text{x}} \in \mathbb{P}^{\text{x}}(i)$ *such that* $X_i = P^{\text{x}}\big(p_i^{\text{x}}|i\big)$*.*

(ii) For all $i \in \mathbb{N}_{[0,N-1]}$ *there is a second-order function* $P^{\text{k}}(\cdot|i) : \mathbb{P}^{\text{k}}(i) \to \big(X_i \times \Theta_i \to \mathbb{U}\big)$ *and there exists a parameter* $p_i^{\text{k}} \in \mathbb{P}^{\text{k}}(i)$ *such that* $K_i = P^{\text{k}}\big(p_i^{\text{k}}|i\big)$*.*

Further, define the shorthand $P(p|i) = \big(P^{\text{x}}(p^{\text{x}}|i), P^{\text{k}}(p^{\text{k}}|i)\big)$ *where* $p = \big(p^{\text{x}}, p^{\text{k}}\big)$*.*

The distinguishing feature of the parameterization proposed in the above definition, and the reason why it is called a *heterogeneous* parameterization, is the fact that the sets $\mathbb{P}(i)$ and functions $P(\cdot|i)$ can be different for every prediction time instant $i \in \mathbb{N}_{[0,N]}$. This can be compared and contrasted with the approach in Chapter 4, where the parameterizations were dependent on $k+i$ in a prescribed periodic fashion. If the sets and functions in Definition 5.1 are chosen to be independent of i, then the setup of Chapter 4 with $M = 1$ (and equivalently, the setup of Section 3.5) is recovered. With the above definition, the notion of *heterogeneous parameterization structure* can be associated.

Definition 5.2. *A heterogeneous parameterization structure* $\mathscr{P}_N$ *is defined as the sequence of 2-tuples*

$$\mathscr{P}_N = \big\{(\mathbb{P}(0), P(\cdot|0)), \ldots, (\mathbb{P}(N), P(\cdot|N))\big\}.$$

The parameterization structure $\mathcal{P}_N$ has to be selected by the user and determines the computational complexity and the achievable performance (level of conservatism) of the resulting tube-based controller.

In the HpT setting, the definition of the set of feasible tubes, the tube synthesis problem, and the domain of attraction are comparable to the general definitions given in 3.4.1. The principal difference is that these are now all defined given a designed parameterization structure $\mathcal{P}_N$. Specifically, in this chapter, we write

$$\mathcal{T}_N\left(x, \mathbf{\Theta} | \mathcal{P}_N\right) = \{\mathbf{T} \,|\, \mathbf{T} \text{ satisfies Def. 5.1 with } X_0 = \{x\} \text{ and } X_N \subseteq X_{\mathrm{f}}\} \tag{5.1}$$

for the set of feasible tubes,

$$V\left(x, \mathbf{\Theta} | \mathcal{P}_N\right) = \min_{\mathbf{T}} J_N\left(\mathbf{T}, \mathbf{\Theta}\right) \text{ subject to } \mathbf{T} \in \mathcal{T}_N\left(x, \mathbf{\Theta} | \mathcal{P}_N\right), \tag{5.2}$$

for the tube synthesis problem, and

$$\mathcal{X}_N\left(\mathbf{\Theta} | \mathcal{P}_N\right) = \{x \in \mathbb{X} \,|\, \mathcal{T}_N\left(x, \mathbf{\Theta} | \mathcal{P}_N\right) \neq \emptyset\} \tag{5.3}$$

for the domain of attraction (DOA). Necessary conditions on the terminal set X_{f} in (5.1) in the HpT setting will be given in this chapter. The definition of the cost function $J_N(\cdot,\cdot,\cdot)$ in (5.2) is the same as (3.9), and the stage cost (3.13) is employed. Like the terminal set, a suitable terminal cost $F(\cdot)$ will be provided in this chapter.

In the next section, it is shown how the HpT-notion of Definition 5.1 can be employed to construct a recursively feasible receding-horizon strategy along the lines of Algorithm 3.1.

5.3 The heterogeneous TMPC algorithm

In this section, the MPC algorithm based on the synthesis of heterogeneously parameterized tubes is developed. Conditions on the tube parameterization and the cost function such that a recursively feasible and stabilizing controller can be obtained are given in Section 5.3.1. In Section 5.3.2, the algorithm is summarized and its properties are proven.

5.3.1 Parameterization conditions

In this subsection, a number of conditions is presented that allows for the derivation of a recursively feasible and stabilizing MPC algorithm. These conditions come as a set of assumptions on (i) the existence of a terminal set and local controller, (ii) the parameterization structure $\mathcal{P}_N$, and (iii) the cost function $J_N(\cdot,\cdot)$.

Given a controller $K : X \times \Theta \to \mathbb{U}$, it is useful to consider controllers that are “the same” as K on a subset of the original domain $X \times \Theta$. Formally, the set of *restrictions* of a controller can be defined as follows.

Definition 5.3. *Let $K : X \times \Theta \to \mathbb{U}$. The set of restrictions of K to the subset $X' \times \Theta' \subseteq X \times \Theta$ is* $\mathcal{R}\left(K \middle| X', \Theta'\right) = \left\{K' : X' \times \Theta' \to \mathbb{U} \,\middle|\, \forall (x,\theta) \in X' \times \Theta' : \; K'(x,\theta) = K(x,\theta)\right\}$.

The first set of assumptions specifies necessary terminal conditions.

Assumption 5.1. *The terminal set X_f in* (5.2) *is controlled λ-contractive in the sense of Definition 2.13. (Recall that a local controller which renders X_f λ-contractive in this sense, is denoted as $K_\mathrm{f} : X_\mathrm{f} \times \Theta \to \mathbb{U}$.)*

In this chapter, finite-step terminal conditions, as used in Chapter 4, are not considered. It should however be possible to extend the results from the present chapter to allow for finite-step terminal conditions as well. This can be a topic for future research.

With the terminal conditions of Assumption 5.1 in place, the first step towards a proof of recursive feasibility can be made. To do this, the following instrumental lemma on the existence of "successor tubes" is required.

Lemma 5.1. *Let $N \in \mathbb{N}_{[1,\infty)}$, let $\mathcal{P}_N$ be a heterogeneous parameterization structure according to Definition 5.2, and let (x,θ) be the current state and scheduling variable values. Furthermore, let $\mathbf{\Theta} = \{\{\theta\}, \{\Theta_i\}_{i=1}^{N-1}\} \subseteq \Theta^N$ and $\mathbf{\Theta}^+ = \{\Theta_i^+\}_{i=0}^{N-1} \subseteq \Theta^N$ be two scheduling tubes satisfying $\mathbf{\Theta}^+ \sqsubseteq \mathbf{\Theta}$. Suppose that there exists a tube $\mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta} | \mathcal{P}_N)$. Then, there always exists a $\gamma \in [0,1]$ and a sequence of sets $\mathbf{X}^+ = \{X_i^+\}_{i=0}^{N}$ that satisfies*

$$\forall i \in \mathbb{N}_{[0,N-2]} : X_i^+ \subseteq X_{i+1}, \tag{5.4a}$$

$$X_{N-1}^+ \subseteq \gamma X_\mathrm{f}, \tag{5.4b}$$

$$X_N^+ \subseteq \lambda\gamma X_\mathrm{f}, \tag{5.4c}$$

$$\forall i \in \mathbb{N}_{[0,N-2]} : \mathcal{G}(X_i^+, \Theta_i^+ | K_{i+1}) \subseteq X_{i+1}^+ \cap \mathbb{X}, \tag{5.4d}$$

$$\mathcal{G}(X_{N-1}^+, \Theta_{N-1}^+ | K_\mathrm{f}) \subseteq X_N^+. \tag{5.4e}$$

Proof. By construction of $\mathbf{T}$, $X_0^+ = \{A(\theta)x + BK_0(x,\theta)\} \subseteq X_1$. Recall that $\mathbf{\Theta}^+ \sqsubseteq \mathbf{\Theta}$ means that $\Theta_i^+ \subseteq \Theta_{i+1}$ for all $i \in \mathbb{N}_{[0,N-2]}$. Thus, from Lemma 2.3, it follows that $\mathcal{G}(X_0^+, \Theta_0^+ | K_1) \subseteq \mathcal{G}(X_1, \Theta_1 | K_1)$. Because it is known that $\mathcal{G}(X_1, \Theta_1 | K_1) \subseteq X_2 \cap \mathbb{X}$, there exists a subset $X_1^+ \subseteq X_2$ such that $\mathcal{G}(X_0^+, \Theta_0^+ | K_1) \subseteq X_1^+ \cap \mathbb{X}$. Repeating this argument for all subsequent time instances $i \in \mathbb{N}_{[1,N-2]}$ yields the existence of a sequence satisfying (5.4a) and (5.4d).

Let $\gamma = \inf\{\gamma' \mid X_N \subseteq \gamma' X_\mathrm{f}\}$. By construction of $\mathbf{T}$, it is known that the inclusion $X_N \subseteq X_\mathrm{f}$ holds, and so we have $\gamma \in [0,1]$. It was previously established that $X_{N-2}^+ \subseteq X_{N-1}$. Thus, $\mathcal{G}(X_{N-2}^+, \Theta_{N-2}^+ | K_{N-1}) \subseteq X_N \subseteq \gamma X_\mathrm{f}$, which gives (5.4b).

Homogeneity of (2.4) combined with Assumption 5.1 imply that for all $\gamma \in [0,1]$ the inclusion $\mathcal{G}(\gamma X_\mathrm{f}, \Theta | K_\mathrm{f}) \subseteq \lambda\gamma X_\mathrm{f}$ holds. Because $X_{N-1}^+ \subseteq \gamma X_\mathrm{f}$ and $\Theta_{N-1}^+ \subseteq \Theta$ this yields (5.4c) and (5.4e), completing the proof. □

In the recursive feasibility proofs considered earlier, i.e., in Proposition 4.2, a successor tube was explicitly constructed in a way analogous to the "shifting of the sequence". The result from the above lemma is different in that it establishes the existence of a successor tube that satisfies certain properties, without explicitly constructing one. As apparent from the proof, a sequence $\mathbf{X}^+$ that satisfies the stronger condition $X_{N-1}^+ \subseteq X_N^+ \subseteq \gamma X_\mathrm{f}$ instead of just (5.4b) can exist. However, for proving recursive feasibility and stability later, condition (5.4b) is sufficient and less restrictive in terms of the permissible designs of $\mathcal{P}_N$.

Remark 5.1 (Successor tube construction)**.** *Two particular constructions of the sequence* $\mathbf{X}^+$ *from Lemma 5.1 are the following. First, it is possible to set "*$\subseteq$*" in* (5.4d)-(5.4e) *to "*$=$*": then, conditions* (5.4a)-(5.4c) *are directly implied. Second, it is also possible to replace inclusion with equality in* (5.4a)-(5.4c) *which, in turn, implies the satisfaction of* (5.4d)-(5.4e)*. This second option is analogous to "shifting the sequence" in standard MPC, and it is the same method that was applied in Chapter 4. In this chapter, both constructions will be used to prove recursive feasibility of the heterogeneous parameterization that will be proposed in Section 5.4.2.*

Under the condition that a feasible tube $\mathbf{T}$ exists, Lemma 5.1 established the existence of a successor tube $\mathbf{T}^+$, but without considering the question if this successor can be parameterized in the structure $\mathcal{P}_N$.

Assumption 5.2. *Suppose the hypotheses of Lemma 5.1 hold true. Assume that* $\mathcal{P}_N$ *is designed such that at least one of the possible sequences* $\mathbf{X}^+$ *fulfilling* (5.4) *satisfies the following conditions:*

(i) For all $i \in \mathbb{N}_{[0,N]}$*, there exists a* $p_i^{\mathrm{x}+} \in \mathbb{P}^{\mathrm{x}}(i)$ *with* $P^{\mathrm{x}}\left(p_i^{\mathrm{x}+} \middle| i\right) = X_i^+$.

(ii) For all $i \in \mathbb{N}_{[1,N-2]}$*, there exists a* $p_i^{\mathrm{k}+} \in \mathbb{P}^{\mathrm{k}}(i)$ *such that* $P^{\mathrm{k}}\left(p_i^{\mathrm{k}+} \middle| i\right) \in \mathcal{R}\left(K_{i+1} \middle| X_i^+, \Theta_i^+\right)$.

(iii) There exists a $p_{N-1}^{\mathrm{k}+} \in \mathbb{P}(N-1)$ *such that* $P^{\mathrm{k}}\left(p_{N-1}^{\mathrm{k}+} \middle| N-1\right) \in \mathcal{R}\left(K_{\mathrm{f}} \middle| X_{N-1}^+, \Theta_{N-1}^+\right)$.

In Assumption 5.2, it can be argued that conditions (ii)-(iii) could be replaced by $P^{\mathrm{k}}\left(p_i^{\mathrm{k}+} \middle| i\right) = K_{i+1}$ and $P^{\mathrm{k}}\left(p_{N-1}^{\mathrm{k}+} \middle| N-1\right) = K_{\mathrm{f}}$, respectively. This is, however, more restrictive in the following sense. Suppose for instance that, for a certain i, X_i is a set with 2 vertices and X_{i+1} is a structurally different set with 4 vertices (this exact situation can occur, e.g., in the "scenario"-parameterization introduced later in Section 5.4.2). If the corresponding controllers (K_i, K_{i+1}) are parameterized as vertex controllers on these sets, then $\mathbb{P}^{\mathrm{k}}(i)$ is a set in dimension $2n_{\mathrm{u}}$ whereas $\mathbb{P}^{\mathrm{k}}(i+1)$ is a set in dimension $4n_{\mathrm{u}}$. Due to this structural difference, it is clearly impossible to find a parameter $p_i^{\mathrm{k}} \in \mathbb{P}(i)$ such that $P^{\mathrm{k}}\left(p_i^{\mathrm{k}+} \middle| i\right) = K_i^+ = K_{i+1}$. In contrast, a parameter $p_i^{\mathrm{k}} \in \mathbb{P}(i)$ could exist such that $P^{\mathrm{k}}\left(p_i^{\mathrm{k}+} \middle| i\right)$ is a controller K_i^+ that produces the same control inputs as K_{i+1} when it would have been applied to the (2-vertex) subset $X_i^+ \subseteq X_{i+1}$. This is precisely the less restrictive condition which, in Assumption 5.2, is captured formally in terms of sets of restrictions $\mathcal{R}(\cdot|\cdot,\cdot)$.

With Assumption 5.2 in place, it is already possible to establish recursive feasibility of a receding-horizon predictive controller based on (5.2). The corresponding proof is given together with the algorithm description and with the proof of closed-loop stability in Theorem 5.1 in Section 5.3.2. To prove closed-loop asymptotic stability, however, the following set of assumptions on the cost and value function is needed first.

Assumption 5.3. *In what follows, let* $K : X \times \Theta \to \mathbb{U}$ *be a controller.*

(i) There exist $\mathcal{K}_\infty$*-functions* $\underline{s}_F, \overline{s}_F$ *such that for all* $X \subseteq X_{\mathrm{f}}$, $\underline{s}_F\left(d_H^0(X)\right) \leq F(X) \leq \overline{s}_F\left(d_H^0(X)\right)$.

(ii) Let $K_\mathrm{f} : X_\mathrm{f} \times \Theta \to \mathbb{U}$ be the local controller from Assumption 5.1. Then, for all $X \subseteq X_\mathrm{f}$, it holds that $F\big(\mathcal{I}(X,\Theta|K_\mathrm{f})\big) - F\big(X\big) \leq -\ell\big(X, K_\mathrm{f}, \Theta\big)$.

(iii) For any set $X \subseteq X_\mathrm{f}$ and any subset $X' \subseteq X$, $F(X') \leq F(X)$. Further, let $\gamma \in [0,1]$ be the infimal γ such that $X \subseteq \gamma X_\mathrm{f}$. Then, $F(X) = F(\gamma X_\mathrm{f})$.

(iv) There exist $\mathcal{K}_\infty$-functions $\underline{s}_V, \overline{s}_V$ such that for all $x \in \mathbb{R}^{n_\mathrm{x}}$ and $\mathbf{\Theta} \subseteq \Theta^N$ for which (5.2) *is feasible, it holds $\underline{s}_V\left(\|x\|\right) \leq V\big(x, \mathbf{\Theta}\big|\mathcal{P}_N\big) \leq \overline{s}_V\left(\|x\|\right)$.*

Observe that Assumptions 5.3.(i)-(ii) are comparable to the "standard" set of assumptions on the terminal set and cost in stabilizing MPC (Mayne et al. 2000). The important difference here is that in the tube-based setting, the function $F(\cdot)$ takes a set as its argument and can be interpreted as being a Lyapunov-type function for set-valued dynamics.

Note that a terminal cost such that Assumption 5.3 is satisfied was already constructed explicitly in Chapter 4. However, in this chapter, a more general approach is taken by proving closed-loop stability under Assumption 5.3 alone without the need to construct a-priori the function $F(\cdot)$. A construction for the terminal cost is provided together with an implementable tube parameterization in Section 5.4.

From Definition 5.3 and Proposition 3.1.(i), the following result on the stage cost function follows directly:

Corollary 5.1. *For all subsets $X' \times \Theta' \subseteq X \times \Theta$, it holds that $\ell\big(X', K, \Theta'\big) = \ell\big(X', K', \Theta'\big) \leq \ell\big(X, K, \Theta\big)$ for any $K' \in \mathcal{R}\big(K|X', \Theta'\big)$.*

This result will also be useful in the proof of stability, which is presented as the main result of this chapter, in the following subsection.

5.3.2 Main result

The receding-horizon heterogeneously parameterized tube MPC algorithm is given in Algorithm 5.1. This algorithm is virtually the same as the prototype receding horizon implementation of Algorithm 3.1, except that in lines 5–6 the parameterization structure $\mathcal{P}_N$ is used.

In the next theorem, the main properties of Algorithm 5.1 are summarized and proven.

Theorem 5.1. *Let the hypotheses of Lemma 5.1 be satisfied. If, at the initial time instant k, there exists a solution $\mathbf{T}^\star = \left(\{X_i^\star\}_{i=0}^{N}, \{K_i^\star\}_{i=0}^{N-1}\right) \in \mathcal{T}_N\left(x(k), \mathbf{\Theta}\big|\mathcal{P}_N\right)$ to* (5.2)*, Algorithm 5.1 achieves the following properties:*

(i) After applying $u(k) = K_0^\star\left(x(k), \theta(k)\right)$ to the system, there exists a guaranteed feasible solution $\mathbf{T}^+ \in \mathcal{T}_N\big(x(k+1), \mathbf{\Theta}^+\big|\mathcal{P}_N\big)$.

(ii) The closed-loop trajectories of (2.4) *converge asymptotically to the origin, i.e., $x(k) \to 0$ as $k \to \infty$.*

Proof. Each statement is proven separately.
Proof of (i). From Definition 5.3, it follows that for any controller $K : X \times \Theta \to \mathbb{U}$ and for arbitrary subsets $(X' \times \Theta') \subseteq (X \times \Theta)$, we have that for all $K' \in \mathcal{R}\big(K\big|X', \Theta'\big)$ there holds

Algorithm 5.1 The receding-horizon HpTMPC algorithm.

Require: $N \in \mathbb{N}_{[1,\infty)}$ and a parameterization structure $\mathscr{P}_N$

$\mathbf{\Theta}_{-1} \leftarrow \Theta^N$
$k \leftarrow 0$
loop
  Construct $\mathbf{\Theta}_k = \left\{\{\theta(k)\}, \left\{\Theta_{i|k}\right\}_{i=1}^{N-1}\right\} \subseteq \Theta^N$ such that $\mathbf{\Theta}_k \sqsubseteq \mathbf{\Theta}_{k-1}$
  if $\mathscr{T}_N\left(x(k), \mathbf{\Theta}_k | \mathscr{P}_N\right) \neq \emptyset$ **then**
    Solve (5.2) to obtain $\mathbf{T}^\star \in \mathscr{T}_N\left(x(k), \mathbf{\Theta}_k | \mathscr{P}_N\right)$
    Apply $u(k) = K_0^\star\left(x(k), \theta(k)\right) = u_0^\star$ to the system (2.4)
    $k \leftarrow k + 1$
  else
    abort
  end if
end loop

$\mathscr{G}(X', \Theta' | K') = \mathscr{G}(X', \Theta' | K)$. Thus, with the sequence $\mathbf{X}^+ = \left\{X_i^+\right\}_{i=0}^{N}$ that is shown to exist in Lemma 5.1, it is possible to associate a sequence of restricted controllers $\mathbf{K}^+ = \left\{K_i^+\right\}_{i=0}^{N-1}$ where

$$
\begin{aligned}
\forall i \in \mathbb{N}_{[0,N-2]} : K_i^+ &\in \mathscr{R}\left(K_{i+1}^\star \middle| X_i^+, \Theta_i^+\right), \\
K_{N-1}^+ &\in \mathscr{R}\left(K_\mathrm{f} \middle| X_{N-1}^+, \Theta_{N-1}^+\right).
\end{aligned}
$$

Then, the successor tube

$$\mathbf{T}^+ = \left(\mathbf{X}^+, \mathbf{K}^+\right)$$

satisfies the initial condition and terminal constraints and Condition (i) of Definition 5.1. By Assumption 5.2 there exists at least one selection of $\mathbf{X}^+$ such that, under the given parameterization structure $\mathscr{P}_N$, conditions (ii)-(iii) of Definition 5.1 are also satisfied. Therefore, there exists a $\mathbf{T}^+ \in \mathscr{T}_N\left(x(k+1), \mathbf{\Theta}^+ \middle| \mathscr{P}_N\right)$.

Proof of (ii). To prove asymptotic closed-loop stability, the standard approach to show that the value function of (5.2) is a Lyapunov function for the closed-loop system will be used. Recall that $\mathbf{T}^\star = \left(\left\{X_i^\star\right\}_{i=0}^{N}, \left\{K_i^\star\right\}_{i=0}^{N-1}\right) \in \mathscr{T}_N\left(x(k), \mathbf{\Theta} \middle| \mathscr{P}_N\right)$ was the tube synthesized at the initial time instant k. Denote $\Delta V_k = V\left(x(k+1), \mathbf{\Theta}^+ \middle| \mathscr{P}_N\right) - V\left(x(k), \mathbf{\Theta} \middle| \mathscr{P}_N\right)$:

$$
\begin{aligned}
\Delta V_k &\leq J_N\left(\mathbf{T}^+, \mathbf{\Theta}^+\right) - J_N\left(\mathbf{T}^\star, \mathbf{\Theta}\right) \\
&= \sum_{i=0}^{N-1} \ell\left(X_i^+, K_i^+, \Theta_i^+\right) - \sum_{i=0}^{N-1} \ell\left(X_i^\star, K_i^\star, \Theta_i\right) + F\left(X_N^+\right) - F\left(X_N^\star\right).
\end{aligned} \tag{5.5}
$$

Using (5.4a)-(5.4c), the order $\mathbf{\Theta}^+ \sqsubseteq \mathbf{\Theta}$, and Proposition 3.1.(i) gives that

$$\sum_{i=0}^{N-1} \ell\left(X_i^+, K_i^+, \Theta_i^+\right) \leq \sum_{i=1}^{N-1} \ell\left(X_i^\star, K_i^\star, \Theta_i\right) + \ell\left(X_{N-1}^+, K_{N-1}^+, \Theta_{N-1}^+\right)$$

and substituting this in (5.5) yields

$$\begin{aligned}\Delta V_k &\le \sum_{i=1}^{N-1} \ell\left(X_i^\star, K_i^\star, \Theta_i\right) + \ell\left(X_{N-1}^+, K_{N-1}^+, \Theta_{N-1}^+\right) - \sum_{i=0}^{N-1} \ell\left(X_i^\star, K_i^\star, \Theta_i\right) \\ &\quad + F\left(X_N^+\right) - F\left(X_N^\star\right) \\ &= -\ell\left(X_0^\star, K_0^\star, \Theta_0\right) + \ell\left(X_{N-1}^+, K_{N-1}^+, \Theta_{N-1}^+\right) + F\left(X_N^+\right) - F\left(X_N^\star\right).\end{aligned} \tag{5.6}$$

It is known that both $X_{N-1}^+ \subseteq \gamma X_\mathrm{f}$ and $X_N^\star \subseteq \gamma X_\mathrm{f}$, so $K_{N-1}^+ \in \mathcal{R}(K_\mathrm{f}|X_{N-1}^+, \Theta_{N-1}^+) \subseteq \mathcal{R}(K_\mathrm{f}|\gamma X_\mathrm{f}, \Theta_{N-1}^+)$. Invoking Proposition 3.1.(i) then gives $\ell(X_{N-1}^+, K_{N-1}^+, \Theta_{N-1}^+) \le \ell(\gamma X_\mathrm{f}, K_\mathrm{f}, \Theta_{N-1}^+)$. Assumption 5.3.(iii) yields $F(X_N^\star) = F(\gamma X_\mathrm{f})$ and since $X_N^+ \subseteq \lambda\gamma X_\mathrm{f}$, also $F(X_N^+) \le F(\lambda\gamma X_\mathrm{f}) = F(\mathcal{S}(\gamma X_\mathrm{f}, \Theta|K_\mathrm{f}))$. Substituting all of this in (5.6) leads to

$$\Delta V_k \le -\ell\left(X_0^\star, K_0^\star, \Theta_0\right) + \ell\left(\gamma X_\mathrm{f}, K_\mathrm{f}, \Theta_{N-1}^+\right) + F\left(\mathcal{S}(\gamma X_\mathrm{f}, \Theta|K_\mathrm{f})\right) - F\left(\gamma X_\mathrm{f}\right),$$

and using the fact that $\Theta_{N-1}^+ \subseteq \Theta$ subsequently gives

$$\Delta V_k \le -\ell\left(X_0^\star, K_0^\star, \Theta_0\right) + \ell\left(\gamma X_\mathrm{f}, K_\mathrm{f}, \Theta\right) + F\left(\mathcal{S}(\gamma X_\mathrm{f}, \Theta|K_\mathrm{f})\right) - F\left(\gamma X_\mathrm{f}\right).$$

From Proposition 3.1.(ii) and Assumption 5.3.(ii) it follows finally that

$$\Delta V_k \le -\ell\left(X_0^\star, K_0^\star, \Theta_0\right) \le -\underline{s}_\ell\left(d_H^0\left(X_0^\star\right)\right) = -\underline{s}_\ell\left(\|x(k)\|\right).$$

This, in conjunction with Assumption 5.3.(iv), is sufficient to conclude that $V(\cdot,\cdot|\mathcal{P}_N)$ is a (regional) Lyapunov function for the closed-loop system according to Definition 2.9. As established in Lemma 2.2, the existence of such a function implies that the origin is a regionally asymptotically stable equilibrium of the closed-loop system. Therefore, $x(k) \to 0$ as $k \to \infty$. □

The properties proven in Theorem 5.1 amount to recursive feasibility and regional asymptotic closed-loop stability of the origin. Hence, Algorithm 5.1 possesses the desirable property of "stability" in the sense of Definition 3.5 (i.e., asymptotic convergence of the state of the closed-loop system to the origin). It is emphasized that this result was obtained on the basis of general conditions alone, without employing explicit constructions of tube parameterization, successor tube, and terminal cost. This is different from the situation in Chapter 4 where specific constructions were considered.

Whether or not the tube synthesis problem (5.2) is computationally tractable, depends on the specific choice of the parameterization structure $\mathcal{P}_N$. One possible choice that leads to tractable problems is presented in the next section.

5.4 Parameterization construction

In this section, constructions are provided that enable the implementation of Algorithm 5.1. First, in Section 5.4.1, a terminal cost that satisfies the necessary assumptions from the previous section is proposed. Then, Section 5.4.2 presents a heterogeneous parameterization structure that leads to a computationally tractable tube synthesis optimization problem.

5.4.1 The HpT cost function

This subsection proposes a terminal cost function such that Assumption 5.3 is satisfied, so it is possible to obtain a stabilizing MPC according to Theorem 5.1. The terminal cost is defined in terms of the set-gauge $\Psi_{X_\mathrm{f}}(\cdot)$ (see Definition 2.2) corresponding to X_f as

$$F(X) = \frac{\bar{\ell}}{1-\lambda^c}\Psi_{X_\mathrm{f}}^c(X), \tag{5.7}$$

where c has the same value as in (3.13), X_f and λ are according to Assumption 5.1, and where

$$\bar{\ell} = \ell(X_\mathrm{f}, K_\mathrm{f}, \Theta) \tag{5.8}$$

is a constant. This terminal cost is very similar to the cost that was considered previously in Chapter 4. However, Chapter 4 considered the case where the norm in (3.13) is the infinity-norm and where $c = 1$, but for a periodic tube parameterization. In contrast, here the tube parameterization is non-periodic, but the norm can be arbitrary and it is allowed that $c \geq 1$.

When the 2-norm is used together with $c = 2$, the cost (3.13) corresponds to a quadratic cost because $\|Mv\|_2^2 = v^\top\left(M^\top M\right)v$.

In the next proposition, it is shown that the cost functions (3.13) and (5.7) satisfy Assumption 5.3, and therefore lead to a stable closed-loop system.

Proposition 5.1. *The stage and terminal cost* (5.7) *together satisfy Assumptions 5.3.(i)-(iii). Furthermore, the value function $V(\cdot,\cdot|\mathcal{P}_N)$ is $\mathcal{K}_\infty$-bounded in the sense of Assumption 5.3.(iv).*

Proof. Satisfaction of Assumption 5.3.(i). Because (5.7) is simply the set-gauge function of the PC-set X_f raised to the power $c \geq 1$ and multiplied with a constant scalar factor, this property follows directly from the existence of the bounds $\underline{s}_\psi, \overline{s}_\psi$ stated in Lemma 2.1.
Satisfaction of Assumption 5.3.(ii). By λ-contractivity of X_f (Assumption 5.1.(i)), $\Psi_{X_\mathrm{f}}\left(\mathcal{G}(X,\Theta|K_\mathrm{f})\right) \leq \lambda\Psi_{X_\mathrm{f}}\left(X\right)$. Also, because of the homogeneity of K_f (Assumption 5.1.(ii)), the stage cost (3.13) is positively homogeneous of degree c in the sense that for any $X \subseteq X_\mathrm{f}$,

$$\ell(X, K_\mathrm{f}, \Theta) \leq \ell\left(\Psi_{X_\mathrm{f}}(X)X_\mathrm{f}, K_\mathrm{f}, \Theta\right) = \Psi_{X_\mathrm{f}}^c(X)\underbrace{\ell(X_\mathrm{f}, K_\mathrm{f}, \Theta)}_{\bar{\ell}}.$$

Thus we have the inequality

$$\begin{aligned} F(\mathcal{G}(X,\Theta|K_\mathrm{f})) - F(X) &= \frac{\bar{\ell}}{1-\lambda^c}\left(\Psi_{X_\mathrm{f}}^c(\mathcal{G}(X,\Theta|K_\mathrm{f})) - \Psi_{X_\mathrm{f}}^c(X)\right) \\ &\leq \frac{\bar{\ell}}{1-\lambda^c}\left(\lambda^c\Psi_{X_\mathrm{f}}^c(X) - \Psi_{X_\mathrm{f}}^c(X)\right) \\ &= -\bar{\ell}\Psi_{X_\mathrm{f}}^c(X) \\ &\leq -\ell(X, K_\mathrm{f}, \Theta), \end{aligned}$$

which was to be proven.
Satisfaction of Assumption 5.3.(iii). This property is immediate from the definition of $\Psi_{X_\mathrm{f}}(\cdot)$, see Definition 2.2.
Satisfaction of Assumption 5.3.(iv). Under the three assumptions that have been proven above, this property has been shown to hold in Proposition 3.2. □

5.4.2 The HpT-SF parameterization

In this subsection, a heterogeneous parameterization that satisfies Assumption 5.2 is proposed. In other words, this subsection presents one possible design of the parameterization structure $\mathcal{P}_N$ in Definition 5.2. The focus is on developing an implementable parameterization leading to a convex finite-dimensional optimization problem (5.2). For this reason, it should be avoided that non-convex constraints arise due to the multiplication of a parameter-dependent input matrix with a controller (i.e., a decision variable) which is dependent on the same scheduling parameter. This is guaranteed by the following assumption.

Assumption 5.4. *Assume that for all k, every set in the scheduling tube constructed in Step 4 of Algorithm 5.1 is a polytope. Denote $K_i = P^{\mathrm{k}}\left(p_i^{\mathrm{k}}|i\right)$ for some $p_i^{\mathrm{k}} \in \mathbb{P}(i)$. Let the elements of Θ be partitioned as*

$$\theta = \begin{bmatrix} \tilde{\theta}_1 \\ \tilde{\theta}_2 \end{bmatrix}, \text{ where } \tilde{\theta}_1 \in \tilde{\Theta}_1 \subseteq \mathbb{R}^{n_{\theta 1}},\ \tilde{\theta}_2 \in \tilde{\Theta}_2 \subseteq \mathbb{R}^{n_{\theta 2}},\ n_{\theta 1} + n_{\theta 2} = n_\theta,$$

with $\tilde{\Theta}_1$ and $\tilde{\Theta}_2$ being the projections of Θ onto the first $n_{\theta 1}$ and last $n_{\theta 2}$ dimensions, respectively. Assume that the structure $\mathcal{P}_N$ is such that for all $i \in \mathbb{N}_{[1,N-1]}$ and for all $\left(p_i^{\mathrm{k}}, \theta\right) \in \mathbb{P}(i) \times \Theta$ the product $B(\theta)K_i(x,\theta)$ can be written as $B(\theta)K_i(x,\theta) = B(\tilde{\theta}_1)K_i(x,\tilde{\theta}_2)$.

Essentially, Assumption 5.4 requires that if the input matrix $B(\cdot)$ of the LPV-PSS representation depends on a scheduling variable θ_1, then the synthesized controllers are independent of θ_1. They are, however, allowed to depend on any other scheduling variable θ_2 that is not a dependency of $B(\cdot)$. Two straightforward possibilities that satisfy this condition are the following:

Remark 5.2. *Two possibilities that guarantee satisfaction of Assumption 5.4 are the following:*

(i) The input matrix of (2.4) *is constant, i.e.,* $\forall \theta \in \Theta :\ B(\theta) = B$.

(ii) The structure $\mathcal{P}_N$ is such that the controllers $K_i : X_i \times \Theta_i \to \mathbb{U}$ are completely independent of θ.

The two options of Remark 5.2 represent two “extreme” cases where either $n_{\theta 1} = 0$ or $n_{\theta 2} = 0$; Option (i) has been used in Chapter 4. However, Assumption 5.4 can also include intermediate cases, where both $n_{\theta 1} > 0$ and $n_{\theta 2} > 0$ (and where, consequently, both $\tilde{\Theta}_1$ and $\tilde{\Theta}_2$ are non-empty sets).

Under Assumption 5.4, a so-called *full scenario tube* can be described in the HpT framework of Definition 5.1.

Definition 5.4 (HpT-S: scenario tube)**.** *A tube $\mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta}|\mathcal{P}_N)$ is a scenario tube or HpT-S if $\forall i \in \mathbb{N}_{[0,N-1]} : X_{i+1} = \mathcal{I}(X_i, \Theta_i|K_i)$ and each K_i is parameterized as a vertex controller on the set $X_i \times \Theta_i$. Let $q_{\mathrm{x}}(i) = q_\theta^{\max\{0,i-1\}}$ and $q_{\mathrm{k}}(i) = q_\theta^i$. Under Assumption 5.4, the parameterization structure $\mathcal{P}_N$ of an HpT-S equivalently satisfies*

(i) For all $i \in \mathbb{N}_{[0,N]}$, it holds $\mathbb{P}^{\mathrm{x}}(i) = \mathbb{R}^{n_{\mathrm{x}} q_{\mathrm{x}}(i)}$, $p_i^{\mathrm{x}} = \left(\bar{x}_i^1, \ldots, \bar{x}_i^{q_{\mathrm{x}}(i)}\right)$, and $P^{\mathrm{x}}\left(p^{\mathrm{x}}|i\right) =$ $\mathrm{convh}\left\{p^{\mathrm{x}}\right\}$.

	Parameterization	DOF
1	$K_i(x,\theta) = c_i + K_\mathrm{f}(x - z_i, \theta)$	1
2	$K_i(x,\theta) = c_i(\theta) + K_\mathrm{f}(x - z_i, \theta)$	q_θ
3	$K_i(x,\theta) = \mathrm{vertpol}\left(p_i^\mathrm{k} \middle\vert X_i, \Theta_i\right)(x,\theta)$	$q_\theta q_\mathrm{s}$

Table 5.1: Some possible control parameterizations that fit in the framework of Definition 5.5. One control DOF corresponds to n_u decision variables in the tube synthesis problem.

(ii) For all $i \in \mathbb{N}_{[0,N-1]}$, it holds $\mathbb{P}^\mathrm{k}(i) = \mathbb{R}^{n_\mathrm{u} q_\mathrm{k}(i)}$, $p_i^\mathrm{k} = \left(\bar{u}_i^1, \ldots, \bar{u}_i^{q_\mathrm{k}(i)}\right)$, and $P^\mathrm{k}\left(p^\mathrm{k} \vert i\right) =$ $\mathrm{vertpol}\left(p^\mathrm{k} \vert X_i, \Theta_i\right)$.

An HpT-S is non-conservative, but the number of vertices of the sets X_i increases exponentially, as each set has q_θ times more vertices than the preceding set. In (Lucia et al. 2013; Maiworm et al. 2015; Muñoz de la Peña et al. 2006), it is proposed to avoid this explosive growth by assuming that the uncertainty resolves after N_0 prediction time instances, at which point the tree stops branching. If this assumption is not met in reality, the scheme loses its feasibility and stability properties. Here the exponential growth is avoided differently: namely, by switching from a scenario parameterization to a parameterization with fixed-complexity cross sections after N_0 prediction steps. (This idea is illustrated in Figure 5.1, which will be explained later in more detail.) In this way, theoretical feasibility and stability guarantees are retained under a more realistic handling of the uncertainty. The specific fixed-complexity parameterization which is employed is defined next.

Definition 5.5 (HpT-F: fixed-cross section heterogeneously parameterized tube)**.** *A tube $\mathbf{T} \in \mathcal{T}_N(x, \boldsymbol{\Theta} \vert \mathcal{P}_N)$ is an HpT-F if for all $i \in \mathbb{N}_{[0,N-1]}$, $X_i = z_i \oplus \alpha_i X_\mathrm{f}$ where $\alpha_i \in \mathbb{R}_+$ is the corresponding* cross-section scaling *and $z_i \in \mathbb{R}^{n_\mathrm{x}}$ is the* cross-section center. *Equivalently, in terms of Definition 5.1, the parameterization structure $\mathcal{P}_N$ of an HpT-F satisfies that*

(i) For all $i \in \mathbb{N}_{[0,N]}$, it holds $\mathbb{P}^\mathrm{x}(i) = \mathbb{R}^{1+n_\mathrm{x}}$, $p_i^\mathrm{x} = (z_i, \alpha_i)$, and $P^\mathrm{x}\left(p^\mathrm{x} \middle\vert i\right) = z \oplus \alpha X_\mathrm{f}$.

(ii) For $i \in \mathbb{N}_{[1,N-1]}$, the sets $\mathbb{P}^\mathrm{k}(i)$ and functions $P^\mathrm{k}(\cdot \vert i)$ are such that if $\exists p_i^\mathrm{k} \in \mathbb{P}^\mathrm{k}(i)$ with $P^\mathrm{k}\left(p_i^\mathrm{k} \middle\vert i\right) = K$, then $\exists p_{i-1}^\mathrm{k} \in \mathbb{P}^\mathrm{k}(i-1)$ satisfying $P^\mathrm{k}\left(p_{i-1}^\mathrm{k} \middle\vert i-1\right) = K$.

In the HpT-F, as the cross-sections X_i are scaled and translated versions of the same set X_f, it is said that the cross sections are *homothetic* to X_f. In contrast to the previous formulation of homothetic LPV TMPC from Section 3.5, the parameterization of the associated control laws is allowed to be time-varying along the prediction horizon, i.e., the sets and functions $(\mathbb{P}^\mathrm{k}(\cdot), P(\cdot\vert\cdot))$ are dependent on i. Thus, an HpT-F satisfying 5.5 can still be called "heterogeneous".

The condition of Definition 5.5.(ii) is required to ensure that an HpT-F satisfies Assumption 5.2.(ii)-(iii), so that it leads to a recursively feasible MPC. In Table 5.1, some control parameterizations are listed for the purpose of illustration.

Next, the HpT-S and HpT-F are combined into a single tube. Such a tube which will be called an HpT-SF (where "SF" stands for "scenario/fixed-complexity").

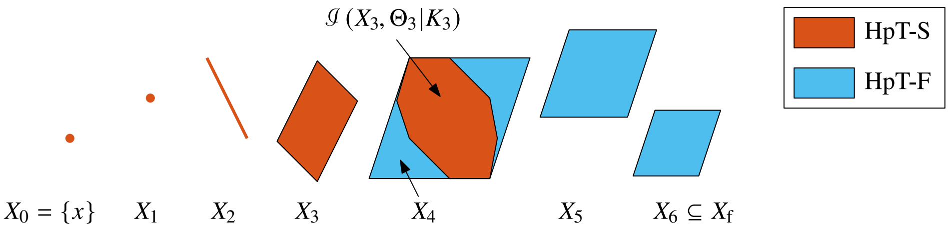


Figure 5.1: Example of a HpT-SF prediction structure in a 2-dimensional state space ($N = 6$).

Definition 5.6 (HpT-SF)**.** *Let $N \in \mathbb{N}_{[2,\infty)}$ and $N_0 \in \mathbb{N}_{[1,N]}$. A heterogeneously parameterized tube $\mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta}|\mathcal{P}_N)$ satisfying Definition 5.1 is an HpT-SF, if it can be decomposed as $\mathbf{T} = (\mathbf{T}^0, \mathbf{T}^1)$ with*

$$\begin{aligned} \mathbf{T}^0 &= \left(\left\{ X_i^0 \right\}_{i=0}^{N_0-1}, \left\{ K_i^0 \right\}_{i=0}^{N_0-1} \right), \\ \mathbf{T}^1 &= \left(\left\{ X_i^1 \right\}_{i=N_0}^{N}, \left\{ K_i^1 \right\}_{i=N_0}^{N-1} \right), \end{aligned} \tag{5.9}$$

where $\mathbf{T}^0$ is an HpT-S according to Definition 5.4 and $\mathbf{T}^1$ is an HpT-F according to Definition 5.5.

In the above definition, since both sections $\mathbf{T}^0$ and $\mathbf{T}^1$ are heterogeneously parameterized according to Definition 5.1, it follows directly that the same holds for the complete tube $\mathbf{T} = (\mathbf{T}^0, \mathbf{T}^1)$.

To clarify the concept, a graphical representation of an example HpT-SF is given in Figure 5.1. For the first prediction time instances, the HpT-S structure is employed. It can be observed that the number of vertices of these cross sections doubles at every prediction step after the first one. Then, at prediction step $N_0 = 4$ a transition is made to the HpT-F structure: from this point forwards, the complexity of the tube cross sections remains constant as they are all homothetic to the same polytope.

The next proposition confirms that the HpT-SF prediction structure satisfies Assumption 5.2, and therefore leads to a recursively feasible MPC as proven in Theorem 5.1.

Proposition 5.2. *Suppose Assumption 5.4 is satisfied. Then, the HpT-SF structure of Definition 5.6 satisfies Assumption 5.2.(i).*

Proof. First, consider the HpT-S part $\mathbf{T}^0$. In reference to Lemma 5.1, recall that $X_0^+ \subseteq X_1$ is a singleton. Applying the first construction of Remark 5.1 for $i \in \mathbb{N}_{[0,N_0-1]}$ (and noting that $\Theta_0^+ = \{\theta(k+1)\}$), gives a sequence $\left\{ X_i^+ \right\}_{i=0}^{N_0-1}$ where the amount of vertices of the i-th set equals $q_{\mathrm{x}}(i) = q_\theta^{\max\{0,i-1\}}$. This is in accordance with Definition 5.4: hence, for all $i \in \mathbb{N}_{[0,N_0-1]}$ there exist $p_i^{\mathrm{x}+}$: $P^{\mathrm{x}}\left(p_i^{\mathrm{x}+}\middle|i\right) = X_i^+$. Next, for all $i \in \mathbb{N}_{[0,N_0-1]}$, the restricted controllers $K_i^+ \in \mathcal{R}\left(K_{i+1}\middle|X_i^+, \Theta_i^+\right)$ are the vertex controllers on the sets X_i^+, which also agrees with Definition 5.4. Because $X_i^+ \subseteq X_{i+1}$, the corresponding vertex control actions in $p_i^{\mathrm{k}+} \in \mathbb{P}(i)$ can be taken as convex combinations of the elements of $p_{i+1}^{\mathrm{k}} \in \mathbb{P}(i+1)$. Therefore, for all $i \in \mathbb{N}_{[0,N_0-1]}$, the existence of $p_i^{\mathrm{k}+} \in \mathbb{P}^{\mathrm{k}}(i)$ such that $P\left(p_i^{\mathrm{k}+}\middle|i\right) = K_i^+$ is guaranteed.

Second, consider the HpT-F part $\mathbf{T}^1$. Applying the second construction of Remark 5.1 for $i \in \mathbb{N}_{[N_0,N-1]}$ gives the sequence $\{X_i^+\}_{i=N_0}^{N}$ where $\{X_i^+\}_{i=N_0}^{N-2} = \{X_i\}_{i=N_0+1}^{N-1}$, $X_{N-1}^+ = \gamma X_{\mathrm{f}}$, and $X_N^+ = \lambda\gamma X_{\mathrm{f}}$. Because all sets in $\{X_i^+\}_{i=N_0}^{N}$ can be represented as $X_i^+ = z_i \oplus \alpha_i X_{\mathrm{f}}$, it follows immediately from Definition 5.5 that for all $i \in \mathbb{N}_{[N_0,N]}$, there exists a $p_i^{\mathrm{x}+}$: $P^{\mathrm{x}}(p_i^{\mathrm{x}+}|i) = X_i^+$. From this construction, it also follows that for $i \in \mathbb{N}_{[N_0,N-2]}$, $K_i^+ = K_{i+1} \in \mathcal{R}(K_{i+1}|X_i^+, \Theta_i^+)$, and that $K_{N-1}^+ \in \mathcal{R}(K_{\mathrm{f}}|\gamma X_{\mathrm{f}}, \Theta_{N-1}^+)$. Thus, Definition 5.5 guarantees that for all $i \in \mathbb{N}_{[N_0,N-1]}$ there exist $p_i^{\mathrm{k}+}$: $P(p_i^{\mathrm{k}+}|i) = K_i^+$. This concludes the proof. □

The HpT-F part in the HpT-SF structure can be implemented similarly to the implementation of Section 4.4.2. Hence, for fixed N_0, the number of variables and constraints in the tube synthesis (5.2) will be in the order of $O(N)$. Therefore:

Corollary 5.2. *Using the HpT-SF parameterization, for fixed N_0, the tube synthesis problem* (5.2) *is computationally tractable in the sense of Definition 3.6.*

On the other hand, for variable N_0, the complexity of the tube synthesis problem is in the order $O(q_\theta^{N_0})$. Therefore, a remaining question is how to design N_0. To obtain the least conservative control law for a given prediction horizon N, one can choose $N_0 \leq N$ as large as computational resources reasonably allow. Another approach is to compare the complexity of an HpT-S of length N_0 with the complexity of an HpT-F of the same length. Suppose that the shape set X_{f} from Definition 5.5 has q_{f} vertices. The last cross-section X_{N_0-1} of an HpT-S of length N_0 has $q_\theta^{N_0-2}$ vertices. Thus it is possible to select N_0 such that $q_\theta^{N_0-2} \approx q_{\mathrm{f}}$, i.e.,

$$N_0 = \operatorname{round}\left(\frac{\log q_{\mathrm{f}}}{\log q}\right) + 2. \tag{5.10}$$

The value of N_0 of (5.10) represents the approximate value where the complexity—in terms of the number of vertices of the tube cross sections—of the HpT-S part starts to grow beyond that of the HpT-F part, implying that it is computationally advantageous to switch to the HpT-F structure after N_0 prediction steps.

5.5 Numerical examples

In this section, two numerical examples are provided to demonstrate the heterogeneously parameterized TMPC (HpTMPC) algorithm. It is shown how different choices in the heterogeneous parameterization structure affect properties such as size of the DOA and computation time.

5.5.1 Parameter-varying double integrator

In this example, the following LPV state-space (LPV-SS) representation is considered

$$x(k+1) = \left(\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix} \theta_1(k) + \begin{bmatrix} 0.5 & 0.5 \\ 0 & 0 \end{bmatrix} \theta_2(k) + \begin{bmatrix} 0 & 0 \\ 0 & 0.2 \end{bmatrix} \theta_3(k) \right) x(k) + \begin{bmatrix} 0.5 \\ 1 \end{bmatrix} u(k)$$

with the constraint and scheduling sets given as

$$\begin{aligned} \mathbb{X} &= \left\{ x \in \mathbb{R}^2 \,\middle|\, \|x\|_\infty \leq 6 \right\}, \\ \mathbb{U} &= \{ u \in \mathbb{R} \mid |u| \leq 1 \}, \\ \Theta &= \left\{ \theta \in \mathbb{R}^3 \,\middle|\, \|\theta\|_\infty \leq 1 \right\}. \end{aligned}$$

The set Θ is a hypercube in 3 dimensions and therefore has 8 vertices.

The purpose of this example is to demonstrate the effect of changing the heterogeneous parameterization structure. Therefore, all other design parameters are set to constant values based on the following choices:

- The prediction horizon is set to $N = 10$.
- The scheduling tubes $\boldsymbol{\Theta} \subseteq \Theta^N$ are constructed such that $\boldsymbol{\Theta}_k = \{\{\theta(k)\}, \Theta, \ldots, \Theta\}$ for all k. In this way, attention is focused on the effect of changing the heterogeneous parameterization structure, and not on different possible ways of constructing a scheduling tube.
- The tuning parameters were set to $Q = I$ and $R = 1$.
- The terminal set X_f was computed to be 0.95-contractive with respect to a robust linear time-invariant (LTI) terminal controller $K_\mathrm{f}(x, \theta) = K_\mathrm{f} x$. This set was computed by Algorithm 2.1 and is described by 10 vertices. The restriction to a robust LTI terminal controller in this case yields a set X_f with a relatively small volume, but also with a relatively low number of vertices. Furthermore, in this example, the relatively small volume of the set allows to illustrate more clearly the effect of the tube parameterization on the DOA of the resulting controllers.
- The maximal "worst-case" DOA is $\mathcal{X}_N^0 = \mathcal{X}_N^{\max}(\boldsymbol{\Theta}^{\mathrm{worst}})$ where $\boldsymbol{\Theta}^{\mathrm{worst}} = \{\Theta, \ldots, \Theta\}$ (see Section 3.4.4 for the definition of maximal DOA). This corresponds to the largest possible DOA that can be achieved by *any* controller.

