Received: 8 July 2020 | Revised: 17 January 2021 | Accepted: 16 February 2021

DOI: 10.1049/cth2.12131

IET Control Theory & Applications

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Stabilizing non-linear model predictive control using linear parameter-varying embeddings and tubes

Jurre Hanema[1] | Roland Tóth[1,2] | Mircea Lazar[1]

[1] Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands (Email: jurre@jhanema.nl; m.lazar@tue.nl)

[2] Systems and Control Laboratory, Institute for Computer Science and Control, Budapest, Hungary

Correspondence
R. Tóth, Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, The Netherlands. Systems and Control Laboratory, Institute for Computer Science and Control, Budapest, Hungary
Email: r.toth@tue.nl

Funding information
European Research Council under Horizon 2020 Framework Programme, Grant/Award Number: 714663

Abstract

This paper proposes a model predictive control (MPC) approach for non-linear systems where the non-linear dynamics are embedded inside a linear parameter-varying (LPV) representation. The non-linear MPC problem is therefore replaced by an LPV MPC problem, without using linearization. Compared to general non-linear MPC, advantages of this approach are that it allows for the tractable construction of a terminal set and cost, and that only a single convex program must be solved online. The key idea that enables proving recursive feasibility and stability, is to restrict the state evolution of the non-linear system to a time-varying sequence of state constraint sets. Because in LPV embeddings, there exists a relationship between the scheduling and state variables, these state constraints are used to construct a corresponding future scheduling tube. Compared to non-time-varying state constraints, tighter bounds on the future scheduling trajectories are obtained. Computing a scheduling tube in this setting requires applying a non-linear function to the sequence of constraint sets. Outer approximations of this non-linear projection-based scheduling tube can be found, e.g., via interval analysis. The computational properties of the approach are demonstrated on numerical examples.

1 | INTRODUCTION

Controlling a non-linear system using MPC typically requires solving a non-convex optimization problem online, which can be computationally demanding. Furthermore, the construction of the terminal set and cost necessary to guarantee closed-loop stability is a challenging task in the general non-linear case. One possible way of coping with these challenges is by *embedding* the non-linear system in an LPV representation [1–3], and to use this LPV model as the prediction model in the controller.

In an LPV model, 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 behaviour. The NMPC 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. Also, tractable approaches for the computation of the 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 a free signal that takes its values independently of the other signals in the embedding, some conservatism is introduced.

To give a brief overview of the state-of-the-art based on this concept, the most relevant NMPC approaches based on LPV embeddings are reviewed next. For a comprehensive survey of MPC methods applicable to LPV systems in general, we refer to [4].

In the scheduling quasi-min-max approach of [5], 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 problem remains feasible, the closed-loop system 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 cannot be established a priori. In the 'gain-scheduling' method of [6], 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 scheduling variable are imposed.

Based on the feedback min-max LPV MPC algorithm of [7] that assumes complete independence of the scheduling variable, the work [8] presents an MPC design applicable to LPV embeddings. The approach relies on the online solutions of linear matrix inequalities (LMIs), and known and bounded ROV on the scheduling variable is assumed. As pointed out in [9], the approach [8] is not guaranteed to satisfy input constraints.

Recently, in [10, 11], an 'iterative' MPC scheme was presented to control non-linear systems embedded in an LPV representation. Therein, knowledge of a scheduling map that describes the non-linear relation between the scheduling variable and the state and input signals 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 generate a new predicted scheduling trajectory, 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 sub-problems is required. If the sub-problems converge and the cost decreases, the closed-loop system can be shown to be asymptotically stable. The concept was applied in [12] to the control of a control moment gyroscope using LPV models in both state-space and input-output representations based on velocity-based linearization. A related approach is [13] where an ARX-based least-squares estimate of the future scheduling trajectory is exploited to construct a single LTV MPC problem to be solved online.