Now, three tube-based controllers based on different parameterization structures $\mathcal{P}_N$ are compared in terms of the achieved DOA and the number of required control DOFs:

Design 1 Homothetic tube with vertex controls, i.e., parameterization 3 from Table 5.1. Identical to the controller developed in Section 3.5.

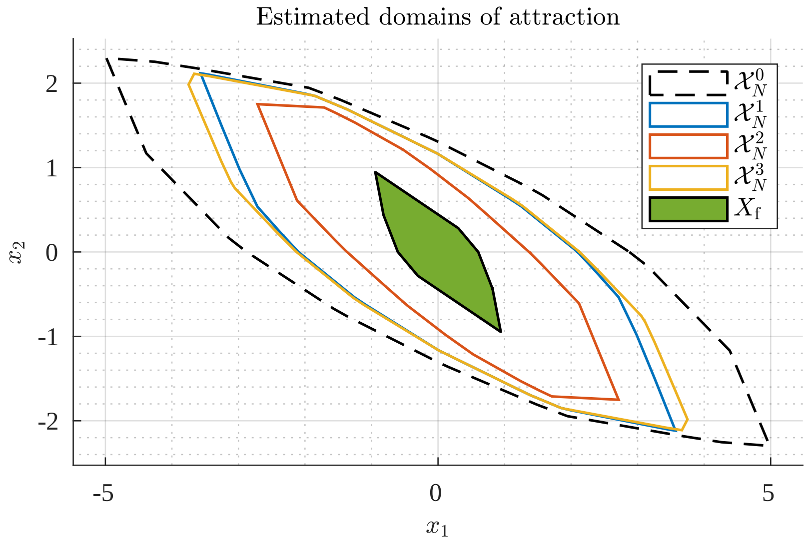


Figure 5.2: Realized (estimated) DOAs for the three designs in Example 1. Here, $\mathcal{X}_N^0 = \mathcal{X}_N^{\max}(\mathbf{\Theta}^{\text{worst}})$, whereas $\mathcal{X}_N^i$ is the DOA corresponding to Design i for $i \in \{1, 2, 3\}$.

	DOA vol.	DOF
Design 1 (homothetic, vertex controls)	12.5	$1 + (N-1)q_\theta q_s = 721$
Design 2 (homothetic, simple controls)	7.51	$N = 10$
Design 3 (heterogeneous)	13.2	$(1 + q_\theta + q_\theta^2) + 3q_\theta q_s + 4 = 317$

Table 5.2: Comparison of the three designs in Example 1.

Design 2 Homothetic tube with "simple" control parametrization, namely parameterization 1 from Table 5.1.

Design 3 A heterogeneous design, consisting of a scenario three for the first 3 prediction time instances, a homothetic tube with vertex control parameterization for the next 3 instances, and a homothetic tube with the simple control parameterization 1 of Table 5.1 for the remaining prediction steps.

It is noted that constructing a full scenario tree of length $N = 10$ for this system would be intractable: the deepest level of the tree would have $8^{10} \approx 10^9$ nodes.

The realized DOAs for the three designs were estimated using the method of Appendix B.2 and are displayed in Figure 5.2. These DOAs are all "worst-case" in the sense that they are computed with respect to a scheduling tube $\mathbf{\Theta}^{\text{worst}} = \{\Theta, \dots, \Theta\}$. The associated set volumes and the number of control DOF are displayed in Table 5.2.

Example closed-loop trajectories for an initial state at which all the three designs are feasible is shown in Figure 5.3, and the corresponding scheduling trajectory is depicted in Figure 5.4.

	Avg. time [ms]	Max. time [ms]
Design 1 (homothetic, vertex controls)	55	69
Design 2 (homothetic, simple controls)	42	45
Design 3 (heterogeneous)	37	44

Table 5.3: Illustration of complexity: solver time per sample in the simulation of Example 1.

CPU	Intel Core i7-4790 @ 3.60 GHz
Memory	8 GB
Operating system	GNU/Linux 64-bit
MATLAB version	9.3.0.713579 (R2017b)
LP solver	Gurobi 7.0.2

Table 5.4: Computer details.

For this initial state, the realized closed-loop inputs are slightly different, but the resulting state trajectories are virtually indistinguishable. This confirms the idea that when a small feasible set is sufficient, one could use a simple controller, but that more complex parameterizations become necessary whenever the DOA is to be enlarged.

For further illustration of the resulting computational complexity of the different designs, a summary of the computation times for the closed-loop simulations is given in Table 5.3. Note that Design 2 has only 10 DOF, but that the complexity of the tube synthesis problem is dominated by the number of constraints necessary to verify the tube set inclusions in that case. The simulations were executed on a computer with the specifications listed in Table 5.4. These figures (especially for Designs 2 and 3) could be further improved by optimizing the linear program (LP) implementation of the tube synthesis problem: the straightforward implementation that was used here, for instance, generates many redundant constraints when using the simple control parameterizations 1 or 2 from Table 5.1.

As a closing remark, it is emphasized that the DOAs in this example were computed with respect to all possible initial scheduling values $\theta \in \Theta$. For a specific measured initial scheduling value $\theta(k)$, the tube synthesis might also be feasible for initial states outside of these domains. Also, the scheduling tubes were constructed in a "worst-case" sense: whenever anticipative knowledge is available to construct refined scheduling tubes $\boldsymbol{\Theta} \subseteq \Theta^N$, the size of the set of stabilizable states can become larger.

5.5.2 Parameter-varying third-order system

In this example, a third-order system is considered. Its dynamics are described by the difference equation

$$x(k+1) = \left(I + \tau \begin{bmatrix} 0 & 1 & 0 \\ -0.7\theta_1(k) & -0.4 & 0.2 \\ 0 & -0.3 & -0.1\theta_2(k) \end{bmatrix} \right) x(k) + \tau \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k)$$

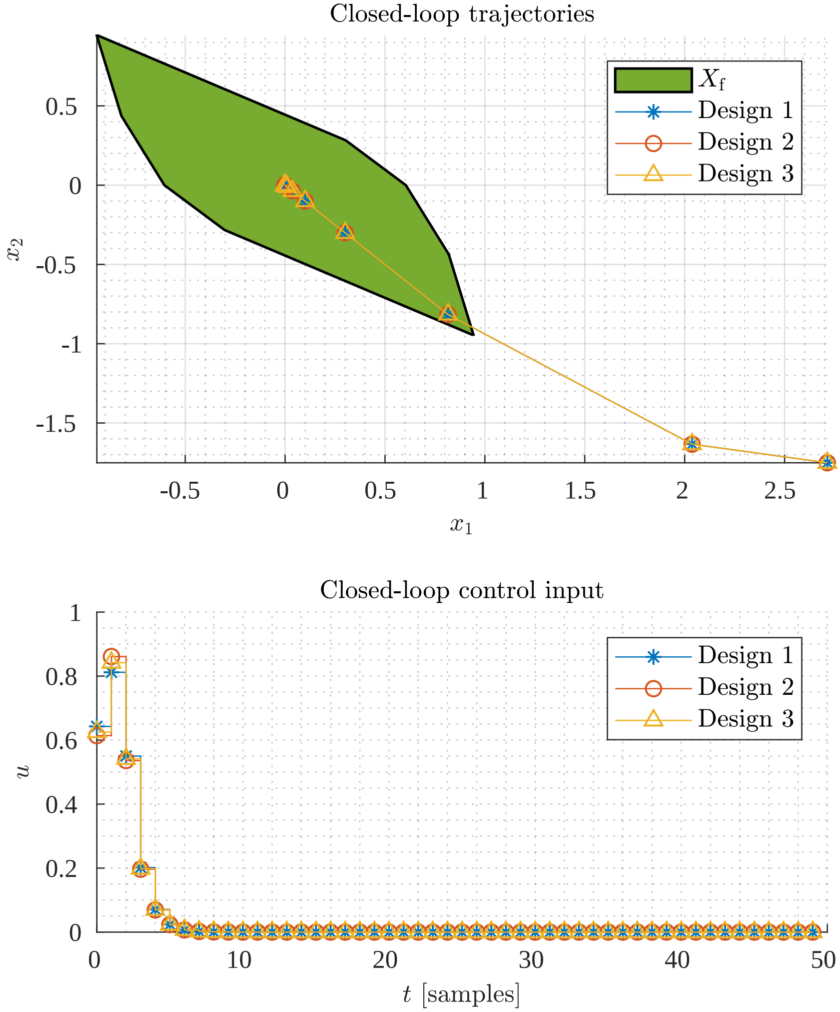


Figure 5.3: Illustrative closed loop trajectories for Example 1.

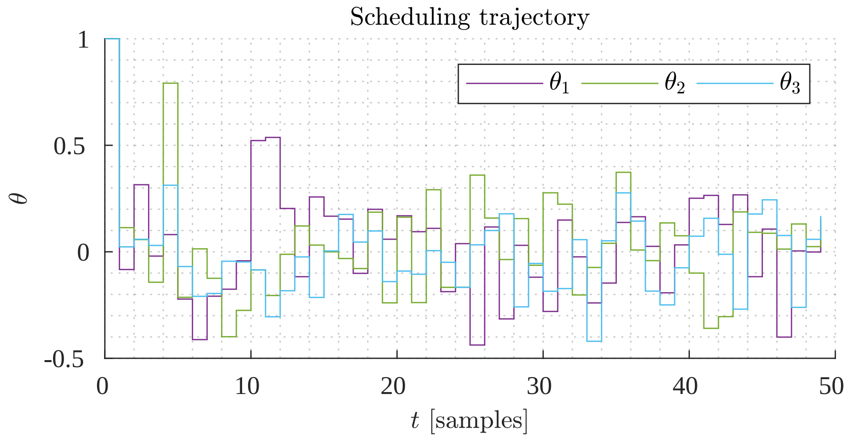


Figure 5.4: Used scheduling trajectory in Example 1.

and the constraint and scheduling sets are

$$\begin{aligned}
\mathbb{X} &= \left\{x \in \mathbb{R}^3 \,\middle|\, |x_1| \leq 0.5, |x_2| \leq 0.1, |x_3| \leq 0.2\right\}, \\
\mathbb{U} &= \{u \in \mathbb{R} \,|\, |u| \leq 0.2\}, \\
\Theta &= \left\{\theta \in \mathbb{R}^2 \,\middle|\, \theta_1 \in [0.5, 1.5], \theta_2 \in [0.8, 1.2]\right\}.
\end{aligned}$$

In the simulations that follow, the sampling time $\tau = 0.36$ s was used. Similar to the previous example, all design parameters except the parameterization structure are kept at constant values. The following choices were made:

- The prediction horizon was set to $N = 8$.
- The scheduling tubes $\boldsymbol{\Theta} \subseteq \Theta^N$ were constructed such that $\boldsymbol{\Theta}_k = \{\{\theta(k)\}, \Theta, \ldots, \Theta\}$ for all k. As in the previous example, this means that attention is focused on the effect of changing the heterogeneous parameterization structure, and not on different possible ways of constructing a scheduling tube.
- The tuning parameters were set to $Q = I$ and $R = 5$.
- The terminal set X_f was computed—using Algorithm 2.1—to be 0.98-contractive with respect to a robust LTI terminal controller $K_\mathrm{f}(x, \theta) = K_\mathrm{f} x$. This set is described by 48 vertices and, equivalently, by 28 hyperplanes.

Now, similarly to the previous example, three tube-based controllers based on different parameterization structures $\mathscr{P}_N$ are compared in terms of the achieved DOA and the number of required control DOFs:

Design 1 Homothetic tube with vertex controls, i.e., parameterization 3 from Table 5.1. Identical to the controller developed in Section 3.5.

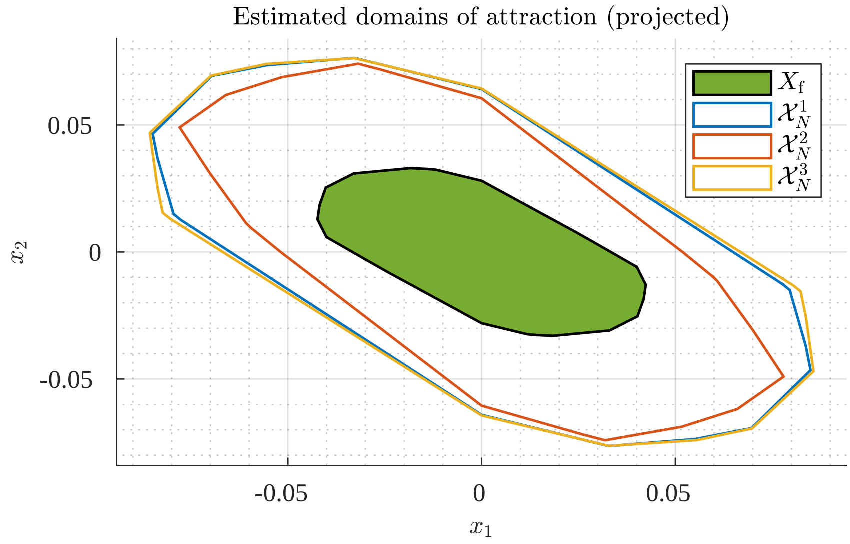


Figure 5.5: Realized (estimated) DOAs for the three designs in Example 2: projection on (x_1, x_2)-space.

	DOA vol.	DOF
Design 1 (homothetic, vertex controls)	$3.13 \cdot 10^{-3}$	$1 + (N-1)q_\theta q_s = 1345$
Design 2 (homothetic, simple controls)	$2.43 \cdot 10^{-3}$	$N = 8$
Design 3 (heterogeneous)	$3.23 \cdot 10^{-3}$	$\left(1 + q_\theta + q_\theta^2\right) + 4 = 25$

Table 5.5: Comparison of the three designs in Example 2.

Design 2 Homothetic tube with "simple" control parametrization, namely parameterization 1 from Table 5.1.

Design 3 A heterogeneous design, consisting of a scenario three for the first 4 prediction time instances, and a homothetic tube with the simple control parameterization 1 of Table 5.1 for the remaining prediction steps.

A projection on the (x_1, x_2)-space of the estimated DOAs realized by the three designs is shown in Figure 5.5 (an attempt to plot the full three-dimensional sets would be illegible). For this example, the DOAs were estimated using the method from Appendix B.3 with the point set $\hat{\mathcal{X}}_N^{\text{init}}$ consisting of the vertices of a one-norm ball, an infinity-norm ball, and the terminal set X_f. The corresponding DOA volumes and the number of control DOF are summarized in Table 5.5. As in the previous example, an illustration of the computation times required to solve the tube synthesis problems for the three different designs is given in Table 5.6. These times were obtained by simulating the closed-loop system with an initial state $x_0 = \begin{bmatrix} 0.05 & 0 & 0 \end{bmatrix}^\top$.

	Avg. time [ms]	Max. time [ms]
Design 1 (homothetic, vertex controls)	613	778
Design 2 (homothetic, simple controls)	310	342
Design 3 (heterogeneous)	317	360

Table 5.6: Illustration of complexity: solver time per sample in the simulation of Example 2.

As was remarked before, it should be possible to improve these figures by considering a more optimized linear programming implementation.

5.6 Concluding remarks

An LPV MPC algorithm based on the construction of so-called heterogeneously parameterized tubes was presented. In an HpT, different cross-section parameterizations can be combined into a single tube. This adds an additional degree of freedom into the design, which can be exploited to achieve different trade-offs between computational complexity and achievable control performance: hence, this chapter has provided one possible affirmative answer to Research Question Q_3. A few directions for further research are suggested:

- One important question for further research is how heterogeneous parameterization structures can be optimally designed. In the LTI case, the design of optimal move blocking strategies that guarantee, e.g., a minimum size of the achieved DOA are considered in (Gondhalekar and Imura 2010; Shekhar and Manzie 2015). It would be of interest to investigate if similar design approaches can be exported to the heterogeneously parameterized tube-setting.
- It would furthermore be interesting to consider also the implementation of various cross section parameterizations next to the "scenario" and "homothetic" cases, such as the "elastic" parameterization from (Raković et al. 2016b). Including the elastic parameterization in the framework of Section 5.2, is straightforward. It must however be investigated if the approximations that are necessary to render the elastic tube synthesis convex, do not introduce possible violations of Lemma 5.1 in an implementation.
- Further improving the computational implementation of HpTMPC is also of interest. For instance, one could eliminate redundant constraints or avoid the use of the vertex representation for those cross sections that use a "simple" control parameterization like $u = c + Ke$.

Chapter 6

Reduced-complexity tube parameterizations

In this chapter, an alternative tube parameterization design is proposed as a second answer to Research Question Q_3. The starting point of the design is a given set of initial conditions that represents an inner-approximation of the desired domain of attraction (DOA) of the controller. Then, an off-line procedure based on sparsity-inducing ℓ_1-optimization is used to determine—one for each vertex of the initial condition set— a collection of tube parameterizations with a reduced number of degrees of freedom (DOFs). The method is developed based on the assumption that only the vertex representation of the sets that define the tube cross section shapes is given: because this representation has to be available anyway to enable the synthesis of vertex control policies, this avoids the need for computing equivalent hyperplane representations. On-line, on the basis of the measured state and scheduling values, a complete tube parameterization is constructed as the combination of a suitable selection from the parameterizations that were computed off-line. It is shown that this construction leads to a DOA that is at least as large as the set of initial states that was considered initially. As it is difficult to guarantee a-priori recursive feasibility and closed-loop stability in a receding-horizon control (RHC) setting, a decreasing-horizon control (DHC) algorithm that does possess these desirable properties is presented. The achievable reduction in the number of decision variables is demonstrated on two numerical examples.

6.1 Introduction

In the tube-based model predictive control (MPC) designs proposed in the previous chapters, as well as in other existing methods in the literature, a certain tube parameterization is selected and its effect on the domain of attraction of the resulting controller is evaluated *a-posteriori*. With respect to MPC designs methods of the literature, the heterogeneous tubes from Chapter 5 introduced an extra degree of freedom in the design by allowing the tube parameterization to be different along the prediction horizon. This heterogeneous parameterization is required to be chosen—subject to conditions that guarantee recursive feasibility—by the control designer before its effect on the DOA can be analyzed.

This chapter proposes an alternative, reversed, procedure. Instead of designing a parameterization first and investigating its effect on the DOA afterwards, it is now proposed to first design a set of initial conditions that corresponds the minimal desired DOA. Subsequently, an efficient parameterization which leads to a controller that is guaranteed to be feasible on the given set is computed by means of an off-line design procedure. This makes it possible to directly optimize the trade-off between the computational complexity of the controller and the achieved DOA as specified in terms of the initial condition set.

A well-known method for reducing the computational complexity of MPC is move blocking (Cagienard et al. 2007). In move blocking, the number of control degrees of freedom is reduced by requiring that control inputs are held constant for some parts–as described in terms of a so-called blocking structure—of the prediction horizon. The work (Shekhar and Manzie 2015) considers "optimal" design of move blocking structures. An off-line design method based on mixed-integer programming is proposed, which aims to find the most efficient blocking structure subject to the constraint that the DOA of the reduced-complexity controller contains a desired set of initial conditions.

In (Oldewurtel et al. 2009), a blocking strategy to decrease the number of degrees of freedom in the robust MPC of (Goulart et al. 2006) is proposed. As in (Goulart et al. 2006), the on-line optimization is carried out over feedback policies that are affine functions of the past disturbances. However, these feedback policies are now parameterized by a vector of decision variables of reduced dimension and a blocking matrix. This blocking matrix is chosen by the designer and determines the structure (e.g., diagonal) of the disturbance feedback matrices. In (Oldewurtel et al. 2009), the word "blocking" does not mean that the control input is held constant along several samples, but instead that the feedback policies are parameterized by a vector of reduced dimension.

An alternative approach that aims to decrease the computational complexity of linear time-invariant (LTI) MPC is given by (Jost et al. 2015b). Therein, a complexity reduction is achieved not by eliminating degrees of freedom (i.e., decision variables), but by eliminating redundant constraints. Based on a measurement of the current state, constraints that are known to be redundant are removed from the on-line optimization problem. This method was extended to tube-based MPC for additively disturbed LTI (LTI+) systems in (Jost et al. 2015a).

In this chapter, a method for reducing the computational complexity of tube-based linear parameter-varying (LPV) MPC by eliminating unnecessary DOF from the tube parameterization is proposed. The approach consists of two parts: an off-line design part, and an on-line part. In the off-line design phase, a reduced-complexity parameterization that guarantees feasibility on a specified set of initial conditions is computed. The set of initial conditions, which can be chosen by the designer, is only subject to the constraint that it is contained inside the DOA corresponding to a full-complexity tube-based MPC. By using reweighted ℓ_1-minimization, it is determined how many degrees of freedom can be eliminated from the control problem that is to be solved on-line, while still guaranteeing feasibility on the initial condition set. This off-line part of the procedure requires the solution of a number of instances of modified version of the tube MPC optimization problem itself, i.e., of a number of linear programs (LPs). The end result is a collection of "sparse" parameterizations, each one corresponding to a single vertex of the initial condition set.

On-line, based on the measured state and scheduling variable value, a tube parameterization is obtained as a combination of a suitable selection from the parameterizations that were designed off-line. This combined parameterization is subsequently used in the solution of the on-line tube synthesis problem. In contrast to the parameterizations from the previous chapters, this sparse parameterization does not by construction guarantee recursive feasibility when used in an RHC scheme. It is however possible to obtain a guaranteed recursively feasible controller by considering a DHC approach.

The proposed approach from this chapter shares conceptual similarities with (Jost et al. 2015b; Oldewurtel et al. 2009; Shekhar and Manzie 2015). As in (Shekhar and Manzie 2015), the purpose of the approach is to find an efficient parameterization that guarantees feasibility on a designed initial condition set. However, the computational reduction is not achieved by forcing control inputs to be constant along parts of the prediction horizon, but by employing a reduced-complexity parameterization like (Oldewurtel et al. 2009). In (Oldewurtel et al. 2009), this parameterization has to be chosen manually, but in this chapter a computational procedure for designing the parameterization is proposed in the context of tube-based MPC for LPV systems. Finally, the similarity between (Jost et al. 2015b) and the proposed approach is that both approaches use a measurement of the current system state to construct the optimization problem that is to be solved. In the case of (Jost et al. 2015b) this involves selecting which constraints to eliminate, while in the approach of this chapter this involves constructing a suitable state-dependent tube parameterization.

The remainder of this chapter is structured as follows. In Section 6.1.1, the problem setting of this chapter is specified. Subsequently, Section 6.2 introduces some preliminary concepts related to sparsity and reweighted ℓ_1-optimization. The reduced-complexity MPC algorithm is described in Section 6.3. In Section 6.4, two numerical examples to demonstrate the reduction in decision variables that can be achieved are given. Finally, the chapter ends with some concluding remarks in Section 6.5.

6.1.1 Problem setting

This chapter considers a constrained polytopic LPV state-space (LPV-PSS) representation of the system that is to be controlled, where similarly to Chapter 4, only the state transition matrix A is parameter-varying, i.e.,

$$x(k+1) = A(\theta(k))x(k) + Bu(k). \tag{6.1}$$

Again, the problem setting is identical to that of Problem 3.1, i.e., it is desired to design an MPC such that the origin is a regionally asymptotically stable equilibrium for the closed-loop system. In this chapter, a solution to this problem is proposed in terms of a tube-based LPV MPC, which employs a tube parameterization that is optimized off-line such that it contains the minimum DOF necessary to achieve feasibility on a selected set of initial conditions.

In this chapter, it is assumed that only the vertex representation of the shape sets that are used to construct the tube are available. In the implementation of the previous chapters, the vertex representation was used to enable the synthesis of vertex control policies, but an equivalent half-space representation was employed to verify set inclusions. Because the V-representation has to be available anyway if vertex policies are to be used, the requirement that the H-representation is also known can be considered undesirable. This is especially true if a set is designed directly in V-representation with a fixed number of vertices using, for instance, the method from Appendix A. Then, it may be the case that the set is represented by many more hyperplanes than vertices, thus negating the benefit of designing a set with a low number of vertices in the first place.

If there is no H-representation available, the LP formulation of Section 4.4.2 can not be used, and therefore this chapter develops an alternative LP.

6.2 Preliminaries and reweighted ℓ_1-optimization

This section introduces a few preliminary concepts related to sparsity. These will be used later in the development of the reduced-complexity tube MPC formulation.

The so-called ℓ_0-"norm" (which is, in fact, not a norm because it is not homogeneous) is used to measure the number of non-zero elements in a vector:

Definition 6.1 (ℓ_0-"norm"). *Let $x \in \mathbb{R}^n$. The ℓ_0-"norm" $\|x\|_0$ equals the number of non-zero elements in x.*

Sometimes, $\|\cdot\|_0$ is also called the "support" of a vector. The "sparsity pattern" of a vector can be defined as follows.

Definition 6.2 (Sparsity pattern). *For a vector $x \in \mathbb{R}^n$, $\operatorname{spar}(x) \in \{0,1\}^n$ is a vector containing only zeros and ones, where the position of the ones corresponds to the positions of the nonzero elements in x.*

The following result, called Carathéodory's theorem, is well-known in combinatorial convexity:

Proposition 6.1 (Carathéodory's theorem). *(Schneider 2013a, Theorem 1.1.4) If $V \subset \mathbb{R}^n$ and $v \in \operatorname{convh}\{V\}$, then v is a convex combination of $n+1$ or fewer points in V.*

In particular, if $V = \{\bar{v}^1, \ldots, \bar{v}^q\}$ is the set of vertices defining a convex polytope $P = \operatorname{convh}\{V\}$, then $v \in P$ if and only if there exists a vector $\lambda \in \mathbb{R}^q_+$ containing at most $n+1$ non-zero elements such that $1^\top \lambda = 1$ and $\sum_{i=1}^q \lambda_i \bar{v}^i = v$. This means that if the inclusion $v \in P$ needs to be verified, it is in principle sufficient to solve a linear feasibility problem in $n+1$ decision variables corresponding to the non-zero elements in $\lambda \in \mathbb{R}^q_+$. However, it is not known a-priori which elements can be set to zero and which should be left as decision variables. In the approach developed in this chapter, suitable sparsity patterns for vectors of convex multipliers $\lambda \in \mathbb{R}^q_+$ are going to be identified as part of the off-line tube parameterization design. As suggested by Proposition 6.1, this can lead to a significant reduction in the number of decision variables.

As the final preliminary, a reweighted ℓ_1-optimization algorithm is introduced. Let $x \in \mathbb{R}^n$ and consider the set of linear (in)equalities

$$\begin{aligned} A_\mathrm{i} x &\leq b_\mathrm{i}, \\ A_\mathrm{e} x &= b_\mathrm{e}. \end{aligned} \tag{6.2}$$

The goal is to find a vector x which satisfies (6.2) and is sparse, i.e., contains relatively few non-zero elements. Because minimizing $\|x\|_0$ directly is intractable, an approximation has to be employed. It is well-known that minimizing the ℓ_1-norm of x, i.e., solving the linear program

$$x = \arg\min_x \|x\|_1 \text{ subject to (6.2)} \tag{6.3}$$

usually gives a sparse solution, though not necessarily the most sparse. To improve upon the sparsity of the solutions obtained by (6.3), the reweighted ℓ_1-minimization algorithm was introduced by (Candès et al. 2008) in the context of sparse signal reconstruction. It is shown in Algorithm 6.1.

Algorithm 6.1 Reweighted ℓ_1-optimization (Candès et al. 2008).

Require: A set of linear (in)equalities (6.2), iteration limit $k_{\max} \in [1..\infty)$, tolerance $\epsilon \in \mathbb{R}_+$, termination tolerance $\delta \in \mathbb{R}_+$

$k \leftarrow 1$
For all $i \in \mathbb{N}_{[1,n]}$, set $w_i^{(1)} = 1$
For all $i \in \mathbb{N}_{[1,n]}$, set $w_i^{(0)} = \infty$ ▷ Or large enough such that $\|w^{(1)} - w^{(0)}\| > \delta$ holds
while $k < k_{\max}$ **and** $\|w^{(k)} - w^{(k-1)}\| > \delta$ **do**
    Solve $x^{(k)} = \arg\min_x \sum_{i=1}^{n} w_i^{(k)} |x_i|$ subject to (6.2)
    For all $i \in \mathbb{N}_{[1,n]}$, set $w_i^{(k+1)} = \frac{1}{|x_i^{(k)}| + \epsilon}$
    $k \leftarrow k + 1$
end while

In Algorithm 6.1, ϵ should be set to a small value (e.g., $\epsilon = 10^{-6}$). The termination tolerance δ is used to terminate the algorithm whenever the values of the weights converge. By applying this iterative algorithm, sparser solutions to (6.2) than those returned by (6.3) are typically obtained. Although there is no a-priori convergence guarantee, it was observed in (Candès et al. 2008) that the algorithm usually needs only a few iterations to converge to a sparse solution of (6.2).

6.3 The reduced-complexity TMPC algorithm

In this section, the reduced-complexity LPV MPC algorithm is presented. Section 6.3.1 introduces the concept of sparsily parameterized tubes (SpTs), which is central to the developed approach. An LP formulation of the tube synthesis problem is developed in Section 6.3.2. The design of a "sparse" tube parameterization that guarantees feasibility on an assigned set of initial conditions consists of two parts: an off-line part and an on-line part. The off-line design procedure is discussed in Section 6.3.3, and the on-line part of the algorithm is presented in Section 6.3.4. Finally, in Section 6.3.5, the complete algorithm is summarized and its main properties are proven.

6.3.1 Sparsely parameterized tubes

In this section, first, the concept of a sparsily parameterized tube (SpT) is introduced. The resulting SpT tube synthesis problem, which will be at the core of the MPC strategy proposed in this chapter, is specified.

Definition 6.3. *Let* $\mathbf{T}$ *be a tube according to Definition 3.4. For all* $i \in \mathbb{N}_{[0,N]}$*, introduce parameter sets* $\mathbb{P}(i) = \mathbb{P}^{\mathrm{x}}(i) \times \mathbb{P}^{\mathrm{k}}(i) \times \mathbb{P}^{\lambda}(i)$*. A tube* $\mathbf{T}$ *is called a* sparsily parameterized tube (SpT) *if it satisfies the following conditions:*

(i) For all $i \in \mathbb{N}_{[0,N]}$ *there is a function* $P^{\mathrm{x}}(\cdot|i) : \mathbb{P}^{\mathrm{x}}(i) \to \mathbb{R}^{n_{\mathrm{x}}}$ *and there exists a parameter* $p_i^{\mathrm{x}} \in \mathbb{P}^{\mathrm{x}}(i)$ *such that* $X_i = P^{\mathrm{x}}(p_i^{\mathrm{x}}|i)$*.*

(ii) For all $i \in \mathbb{N}_{[0,N-1]}$ *there is a second-order function* $P^{\mathrm{k}}(\cdot|i) : \mathbb{P}^{\mathrm{k}}(i) \to (X_i \times \Theta_i \to \mathbb{U})$ *and there exists a parameter* $p_i^{\mathrm{k}} \in \mathbb{P}^{\mathrm{k}}(i)$ *such that* $K_i = P^{\mathrm{k}}(p_i^{\mathrm{k}}|i)$*.*

(iii) For all $i \in \mathbb{N}_{[0,N-1]}$ *and for all* $x \in \mathcal{I}(X_i, \Theta_i|K_i)$ *there exists a vector* $p_i^{\lambda} \in \mathbb{P}^{\lambda}(i)$ *satisfying* $p_i^{\lambda} \in \mathrm{Convm}(x|X_{i+1})$*.*

Further, define the shorthand $P(p|i) = \left(P^{\mathrm{x}}(p^{\mathrm{x}}|i), P^{\mathrm{k}}(p^{\mathrm{k}}|i)\right)$ *where* $p = (p^{\mathrm{x}}, p^{\mathrm{k}}, p^{\lambda})$*.*

The SpTs of Definition 6.3 look similar to the heterogeneously parameterized tubes (HpTs) of Definition 5.1. The main difference is that in Definition 6.3 the convex multipliers that are necessary to verify the tube inclusions $\mathcal{I}(X_i, \Theta_i|K_i) \subseteq X_{i+1}$ are also considered to be part of the tube parameterization, as described in terms of the sets $\mathbb{P}^{\lambda}(\cdot)$ (see Definition 2.1 for the definition of $\mathrm{Convm}(\cdot|\cdot)$). Although in Definition 6.3 a requirement on "sparsity" is not yet imposed, the name SpT is motivated by the specific approach that is developed in this chapter to design the parameterization structure

$$\mathcal{P}_N = \left\{(\mathbb{P}(0), P(\cdot|0)), \ldots, (\mathbb{P}(N), P(\cdot|N))\right\}. \tag{6.4}$$

In particular, sparsity-promoting ℓ_1-optimization will be employed to determine which DOF can be removed from (6.4) when compared to a given full-complexity parameterization structure (this full-complexity or "dense" structure is introduced later in Section 6.3.2).

Similarly to Chapter 5, the set of feasible tubes is denoted by

$$\mathcal{T}_N(x, \mathbf{\Theta}|\mathcal{P}_N) = \{\mathbf{T} \,|\, \mathbf{T} \text{ satisfies Def. 5.1 with } X_0 = \{x\} \text{ and } X_N \subseteq X_{\mathrm{f}}\} \tag{6.5}$$

the tube synthesis problem by

$$V(x, \mathbf{\Theta}|\mathcal{P}_N) = \min_{\mathbf{T}} J_N(\mathbf{T}, \mathbf{\Theta}) \text{ subject to } \mathbf{T} \in \mathcal{T}_N(x, \mathbf{\Theta}|\mathcal{P}_N), \tag{6.6}$$

and the corresponding DOA by

$$\mathcal{X}_N(\mathbf{\Theta}|\mathcal{P}_N) = \{x \in \mathbb{X} \,|\, \mathcal{T}_N(x, \mathbf{\Theta}|\mathcal{P}_N) \neq \emptyset\}. \tag{6.7}$$

In this chapter, the focus is on the design of a computationally efficient parameterization structure (6.4), such that the resulting controller achieves a DOA (6.7) that is guaranteed to contain a desired set of initial states X_0. The issues of recursive feasibility and closed-loop asymptotic stability whenever the controller is executed in an RHC fashion are not considered in this chapter. Therefore, the selection of the cost function $J_N(\cdot,\cdot,\cdot)$ in (5.2) is not constrained by stability considerations. It is however suggested to use the cost function from Section 3.4.5 with the terminal cost from Chapter 4.

It is furthermore assumed that the terminal set X_f is controlled λ-contractive for the system (6.1) in the sense of Definition 2.13. In the absence of recursive feasibility and stability guarantees in the RHC setting, this property can be exploited to easily obtain a stabilizing DHC implementation instead, as shown later in Section 6.3.5.

6.3.2 Dense parameterization and LP formulation

In this section, an LP that solves (6.6) is derived. To make this possible, it is first necessary to define a concrete parameterization structure $\mathscr{P}_N$. The tube cross sections are parameterized in the familiar homothetic manner, i.e., there holds

$$\begin{aligned} p_i^\mathrm{x} &= (\alpha_i, z_i) \in \overline{\mathbb{P}}^\mathrm{x}(i), \\ \overline{\mathbb{P}}^\mathrm{x}(i) &= \mathbb{R}_+ \times \mathbb{R}^{n_\mathrm{x}}, \\ \overline{P}^\mathrm{x}\left(p_i^\mathrm{x} \middle| i\right) &= z_i \oplus \alpha_i S, \end{aligned} \tag{6.8}$$

where S is a polytopic PC-set specified in V-representation as

$$S = \operatorname{convh}\left\{\bar{s}_i^1, \ldots, \bar{s}_i^{q_\mathrm{s}}\right\}. \tag{6.9}$$

An augmented control parameterization is defined as

$$\begin{aligned} p_i^\mathrm{k} &= \left(\bar{u}_i, u_i^{(1,1)}, \ldots, u_i^{(q_\mathrm{s},1)}, \ldots, u_i^{(q_\mathrm{s},q_\theta)}\right) \in \overline{\mathbb{P}}^\mathrm{k}(i), \\ \overline{\mathbb{P}}^\mathrm{k}(i) &= \mathbb{R}^{(q_\theta q_\mathrm{s}+1)n_\mathrm{u}}, \\ \overline{P}^\mathrm{k}\left(p_i^\mathrm{k} \middle| i\right) &= \bar{u}_i + K_\mathrm{f}(\cdot - z_i, \cdot) + \operatorname{vertpol}\left(p_i^\mathrm{k} \middle| X_i, \Theta_i\right)(\cdot,\cdot), \end{aligned} \tag{6.10}$$

where $K_\mathrm{f} : S \times \Theta \to \mathbb{U}$ is a pre-designed $\mathcal{CH}_1$ control law and $\bar{u}_i$ are "nominal" control actions to be optimized. This augmented policy does not affect the DOA with respect to a standard gain-scheduled vertex control policy. Therefore, at first sight, the addition of the additional control input $\bar{u}_i$ and the local controller K_f can seem unnecessary. The purpose of this augmented policy is to aid the off-line parameterization design in its capability to remove as many of the degrees of freedom $u_i^{(j,l)}$ that define the vertex policy $\operatorname{vertpol}(\cdot|\cdot,\cdot)$ as possible, while still retaining feasibility on the desired set of initial conditions.

In Definition 6.3, the convex multipliers necessary to verify the tube set inclusions if only the vertex representation (6.9) is available, are also considered to be part of the tube parameterization. For now, the following full parameterization for the convex multipliers is used:

$$\begin{aligned} p_i^\lambda &= \left(\lambda_i^{(1,1)}, \ldots, \lambda_i^{(q_\mathrm{s},1)}, \ldots, \lambda_i^{(q_\mathrm{s},q_\theta)}\right) \in \overline{\mathbb{P}}^\lambda(i), \\ \overline{\mathbb{P}}^\lambda(i) &= \mathbb{R}_+^{q_\theta q_\mathrm{s}^2}. \end{aligned} \tag{6.11}$$

The tube parameterization structure as defined by (6.8), (6.10), and (6.11) will be referred to as the "dense" structure and will be denoted by $\overline{\mathcal{P}}_N$. At first sight, the dependence of $\left(\overline{\mathbb{P}}^{\mathrm{x}}(\cdot), \overline{\mathbb{P}}^{\mathrm{k}}(\cdot), \overline{\mathbb{P}}^{\lambda}(\cdot)\right)$ on i seems unnecessary. However, in the off-line design procedure, a time-dependent sparse parameterization will be computed based on the time-invariant dense parameterization defined above: therefore, for consistency with later developments, the time-dependence is kept in the notation.

A construction of the optimization problem (6.6) and the corresponding number of variables and constraints is now provided along the same line as in Section 4.4.2, but using the parameterization structure $\overline{\mathcal{P}}_N$ and using only the vertex representation (6.9). The cost function is omitted from the formulation presented here for the sake of simplicity, but the cost function of Section 3.4.5 can be included according to the construction in Section 4.4.2.

Cross section parameterization According to (6.8), each tube cross section is characterized by a scaling $\alpha_i \in \mathbb{R}_+$ and a center $z_i \in \mathbb{R}^{n_{\mathrm{x}}}$. As there are $N+1$ cross sections in the tube (counting X_0 up to and including X_N), this introduces $(N+1)(1+n_{\mathrm{x}})$ decision variables. Furthermore, the scalings must be non-negative, leading to the $N+1$ inequality constraints

$$\forall i \in [0..N]: \ \alpha_i \geq 0. \tag{6.12}$$

Control parameterization At each prediction time instant $i \in [0..N-1]$, the parameterization (6.10) consists of $q_{\mathrm{s}}q_\theta + 1$ control actions, i.e., $n_{\mathrm{u}}(q_{\mathrm{s}}q_\theta + 1)$ decision variables. Thus the control parameterization introduces a total of $N(n_{\mathrm{u}}q_{\mathrm{s}}q_\theta + 1)$ variables. However, because of the constraint $X_0 = \{x\}$ and the availability of the measurement $\theta(k)$ all control actions for $i = 0$ are equal. Thus the number of introduced decision variables can be slightly reduced to $n_{\mathrm{u}} + (N-1)(n_{\mathrm{u}}q_{\mathrm{s}}q_\theta + 1)$.

Initial condition constraint The constraint $X_0 = \{x\}$ in (6.5) is implemented by $n_{\mathrm{x}} + 1$ equality constraints

$$\begin{aligned} z_0 &= x, \\ \alpha_0 &= 0. \end{aligned} \tag{6.13}$$

Tube dynamics constraints The transition constraints $\mathcal{G}(X_i, \Theta_i | K_i) \subseteq X_{i+1}$ from Definition 3.4 are equivalent to the existence of convex multipliers $\lambda_i^{(j,l)} \in \mathbb{R}_+^{q_{\mathrm{s}}}$, that for all $(i,j,l) \in [0..N-1] \times [1..q_{\mathrm{s}}] \times [1..q_\theta]$ satisfy the conditions

$$\begin{aligned} \sum_{t=1}^{q_{\mathrm{s}}} \left[\lambda_i^{(j,l)}\right]_t \bar{s}^t &= A(\bar{\theta}^l)\left(z_i + \alpha_i \bar{s}^j\right) + B\left(\bar{u}_i + \alpha_i K_{\mathrm{f}}(\bar{s}^j, \bar{\theta}^l) + u_i^{(j,l)}\right) - z_{i+1}, \\ 1^\top \lambda_i^{(j,l)} &\leq \alpha_{i+1}, \end{aligned} \tag{6.14}$$

which amount to $Nq_{\mathrm{s}}q_\theta n_{\mathrm{x}}$ equality constraints and $Nq_{\mathrm{s}}q_\theta$ inequalities. Furthermore, the convex multipliers $\lambda_i^{(j,l)} \in \mathbb{R}_+^{q_{\mathrm{s}}}$ correspond to $Nq_{\mathrm{s}}^2 q_\theta$ decision variables.

State constraints The state constraints $\mathcal{I}(X_i, \Theta_i | K_i) \subseteq \mathbb{X}$ in Definition 3.4 can be expressed by $N r_{\mathrm{x}} q_{\mathrm{s}} q_\theta$ linear inequalities

$$\forall i \in [0..N-1] : \ \forall (j,l) \in [1..q_{\mathrm{s}}] \times [1..q_\theta] : \\ H_{\mathrm{x}} \left(A\left(\bar{\theta}_i^l\right) \left(z_i + \alpha_i \bar{s}^j\right) + B\left(\bar{u}_i + \alpha_i K_{\mathrm{f}}(\bar{s}^j, \bar{\theta}^l) + u_i^{(j,l)}\right)\right) \le 1. \quad (6.15)$$

Input constraints The input constraints are realized by $N r_{\mathrm{u}} q_{\mathrm{s}} q_\theta$ linear inequalities

$$\forall i \in [0..N-1] : \ \forall (j,l) \in [1..q_{\mathrm{s}}] \times [1..q_\theta] : \ H_{\mathrm{u}} \left(\bar{u}_i + \alpha_i K_{\mathrm{f}}(\bar{s}^j, \bar{\theta}^l) + u_i^{(j,l)}\right) \le 1. \quad (6.16)$$

Terminal constraint The terminal constraint $X_N \subseteq X_{\mathrm{f}}$ is equivalent to the condition that for all $j \in \mathbb{N}_{[1,q_s]}$, there exist multipliers $\eta^j \in \mathbb{R}_+^{q_{\mathrm{s}}}$ and a $\gamma \in \mathbb{R}_+$ such that

$$\begin{aligned} \sum_{t=1}^{q_s} \left[\eta^j\right]_t \bar{s}^t &= z_N + \alpha_N \bar{s}^j, \\ 1^\top \eta^j &\le \gamma, \\ \eta^j &\ge 0, \\ 0 \le \gamma &\le 1, \end{aligned} \quad (6.17)$$

which correspond to $n_{\mathrm{x}} q_{\mathrm{s}}$ equalities and $q_{\mathrm{s}}(1 + q_{\mathrm{s}}) + 2$ inequalities. Furthermore, the convex multipliers η^j and the scaling γ correspond to $q_{\mathrm{s}}^2 + 1$ decision variables.

The total LP Given the above constructions, the total LP to be solved-online becomes

$$V\left(x, \mathbf{\Theta} \middle| \overline{\mathcal{P}}_N\right) = \min_{\mathbf{d}} J_N(\mathbf{d}) \text{ subject to (6.12)–(6.17)} \quad (6.18)$$

with the decision variable $\mathbf{d}$ being the collection of all the variables introduced above, and possible additional slack variables to realize the cost function $J_N(\mathbf{d}) = J_N(\mathbf{T})$ along the same lines as in Section 4.4.2. Summing up the numbers of decision variables n_{d}, equality constraints n_{eq}, and inequality constraints n_{ineq} given above for each sub-part of the problem (excluding, as stated before, the cost function for simplicity) yields

$$\begin{aligned} n_{\mathrm{d}} &= (N+1)(1+n_{\mathrm{x}}) + n_{\mathrm{u}} + (N-1)(n_{\mathrm{u}} q_{\mathrm{s}} q_\theta + 1) + N q_{\mathrm{s}}^2 q_\theta + q_{\mathrm{s}}^2 + 1 \\ &= O(N), \\ n_{\mathrm{eq}} &= n_{\mathrm{x}} + 1 + N q_{\mathrm{s}} q_\theta n_{\mathrm{x}} + n_{\mathrm{x}} q_{\mathrm{s}} \\ &= O(N), \\ n_{\mathrm{ineq}} &= N + 1 + N q_{\mathrm{s}} q_\theta + N r_{\mathrm{x}} q_{\mathrm{s}} q_\theta + N r_{\mathrm{u}} q_{\mathrm{s}} q_\theta + q_{\mathrm{s}}(1+q_{\mathrm{s}}) + 2 \\ &= O(N). \end{aligned}$$

It can be observed that the dominant component of n_d is

$$(N-1)(n_\mathrm{u} q_\mathrm{s} q_\theta + 1) + N q_\mathrm{s}^2 q_\theta \tag{6.19}$$

which corresponds to the number of control actions in the augmented parameterization (6.10) and to the convex multipliers in (6.11) necessary to verify the tube dynamics constraints. In order to improve the computational complexity of the tube synthesis problem (6.6), decreasing this term would be most beneficial. Indeed, the approach developed in the next section is aimed at decreasing this number.

6.3.3 Off-line parameterization design

The purpose of the proposed design is to determine a parameterization structure $\mathscr{P}_N$ that corresponds to a reduced number of DOF with respect to the "dense" parameterization introduced previously, under the condition that for all $\boldsymbol{\Theta} \subseteq \Theta^N$, there holds $X_0 \subseteq \mathcal{X}_N(\boldsymbol{\Theta}|\mathscr{P}_N)$ where $X_0 \subseteq \mathbb{X}$ is an initial condition set that can be selected by the user. In this section, the off-line part of this design procedure is discussed. First, the requirements on X_0 are given, and a way of selecting X_0 is proposed. Then, as the main step in the off-line part of the design, reweighted ℓ_1-optimization is employed to determine sparsity patterns indicating which DOF can be eliminated from the tube parameterization without affecting feasibility on X_0. Later, in the on-line part of the algorithm that is described in the next section, these patterns will be combined into the final parameterization structure based on measurements of the state and scheduling variable values.

The requirements on the design of the initial condition set X_0 are that

(i) It must be a polytopic PC-set;

(ii) It must not be larger than the DOA achieved using the dense structure $\overline{\mathscr{P}}_N$, i.e., there must hold $X_0 \subseteq \mathcal{X}_N\left(\boldsymbol{\Theta}\middle|\overline{\mathscr{P}}_N\right)$.

A simple approach for the design of the set X_0 such that these conditions are always satisfied, based on the scaling of a fixed "initial condition shape"-set, is proposed next. Let a polytopic PC-set $\hat{X}_0 \subset \mathbb{R}^{n_\mathrm{x}}$ be given as

$$\hat{X}_0 = \operatorname{convh}\left\{\hat{x}_0^1, \dots, \hat{x}_0^{q_0}\right\}. \tag{6.20}$$

Reasonable choices that can be made for $\hat{X}_0$ are e.g., the terminal set, the unit cube- or rhombus, or a combination of these. For all $i \in \mathbb{N}_{[1,q_0]}$, solve the optimization problems

$$\beta_i = \max_{\beta} \beta \text{ subject to } \mathcal{T}_N\left(\beta \hat{x}_0^i, \Theta^N \middle| \mathscr{P}_N\right) \neq \emptyset \tag{6.21}$$

and compute the initial condition set X_0 as

$$X_0 = \mathbb{X} \cap \operatorname{convh}\left\{\beta_1 \hat{x}_0^1, \dots, \beta_{q_0} \hat{x}_0^{q_0}\right\} := \operatorname{convh}\left\{\bar{x}_0^1, \dots, \bar{x}_0^{q_0}\right\}. \tag{6.22}$$

Note that the maximization in (6.21) can be formulated as an LP simply by replacing the constraint (6.13) with

$$z_0 = \beta \hat{x}_0^j. \tag{6.23}$$

Once an initial condition set X_0 has been determined, a computationally efficient parameterization structure that guarantees initial feasibility on this set can be determined. For the subsequent developments, it is convenient to partition the tube parameters and the corresponding parameter sets as $p_i = (p_i^1, p_i^2)$ and $\mathbb{P}(i) = \left(\mathbb{P}^1(i), \mathbb{P}^2(i)\right)$ where

$$\begin{aligned} p_i^1 &= \left(p_i^{\mathrm{k}}, p_i^{\lambda}\right), & \mathbb{P}^1(i) &= \left(\mathbb{P}^{\mathrm{k}}(i), \mathbb{P}^{\lambda}(i)\right), \\ p_i^2 &= p_i^{\mathrm{x}}, & \mathbb{P}^2(i) &= \mathbb{P}^{\mathrm{x}}(i). \end{aligned} \tag{6.24}$$

Furthermore, for a tube of length N, let all the corresponding tube parameters be denoted as

$$\mathbf{p}^1 = \left(p_0^1, \ldots, p_{N-1}^1\right), \; \mathbf{p}^2 = \left(p_0^2, \ldots, p_N^2\right), \; \mathbf{p} = \left(\mathbf{p}^1, \mathbf{p}^2\right), \tag{6.25}$$

and let $\mathbf{T}(\mathbf{p})$ denote the tube that is parameterized by the parameters collected in $\mathbf{p}$.

In the partitioning introduced in (6.24)–(6.25), the parameters in $\mathbf{p}^1$ correspond to DOF that will possibly be eliminated from the sparse tube parameterization structure that is going to be designed. The parameters in $\mathbf{p}^2$ are not included in the design, and therefore these parameters will be the same in both the "dense" structure and in the sparse structure.

The parameterization design proceeds as follows. Define $\mathbf{\Theta}(\theta) = \{\{\theta\}, \Theta, \ldots, \Theta\} \subseteq \Theta^N$. Given the set X_0 which was designed previously, for all $(i, j) \in \mathbb{N}_{[1,q_0]} \times \mathbb{N}_{[1,q]}$ the optimization problems

$$\mathbf{p}_{\mathrm{sp}}\left(\bar{x}_0^i, \bar{\theta}^j\right) = \arg\min_{\mathbf{p}} \|\mathbf{p}^1\|_0 \text{ subject to } \mathbf{T}(\mathbf{p}) \in \mathcal{T}_N\left(\bar{x}_0^i, \mathbf{\Theta}\left(\bar{\theta}^j\right) \middle| \overline{\mathcal{P}}_N\right) \tag{6.26}$$

are approximately solved using the reweighted ℓ_1-minimization described in Algorithm 6.1. Each iteration in the reweighted ℓ_1-minimization algorithm requires the solution of one LP. With respect to (6.18), minimizing the ℓ_1-norm of $\mathbf{p}^1$ only requires the introduction of slack variables corresponding to the absolute values of the respective parameters. If $k_{\max}$ is the maximum number of iterations of the reweighted algorithm, then this part of the design procedure requires solving at most $k_{\max} q_0 q$ LPs, where the size of each LP is approximately the same as that of the tube synthesis problem with the parameterization structure $\overline{\mathcal{P}}_N$.

After solving all problems, the resulting sparsity patterns

$$\operatorname{spar}\left(\mathbf{p}_{\mathrm{sp}}^1\left(\bar{x}_0^i, \bar{\theta}^j\right)\right), \; (i, j) \in \mathbb{N}_{[1,q_0]} \times \mathbb{N}_{[1,q]} \tag{6.27}$$

are stored. In the on-line part of the algorithm, which is described in the next subsection, these stored patterns are used to construct the final tube parameterization that is used in solving (6.6) based on the measured state and scheduling values.

It is noted that reweighted ℓ_1-optimization is not the only possible approach that can be used to obtain sparse approximate solutions to (6.26). Other methods that can be considered are, e.g., Lasso and Non-negative garrotte (NNG) (Tibshirani 1996; Yuan and Lin 2007).

6.3.4 On-line parameterization construction

In this section, the on-line part of the algorithm is presented. On-line, at each sampling instance, a measurement (x, θ) is obtained. Then, the patterns (6.27) that were designed off-line are combined to form one final parameterization structure such that the tube synthesis problem is feasible for the measured state and scheduling values.

To construct this parameterization, the first step is to find multipliers $\xi \in \mathbb{R}_+^{q_0}$ to certify that $x \in \beta X_0$ for some $\beta \geq 0$, and convex multipliers $\zeta \in \mathbb{R}_+^q$ proving that $\theta \in \Theta$. This is done by solving the two small LPs

$$\xi(x) = \arg \min_{\xi \in \mathbb{R}_+^{q_0}} \left\{ \|\xi\|_1 \,\middle|\, \sum_{i=1}^{q_0} \xi_i \bar{x}_0^i = x \right\}, \tag{6.28a}$$

$$\zeta(\theta) = \arg \min_{\zeta \in \mathbb{R}_+^{q}} \left\{ \|\zeta\|_1 \,\middle|\, \sum_{i=1}^{q} \zeta_i \bar{\theta}^i = \theta, \|\zeta\|_1 = 1 \right\}. \tag{6.28b}$$

Note that because X_0 is PC, if $x \in X_0$, then the solution of (6.28a) satisfies $\|\xi\|_1 \leq 1$. Furthermore, the solution is sparse due to 1-norm objective, and typically does not contain more than $n_\mathrm{x} + 1$ non-zeros in agreement with Proposition 6.1. The condition $\|\xi\|_1 \leq 1$ is not included as a constraint, because it allows to also consider the case that $x \neq X_0$ (equivalently, $x \in \beta X_0$ for some $\beta > 1$). This allows the construction of a parameterization also for initial states $x \notin X_0$, thus possibly rendering the DOA of the controller larger than X_0.

The formulation of problem (6.28b) seems counterintuitive, because it is asked to "minimize" $\|\zeta\|_1$ under the constraint that $\|\zeta\| = 1$. However, by formulating the problem in this way as an LP, in practice typically a sparse solution having at most $n_\theta + 1$ non-zeros—according to Proposition 6.1—is obtained. In (6.28b), the equality $\|\zeta\|_1 = 1$ must be used instead of $\|\zeta\| \leq 1$ because Θ is not necessarily a PC-set.

Note that the sparsity of the solutions (6.28a)–(6.28a) could possibly be further improved by employing reweighted ℓ_1-minimization here as well. This is not further pursued here.

The next step is to construct the sets of indices

$$\begin{aligned} \mathcal{I}_\xi(x) &= \{i \in \mathbb{N}_{[1,q_0]} \mid \xi_i(x) \neq 0\}, \\ \mathcal{I}_\zeta(\theta) &= \{i \in \mathbb{N}_{[1,q_\theta]} \mid \zeta_i(\theta) \neq 0\}, \end{aligned} \tag{6.29}$$

corresponding to the locations of the non-zero multipliers in $\xi(x)$ and $\zeta(\theta)$, and to compute the combined sparsity patterns

$$\forall i \in [0..N-1] : \ \mathrm{SPAR}_i(x, \theta) = \bigvee_{r \in \mathcal{I}_\xi(x)} \bigvee_{s \in \mathcal{I}_\zeta(\theta)} \mathrm{spar}\left(p_i^1\left(\bar{x}_0^r, \bar{\theta}^s\right)\right) \tag{6.30}$$

where the patterns $\mathrm{spar}\left(p_i^1\left(\bar{x}_0^r, \bar{\theta}^s\right)\right)$ are extracted from the off-line designed patterns (6.27) according to the partitioning of (6.25). The element-wise disjunction operator $\vee$ is defined as follows:

Definition 6.4 (Element-wise disjunction)**.** *Let $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ be vectors. Then, $x \vee y = z \in \{0,1\}^n$ is defined as*

$$\forall i \in \mathbb{N}_{[1,n]} : \; z_i = \begin{cases} 1, & x_i = 1 \text{ or } y_i = 1, \\ 0, & \text{otherwise.} \end{cases}$$

Based on (6.30), the final sparse tube parameterization that will be used in solving the tube synthesis problem (6.6) can be defined. First, construct the sets of tube parameters corresponding to the patterns $\mathrm{SPAR}_i(x,\theta)$ as

$$\mathbb{P}^1(i|x,\theta) = \left\{ p^1 \in \overline{\mathbb{P}}(i) \,\middle|\, \mathrm{spar}\left(p^1\right) \leq \mathrm{SPAR}_i(x,\theta) \right\}. \tag{6.31}$$

This leads to the initial state- and scheduling-dependent parameterization structure

$$\mathcal{P}_N(x,\theta) = \left\{ \left(\mathbb{P}(0|x,\theta), P(\cdot|0)\right), \ldots, \left(\mathbb{P}(N-1|x,\theta), P(\cdot|N-1)\right), \left(\overline{P}(N), P(\cdot|N)\right) \right\} \tag{6.32}$$

where $\mathbb{P}(i|x,\theta) = \mathbb{P}^1(i|x,\theta) \times \mathbb{P}^2(i)$.

Finally, the on-line tube synthesis problem is constructed as in (6.6), given the parameterization structure $\mathcal{P}_N(x,\theta)$ of (6.32). Decision variables that correspond to parameters that are identically zero in the structure $\mathcal{P}_N(x,\theta)$ can be directly eliminated from the optimization problem, resulting in reduced computational complexity of the proposed method.

6.3.5 Main result

The full algorithm, consisting of both the on- and off-line parts described in the preceding sections, is summarized in Algorithm 6.2. In Algorithm 6.2, Step 4 involves the solution of two small LPs, and Step 9 requires the solution of one larger LP. The tube synthesis problem as solved in Step 9 is guaranteed to be feasible for all the initial states on the designed set X_0 and for all measured scheduling variable values in the set Θ, as stated in the next theorem.

Theorem 6.1. *The tube synthesis problem* (6.6)*, given the parameterization structure* (6.32)*, is feasible for all initial conditions $(x,\theta) \in X_0 \times \Theta$. That is, for all $(x,\theta) \in X_0 \times \Theta$, $\mathcal{T}_N\left(x, \boldsymbol{\Theta}(\theta) | \mathcal{P}_N(x,\theta)\right) \neq \emptyset$, where $\boldsymbol{\Theta}(\theta) = \{\{\theta\}, \Theta, \ldots, \Theta\} \subseteq \Theta^N$.*

Proof. Recall the vertex representations $X_0 = \mathrm{convh}\left\{\bar{x}_0^1, \ldots, \bar{x}_0^{q_0}\right\}$ and $\Theta = \left\{\bar{\theta}^1, \ldots, \bar{\theta}^{q_\theta}\right\}$. Suppose that $(x,\theta) \in X_0 \times \Theta$. The construction of Steps 4–6 of Algorithm 8.1 yields a tube synthesis problem that is feasible for every initial point $(\bar{x},\bar{\theta}) \in \mathcal{F} = \left\{\left(\bar{x}^i, \bar{\theta}^j\right) \mid i \in \mathcal{I}_\xi(x), j \in \mathcal{I}_\zeta(\theta)\right\}$. In other words, $\forall (\bar{x},\bar{\theta}) \in \mathcal{F} : \; \mathcal{T}_N(x,\theta | \mathcal{P}_N(x,\theta)) \neq \emptyset$. Because (6.6) is an LP, this implies feasibility on $\mathrm{convh}\{\mathcal{F}\}$ as well. By construction, $(x,\theta) \in \mathrm{convh}\{\mathcal{F}\}$ and therefore (6.6) is feasible for any $(x,\theta) \in X_0 \times \Theta$. □

The sparse parameterization structure $\mathcal{P}_N(x,\theta)$ guarantees initial feasibility on $X_0 \times \Theta$, but it does not by construction lead to guaranteed recursive feasibility. Based on the assumption that the terminal set X_f is controlled λ-contractive, it is however possible to formulate a DHC variant of Algorithm 6.2 that does possess the desirable properties of guaranteed recursive feasibility and regional asymptotic stability. This DHC modification is summarized in Algorithm 6.3 and its properties are summarized in the following Theorem.

Algorithm 6.2 RHC sparsely parameterized TMPC.

Require: $N \in [1,\infty)$ and an initial condition set $X_0 \subseteq \mathbb{X}$

Off-line part

Compute the patterns $\text{spar}\left(\mathbf{p}^1_{\text{sp}}\left(\bar{x}^i_0, \bar{\theta}^j\right)\right)$, $(i,j) \in \mathbb{N}_{[1,q_0]} \times \mathbb{N}_{[1,q]}$ as described in Section 6.3.3

On-line part

$k \leftarrow 0$
loop
  Find $\xi\,(x(k))$ and $\zeta\,(\theta(k))$ by solving (6.28a)-(6.28b)
  For all $i \in [0..N-1]$, compute $\text{SPAR}_i(x(k),\theta(k))$ according to (6.30)
  Construct $\mathcal{P}_N(k) := \mathcal{P}_N(x(k),\theta(k))$ according to (6.32)
  Construct $\mathbf{\Theta}_k = \left\{\{\theta(k)\}, \left\{\Theta_{i|k}\right\}_{i=1}^{N-1}\right\} \subseteq \Theta^N$ such that $\mathbf{\Theta}_k \sqsubseteq \mathbf{\Theta}_{k-1}$
  if $\mathcal{T}_N\,(x(k),\mathbf{\Theta}_k|\mathcal{P}_N(k)) \neq \emptyset$ **then**
    Solve (6.6) to obtain $\mathbf{T}^\star \in \mathcal{T}_N\,(x(k),\mathbf{\Theta}_k|\mathcal{P}_N(k))$
    Apply $u(k) = K^\star_0\,(x(k),\theta(k)) = u^\star_0$ to the system (6.1)
    $k \leftarrow k+1$
  else
    abort
  end if
end loop

Theorem 6.2. *Suppose that the terminal set X_f is controlled λ-contractive in the sense of Definition 2.13 with $K_\text{f} : X_\text{f} \times \Theta \to \mathbb{U}$ being the corresponding local controller. Furthermore, let $\mathbf{\Theta} = \left\{\{\theta\}, \left\{\Theta_i\right\}_{i=1}^{N-1}\right\} \subseteq \Theta^N$ and $\mathbf{\Theta}^+ = \left\{\Theta^+_i\right\}_{i=0}^{N-1} \subseteq \Theta^N$ be two scheduling tubes satisfying $\mathbf{\Theta}^+ \sqsubseteq \mathbf{\Theta}$. Then, the DHC Algorithm 6.3 satisfies the following properties:*

(i) It is recursively feasible, i.e., if $\exists \mathbf{T} \in \mathcal{T}_N\,(x,\mathbf{\Theta}|\mathcal{P}_N)$, then $\exists \mathbf{T}^+ \in \mathcal{T}_{N-1}\left(x^+,\mathbf{\Theta}^+\middle|\mathcal{P}^+_{N-1}\right)$ where $x^+ = A(\theta)x + BK_0(x,\theta)$ and where $\mathcal{P}^+_{N-1}$ is constructed as in Step 12 of Algorithm 6.3.

(ii) The state of the controlled system reaches X_f in N steps or less, i.e., $\exists k_\star \in \mathbb{N}_{[0,N]}$ such that $\forall k \geq k_\star :\ x(k) \in X_\text{f}$.

(iii) The origin of (8.1) *is regionally asymptotically stabilized in the sense of Definition 2.8.*

Proof of (i). Recall that the tube $\mathbf{T} \in \mathcal{T}_N\,(x,\mathbf{\Theta}|\mathcal{P}_N(k))$ synthesized at time k is

$$\mathbf{T} = (\{\{x\}, X_1, \ldots, X_N\}, \{K_0, \ldots, K_{N-1}\})$$

with $X_N \subseteq X_\text{f}$. Then, at time $k+1$, the length of the synthesized tube is reduced to $N-1$. Therefore, and because $\mathbf{\Theta}^+ \sqsubseteq \mathbf{\Theta}$ and $x^+ \in X_1$, a successor tube can be constructed as

$$\mathbf{T}^+ = \left(\left\{\left\{x^+\right\}, X_2, \ldots, X_N\right\}, \{K_1, \ldots, K_{N-1}\}\right).$$

Algorithm 6.3 DHC sparsely parameterized TMPC.

Require: $N \in [1, \infty)$ and an initial condition set $X_0 \subseteq \mathbb{X}$

Off-line part

Compute the patterns $\mathrm{spar}\left(\mathbf{p}_{\mathrm{sp}}^{1}\left(\bar{x}_0^i, \bar{\theta}^j\right)\right)$, $(i, j) \in \mathbb{N}_{[1,q_0]} \times \mathbb{N}_{[1,q]}$ as described in Section 6.3.3

On-line part

$k \leftarrow 0$
Find $\xi\,(x(k))$ and $\zeta\,(\theta(k))$ by solving (6.28a)-(6.28b)
For all $i \in [0..N-1]$, compute $\mathrm{SPAR}_i(x(k), \theta(k))$ according to (6.30)
Construct $\mathscr{P}_N(k) := \mathscr{P}_N(x(k), \theta(k))$ according to (6.32)
loop
    if $N > 0$ **then**
        Construct $\mathbf{\Theta}_k = \left\{\{\theta(k)\}, \left\{\Theta_{i|k}\right\}_{i=1}^{N-1}\right\} \subseteq \Theta^N$ such that $\mathbf{\Theta}_k \sqsubseteq \mathbf{\Theta}_{k-1}$
        if $\mathcal{T}_N\,(x(k), \mathbf{\Theta}_k | \mathscr{P}_N(k)) \neq \emptyset$ **then**
            Solve (6.6) to obtain $\mathbf{T}^\star \in \mathcal{T}_N\,(x(k), \mathbf{\Theta}_k | \mathscr{P}_N(k))$
            Apply $u(k) = K_0^\star\,(x(k), \theta(k)) = u_0^\star$ to the system (6.1)
            $\mathscr{P}_{N-1}(k+1) \leftarrow \{(\mathbb{P}(1), P(\cdot|1)), \ldots, (\mathbb{P}(N), P(\cdot|N))\}$
            $N \leftarrow N - 1$
        else
            abort
        end if
    else
        Apply $u(k) = K_{\mathrm{f}}\,(x(k), \theta(k))$ to the system (6.1)
    end if
    $k \leftarrow k + 1$
end loop

By its construction, the tube $\mathbf{T}^+$ can be parameterized in the shifted parameterization structure $\mathcal{P}^+_{N-1}$, i.e., $\mathbf{T}^+ \in \mathcal{T}_{N-1}\left(x^+, \mathbf{\Theta}^+ \middle| \mathcal{P}^+_{N-1}\right)$. Observe that this result is independent of how the parameterization structure $\mathcal{P}_N$ was chosen initially.

Proof of (ii). Given the property of recursive feasibility, finite-time convergence to X_f is guaranteed by the fact that at every time instant the last cross section of the constructed tube is inside X_f. Eventually the length of the tube is reduced to 1, which implies that after at most N steps the initial state of the closed-loop system is in X_f as well.

Proof of (iii). After N time steps, once the state of the controlled system has reached X_f, asymptotic convergence of the state to the origin is ensured by the local controller $K_\mathrm{f}(\cdot,\cdot)$. □

6.4 Numerical examples

The purpose of this section is to demonstrate the computational properties of the proposed method on two different example systems. For each considered system, the following sets are calculated:

- The terminal set X_f. Specific details of how this set is calculated are discussed for each example.
- The initial state set X_0 is computed according to the procedure of (6.20)–(6.22), starting with $\hat{X}_0 = X_\mathrm{f}$.
- The controlled invariant set $\mathcal{X}_N^{\max}(\Theta^N)$ is computed using the iterative approach of Algorithm 2.1.
- An approximate domain of attraction $\mathcal{X}_N^{\mathrm{dense}} = \mathcal{X}_N(\Theta^N | \overline{\mathcal{P}}_N)$ is calculated for the full complexity controller (i.e., a controller using the dense structure defined by (6.8), (6.10), and (6.11)). The approximation is computed according to the procedure of Appendix B.3 with the initial set $\hat{\mathcal{X}}_N^{\mathrm{init}}$ being equal to the union of the vertices of X_f, $\mathcal{X}_N^{\max}(\Theta^N)$ and $\mathbb{X}$.
- An approximate domain of attraction $\mathcal{X}_N^{\mathrm{red}} = \mathcal{X}_N(\Theta^N | \mathcal{P}_N(\cdot,\cdot))$ of the reduced complexity controller (i.e., a controller using the sparse parameterization (6.32) proposed in this chapter) is obtained by using again the method of Appendix B.3. The choice for $\hat{\mathcal{X}}_N^{\mathrm{init}}$ is the same as for the computation of $\mathcal{X}_N^{\mathrm{dense}}$.

After the design has been carried out, the computational performance is verified by solving the tube synthesis problem for randomly generated samples (x, θ) from $\mathcal{X}_N^{\mathrm{red}} \times \Theta$. Details of the computer used for this simulation study are displayed in Table 6.1. The computation times for solving the LPs as reported by the solver are shown (Gurobi Optimization 2017). In case of the reduced-complexity controllers, the time needed for solving (6.28a)-(6.28b) is also included.

In order to investigate the computational properties of the proposed method, the tube synthesis problems are solved using different LP algorithms that are implemented in the solver: dual simplex, primal simplex, interior point (barrier), and automatic. In the automatic case the solver decides which one of the three methods to use based on its own analysis of the problem. The small LPs (6.28) are always solved using the automatic setting.

CPU	Intel Core i7-4790 @ 3.60 GHz
Memory	8 GB
Operating system	GNU/Linux 64-bit
MATLAB version	9.3.0.713579 (R2017b)
LP solver	Gurobi 7.0.2

Table 6.1: Computer details.

	X_{f}	$\mathcal{X}_N^{\max}$	$\hat{\mathcal{X}}_N^{\mathrm{dense}}$	$\hat{\mathcal{X}}_N^{\mathrm{red}}$
Volume	2.747	14.79	13.87	13.72

Table 6.2: Example 1: set volumes.

6.4.1 Numerical example 1: second-order system

Consider the system from (Casavola et al. 2012)

$$x(k+1) = \left(\begin{bmatrix} 2 & -0.1 \\ 0.5 & 1 \end{bmatrix} \theta_1(k) + \begin{bmatrix} 1 & 0.1 \\ 2.5 & 1 \end{bmatrix} \theta_2(k) \right) x(k) + \begin{bmatrix} 1 \\ -0.3 \end{bmatrix} u(k),$$

with the corresponding scheduling and constraint sets given by

$$\begin{aligned} \Theta &= \operatorname{convh} \left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}, \\ \mathbb{U} &= \{ u \in \mathbb{R} \mid |u| \leq 1 \}, \\ \mathbb{X} &= \left\{ x \in \mathbb{R}^2 \,\middle|\, |x_1| \leq 1, |x_2| \leq 5 \right\}. \end{aligned}$$

A 0.90-contractive terminal set with the specified fixed complexity of 10 vertices was computed using the method of Appendix A. The prediction horizon is set to $N = 6$.

The sets shown in Figure 6.1 were computed as explained before in the introduction of Section 6.4. It is observed that, as expected, the DOA $\hat{\mathcal{X}}_N^{\mathrm{red}}$ for the reduced-complexity controller is larger than X_0, but slightly smaller than the DOA $\hat{\mathcal{X}}_N^{\mathrm{dense}}$ of the full-complexity controller. The corresponding set volumes are shown in Table 6.2.

In Table 6.3, the achieved reduction in decision variables in the tube synthesis problem with respect to the dense parameterization structure $\overline{\mathcal{P}}_N$ is shown. The number of DOF refers to the vertex control actions and convex multipliers that are possibly eliminated from the sparse reduced-complexity parameterization structure, whereas the total number n_{d} of decision variables in the tube synthesis problem is slightly higher because this also includes the parameters $\mathbf{p}^2$ that are not eliminated by the sparse parameterization design (e.g., the centers and scalings of the tube cross sections). The "simulated worst"-column shows the greatest number of decision variables that resulted when solving the tube synthesis problem for 125 randomly generated initial states and scheduling values. In addition, the column "guaranteed worst" in Table 6.3 lists the number of variables that would result when combining *all* the

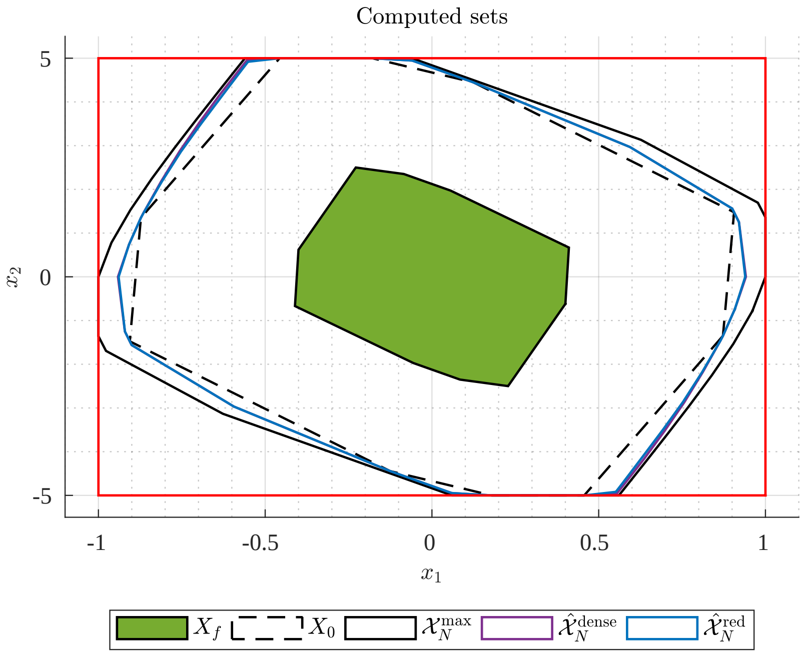


Figure 6.1: Example 1: computed sets.

	Dense	Reduced (simulated worst)	Reduced (guaranteed worst)
Vertex control actions $u_i^{(j,l)}$	100	99	100
Nominal control actions $\bar{u}_i$	1	6	6
Convex multipliers	1110	301	462
Total DOF	1211	406	562
Ratio DOF	1	0.34	0.46
Total n_d	1251	441	603
Ratio n_d	1	0.35	0.48
Total n_eq	224	224	224
Total n_ineq	1128	1128	1128

Table 6.3: Example 1: complexity of the tube synthesis problems.

patterns that were discovered in the design phase. This gives an obvious but pessimistic upper bound on the maximum complexity that can possibly occur in practice. The amount of equality and inequality constraints (n_eq and n_ineq) in the tube synthesis problem, which is not affected by the sparse design, is also shown.

In this example, by applying the sparse parameterization, the number of decision variables could be reduced from 1251 to 441. That is, the number of decision variable in the tube synthesis problem when using the sparse parameterization is approximately 35% of the number of variables when using the "dense" parameterization. It is noted that almost the full reduction in the number of variables in this case results from the elimination of unnecessary convex multipliers. Furthermore, it is worth noting that by using the sparse parameterization the number of variables in the tube synthesis problem is always guaranteed to be reduced to at most 48% of its original value.

The effect of the reduction of decision variables on the computation time was investigated by solving the tube synthesis problem for 125 random initial state and scheduling values sampled from $\hat{\mathcal{X}}^\mathrm{red} \times \Theta$. The performance of three different LP algorithms implemented in the Gurobi solver is compared as explained in the introduction to this section. The computational results are visualized in Figure 6.2 by means of box plots. In these plots, the bottom and top of the box respectively indicate the 25th and 75th percentiles of the measured computation times (i.e., 25% of the measured times are below the lower edge of the box, and 75% of the measured times are below the top edge of the box). A line inside of each box indicates the median. The lines that extend from the bottom and top of the boxes indicate the full range of the measured computation times, but excluding times that are considered to be outliers. Outliers (i.e., unusually high or low computation times) are depicted by the "+"-symbol.

On this problem, the dual simplex method shows the best performance both in the dense- and reduced-complexity cases. A concise summary of the computation times for the dual simplex-case is given in Table 6.4. From Table 6.2 it can be concluded that using the sparse tube parameterization from this chapter leads to a reduction of approximately 29% in the mean

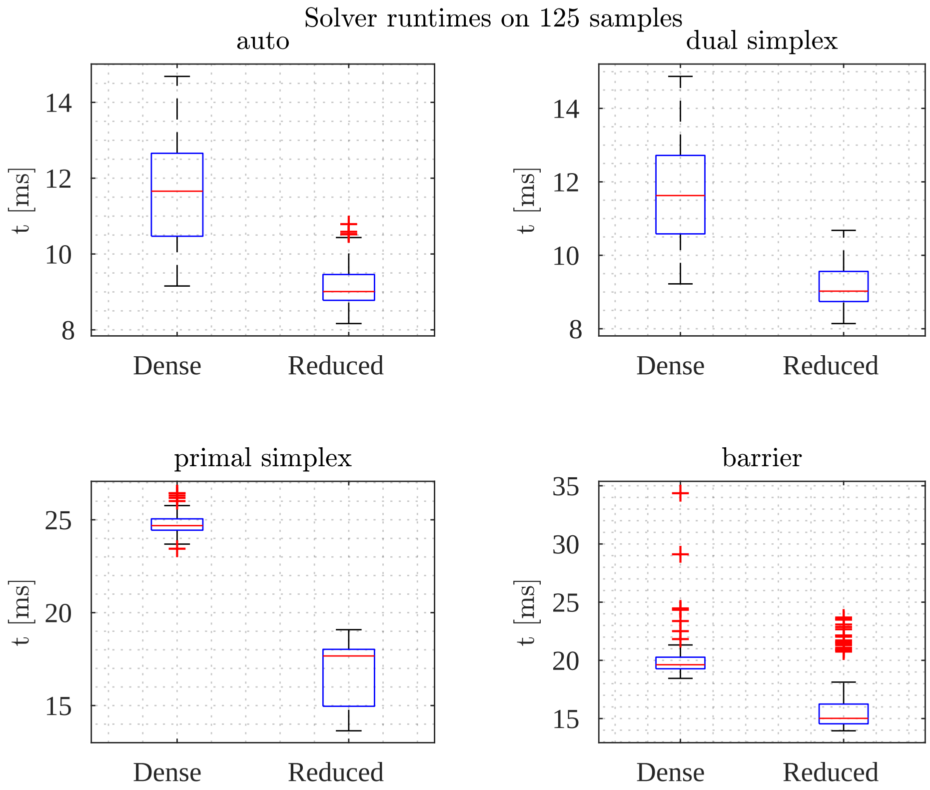


Figure 6.2: Example 1: reported solver times for various LP algorithms.

computation time required to solve the tube synthesis problem, and to a reduction of 35% of the worst-case measured computation time.

6.4.2 Numerical example 2: third-order system

Consider the third-order system

$$x(k+1) = \left(\begin{bmatrix} 1 & 1 & 0.5 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \\ 0 & 0 & 0 \end{bmatrix} \theta_1(k) + \begin{bmatrix} 0 & 0 & 0.2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \theta_2(k) \right) x(k) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(k),$$

	Mean solvertime (ms)	Max. solvertime (ms)
Dense $\overline{\mathcal{P}}_N$ with dual simplex	11.8	15.9
Reduced $\mathcal{P}_N(x,\theta)$ with dual simplex	8.37	10.2

Table 6.4: Example 1: summary of reported solver times.

	X_f	$\mathcal{X}_N^{\max}$	$\hat{\mathcal{X}}_N^{\mathrm{dense}}$	$\hat{\mathcal{X}}_N^{\mathrm{red}}$
Volume	18.79	170.3	170	169.5

Table 6.5: Example 2: set volumes.

with the corresponding scheduling and constraint sets given by

$$\begin{aligned}
\Theta &= \mathrm{convh}\left\{\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix}\right\}, \\
\mathbb{U} &= \{u \in \mathbb{R} \mid |u| \leq 4\}, \\
\mathbb{X} &= \left\{x \in \mathbb{R}^2 \,\middle|\, |x_1| \leq 5, |x_2| \leq 3, |x_3| \leq 8\right\}.
\end{aligned}$$

A 0.90-contractive terminal set with the specified fixed complexity of 10 vertices was computed using the method of Appendix A. The prediction horizon is $N = 8$.

The volumes of the resulting sets, which were computed as explained before in the introduction of Section 6.4, are shown in Table 6.5. The numbers of decision variables and constraints in the tube synthesis problems are shown in Table 6.6. The total number of decision variables in the tube synthesis problem is reduced from 7586 to 2566, i.e., to 34% of the original value. In contrast to the previous example, some complexity reduction is also achieved from the elimination of unnecessary vertex control actions.

The computational results are visualized in Figure 6.3 in terms of box plots (see the previous example in Section 6.4.1 for an explanation of the box plots). A summary of these results is given in Table 6.7. In this example, it is noted that the interior point (barrier) algorithm performs the fastest on the full-size problem. However, for the reduced-complexity case, the dual simplex method gives the least computation times. From Table 6.7, it is concluded that in this case the sparse tube parameterization leads to a reduction of approximately 30% in the mean computation time required to solve the tube synthesis problem, and to a reduction of about 44% in the worst-case measured computation time.

In this example, it is interesting and somewhat surprising to note that the computational performance of the interior point method actually deteriorates when using the sparse parameterization. It can therefore be concluded that the obtained performance improvements can be dependent on the algorithm that is used to solve the LP.

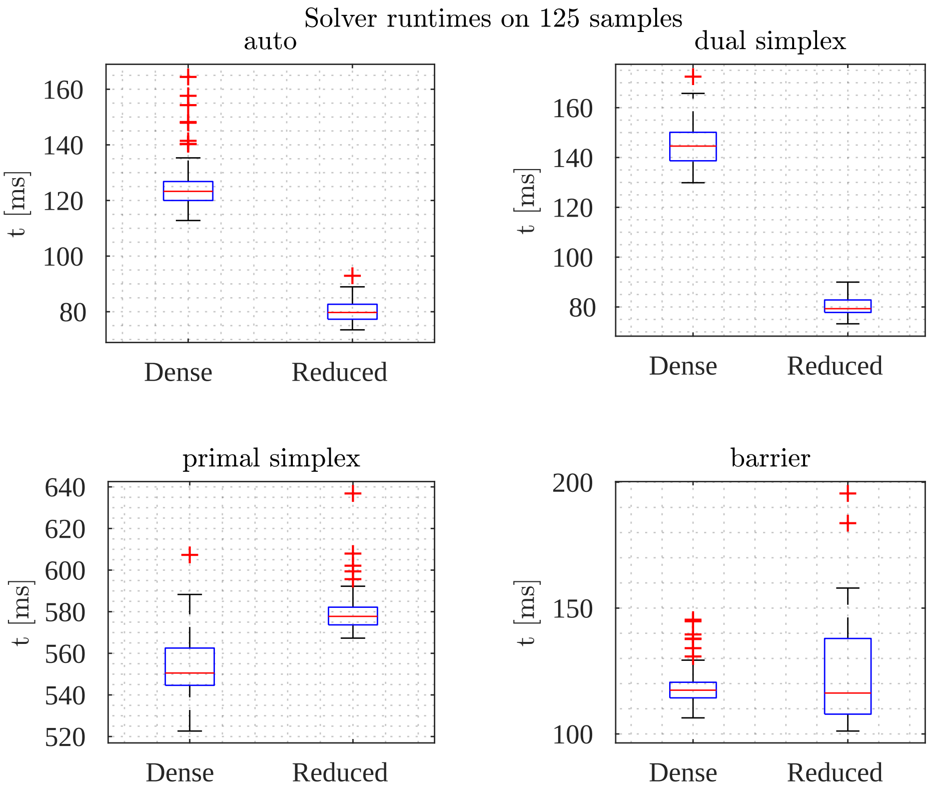


Figure 6.3: Example 2: reported solver times for different LP algorithms.

	Dense	Reduced (simulated worst)	Reduced (guaranteed worst)
Vertex control actions	378	204	308
Nominal control actions	1	8	8
Convex multipliers	7146	2300	3063
Total DOF	7525	2512	3371
Ratio DOF	1	0.33	0.45
Total n_d	7586	2566	3433
Ratio n_d	1	0.34	0.45
Total n_eq	1194	1194	1194
Total n_ineq	4956	4956	4956

Table 6.6: Example 2: complexity of the tube synthesis problems.

	Mean solvertime (ms)	Max. solvertime (ms)
Dense $\overline{\mathscr{P}}_N$ with interior point	121	168
Reduced $\mathscr{P}_N(x,\theta)$ with dual simplex	84.8	93.6

Table 6.7: Example 2: summary of reported solver times.

6.5 Concluding remarks

An approach for designing a computationally efficient tube parameterization that guarantees initial feasibility on an assigned set of initial conditions was presented. Sparsity-promoting reweighted ℓ_1-optimization was employed to determine which decision variables can be eliminated from the tube parameterization. Although the method as presented does not guarantee recursive feasibility when used in an RHC scheme, a recursively feasible DHC variant was constructed.

The result of this chapter is an alternative answer to Research Question Q_3, besides the answer that was provided in Chapter 5. In Chapter 5 a new degree of freedom was introduced in the tube parameterization design by allowing the parameterization to be heterogeneous along the prediction horizon. The approach of the current chapter introduces a different design degree of freedom, namely the choice of an initial condition set: an efficient tube parameterization that guarantees feasibility on this set is then computed automatically. Some directions for further research on this approach are the following:

- In the approach of this chapter, to achieve a reduction of computational complexity, decision variables were eliminated from the tube synthesis problem. An important next step that has to be made to improve the method further, is to study the elimination of redundant constraints as well.
- In numerical experiments, it was observed that the computation time required to solve

the "reduced" tube synthesis problem can sometimes increase when using an interior point solver. It would be interesting to study further why this is the case, and to obtain a more complete understanding of how the structure of the tube synthesis problem affects the relative performance of different linear programming algorithms.

- The effect of the reduced-complexity parameterization on control performance as measured in terms of the closed-loop cost must be studied as well.

Lastly, it is remarked that if the H-representation of the set S is available, the multipliers p^λ can be removed from the tube parameterization in Definition 6.3. Then, the tube dynamics constraints can be checked in the same way as was done in Section 4.4.2. It is then still possible to apply the method from this chapter to obtain sparse control parameterizations, but the reduction in decision variables will not be as large.

Chapter 7

Towards constrained LPV reference tracking

This chapter addresses constrained output tracking of constant references for linear parameter-varying systems. In model predictive control based on linear parameter-varying state-space representations, it is recognized that offset-free output tracking is generally not possible to achieve due to the variations in the scheduling signal. Therefore, in this chapter, it is proposed instead to guarantee a pre-specified tracking error bound ϵ which is achievable for all admissible variations of the scheduling variable. The construction of an invariant set in which such a bound can be satisfied is described. Subsequently a tube-based model predictive controller is designed which brings the state of the system inside this set in finite time. The properties of the approach are demonstrated on two numerical examples.

7.1 Introduction

The model predictive control (MPC) approaches described in the previous chapters were all aimed at asymptotically stabilizing the origin of a given linear parameter-varying (LPV) system. Besides stabilization of the state around the origin, it is often desired to bring the *output* of the system to a specified value, leading to a reference tracking problem. Related to the LPV case, robust tracking MPC approaches for linear parametrically uncertain (LPU) systems were developed in, e.g., (Pannocchia 2004; Y. J. Wang and Rawlings 2004). It was shown that, with these methods, offset-free tracking is achieved if the uncertainty (corresponding to θ in an LPV system) is time-invariant. However, no hard bounds on the tracking error have been given for the general case when θ is a time-varying signal.

When controlling a constrained LPV system described by a state-space representation, it is usually impossible to achieve offset-free output tracking even of constant reference signals because variations in θ also influence the output. Indeed, in MPC, offset-free tracking can generally only be achieved for (asymptotically) constant disturbances or uncertainties (Pannocchia et al. 2015). Then, it becomes a natural approach to look for a set *around* the desired reference to which convergence can be established.

Previously, in the linear time-invariant (LTI) case, (Alvarado et al. 2007) considered the tracking of piecewise constant references for LTI systems subject to additive disturbances using a tube-based approach. Asymptotic convergence of the output towards a bounded region around the reference was established. The paper (Betti et al. 2013) presents similar results using prediction models in the velocity form. In (Falugi and Mayne 2013), undisturbed LTI systems with randomly time-varying references are considered: again, a bounded set to which the tracking error converges is characterized.

An alternative form of LTI tracking MPC, guaranteeing strict error bounds, has been recently proposed (Di Cairano and Borrelli 2016). The reference is assumed to be generated by a *reference generator*, which itself is a constrained LTI system subject to bounded additive disturbances. An invariant set is designed in a lifted state/reference space. When the state of the system and the state of the reference generator are inside this set, the tracking error is contained within an ϵ-ball for all possible reference trajectories. An MPC law is designed which brings the state of the system inside this set, provided that the initial tracking error is already within the specified ϵ-ball.

To resolve the MPC tracking problem for LPV systems, this chapter adopts the basic reasoning of (Di Cairano and Borrelli 2016) and modifies it to handle the LPV setting. It is recognized that offset-free tracking is generally impossible when θ is time-varying. An invariant set in which the tracking error with respect to a given *constant* reference satisfies a pre-specified bound is derived. Then, a tube-based predictive controller based on the concepts detailed in Chapter 3 is designed to control the state of the system towards this set. To ensure finite-time convergence, a time-varying terminal constraint is employed. The finite-horizon cost function is designed to optimize local tracking performance. In this proposed method, which can serve as a stepping stone towards the development of more sophisticated constrained LPV output reference tracking controllers, only constant references are considered in lieu of the more general signal class from (Di Cairano and Borrelli 2016).

The remainder of this chapter is structured as follows. The notation and the problem setting are introduced in Section 7.1.1 and Section 7.1.2, respectively. Next, Section 7.2 introduces the concept of bounded-error invariant sets, which will be used as the basis of the MPC approach that is developed in Section 7.3. The properties of the approach are demonstrated on some numerical examples in Section 7.4, and the chapter ends with some concluding remarks in Section 7.5.

7.1.1 Notation

In this chapter, all vector norms refer to the ∞-norm, i.e., $\|\cdot\| = \|\cdot\|_\infty$. A closed ϵ-ball is defined as

$$\mathbb{B}(\epsilon) := \{x \mid \|x\| \leq \epsilon\}$$

where the dimension of the space will be clear from the context.

The abbreviation "CPI set" refers to a controlled positively invariant set in the sense of Definition 2.13.

7.1.2 Problem setting

This chapter considers constrained polytopic LPV state-space (LPV-PSS) representations (see Definitions 2.6 and 2.7) of the form

$$\begin{aligned} x(k+1) &= A(\theta(k))x(k) + Bu(k), \\ y(k) &= C(\theta(k))x(k). \end{aligned} \tag{7.1}$$

For simplicity, the direct feedthrough matrix is omitted and similarly to Chapter 4 the input matrix is assumed to be constant. This latter choice is made mainly for implementation reasons; the discussion of Section 4.6.1 on how to handle system representations with a parameter-varying B applies to the situation in the current chapter as well. In contrast to the previous chapters, an output equation $y(k) = C(\theta(k))x(k)$ is present, and constraints on the output signal $y : \mathbb{N} \to \mathbb{Y}$ are assumed.

Assumption 7.1. *With respect to the output constraint set $\mathbb{Y}$, assume the following:*

(i) The output constraint set $\mathbb{Y} \subset \mathbb{R}^{n_y}$ is a PC-set.

(ii) The state- and output constraint sets are consistent, i.e., $\forall (x, \theta) \in \mathbb{X} \times \Theta : \; y = C(\theta)x \in \mathbb{Y}$.

Due to Assumption 7.1.(ii), it is sufficient to consider only the two constraint sets $\mathbb{X}$ and $\mathbb{U}$ explicitly, as any output constraints $\mathbb{Y}$ are implicitly satisfied through satisfaction of $\mathbb{X}$. The following assumption is also made.

Assumption 7.2. *In* (7.1) *the IO dimensions satisfy* $1 \le n_y \le n_u$, *and*

$$\forall \theta \in \Theta : \; \operatorname{rank} \begin{bmatrix} A(\theta) - I & B \\ C(\theta) & 0 \end{bmatrix} = n_x + n_y .$$

Assumption 7.2 implies that for any reference $r \in \mathbb{Y}$ and any constant scheduling value $\theta \in \Theta$, it is possible to find a steady state/input pair $(\bar{x}(\theta, r), \bar{u}(\theta, r))$ such that $C(\theta)\bar{x}(\theta, r) = r$. This will prove to be useful later in the design of a tracking MPC cost function. The following problem is considered in this chapter.

Problem 7.1. *Given a reference value $r \in \mathbb{Y}$ and a bound $\epsilon \in \mathbb{R}_+$, design a controller which (i) brings the tracking error $e(k) = y(k) - r$ into the ϵ-ball $\mathbb{B}(\epsilon)$ in finite time, and (ii) subsequently keeps it there despite of the variations of $\theta \in \Theta$.*

This problem is different from Problem 3.1, because asymptotic stability of the origin is now not required.

7.2 Bounded-error invariant set

This section introduces the concept of a bounded-error invariant set, which will be used in the development of a tracking tube-based LPV MPC strategy. In an LPV system, due to the variations in θ, it is generally impossible to achieve offset-free tracking even for constant references r. Instead, it is proposed to keep the tracking error

$$e(k) := y(k) - r = C(\theta(k))x(k) - r$$

within a certain bound which can be maintained for all admissible scheduling trajectories. This bound will be characterized in terms of an ϵ-ball $\mathbb{B}(\epsilon)$. It is first assumed that a reference r and a bound $\epsilon \in \mathbb{R}_+$ are given. The case when only a desired error bound is specified a-priori is considered later. To construct a bounded-error invariant set, first define augmented state constraints as

$$\mathbb{X}_{\mathcal{E}}(r,\epsilon) := \mathbb{X} \cap \mathcal{E}(r,\epsilon)$$

where

$$\mathcal{E}(r,\epsilon) := \{x \in \mathbb{R}^{n_\mathrm{x}} \mid \forall\theta \in \Theta: \ (C(\theta)x - r) \in \mathbb{B}(\epsilon)\}$$

contains all states that render the constraint $e(k) \in \mathbb{B}(\epsilon)$ satisfied irrespective of θ. Then, the set

$$P(r,\epsilon) := \text{CPI set for (7.1) w.r.t. } \mathbb{X}_{\mathcal{E}}(r,\epsilon) \times \mathbb{U} \tag{7.2}$$

is a controlled invariant set inside of which the tracking error constraint $e(k) \in \mathbb{B}(\epsilon)$ is satisfied. Therefore, this set can be used as the target set that is to be reached within finite time using the proposed MPC solution.

It is possible that there exist combinations of reference values r and error bounds ϵ for which such a set does not exist (e.g., if the ϵ is chosen too small). This gives rise to the following notion of ϵ-achievable reference.

Definition 7.1. *Let $\epsilon \in \mathbb{R}_+$ be given. A reference value $r \in \mathbb{Y}$ is called* ϵ-achievable *for the system* (7.1) *and with respect to the constraints $\mathbb{X} \times \mathbb{U}$ if $P(r,\epsilon) \neq \emptyset$.*

For the computation of (7.2) it was assumed that both the reference and the error bound were known and fixed. It is also possible to define the set of all references which are ϵ-achievable for (7.1) in the sense of Definition 7.1 as

$$\mathcal{R}(\epsilon) := \{r \in \mathbb{Y} \mid P(r,\epsilon) \neq \emptyset\}. \tag{7.3}$$

To characterize this set, consider the extended system

$$\begin{aligned}\begin{bmatrix} x(k+1) \\ r(k+1) \end{bmatrix} &= \begin{bmatrix} A(\theta(k)) & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} x(k) \\ r(k) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(k) \\ e(k) &= \begin{bmatrix} C(\theta(k)) & -I \end{bmatrix} \begin{bmatrix} x(k) \\ r(k) \end{bmatrix}\end{aligned} \tag{7.4}$$

and the associated $(n_\mathrm{x} + n_\mathrm{y})$-dimensional constraint set

$$\tilde{\mathbb{X}}_{\mathcal{E}}(\epsilon) := (\mathbb{X} \times \mathbb{Y}) \cap \tilde{\mathcal{E}}(\epsilon)$$

with

$$\tilde{\mathcal{E}}(\epsilon) := \left\{ \begin{bmatrix} x \\ r \end{bmatrix} \in \mathbb{R}^{n_\mathrm{x}+n_\mathrm{y}} \middle| \forall\theta \in \Theta: \ \begin{bmatrix} C(\theta) & -I \end{bmatrix} \begin{bmatrix} x \\ r \end{bmatrix} \in \mathbb{B}(\epsilon) \right\}$$

containing all pairs (x, r) satisfying the error bound for all $\theta \in \Theta$. Note that in (7.4), if the "initial" reference is $r(0) = r_0$, then $r(k) = r_0$ for all $k \geq 0$. This system can be viewed as a special case of the setup from (Di Cairano and Borrelli 2016), where the reference evolves according to more general non-autonomous constrained LTI dynamics. The invariant set

$$\tilde{P}(\epsilon) := \text{CPI set for (7.4) w.r.t. } \tilde{\mathbb{X}}_{\mathcal{E}}(\epsilon) \times \mathbb{U} \tag{7.5}$$

can now be computed. It is a subset of an extended state/reference space of dimension $n_{\text{x}} + n_{\text{y}}$. The corresponding set $\mathcal{R}(\epsilon)$ is found by projecting $\tilde{P}(\epsilon)$ onto the reference space, i.e.,

$$\mathcal{R}(\epsilon) = \left\{ r \in \mathbb{Y} \;\middle|\; \exists x \in \mathbb{X} : \begin{bmatrix} x \\ r \end{bmatrix} \in \tilde{P}(\epsilon) \right\}.$$

Similarly, given $\tilde{P}(\epsilon)$ and a $r \in \mathcal{R}(\epsilon)$, the corresponding set $P(r, \epsilon)$ from (7.2) can be recovered by projection as

$$P(r, \epsilon) = \left\{ x \in \mathbb{X} \;\middle|\; \begin{bmatrix} x \\ r \end{bmatrix} \in \tilde{P}(\epsilon) \right\}. \tag{7.6}$$

The projection in (7.6) can be computed efficiently, because it only considers one fixed reference r. If $r \neq \mathcal{R}(\epsilon)$, i.e., the reference is not ϵ-achievable, the result will be $P(r, \epsilon) = \emptyset$.

Finally, for a given reference r, it can be useful to know the minimal ϵ for which $P(r, \epsilon) \neq \emptyset$. This minimal achievable error bound is defined as

$$\epsilon^\star(r) := \inf \{\epsilon \geq 0 \mid P(r, \epsilon) \neq \emptyset\} \tag{7.7}$$

and the corresponding invariant set is denoted by

$$P^\star(r) := P(r, \epsilon^\star(r)). \tag{7.8}$$

In principle, the optimal value $\epsilon^\star(r)$ in (7.7) can be computed by a bisection search. Such a procedure can be carried out relatively efficiently for low-order systems, where invariant sets of the form (7.2) can be computed reasonably fast using one of the methods described in Section 2.2.5.

7.3 The tracking TMPC algorithm

When a set $P(r, \epsilon)$ has been obtained using the procedures from the previous section, it can be used as a terminal set in an MPC algorithm. If the MPC manages to bring the state of the system into this set and keep it there (which should be possible due to the invariance property), it cam provide a solution to Problem 7.1. In this section, such an MPC is developed based on the synthesis of homothetic tubes which were introduced in Section 3.5.