In the current paper, an alternative non-linear MPC approach is developed by utilizing LPV embeddings of the non-linear dynamics in a tube-based LPV-MPC framework, as a further development of the ideas first described in [14]. The controller requires at each sampling instant the solution of a single linear program (LP) or quadratic program (QP). As a key contribution of the paper, it is shown that by bounding the trajectories of the scheduling variable of the embedding, strong guarantees on recursive feasibility and stability are inherited from the tube-based MPC framework for LPV systems [15]. The principal idea that enables bounding the evolution of the scheduling variable, is to first restrict the state evolution of the non-linear system to an initial time-varying sequence of state constraint sets. By applying the known scheduling map to these constraint sets, a corresponding set sequence of possible scheduling trajectories is generated. During the MPC optimization, the scheduling variable is considered as a free signal that can vary inside the constructed scheduling sets.

The approach is not restricted to a specific method of constructing the initial state constraint sequence. In practice, it is proposed to construct it either as a sequence centred around an initially feasible trajectory, or based on a maximum allowed ROV of the state variable. Different ways to compute the corresponding scheduling sets from the selected constraint sets are also proposed: these include straightforward computation of bounds using global optimization or gridding, and a more advanced approach based on interval analysis previously employed in non-linear reachability analysis.

In summary, the usage of an LPV model to embed the non-linear dynamics allows us to formulate the control problem in terms of a single LP or QP to be solved online with an a priori guarantee on recursive feasibility and closed-loop stability. In return, an additional mechanism to bound the future evolution of the introduced scheduling variable is now required.

Both the controllers presented here and in [10, 11] make use of a scheduling map to take into account the relationship between the state and scheduling variables. Where [10] solves a sequence of LTV MPC problems at each time instant, the tube-based method that this paper proposes solves a single LP or QP which is, however, more complex than the problems solved in [10]. Therefore, both approaches provide a different trade-off between computational efficiency and a priori stability guarantees. The stability guarantees of the method presented herein follow from a suitable construction of the LPV embedding combined with the properties of the tube-based LPV MPC framework. The guarantees of [10, 11] rely on convergence of a sequence of QPs to a local minimum that results in a cost decrease, which is not a priori verifiable in general.

As the final part of this introduction, we elaborate on the differences between the current paper and our earlier work on tube-based MPC approaches for LPV systems [14–16].

The paper [15] presents the construction of a terminal set and cost, based on the concept of finite-step contraction, to guarantee stability of a tube model predictive control (TMPC) scheme for LPV systems. Therein, the scheduling variable is considered as an exogenous signal that is not dependent on any state or input variables. As a result, the bounds on the future evolution of the scheduling variable can be arbitrarily fixed. In the current work, the scheduling variable is a signal that depends on the internal state variables of an embedded non-linear system. This means that the bounds on the scheduling variable have to be computed dynamically in order to be consistent with the relationship that exists between the state and scheduling variables. Therefore, the conditions under which recursive feasibility and closed-loop stability can be proven are new and different from [15].

Next, the paper [16] presents alternative tube parameterizations that can be used to enhance the trade-off between computational complexity and control performance. The scheduling variable is considered to be an exogenous signal in the same way as in [15]. The development of the parameterizations in [16] was driven by optimizing the trade-off between computational complexity and the volume of the closed-loop region of attraction. In the current paper, the main challenge for the tube construction lies in 'over-approximating' embedded non-linear dynamics by variations of the scheduling variable without introducing too much conservatism. As such, the conditions that the tube must satisfy are different from [16].

The work presented in [14] was an initial development leading up to the work presented in the current paper. The main differences and improvements over [14] are (i) strengthening of all theoretical results through, for example the introduction of the notion of 'convex outer approximator' and a new definition of λ-contractivity for LPV embeddings, (ii) addition of a section on how to compute the bounds on the scheduling variable, which is critical to practical applicability of the proposed approach, and (iii) new numerical simulations on more challenging examples and a comparison with existing controllers from the literature.

Outline

The remainder of this paper is structured as follows. Preliminaries are discussed in Section 2, which includes notation, an overview of LPV embeddings, and the problem setting considered herein. Section 3 presents the developed algorithm, Section 4 discusses possible initialization procedures to find an initial state constraint sequence, and Section 5 proposes methods to compute the required bounds on the scheduling sets based on the selected state constraints. The approach is demonstrated on two numerical examples (a pendulum driven by a DC motor and a two-tank system) in Section 6. The paper ends with concluding remarks given in Section 7.