To obtain the desired finite-time convergence to $P(r, \epsilon)$, for a tube synthesized at time instant k, a time-varying terminal constraint

$$\forall i \in \mathbb{N}_{[\max\{0, N-k\}, N]} : \; X_i \subseteq P(r, \epsilon) \tag{7.9}$$

is proposed which at the initial time $k = 0$ constrains the state of the system to be inside $P(r, \epsilon)$ after N steps. At time $k + 1$, the state is required to reach $P(r, \epsilon)$ in $N - 1$ steps: the pattern continues until it is finally required that the initial state x is inside $P(r, \epsilon)$.

This leads to a set of feasible tubes which depends on the reference r, on the selected error bound ϵ, and on the current time instant k, i.e., to

$$\mathcal{T}_N\left(x, \boldsymbol{\Theta}|r, \epsilon, k\right) = \big\{\mathbf{T} \,\big|\, \mathbf{T} \text{ satisfies Def. 3.9 with } X_0 = \{x\} \\ \text{and } \forall i \in \mathbb{N}_{[\max\{0, N-k\}, N]} : \; X_i \subseteq P(r, \epsilon)\big\} \quad (7.10)$$

and an associated tube synthesis problem

$$V\left(x, \boldsymbol{\Theta}|r, \epsilon, k\right) = \min_{\mathbf{T}} J_N\left(\mathbf{T}, \boldsymbol{\Theta}|r, k\right) \text{ subject to } \mathbf{T} \in \mathcal{T}_N\left(x, \boldsymbol{\Theta}|r, \epsilon, k\right). \quad (7.11)$$

The cost function used in this chapter does not fit into the class presented in Section 3.4.5. Instead the tracking stage cost

$$\ell\left(X, K, \Theta|r\right) = \max_{(x,\theta)\in X\times\Theta} \|Q\left(x - \bar{x}(r,\theta)\right)\| + \|R\left(K(x,\theta) - \bar{u}(r,\theta)\right)\| \quad (7.12)$$

is used, where $\theta = \Theta_0 = \{\theta(k)\}$ is the measured value of the scheduling variable (see Assumption 2.1). The pair $(\bar{x}(r,\theta), \bar{u}(r,\theta))$ is computed according to

$$\begin{aligned}(\bar{x}(r,\theta), \bar{u}(r,\theta)) = \arg\min_{(\bar{x},\bar{u})} &\left\|\begin{bmatrix}\bar{x}^\top & \bar{u}^\top\end{bmatrix}\right\|_2^2 \\ \text{s.t.} &\begin{bmatrix}A(\theta) - I & B \\ C(\theta) & 0\end{bmatrix}\begin{bmatrix}\bar{x} \\ \bar{u}\end{bmatrix} = \begin{bmatrix}0 \\ r\end{bmatrix}\end{aligned} \quad (7.13)$$

which always has a solution due to Assumption 7.2. In (7.13), a least-norm cost function is selected, preferring solutions with small input energy whenever $n_\mathrm{u} > n_\mathrm{y}$. If $n_\mathrm{y} = n_\mathrm{u}$, there exists only one unique solution regardless of the minimization objective. Note that if $n_\mathrm{y} > \mathrm{u}$, Assumption 7.2 is violated, and it is possible that no solution to (7.13) exists.

The main reasoning behind selecting the stage cost (7.12) is that it aims to bring the output close to the reference if θ is slowly varying, and possibly even yields offset-free tracking if θ stops varying (i.e., if $\exists k_\star \in \mathbb{N}$ such that $\forall k \geq k_\star : \; \theta(k) = \bar{\theta}$).

Remark 7.1. *Offset-free tracking under constant θ is not guaranteed in the considered control solution. One principal reason is that this objective is not necessarily compatible with the bounded-error requirement that $e(k) \in \mathbb{B}(\epsilon)$ for all k. Indeed, it can happen that there exists $\theta \in \Theta$ for which $\bar{x}(r,\theta) \notin P(r,\epsilon)$: an example of such a case is given in Section 7.4.*

Because stability of the origin is of no concern, no terminal cost is being used and the stage cost does not have to satisfy any further properties such as being $\mathcal{K}_\infty$-lower bounded. Instead, finite-time convergence to the target set $P(r, \epsilon)$ is ensured by (7.9). The resulting reference-tracking algorithm is summarized in Algorithm 7.1 and its important properties are stated in Theorem 7.1.

Algorithm 7.1 The reference-tracking receding-horizon TMPC algorithm.

Require: $N \in [1..\infty)$, $\epsilon \in \mathbb{R}_+$, $r \in \mathcal{R}(\epsilon)$

$\boldsymbol{\Theta}_{-1} \leftarrow \Theta^N$

$k \leftarrow 0$

loop

  Construct $\boldsymbol{\Theta}_k = \left\{ \{\theta(k)\}, \left\{\Theta_{i|k}\right\}_{i=1}^{N-1} \right\} \subseteq \Theta^N$ such that $\boldsymbol{\Theta}_k \sqsubseteq \boldsymbol{\Theta}_{k-1}$

  if $\mathcal{T}_N\left(x(k), \boldsymbol{\Theta}_k | r, \epsilon, k\right) \neq \emptyset$ **then**

    Compute $(\bar{x}(r,\theta), \bar{u}(r,\theta))$ according to (7.13)

    Solve (7.11) to obtain $\mathbf{T}^\star \in \mathcal{T}_N\left(x(k), \boldsymbol{\Theta}_k | r, \epsilon, k\right)$

    Apply $u(k) = K_0^\star\left(x(k), \theta(k)\right) = u_0^\star$ to the system (7.1)

    $k \leftarrow k + 1$

  else

    abort

  end if

end loop

Theorem 7.1. *Let $\epsilon \in \mathbb{R}_+$ and $r \in \mathcal{R}(\epsilon)$ be given, and let $\boldsymbol{\Theta} \subseteq \Theta^N$, $\boldsymbol{\Theta}^+ \subseteq \Theta^N$ be two sequences related as $\boldsymbol{\Theta}^+ \sqsubseteq \boldsymbol{\Theta}$. Then, Algorithm 7.1 achieves the following properties:*

(i) It is recursively feasible, i.e., if $\exists \mathbf{T} \in \mathcal{T}_N(x, \boldsymbol{\Theta} | r, \epsilon, k)$, then $\exists \mathbf{T}^+ \in \mathcal{T}_N(x^+, \boldsymbol{\Theta}^+ | r, \epsilon, k+1)$ where $x^+ = A(\theta)x + BK_0(x,\theta)$. Here, (x,θ) are the measured values of the state and scheduling variables, and K_0 is the first controller in the tube $\mathbf{T}$ according to Definition 3.4.

(ii) The state of the controlled system reaches $P(r,\epsilon)$ in N steps or less, i.e., $\exists k_\star \in \mathbb{N}_{[0,N]}$ such that $\forall k \geq k_\star :\ x(k) \in P(r,\epsilon)$.

(iii) The tube synthesis problem (7.11) *is a linear program. The numbers of variables and constraints are linear functions of the prediction horizon N.*

Proof of (i). Recursive feasibility is established exactly in the same way as in Proposition 4.2 (with $M = 1$) or Theorem 5.1 (with a fully homothetic parameterization structure). The candidate feasible tubes used in the proofs of Proposition 4.2 and Theorem 5.1 are by construction already compatible with the time-varying terminal constraint that is employed in the current chapter. The only difference is that now the terminal set $P(r,\epsilon)$ is invariant instead of λ-contractive with $\lambda < 1$, but this does not have any consequences for the recursive feasibility proof.

Proof of (ii). When Property (i) is satisfied, this property is implied by the construction of the time-varying terminal constraint in the set of feasible tubes (7.10).

Proof of (iii). Implementation of (7.11) proceeds in the same way as described in Section 4.4, leading to Property (iii). The modifications necessary to use the tracking cost function (7.12) are straightforward and do not influence the $O(N)$-scaling of the number of variables and constraints. □

The property Theorem 7.1.(ii) is similar to that of the "decreasing horizon tube controller" from (Langson et al. 2004, Proposition 7), where finite-time convergence to a robustly invariant set is established for an LTI system subject to additive disturbances. The approach of this chapter, in contrast, varies the time at which the terminal constraint becomes active instead of the horizon length N itself. This preserves degrees of freedom to optimize local performance once $P(r, \epsilon)$ has been reached. Because the finite-time convergence of the tracking error is obtained by construction of the constraints and asymptotic stability is not of interest, no terminal cost is necessary in this particular MPC formulation.

Finally, observe that the proof of Theorem 7.1 does not require $e(k) \in \mathbb{B}(\epsilon)$, where $e(k) = x(k) - r$.

7.4 Numerical examples

Two numerical examples that demonstrate the tracking LPV algorithm developed in this chapter are now provided.

7.4.1 Minimal ϵ for fixed r

In this example, a fixed constant reference r is given, which is required to be tracked with the minimal achievable error bound $\epsilon^\star$ defined in (7.7). The system used is of the form (7.1) with

$$\begin{aligned} x(k+1) &= \left(\begin{bmatrix} 0.95 & 1 \\ 0 & -0.59 \end{bmatrix} + \begin{bmatrix} 0 & 0.5 \\ 0 & 0 \end{bmatrix} \theta_1(k) + \begin{bmatrix} 0 & 0 \\ 0.2 & 0 \end{bmatrix} \theta_2(k) \right) x(k) + \begin{bmatrix} 1 \\ 0.5 \end{bmatrix} u(k) \\ y(k) &= \left(\begin{bmatrix} 0.8 & -0.6 \end{bmatrix} + \begin{bmatrix} 0 & -0.03 \end{bmatrix} \theta_1(k) + \begin{bmatrix} 0.04 & 0 \end{bmatrix} \theta_2(k) \right) x(k) \end{aligned}$$

and with the scheduling- and constraint sets

$$\begin{aligned} \Theta &= \left\{ \theta \in \mathbb{R}^2 \mid \|\theta\| \le 1 \right\}, \\ \mathbb{X} &= \left\{ x \in \mathbb{R}^2 \mid -6 \le x_1 \le 4,\ -4 \le x_2 \le 6 \right\}, \\ \mathbb{U} &= \left\{ u \in \mathbb{R} \mid -1 \le u \le 2 \right\}. \end{aligned} \tag{7.14}$$

The constant reference to be tracked is chosen to be $r = -0.85$.

Using the bisection procedure proposed in Section 7.2, it was found that $\epsilon^\star = 0.37$. The corresponding set $P^\star(r)$ is shown in Figure 7.1. In light of Remark 7.1, the convex hull of the set of all steady states $\bar{\mathbb{X}}(r) = \left\{ \bar{x}(r, \theta) \mid \theta \in \Theta \right\}$ is also depicted. It can be clearly seen that there exist $\theta \in \Theta$ for which offset-free tracking of r is, by definition, impossible to achieve without violating the guarantee that $e(k) \in \mathbb{B}(\epsilon^\star)$ for all time given all possible trajectories of θ.

A tracking tube-based MPC was implemented according to the construction of Section 7.3. In this simulation, the following parameters were used:

- The prediction horizon was $N = 7$;
- The tuning parameters for the stage cost (7.12) were set as $Q = I$ and $R = 10$.

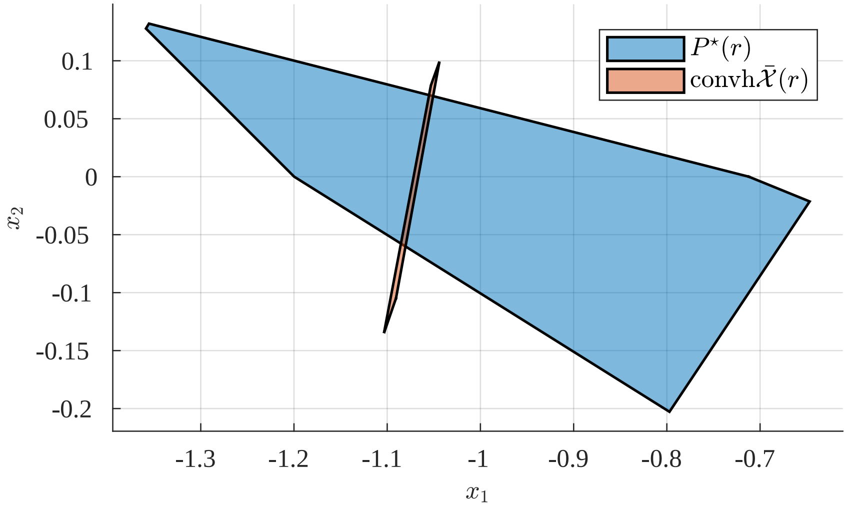


Figure 7.1: The sets $P^\star(r)$ and $\mathrm{convh}\{\bar{\mathbb{X}}(r)\}$ in the example of Section 7.4.1.

A simulation result for the initial state $x(0) = \begin{bmatrix} -4 & 3 \end{bmatrix}^\top$ is shown in Figure 7.2. It can be observed that the tracking error converges to $\mathbb{B}(\epsilon)$ in less than N steps as given by Theorem 7.1. The scheduling signal used in this simulation corresponds to a "harsh" scenario, with rapid variations consisting of random switches between the vertices of Θ.

7.4.2 All ϵ-achievable references

In this second example, a desired error bound ϵ is specified. Now, the purpose is to determine the set $\mathcal{R}(\epsilon)$ of all ϵ-achievable references according to (7.3). The system used in this example has the same dynamics as the system of Example 1. To make the computation of $\mathcal{R}(\epsilon)$ more interesting, a second input and output are added, which gives the system

$$\begin{aligned} x(k+1) &= \left(\begin{bmatrix} 0.95 & 1 \\ 0 & -0.59 \end{bmatrix} + \begin{bmatrix} 0 & 0.5 \\ 0 & 0 \end{bmatrix} \theta_1(k) + \begin{bmatrix} 0 & 0 \\ 0.2 & 0 \end{bmatrix} \theta_2(k) \right) x(k) + \begin{bmatrix} 1 & 0.7 \\ 0.5 & -1 \end{bmatrix} u(k) \\ y(k) &= \left(\begin{bmatrix} 1.2 & 1.0 \\ 0.7 & 0 \end{bmatrix} + \begin{bmatrix} 0 & -0.03 \\ 0.08 & 0 \end{bmatrix} \theta_1(k) + \begin{bmatrix} 0.04 & 0 \\ 0 & -0.06 \end{bmatrix} \theta_2(k) \right) x(k). \end{aligned}$$

The corresponding input constraint set is

$$\mathbb{U} = \left\{ u \in \mathbb{R}^2 \mid -1 \le u_1 \le 1,\ -0.4 \le u_2 \le 0.4 \right\}.$$

and the state constraints and scheduling set are the same as in (7.14).

A desired tracking error bound of $\epsilon = 0.1$ is specified. The corresponding set $\mathcal{R}(\epsilon) \subset \mathbb{R}^{n_y}$ was calculated according to the procedure in Section 7.3 and the result is shown in Figure 7.3.

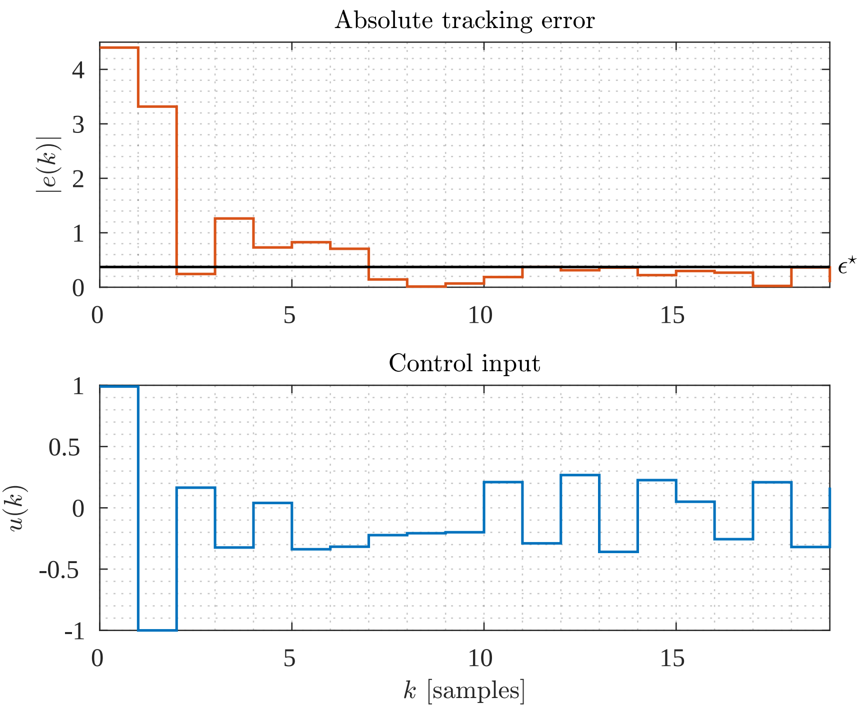


Figure 7.2: Simulation result for the example of Section 7.4.1.

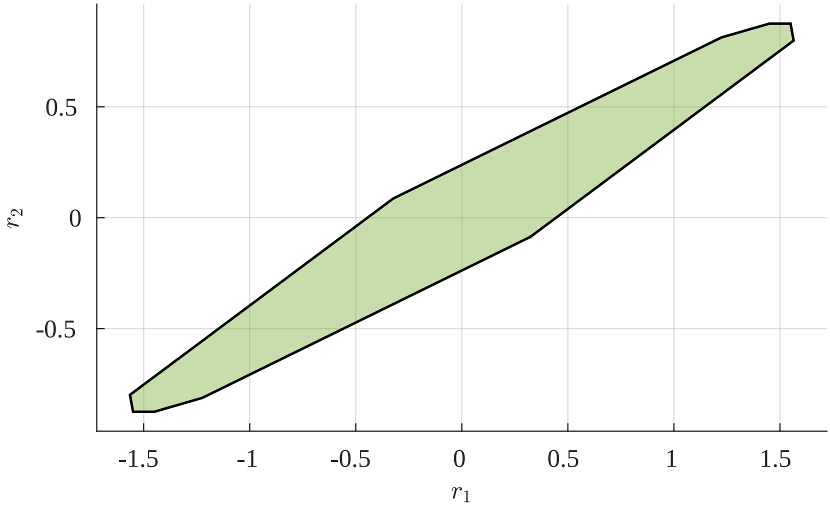


Figure 7.3: The set $\mathcal{R}(\epsilon)$ obtained in the example of Section 7.4.2.

Based on this analysis, the constant reference $r = \begin{bmatrix}0.5 & 0.3\end{bmatrix}^\top$ was chosen, which is in the interior of $\mathcal{R}(\epsilon)$. A corresponding simulation result for the initial state $x(0) = \begin{bmatrix}2 & 2\end{bmatrix}^\top$ is displayed in Figure 7.4. The prediction horizon $N = 7$ and tuning settings $Q = I$ and $R = 10I$ were equal to the previous experiment. The scheduling signal consisted of random switching between the vertices of Θ until the 18th sample, after which $\theta(k)$ remained constant at its last value.

In Figure 7.4, it can be observed that the tracking error goes to zero: thus, in this special case the controller achieves offset-free tracking when the scheduling variable stops varying. However, there also exist references $r \in \mathcal{R}(\epsilon)$ for which this is not possible (see Remark 7.1). An example of such a case was already considered in the previous experiment of Section 7.4.1, where in Figure 7.1, it was shown that $\text{convh}\{\bar{\mathbb{X}}(r)\} \not\subseteq P(r, \epsilon)$.

7.5 Concluding remarks

A tube-based MPC algorithm to track constant references for LPV systems with a guaranteed error bound was presented. The results of this chapter are a possible first step towards more sophisticated reference-tracking LPV predictive controllers. Some ideas for further development are the following:

- It would be of interest to develop an approach that combines a guaranteed ϵ-error bound with the ability to always achieve asymptotically offset-free tracking if $\theta(k)$ stops varying.
- An issue with the current method is that the domain of attraction of the controller can

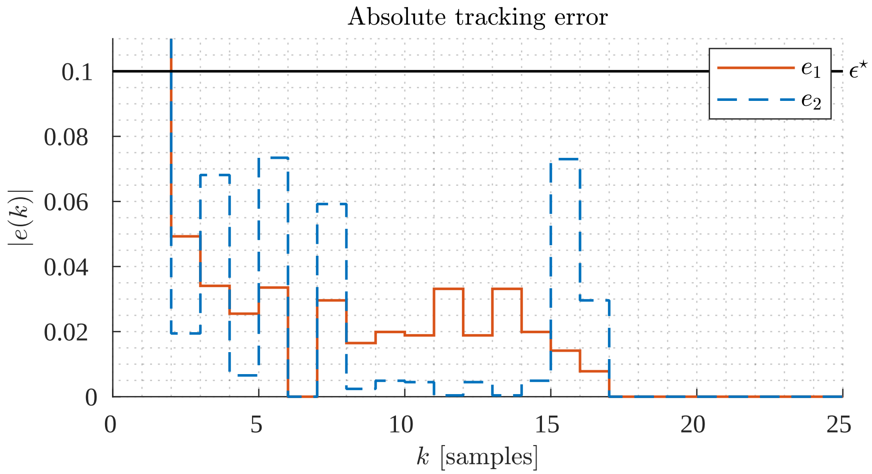


Figure 7.4: Tracking error of the MPC in the example of Section 7.4.2. In this special case, the tracking error goes to zero after θ stops varying.

be small if ϵ is chosen small. Hence, it would also be relevant to see if it is possible to construct an extended terminal set around $P(r, \epsilon)$ and use an appropriate terminal cost to ensure convergence, possibly without the time-varying terminal constraint.

- The more advanced tubes of, e.g., Chapters 4 and 5 could be considered for the construction of a reference-tracking LPV MPC algorithm as well.
- The approach could be extended to handle a larger class of references, e.g., piecewise-constant signals or those generated by *reference generators* as in (Di Cairano and Borrelli 2016).

Chapter 8

Non-linear MPC through LPV embeddings

This chapter addresses Research Question Q_5, and proposes a model predictive control (MPC) approach for non-linear systems based on linear parameter-varying (LPV) representations. The non-linear dynamics are assumed to be embedded inside an LPV representation. Hence, the non-linear MPC problem is replaced by an LPV MPC problem. When compared to general non-linear model predictive control (NMPC), the two main advantages of this approach are that it requires the on-line solution of a convex program, and that it allows for the systematic derivation of a terminal set and cost. Naturally, this may come at the sacrifice of achievable performance with respect to NMPC. The key idea behind the approach that enables a proof of recursive feasibility and stability, is to first restrict the the state evolution of the non-linear system to belong to a time-varying sequence of state constraint sets. Because in an LPV embedding, there exists a relationship between the scheduling and state variables, these state constraints are then used to construct a corresponding future scheduling tube. Compared to non-time-varying state constraints, tighter bounds on the future scheduling trajectories can be obtained. Two simple approaches to compute the initial sequence of state constraint sets are presented. Furthermore, the computation of a scheduling tube in this setting requires the application of a non-linear function to this sequence of constraint sets. Hence, outer approximations of this non-linear projection-based scheduling tube can be found, e.g., via methods based on interval analysis that were previously employed in reachability analysis of non-linear systems. Finally, the computational properties of the approach are demonstrated on numerical examples.

8.1 Introduction

In an LPV system, it is assumed that the scheduling variable is a signal that is independent from the states or control inputs. This assumed freedom enables the use of computationally efficient control design procedures. If an LPV model is used to embed an underlying non-linear system, the scheduling signal represents the non-linear behavior. Therefore, a controller designed on the basis of this linear model can be used to control the original non-linear system.

This idea is illustrated in Figure 8.1, in a way similar to the exposition in Section 2.3. The starting point in Figure 8.1.(a) is a non-linear system connected in closed-loop with a non-linear predictive controller. Next, scheduling variables $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ are introduced such

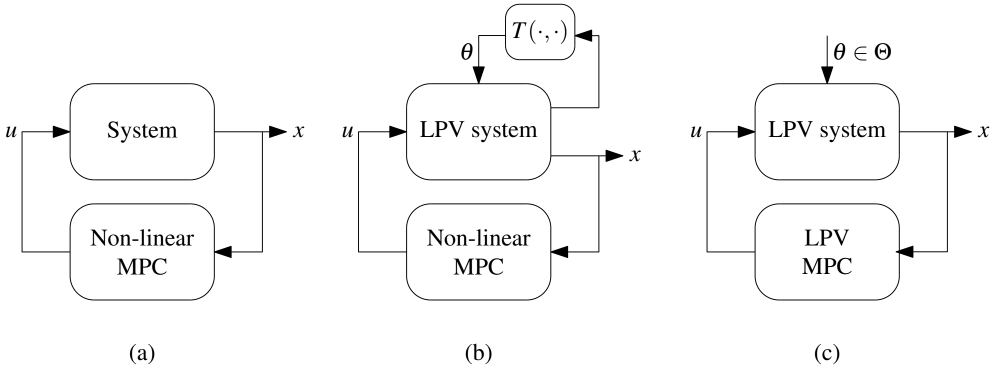


Figure 8.1: Principle of non-linear MPC via LPV embeddings. (a) The true non-linear system connected in closed-loop with a non-linear MPC. (b) The non-linear system is equivalently represented as an LPV system by describing the non-linearities in terms of a scheduling variable: the control problem is still non-linear. (c) The scheduling variable is "disconnected" to obtain an LPV embedding: the non-linear MPC is replaced by an LPV MPC.

that the non-linear system is equivalently represented in an LPV form. Typically, the introduced scheduling variables are related to the state and input signals, i.e., there exists a map $T(\cdot,\cdot)$ such that $T(x,u) = \theta$ (Figure 8.1.(b)). Because this is just an equivalent representation of the same system, the MPC problem is still non-linear. Then, to construct the embedding, the relation $T(\cdot,\cdot)$ is discarded and only an inclusion $\theta(k) \in \Theta$ is assumed (Figure 2.1.(c)).

The NMPC—which typically requires non-convex optimization problems to be solved on-line—can therefore be replaced by an LPV MPC. Although the optimization problems in LPV MPC often have more decision variables and constraints, they can still be solved efficiently due to their convex structure. Furthermore, systematic approaches for the computation of the necessary terminal set and cost to guarantee recursive feasibility and closed-loop stability are available in the LPV setting. However, because the scheduling variable is considered to be a free signal that can take any value inside of the set Θ independently of the other signals in the system, some conservatism is introduced. A few MPC approaches for LPV embeddings from the literature are reviewed next.

In the MPC scheme of (Lu and Arkun 2002), which is based on the quasi min-max scheme (Lu and Arkun 2000b), it is assumed that the underlying non-linear system can be represented by an LPV model obtained as a family of linearized models around various operating points. It is shown that if the optimization remains feasible, the closed loop is asymptotically stable. However, because the family of linearized models is not guaranteed to embed the true non-linear dynamics along closed-loop trajectories, recursive feasibility can not be established a-priori. In the method of (Chisci et al. 2003), the non-linear dynamics are embedded in an LPV representation with a discrete scheduling set, effectively representing a collection of uncertain linear models. Rate-of-variation (ROV) bounds on the state—and hence, on the scheduling variable—are imposed in a way reminiscent of classical gain scheduling (Rugh and Shamma

2000).

Based on the pure LPV MPC algorithm of (Casavola et al. 2002), the work (Casavola et al. 2003) presents an MPC algorithm applicable to LPV embeddings. A known and bounded ROV on the scheduling variable is assumed. The LPV MPC algorithm used in (Casavola et al. 2003) contains a mistake that can cause input constraint violation, as pointed out in (Ding and Huang 2007). Even when this mistake is corrected, there is no guarantee that the assumed ROV on the scheduling variable will hold along realized closed-loop trajectories. Recursive feasibility and stability can therefore only be established under the assumption that the controller will not excite the system in a way that violates the assumed ROV bounds.

In (Cisneros et al. 2016; Cisneros and Werner 2017), an "iterative" MPC scheme is presented to control non-linear systems embedded in an LPV representation. Unlike the previously discussed approaches, knowledge of the scheduling map is exploited in the prediction stage. Based on an initial guess of the future scheduling trajectory, a simple linear time-varying (LTV) MPC problem is solved. The resulting optimal state and input trajectories are used to generated a new future predicted scheduling trajectory through the scheduling map, and the procedure is iterated until the predicted trajectories converge. This is computationally highly efficient because at each time instant, only the solution of a sequence of LTV MPC problems is required. The authors demonstrated that it works well on some examples, but there are no guarantees concerning convergence of the iterations: hence, closed-loop stability can not be guaranteed.

In this chapter, a non-linear MPC based on LPV embeddings is developed, which integrates the explicit use of a scheduling map from (Cisneros et al. 2016; Cisneros and Werner 2017) into a tube-based LPV MPC formulation. In this way, achievable performance with respect to NMPC is traded off for computational efficiency, without sacrificing guarantees on recursive feasibility and stability. In the proposed approach, the state constraints are allowed to be time-varying along the prediction horizon. These constraints are computed for a measured initial state as part of the "initialization" of the controller. The key feature that enables an a-priori guarantee of recursive feasibility and stability, is the use of the scheduling map to construct the scheduling sets that are valid along all possible closed-loop trajectories that are allowed by the designed state constraints. Such a guarantee is missing in, e.g., the approaches of (Casavola et al. 2003; Lu and Arkun 2002).

Previously, tube-based methods have been used to derive efficient non-linear model predictive controllers based on linearization. In (Cannon et al. 2011; Y. I. Lee et al. 2002) the non-linear dynamics are linearized around a given feasible "seed" trajectory, and the arising linearization errors are represented as additive perturbations. Although both the proposed method of this chapter and (Cannon et al. 2011; Y. I. Lee et al. 2002) belong to the class of tube-based policies, the linearized system representation adopted in (Cannon et al. 2011; Y. I. Lee et al. 2002) is different from the LPV embedding considered in the current chapter.

The remainder of this chapter is structured as follows. Section 8.1.1 specifies the problem setting considered in this chapter. The tube-based MPC algorithm applicable to LPV embeddings is developed in Section 8.2. Several implementable methods for initializing the controller and for computing the scheduling tubes, are given in Section 8.3 and Section 8.4 respectively. The control performance of the method is demonstrated on some numerical examples in Section 8.5. Finally, a few concluding remarks are provided in Section 8.6.

8.1.1 Problem setting

This chapter considers constrained non-linear systems represented as

$$x(k+1) = f\left(x(k), u(k)\right), \; k \in \mathbb{N}, \; x(0) = x_0, \tag{8.1}$$

that can be embedded in a constrained polytopic LPV state-space (LPV-PSS) representation of the form

$$x(k+1) = A\left(\theta(k)\right) x(k) + Bu(k), \tag{8.2a}$$

$$\theta(k) = T\left(x(k)\right). \tag{8.2b}$$

The following assumptions on the system (8.1) are made.

Assumption 8.1. *The system* (8.1) *satisfies the following properties:*

(i) The function $f : \mathbb{X} \times \mathbb{U} \to \mathbb{R}^{n_x}$ *is continuously differentiable, and it is bounded on* $\mathbb{X} \times \mathbb{U}$.

(ii) The origin is an equilibrium of (8.1)*, i.e.,* $f(0,0) = 0$.

An overview of the concept of LPV embedding was provided in Section 2.3. In (8.2), the function $T : \mathbb{X} \to \Theta$ is called the *scheduling map*. This function describes the relationship between the state and scheduling variables. In the definition of an LPV embedding, the scheduling map is important because it is connected to the validity of the embedding on $\mathbb{X} \times \mathbb{U}$. In contrast, when controlling an LPV embedding, $T(\cdot)$ typically appears only in a limited and implicit way, because the scheduling variable is considered to be a free signal. Knowledge of $T(\cdot)$ can however be exploited to obtain refined sets of all possible scheduling trajectories based on the possible future state evolution; this is what will be done in the MPC developed in this chapter.

The class of polytopic LPV embeddings described by (8.2) is somewhat restricted, because the map $T(\cdot)$ does not depend on u and because the input matrix B does not depend on θ. Some remarks on the extension of the results of this chapter to more general embeddings are provided later in Section 8.6.1. A class of systems that can always be embedded in the required form of (8.2) are, for instance, Lur'e systems of the form

$$x(k+1) = Ax + \phi\left(x\right) + Bu$$

where $\phi(\cdot)$ is Lipschitz continuous and bounded on a compact set $\mathbb{X} \subset \mathbb{R}^{n_x}$. Sufficient conditions under which an embedding of the form (8.2) can be constructed for an arbitrary continuous-time non-linear system can be found in (Abbas et al. 2014, 2017). The discrete-time case is still a topic of ongoing research. Some further remarks on the possible extension of the methods presented in this chapter to more general classes of embeddings, are provided in Section 8.6.1.

The control problem considered in this chapter is equal to Problem 3.1, i.e., it is desired to design a controller $K_{\text{mpc}}(\cdot,\cdot,\cdot)$ that renders the origin of the representation (8.2) regionally asymptotically stable. From the basic properties of embeddings (see again Section 2.3), it follows directly that if the developed MPC stabilizes the LPV representation (8.2), then it stabilizes the original non-linear system (8.1) as well. This work can be seen as a necessary first step towards the development of an MPC approach for LPV embeddings that can stabilize other equilibrium points besides the origin.

8.2 The non-linear TMPC algorithm

In this section, the tube-based MPC algorithm for polytopic LPV embeddings is presented. First, some necessary modifications to the basic tube synthesis approach are given in Section 8.2.1. Then, in Section 8.2.2, an alternative class of stage cost function is presented in addition to the class that was introduced in Section 3.4.5. Necessary conditions on the terminal set are also given to enable the proof of recursive feasibility and stability. Finally, the complete algorithm together with a proof of its main properties is provided in Section 8.2.3.

8.2.1 Tubes for LPV embeddings

The type of tubes employed in this chapter are modified slightly with respect to those of Definition 3.4, which were used in the previous chapters:

Definition 8.1. *Let* $\mathbf{\Theta} \subseteq \Theta^N$ *be given according to Definition 3.1. A* tube for embeddings *of length N is a pair*

$$\mathbf{T} = (\mathbf{X}, \mathbf{K}) = \left(\{X_i\}_{i=0}^{N}, \{K_i\}_{i=0}^{N-1} \right)$$

where $X_i \subseteq \mathbb{R}^{n_x}$ *are sets and where* $K_i : X_i \times \Theta_i \rightarrow \mathbb{U}$ *are* $\mathcal{CH}_1$ *control laws such that for all* $i \in [0..N-1]$, *the conditions* $\mathcal{I}(X_i, \Theta_i | K_i) \subseteq X_{i+1}$ *and* $X_i \subseteq \mathbb{X}_i$ *hold. Each set* X_i *is called a cross section.*

With respect to Definition 3.4, the following two differences exist:

- The state constraints are allowed to be time-dependent sequences $\vec{\mathbb{X}} = \{\mathbb{X}_0, \ldots, \mathbb{X}_{N-1}\}$.
- The state constraint inclusion condition $X_i \subseteq \mathbb{X}_i$ is more stringent than the condition $\mathcal{I}(X_i, \Theta_i | K_i) \subseteq \mathbb{X}$ which was used previously. In the setting of this chapter, as will become clear later, this stricter condition is necessary to guarantee consistency between the transition constraints $\mathcal{I}(X_i, \Theta_i | K_i) \subseteq X_{i+1}$ and the system dynamics (8.2).

Note that with these modifications, all the conditions appearing in Definition 8.1 can still be expressed in terms of linear inequalities along the line of Section 4.4.2. The role of the scheduling tube $\mathbf{\Theta} \subseteq \Theta^N$ is to bound all possible behaviors of θ in the embedding (8.2). To this end, the sequence of time-varying state constraints allowed by Definition 8.1

$$\vec{\mathbb{X}} = \{\mathbb{X}_0, \ldots, \mathbb{X}_{N-1}\} \subseteq \mathbb{X}^N \tag{8.3}$$

is constructed at each time instant k, and an associated scheduling sequence $\mathbf{\Theta}$ is generated by applying the scheduling map $T(\cdot)$. So, the signal θ is still free, but all its possible future trajectories are known to belong to the scheduling tube $\mathbf{\Theta}$ in a way that is consistent with all possible future state trajectories allowed by the constraints (8.3). This setup, that allows constraint sequences consisting of arbitrary sets, is more general than the bounds on the state ROV assumed in (Chisci et al. 2003).

For simplicity, this chapter considers homothetic tubes similar to those of Definition 3.9:

Definition 8.2. *A tube for embeddings* **T** *according to Definition 8.1 is called a* homothetic tube for embeddings *if, in addition, it satisfies Definition 3.9.*

The only difference between Definition 8.2 and Definition 3.9, is the fact that Definition 8.1 is with respect to tubes of the type of Definition 8.1 instead of Definition 3.4. This does not cause any contradiction, because a tube that satisfies Definition 8.1 also satisfies Definition 3.4.

To obtain a scheduling tube consistent with the state constraints (8.3), the map $T(\cdot)$ has to be applied to set-valued arguments $X \subseteq \mathbb{X}$ according to

$$T(X) := \{T(x) \mid x \in X\}\,. \tag{8.4}$$

Because $T(\cdot)$ is a non-linear function in general, Equation 8.4 is not straightforward to compute directly. If X is a polytope then there is no guarantee that $T(X)$ is also a polytope, or even convex. Therefore, in the sequel, it is assumed that there exists a so-called convex outer-approximator $\hat{T}(\cdot)$ of $T(\cdot)$.

Definition 8.3. *The function $\hat{T} : 2^{\mathbb{X}} \to 2^{\Theta}$ is a convex outer-approximator of $T : 2^{\mathbb{X}} \to 2^{\Theta}$ if*

(i) For a singleton set $X = \{x\}$, $\hat{T}(X) = \hat{T}(\{x\}) = T(x)$.

(ii) For all convex and compact sets $X \subseteq \mathbb{X}$, $\hat{T}(X)$ is a convex set and $T(X) \subseteq \hat{T}(X)$.

(iii) For all convex and compact sets $X_0 \subseteq X_1 \subseteq \mathbb{X}$, $\hat{T}(X_0) \subseteq \hat{T}(X_1)$.

(iv) If $X \subseteq \mathbb{X}$ is a polytope, then $\hat{T}(X)$ is a polytope as well.

In this section, the existence of a convex outer-approximator is simply assumed. Possible constructions of such an approximator are presented later in Section 8.4.

In the MPC approach for LPV embeddings, the set of feasible tubes depends on the time-varying state constraints as

$$\mathcal{T}_N\left(x\middle|\vec{\mathbb{X}}\right) = \Big\{\mathbf{T}\,\Big|\,\mathbf{T} \text{ satisfies Def. 8.2 with } X_0 = \{x\}, X_N \subseteq X_{\mathrm{f}}, \\ \text{and } \forall i \in [0..N-1] : \Theta_i = \hat{T}\left(\mathbb{X}_i\right)\Big\}. \tag{8.5}$$

Hence, the associated tube synthesis problem is

$$V\left(x\middle|\vec{\mathbb{X}}\right) = \min_{\mathbf{T}} J_N\left(\mathbf{T}, \vec{\mathbb{X}}\right) \text{ subject to } \mathbf{T} \in \mathcal{T}_N\left(x\middle|\vec{\mathbb{X}}\right). \tag{8.6}$$

In contrast to the setup described in Section 3.4.1, (8.5)–(8.6) are not dependent on $\boldsymbol{\Theta}$, but on $\vec{\mathbb{X}}$. This is because in the proposed approach, the scheduling sets in the sequence $\boldsymbol{\Theta} \subseteq \Theta^N$ are generated by applying the outer-approximator $\hat{T}(\cdot)$ to the sequence $\vec{\mathbb{X}} \subseteq \mathbb{X}^N$.

This construction is also the reason why the state constraints in Definition 8.1 have to be of the form $X_i \subseteq \mathbb{X}_i$ instead of the more relaxed form adopted previously in Definition 3.4. The tube transition constraints are of the form $\mathcal{G}(X_i, \Theta_i | K_i) \subseteq X_{i+1}$ where, according to (8.5), $\Theta_i = \hat{T}(\mathbb{X}_i)$. If X_i would not be a subset of $\mathbb{X}_i$, there could exist $x \in X_i$ such that $T(x) \notin \Theta_i$, and therefore the transition constraint $\mathcal{G}(X_i, \Theta_i | K_i) \subseteq X_{i+1}$ would not be consistent with the dynamics (8.2). Therefore the condition $X_i \subseteq \mathbb{X}_i$ is necessary.

In the this chapter, the cost function takes the standard form of Section 3.4.1, i.e.,

$$J_N\left(\mathbf{T}, \vec{\mathbb{X}}\right) = \sum_{i=0}^{N} \ell\left(X_i, K_i, \hat{T}\left(\mathbb{X}_i\right)\right) + F\left(X_N\right) \tag{8.7}$$

with $\ell(\cdot,\cdot,\cdot)$ being the stage cost and $F(\cdot)$ being the terminal cost. The type of stage cost employed in this chapter can be that of Section 3.4.5, but an alternative choice of cost will also be presented in Section 8.2.2. Furthermore the conditions on the terminal set in (8.5) need to be adapted to take into account the relationship $\theta = T(x)$ between state and scheduling variables. These elements will be presented in the next subsection.

8.2.2 Center-based cost function and terminal set

In the MPC algorithms presented previously, the worst-case stage cost function introduced in Section 3.4.5 was used. This cost can be used in the setting of this chapter without modification. However, it can also be of interest to optimize an average- or expected cost instead of the worst-case one. For this purpose, this section introduces an alternative, center-based, stage cost that can be used if desired.

It was found that the center-based cost is especially useful when controlling a non-linear system represented by an LPV embedding, because the worst-case cost of Section 3.4.5 can sometimes lead to non-smooth closed-loop trajectories. For this reason, it is introduced here in this chapter. It should, however, be noted that this new center-based cost can also be employed in the controllers from the previous chapters with suitable modifications made to the stability proofs.

Define $r(2) := 2$ and $r(\infty) := 1$. The alternative center-based stage cost function, which is applicable to sets of the form $X = z \oplus \alpha S$ where $S = \mathrm{convh}\{\bar{s}^1, \ldots, \bar{s}^{q_s}\}$, is then designed as

$$\ell_p\left(X, K, \Theta\right) = \|Qz\|_p^{r(p)} + P|\alpha|^{r(p)} + \frac{1}{q_\theta q_s} \sum_{i=1}^{q_s} \sum_{j=1}^{q_\theta} \|RK\left(\bar{x}^i, \bar{\theta}^j\right)\|_p^{r(p)} \tag{8.8}$$

where $p \in \{2, \infty\}$, and where $Q \in \mathbb{R}^{n_q \times n_x}$, $R \in \mathbb{R}^{n_r \times n_u}$ and the scalar $P > 0$ are tuning parameters. The matrices (Q, R) must be of full column rank. In (8.8), the vectors $\{\bar{\theta}^1, \ldots, \bar{\theta}^{q_\theta}\}$ represent the vertices of the set Θ that is being passed as the third argument of $\ell(\cdot,\cdot,\cdot)$.

Because the center-based cost (8.8) is only defined for sets represented as $X = z \oplus \alpha S$, it is less general than the worst-case cost of Section 3.4.5 used previously, which could be applied to arbitrary sets. Like the worst-case stage cost, the cost (8.8) has some useful properties:

Proposition 8.1. *In what follows, let $K : X \times \Theta \to \mathbb{U}$ be a $\mathcal{CH}_1$-controller. The cost function* (8.8) *satisfies the following properties:*

(i) For $p \in \{2, \infty\}$, there exists a $\mathcal{K}_\infty$-function $\underline{\ell}$ such that $\underline{\ell}\left(d_H^0\left(X\right)\right) \leq \ell_p(X, K, \Theta)$

(ii) For $p \in \{2, \infty\}$, the stage cost is homogeneous of degree $r(p)$ in the sense that for all $\alpha \in \mathbb{R}_+$, $\ell(\alpha X, K, \Theta) = \alpha^{r(p)} \ell(X, K, \Theta)$.

Proof of (i). Trivially, $\|Qz\|_p^{r(p)} + P|\alpha|^{r(p)} \leq \ell\big(X, K\big)$. As Q is assumed to be full column rank, $x \mapsto \|Qx\|$ is a norm (in particular, $\|Qx\| = 0$ if and only if $x = 0$). Therefore, for $p \in \{2, \infty\}$, $\exists c_p > 0$ such that $\forall z \in \mathbb{R}^{n_x}: \; c_p\|z\|_p \leq \|Qz\|_p$. Let $X = z \oplus \alpha S \in \Omega$ where S is a polytope as stated in Definition 8.2. Define $M := \max_{i \in \mathbb{N}_{[1,q_S]}} \|\bar{s}^i\|_p$. By the definition of the Hausdorff distance in the relevant norm,

$$c_p d_H^0(X) = c_p \max_{i \in \mathbb{N}_{[1,q_S]}} \|z + \alpha \bar{s}^i\|_p \leq c_p\|z\|_p + c_p M|\alpha|,$$

so that if $c_p M \leq P$, $c_p d_H^0(X) \leq \|Qz\|_p + P|\alpha|$ is satisfied. Because the set S can always be scaled by an arbitrary factor $\beta > 0$, the property $c_p M \leq P$ can be assumed without loss of generality. This proves the existence of a class-$\mathcal{K}_\infty$ lower bound $\underline{\ell}(d_H(X))$ for $\ell\big(X, K\big)$ if $p = \infty$. If $p = 2$,

$$\frac{1}{2}\left(c_p d_H^0(X)\right)^2 = \frac{1}{2}c_p^2 \max_{i \in \mathbb{N}_{[1,q_S]}} \|z + \alpha \bar{s}^i\|_p^2 \leq \frac{1}{2}c_p^2\left(\|z\|_p + M|\alpha|\right)^2 \leq c_p^2\left(\|z\|_p^2 + M^2|\alpha|^2\right)$$

where again it can be assumed without loss of generality that the set S is scaled such that $c_p^2 M^2 \leq P$, proving the existence of a lower bound $\underline{\ell}(d_H(X))$ also for $p = 2$.
Proof of (ii). Let $X \subset \mathbb{R}^{n_x}$ be an arbitrary set represented as $X = z \oplus \alpha S$ with $S \subset \mathbb{R}^{n_x}$ being convex and compact. Then, $\beta X = \beta z \oplus \beta \alpha S$, and therefore Property (ii) follows from the $\mathcal{CH}_1$-property of the controller $K(\cdot, \cdot)$ combined with homogeneity of the vector norm $\|\cdot\|$. □

It is important to note that the cost (8.8) does not satisfy the subset inclusion property of Proposition 3.1.(i). This has a consequence for the proof of closed-loop stability. In particular, if the cost (8.8) is used, it is necessary to relax the initial condition constraint in the set of feasible tubes (8.5) from $X_0 = \{x\}$ to $x \in X_0$. That is, it is necessary to replace (8.5) with

$$\mathcal{T}_N\left(x \middle| \vec{\mathbb{X}}\right) = \left\{\mathbf{T} \,\middle|\, \mathbf{T} \text{ satisfies Def. 8.2 with } x \in X_0, X_N \subseteq X_\mathrm{f}, \text{and } \Theta_i = \hat{T}\left(\mathbb{X}_i\right)\right\}. \tag{8.9}$$

The full proof of stability, where this point is taken into account, will be presented later for Theorem 8.1.

As is usual, to guarantee recursive feasibility of (8.6), a terminal set constraint $X_N \subseteq X_\mathrm{f}$ is included in (8.5). The set X_f must be designed to be controlled λ-contractive for (8.2) in a sense similar to Definition 2.13. However, because in the current setting there exists a relationship between the state and scheduling variables that definition has to be adapted slightly:

Definition 8.4. *Let X and $\breve{\mathbb{X}}$ both be PC-sets such that $X \subseteq \breve{\mathbb{X}} \subseteq \mathbb{X}$. Then, the set X is called controlled λ-contractive for an LPV-PSS embedding* (8.2)*, if there exists a local $\mathcal{CH}_1$ controller $K : X \times \hat{T}\left(\breve{\mathbb{X}}\right) \to \mathbb{U}$ such that $\lambda = \inf\{\mu \,|\, \mathcal{G}(X, \hat{T}\left(\breve{\mathbb{X}}\right)|K) \subseteq \mu X\} < 1$.*

In light of the above definition, the procedure for computing the terminal set X_f consists of first choosing a constraint set $\breve{\mathbb{X}}$, after which X_f can be computed using any of the standard approaches (e.g., Algorithm 2.1 or the method from Appendix A) available for LPV systems. If no X_f can be found or if its volume is smaller than desired, the constraint set $\breve{\mathbb{X}}$ can be changed and the procedure repeated. Observe that X_f is always bounded as $X_\mathrm{f} \subseteq \tilde{\mathbb{X}}$, but that a large $\tilde{\mathbb{X}}$ also results in a large scheduling set $\hat{T}\left(\breve{\mathbb{X}}\right)$.

Suppose that X_f is λ-contractive according to Definition 8.4. Then a suitable terminal cost $F(\cdot)$ to be used in (8.7) can be computed with the same basic structure as the terminal costs used in Chapters 4 and 5. Define

$$F(X) = \frac{\Psi_{X_\mathrm{f}}(X)}{1-\lambda} \max_{\tilde{X} \subseteq X_\mathrm{f}, \theta \in T(\breve{\mathbb{X}})} \ell_p\left(\tilde{X}, K_\mathrm{f}\right) = \frac{\bar{\ell}}{1-\lambda} \Psi_{X_\mathrm{f}}(X), \tag{8.10}$$

where the maximization is done with respect to sets $X = z \oplus \alpha S$. The function $\Psi_{X_\mathrm{f}}(\cdot)$ is the set-gauge of X_f, as specified in Definition 2.2, and $K_\mathrm{f} : X_\mathrm{f} \times T(\breve{\mathbb{X}}) \to \mathbb{U}$ is a local set-induced controller that renders X_f λ-contractive. Because $\ell_p(\cdot,\cdot,\cdot)$ is convex, (8.10) reduces to a finite-dimensional optimization problem provided that $T(\breve{\mathbb{X}})$ is a polytope, as then $\ell_p(\cdot,\cdot,\cdot)$ takes its maximum on one of the vertices of $X_\mathrm{f} \times T(\breve{\mathbb{X}})$. For the stage cost of Section 3.4.5, the terminal cost (8.10) is exactly the same as the cost from Chapter 5. For the center-based stage cost of the present section, however, the numerical value can be different.

It is important to note that the stage cost (8.8) and the terminal cost (8.10) satisfy the necessary conditions that enable the construction of $\mathcal{K}_\infty$-bounds on the value function according to Proposition 3.2.

8.2.3 Main result

Algorithm 8.1 The MPC algorithm for LPV embeddings.

Require: $N \in [1..\infty)$
Require: A given sequence $\overrightarrow{\mathbb{X}}(0)$

$k \leftarrow 0$
loop
  if $\mathcal{T}_N\left(x(k) \middle| \overrightarrow{\mathbb{X}}_k\right) \neq \emptyset$ **then**
    Solve (8.6) to obtain $\mathbf{T}^\star \in \mathcal{T}_N\left(x(k) \middle| \overrightarrow{\mathbb{X}}(k)\right)$
    Apply $u(k) = K_0^\star(x(k), \theta(k)) = u_0^\star$ to the system (8.1)
    $\forall i \in \mathbb{N}_{[0,N-1]} : \mathbb{X}_i(k+1) \leftarrow \begin{cases} \mathbb{X}_{i+1}(k), & i < N-1, \\ \breve{\mathbb{X}}, & i = N-1. \end{cases}$
    $k \leftarrow k+1$
  else
    abort
  end if
end loop

The proposed MPC approach is summarized in Algorithm 8.1, and Theorem 8.1 gives the main result concerning its properties:

Theorem 8.1. *Suppose that $\hat{T}(\cdot)$ is a convex outer-approximator of the scheduling map $T(\cdot)$ according to Definition 8.3. Let X_f satisfy Definition 8.4, let $F(\cdot)$ be as in* (8.10)*, and in Definition 8.2 set $S = X_\mathrm{f}$. Furthermore let $\overrightarrow{\mathbb{X}} = \{\mathbb{X}_0, \ldots, \mathbb{X}_{N-1}\}$ and let $\overrightarrow{\mathbb{X}}^+ \subseteq \mathbb{X}^N$ be related* *to* $\vec{\mathbb{X}}$ *as* $\vec{\mathbb{X}}^+ = \left\{\mathbb{X}_1, \ldots, \mathbb{X}_{N-1}, \breve{\mathbb{X}}\right\}$ *according to Step 6 of Algorithm 8.1. Then, the algorithm satisfies the following properties:*

(i) It is recursively feasible, i.e., if $\exists \mathbf{T} \in \mathcal{T}_N\left(x \middle| \vec{\mathbb{X}}\right)$*, then* $\exists \mathbf{T}^+ \in \mathcal{T}_N\left(x^+ \middle| \vec{\mathbb{X}}^+\right)$ *where* $x^+ = A(\theta)x + BK_0(x,\theta) = A(T(x))x + BK_0(x,T(x))$*.*

(ii) The state of the controlled system reaches $\breve{\mathbb{X}}$ *in N steps or less, i.e.,* $\exists k_\star \in \mathbb{N}_{[0,N]}$ *such that* $\forall k \geq k_\star : \ x(k) \in \breve{\mathbb{X}}$*.*

(iii) The origin of (8.1) *is asymptotically stabilized in the sense of Definition 2.8.*

(iv) The tube synthesis problem (8.6) *is a quadratic program (QP) if* $p = 2$ *and a linear program (LP) if* $p = \infty$*. The numbers of variables and constraints are linear functions of the prediction horizon* N*.*

Proof of (i). The particular construction of $\vec{\mathbb{X}}^+$ in Step 6 of Algorithm 8.1 means that the relationship $\Theta^+ \sqsubseteq \Theta$ holds, where

$$\Theta = \left\{\hat{T}\left(\mathbb{X}_1\right), \ldots, \hat{T}\left(\mathbb{X}_{N-1}\right)\right\}, \ \Theta^+ = \left\{\hat{T}\left(\mathbb{X}_1^+\right), \ldots, \hat{T}\left(\mathbb{X}_{N-1}^+\right)\right\}.$$

Therefore, Property (i) is proven in the same way as was done in Proposition 4.2. If the tube $\mathbf{T}$ satisfies the state constraints of Definition 8.1, then the feasible candidate tube $\mathbf{T}^+$ constructed therein also satisfies the state constraints $\vec{\mathbb{X}}^+$.

If the center-based stage cost (8.8) is used, the set of feasible tubes has a different initial condition constraint, and it is given by (8.9). The tube $\mathbf{T}^+$ used to establish recursive feasibility consequently has to be changed into

$$\mathbf{T}^+ = \left(\{X_1, X_2, \ldots, X_{N-1}, \gamma X_\mathrm{f}, \gamma\lambda X_\mathrm{f}\}, \ \{K_1, \ldots, K_{N-1}, K_\mathrm{f}\}\right). \tag{8.11}$$

No other modifications are necessary.

Proof of (ii). Given Property (i), the satisfaction of Property (ii) follows by construction of the constraint set sequences $\vec{\mathbb{X}}_k$ in Step 6 of Algorithm 8.1.

Proof of (iii). The first step is to show closed-loop asymptotic local stability of the origin of the LPV embedding (8.2). Whenever the stage cost of Section 3.4.5 is used, this proof is equal to the proofs presented in Theorem 4.1 (with $M = 1$) and Theorem 5.1 (with a fully homothetic parameterization structure). If use is made of the center-based stage cost (8.8), the only necessary modification stems from the fact that (8.8) does not satisfy a subset inclusion property like Proposition 3.1.(i). Therefore, the feasible, but not necessarily optimal tube $\mathbf{T}^+$ used in the proof must be changed into (8.11).

Once closed-loop stability of the origin (8.2) is established, stability of the origin of the non-linear system (8.1) follows by the fundamental properties of embeddings as outlined in Section 2.3.

Proof of (iv). A construction similar to that of Section 4.4.2 applies. The necessary modifications to accommodate the more stringent constraint inclusion condition of Definition 8.1 and—if applicable—the alternative cost function (8.8) are straightforward. The modifications do not affect the $O(N)$-scaling of the number of variables and constraints. □

8.3 Initializing the state constraints

This section presents two possible methods for determining the initial state constraint sequence $\overrightarrow{\mathbb{X}}$. The first method of Section 8.3.1 is based on restricting the ROV of the state variable, whereas the second method of Section 8.3.2 constructs the sets around a given initially feasible "seed" trajectory.

Note that although this section presents two particular constructions, the feasibility and stability properties established in the previous section are valid for any arbitrary construction of the sequence $\overrightarrow{\mathbb{X}}$.

8.3.1 Bounded rate of variation

Let $\delta \in \mathbb{R}_+^{n_\mathrm{x}}$ denote a bound on the ROV of the state variable and define

$$\Delta(\delta) = \left\{ x \in \mathbb{R}^{n_\mathrm{x}} \,\middle|\, \forall i \in \mathbb{N}_{[1,n_\mathrm{x}]} : \ |x_i| \le \delta_i \right\}. \tag{8.12}$$

Then, the initial constraint sequence $\overrightarrow{\mathbb{X}}$ can be computed as

$$\forall i \in \mathbb{N}_{[0,N]} : \mathbb{X}_i = \begin{cases} \{x_0\}, & i = 0, \\ (\mathbb{X}_{i-1} \oplus \Delta(\delta)) \cap \mathbb{X}, & i > 0. \end{cases} \tag{8.13}$$

The bounded ROV-construction is similar to what was used in (Chisci et al. 2003). A relevant question is how to choose the value δ. A large δ allows a lot of freedom in the state variation, but also leads to large corresponding scheduling sets, which can lead to infeasibility as the uncertainty in the future evolution of θ becomes too great. On the other hand, a small δ may give state constraints that are too restrictive, also causing infeasibility. Therefore, a trade-off between these two undesirable extremes has to be found.

Note that if there exists a partitioning of the state vector $x = (x_1, x_2)$ such that $T(x) = T(x_1)$, then it is sufficient to impose a bounded ROV on the x_1-component only.

8.3.2 Initial feasible trajectory

Suppose that a feasible initial state and input trajectory

$$\tilde{\mathbf{T}}_0 = (\{\tilde{x}_0, \tilde{x}_1, \ldots, \tilde{x}_N\}, \{\tilde{u}_0, \ldots, \tilde{u}_{N-1}\})$$

is known where $\tilde{x}_N \in X_\mathrm{f}$ and where for all $i \in \mathbb{N}_{[0,N-1]}$: $\tilde{x}_i \in \mathbb{X}$, $\tilde{u}_i \in \mathbb{U}$, and $\tilde{x}_{i+1} = A\big(T(\tilde{x}_i)\big)\tilde{x}_i + B\tilde{u}_i$. Then, by defining some tolerance $\delta \in \mathbb{R}_+^{n_\mathrm{x}}$, the initial state constraint sets can be computed according to the relationship

$$\forall i \in \mathbb{N}_{[0,N]} : \ \mathbb{X}_i = \begin{cases} \{x_0\}, & i = 0, \\ (\tilde{x}_i \oplus \Delta(\delta)) \cap \mathbb{X}, & i > 0, \end{cases} \tag{8.14}$$

with $\Delta(\cdot)$ as in (8.12). In comparison to the bounded ROV initialization discussed earlier, this approach is generally more likely to yield a feasible solution to (8.6), but the assumed availability of the initially feasible trajectory $\tilde{\mathbf{T}}_0$ can be considered a disadvantage (note that linearization-based approaches (Cannon et al. 2011; Y. I. Lee et al. 2002) also presume the knowledge of such a trajectory). If $\overrightarrow{\mathbb{X}}(0)$ is constructed according to (8.14) with $\delta = 0$, then (8.6) is always feasible at $k = 0$. Due to the constraint update Step 7 of Algorithm 8.1, a value of $\delta = 0$ however means that the controller can never deviate from the initial trajectory, thus blocking the feedback action of MPC. On the other hand, a value δ that is too large can lead to infeasibility as the corresponding scheduling sets become too large. Similarly to the situation with the ROV bound from the previous subsection, a trade-off for the value of δ has to be found.

The trajectory $\tilde{\mathbf{T}}_0$ can be obtained by solving a non-linear MPC problem. This could be computationally expensive, but this problem only needs to be solved once at the first sample. An efficient approach that can be used to generate an initial trajectory is (Cisneros et al. 2016). The method (Cisneros et al. 2016) is not guaranteed to converge to a solution, but if it converges, the result can be used to initialize the MPC proposed in this chapter which subsequently guarantees recursive feasibility and stability.

8.4 Bounding the scheduling tube

In this section, several ways of constructing outer-approximations $\hat{T}(X)$ are discussed. The easy special case of an affine scheduling map is presented first in Section 8.4.1. For general non-linear scheduling maps, some methods to construct the approximations are discussed in Section 8.4.2. These methods include the straightforward computation of bounds using global optimization or gridding, and a more advanced approach based on Taylor expansion and interval analysis. This latter approach was used previously—in a slightly different form—for the purpose of reachability analysis of non-linear systems.

Throughout this section, $X \subseteq \mathbb{X}$ is assumed to be a convex and compact set. This is in agreement with the state constraint sets that are being used in the MPC algorithm of Section 8.2.

8.4.1 Affine scheduling map

Suppose that $T : \mathbb{X} \to \Theta$ is affine, i.e., of the form

$$T(x) = \mathcal{T} x + b,$$

where $\mathcal{T} \in \mathbb{R}^{n_\theta \times n_\mathrm{x}}$ and $b \in \mathbb{R}^{n_\theta}$. An affine scheduling map will typically result when constructing an embedding of a bilinear system (8.1). Furthermore suppose that the vertex representation of X is available as

$$X = \left\{\bar{x}^1, \ldots, \bar{x}^{q_\mathrm{x}}\right\}. \tag{8.15}$$

Then, the outer-approximation $\hat{T}(X)$ is exact and is given by

$$\hat{T}(X) = T(X) = \operatorname{convh}\left\{\mathcal{T}\bar{x}^1 + b, \ldots, \mathcal{T}\bar{x}^{q_\mathrm{x}} + b\right\}. \tag{8.16}$$

The assumption that the vertex representation (8.15) is available is not restrictive, because the H- and V-representations of the sets in the sequence $\overrightarrow{\mathbb{X}}$ can be constructed simultaneously in the constructions of Section 8.3. Hence, no computationally expensive conversion between the two representations has to be performed.

8.4.2 General scheduling map

In the general case, the scheduling map $T : \mathbb{X} \to \Theta$ is a non-linear function of the form

$$T(x) = \begin{bmatrix} t_1(x) \\ \vdots \\ t_{n_\theta}(x) \end{bmatrix} \tag{8.17}$$

where for each $i \in [1..n_\theta]$, $t_i : \mathbb{X} \to \mathbb{R}$. For a possibly non-linear and non-convex function $T(\cdot)$, obtaining a convex outer-approximator $\hat{T}(\cdot)$ is not always straightforward. This section suggest a few ways that can be used to obtain these approximations in a practical setting.

Perhaps the simplest way, conceptually, to get an outer approximation $\hat{T}(X)$ of $T(X)$ is to use gridding or sampling of X. Suppose that a number of M points $\hat{x}_i \in X$, $i \in [1..M]$, has been selected. Then,

$$\hat{T}(X) = \text{convh}\left\{T\left(\hat{x}^1\right), \ldots, T\left(\hat{x}^M\right)\right\} \tag{8.18}$$

where the computation of the convex hull only requires the elimination of redundant vertices (Fukuda 2004). In low dimensions, gridding can work efficiently. The main caveat is that the grid has to be dense enough in order to guarantee that $\hat{T}(\cdot)$ is indeed a valid outer-approximation in the sense of Definition 8.3.

Another conceptually simple way of constructing the convex outer-approximation $\hat{T}(X)$ is to compute the interval hypercube

$$\hat{T}(X) = \left[\min_{x \in X} t_1(x), \max_{x \in X} t_1(x)\right] \times \cdots \times \left[\min_{x \in X} t_{n_\theta}(x), \max_{x \in X} t_{n_\theta}(x)\right]. \tag{8.19}$$

Because the functions $t_i(\cdot)$ are non-convex in general, computing the minima and maxima in (8.19) requires the use of an optimization algorithm that is guaranteed to find global optima, e.g., (Breiman and Cutler 1989). This optimization has to be done on-line, but only once when the controller is initialized with the constructed initial constraint sequence. Besides the potentially heavy computational burden of global optimization, a possible downside of this approach is that the restriction to an interval hypercube may lead to a conservative over-approximation. Possibly, this conservatism can be reduced by also considering rotations.

An alternative approach that avoids both the need for global optimization and gridding, can be derived based on the method for non-linear reachability analysis presented in (Althoff et al. 2008). Therein, the idea is to construct outer approximations of non-linear functions applied to sets using a combination of Taylor expansion and interval analysis (Jaulin et al. 2001). To apply this approach to the present setting, the following assumption is necessary.

Assumption 8.2. *The function* $T(\cdot)$ *in* (8.17) *is twice continuously differentiable.*

Let $Df(z)$ denote the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ evaluated at the point $x = z$ with $z \in \mathbb{R}^n$, and let $D^2 f(z)$ denote its Hessian matrix evaluated at $x = z$. The functions $t_i(\cdot)$ can be approximated by a first-order Taylor series plus a Lagrange remainder term as

$$t_i(x) \in t_i\left(x^\star\right) + \left(x - x^\star\right)^\top Dt_i\left(x^\star\right) \oplus \mathcal{L}_i\left(x, x^\star\right) \tag{8.20}$$

where the linearization point $x^\star \in X$ is arbitrary and where the set of possible values of the Lagrange remainder is given by

$$\mathcal{L}_i\left(x, x^\star\right) = \left\{\frac{1}{2}\left(x - x^\star\right)^\top D^2 t_i\left(x^\star + \alpha\left(x - x^\star\right)\right)\left(x - x^\star\right) \,\middle|\, \alpha \in [0,1]\right\}, \tag{8.21}$$

see (Berz and Hoffstätter 1998). The idea is to construct a "global" bound $\bar{\mathcal{L}}_i$ for the remainder, such that

$$\forall (x, x^\star) \in X \times X:\ \mathcal{L}_i\left(x, x^\star\right) \subseteq \bar{\mathcal{L}}_i$$

and, evidently,

$$t_i(x) \in t_i\left(x^\star\right) + \left(x - x^\star\right)^\top Dt_i\left(x^\star\right) \oplus \bar{\mathcal{L}}_i.$$

An expression for the set $\bar{\mathcal{L}}_i$ can be derived following the same basic approach as (Althoff et al. 2008). Nonetheless, due to the different problem setting adopted in this chapter, the result is summarized together with a self-contained proof in the following proposition.

Proposition 8.2 (Based on (Althoff et al. 2008, Propositions 1–2)). *An overbound $\bar{\mathcal{L}}_i$ for the Lagrange remainder $\mathcal{L}\left(x, x^\star\right)$ is*

$$\begin{aligned}\forall (x, x^\star) \in X \times X : \mathcal{L}_i\left(x, x^\star\right) &\in \bar{\mathcal{L}}_i \\ &= \left[-\frac{1}{2} e^\top\left(x^\star\right) \max_{x \in X}\left|D^2 t_i(x)\right| e\left(x^\star\right),\ \frac{1}{2} e^\top\left(x^\star\right) \max_{x \in X}\left|D^2 t_i(x)\right| e\left(x^\star\right)\right]\end{aligned}$$

where

$$e\left(x^\star\right) = \begin{bmatrix} \max_{i \in [1..q_\mathrm{x}]}\left|\bar{x}_1^i - x_1^\star\right| \\ \vdots \\ \max_{i \in [1..q_\mathrm{x}]}\left|\bar{x}_{n_\mathrm{x}}^i - x_{n_\mathrm{x}}^\star\right| \end{bmatrix}.$$

Furthermore, the choice of linearization point

$$x^\star = \frac{1}{q_\mathrm{x}} \sum_{i=1}^{q_\mathrm{x}} \bar{x}^i \tag{8.22}$$

is "optimal" in the sense that it gives the smallest possible interval $\bar{\mathcal{L}}_i$.

Proof. Starting from the definition of $\mathcal{L}\left(x, x^\star\right)$ in (8.21), for fixed $x^\star \in X$, the following chain of inclusions can be derived:

$$\begin{aligned}
\left|\mathcal{L}\left(x,x^\star\right)\right| &= \left\{\frac{1}{2}\left|\left(x-x^\star\right)^\top D^2 t_i\left(x^\star+\alpha\left(x-x^\star\right)\right)\left(x-x^\star\right)\right| \middle| \alpha\in[0,1]\right\} \\
&\subseteq \frac{1}{2}\left\{\left|\left(x-x^\star\right)^\top D^2 t_i\left(x^\star+\alpha\left(x-x^\star\right)\right)\left(x-x^\star\right)\right| \middle| \alpha\in[0,1], x\in X\right\} \\
&\subseteq \frac{1}{2}\left[0, \max_{x\in X,\alpha\in[0,1]}\left|\left(x-x^\star\right)^\top D^2 t_i\left(x^\star+\alpha\left(x-x^\star\right)\right)\left(x-x^\star\right)\right|\right] \\
&\subseteq \frac{1}{2}\left[0, \max_{x\in X,\alpha\in[0,1]}\left|x-x^\star\right|^\top \left|D^2 t_i\left(x^\star+\alpha\left(x-x^\star\right)\right)\right|\left|x-x^\star\right|\right] \\
&\subseteq \frac{1}{2}\left[0, \max_{x\in X}\left|x-x^\star\right|^\top \max_{x\in X,\alpha\in[0,1]}\left|D^2 t_i\left(x^\star+\alpha\left(x-x^\star\right)\right)\right| \max_{x\in X}\left|x-x^\star\right|\right] \\
&\subseteq \frac{1}{2}\left[0, \max_{x\in X}\left|x-x^\star\right|^\top \max_{x\in X}\left|D^2 t_i\left(x\right)\right| \max_{x\in X}\left|x-x^\star\right|\right]
\end{aligned} \tag{8.23}$$

where all max-operations are done element-wise. The last line follows because

$$S(x^\star) = \left\{x^\star+\alpha\left(x-x^\star\right) \,\middle|\, x\in X, \alpha\in[0,1]\right\} = X$$

independently of $x^\star$. (It can be shown that for any $x^\star \in X$, $S(x^\star) \subseteq X$ and $X \subseteq S(x^\star)$, and therefore $S(x^\star) = X$). Considering that a vertex representation of X is assumed to be available in terms of (8.15), it holds that

$$\max_{x\in X}\left|x-x^\star\right| = \begin{bmatrix} \max_{x\in X}\left|x_1-x_1^\star\right| \\ \vdots \\ \max_{x\in X}\left|x_{n_\mathrm{x}}-x_{n_\mathrm{x}}^\star\right| \end{bmatrix} = \begin{bmatrix} \max_{i\in[1..q_\mathrm{x}]}\left|\bar{x}_1^i-x_1^\star\right| \\ \vdots \\ \max_{i\in[1..q_\mathrm{x}]}\left|\bar{x}_{n_\mathrm{x}}^i-x_{n_\mathrm{x}}^\star\right| \end{bmatrix}.$$

In the above relation, the linearization point $x^\star$ is arbitrary, and the result holds for any choice $x^\star \in X$. However, it makes sense to find the point $x^\star \in X$ that minimizes the size of the interval to which $\left|\mathcal{L}\left(x,x^\star\right)\right|$ belongs according to (8.23). This is done by taking $x^\star$ as the centroid of X, i.e., as the arithmetic mean of its vertices

$$x^\star = \frac{1}{q_\mathrm{x}}\sum_{i=1}^{q_\mathrm{x}} \bar{x}^i. \tag{8.24}$$

This can be seen by noting that in $\bar{\mathcal{L}}_i$, the function $D^2 t_i(\cdot)$ is independent of $x^\star$. Thus, minimizing size of the interval $\bar{\mathcal{L}}_i$ as a function of $x^\star$ amounts to minimizing the absolute values of the elements of $e(x^\star)$: this is accomplished by choosing $x^\star$ as (8.24). □

In Proposition 8.2, evaluation of the intervals $\bar{\mathcal{L}}_i$ still requires computing the maximum $\max_{x\in X}|D^2 t_i(x)|$ where $D^2 t_i(\cdot)$ is the second derivative of $t_i(\cdot)$. In (Althoff et al. 2008), it is proposed to compute this bound using interval analysis (Jaulin et al. 2001). The principal idea of interval analysis is to determine, given a (multi-dimensional) interval $[a, b] \subset \mathbb{R}^n$ and a function $f : \mathbb{R}^n \to \mathbb{R}^m$ the bounds

$$f([a, b]) = \left[\min_{x \in [a,b]} f(x), \max_{x \in [a,b]} f(x) \right]$$

in a computationally efficient manner, i.e., without using global optimization (the max- and min-operations in the above given equation are taken element-wise). A list of how interval approximation for a number of common functions can be computed is given in (Althoff and Grebenyuk 2016), and a commercially available toolbox implementing computationally efficient interval methods is (Rump 1999).

It can be expected that this approximation works well for weakly non-linear functions with small second derivatives, such that the linearization error bound from Proposition 8.2 is small. In (Althoff et al. 2008) it is shown that, if necessary, the linearization error bounds can be reduced by splitting the sets X. This approach is not further pursued here.

Based on the discussion above, some other approaches that could be considered in a practical scenario are the following:

- Apply interval analysis directly on the non-linear function (8.17), using, e.g., the implementation of (Rump 1999). This yields an interval hypercube of the form (8.19), but depending on function, it may result in a coarse over-approximation.
- Consider the first-order approximation (8.20). Instead of the interval result in Proposition 8.2, another method (such as gridding or sampling) to bound the Lagrange remainder could be used.

8.5 Numerical examples

In this section, the properties of the approach are demonstrated on numerical examples: a controlled Van der Pol oscillator, an inverted pendulum driven by a DC motor, and a two-tank system.

8.5.1 Controlled Van der Pol oscillator

This example considers a controlled Van der Pol oscillator described by $\ddot{q}(t) = \mu \left(1 - q^2\right) \dot{q}(t) - q(t) + u(t)$, similar to the system used in (Cisneros et al. 2016). An equivalent LPV form of this non-linear model is

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} 0 & 1 \\ -1 & \mu(1-\theta) \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \\ \theta(t) &= T(x(t)) = \left(\begin{bmatrix} 1 & 0 \end{bmatrix} x(t) \right)^2 \end{aligned}$$

with the state vector $x = \begin{bmatrix} q & \dot{q} \end{bmatrix}^\top$. For control and simulation, the continuous-time LPV representation above is discretized with sampling time τ using the Euler approach, which gives

Case	Mean	Min	Max	Standard deviation
NMPC (`fmincon`)	48	34	74	13
TMPC (proposed method)	10	9.1	12	1.0

Table 8.1: Computation times per sample for the Van der Pol-example (values in ms).

the discrete-time representation

$$x(k+1) = \begin{bmatrix} 1 & \tau \\ -\tau & 1+\tau\mu(1-\theta) \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \tau \end{bmatrix} u(k)$$
$$\theta(k) = T(x(k)) = \left(\begin{bmatrix} 1 & 0 \end{bmatrix} x(k)\right)^2 .$$

The used damping parameter is $\mu = 2$ and the sampling time is set to $\tau = 0.1$. The scheduling map is not affine, however it is convex and exact upper and lower bounds on $T(X)$ can be obtained for polytopes $X \subseteq \mathbb{X}$. In this example, the following settings were used:

- The stage cost was of the center-based quadratic type, i.e., of the form (8.8) with $p = 2$;
- The MPC tuning parameters were set to $Q = I$, $R = 0.1$ and $P = 15$;
- The prediction horizon was $N = 10$;
- The state constraints are $\mathbb{X} = \{x \mid \|x\|_\infty \leq 1\}$;
- The input constraint is $\mathbb{U} = \{u \mid |u| \leq 1\}$;
- The set $\breve{\mathbb{X}}$ was chosen to be $\breve{\mathbb{X}} = 0.5\mathbb{X}$.
- A controlled 0.96-contractive set X_f satisfying Definition 8.4 was computed using the method of Appendix A. The set has 4 vertices.

The simulation results are shown in Figure 8.2. For this example, no special initialization of $\vec{\mathbb{X}}(0)$ was necessary: the method already works well with $\vec{\mathbb{X}}(0) = \mathbb{X}^N$. The performance of the proposed LPV-embedding based MPC is compared to that of a full NMPC formulation. The same terminal set and terminal cost (8.10) were used in both the proposed tube-based and in the NMPC. In Figure 8.2, it can be seen that the obtained closed-loop trajectories are comparable. The computation time per sample of the proposed method was lower than that of the NMPC implemented with the `fmincon` function of MATLAB R2017b, as summarized in Table 8.1. The greater efficiency comes at the cost of a slightly reduced domain of attraction $\mathcal{X}_N(\cdot,\cdot)$, as visible in Figure 8.2.

8.5.2 Electrically driven inverted pendulum

In this numerical example, the purpose is to control the angle q of the electrically driven inverted pendulum shown in Figure 8.3. Define the state vector

$$x = \begin{bmatrix} q \\ \dot{q} \\ i \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{8.25}$$

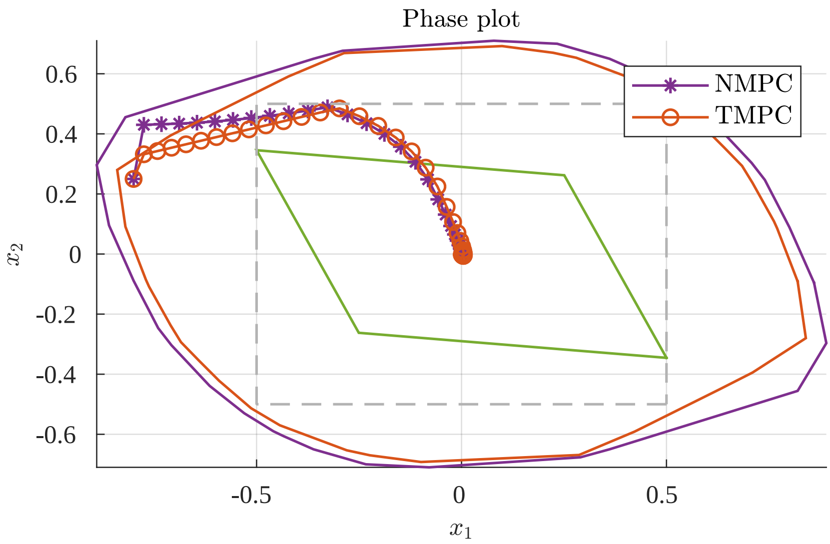


Figure 8.2: Phase plot of the closed-loop trajectory in the Van der Pol-example with terminal set (green), set $\check{\mathbb{X}}$ (dashed grey), and estimated domains of attraction.

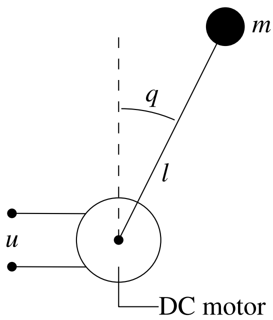


Figure 8.3: The electrically driven inverted pendulum.

Description	Symbol	Value	Unit
Electrical resistance	R	1.0	Ω
Electrical inductance	L	$1.0 \cdot 10^{-3}$	H
Motor constant	k	$6.0 \cdot 10^{-2}$	NA^{-1}
Friction coefficient	b	$1.0 \cdot 10^{-3}$	Nsm^{-1}
Pendulum mass	m	$7.0 \cdot 10^{-2}$	kg
Pendulum length	l	$1.0 \cdot 10^{-1}$	m
Pendulum inertia	$J = ml^2$	$7.0 \cdot 10^{-4}$	kgm^2
Standard gravity	g	9.81	ms^{-2}

Table 8.2: Model parameters for Example 2.

where q is the angle of the pendulum, $\dot{q}$ is the angular velocity, and i is the motor current. Then, the system is described by the non-linear differential equation

$$\dot{x}(t) = \begin{bmatrix} x_2(t) \\ \frac{mgl}{J}\sin x_1(t) - \frac{b}{J}x_2(t) + \frac{k}{J}x_3(t) \\ -\frac{k}{L}x_2(t) - \frac{R}{L}x_3(t) + \frac{1}{L}u(t) \end{bmatrix}$$

where u is the input voltage that is applied to the motor. The model parameters are displayed in Table 8.2. Introduce a scheduling variable

$$\theta(k) = T(x(k)) = \operatorname{sinc}(x_1(k)),$$

where

$$\operatorname{sinc}(x) := \begin{cases} 1, & x = 0, \\ \frac{\sin(x)}{x}, & x \neq 0. \end{cases}$$

The dynamics (8.25) can now be written in the equivalent LPV embedding form as

$$\begin{aligned} \dot{x}(t) &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\frac{b}{J} & \frac{k}{J} \\ \frac{mgl}{J}\theta(t) & -\frac{k}{L} & \frac{-R}{L} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u(t), \\ \theta(t) &= \operatorname{sinc}(x_1(t)). \end{aligned} \tag{8.26}$$

For control design, (8.26) is discretized using a second-order polynomial approximation (Tóth 2010, Chapter 6). This polynomial approximation is more accurate than the first-order (Euler) approximation that is frequently used, but has the disadvantage that it can introduce polynomial parameter dependencies and a parameter-varying input matrix in the discrete-time model. Due to the particular structure of (8.26), however, the second-order discretization still has an affine dependency on θ and a constant B-matrix. The sampling time that is used in this experiment is $\tau = 0.04$ s, which leads to a discrete-time embedding with the following numerical values:

$$x(k+1) = \left(\begin{bmatrix} 1.00 & 0.04 & 0.07 \\ 0 & 0.90 & 2.64 \\ 0 & -0.02 & 0.64 \end{bmatrix} + \begin{bmatrix} 0.08 & 0 & 0 \\ 3.81 & 0.08 & 0 \\ -0.05 & 0 & 0 \end{bmatrix} \theta(k) \right) x(k) + \begin{bmatrix} 0 \\ 0.69 \\ 0.32 \end{bmatrix} u(k),$$
$$\theta(k) = \operatorname{sinc}(x_1(k)). \tag{8.27}$$

The system is subject to the constraints

$$\mathbb{X} = \left\{ x \in \mathbb{R}^3 \,\middle|\, |x_1| \le 2\pi, |x_2| \le 18, |x_3| \le 6 \right\},$$
$$\mathbb{U} = \{ u \in \mathbb{R} \mid |u| \le 8 \}.$$

For the simulation, the following settings were used:

- The prediction horizon and tuning parameters were set to $N = 6$, $Q = I$, and $R = 10$. The infinity-norm based cost (3.13) was used.
- A 0.98-contractive terminal set X_f was computed using Algorithm 2.1, with $\breve{\mathbb{X}} = \left\{ x \in \mathbb{R}^3 \,\middle|\, |x_1| \le \frac{1}{2}\pi, |x_2| \le 18, |x_3| \le 2 \right\}$. The set X_f is contractive under a robustly stabilizing linear state feedback $u = Kx$, and has 32 vertices.
- The initial state constraints sequence and the corresponding scheduling sets were generated using the bounded ROV method of Section 8.3.1 with $\delta = \begin{bmatrix} \frac{1}{4}\pi & \infty & \infty \end{bmatrix}^\top$. Because in (8.27) the scheduling map $T(\cdot)$ is only dependent on x_1, it is only necessary to bound the ROV of x_1.

The obtained simulation result is shown in Figure 8.4. The initial state was $x_0 = \begin{bmatrix} \pi & 0 & 0 \end{bmatrix}^\top$. It was found that the tube synthesis problem is infeasible for the considered initial state if all state variables are allowed to vary arbitrarily fast. Thus, the more complex initialization procedure of Sections 8.3–8.4 in this case is really necessary. As expected, the pendulum is driven to its upright equilibrium position $x_1 = q = 0$ and state and input constraints are respected. Note that the discrete-time embedding (8.27) was used as a prediction model in the controller, while the non-linear continuous-time model was used for simulation.

8.5.3 Two-tank system

The two-tank system from (Angeli et al. 2000; Casavola et al. 2003) is depicted in Figure 8.5. A flow of liquid u with density ρ is being pumped into the upper tank. This tank has a cross-sectional area (CSA) S_1. Through a pipe with CSA A_1, liquid flows out into the lower tank which has a CSA S_2. From the lower tank, in turn, liquid flows out through a pipe with CSA A_2. It is desired to regulate the liquid levels h_1 and h_2 at a given setpoint. For this purpose, the input flow u is available as a control input.

The dynamics of the system are described by the non-linear differential equations

$$\rho S_1 \dot{h}_1 = -\rho A_1 \sqrt{2 g h_1} + u,$$
$$\rho S_2 \dot{h}_2 = \rho A_1 \sqrt{2 g h_1} - \rho A_2 \sqrt{2 g h_2}, \tag{8.28}$$

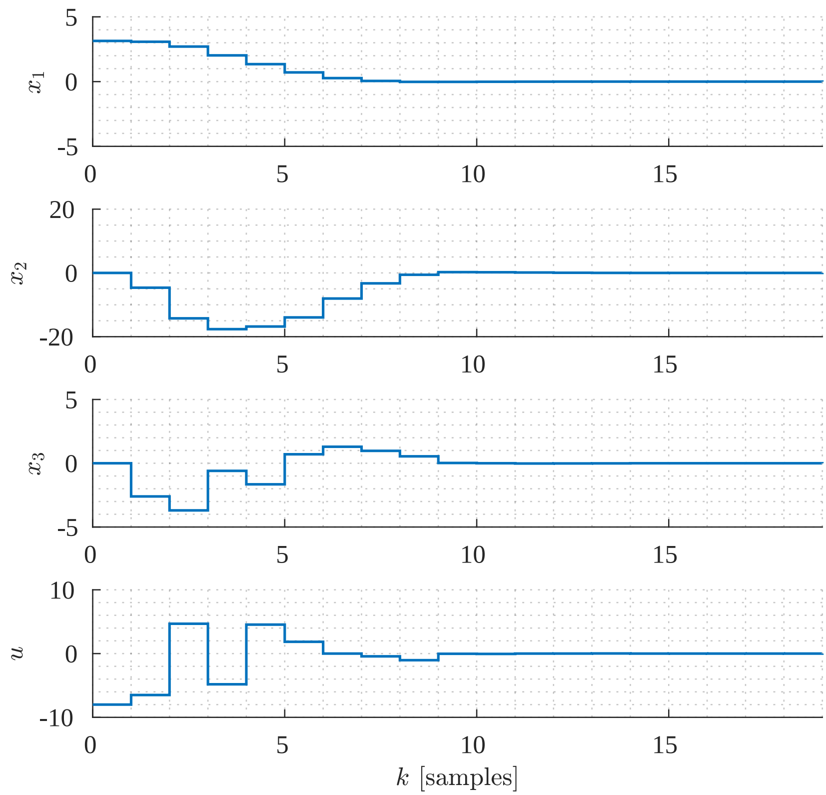


Figure 8.4: Simulation result for the electrically driven inverted pendulum.

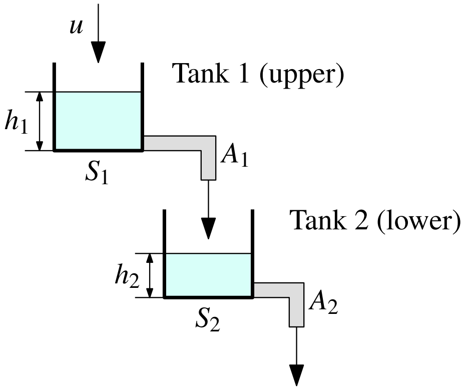


Figure 8.5: The two-tank system.

Description	Symbol	Value	Unit
Tank 1 cross-sectional area	S_1	2500	cm^2
Pipe 1 cross-sectional area	A_1	9	cm^2
Tank 2 cross-sectional area	S_2	1600	cm^2
Pipe 2 cross-sectional area	A_2	4	cm^2
Standard gravity	g	980	cms^{-2}
Density of the liquid	ρ	0.001	$\mathrm{kg} \cdot \mathrm{cm}^{-3}$

Table 8.3: Parameters of the two-tank system, the same as used in (Casavola et al. 2003).

Description	Symbol	Value	Unit
Maximum input flow	$\overline{u}$	4	kgs^{-1}
Tank 1 maximum level	$\overline{h}_1$	35	cm
Tank 1 minimum level	$\underline{h}_1$	1	cm
Tank 2 maximum level	$\overline{h}_2$	200	cm
Tank 2 minimum level	$\underline{h}_2$	10	cm

Table 8.4: Constraints of the two-tank system, the same as used in (Casavola et al. 2003).

with the parameters shown in Table 8.3. The input flow is subject to a constraint

$$\mathbb{U} = \{u \,|\, 0 \leq u \leq \overline{u}\}$$

and the liquid levels have to satisfy the bounds described by the state constraint set

$$\mathbb{X} = \left\{ x = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \middle| \begin{bmatrix} \underline{h}_1 \\ \underline{h}_2 \end{bmatrix} \leq \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \leq \begin{bmatrix} \overline{h}_1 \\ \overline{h}_2 \end{bmatrix} \right\},$$

where the values of the constraints are shown in Table 8.4.

In (Casavola et al. 2003), it is shown that the non-linear dynamics (8.28) can be embedded in a (continuous-time) LPV state-space (LPV-SS) representation by introducing the scheduling variables

$$\theta_1(h_1) = \frac{1/\sqrt{h_1} - 1/\sqrt{\underline{h}_1}}{1/\sqrt{\overline{h}_1} - 1/\sqrt{\underline{h}_1}}, \quad \theta_2(h_2) = \frac{1/\sqrt{h_2} - 1/\sqrt{\underline{h}_2}}{1/\sqrt{\overline{h}_2} - 1/\sqrt{\underline{h}_2}}, \tag{8.29}$$

where the scheduling set corresponding to (8.29) is

$$\Theta = \left\{ \begin{bmatrix} \theta_1(h_1) \\ \theta_2(h_2) \end{bmatrix} \middle| \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \in \mathbb{X} \right\} = \left\{ \theta \in \mathbb{R}^2 \,\middle|\, 0 \leq \theta \leq 1 \right\}. \tag{8.30}$$

After discretizing the resulting continuous-time embedding using the Euler approach with a sampling time of $\tau = 0.9$ seconds, the discrete-time LPV embedding

$$\begin{aligned} x(k+1) &= \left(\begin{bmatrix} 0.86 & 0 \\ 0.22 & 0.97 \end{bmatrix} + \begin{bmatrix} 0.12 & 0 \\ -0.19 & 0 \end{bmatrix} \theta_1(k) + \begin{bmatrix} 0 & 0 \\ 0 & 0.02 \end{bmatrix} \theta_2(k) \right) x(k) + \begin{bmatrix} 0.36 \\ 0 \end{bmatrix} u(k), \\ \theta(k) &= T(x(k)), \end{aligned} \tag{8.31}$$

is obtained where $T(\cdot)$ is the scheduling map corresponding to (8.29).

The goal in the simulation is to bring the level h_2 of the second tank to a reference value of 115 cm. By solving (8.28), it is found that this corresponds to the equilibrium state and input values

$$x_{\mathrm{ss}} = \begin{bmatrix} 22.72 \\ 115 \end{bmatrix}, \quad u_{\mathrm{ss}} = 1.90.$$

As was done in (Casavola et al. 2003), the problem of regulating the state to a non-zero equilibrium is converted into a "stabilization" problem by introducing the translated state and input variables

$$\tilde{x} = x - x_{\mathrm{ss}}, \quad \tilde{u} = u - u_{\mathrm{ss}},$$

so that regulating $\tilde{x}$ to zero becomes equivalent to regulating x to the desired setpoint. In all of the simulations, the following settings were used:

- The tuning parameters were set to $Q = \mathrm{diag}\{0, 2\}$, $R = 0.1$, and—for the center-based stage cost (8.8)—$P = 5$;
- For the tube-based MPC, a controlled 0.995-contractive terminal set X_{f} with 6 vertices was computed using the method in Appendix A with $\check{\mathbb{X}} = \mathbb{X}$;
- The tube-based MPC was initialized according to the bounded ROV approach of Section 8.3.1, with $\delta = \begin{bmatrix} 2 & 25 \end{bmatrix}^\top$;
- With the scheduling map $T(\cdot)$ from (8.31), even though it is not an affine function, the scheduling set sequence can be computed by evaluating $T(\cdot)$ at the vertices of the constraints sets as done in Section 8.4.1.

For the two-tank system, a total of four different simulations was executed with the following corresponding MPC designs:

Casavola The controller from (Casavola et al. 2003), with $N = 2$ and $x_0 = \begin{bmatrix} 15 & 40 \end{bmatrix}^\top$.

TMPC 1 The tube-based MPC from this chapter with infinity-norm cost function (3.13), and with $N = 4$ and $x_0 = \begin{bmatrix} 15 & 40 \end{bmatrix}^\top$. This simulation allows to compare the behavior of the TMPC with an infinity-norm cost function to (Casavola et al. 2003), which uses a quadratic cost.

TMPC 2 The tube-based MPC from this chapter with infinity-norm cost function (3.13), and with $N = 4$ and $x_0 = \begin{bmatrix} 8 & 40 \end{bmatrix}^\top$. This represents an initial state for which (Casavola et al. 2003) is infeasible. For this initial state, the TMPC was also infeasible when an arbitrarily fast scheduling ROV was assumed: thus, the initialization of the state constraint sets with a bounded ROV δ was necessary in this case.

TMPC 3 The tube-based MPC from this chapter with quadratic center-based cost function, and with $N = 10$ and $x_0 = \begin{bmatrix} 15 & 40 \end{bmatrix}^\top$. This compares the behavior of the TMPC with a quadratic cost function and a longer horizon to (Casavola et al. 2003).

The suggested correction from (Ding and Huang 2007) to the algorithm (Casavola et al. 2003) was applied, requiring that the initial state is contained inside of an ellipsoidal invariant set. This means that the length of the prediction horizon and the assumed scheduling ROV do not influence the domain of attraction (DOA) of (Casavola et al. 2003). For this reason, and because it does not provide a mechanism to ensure that the assumed scheduling ROV is actually satisfied along realized closed-loop trajectories, the controller (Casavola et al. 2003) is run with an assumed arbitrarily fast scheduling ROV.

The simulation results are depicted in Figure 8.6. All controllers eventually bring the system to the desired set point, but in slightly different ways. In particular, it can be observed that the TMPC with the infinity-norm based cost function pushes the system closer to its constraints than the controllers with quadratic cost functions. The TMPC with quadratic cost and longer prediction horizon achieves convergence a bit faster than (Casavola et al. 2003), which uses a short horizon. Note that the computational complexity of (Casavola et al. 2003) grows exponentially in the horizon length: therefore, its performance for longer horizons was not investigated.

For the purpose of further illustration, the realized closed-loop trajectories are shown in a phase plot in Figure 8.7. The initial tube for "TMPC 2" is shown in Figure 8.8, together with the generated state constraint sets: as expected, every tube cross section X_i is contained inside of the corresponding constraint set $\mathbb{X}_i$.

8.6 Concluding remarks

This concluding section starts with Section 8.6.1, which contains a short discussion on the possibility of including input non-linearities in the embedding (8.2). A summary of the chapter and suggestions for future work are presented in Section 8.6.2.

8.6.1 Extension to input non-linearities

The extension of the approach presented in this chapter to more general embeddings described by

$$\begin{aligned} x(k+1) &= A\left(\theta(k)\right) x(k) + B\left(\theta(k)\right) u(k), \\ \theta(k) &= T\left(x(k), u(k)\right) \end{aligned}$$

should be straightforward. The dependency of $T(\cdot,\cdot)$ on the control input can be dealt with by introducing time-varying sequences of input constraints similar to the state constraint sequence (8.3). With regards to the parameter dependency of the input matrix B, the discussion of Section 4.6.1 applies.

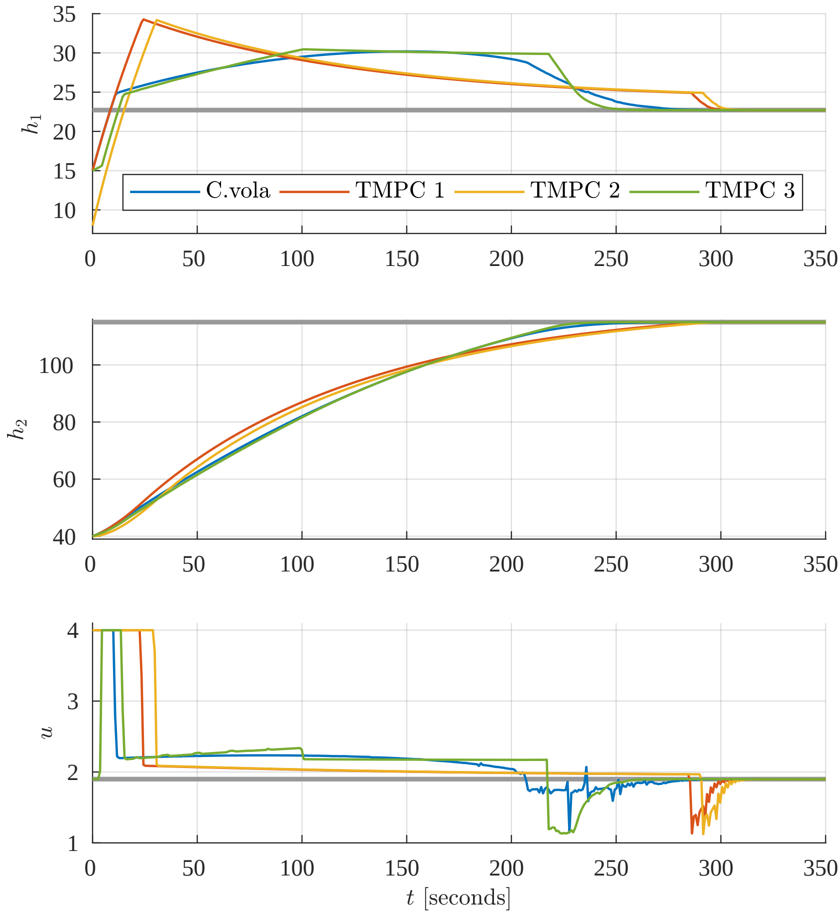


Figure 8.6: Closed-loop state and input trajectories in the two-tank example.

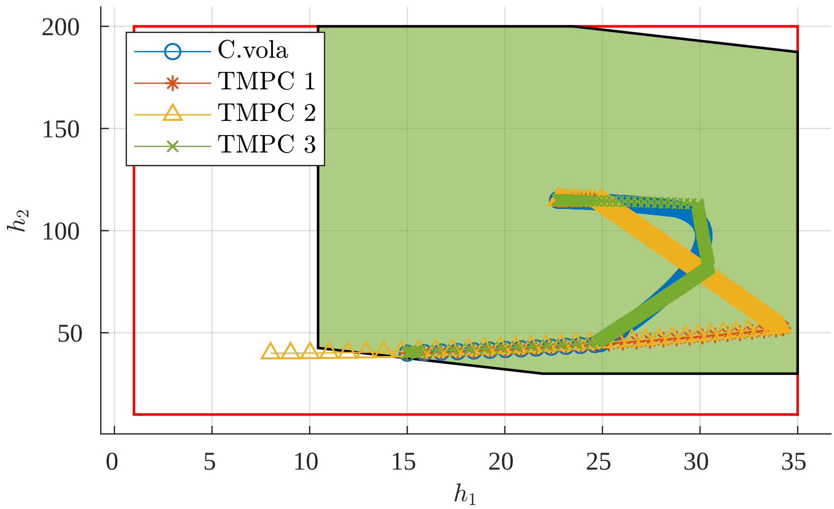


Figure 8.7: Closed-loop state trajectories for the two-tank example in a phase plot.

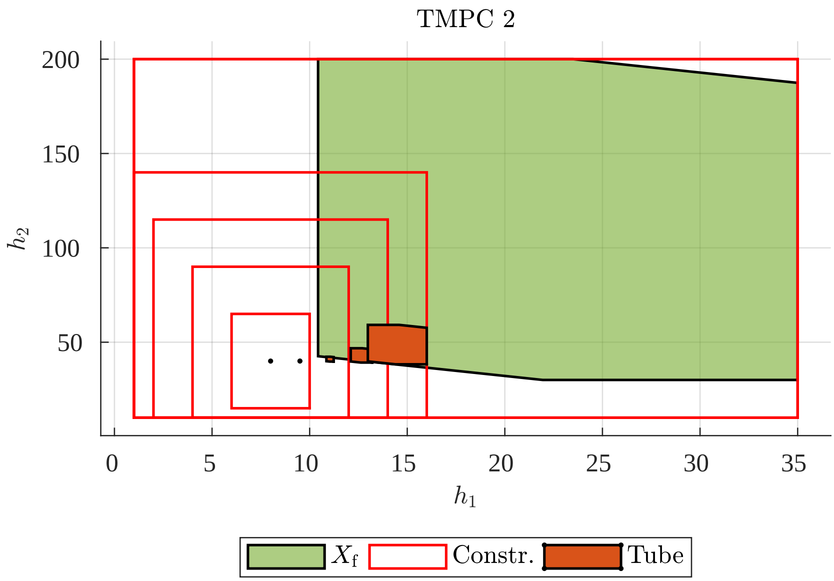


Figure 8.8: Tube $\mathbf{T}_0$ computed at $k = 0$ in the two-tank simulation “TMPC 2”.

8.6.2 Summary

In this chapter, a tube-based MPC strategy applicable to the control of non-linear systems embedded in an LPV-PSS representation was presented. The principal feature of the approach is the bounding of predicted closed-loop state trajectories by a sequence of state constraints that has to be chosen when initializing the controller. The bounding of state trajectories in this way enables the computation of a corresponding sequence of scheduling sets that are valid along all possible realized closed-loop trajectories, thus allowing the derivation of strong feasibility and stability guarantees.

The determination of a sequence of state constraints and corresponding scheduling sets adds a degree of complexity in the initialization stage of the controller that is not present in approaches such as (Casavola et al. 2003; Lu and Arkun 2002), where it is simply *assumed* that the scheduling sets will be valid along realized closed-loop trajectories. This increased complexity appears, however, to be a price that has to be paid for guaranteed recursive feasibility.

The development of this chapter has provided an answer to Research Question Q_5. As summarized in Section 1.2.1, the LPV concept was originally introduced to enable the application of linear control design approaches to non-linear problems. In this chapter, this reasoning was applied to MPC design. The proposed tube-based controller can be used to control a non-linear system, but only requires the on-line solution of convex optimization problems. Naturally, this comes at the cost of reduced achievable performance with respect to NMPC.

A few relevant topics for future research are the following:

- To enlarge the applicability of the approach, the development of more different and improved ways of selecting the initial state constraint sequences are of interest. Related to this, improved and computationally efficient ways for generating the associated scheduling sets should also be investigated.
- It could be investigated if recursive feasibility can still be established under the less-restrictive state constraint condition of Definition 3.4 (see also the discussion below Definition 8.1).
- Modify the method to use of the more advanced tubes of Chapters 4 and 5, as opposed to just the simple homothetic tubes of Section 3.5.

Chapter 9

Anticipative control of a thermal system

THE DEVELOPMENT of the anticipative linear parameter-varying (LPV) model predictive control (MPC) concept was partially motivated in Chapter 1 by a thermal control problem from semiconductor lithography. To study if the proposed tube-based MPC solutions can indeed be beneficial for this application, their performance is investigated in a simulation study on a simplified version of this control problem. The effect of including knowledge on the possible future immersion hood (IH) trajectories is investigated for different system configurations. It is found that in certain cases, anticipative control can give some performance benefit as measured in terms of the realized closed-loop cost. The initial results from this chapter can serve as a basis for further research on the application of anticipative control techniques to related thermal problems.

9.1 Introduction

The main steps in the production of integrated circuits (ICs) are depicted in Figure 9.1. In the central "exposure" or "lithography" step, a pattern that corresponds to the IC that is being manufactured is projected onto a silicon wafer that is coated with a layer of light-sensitive material (photoresist) (Lawson and Robinson 2016; Moreau 1988; Singh et al. 2013). In subsequent processing steps, the parts of the wafer that were not exposed to light are chemically etched away, thus creating the structure of the IC [1]. To manufacture an IC, this whole process is repeated several times. Every time when the same wafer is exposed, the pattern has to be projected precisely on top of the previously developed pattern with a maximum overlay error in the nanometer range.

In *immersion lithography*, a thin layer of water is held in between the wafer being exposed and the so-called immersion hood (IH) (Figure 9.2). Because water has a higher refractive index than air, this allows the projection of smaller details for the same light wavelength (Lawson and Robinson 2016, Chapter 1). During the exposure, the IH moves over the wafer according to a prescribed trajectory. Whenever the IH crosses over the edge of the wafer, air bubbles are formed in the water and these have to be removed. To achieve this, a so-called bubble extraction system (BES) sucks away the air/water mixture along the wafer edge (Figure 9.3). This process cools the wafer down locally (Kou 2013). The resulting non-uniform temperature

[1] Or vice-versa, depending on whether a "positive" or "negative" photoresist is used.

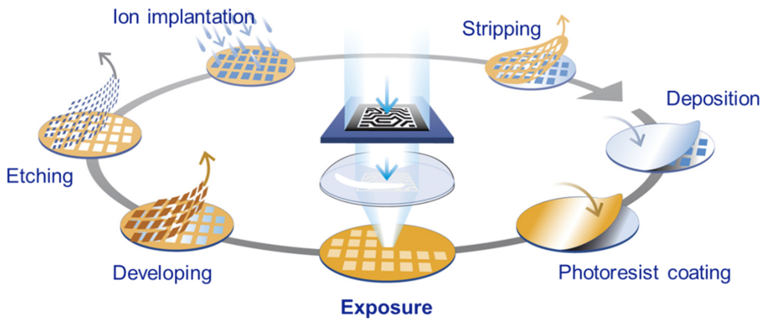


Figure 9.1: Main steps of semiconductor manufacturing. (Figure from (Boef 2016))

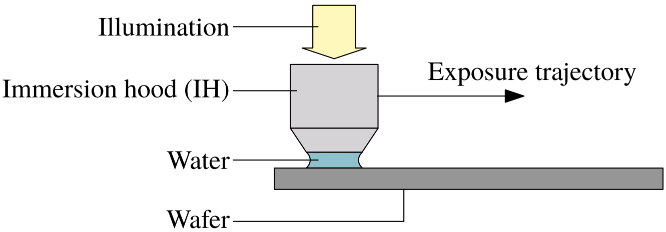


Figure 9.2: Immersion lithography: side view.

distribution leads to mechanical deformations at the wafer surface, which can result in imaging errors causing defects in the produced ICs.

It is therefore desired to control the thermally induced deformations in order to keep them as small as possible. This can be accomplished by using heaters that are located around edges of the wafer (Figure 9.4). These actuators are constrained, because they can only supply heat to the system but not actively cool it down. For this application, this motivates the usage of MPC.

In this system, the position of the IH is an exogenous parameter that influences the thermal behavior of the system. As will be shown in detail later in Section 9.2, this means that the thermal dynamics can be described by an LPV model where the scheduling variable depends on the position of the IH. When an IC is exposed, the IH is controlled to move with respect to

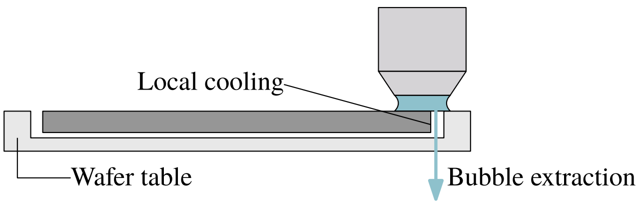


Figure 9.3: Immersion lithography: idea of the bubble extraction system.

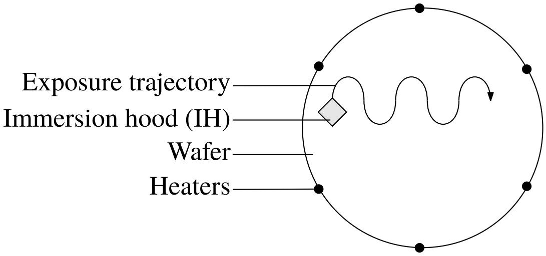


Figure 9.4: Immersion lithography: top view with heaters.

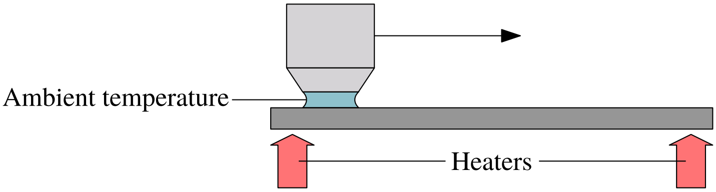


Figure 9.5: The simplified thermal system, a 1-D slice of wafer.

the wafer surface according to a prescribed reference trajectory. This implies that at each time instant, the future position of the IH (i.e., the future scheduling trajectory) is approximately known in advance. Therefore, an "anticipative" LPV MPC approach that can exploit this future information could be an attractive control solution.

The purpose of this chapter is to present a preliminary study to investigate the possible utility of anticipative LPV MPC for this thermal application, and to compare the resulting closed-loop behavior to that obtained with "non-anticipative" control. In particular, the effect of the size of the uncertainty on future IH trajectories is considered for different actuator configurations (maximum heating power and placement).

The remainder of this chapter is structured as follows. A thermal model that will be used for control and simulation is developed in Section 9.2. Next, in Section 9.3, the setup of the simulation study is described. The obtained results are presented in Section 9.4, and some concluding remarks with recommendations for future work are given in Section 9.5.

9.2 Thermal modeling

In this section, a highly simplified thermal model of the wafer is developed. A "slice" of the wafer is modeled as a one-dimensional beam, and spatially discretized using a finite-difference approach. This is illustrated in Figure 9.5. The beam has dimensions $L \times h \times d$ (length $\times$ height $\times$ depth) and is split into M identical elements along the first dimension (length). The volume of a single element is

$$V = \frac{L}{M} hd \tag{9.1}$$

and the heat capacity of a single element is therefore

$$C = V\rho C_{\mathrm{m}} \tag{9.2}$$

where the density ρ ($\mathrm{kgm^{-3}}$) and the specific heat capacity per unit of mass C_{m} ($\mathrm{J \cdot kg^{-1} \cdot K^{-1}}$) are material properties. Let T_i (K) denote the temperature of the i-th element. Its rate of change $\dot{T}_i$ ($\mathrm{Ks^{-1}}$) is governed by the differential equation

$$C\dot{T}_i = \dot{Q}_i \tag{9.3}$$

where $\dot{Q}_i$ (W) is the net flow of heat into the element. There are three sources that are contributing to the heat flow $\dot{Q}_i$:

- Each element exchanges heat with its two neighbors (one neighbor in case of the first and last elements). According to Fourier's law (Hahn and Özisik 2012), the heat that flows into the i-th element from its neighbor j is

$$\dot{Q}_{i,j} = \kappa \frac{Mhd}{L} \left(T_j - T_i\right) \tag{9.4}$$

  where κ ($\mathrm{Wm^{-1}K^{-1}}$) is the thermal conductivity, a material property.

- The contact of an element with the IH is the second contribution to the heat flow $\dot{Q}_i$. Let w (m) denote the width of the IH. Also, the IH is considered to be an ideal heat sink, i.e., a body of constant temperature T_{amb}. Without loss of generality, it is assumed that $T_{\mathrm{amb}} = 0$. Then, the heat flow between the i-th element and the IH is given by

$$\dot{Q}_{i,\mathrm{ih}} = -\theta_i \kappa \frac{wd}{h} T_i \tag{9.5}$$

  where θ_i is the ratio of the IH contact area wd that is in contact with element i. The precise modeling of the scheduling variables θ_i is discussed later.

- The third and last contribution to the heat flow into an element, are heaters. A heater can be attached to any element, providing an external heat flow u_i (W) into that element. These heaters are available as actuators for active control of the temperature distribution.

Note that heat exchange with the environment is not modeled separately, i.e., it is assumed that all heat exchanged with the ambient environment goes through the IH. All of this leads to the following differential equations governing the dynamics of the simplified thermal system:

$$\begin{aligned} C\dot{T}_0 &= \kappa \frac{Mhd}{L} \left(T_1 - T_0\right) - \theta_0 \kappa \frac{wd}{h} T_0 + u_0, \\ \forall i \in [1..M-2]: \quad C\dot{T}_i &= \kappa \frac{Mhd}{L} \left(T_{i-1} + T_{i+1} - 2T_i\right) - \theta_i \kappa \frac{wd}{h} T_i + u_i, \\ C\dot{T}_{M-1} &= \kappa \frac{Mhd}{L} \left(T_{M-2} - T_{M-1}\right) - \theta_{M-1} \kappa \frac{wd}{h} T_{M-1} + u_{M-1}. \end{aligned} \tag{9.6}$$

Besides the simplification of modeling a one-dimensional slice instead of the full three-dimensional wafer structure, another major simplification in the model is the absence of additive disturbance terms. In reality, such terms appear when considering the local cooling effect due to the BES, see Figure 9.3. Because the asymptotic stability results developed in this thesis are applicable only to the case when no persistent additive disturbances are present, these are omitted from the model in this case study. Simulations are instead performed for non-uniform initial temperature distributions, which could represent the distribution that results *after* a BES-induced disturbance has acted on the system.

For MPC, the continuous-time model (9.6) must be temporally discretized. In this chapter, the first-order Euler approximation is used, which leads to the discrete-time model

$$\begin{aligned}
T_0(k+1) &= \tau\frac{\kappa M h d}{CL}\left(T_1(k) - T_0(k)\right) + \left(1 - \tau\theta_0\frac{\kappa w d}{Ch}\right)T_0(k) \\
&\quad + \tau\frac{1}{C}u_0(k), \\
\forall i \in [1..M-2] : T_i(k+1) &= \tau\frac{\kappa M h d}{CL}\left(T_{i-1}(k) + T_{i+1}(k) - 2T_i(k)\right) \\
&\quad + \left(1 - \tau\theta_i\frac{\kappa w d}{Ch}\right)T_i(k) + \tau\frac{1}{C}u_i(k), \\
T_{M-1}(k+1) &= \tau\frac{\kappa M h d}{CL}\left(T_{M-2}(k) - T_{M-1}(k)\right) \\
&\quad + \left(1 - \tau\theta_{M-1}\frac{\kappa w d}{Ch}\right)T_{M-1}(k) + \tau\frac{1}{C}u_{M-1}(k),
\end{aligned} \tag{9.7}$$

where τ is the sampling time in seconds. The above discrete-time model can be written in the standard LPV state-space (LPV-SS) form

$$x(k+1) = A\left(\theta(k)\right)x(k) + Bu(k).$$

In this particular model, $A(\cdot)$ is a tridiagonal matrix and for every frozen scheduling value $\bar{\theta}$ its eigenvalues are distinct, real, and have a magnitude strictly less than one. Because the scheduling variables $\theta_0, \ldots, \theta_{M-1}$ are used to model the effect of the IH touching the wafer, they are functions of its position. This functional dependence is omitted from the notation in (9.7) for simplicity.

As stated before below (9.5), the value of θ_i is defined as the ratio of the area of the IH that is in contact with the i-th element. A similar definition of the scheduling variable was considered before in the 2D-case in (Kou 2013). Let

$$p \in [\frac{w}{2}, L - \frac{w}{2}] = \Pi \tag{9.8}$$

denote the position of the center of the IH with respect to the beam. An expression for the relationship $p \mapsto (\theta_0, \ldots, \theta_{M-1})$ will now be derived. For simplicity, the following assumption is made.

Assumption 9.1. *The width w of the IH is less than or equal to the width of a single element, i.e., $w \leq \frac{L}{M}$.*

Based on Assumption 9.1, to model the dependency of the vector of scheduling variables on the IH position, introduce $a, b \in \mathbb{R}$ as

$$\begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \frac{L}{2M} & 1 \\ \frac{L}{2M} - \frac{w}{2} & 1 \end{bmatrix}^{-1} \begin{bmatrix} \frac{1}{2} \\ 1 \end{bmatrix}$$

and define

$$\bar{\theta}(q) = \begin{cases} \max\{\min\{-aq + b, 1\}, 0\}, & q < 0, \\ \max\{\min\{aq + b, 1\}, 0\}, & q \geq 0. \end{cases}$$

The desired functions $\theta_i(\cdot)$ are then constructed as

$$\forall i \in [0..M-1]: \ \theta_i(p) = \bar{\theta}\left(p - i\frac{L}{M} - \frac{L}{2M}\right).$$

These functions are also referred to as "activation functions". An illustrative plot of their values for the case that $M = 4$ and $w = \frac{L}{2M}$ is provided in Figure 9.6.

Denote

$$\theta(p) = \begin{bmatrix} \theta_0(p) & \theta_1(p) & \cdots & \theta_{M-1}(p) \end{bmatrix}^\top \tag{9.9}$$

and observe that

$$\forall p \in \Pi: \ \sum_{i=0}^{M-1} \theta_i(p) = 1,$$

where Π is the interval defined in (9.8). This implies that

$$\forall p \in \Pi: \ \theta(p) \in \Theta$$

with $\Theta \subseteq \mathbb{R}^M$ being the M-dimensional standard simplex, i.e.,

$$\Theta = \operatorname{convh}\left\{ \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\}. \tag{9.10}$$

The scheduling set Θ in (9.10) has M vertices.

The IH is controlled to follow a known reference exposure trajectory. Therefore, at any time instant, the future trajectory for the position p can be predicted up to a small uncertainty. In the sequel, it is assumed that at each time instant $k \in \mathbb{N}$, the current position $p(k)$ is measured exactly. Furthermore, it is assumed for its future trajectories that it is known that

$$\forall i \in [1..N-1]: \ p(k+i) \in \left(\bar{p}(k+i) \oplus \Delta\right) \cap \Pi \tag{9.11}$$

where $\bar{p}: \mathbb{N} \to \Pi$ is a known reference signal, and the uncertainty interval is

$$\Delta = [-\delta, \delta]$$

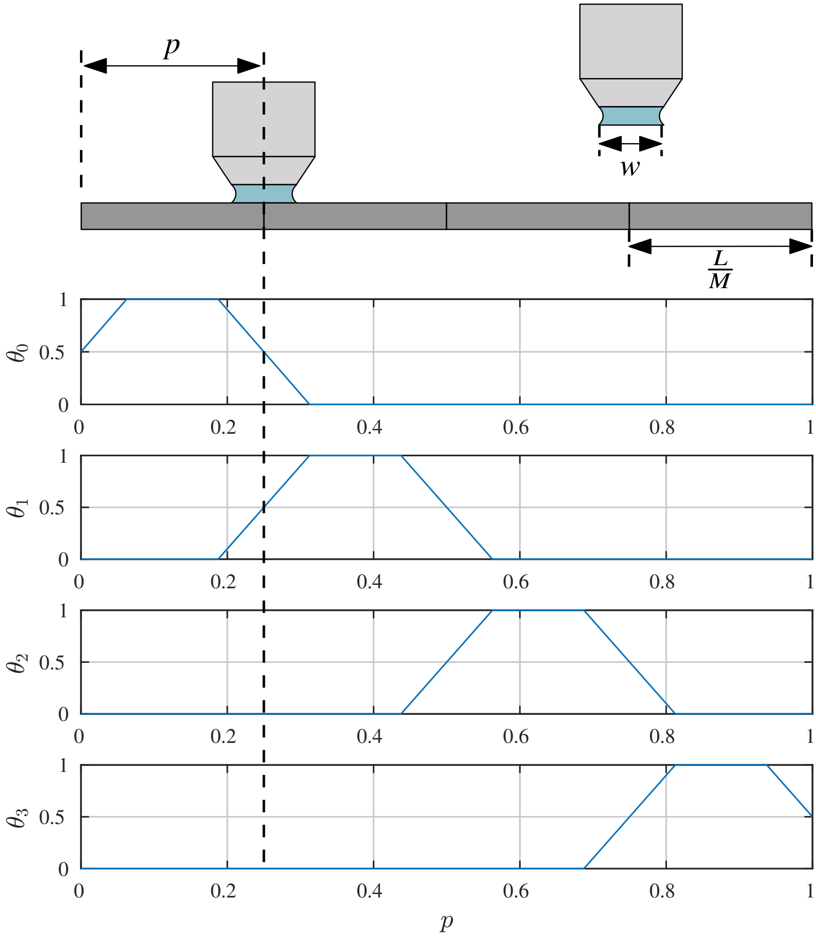


Figure 9.6: Illustration of activation function values as a function of p for the case that $M = 4$ and $w = \frac{L}{2M}$.

Description	Symbol	Value	Unit
Sampling time	τ	50	ms
Prediction horizon	N	20	samples
State weight	Q	I	
Input weight	R	$0.01I$	

Table 9.1: Model settings and controller tuning.

with $\delta \in \mathbb{R}_+$. This description of uncertain future IH positions is like the description of uncertain future scheduling trajectories of Equation 3.4. The difference is that in the current model, an extra step has to be performed in order to translate the sets of future positions into corresponding M-dimensional sets of scheduling values $\Theta_i \subseteq \Theta \subset \mathbb{R}^M$. These sets are defined for all $k \in \mathbb{N}$ as

$$\forall i \in [0..N-1] : \; \Theta_i = \begin{cases} \{\theta\,(p\,(k))\}, & i = 0, \\ \operatorname{convh}\{\theta(p) \,|\, p \in \bar{p}(k+i) \oplus \Delta\}, & i > 0, \end{cases} \tag{9.12}$$

where $p(k)$ is the current IH position that is measured exactly, and $\theta(\cdot)$ is the function defined in (9.9). The computation of the sets Θ_i, $i \in [1..N-1]$ is difficult in principle, because $\theta(\cdot)$ is a non-convex piecewise affine (PWA) function. However, because the position p is scalar, good approximations can be obtained relatively efficiently by gridding over p and eliminating the redundant points to obtain vertex representations of the convex hulls of Θ_i.

9.3 Experiment description

The purpose of the experiment proposed in this section is to investigate the effect of including the "anticipative" knowledge about possible future scheduling trajectories in the thermal control problem. It is not the intention to present a realistic simulation study, but instead to gain some insight in the circumstances under which anticipative control can lead to differences and improvements in the achieved control performance. For this purpose, the simplified thermal model developed previously in Section 9.2 is employed. The considered control problem is regulation of the temperature to the origin, while the IH is moving according to a given trajectory. The system (9.7) is open-loop asymptotically stable, so even when applying no control inputs the temperature will converge to zero eventually. However, through the use of heaters, this convergence can possibly be achieved faster.

Because the main purpose of the simulations presented here is to study the effect of including "anticipative" knowledge on achievable control performance, the influence of other design variables (such as the tube parameterization) is not considered. For this reason, the used controller is the most simple one from this thesis, namely the homothetic TMPC of Section 3.5. The controller tuning and sampling time are set to the values displayed in Table 9.1. Observe that with the given settings, the prediction horizon of $N = 20$ samples corresponds to 1 second.

Description	Symbol	Value	Unit
Thermal conductivity	κ	150	$\mathrm{Wm^{-1}K^{-1}}$
Density	ρ	2330	$\mathrm{kg \cdot m^{-3}}$
Specific heat	C_m	710	$\mathrm{J \cdot kg^{-1} \cdot K^{-1}}$

Table 9.2: Material properties of silicon.

Description	Symbol	Value	Unit
Beam length	L	10	cm
Beam height	h	1.0	mm
Beam depth	d	2.0	mm
IH width	w	1.0	cm
Number of elements	M	4	

Table 9.3: Dimensions.

In the simulations, the material properties that appear in (9.6) are taken to be those of silicon. These properties are summarized in Table 9.2. The assumed physical dimensions of the components are given in Table 9.3. The small number of elements $M = 4$ means that the model (9.7) can only provide a crude approximation of the true thermal dynamics of the beam. This choice is unfortunately necessary due to computational complexity considerations related to the representation of the terminal set (which corresponds also to the shape of the tube cross sections).

For the model (9.7), it is found that the M-dimensional hypercube

$$X_\mathrm{f} = \alpha_\mathrm{f} \left\{ x \in \mathbb{R}^M \;\middle|\; \|x\|_\infty \leq 1 \right\} \tag{9.13}$$

is controlled invariant (i.e., $\lambda = 1$) in the sense of Definition 2.13, where α_f is a scaling such that $X_\mathrm{f} \subseteq \mathbb{X}$. Physically, this follows from the fact that the temperature of an element can never exceed the maximum of the initial temperatures, even if no control inputs are applied. It was unfortunately found that computing controlled contractive sets (i.e., $\lambda < 1$) was difficult[2]. Therefore, the invariant set (9.13) was used as a terminal and cross section shape set. Strictly speaking this is a violation of the developed stability conditions, but as will be seen later, no particular problems were observed in the simulations. The value of the terminal cost (3.19) was computed using a hypothetical value of $\lambda = 0.99$ to avoid a division by zero.

In the sequel, the state constraints are chosen to be

$$\mathbb{X} = \left\{ x \in \mathbb{R}^M \;\middle|\; \|x\|_\infty \leq 15 \right\}$$

and the scaling α_f in (9.13) is set to $\alpha_\mathrm{f} = 5$.

Remark 9.1. *The control approach of Section 3.5 relies, like the other approaches from this thesis, on the availability of the vertex representation of* (9.13). *Because the number of vertices*

[2]For $\lambda < 1$, Algorithm 2.1 did not converge before memory was exhausted.

of the M-dimensional hypercube is 2^M, this automatically leads to a restriction to small values of M.

The movement of the IH is modeled as follows. The reference position $\bar{p}(\cdot)$ in (9.11) is generated according to

$$\bar{p}(0) = \frac{w}{2}, \tag{9.14a}$$

$$v(0) = \bar{v}, \tag{9.14b}$$

$$v(k) = \begin{cases} v(k), & \frac{w}{2} < p(k) < L - \frac{w}{2}, \\ -v(k), & \text{otherwise,} \end{cases} \tag{9.14c}$$

$$\bar{p}(k+1) = p(k) + v(k)\tau, \tag{9.14d}$$

i.e., the IH is controlled to move back-and-forth with a constant speed $\bar{v}$ (m/s), starting from the initial position $\bar{p}(0)$. In particular, (9.14c) means that the IH reverses direction when it reaches the edge of the wafer. In the presented simulations, the value $\bar{v} = 3.0$ cm/s is used.

The value of δ in the description of possible future IH positions in (9.11) represents the uncertainty in its future trajectory. In the simulations, this parameter is going to be varied to study its effect on the achieved control performance. The true realized IH trajectories are generated according to

$$p(k) = (\bar{p}(k) + n(k)) \cap \Pi \tag{9.15}$$

where $n : \mathbb{N} \to \mathcal{N}$ is a random signal with values drawn uniformly from the interval

$$\mathcal{N} = \left[-\frac{w}{10}, \frac{w}{10}\right].$$

In order to be consistent with (9.15), the minimum value of δ in the description (9.11) is therefore $\underline{\delta} = \min\{|\delta| \mid \delta \in \mathcal{N}\} = \frac{w}{10}$.

The heaters are subject to an input constraint of the form

$$\mathbb{U} = \{u \in \mathbb{R}^{n_u} \mid 0 \leq u \leq \bar{u}\}$$

where $\bar{u} \in \mathbb{R}_+$ represents the maximal heating power. The dimension n_u depends on the number of elements in (9.7) to which a heater is attached. In everything that follows, the following two cases will be considered:

Endpoint actuation In the endpoint actuation case, there is a heater connected to the first and last element, so there are $n_u = 2$ control inputs.

Full actuation In this case, there is a heater connected to every element, so there are $n_u = M$ control inputs.

All the simulations are performed for an initial state of the form

$$x_0 = -\tilde{x}_0 \begin{bmatrix} -1 & -1 + 1\frac{4}{3(M-1)} & -1 + 2\frac{4}{3(M-1)} & \cdots & -1 + (M-1)\frac{4}{3(M-1)} \end{bmatrix}^\top \tag{9.16}$$

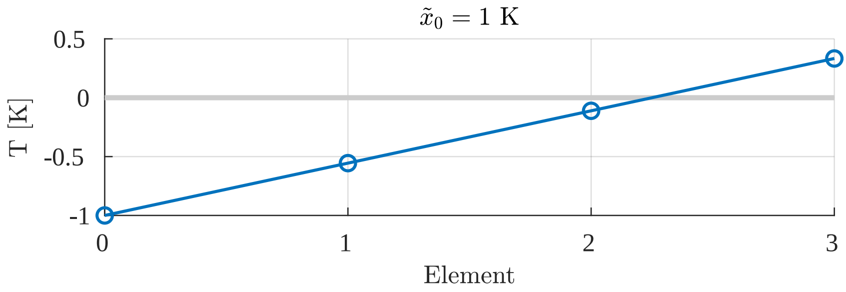


Figure 9.7: Initial temperature distribution according to (9.16), for the case that $\tilde{x}_0 = 1$ K and $M = 4$.

with $\tilde{x}_0 \in \mathbb{R}_+$. This corresponds to a linear temperature distribution in the beam. Such a distribution could result after a BES disturbance (Figure 9.3) has acted on the system, and has cooled down the beam at one side. This initial distribution for $\tilde{x}_0 = 1$ and $M = 4$ is illustrated in Figure 9.7.

The purpose of the simulations that follow is to compare the obtained control performance between anticipative and non-anticipative control. To reiterate, the difference between anticipative and non-anticipative control is as follows:

Anticipative control In the "anticipative" case, the description (9.11) of future IH trajectories is used. The value of δ can be changed according to the level of knowledge on future IH positions that is assumed to be available.

Non-anticipative or "classical" control In this case, no use is made of the description (9.11) of future IH trajectories. Instead, the only assumption that is made is that $p(k+i) \in \Pi$ (see Equation 9.8) for all future time instances $k+i$ with $i \in [1..\infty)$.

In all other respects, the two controllers are the same. The stage cost is of the max-type from Section 3.4.5 with the infinity norm and $c = 1$. Hence, the criterion that will be used to compare the control performance is the simulated closed-loop cost

$$J_{\text{sim}} = \sum_{k=0}^{N_{\text{sim}}} \|Qx(k)\|_\infty + \|Ru(k)\|_\infty \tag{9.17}$$

where $N_{\text{sim}} > 0$ is the length of one simulation in samples. For every simulation scenario, the relative difference

$$\eta = \frac{J^{\text{c}}_{\text{sim}} - J^{\text{a}}_{\text{sim}}}{J^{\text{c}}_{\text{sim}}} \tag{9.18}$$

in the achieved cost for an anticipative ($J^{\text{a}}_{\text{sim}}$) and a non-anticipative ($J^{\text{c}}_{\text{sim}}$) controller is computed. A positive value of η indicates that the anticipative controller, i.e., the controller using the description (9.11) of possible future IH trajectories, outperforms the non-anticipative controller. Because the anticipative controller exploits more detailed information about what is going to happen in the system, it is expected that $\eta \geq 0$ in all cases (i.e., it is expected that using the description (9.11) does not deteriorate performance).

In the next section, some simulation results showing the realized differences between anticipative and non-anticipative control are presented.

9.4 Simulation results

This section presents the results of the simulations carried out in accordance with the experiment description of the previous section. In Section 9.4.1, the endpoint actuation case is considered. The effect of the values $\bar{u}$ and δ on the relative cost difference η are investigated. In Section 9.4.2, the same thing is done, but for the full actuation case. The computational properties of the approaches and the difference between linear time-varying (LTV) and LPV control is highlighted in Section 9.4.3. Finally, in Section 9.4.4, the effect of increasing the IH heat transfer coefficient on the relative cost difference η is considered.

9.4.1 Endpoint actuation case

As a first step, the effect of the size of the input constraint $\bar{u}$ and the uncertainty δ in future IH positions on the relative cost difference (9.18) is investigated. This is done for an endpoint actuator configuration, where a heater is connected to the first and last element of the beam. The used simulation length is $N_{\text{sim}} = 100$ samples, corresponding to 5 seconds.

The results, for different initial state magnitudes $\tilde{x}_0$, are illustrated graphically in Figure 9.8 (see Equation 9.16 for the definition of initial state). In these plots, each square corresponds to a particular combination of values $(\bar{u}, \delta)$. The color of the square indicates the corresponding value of η. A white square indicates that at least one of the tube synthesis problems was infeasible for the given initial state and corresponding combination of $(\bar{u}, \delta)$, so that no η-value could be determined.

From Figure 9.8, it can be observed that there is a region of values of $\bar{u}$ for which anticipative control provides the highest relative improvements. This region approximately covers the range $\bar{u} \in [0.07, 0.32]$. If $\bar{u}$ becomes too large or too small, however, the relative benefit decreases. The maximum relative improvement η for any constraint size is about 10%.

It appears that for small values of $\bar{u}$ (relative to the magnitude of the initial condition), the achievable control performance is limited by the input constraints, and it is almost impossible to attain improvements by means of better control. Likewise, for large $\bar{u}$, the system seems easy to control, and any controller is able to achieve good performance. Only somewhere in between these two extremes there is a "sweet spot" where an anticipative controller can provide some benefit.

A representative scheduling trajectory that was used for these computations, corresponding to the disturbed IH position (9.15), is shown in Figure 9.9. Recall that the scheduling signal $\theta(\cdot)$ is generated from the IH position $p(\cdot)$ according to (9.9), and that $p(\cdot)$ behaves as described by (9.15).

From Figure 9.8, it also appears that the relative improvements increase with improved knowledge (i.e., smaller δ) of the future IH positions. The relationship between δ and η, for one

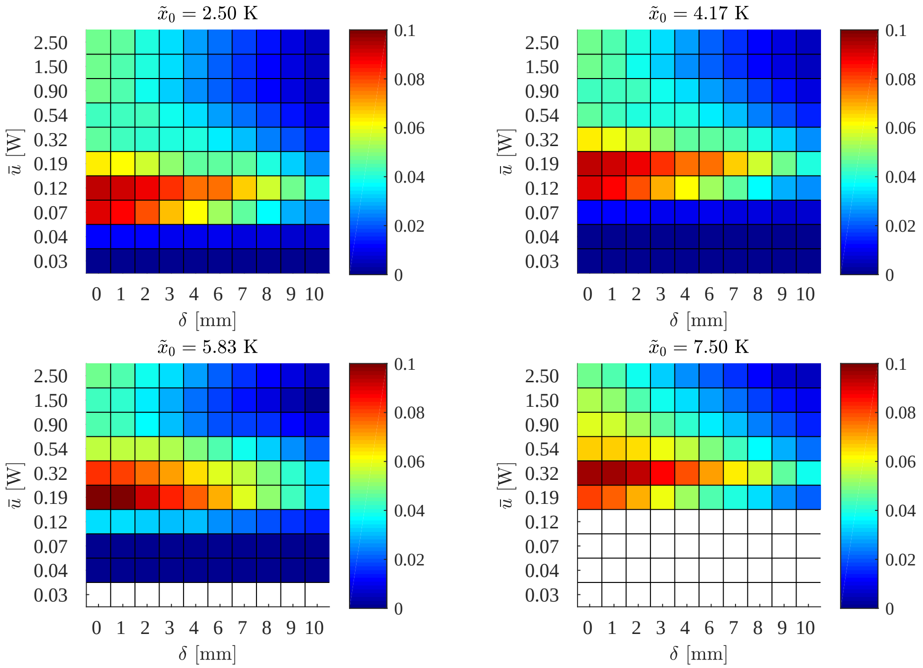


Figure 9.8: Relative differences η between realized closed-loop costs in the "classical" and "anticipative" cases with endpoint actuation, for various combinations of input constraint size $\bar{u}$ and size of uncertainty δ in the future IH positions, and for different initial states $\tilde{x}_0$. A white square indicates that at least one of the tube synthesis problems was infeasible for the given combination $(\tilde{x}_0, \bar{u})$, so that the corresponding η value could not be computed.

particular choice of the input constraint, is shown in Figure 9.10. It is confirmed that smaller δ lead to improved η, which is in agreement with the expectations.

In these simulations, the case that $\delta = 0$ represents an "oracle"-approach in which it was assumed that the uncertain future scheduling trajectory generated by (9.15) is known exactly in advance. Equivalently, this means that the future behavior of the noise signal $n(\cdot)$ is known. This is unrealistic, but represents the best possible closed-loop performance that can be achieved.

For the cases that $\delta > 0$, the value of δ represents the uncertainty in future IH trajectories in terms of a maximum deviation around a nominal trajectory according to (9.11). In Figure 9.10, it can be seen that for small δ the performance of anticipative control is quite close to this theoretical maximum. All considered values $\delta > 0$ are such that $\delta \geq \underline{\delta}$, i.e., are consistent with the true realized IH trajectories (9.15).

The realized closed-loop state and input trajectories for the configuration from Figure 9.10 where a relative improvement of almost 10% is attained, are depicted in Figure 9.11. It is observed that the anticipative controller keeps the first, second, and fourth state variables closer

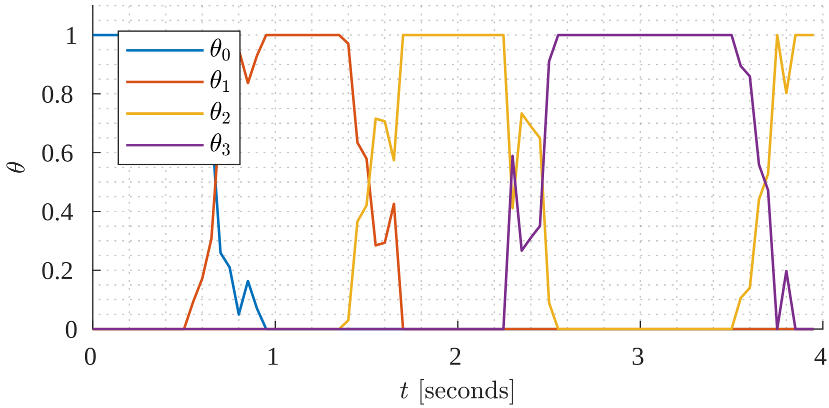


Figure 9.9: Representative scheduling trajectory corresponding to (9.15). The IH speed is $v = 3.0$ m/s.

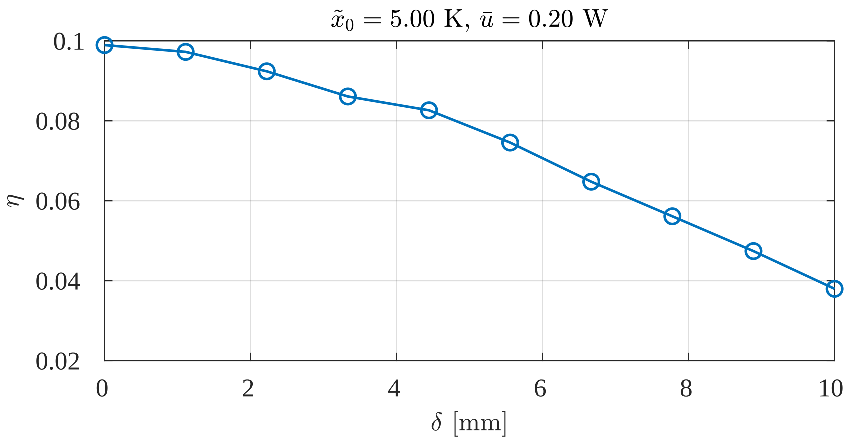


Figure 9.10: Relative difference η in closed-loop cost as function of δ, for fixed initial state $\tilde{x}_0$ and input constraint $\bar{u}$. (Endpoint actuation)

to zero and thereby achieves an improved closed-loop cost (9.17).

9.4.2 Full actuation case

To show what happens with a different actuator configuration, the simulations of Section 9.4.1 are repeated for the full actuation case. As previously, the used simulation length is $N_{\mathrm{sim}} = 100$ samples, corresponding to 5 seconds. The obtained relative differences η between anticipative and non-anticipative control for different values of $\bar{u}$, δ, and $\tilde{x}_0$ are shown in Figure 9.12. As in the previous section, each square corresponds to a particular combination of values $(\bar{u}, \delta)$ and the color of the square indicates the corresponding measured value of η.

As in the endpoint actuation case, there is a region of values of $\bar{u}$ for which anticipative control is able to provide some performance benefit when compared to non-anticipative control. The principal difference with respect to endpoint actuation is that this region is much narrower. The maximum relative improvement η for any initial state magnitude $\tilde{x}_0$ and for any constraint size is about 9.5%.

The relationship between δ and η, for one particular choice of the input constraint, is shown in Figure 9.13. Like in the endpoint actuation case considered previously, the measured value of η is increasing as δ becomes smaller. A representative closed-loop trajectory, showing the difference between non-anticipative and anticipative control, is provided in Figure 9.14.

9.4.3 Computational load and LTV control

A box plot summarizing the computation times necessary to solve the on-line tube synthesis problems for the endpoint actuation case is shown in Figure 9.15. In these plots, the bottom and top of the box respectively indicate the 25th and 75th percentiles of the measured computation times (i.e., 25% of the measured times are below the lower edge of the box, and 75% of the measured times are below the top edge of the box). A line inside of each box indicates the median. The lines that extend from the bottom and top of the boxes indicate the full range of the measured computation times, but excluding times that are considered to be outliers. Outliers (i.e., unusually high or low computation times) are depicted by the "+"-symbol.

It is observed that the anticipative controller is on average somewhat faster than the non-anticipative one: this is because for small δ, the scheduling sets Θ_i generated according to (9.12) have less vertices than the full set Θ. It is emphasized that, due to the exponential complexity in M of representation of the used invariant set, the computation times grow rapidly with M (see Remark 9.1). The computations were carried out on the same computer as the other simulations in this thesis, and its specifications can be found in, e.g., Table 6.1.

One way of significantly decreasing the computational load is to consider an LTV controller, in which it is assumed that the full future scheduling trajectory (9.15) is known exactly in advance. This is called an "oracle LTV" approach and corresponds to the situation with $\delta = 0$ in Section 9.4.1 and Section 9.4.2. The assumption that the future IH trajectory (9.15) is known exactly in advance is, however, unrealistic.

It is also possible to consider an LTV controller based only on the nominal IH position $\bar{p}(\cdot)$. In the sequel, this will be referred to as a "nominal LTV" approach. In this approach, the

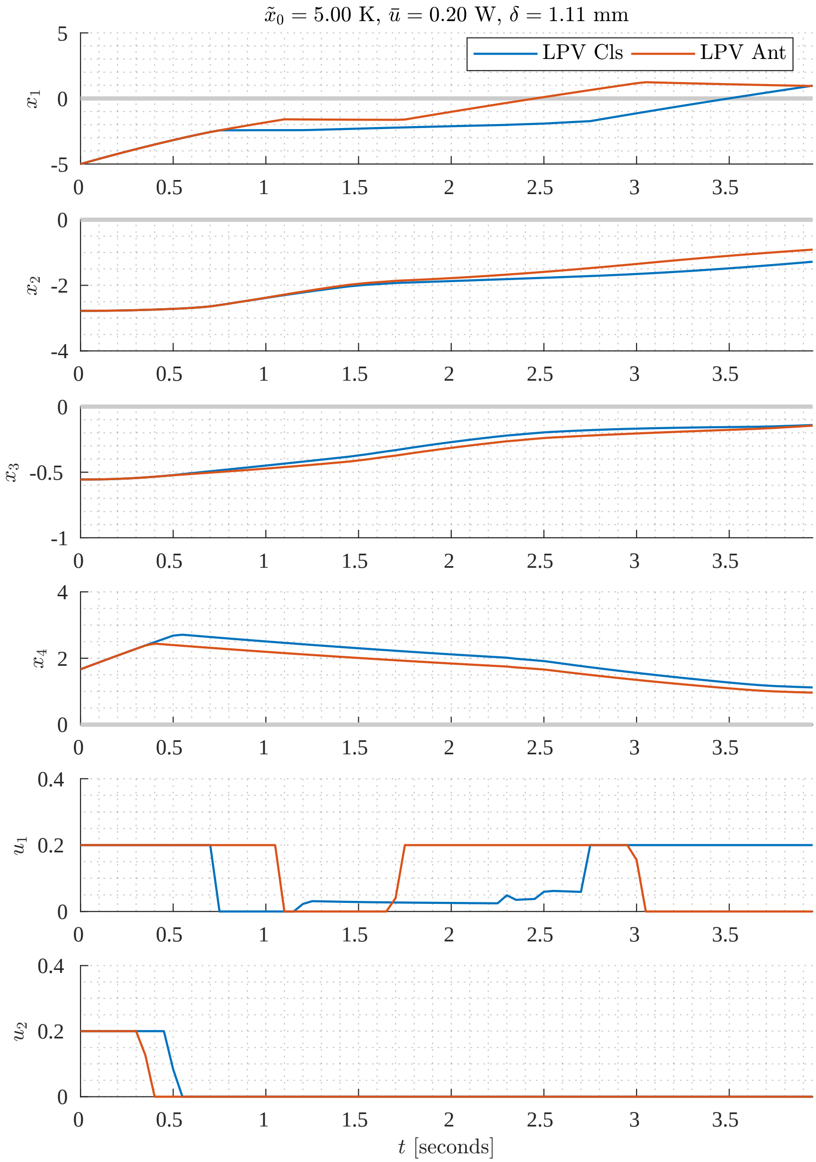


Figure 9.11: Closed-loop state and input trajectories for one particular setting of $(\tilde{x}_0, \bar{u}, \delta)$, with endpoint actuation.

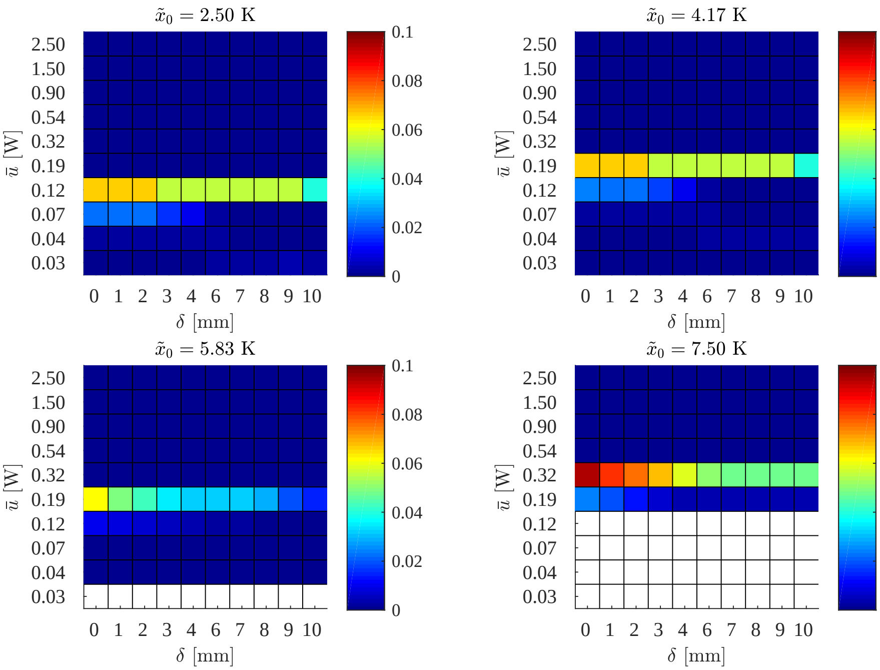


Figure 9.12: Relative differences η between realized closed-loop costs in the "classical" and "anticipative" cases with full actuation, for various combinations of input constraint size $\bar{u}$ and size of uncertainty δ in the future IH positions, and for different initial states $\tilde{x}_0$. A white square indicates that at least one of the tube synthesis problems was infeasible for the given combination $(\tilde{x}_0, \bar{u})$, so that the corresponding η value could not be computed.

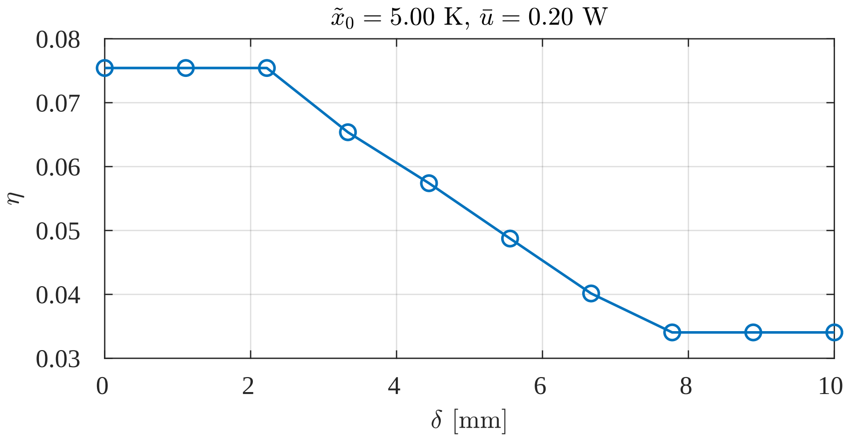


Figure 9.13: Relative difference η in closed-loop cost as function of δ, for fixed initial state $\tilde{x}_0$ and input constraint $\bar{u}$. (Full actuation)

controller ignores any uncertainty in the future IH trajectory, and assumes at every sampling instant k that

$$\forall i \in [1..N-1]: \; p(k+i) = \bar{p}(k+i)$$

where $\bar{p}(\cdot)$ is the nominal trajectory generated by (9.14). This leads to the loss of theoretical guarantees on recursive feasibility and stability because the actual IH trajectory is generated by (9.15), but can still be attractive for a practical application due to the reduced computational load.

Simulated closed-loop trajectories, including the non-anticipative ("classical") LPV, anticipative LPV, oracle LTV, and nominal LTV controllers, are shown in Figure 9.16. Only the trajectories for the first state variable and first control input are shown, because these are the signals where the difference between the four approaches is most clearly seen. Because the oracle LTV controller exploits the availability of the exact future IH positions (9.15), this represents the best possible achievable closed-loop performance. The trajectory with the nominal LTV controller is almost indistinguishable from this best possible trajectory, and in fact leads to the same realized closed-loop simulated cost over a simulation length of $N_{\text{sim}} = 120$ samples. The obtained simulated cost values and the corresponding η-values are presented in Table 9.4.

The solver run times for both LPV and LTV control are summarized in the box plot of Figure 9.17, showing the dramatic difference in complexity. It is, however, again emphasized that the much lower complexity of nominal LTV control comes at the price of losing theoretical guarantees on recursive feasibility and closed-loop stability—even though in this specific example, it performs very well.

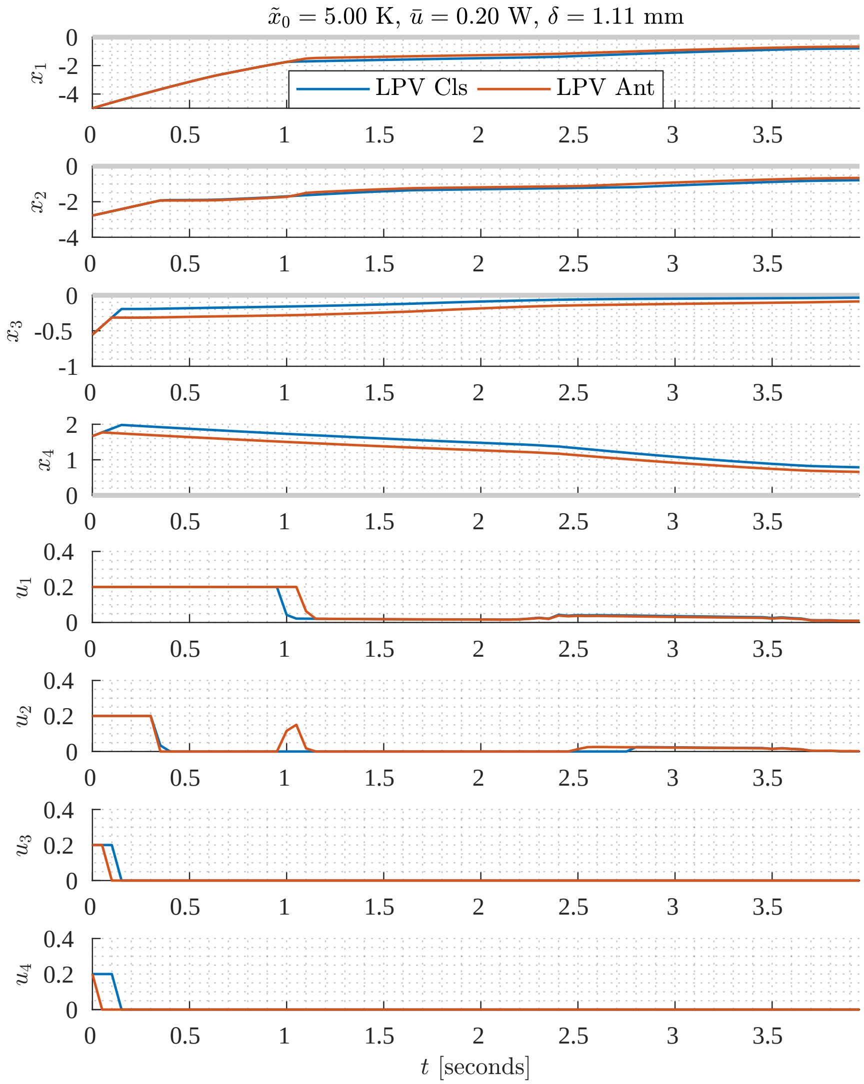


Figure 9.14: Closed-loop state and input trajectories for one particular setting of $(\tilde{x}_0, \bar{u}, \delta)$, with full actuation.

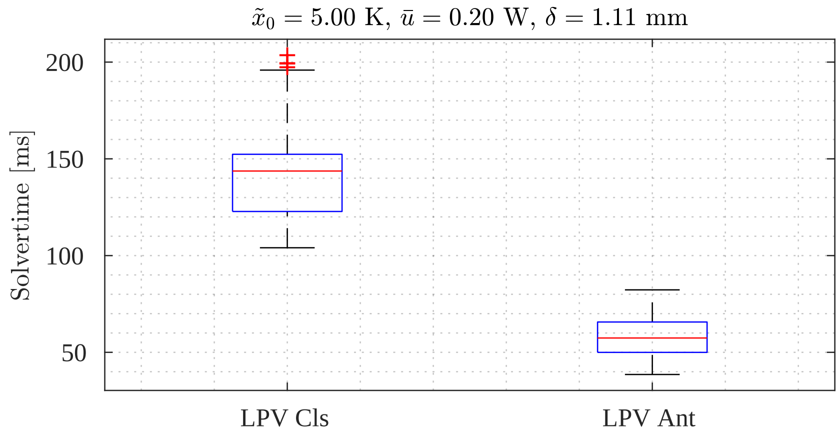


Figure 9.15: Illustration of solver run times per sample: comparison of non-anticipative LPV and anticipative LPV control. (Endpoint actuation)

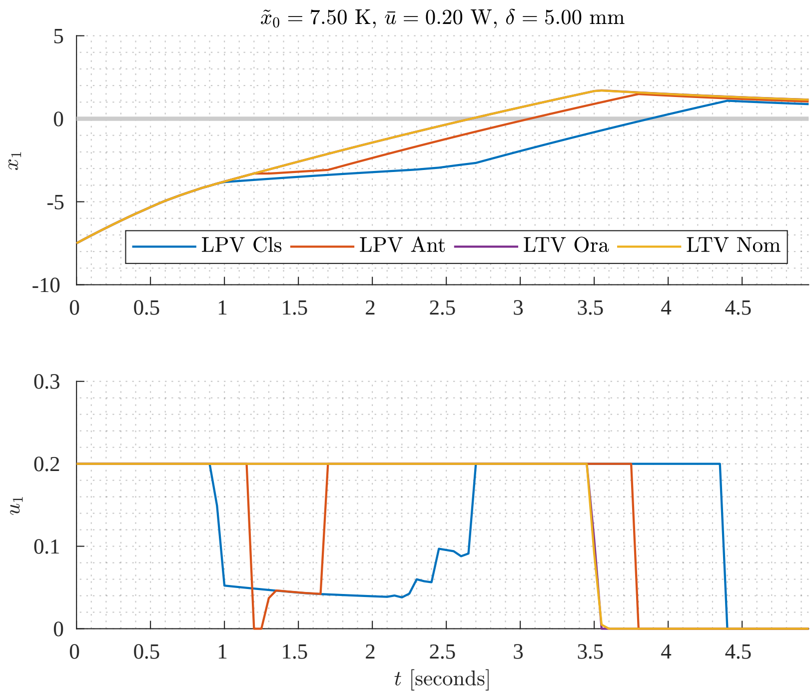


Figure 9.16: Closed-loop state and input trajectories for one particular setting of $(\tilde{x}_0, \bar{u}, \delta)$, including LTV control. The trajectories obtained with the "oracle LTV" and "nominal LTV" controllers are virtually identical. (Endpoint actuation)

Controller	J_{sim}	η
LPV Classical	320	0
LPV Anticipative (δ = 5 mm)	299	0.066
LTV Oracle	288	0.100
LTV Nominal	288	0.100

Table 9.4: Simulated cost values for different controllers. (Endpoint actuation)

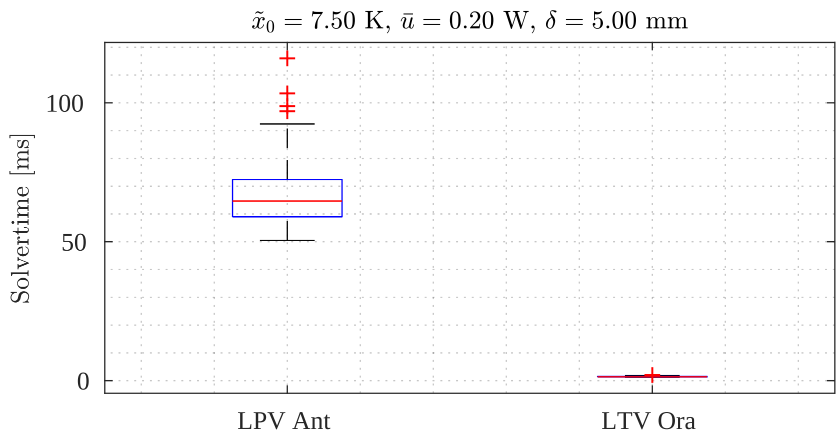


Figure 9.17: Illustration of solver run times per sample: comparison of LPV and LTV control. (Endpoint actuation)

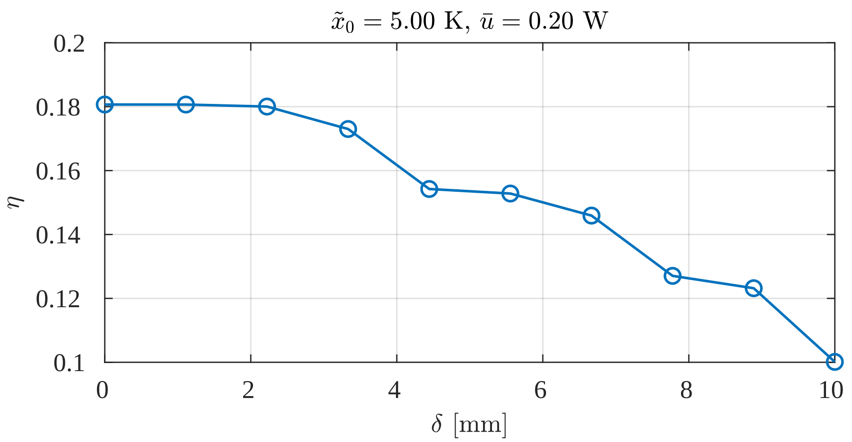


Figure 9.18: Relative difference η in closed-loop cost as function of δ, for fixed initial state $\tilde{x}_0$ and input constraint $\bar{u}$. (Endpoint actuation with k_{ih} multiplied by $\alpha_\kappa = 5$)

9.4.4 Effect of the IH heat transfer coefficient

The heat exchange between the wafer and the IH is modeled in (9.5). This heat flow is proportional to the constant κ ($\text{Wm}^{-1}\text{K}^{-1}$), which is the thermal conductivity of the wafer material. To study what is the effect of increasing the rate of heat transfer between the wafer and the IH, relation (9.5) is replaced by

$$\dot{Q}_{i,\text{ih}} = -\theta_i \alpha_\kappa \kappa \frac{wd}{h} T_i$$

where $\alpha_\kappa > 1$ is a scaling factor. A higher value of α means that the effect of scheduling variations on the system dynamics is increased.

It can be expected that the relative performance improvement of anticipative over non-anticipative control will increase as the influence of the scheduling variable on the system becomes more significant. To see if this is indeed the case, the values of η are computed as a function of δ for one choice of input constraint and initial state, and for $\alpha_\kappa = 5$. A simulation length of $N_{\text{sim}} = 100$ samples is used.

The result is shown in Figure 9.18. The realized values of η are higher than those obtained in Section 9.4.1 and Section 9.4.2. It can therefore be concluded that the relative benefit of anticipative control indeed becomes greater whenever the influence of the scheduling variable on the system dynamics is more severe.

9.5 Concluding remarks

In this chapter, the application of tube-based anticipative LPV MPC to a simplified thermal control problem was considered. It was found that, for the particular model parameters used, anticipative control could achieve a modest improvement in the realized closed-loop cost when considering a linear initial temperature distribution. For both the endpoint and full actuation cases, the magnitude of this improvement was found to be dependent on the size of the heater input constraint in relation to the initial state. This preliminary result indicates that, for the considered system, there exist some situations in which an anticipative MPC strategy can be beneficial with respect to a non-anticipative MPC strategy.

The principal limitation for applying the proposed anticipative control methods to a more accurate thermal model with a large number of elements M, is computational. In the tube-based MPC approach, a contractive terminal set is necessary to guarantee recursive feasibility and closed-loop stability. This same set has to be used as the tube cross section shape set. For this particular system, it was found that the M-dimensional hypercube is a suitable set for this purpose. Unfortunately, the complexity of the required vertex representation of this set grows as 2^M, limiting application of the approach to small values of M.

To overcome this computational challenge, for a practical thermal control application where the size of δ in (9.11) is small, it is suggested to consider LTV MPC based on the nominal trajectory $\bar{p}(k)$ instead of LPV MPC. By employing such an LTV strategy, theoretical guarantees on recursive feasibility and stability in the presence of uncertain future scheduling trajectories can not be given. Nonetheless, if the uncertainty is small, good performance can be achieved at a computational cost that is very much reduced with respect to "theoretically correct" LPV MPC.

Finally, it was illustrated that artificially increasing the effect of the scheduling variable on the system dynamics by scaling the IH heat transfer coefficient leads to greater relative differences between anticipative and non-anticipative control.

The results from this chapter are of a preliminary character and are meant to provide a basis for further work on the application of anticipative techniques to related thermal control problems. Some possible future research directions are suggested as follows:

- It could be possible that there exist (polytopic) contractive sets for a thermal system of the form (9.7) with a lower representation complexity in terms of the number of vertices. Finding such sets can be an objective of further applied research.
- The complexity-reduction approach of Chapter 6 could be considered to reduce the on-line computational complexity of the controller.
- In this chapter, an infinity-norm based cost function was used. For the thermal system, the effect of using a cost function based on a different norm (e.g., the one-norm) should be investigated.
- It is emphasized that the results presented in this chapter are valid for one particular set of parameters (number of elements, material properties, dimensions, controller tuning, IH trajectory, ...) and the results can be different for other values of these parameters. Therefore, it is recommended to study if comparable results are achieved when considering different and perhaps more realistic model parameters.

Chapter 10

Conclusion

THIS FINAL CHAPTER presents an overview of the research that was reported in this thesis. Its main contributions are highlighted, and based on these results some possible directions for future work are suggested.

10.1 Overview

The principal goal of the research presented in this thesis was to provide an answer to the following main research question:

> Q_0. How to design a stable and tractable LPV MPC strategy for regulation and tracking that can exploit knowledge about the possible future evolution of the scheduling variable?

In order to come to a satisfactory answer to Question Q_0, and based on the overview of the current literature in Section 1.3, several subquestions were proposed in Section 1.4. These were addressed separately in order to formulate a solution that addresses Question Q_0. These subquestions are now recalled and the developed answers—which together make up the contributions of the thesis—are summarized.

How to construct a practically useful description of the uncertain future evolution of the scheduling variable?

Question Q_1 was addressed in Chapter 3. It was proposed to describe possible future scheduling trajectories in terms of a sequence of sets. The notion of "anticipative" control was introduced to describe a predictive controller that can exploit this future information to possibly improve its achievable control performance. It was shown that the paradigm of tube-based model predictive control (MPC) offers the potential to construct anticipative controllers for linear parameter-varying (LPV) systems. As the main contribution of this chapter, this has led to the development of a framework for anticipative predictive control of LPV systems with a computational complexity that scales linearly in N.

How to use the model developed as the answer to Question Q_1 in a stable LPV MPC strategy for stabilization?

Question Q_2 was considered in Chapter 3 and, in a more general fashion, in Chapter 4. In Chapter 3, a tube-based MPC algorithm for LPV systems was presented based on the construction of so-called homothetic tubes. To guarantee recursive feasibility and closed-loop stability, a controlled λ-contractive set was employed.

In Chapter 4, it was proposed to relax the requirement that the terminal set must be λ-contractive by considering finite-step contractive sets instead. Thus, the main contribution of Chapter 4 has been the development of more general stability conditions for tube-based MPC for LPV systems based on the notion of finite-step contraction. The development of efficient methods for constructing finite-step contractive sets for LPV systems is a possible topic of future research.

Also, in Chapter 9, it was shown how the developed description of possible future scheduling trajectories can be used in an MPC to stabilize a thermal system. Under some circumstances, the use of "anticipative" control can improve the realized closed-loop performance. Although the results were obtained on a highly simplified model, the presented work could serve as a basis for further research on the application of anticipative control techniques to thermal problems.

Is it possible to introduce an extra degree-of-freedom in the design of the tube parameterization?

One answer to Question Q_3 was proposed in Chapter 5. The main contribution of this chapter has been the development of a heterogeneously parameterized tube (HpT) framework that allows the combination of different cross-section parameterization in a single tube. The choice of parameterizations can be considered as an additional degree of freedom (DOF) in the overall tube design. It was shown that this extra design DOF has the potential to improve the trade-off between computational complexity and size of the domain of attraction (DOA) of the resulting controllers. Topics for future research include the optimal design of heterogeneous parameterization structures, the implementation of different cross sections parameterizations, and further improvements to computational efficiency.

An alternative answer to Question Q_3 was presented in Chapter 6. The contribution of this chapter has been the development of a computational method that allows for an off-line design of the tube parameterization, such that the resulting controller is guaranteed to be feasible on a pre-specified set of initial conditions. An open problem that can be addressed in the future is the elimination of redundant constraints in addition to decision variables. Further, the effect of the used linear program (LP) optimization algorithm on the computational performance of the proposed method should be studied in more detail, and the effect of the parameterization on the realized closed-loop cost should be considered.

How can the developed strategies for "stabilization" be adapted to the "tracking" case?

Question Q_4 was considered in Chapter 7. The contribution of this chapter was the development of a tube-based MPC strategy for tracking constant references based on so-called bounded-error invariant sets. The proposed method guarantees an a-priori specified bound on the tracking error in the face of all possible variations of the scheduling signal. Topics for future work include studying the possibility of guaranteeing offset-free tracking if the scheduling variable stops varying, enlarging the DOA, and extending the method to handle a larger class of references such as those generated by so-called reference governors.

How to apply the developed LPV MPC approaches to the control of LPV embeddings?

Research Question Q_5 was the last one to be addressed in Chapter 8. The feature of the approach is the bounding of predicted closed-loop state trajectories by a sequence of state constraints. This enables the computation of a corresponding sequence of scheduling sets that are valid along all possible realized closed-loop trajectories, which can be used in an anticipative LPV MPC approach. As the contribution of this chapter, this gives a predictive control approach for non-linear systems described in terms of an LPV embedding with guaranteed recursive feasibility and closed-loop stability. The main questions that should be addressed in the future are how to systematically select the initial state constraint sequence and how to efficiently compute the corresponding scheduling sets. Studying the possibility of proving recursive feasibility under less restrictive state constraint conditions is also of interest.

10.2 Main contributions and future work

Based on the overview given in Section 10.1, the main contributions of this thesis can be succinctly listed as follows:

- Introduced a framework for "anticipative" control for LPV systems with a complexity that scales linearly in the prediction horizon N (Chapter 3).
- Developed new stability conditions for LPV tube MPC based on finite-step contraction (Chapter 4). This presents a relaxation of the usual stability conditions in tube-based MPC based on "normal" controlled contractive sets.
- Proposed new "heterogeneous" and "sparse" tube parameterizations (Chapter 5 and Chapter 6). Both of these approaches provide new freedoms in the design of tube parameterizations, which can be exploited to achieve different and more favorable trade-offs between computational complexity and achieved control performance as measured in terms of the size of the DOA.
- Created an opening towards the development of a constrained LPV reference tracking approach (Chapter 7). In contrast to previously considered approaches, the proposed

method guarantees an a-priori specified bound on the tracking error for all possible variations of the scheduling signal.

- Presented a new approach for the constrained control of non-linear systems represented by an LPV embedding (Chapter 8). The main novel feature of the method is the integration of sequences of constraint sets that are used to bound possible future scheduling trajectories by means of a scheduling map: in this way, it becomes possible to guarantee closed-loop stability with the LPV MPC controller and the original non-linear system.

Together, these contributions form the elements of an answer to Research Question Q_0. Naturally, this answer is not final or even complete, and there is ample opportunity for extension and improvement. Specific suggestions for future work relating to the individual research questions were already provided at the ends the corresponding chapters, and these were also summarized in the overview of Section 10.1. Some more general pointers for future research are given below:

- Several chapters in this thesis have introduced new design DOFs in tube parameterizations. However, it was not always completely explored how this new freedom can be exploited to the fullest possible advantage. This holds true particularly for the developments of Chapter 4 (how to compute S_0), Chapter 5 (how to choose the heterogeneous parameterization structure), and Chapter 8 (how to pick the initial state constraint sets). Hence, it is suggested to develop procedures that can systematically exploit the new design freedoms to obtain controllers with favorable complexity/performance trade-offs. A first idea for this is to consider evolutionary optimization methods (such as PSO, which is employed in Appendix A) to design heterogeneous tube parameterizations.

- With respect to cross section parameterizations, only the homothetic case was considered for implementation, even though the developed theoretical results allow for other possibilities. These include elastic tubes, and perhaps novel parameterizations yet to be invented. It is therefore suggested to pursue the implementation of different cross section parameterizations in the tube-based anticipative LPV MPC setting.

- In Chapter 9, the anticipative control approach based on homothetic tubes was used on a simple thermal model. To study further the applicability and limitations of the proposed methods, it is suggested to implement them on more case studies. Besides the homothetic tube-based approach of Section 3.5, these studies should include the periodic, heterogeneous, and sparse tubes as well.

- A disadvantage of the developed approaches is that the focus on theoretical correctness (guaranteed recursive feasibility and closed-loop stability) leads to high computational burdens. In particular, the complexity of the developed algorithms scales badly in the state dimension. For systems with more than three states, the overall computational load is high despite the formulation as an LP with linear scaling in N. Computational complexity has been a longstanding issue in robust and LPV MPC that has not yet been resolved in a completely satisfactory manner. Therefore, any new research in the broad area of MPC for LPV systems—not necessarily restricted to tube-based methods—is always to be encouraged.

It is hoped that this thesis will inspire more research on anticipative model predictive control for linear parameter-varying systems, either following the above suggestions or along novel—“unanticipated”—lines.

Appendix A

Synthesizing finite-step contractive sets with PSO

This appendix describes a method, based on particle swarm optimization (PSO), for the computation of (polytopic and symmetric) finite-step contractive sets. The complexity of the first set in the sequence is fixed by specifying its number of vertices. The computed sets can be used in the tube model predictive control (TMPC) approach of Chapter 4. The method from this appendix has also been used in various other locations in the thesis to compute "normal" controlled λ-contractive sets with pre-specified complexity.

A.1 Problem setting

Consider a polytopic LPV state-space (LPV-PSS) representation of the form

$$x(k+1) = A(\theta(k))x(k) + Bu(k). \tag{A.1}$$

Furthermore let $(M, \lambda) \in [1, \infty) \times [0, 1)$ be given. The problem addressed in this appendix is to find a set, with the largest possible volume, that is controlled (M, λ)-contractive with respect to the system (A.1) (see Definition 4.1). This objective corresponds to

$$\begin{aligned} S^\star = \arg \max_{S \in 2^{\mathbb{R}^{n_x}}} \quad & \text{volume}\,(S) \\ \text{subject to} \quad & S \text{ satisfies Def. 4.1 w.r.t. (A.1) for the given } (M, \lambda). \end{aligned} \tag{A.2}$$

This problem is intractable, as the optimization is over arbitrary PC-sets in $\mathbb{R}^{n_x}$. Therefore, in this appendix, an approximate solution to (A.2) is proposed where the optimization is over symmetric polytopes. Even with this restriction, the problem (A.2) is difficult to solve because it involves the maximization of the volume of a polytope.

For this reason, it is proposed to employ a non-linear optimization method called particle swarm optimization (PSO) in order to find (approximate) solutions to (A.2). PSO is an unconstrained evolutionary optimization method that aims to find a local minimum of a problem of the form $\min_x f(x)$ (Kennedy and Eberhart 1995; Poli et al. 2007; Sandou 2009).

PSO was used previously for the computation of "normal" controlled contractive sets in (Munir et al. 2016). In this appendix, the more general case of controlled (M, λ)-contraction is considered. Furthermore, in (Munir et al. 2016) the polytopic sets are described by an H-representation, whereas the method from this appendix uses V-representations. This allows the computation of sets that are controlled contractive under a vertex control policy, whereas the H-representation used by (Munir et al. 2016) necessarily restricts attention to linear state feedback policies.

A.2 The algorithm

In this section, the proposed algorithm for computing finite-step contractive sets is presented. First, Section A.2.1 gives a more tractable approximation of (A.2) by restricting the class of considered sets to symmetric polytopes. Next, in Section A.2.2, an approach for quantifying "how close" a given set of the form (A.3) is to being controlled (M, λ)-contractive is developed in terms of an elastic linear program (LP). This LP is subsequently used as a constraint test in the overall optimization algorithm, which is presented in Section A.2.3. A few remarks on the required polytope volume computations are provided in Section A.2.4.

A.2.1 Set parameterization

To render the problem (A.2) tractable, the optimization is restricted to symmetric polytopes of the form

$$S(\mathbf{v}) = \operatorname{convh}\left\{v_1, \dots, v_{\frac{1}{2}n_\mathrm{v}}, -v_1, \dots, -v_{\frac{1}{2}n_\mathrm{v}}\right\} \tag{A.3}$$

where $\frac{1}{2}n_\mathrm{v}$ is a design parameter fixing the number of vertices of $S(\mathbf{v})$. The decision variable $\mathbf{v}$ is therefore

$$\mathbf{v} = \left(v_1, v_2, \dots, v_{\frac{1}{2}n_\mathrm{v}}\right) \in \mathbb{V} = \mathbb{R}^{\frac{1}{2}n_\mathrm{v}n_\mathrm{x}}.$$

By restricting the sets $S(\mathbf{v})$ to be symmetric, it is ensured that they contain the origin and are therefore PC-sets by construction.

Conceptually it is also possible to consider a simpler parameterization of $S(\mathbf{v})$ leading to a lower-dimensional decision variable $\mathbf{v}$. For instance, the set $S(\mathbf{v})$ can be parameterized as being a linearly transformed version of some given polytope.

Given the parameterization (A.3), the following optimization problem can be formulated as a more tractable approximation of (A.2):

$$\begin{aligned} \mathbf{v}^\star = \arg\max_{\mathbf{v}\in\mathbb{V}} \quad & \operatorname{volume}\left(S(\mathbf{v})\right) \\ \text{subject to} \quad & S(\mathbf{v}) \text{ satisfies Def. 4.1 w.r.t. (A.1) for the given } (M, \lambda). \end{aligned} \tag{A.4}$$

This optimization problem is still not convex because it involves maximization of the volume of a polytope. In the sequel, it is shown how (A.4) can be formulated in a way such that PSO can be used to find approximate solutions (i.e., local maxima).

A.2.2 Verifying finite-step contraction

In this subsection, the question of deciding if a given PC-set S is controlled (M, λ)-contractive is formulated in terms of an elastic LP. Verifying if a set $S(\mathbf{v})$ of the form (A.3) is (M, λ)-contractive can be done by checking the conditions of the following lemma, which is a straightforward consequence of Definition 4.1:

Lemma A.1. *Let $S(\mathbf{v})$ be given as in* (A.3)*. Define $S_0 = S(\mathbf{v})$, and for $i \in [0..M-2]$ let $S_{i+1} = \mathcal{I}(S_i, \Theta | K_i)$ where $K_i : S_i \times \Theta \to \mathbb{U}$ are control laws. The set $S(\mathbf{v})$ is controlled (M, λ)-contractive if there exists $K_i : S_i \times \Theta \to \mathbb{U}$ such that $\forall i \in [0..M-1] : \{0\} \subseteq S_i \subseteq \mathbb{X}$ and $S_{M-1} \subseteq \lambda S_0$.*

By parameterizing the controllers $K_i(\cdot,\cdot)$ as vertex controllers corresponding to the sets S_i, the conditions of Lemma A.1 can be verified by solving a single large LP. Equivalently, the conditions can be verified by solving n_v smaller LPs: one for each vertex of $S(\mathbf{v})$. In what follows, the latter approach is chosen, i.e., Lemma A.1 is verified by solving n_v LPs of the general form

$$\begin{aligned} \lambda^\star(v) = \min_{d, \tilde{\lambda}} \quad & \tilde{\lambda}, \\ \text{subject to} \quad & A_\mathrm{eq}(v) d = b_\mathrm{eq}(v), \\ & A_\mathrm{ineq}(v) d \leq b_\mathrm{ineq}(v), \end{aligned} \tag{A.5}$$

where $v \in \mathbb{X}$ is a vertex of $S(\mathbf{v})$, while $d \in \mathbb{R}^{n_\mathrm{d}}$ and $\tilde{\lambda} \in \mathbb{R}_+$ are decision variables. The precise construction of the LP (A.5) is straightforward and is not discussed here in detail: it is done along the same line as constructing the LP for the "scenario tube" from Chapter 5 or for the scenario trees in (Muñoz de la Peña et al. 2005; Scokaert and Mayne 1998). The complexity of the LP (A.5) is exponential in M. If for every vertex v of $S(\mathbf{v})$ it is true that $\lambda^\star(v) \leq \lambda$, then $S(\mathbf{v})$ is controlled (M, λ)-contractive. Else, if $\lambda^\star(v) > \lambda$ for some v, the value of $\lambda^\star(v)$ can provide a measure of how close $S(\mathbf{v})$ is to being controlled finite-step contractive.

However, if for the given initial set $S(\mathbf{v})$ it is not possible to satisfy all of the constraints $\{0\} \subseteq S_i \subseteq \mathbb{X}$, or if the input constraints can not be satisfied by the controllers $K_i(\cdot,\cdot)$, then no useful information can be extracted from the infeasibility of (A.5). To remedy this situation, an elasticized version of (A.5) can be used (Chinneck 2008, Chapter 6). Introduce a vector of non-negative slack variables $[\, s^\top \; t \,]^\top$ and consider the elasticized LP

$$\begin{aligned} \epsilon(v) = \min_{d, \tilde{\lambda}, s, t} \quad & 1^\top \begin{bmatrix} s \\ t \end{bmatrix}, \\ \text{subject to} \quad & A_\mathrm{eq}(v) d = b_\mathrm{eq}(v), \\ & A_\mathrm{ineq}(v) d - s \leq b_\mathrm{ineq}(v), \\ & \lambda - t < 1, \\ & s \geq 0, t \geq 0, \end{aligned} \tag{A.6}$$

where again $v \in \mathbb{X}$ is a vertex of $S(\mathbf{v})$. The equality constraints describe the system dynamics, and therefore must not be elasticized. Next, define

$$\bar{\epsilon}(\mathbf{v}) = \sum_{i=1}^{\frac{1}{2}n_{\mathrm{v}}} \epsilon(v_i) + \epsilon(-v_i),$$

i.e., the sum of all $\epsilon(v)$ for every vertex of $S(\mathbf{v})$. If $\bar{\epsilon}(\mathbf{v}) = 0$, it is known that $S(\mathbf{v})$ is $(M, \tilde{\lambda})$-contractive for some $\tilde{\lambda} \in [0, \lambda)$. Otherwise, the magnitude of $\bar{\epsilon}(\mathbf{v})$ provides a measure of "how close" $S(\mathbf{v})$ is to being at least (M, λ)-contractive in the sense of Definition 4.1. The problem (A.4) is now equivalent to

$$\begin{aligned} \mathbf{v}^{\star} = \arg\max_{\mathbf{v} \in \mathbb{V}} \quad & \text{volume}\left(S(\mathbf{v})\right) \\ \text{subject to} \quad & \bar{\epsilon}(\mathbf{v}) = 0. \end{aligned} \tag{A.7}$$

The introduction of the quantity $\bar{\epsilon}(\mathbf{v})$ allows (A.7) to be approximately solved using PSO, as will be shown next.

A.2.3 Main result

The problem (A.7) is subject to an equality constraint. In PSO, constraints need to be dealt with by augmenting the cost function (Sedlaczek and Eberhard 2006). The idea is to construct the cost in such a way that constraint violations are heavily penalized, such that minimizing it tends to result in feasible solutions. In this appendix, for this purpose, a discontinuous cost function

$$J(\mathbf{v}) = \begin{cases} -\text{volume}\left(S(\mathbf{v})\right), & \bar{\epsilon}(\mathbf{v}) = 0, \\ \bar{\epsilon}(\mathbf{v}), & \text{otherwise}, \end{cases} \tag{A.8}$$

is employed. This leads to the final optimization problem

$$\mathbf{v}^{\star} = \arg\min_{\mathbf{v} \in \mathbb{V}} J(\mathbf{v}) \tag{A.9}$$

to be solved using PSO. The cost function (A.8) is always positive when the constraint $\bar{\epsilon}(\mathbf{v})$ is violated, and negative otherwise. Therefore, by minimizing $J(\mathbf{v})$ the PSO algorithm tends to produce feasible solutions while maximizing the volume of $S(\mathbf{v})$ at the same time. If the PSO algorithm terminates with a negative objective function value, then it is guaranteed that the set $S(\mathbf{v}^{\star})$ is controlled (M, λ)-contractive.

In PSO, more sophisticated ways of constructing a cost function that penalizes constraint violations are available, e.g., the augmented Lagrangian method of (Sedlaczek and Eberhard 2006). However, in numerical trials the performance of the simple cost function (A.8) was found to be satisfactory.

Lastly, it is interesting to note that evaluating the objective function (A.8) involves solving n_{v} LPs and computing the volume of a polytope. One of the attractive features of PSO for this application is that it can deal with this type of complex (no closed expression available, non-differentiable, discontinuous) cost and constraint functions without particular challenges.

It is however not a-priori guaranteed that a feasible solution to (A.9) will be found, and in general the returned solutions will not be globally optimal.

Once a set $S_0 = S(\mathbf{v}^\star)$ that is (M, λ)-contractive is found, the rest of the sets in the sequence $\{S_0, \ldots, S_{M-1}\}$ (see Lemma A.1 and Definition 4.1) can be extracted from the solution of (A.6) or (A.5). The number of vertices of the other sets in the sequence grows exponentially in principle, but in practice it is often found that many of the generated candidate vertices are redundant and can be eliminated; compare the similar situation in (Athanasopoulos and Lazar 2014).

A.2.4 Note on volume computation

The exact volume of a polytope represented by its vertices can be computed, e.g., using a triangulation-based algorithm (Bueler et al. 2000). Such a method is implemented in MPT (Herceg et al. 2013) and works well in relatively low dimensions.

In higher dimensions, it is proposed to replace the exact computation of volume$(\cdot)$ by a function $\widehat{\text{volume}}(\cdot)$, such that maximizing $\widehat{\text{volume}}(\cdot)$ approximately maximizes volume$(\cdot)$ as well. A common approximation employed for this purpose involves computing the maximum inscribed ellipsoid (Boyd and Vandenberghe 2004, Section 8.4.2). Unfortunately, this is a non-convex problem if the polytope is described by its V-representation, as is the case in this appendix. This can be overcome by computing the equivalent H-representation first, but in higher dimensions this is a hard problem by itself (Fukuda 2004).

An alternative approximation that can be used to maximize volume$(S(\mathbf{v}))$ that does lead to a convex problem is

$$\widehat{\text{volume}}\,(S(\mathbf{v})) = \arg\max_{\alpha \in \mathbb{R}_+} \alpha \text{ subject to } \alpha X \subseteq S(\mathbf{v}),$$

where $X = \text{conv}\{\bar{x}^1, \ldots, \bar{x}^{q_x}\}$ is a pre-designed polytope: X could be selected as, e.g., a norm ball in the one- or infinity-norms.

Appendix B

Estimating domains of attraction

The purpose of this appendix is to explain the methods that were used throughout this thesis to estimate domains of attraction (DOAs) of the developed tube-based predictive controllers.

B.1 Problem setting

Recall from (3.11) that the DOA corresponding to a tube-based controller is defined as

$$\mathcal{X}_N(\boldsymbol{\Theta}) = \{x \in \mathbb{X} \mid \mathcal{T}_N(x, \boldsymbol{\Theta}) \neq \emptyset\} \tag{B.1}$$

where $\boldsymbol{\Theta} \subseteq \Theta^N$ is a sequence of scheduling sets. An exact representation of the DOA can, in principle, be found by solving a multi-parametric linear program (MPLP) (Herceg et al. 2013; Pistikopoulos et al. 2011). However, the linear programs (LPs) encountered in tube model predictive control (TMPC) are typically too large for this to be computationally feasible. Therefore, computing an exact representation of the DOA is usually not possible, and it must be approximated instead.

This appendix presents two approaches that have been used to compute inner approximations of the domains of attraction for the controllers described in this thesis. One approach is based on gridding and a search procedure (Section B.2), whereas the other directly scales the vertices of a given set to approximate the DOA (Section B.3).

In this appendix, attention is focused on approximation of the worst-case domain of attraction (WC-DOA), defined as

$$\mathcal{X}_N^{\mathrm{wc}} = \mathcal{X}_N\left(\Theta^N\right). \tag{B.2}$$

Further, define

$$\boldsymbol{\Theta}(\theta) = \{\{\theta\}, \Theta, \ldots, \Theta\} \subseteq \Theta^N,$$

and assume that, as usual, the scheduling set $\Theta \subseteq \mathbb{R}^{n_\theta}$ is given by

$$\Theta = \operatorname{convh}\left\{\bar{\theta}^1, \ldots, \bar{\theta}^{q_\theta}\right\}.$$

B.2 Approach 1: gridding and testing

The method is described in Algorithm B.1. The idea of the algorithm is simple: starting with an initial grid of points in the state space, scale each point such that a feasible tube can be found when the scaled point is used as the initial state. Its main advantage is that to apply it, no changes have to be made to the underlying tube synthesis problem (see Equation (3.8)).

Increasing the density of the initial grid $\mathcal{L}_{\text{init}}$ can lead to tighter estimates of the true WC-DOA, at the cost of a longer computation time.

Algorithm B.1 The WC-DOA approximation algorithm based on gridding.

Require: A discrete set of M points $\mathcal{L}_{\text{init}} = \{x_0, \ldots, x_{M-1}\} \subseteq \mathbb{X}$
Require: A step size $\delta \in (0, 1]$
1: $\hat{\mathcal{X}}_N^{\text{wc}} \leftarrow \mathbb{X}$
2: **for all** $i \in [1..q_\theta]$ **do**
3: $\quad \hat{\mathcal{X}}_N^{i} \leftarrow$ DOA-SINGLE-THETA$(\bar{\theta}^i)$ ▹ See Line 6
4: $\quad \hat{\mathcal{X}}_N^{\text{wc}} \leftarrow \hat{\mathcal{X}}_N^{\text{wc}} \cap \hat{\mathcal{X}}_N^{i}$
5: **end for**
6: **function** DOA-SINGLE-THETA(θ)
7: $\quad \mathcal{L}_{\text{infeas}} \leftarrow \mathcal{L}_{\text{init}}$
8: $\quad \mathcal{L}_{\text{feas}} \leftarrow \{\}$
9: $\quad$ **while** $\mathcal{L}_{\text{infeas}} \neq \emptyset$ **do**
10: $\quad\quad x \leftarrow$ last element from $\mathcal{L}_{\text{infeas}}$
11: $\quad\quad$ **for all** $j \in [0..\lceil \frac{1}{\delta} \rceil]$ **do**
12: $\quad\quad\quad \alpha \leftarrow \max\{1 - j\delta, 0\}$
13: $\quad\quad\quad$ **if** $\mathcal{T}_N(\alpha x, \mathbf{\Theta}(\theta)) \neq \emptyset$ **then**
14: $\quad\quad\quad\quad \mathcal{L}_{\text{feas}} \leftarrow \mathcal{L}_{\text{feas}} \cup \{\alpha x\}$
15: $\quad\quad\quad\quad \mathcal{L}_{\text{infeas}} \leftarrow \mathcal{L}_{\text{infeas}} \setminus \{x\}$
16: $\quad\quad\quad\quad$ **break** ▹ From the *for all*-loop on Line 11
17: $\quad\quad\quad$ **end if**
18: $\quad\quad$ **end for**
19: $\quad$ **end while**
20: $\quad$ **return** convh $\{\mathcal{L}_{\text{feas}}\}$
21: **end function**

Note that Algorithm B.1 is guaranteed to terminate, because the tube synthesis problem is always feasible for the initial state $x = 0$ (i.e., $\mathcal{T}_N(0, \mathbf{\Theta}) \neq \emptyset$ for any $\mathbf{\Theta} \subseteq \Theta^N$).

B.3 Approach 2: scaling

Consider a point set $\hat{\mathcal{X}}_N^{\text{init}} \subseteq \mathbb{R}^{n_x}$ represented as

$$\hat{\mathcal{X}}_N^{\text{init}} = \left\{\bar{x}^1, \ldots, \bar{x}^{q_x}\right\}.$$

It is assumed that $\operatorname{convh}\{\hat{\mathcal{X}}_N^{\text{init}}\}$ is a PC-set. Then, for each $i \in [1..q_{\text{x}}]$, compute the scalings

$$\alpha_i = \max_{\alpha \in \mathbb{R}_+} \alpha \text{ subject to } \mathcal{T}_N\left(\alpha \bar{x}^i, \Theta^N\right) \neq \emptyset \tag{B.3}$$

and construct the inner approximation of the WC-DOA as

$$\hat{\mathcal{X}}_N^{\text{wc}} = \operatorname{convh}\left\{\alpha_1 \bar{x}^1, \ldots, \alpha_{q_{\text{x}}} \bar{x}^{q_{\text{x}}}\right\}. \tag{B.4}$$

It is noted that the problems (B.3) are equivalent to

$$\begin{aligned}
\alpha_i &= \max_{\alpha \in \mathbb{R}_+} \alpha \text{ subject to } \forall j \in [1..q_\theta] : \mathcal{T}_N\left(\alpha \bar{x}^i, \boldsymbol{\Theta}(\bar{\theta}^j)\right) \neq \emptyset \\
&= \min_{j \in [1..q_\theta]} \left(\max_{\alpha \in \mathbb{R}_+} \alpha \text{ subject to } \mathcal{T}_N\left(\alpha \bar{x}^i, \boldsymbol{\Theta}(\bar{\theta}^j)\right) \neq \emptyset\right).
\end{aligned}$$

In other words, for each $i \in [1..q_{\text{x}}]$, a solution to (B.3) can be found by solving q_θ LPs. Therefore, constructing the approximation (B.4) requires the solution of $q_{\text{x}} q_\theta$ LPs.

To apply this method, the tube synthesis problem needs to be changed in order to maximize the objective function (B.3). This is in contrast to the “gridding” method from Section B.2, which can be applied without any modifications to the tube synthesis problem (but may require more computational effort).

Bibliography

Abbas, H. S., A. Ali, S. M. Hashemi, and H. Werner (2013). “LPV Gain-Scheduled Control of a Control Moment Gyroscope”. In: pp. 6841–6846 (cit. on p. 11).

Abbas, H. S. et al. (2014). “Embedding of Nonlinear Systems in a Linear Parameter-Varying Representation”. In: *Proc. of the 19th IFAC World Congress*, pp. 6907–6913 (cit. on pp. 9, 46, 158).

Abbas, H. S. et al. (2015). “An MPC Approach for LPV Systems in Input-Output Form”. In: *Proc. of the 54th IEEE Conference on Decision and Control*, pp. 91–96 (cit. on p. 242).

Abbas, H. S. et al. (2016). “A Robust MPC for Input-Output LPV Models”. In: *IEEE Transactions on Automatic Control* 61, pp. 4183–4188 (cit. on p. 241).

Abbas, H. S. et al. (2017). “LPV Modeling of Nonlinear Systems: A Multi-Path Feedback Linearization Approach”. In: *Submitted to IEEE Transactions on Automatic Control* (cit. on p. 158).

Abbas, H. S. et al. (2018a). “A New Approach to Robust MPC Design for LPV Systems in Input-Output Form”. In: *Accepted for presentation at the joint 9th IFAC Symposium on Robust Control Design and 2nd IFAC Workshop on LPV Systems* (cit. on p. 241).

Abbas, H. S. et al. (2018b). “An Improved Robust Model Predictive Control for Linear Parameter-Varying Input-Output Models”. In: *Int. J. of Robust and Nonlinear Control* 28, pp. 859–880 (cit. on p. 241).

Abelson, H., G. J. Sussman, and J. Sussman (1996). *Structure and Interpretation of Computer Programs*. 2nd ed. MIT Press (cit. on p. 35).

Aeyels, D. and J. Peuteman (1998). “A New Asymptotic Stability Criterion for Nonlinear Time-Variant Differential Equations”. In: *IEEE Transactions on Automatic Control* 43, pp. 968–971 (cit. on pp. 40, 78).

Alessio, A. and A. Bemporad (2009). “A Survey on Explicit Model Predictive Control”. In: *Nonlinear Model Predictive Control. Towards New Challenging Applications*. Ed. by L. Magni, D. M. Raimondo, and F. Allgöwer. Springer, pp. 345–369 (cit. on p. 20).

Althoff, M. and D. Grebenyuk (2016). “Implementation of Interval Arithmetic in CORA 2016”. In: *Proc. of the 3rd International Workshop on Applied Verification for Continuous and Hybrid Systems*, pp. 91–105 (cit. on p. 170).

Althoff, M., O. Stursberg, and M. Buss (2008). "Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization". In: *Proc. of the 47th IEEE Conference on Decision and Control*, pp. 4042–4048 (cit. on pp. 167–170).

Alvarado, I. et al. (2007). "Robust tube based MPC for tracking of piece-wise constant references". In: *Proc. of the 46th IEEE Conference on Decision and Control*, pp. 1820–1825 (cit. on pp. 25, 143).

Anderson, B. D. O. and J. B. Moore (1989). *Optimal Control: Linear Quadratic Methods*. Prentice Hall (cit. on p. 38).

Angeli, D., A. Casavola, and E. Mosca (2000). "Constrained predictive control of nonlinear plants via polytopic linear system embedding". In: *Int. J. of Robust and Nonlinear Control* 10, pp. 1091–1103 (cit. on p. 174).

Artstein, Z. and S. V. Raković (2011). "Set invariance under output feedback: A set-dynamics approach". In: *Int. J. of Systems Science* 42, pp. 539–555 (cit. on p. 71).

Athanasopoulos, N. and G. Bitsoris (2010). "Invariant set computation for constrained uncertain discrete-time linear systems". In: *Proc. of the IEEE Conference on Decision and Control*, pp. 5227–5232 (cit. on p. 70).

Athanasopoulos, N., A. I. Doban, and M. Lazar (2013). "On Constrained Stabilization of Discrete-Time Linear Systems". In: *Proc. of the 21st Mediterranean Conference on Control & Automation*, pp. 831–839 (cit. on p. 70).

Athanasopoulos, N. and M. Lazar (2013). "Scalable Stabilization of Large Scale Discrete-Time Linear Systems via the 1-Norm". In: *Proc. of the 4th IFAC Workshop on Distributed Estimation and Control in Networked Systems*, pp. 277–284 (cit. on p. 84).

Athanasopoulos, N. and M. Lazar (2014). "Alternative Stability Conditions for Switched Discrete Time Linear Systems". In: *Proc. of the 19th IFAC World Congress*, pp. 6007–6012 (cit. on pp. 84, 217).

Bachnas, A. A., R. Tóth, J. H. A. Ludlage, and A. Mesbah (2014). "A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study". In: *Journal of Process Control* 24, pp. 272–285 (cit. on pp. 9, 11).

Bacic, M., M. Cannon, Y. I. Lee, and B. Kouvaritakis (2003). "General Interpolation in MPC and Its Advantages". In: *IEEE Transactions on Automatic Control* 48, pp. 1092–1096 (cit. on pp. 22, 23).

Bemporad, A., F. Borrelli, and M. Morari (2003). "Min-max Control of Constrained Uncertain Discrete-Time Linear Systems". In: *IEEE Transactions on Automatic Control* 40, pp. 1234–1236 (cit. on p. 20).

Bemporad, A. and M. Morari (1999). "Robust Model Predictive Control: A Survey". In: *Robustness in identification and control*. Ed. by A. Garulli, A. Tesi, and A. Vicino. Springer, pp. 207–226 (cit. on p. 20).

Ben-Tal, A., A. Goryashko, E. Guslitzer, and A. Nemirovski (2004). “Adjustable robust solutions of uncertain linear programs”. In: *Mathematical Programming* 99, pp. 351–376 (cit. on p. 24).

Bertsekas, D. P. (2005). *Dynamic programming and optimal control*. 3rd ed. Athena Scientific (cit. on p. 17).

Bertsekas, D. P. and I. B. Rhodes (1971). “On the minimax reachability of target sets and target tubes”. In: *Automatica* 7, pp. 233–247 (cit. on p. 56).

Berz, M. and G. Hoffstätter (1998). “Computation and Application of Taylor Polynomials with Interval Remainder Bounds”. In: *Reliable Computing* 4, pp. 83–97 (cit. on p. 168).

Besselmann, T., J. Löfberg, and M. Morari (2012). “Explicit MPC for LPV Systems: Stability and Optimality”. In: *IEEE Transactions on Automatic Control* 57, pp. 2322–2332 (cit. on pp. 20, 68, 93).

Betti, G., M. Farina, and R. Scattolini (2013). “A Robust MPC Algorithm for Offset-Free Tracking of Constant Reference Signals”. In: *IEEE Transactions on Automatic Control* 58, pp. 2394–2400 (cit. on p. 144).

Bianchi, F. D., H. De Battista, and R. J. Mantz (2007). *Wind Turbine Control Systems*. Springer (cit. on p. 11).

Blanchini, F. and S. Miani (2015). *Set-Theoretic Methods in Control*. 2nd ed. Birkhäuser (cit. on pp. 41, 43, 69, 84).

Blanchini, F., S. Miani, and C. Savorgnan (2007). “Stability results for linear parameter varying and switching systems”. In: *Automatica* 43, pp. 1817–1823 (cit. on p. 93).

Blanco, T. B., M. Cannon, and B. De Moor (2010). “On efficient computation of low-complexity controlled invariant sets for uncertain linear systems”. In: *Int. J. of Control* 83, pp. 1339–1346 (cit. on p. 70).

Boef, A. J. den (2016). “Optical wafer metrology sensors for process-robust CD and overlay control in semiconductor device manufacturing”. In: *Surface Topography: Metrology and Properties* 4 (cit. on p. 184).

Boyd, S., L. El Ghaoui, E. Feron, and V. Balakrishnan (1994). *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics (cit. on p. 45).

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press (cit. on pp. 18, 217).

Breiman, L. and A. Cutler (1989). *A Deterministic Algorithm for Global Optimization*. Tech. rep. University of California Berkeley (cit. on p. 167).

Broomhead, T. J. et al. (2014). “A robust model predictive control framework for diesel generators”. In: *Proc. of the 19th IFAC World Congress*, pp. 11848–11853 (cit. on p. 58).

Brunner, F. D., M. Lazar, and F. Allgöwer (2013). “An Explicit Solution to Constrained Stabilization via Polytopic Tubes”. In: *Proc. of the 52nd IEEE Conference on Decision and Control*, pp. 7721–7727 (cit. on pp. 57, 58, 60, 67, 68, 73, 79).

Bueler, B., A. Enge, and K. Fukuda (2000). “Exact Volume Computation for Polytopes: A Practical Study”. In: *Polytopes - Combinatorics and Computation*. Birkhäuser, pp. 131–154 (cit. on p. 217).

Bumroongsri, P. (2015). “Tube-based robust MPC for linear time-varying systems with bounded disturbances”. In: *Int. J. of Control, Automation and Systems* 13, pp. 620–625 (cit. on p. 59).

Cagienard, R., P. Grieder, E. C. Kerrigan, and M. Morari (2007). “Move blocking strategies in receding horizon control”. In: *Journal of Process Control* 17, pp. 563–570 (cit. on pp. 98, 120).

Candès, E. J., M. B. Wakin, and S. P. Boyd (2008). “Enhancing sparsity by reweighted ℓ_1 minimization”. In: *Journal of Fourier Analysis and Applications* 14, pp. 877–905 (cit. on p. 123).

Cannon, M., J. Buerger, B. Kouvaritakis, and S. V. Raković (2011). “Robust Tubes in Nonlinear Model Predictive Control”. In: *IEEE Transactions on Automatic Control* 56, pp. 1942–1947 (cit. on pp. 157, 166).

Cannon, M. and B. Kouvaritakis (2005). “Optimizing prediction dynamics for robust MPC”. In: *IEEE Transactions on Automatic Control* 50, pp. 1892–1897 (cit. on p. 23).

Casavola, A., D. Famularo, and G. Franzè (2003). “Predictive control of constrained nonlinear systems via LPV linear embeddings”. In: *Int. J. of Robust and Nonlinear Control* 13, pp. 281–294 (cit. on pp. 25, 157, 174, 176–178, 181).

Casavola, A., D. Famularo, and G. Franzè (2008). “A predictive control strategy for norm-bounded LPV discrete-time systems with bounded rates of parameter change”. In: *Int. J. of Robust and Nonlinear Control* 18, pp. 714–740 (cit. on p. 22).

Casavola, A., D. Famularo, and G. Franzè (2002). “A Feedback Min-Max MPC Algorithm for LPV Systems Subject to Bounded Rates of Change of Parameters”. In: *IEEE Transactions on Automatic Control* 47, pp. 1147–1153 (cit. on pp. 22, 25, 157).

Casavola, A., D. Famularo, and G. Franzè (2004). “Robust constrained predictive control of uncertain norm-bounded linear systems”. In: *Automatica* 40, pp. 1865–1876 (cit. on p. 22).

Casavola, A., D. Famularo, G. Franzè, and E. Garone (2006). “An improved predictive control strategy for polytopic LPV linear systems”. In: *Proc. of the 45th IEEE Conference on Decision & Control*, pp. 5820–5825 (cit. on p. 22).

Casavola, A., D. Famularo, G. Franzè, and E. Garone (2007). “A dilated MPC control strategy for LPV linear systems”. In: *Proc. of the 2007 European Control Conference*, pp. 460–466 (cit. on p. 22).

Casavola, A., D. Famularo, G. Franzè, and E. Garone (2012). “A fast ellipsoidal MPC scheme for discrete-time polytopic linear parameter varying systems”. In: *Automatica* 48, pp. 2620–2626 (cit. on pp. 93, 135).

Casavola, A., M. Giannelli, and E. Mosca (2000). “Min-max predictive control strategies for input-saturated polytopic uncertain systems”. In: *Automatica* 36, pp. 125–133 (cit. on p. 22).

Castillo, F. et al. (2015). “Fresh Air Fraction Control in Engines Using Dynamic Boundary Stabilization of LPV Hyperbolic Systems”. In: *IEEE Transactions on Control Systems Technology* 23, pp. 963–974 (cit. on p. 11).

Cheng, Q., M. Cannon, and B. Kouvaritakis (2013). “The design of dynamics in the prediction structure of robust MPC”. In: *Int. J. of Control* 86, pp. 2096–2103 (cit. on p. 23).

Chinneck, J. W. (2008). *Feasibility and Infeasibility in Optimization: Algorithms and Computational Methods*. Springer (cit. on p. 215).

Chisci, L., P. Falugi, and G. Zappa (2003). “Gain-scheduling MPC of nonlinear systems”. In: *Int. J. of Robust and Nonlinear Control* 13, pp. 295–308 (cit. on pp. 25, 156, 159, 165).

Cisneros, P. S. G., S. Voss, and H. Werner (2016). “Efficient Nonlinear Model Predictive Control via quasi-LPV Representation”. In: *Proc. of the 55th IEEE Conference on Decision and Control*, pp. 3216–3221 (cit. on pp. 25, 157, 166, 170).

Cisneros, P. S. G. and H. Werner (2017). “Parameter-Dependent Stability Conditions for Quasi-LPV Model Predictive Control”. In: *Proc. of the 2017 American Control Conference*, pp. 5032–5037 (cit. on p. 157).

Colmegna, P., R. S. Sánchez-Peña, and R. Gondhalekar (2018). “Linear parameter-varying model to design control laws for an artificial pancreas”. In: *Biomedical Signal Processing and Control* 40, pp. 204–213 (cit. on p. 11).

Cox, P. B. and R. Tóth (2016). “LPV State-Space Model Identification in the Bayesian Setting: A 3-step Procedure”. In: *Proc. of the 2016 American Control Conference*, pp. 4604–4610 (cit. on p. 11).

Daafouz, J. and J. Bernussou (2001). “Parameter Dependent Lyapunov Functions for Discrete Time Systems With Time-Varying Parametric Uncertainties”. In: *Systems & Control Letters* 43, pp. 355–359 (cit. on p. 93).

Dantzig, G. B. and B. C. Eaves (1973). “Fourier-Motzkin Elimination and Its Dual”. In: *Journal of Combinatorial Theory (A)* 14, pp. 288–297 (cit. on p. 41).

Dettori, M. and C. W. Scherer (2001). “LPV design for a CD player: An experimental evaluation of performance”. In: *Proc. of the 40th IEEE Conference on Decision and Control*, pp. 4711–4716 (cit. on p. 11).

Di Cairano, S. and F. Borrelli (2016). “Reference Tracking With Guaranteed Error Bound for Constrained Linear Systems”. In: *IEEE Transactions on Automatic Control* 61, pp. 2245–2250 (cit. on pp. 144, 147, 154).

Diehl, M. and J. Björnberg (2004). “Robust dynamic programming for min-max model predictive control of constrained uncertain systems”. In: *IEEE Transactions on Automatic Control* 49, pp. 2253–2257 (cit. on p. 20).

Ding, B.-C. and B. Huang (2007). “Comments on A Feedback Min-Max MPC Algorithm for LPV Systems Subject to Bounded Rates of Change of Parameters”. In: *IEEE Transactions on Automatic Control* 52, pp. 970–970 (cit. on pp. 22, 157, 178).

Do, A. L., C. Poussot-Vassal, O. Sename, and L. Dugard (2013). “LPV Control Approaches in View of Comfort Improvement of Automotive Suspensions Equipped with MR Dampers”. In: *Robust Control and LPV Approaches. Application to Vehicle Dynamics*. Ed. by O. Sename, P. Gáspár, and J. Bokor. Springer. Chap. 7, pp. 183–212 (cit. on p. 11).

Donida, F., C. Romani, F. Casella, and M. Lovera (2009). “Towards integrated modelling and parameter estimation: an LFT-Modelica approach”. In: *Proc. of the 15th IFAC Symposium on System Identification*, pp. 1286–1291 (cit. on pp. 11, 46).

Dorf, R. C. and R. H. Bishop (2011). *Modern Control Systems*. 12th ed. Prentice Hall (cit. on p. 6).

Falugi, P., L. Giarré, L. Chisci, and G. Zappa (2001). “LPV predictive control of the stall and surge for jet engine”. In: *Proc. of the IFAC Workshop on Periodic Control Systems*, pp. 31–36 (cit. on p. 11).

Falugi, P. and D. Q. Mayne (2013). “Model predictive control for tracking random references”. In: *Proc. of the 2013 European Control Conference*, pp. 518–523 (cit. on p. 144).

Fiacchini, M. and M. Alamir (2017). “Computing control invariant sets is easy”. In: eprint: `arXiv:1708.04797v1` (cit. on p. 70).

Fleming, J., B. Kouvaritakis, and M. Cannon (2015). “Robust Tube MPC for Linear Systems With Multiplicative Uncertainty”. In: *IEEE Transactions on Automatic Control* 60, pp. 1087–1092 (cit. on pp. 24, 58, 60, 97).

Franklin, G. F., J. D. Powell, and A. Emami-Naeini (2014). *Feedback Control of Dynamic Systems*. 7th ed. Pearson (cit. on p. 6).

Fukuda, K. (2004). *Frequently Asked Questions in Polyhedral Computation*. URL: `https://www.inf.ethz.ch/personal/fukudak/polyfaq/polyfaq.html` (cit. on pp. 33, 167, 217).

Gahinet, P., P. Apkarian, and G. Becker (1995). “Self-scheduled H_∞ control of linear parameter-varying systems: a design example”. In: *Automatica* 31, pp. 1251–1261 (cit. on p. 11).

Gallieri, M. (2016). *Lasso-MPC—Predictive Control with L1-Regularised Least Squares*. Springer (cit. on p. 68).

Geiselhart, R., R. H. Gielen, M. Lazar, and F. R. Wirth (2014). “An alternative converse Lyapunov theorem for discrete-time systems”. In: *Systems & Control Letters* 70, pp. 49–59 (cit. on p. 75).

Gielen, R. H. et al. (2010). “On polytopic inclusions as a modeling framework for systems with time-varying delays”. In: *Automatica* 46, pp. 615–619 (cit. on p. 45).

Gilbert, E. and I. Kolmanovsky (2002). “Nonlinear tracking control in the presence of state and control constraints: A generalized reference governor”. In: *Automatica* 38, pp. 2063–2073 (cit. on pp. 27, 71).

Gondhalekar, R. and J.-I. Imura (2010). “Least-restrictive move-blocking model predictive control”. In: *Automatica* 46, pp. 1234–1240 (cit. on p. 117).

Gondhalekar, R. and C. N. Jones (2011). “MPC of constrained discrete-time linear periodic systems—A framework for asynchronous control: Strong feasibility, stability and optimality via periodic invariance”. In: *Automatica* 47, pp. 326–333 (cit. on p. 70).

Gonzalez, R. et al. (2011). “Online robust tube-based MPC for time-varying systems: a practical approach”. In: *Int. J. of Control* 84, pp. 1157–1170 (cit. on p. 58).

Goulart, P. J., E. C. Kerrigan, and J. M. Maciejowski (2006). “Optimization over state feedback policies for robust control with constraints”. In: *Automatica* 42, pp. 523–533 (cit. on pp. 24, 57, 120).

Goulart, P. J., E. C. Kerrigan, and D. Ralph (2008). “Efficient robust optimization for robust control with constraints”. In: *Mathematical Programming* 114, pp. 115–147 (cit. on p. 24).

Groot Wassink, M., M. van de Wal, C. W. Scherer, and O. Bosgra (2005). “LPV control for a wafer stage: beyond the theoretical solution”. In: *Control Engineering Practice* 13, pp. 231–245 (cit. on p. 11).

Grünbaum, B. (2003). *Convex polytopes*. 2nd ed. Springer (Originally published by John Wiley, 1967) (cit. on p. 33).

Grüne, L. and J. Pannek (2011). *Nonlinear Model Predictive Control: Theory and Algorithms*. Springer (cit. on pp. 40, 48).

Gunes, B., J. W. van Wingerden, and M. Verhaegen (2017). “Predictor-Based Tensor Regression (PBTR) for LPV subspace identification”. In: *Automatica* 79, pp. 235–243 (cit. on p. 11).

Gurobi Optimization (2017). *Gurobi optimizer reference manual (version 7.0)*. URL: `http://www.gurobi.com/documentation/7.0/refman` (cit. on p. 134).

Hahn, D. W. and M. N. Özisik (2012). “Heat Conduction Fundamentals”. In: *Heat Conduction*. 3rd ed. John Wiley & Sons Ltd. Chap. 1, pp. 1–39 (cit. on p. 186).

Hanema, J., M. Lazar, and R. Tóth (2017a). “Stabilizing Tube-Based Model Predictive Control: Terminal Set and Cost Construction for LPV Systems”. In: *Automatica* 85, pp. 137–144 (cit. on p. 241).

Hanema, J., R. Tóth, and M. Lazar (2016a). “Tube-based anticipative model predictive control for linear parameter-varying systems”. In: *Proc. of the 55th IEEE Conference on Decision and Control*, pp. 1458–1463 (cit. on p. 242).

Hanema, J., M. Lazar, and R. Tóth (2017b). "Tube-based LPV constant output reference tracking MPC with error bound". In: *Proc. of the 20th IFAC World Congress*, pp. 8612–8617 (cit. on p. 241).

Hanema, J., M. Lazar, and R. Tóth (2018a). "Heterogeneously parameterized tube model predictive control of LPV systems". In: *Journal paper in preparation for submission to Automatica* (cit. on p. 241).

Hanema, J., R. Tóth, and M. Lazar (2017c). "Stabilizing Non-linear MPC using Linear Parameter-Varying Representations". In: *Proc. of the 56th IEEE Conference on Decision and Control*, pp. 3582–3587 (cit. on p. 241).

Hanema, J., R. Tóth, and M. Lazar (2017d). "Tube-based anticipative linear parameter-varying MPC: application to non-linear systems". In: *The 36th Benelux Meeting on Systems and Control* (cit. on p. 242).

Hanema, J., R. Tóth, and M. Lazar (2018b). "Tube-based linear parameter-varying MPC for a thermal system". In: *The 37th Benelux Meeting on Systems and Control* (cit. on p. 242).

Hanema, J., R. Tóth, M. Lazar, and H. S. Abbas (2016b). "MPC for Linear Parameter-Varying Systems in Input-Output Representation". In: *Proc. of the 2016 IEEE International Symposium on Intelligent Control*, pp. 354–359 (cit. on p. 242).

Hanema, J., R. Tóth, M. Lazar, and S. Weiland (2015). "Anticipative linear parameter-varying model predictive control". In: *The 34th Benelux Meeting on Systems and Control* (cit. on p. 242).

Hanema, J., R. Tóth, M. Lazar, and S. Weiland (2016c). "Towards anticipative LPV tube model predictive control". In: *The 35th Benelux Meeting on Systems and Control* (cit. on p. 242).

Hanema, J., M. Lazar, and R. Tóth (2017e). "Stabilizing Tube-Based Model Predictive Control: Terminal Set and Cost Construction for LPV Systems (extended version) [arXiv:1702.05393]". In: *arXiv*. eprint: 1702.05393 (cit. on p. 242).

Herceg, M., M. Kvasnica, C. N. Jones, and M. Morari (2013). "Multi-Parametric Toolbox 3.0". In: *Proc. of the European Control Conference*, pp. 502–510. URL: http://control.ee.ethz.ch/~mpt (cit. on pp. 217, 219).

Hjartarson, A., P. Seiler, and G. J. Balas (2014). "LPV analysis of a gain scheduled control for an aeroelastic aircraft". In: *Proc. of the 2014 American Control Conference*, pp. 3778–3783 (cit. on p. 11).

Hoffmann, C., S. M. Hashemi, H. S. Abbas, and H. Werner (2014). "Synthesis of LPV Controllers With Low Implementation Complexity Based on a Reduced Parameter Set". In: *IEEE Transactions on Control Systems Technology* 22, pp. 2393–2398 (cit. on p. 11).

Hoffmann, C. and H. Werner (2014). "Linear Parameter-Varying Control of Complex Mechanical Systems". In: *Proc. of the 19th IFAC World Congress*, pp. 6147–6152 (cit. on p. 46).

Hoffmann, C. and H. Werner (2015a). “Compact LFT-LPV modeling with automated parameterization for efficient LPV controller synthesis”. In: *Proc. of the 2015 American Control Conference*, pp. 119–124 (cit. on p. 46).

Hoffmann, C. and H. Werner (2015b). “LFT-LPV Modeling and Control of a Control Moment Gyroscope”. In: *Proc. of the 54th IEEE Conference on Decision and Control*, pp. 5328–5333 (cit. on p. 46).

Jaulin, L., M. Kieffer, O. Didrit, and E. Walter (2001). *Applied Interval Analysis*. 1st ed. Springer (cit. on pp. 167, 169).

Jiang, Z. P. and Y. Wang (2002). “A converse Lyapunov theorem for discrete-time systems with disturbances”. In: *Systems & Control Letters* 45, pp. 49–58 (cit. on p. 40).

Jost, M., G. Pannocchia, and M. Monnigmann (2015a). “Accelerating tube-based model predictive control by constraint removal”. In: *Proc. of the 54th IEEE Conference on Decision and Control*, pp. 3651–3656 (cit. on p. 120).

Jost, M., G. Pannocchia, and M. Mönnigmann (2015b). “Online constraint removal: Accelerating MPC with a Lyapunov function”. In: *Automatica* 57, pp. 164–169 (cit. on pp. 120, 121).

Kang, C. G. (2016). “Origin of Stability Analysis: “On Governors” by J.C. Maxwell”. In: *IEEE Control Systems Magazine* 36, pp. 77–88 (cit. on p. 3).

Kellett, C. M. (2014). “A compendium of comparison function results”. In: *Mathematics of Control, Signals, and Systems* 26, pp. 339–374 (cit. on pp. 34, 76).

Kennedy, J. and R. Eberhart (1995). “Particle swarm optimization”. In: *Proc. of the IEEE International Conference on Neural Networks*, pp. 1942–1948 (cit. on p. 213).

Kerrigan, E. C. and J. M. Maciejowski (2004). “Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution”. In: *Int. J. of Robust and Nonlinear Control* 14, pp. 395–413 (cit. on p. 19).

Kothare, M. V., V. Balakrishnan, and M. Morari (1996). “Robust Constrained Model Predictive Control using Linear Matrix Inequalities”. In: *Automatica* 32, pp. 1361–1379 (cit. on pp. 21–23, 59, 85, 90).

Kou, W. (2013). “Modeling and observer design for the prediction of wafer table temperature distributions”. MSc. thesis (company confidential). Eindhoven University of Technology (cit. on pp. 183, 187).

Kouvaritakis, B. and M. Cannon (2016). *Model Predictive Control. Classical, Robust and Stochastic*. Springer (cit. on p. 20).

Kouvaritakis, B., J. A. Rossiter, and J. Schuurmans (2000). “Efficient robust predictive control”. In: *IEEE Transactions on Automatic Control* 45, pp. 1545–1549 (cit. on p. 23).

Kovács, L. (2017). “Linear parameter varying (LPV) based robust control of type-I diabetes driven for real patient data”. In: *Knowledge-Based Systems* 122, pp. 199–213 (cit. on p. 11).

Kvasnica, M., B. Takács, J. Holaza, and D. Ingole (2015). “Reachability analysis and control synthesis for uncertain linear systems in MPT”. In: *Proc. of the 8th IFAC Symposium on Robust Control Design*, pp. 302–307 (cit. on p. 43).

Kwiatkowski, A., M. T. Boll, and H. Werner (2006). “Automated Generation and Assessment of Affine LPV Models”. In: *Proc. of the 45th IEEE Conference on Decision and Control*, pp. 6690–6695 (cit. on pp. 9, 11, 46).

Langson, W., I. Chryssochoos, S. V. Raković, and D. Q. Mayne (2004). “Robust model predictive control using tubes”. In: *Automatica* 40, pp. 125–133 (cit. on pp. 24, 57, 58, 60, 67, 79, 149).

Lawson, R. A. and A. P. G. Robinson (2016). “Overview of materials and processes for lithography”. In: *Materials and Processes for Next Generation Lithography*. Ed. by A. P. G. Robinson and R. A. Lawson. Elsevier. Chap. 1, pp. 1–90 (cit. on pp. 5, 183).

Lazar, M. and V. Spinu (2015). “Finite–step Terminal Ingredients for Stabilizing Model Predictive Control”. In: *Proc. of the 5th IFAC Conference on Non-linear Model Predictive Control*, pp. 9–15 (cit. on pp. 70, 84).

Lee, J. H. and Z. Yu (1997). “Worst-case Formulations of Model Predictive Control for Systems with Bounded Parameters”. In: *Automatica* 33, pp. 763–781 (cit. on p. 19).

Lee, Y. I., M. Cannon, and B. Kouvaritakis (2005). “Extended invariance and its use in model predictive control”. In: *Automatica* 41, pp. 2163–2169 (cit. on p. 70).

Lee, Y. I. and B. Kouvaritakis (2006). “Constrained robust model predictive control based on periodic invariance”. In: *Automatica* 42, pp. 2175–2181 (cit. on p. 70).

Lee, Y. I., B. Kouvaritakis, and M. Cannon (2002). “Constrained receding horizon predictive control for nonlinear systems”. In: *Automatica* 38, pp. 2093–2102 (cit. on pp. 157, 166).

Leith, D. J. and W. E. Leithead (1998). “Gain-scheduled and nonlinear systems: Dynamic analysis by velocity-based linearization families”. In: *Int. J. of Control* 70, pp. 289–317 (cit. on p. 46).

Leith, D. J. and W. E. Leithead (2000a). “On formulating nonlinear dynamics in LPV form”. In: *Proc. of the 39th IEEE Conference on Decision and Control* 4, pp. 3526–3527 (cit. on p. 9).

Leith, D. J. and W. E. Leithead (2000b). “Survey of gain-scheduling analysis and design”. In: *Int. J. of Control* 73, pp. 1001–1025 (cit. on p. 9).

Lhachemi, H., D. Saussié, and G. Zhu (2015). “A structured H_∞-based optimization approach for integrated plant and self-scheduled flight control system design”. In: *Aerospace Science and Technology* 45, pp. 30–38 (cit. on p. 11).

Limón Marruedo, D., T. Álamo, and E. F. Camacho (2002). “Input-to-state stable MPC for constrained discrete-time nonlinear systems with bounded additive uncertainties”. In: *Proc. of the 41st IEEE Conference on Decision and Control*, pp. 4619–4624 (cit. on p. 19).

Limon, D., I. Alvarado, T. Alamo, and E. F. Camacho (2010). “Robust tube-based MPC for tracking of constrained linear systems with additive disturbances”. In: *Journal of Process Control* 20, pp. 248–260 (cit. on p. 25).

Löfberg, J. (2003). “Approximations of closed-loop minimax MPC”. In: *Proc. of the 42nd IEEE Conference on Decision and Control*, pp. 1438–1442 (cit. on p. 24).

Lu, Y. and Y. Arkun (2000a). “A Quasi-min-max MPC Algorithm for Linear Parameter Varying Systems with Bounded Rate of Change of Parameters”. In: *Proc. of the 2000 American Control Conference*, pp. 3234–3238 (cit. on pp. 21, 22).

Lu, Y. and Y. Arkun (2000b). “Quasi-Min-Max MPC algorithms for LPV systems”. In: *Automatica* 36, pp. 527–540 (cit. on pp. 21, 22, 25, 59, 92, 156).

Lu, Y. and Y. Arkun (2002). “A scheduling quasi–min-max model predictive control algorithm for nonlinear systems”. In: *Journal of Process Control* 12, pp. 589–604 (cit. on pp. 25, 156, 157, 181).

Lucia, S., T. Finkler, and S. Engell (2013). “Multi-stage nonlinear model predictive control applied to a semi-batch polymerization reactor under uncertainty”. In: *Journal of Process Control* 23, pp. 1306–1319 (cit. on p. 108).

Luspay, T. and K. Grigoriadis (2015). “Robust linear parameter-varying control of blood pressure using vasoactive drugs”. In: *Int. J. of Control* 88, pp. 2013–2029 (cit. on p. 11).

Maciejowski, J. M. (2002). *Predictive Control with Constraints*. Prentice Hall (cit. on p. 5).

Magni, L., D. M. Raimondo, and R. Scattolini (2006). “Regional Input-to-State Stability for Nonlinear Model Predictive Control”. In: *IEEE Transactions on Automatic Control* 51, pp. 1548–1553 (cit. on p. 40).

Maiworm, M., T. Bäthge, and R. Findeisen (2015). “Scenario-based Model Predictive Control: Recursive Feasibility and Stability”. In: *Proc. of the 9th IFAC Symposium on Advanced Control of Chemical Processes*, pp. 50–56 (cit. on p. 108).

Majecki, P. et al. (2015). “Real-Time Predictive Control for SI Engines Using Linear Parameter-Varying Models”. In: *Proc. of the 5th IFAC Conference on Non-linear Model Predictive Control* (cit. on p. 11).

Mangasarian, O. L. (1994). “Linear Inequalities and Theorems of the Alternative”. In: *Nonlinear Programming*. Society for Industrial and Applied Mathematics. Chap. 2, pp. 16–37 (cit. on p. 59).

Maxwell, J. C. (1868). “On governors”. In: *Proc. of the Royal Society* 100 (cit. on p. 3).

Mayne, D. Q. (2014). “Model predictive control: Recent developments and future promise”. In: *Automatica* 50, pp. 2967–2986 (cit. on pp. 20, 21).

Mayne, D. Q. (2016). “Robust and stochastic model predictive control: Are we going in the right direction?” In: *Annual Reviews in Control* 41, pp. 184–192 (cit. on p. 20).

Mayne, D. Q., J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert (2000). “Constrained model predictive control: Stability and optimality”. In: *Automatica* 36, pp. 789–814 (cit. on pp. 48, 72, 103).

Mayne, D. Q., M. M. Seron, and S. V. Raković (2005). “Robust model predictive control of constrained linear systems with bounded disturbances”. In: *Automatica* 41, pp. 219–224 (cit. on pp. 24, 56, 58–60).

Miani, S. and C. Savorgnan (2005). “MAXIS-G: A software package for computing polyhedral invariant sets for constrained LPV systems”. In: *Proc. of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, pp. 7609–7614 (cit. on pp. 43, 69).

Mohammadpour, J. and C. W. Scherer (2012). *Control of linear parameter varying systems with applications*. Springer (cit. on p. 11).

Moreau, W. M. (1988). *Semiconductor Lithography. Principles, Practices, and Materials*. 1st ed. Plenum Press (cit. on pp. 5, 183).

Munir, S., M. Hovd, G. Sandou, and S. Olaru (2016). “Controlled contractive sets for low-complexity constrained control”. In: *Proc. of the 2016 IEEE Multi-Conference on Systems and Control* (cit. on pp. 70, 213, 214).

Muñoz de la Peña, D., T. Alamo, and A. Bemporad (2005). “A decomposition algorithm for feedback min-max model predictive control”. In: *Proc. of the 44th IEEE Conference on Decision and Control*, pp. 5126–5131 (cit. on pp. 19, 215).

Muñoz de la Peña, D., T. Alamo, A. Bemporad, and E. F. Camacho (2006). “Feedback Min-Max Model Predictive Control Based on a Quadratic Cost Function”. In: *Proc. of the 2006 American Control Conference*, pp. 1575–1580 (cit. on p. 108).

Muñoz-Carpintero, D., M. Cannon, and B. Kouvaritakis (2015). “Robust MPC strategy with optimized polytopic dynamics for linear systems with additive and multiplicative uncertainty”. In: *Systems & Control Letters* 81, pp. 34–41 (cit. on pp. 24, 58).

Oldewurtel, F., R. Gondhalekar, C. N. Jones, and M. Morari (2009). “Blocking parameterizations for improving the computational tractability of affine disturbance feedback MPC problems”. In: *Proc. of the Joint 48th IEEE CDC and 28th CCC*, pp. 7381–7386 (cit. on pp. 120, 121).

Pandey, A. P. and M. C. de Oliveira (2017). “A new discrete-time stabilizability condition for Linear Parameter-Varying systems”. In: *Automatica* 79, pp. 214–217 (cit. on p. 96).

Pannocchia, G. (2004). “Robust model predictive control with guaranteed setpoint tracking”. In: *Journal of Process Control* 14, pp. 927–937 (cit. on pp. 25, 143).

Pannocchia, G., M. Gabiccini, and A. Artoni (2015). “Offset-free MPC explained: novelties, subtleties, and applications”. In: *Proc. of the 5th IFAC Conference on Non-linear Model Predictive Control*, pp. 342–351 (cit. on p. 143).

Papageorgiou, G., K. Glover, G. D'Mello, and Y. Patel (2000). "Taking robust LPV control into flight on the VAAC Harrier". In: *Proc. of the 39th IEEE Conference on Decision and Control* 5, pp. 4558–4564 (cit. on p. 11).

Petersson, D. and J. Löfberg (2009). "Optimization Based LPV-Approximation of Multi-Model Systems". In: *Proc. of the European Control Conference*, pp. 3172–3177 (cit. on p. 46).

Pin, G., D. M. Raimondo, L. Magni, and T. Parisini (2009). "Robust Model Predictive Control of Nonlinear Systems With Bounded and State-Dependent Uncertainties". In: *IEEE Transactions on Automatic Control* 54, pp. 1681–1687 (cit. on p. 19).

Pistikopoulos, E. N., M. C. Georgiadis, and V. Dua (2011). *Multi-Parametric Programming: Theory, Algorithms, and Applications*. Wiley (cit. on pp. 20, 219).

Pistikopoulos, E. N. (2012). "From multi-parametric programming theory to MPC-on-a-chip multi-scale systems applications". In: *Computers and Chemical Engineering* 47, pp. 57–66 (cit. on p. 20).

Pluymers, B. (2006). "Robust Model Based Predictive Control. An Invariant Set Approach". PhD thesis. Katholieke Universiteit Leuven (cit. on pp. 22, 23, 70).

Pluymers, B., J. A. Rossiter, J. A. K. Suykens, and B. De Moor (2005a). "Interpolation based MPC for LPV systems using polyhedral invariant sets". In: *Proc. of the 2005 American Control Conference*, pp. 2–7 (cit. on p. 23).

Pluymers, B., J. A. K. Suykens, and B. De Moor (2005b). "Min-max feedback MPC using a time-varying terminal constraint set and comments on "Efficient robust constrained model predictive control with a time-varying terminal constraint set"". In: *Systems & Control Letters* 54, pp. 1143–1148 (cit. on p. 23).

Poli, R., J. Kennedy, and T. Blackwell (2007). "Particle swarm optimization. An overview". In: *Swarm Intelligence* 1, pp. 33–57 (cit. on p. 213).

Rahme, S. et al. (2016). "LPV model development and control of a solution copolymerization reactor". In: *Control Engineering Practice* 48, pp. 1044–1050 (cit. on p. 11).

Raković, S. V. (2015). "Robust Model-Predictive Control". In: *Encyclopedia of Systems and Control*. Springer, pp. 1225–1233 (cit. on p. 20).

Raković, S. V., B. Kouvaritakis, R. Findeisen, and M. Cannon (2012a). "Homothetic tube model predictive control". In: *Automatica* 48, pp. 1631–1638 (cit. on pp. 24, 57, 60, 67, 79).

Raković, S. V. and M. Lazar (2012). "Minkowski terminal cost functions for MPC". In: *Automatica* 48, pp. 2721–2725 (cit. on pp. 35, 68).

Raković, S. V., W. S. Levine, and B. Açıkmeşe (2016a). "Discretely Generalized Model Predictive Control". In: *Proc. of the 2016 American Control Conference*, pp. 26–31 (cit. on pp. 58, 60, 71).

Raković, S. V., W. S. Levine, and B. Açıkmeşe (2016b). "Elastic Tube Model Predictive Control". In: *Proc. of the 2016 American Control Conference*, pp. 3594–3599 (cit. on pp. 57, 58, 117).

Raković, S. V. et al. (2012b). "Parameterized tube model predictive control". In: *IEEE Transactions on Automatic Control* 57, pp. 2746–2761 (cit. on pp. 24, 57).

Rawlings, J. B. and D. Q. Mayne (2009). *Model Predictive Control: Theory and Design*. Nob Hill Publishing (cit. on pp. 5, 6, 15, 48, 58).

Richards, A. (2005). "Robust model predictive control for time-varying systems". In: *Proc. of the 44th IEEE Conference on Decision and Control*, pp. 3747–3752 (cit. on p. 58).

Richter, H. (2012). *Advanced Control of Turbofan Engines*. Springer (cit. on p. 11).

Rizvi, S. Z. et al. (2018). "State-space LPV model identification using kernelized machine learning". In: *Automatica* 88, pp. 38–47 (cit. on p. 11).

Rossiter, J. A., B. Pluymers, and B. De Moor (2007). "The Potential of Interpolation for Simplifying Predictive Control and Application to LPV Systems". In: *Assessment and Future Directions of Nonlinear Model Predictive Control*. Ed. by R. Findeisen, F. Allgöwer, and L. T. Biegler. Springer, pp. 63–76 (cit. on p. 23).

Rugh, W. J. and J. S. Shamma (2000). "Research on gain scheduling". In: *Automatica* 36, pp. 1401–1425 (cit. on pp. 6, 9, 11, 25, 156).

Rump, S. M. (1999). "INTLAB - INTerval LABoratory". In: *Developments in Reliable Computing*. Ed. by T. Csendes. Kluwer Academic Publishers, pp. 77–104 (cit. on p. 170).

Saltık, M. B. et al. (2018). "An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects". In: *Journal of Process Control* 61, pp. 77–102 (cit. on pp. 20, 21).

Sandou, G. (2009). "Particle Swarm Optimization: an Efficient Tool for the Design of Automatic Control Laws". In: *Proc. of the European Control Conference 2009*, pp. 2670–2675 (cit. on p. 213).

Satzger, C., R. de Castro, and A. Knoblach (2017). "Robust Linear Parameter Varying Model Predictive Control and its Application to Wheel Slip Control". In: *Proc. of the 20th IFAC World Congress*, pp. 1550–1556 (cit. on pp. 58, 60).

Scherer, C. W., R. G. E. Njio, and S. Bennani (1997). "Parametrically varying flight control system design with full block scalings". In: *Proc. of the 36th Conference on Decision and Control*, pp. 1510–1515 (cit. on pp. 9, 11, 45).

Schetzen, M. (2002). *Linear Time-Invariant Systems*. 1st ed. Wiley (cit. on p. 6).

Schneider, R. (2013a). "Basic Convexity". In: *Convex Bodies: The Brunn-Minkowski Theory*. 2nd ed. Cambridge University Press, pp. 1–73 (cit. on pp. 75, 122).

Schneider, R. (2013b). *Convex Bodies: The Brunn-Minkowski Theory*. 2nd ed. Cambridge University Press (cit. on p. 32).

Schuurmans, J. and J. A. Rossiter (2000). “Robust predictive control using tight sets of predicted states”. In: *IEE Proc.-Control Theory Appl.* 147, pp. 13–18 (cit. on p. 22).

Scokaert, P. O. M. and D. Q. Mayne (1998). “Min-max Feedback Model Predictive Control for Constrained Linear Systems”. In: *IEEE Transactions on Automatic Control* 43, pp. 1136–1142 (cit. on pp. 19, 215).

Sedlaczek, K. and P. Eberhard (2006). “Using augmented Lagrangian particle swarm optimization for constrained problems in engineering”. In: *Structural and Multidisciplinary Optimization* 32, pp. 277–286 (cit. on p. 216).

Shamma, J. S. and M. Athans (1990). “Analysis of Gain Scheduled Control for Nonlinear Plants”. In: *IEEE Transactions on Automatic Control* 35, pp. 898–907 (cit. on p. 9).

Shamma, J. S. and M. Athans (1991). “Guaranteed Properties of Gain Scheduled Control for Linear Parameter-varying Plants”. In: *Automatica* 27, pp. 559–564 (cit. on p. 9).

Shamma, J. S. and M. Athans (1992). “Gain Scheduling: Potential Hazards and Possible Remedies”. In: *IEEE Control Systems* 12, pp. 101–107 (cit. on p. 9).

Shamma, J. S. and J. R. Cloutier (1993). “Gain-Scheduled Missile Autopilot Design Using Linear Parameter Varying Transformations”. In: *Journal of Guidance, Control, and Dynamics* 16, pp. 256–263 (cit. on p. 46).

Shamma, J. S. (2012). “An Overview of LPV Systems”. In: *Control of Linear Parameter Varying Systems with Applications*. Ed. by J. Mohammadpour and C. W. Scherer. Springer, pp. 3–26 (cit. on pp. 6, 8, 14).

Sheer, S. and P. O. Gutman (2016). “A Novel Approach to the Computation of Polyhedral Invariant Sets for Constrained Systems”. In: *Proc. of the 2016 IEEE Multi-Conference on Systems and Control*, pp. 1428–1433 (cit. on p. 70).

Shekhar, R. C. and J. M. Maciejowski (2012). “Robust variable horizon MPC with move blocking”. In: *Systems & Control Letters* 61, pp. 587–594 (cit. on p. 98).

Shekhar, R. C. and J. M. Maciejowski (2013). “Corrigendum to “Robust variable horizon MPC with move blocking” [Systems Control Lett. 61 (4) (2012) 587–594]”. In: *Systems & Control Letters* 62, pp. 451–452 (cit. on p. 98).

Shekhar, R. C. and C. Manzie (2015). “Optimal move blocking strategies for model predictive control”. In: *Automatica* 61, pp. 27–34 (cit. on pp. 117, 120, 121).

Shirazi, F. A., K. M. Grigoriadis, and D. Viassolo (2012). “Wind turbine integrated structural and LPV control design for improved closed-loop performance”. In: *Int. J. of Control* 85, pp. 1178–1196 (cit. on p. 11).

Singh, R. et al. (2013). “Semiconductor Manufacturing”. In: *Guide to State-of-the-Art Electron Devices*. Ed. by J. N. Burghartz. John Wiley & Sons Ltd. Chap. 10, pp. 121–132 (cit. on pp. 5, 183).

Skogestad, S. and I. Postlethwaite (2005). *Multivariable Feedback Control. Analysis and Design*. Wiley (cit. on p. 1).

Smith, R. S. (2004). “Robust model predictive control of constrained linear systems”. In: *Proc. of the 2004 American Control Conference*, pp. 245–250 (cit. on p. 59).

Su, Y., K. K. Tan, and T. H. Lee (2012). “Tube based quasi-min-max output feedback MPC for LPV systems”. In: *Proc. of the 8th IFAC Symposium on Advanced Control of Chemical Processes*, pp. 186–191 (cit. on p. 59).

Suzuki, H. and T. Sugie (2006). “MPC for LPV systems with bounded parameter variation using ellipsoidal set”. In: *Proc. of the American Control Conference*, pp. 5251–5256 (cit. on pp. 59, 60).

Tahir, F. and I. M. Jaimoukha (2015). “Low-Complexity Polytopic Invariant Sets for Linear Systems Subject to Norm-Bounded Uncertainty”. In: *IEEE Transactions on Automatic Control* 60, pp. 1416–1421 (cit. on p. 70).

Theis, J., C. Radisch, and H. Werner (2014). “Self-scheduled control of a gyroscope”. In: *Proc. of the 19th IFAC World Congress*, pp. 6129–6134 (cit. on p. 11).

Tibshirani, R. (1996). “Regression Shrinkage and Selection via the Lasso”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 58, pp. 267–288 (cit. on p. 129).

Tóth, R. (2010). *Modeling and Identification of Linear Parameter-Varying Systems*. Springer (cit. on pp. 6, 36, 37, 46, 90, 173).

Tóth, R., P. S. C. Heuberger, and P. M. J. Van den Hof (2012). “Prediction-Error Identification of LPV Systems: Present and Beyond”. In: *Control of Linear Parameter Varying Systems with Applications*. Ed. by J. Mohammadpour and C. W. Scherer. Springer (cit. on p. 11).

Tóth, R., J. C. Willems, P. S. C. Heuberger, and P. M. J. Van den Hof (2011). “The Behavioral Approach to Linear Parameter-Varying Systems”. In: *IEEE Transactions on Automatic Control* 56, pp. 2499–2514 (cit. on p. 9).

Tudón-Martínez, J. C., D. Hernández-Alcántara, and R. Morales-Menendez (2015). “Semi-Active Suspension Control with LPV Mass Adaptation”. In: *Proc. of the 1st IFAC Workshop on LPV Systems*, pp. 67–72 (cit. on p. 11).

Vízer, D. and G. Mercère (2014). “An H_∞-norm-based approach for operating point selection and LPV model identification from local experiments”. In: *Periodica Polytechnica Electrical Engineering and Computer Science* 58, pp. 121–131 (cit. on p. 46).

Vízer, D., G. Mercère, and E. Laroche (2015). “Gray-box LPV model identification of a 2-DoF surgical robotic manipulator by using an H_∞-norm-based local approach”. In: *Proc. of the 1st IFAC Workshop on LPV Systems*, pp. 79–84 (cit. on p. 46).

Wada, N., K. Saito, and M. Saeki (2006). “Model Predictive Control for Linear Parameter-Varying Systems Using Parameter Dependent Lyapunov Function”. In: *IEEE Transactions on Circuits and Systems II: Express Briefs* 53, pp. 1446–1450 (cit. on p. 22).

Wan, Z. and M. V. Kothare (2003). “Efficient robust constrained model predictive control with a time varying terminal constraint set”. In: *Systems and Control Letters* 48, pp. 375–383 (cit. on p. 23).

Wan, Z., B. Pluymers, M. V. Kothare, and B. De Moor (2006). “Comments on: “Efficient robust constrained model predictive control with a time varying terminal constraint set” by Wan and Kothare”. In: *Automatica* 55, pp. 618–621 (cit. on p. 23).

Wang, Y. J. and J. B. Rawlings (2004). “A new robust model predictive control method I: Theory and computation”. In: *Journal of Process Control* 14, pp. 231–247 (cit. on pp. 25, 143).

Wei, X., L. del Re, and L. Liu (2008). “Air path identification of diesel engines by LPV techniques for gain scheduled control”. In: *Mathematical and Computer Modelling of Dynamical Systems* 14, pp. 495–513 (cit. on p. 11).

Wingerden, J. W. van, I. Houtzager, F. Felici, and M. Verhaegen (2009). “Closed-loop identification of the time-varying dynamics of variable-speed wind turbines”. In: *Int. J. of Robust and Nonlinear Control* 19, pp. 4–21 (cit. on p. 11).

Yu, S., C. Böhm, H. Chen, and F. Allgöwer (2012). “Model predictive control of constrained LPV systems”. In: *Int. J. of Control* 85, pp. 671–683 (cit. on p. 22).

Yuan, M. and Y. Lin (2007). “On the non-negative garrotte estimator”. In: *Journal of the Royal Statistical Society. Series B (Methodological)* 69, pp. 143–161 (cit. on p. 129).

Ziegler, G. M. (1995). *Lectures on Polytopes*. 1st ed. Springer (cit. on pp. 33, 83).

List of publications

The following is a list of publications that have been (co-)authored by the author of this thesis.

Journal papers

- J. Hanema, M. Lazar, and R. Tóth (2018a). "Heterogeneously parameterized tube model predictive control of LPV systems". In: *Journal paper in preparation for submission to Automatica*. (This thesis: Chapters 3, 5)
- H. S. Abbas, J. Hanema, R. Tóth, J. Mohammadpour, and N. Meskin (2018b). "An Improved Robust Model Predictive Control for Linear Parameter-Varying Input-Output Models". In: *Int. J. of Robust and Nonlinear Control* 28, pp. 859–880. (Not part of this thesis)
- J. Hanema, M. Lazar, and R. Tóth (2017a). "Stabilizing Tube-Based Model Predictive Control: Terminal Set and Cost Construction for LPV Systems". In: *Automatica* 85, pp. 137–144. (This thesis: Chapter 4)
- H. S. Abbas, R. Tóth, N. Meskin, J. Mohammadpour, and J. Hanema (2016). "A Robust MPC for Input-Output LPV Models". In: *IEEE Transactions on Automatic Control* 61, pp. 4183–4188. (Not part of this thesis)

Conference proceedings

- H. S. Abbas, J. Hanema, R. Tóth, J. Mohammadpour, and N. Meskin (2018a). "A New Approach to Robust MPC Design for LPV Systems in Input-Output Form". In: *Accepted for presentation at the joint 9th IFAC Symposium on Robust Control Design and 2nd IFAC Workshop on LPV Systems*. (Not part of this thesis)
- J. Hanema, R. Tóth, and M. Lazar (2017c). "Stabilizing Non-linear MPC using Linear Parameter-Varying Representations". In: *Proc. of the 56th IEEE Conference on Decision and Control*, pp. 3582–3587. (This thesis: Chapter 8)
- J. Hanema, M. Lazar, and R. Tóth (2017b). "Tube-based LPV constant output reference tracking MPC with error bound". In: *Proc. of the 20th IFAC World Congress*, pp. 8612–8617. (This thesis: Chapter 7)

- J. Hanema, R. Tóth, and M. Lazar (2016a). "Tube-based anticipative model predictive control for linear parameter-varying systems". In: *Proc. of the 55th IEEE Conference on Decision and Control*, pp. 1458–1463. (This thesis: Chapter 3)

- J. Hanema, R. Tóth, M. Lazar, and H. S. Abbas (2016b). "MPC for Linear Parameter-Varying Systems in Input-Output Representation". In: *Proc. of the 2016 IEEE International Symposium on Intelligent Control*, pp. 354–359. (Not part of this thesis)

- H. S. Abbas, R. Tóth, N. Meskin, J. Mohammadpour, and J. Hanema (2015). "An MPC Approach for LPV Systems in Input-Output Form". In: *Proc. of the 54th IEEE Conference on Decision and Control*, pp. 91–96. (Not part of this thesis)

Preprints

- J. Hanema, M. Lazar, and R. Tóth (2017e). "Stabilizing Tube-Based Model Predictive Control: Terminal Set and Cost Construction for LPV Systems (extended version) [arXiv:1702.05393]". In: *arXiv*. eprint: `1702.05393`. (This thesis: Chapter 4)

Peer-reviewed abstracts

- J. Hanema, R. Tóth, and M. Lazar (2018b). "Tube-based linear parameter-varying MPC for a thermal system". In: *The 37th Benelux Meeting on Systems and Control*. (This thesis: Chapter 9)

- J. Hanema, R. Tóth, and M. Lazar (2017d). "Tube-based anticipative linear parameter-varying MPC: application to non-linear systems". In: *The 36th Benelux Meeting on Systems and Control*. (This thesis: Chapter 8)

- J. Hanema, R. Tóth, M. Lazar, and S. Weiland (2016c). "Towards anticipative LPV tube model predictive control". In: *The 35th Benelux Meeting on Systems and Control*. (This thesis: Chapter 3)

- J. Hanema, R. Tóth, M. Lazar, and S. Weiland (2015). "Anticipative linear parameter-varying model predictive control". In: *The 34th Benelux Meeting on Systems and Control*. (Not part of this thesis)

Acknowledgments

The completion of this thesis marks the end of a decade spent with the Department of Electrical Engineering of the Eindhoven University of Technology. Naturally, when I started as a bachelor student ten years ago, I was not aware of such things as linear parameter-varying systems or model predictive control. It is only natural that a number of people from this environment have contributed to my development which has culminated in the realization of this thesis.

First of all, I like to thank my daily supervisor Roland. We started working together for the first time during my MSc graduation research, and it was you who presented me with the opportunity to continue as a PhD candidate. During these years I have learned a lot from you, from esoteric aspects of LPV system theory to how to write better papers. It was not always fun to find my drafts being returned full of red ink, but in the end, the results were worth the effort. I have also valued your constant support in organizational matters, making sure that the writing of this thesis went smoothly and according to plan, and your positive attitude in times when my motivation was running low.

An equal amount of appreciation goes out, naturally, to my other daily supervisor, Mircea. Your MSc course was my first introduction to the fascinating world of model predictive control. During the first year of my PhD, our collaboration had a bit of a slow start, but I like to think that we more than made up for this. I always enjoyed our deep technical discussions. Your thorough knowledge of the MPC literature and your eye for mathematical rigor have been invaluable in shaping my rough ideas into the final results that are presented in this thesis.

All in all, I have been very happy to work with a team of two dedicated and supportive daily supervisors. I feel that your different areas of expertise and often complimentary viewpoints were truly helpful in strengthening my work. The constructive and fruitful collaboration that we had over the last years has been a truly rewarding experience.

Next, I want to thank my first promotor, Siep. We did not have as much day-to-day collaboration, yet I feel that your contribution to this thesis is significant. By asking the right questions at the right moments, you were always able to point out areas of improvement, in particular with regards to the motivation and positioning of the research and the formulation of clear research questions. Looking back into the distant past, I also recall your lectures on optimization as being one factor that influenced me in selecting the MSc specialization of Control Systems in the first place.

I am also grateful to the members of my PhD committee—Alberto Bemporad, Mark Cannon, Maurice Heemels and Marc van de Wal—for taking the time to read my thesis and for their positive and helpful feedback. In particular, I value the discussions that I have had with Marc on the topic of Chapter 9. I appreciate your critical but constructive attitude towards my results, even though they are very far from being applicable to real industrial thermal control problems with hundreds to thousands of state variables.

Furthermore I thank Ruben Merks for his assistance with defining the thermal model of Chapter 9, and for submitting my papers to the Technical Publication Board of ASML in order to get the necessary permission for publication. (A formality, often accompanied by the disclaimer "purely theoretical contribution, paper does not contain any ASML-related content"!)

I may be among a small number of people who claim that their PhD was, on the whole, not a stressful period. This has been made possible by the fact that I was able to carry out my work in a friendly environment. For this, I first would like to mention first a few people that have shared an office with me during periods of my PhD: Alejandro, Carlos, Harm, Ioannis, Marcella, Mohamed, and Yanin. I hope that you were not too annoyed by my love of mechanical keyboards. Ioannis, I respect your highly critical scientific attitude, and I enjoyed the many discussions that we had on teaching and on robust control. Yanin, I will fondly remember our friendly interaction both inside and outside the office.

Of course, many other people from the Control Systems Group also contributed to creating a nice atmosphere at occasions such as conference trips, Benelux meetings, or simply lunch breaks. These names certainly include—but are not necessarily limited to—Daming, Diana, Henrik, Pepijn, Ruben, Shengling, and Tuan.

Sufficient relaxation outside of work is vital to regenerate mental energy and to come up with fresh ideas. For this, my regular holidays with "de SWS" (you know who you are) were instrumental. In fact, the initial idea that eventually led to the development of Chapter 4 materialized during one of these trips. I look forward to the next one.

Finally, I want to express my gratitude to my family—my parents, brother and sister—for giving me a nice and stable home and for unconditionally supporting me through all these years.

Jurre Hanema
Eindhoven, September 2018

Curriculum vitae

Jurre Hanema was born in 1990 in Heerenveen, The Netherlands. He obtained the BSc and MSc degrees in Electrical Engineering from Eindhoven University of Technology in 2012 and 2014, respectively. His MSc graduation research was on model predictive control of linear parameter-varying systems described in input-output representation, and was titled "Data-driven linear parameter-varying predictive control for process systems". It was done under the supervision of Roland Tóth. As part of his master's education, he also completed an internship at DSM Polymer Intermediates studying the implementation of model predictive control on a distillation column.

In 2014, Jurre joined the Control Systems Group at the Department of Electrical Engineering of the Eindhoven University of Technology as a doctoral candidate. During his PhD, he was supervised by Roland Tóth, Mircea Lazar, and Siep Weiland. His research deals with model predictive control of linear parameter-varying systems, focusing on tube-based approaches with the capability of exploiting available information on future scheduling trajectories ("anticipation"). His PhD thesis—which you are currently reading—is titled "Anticipative model predictive control for linear parameter-varying systems" and was defended in 2018.