2 | PRELIMINARIES

This section introduces notation and basic definitions, gives an overview of LPV embeddings, and introduces the problem setting considered by the authors.

2.1 | Notation and basic definitions

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 $[a..b] = \{i \in \mathbb{N} \mid a \leq i \leq b\}$ and $[a..b) = \{i \in \mathbb{N} \mid a \leq i < b\}$, respectively. A set that is convex, compact, and has a non-empty interior that contains the origin is called a PC-set. A polyhedron is a convex set that can be represented as the intersection of finitely many half-spaces. A polytope is a compact polyhedron and can equivalently be described as the convex hull of finitely many vertices. The power set of $\mathcal{A} \subseteq \mathbb{R}^n$ is the set of all subsets of $\mathcal{A}$ (including the empty set and $\mathcal{A}$ itself), and is denoted by $2^{\mathcal{A}}$. Sequences are denoted compactly as $\{X_i\}_{i=a}^{b} = \{X_a, X_{a+1}, \dots, X_b\}$. 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, define $a \oplus \mathcal{B} = \{a + b \mid b \in \mathcal{B}\}$. The Hausdorff distance between a set $\mathcal{A} \subseteq \mathbb{R}^n$ and the origin is

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

where $\|\cdot\|$ can be any vector norm on $\mathbb{R}^n$.

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, next 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 $\lim_{\xi \to 0} g(\xi) = 0$. Lastly, 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.

2.2 | LPV embeddings of non-linear systems

This section describes the general concept of an LPV embedding, which will be used throughout this paper. Consider a general constrained system represented by a state-space (SS) representation

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

and an LPV state-space (LPV-SS) representation of the form

$$x(k+1) = A\big(\theta(k)\big)x(k) + B(\theta(k))u(k), \; x(0) = x_0, \tag{2}$$

where $k \in \mathbb{N}$. In the above equations $u : \mathbb{N} \to \mathbb{U} \subset \mathbb{R}^{n_u}$ is the control input, $x : \mathbb{N} \to \mathbb{X} \subset \mathbb{R}^{n_x}$ is the state variable, and $\theta : \mathbb{N} \to \Theta \subset \mathbb{R}^{n_\theta}$ is called the scheduling signal. Constraints on the input and state variables are represented by the compact sets $(\mathbb{X}, \mathbb{U})$.

The representation (2) can be constructed such that it is an *embedding* of the non-linear system (1). The general process of constructing an LPV embedding is depicted in Figure 1. Starting with the original non-linear system, scheduling variables $\theta \in \Theta \subset \mathbb{R}^{n_\theta}$ are introduced such that the non-linear system (1) is equivalently represented by an LPV model as

$$f(x, u) = A(\theta)x + B(\theta)u, \tag{3}$$

where the dependence of the matrix functions $(A, B) : \Theta \times \Theta \to \mathbb{R}^{n_x \times n_x} \times \mathbb{R}^{n_x \times n_u}$ can be non-linear. Typically, the introduced scheduling variables are related to the state and input signals, that is there exists a scheduling map $\mu : \mathbb{X} \times \mathbb{U} \to \Theta$ such that $\theta = \mu(x, u)$ (Figure 1.(b)). When an embedding is constructed on a bounded operating region, 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 1.(c)).

Following the above high-level explanation, the concept of an LPV embedding that establishes the link between (1) and (2) can be formally defined as follows:

Definition 1. Let $\mathcal{X} \subseteq \mathbb{X}$ be an open, bounded, and invariant set corresponding to an operating region of the non-linear system (1). Define $\mathcal{U} = \{u \in \mathbb{U} \mid \exists x \in \mathcal{X} : f(x, u) \in \mathcal{X}\}$. Let $\mu : \mathcal{X} \times \mathcal{U} \to \Theta$ be a scheduling map with $\Theta \subset \mathbb{R}^{n_\theta}$ a bounded set. An LPV-SS representation (2) is an embedding of (1) on $\mathcal{X}$, if for all $(x, u) \in \mathcal{X} \times \mathcal{U}$ there holds $f(x, u) = A(\mu(x, u))x + B(\mu(x, u))u$.

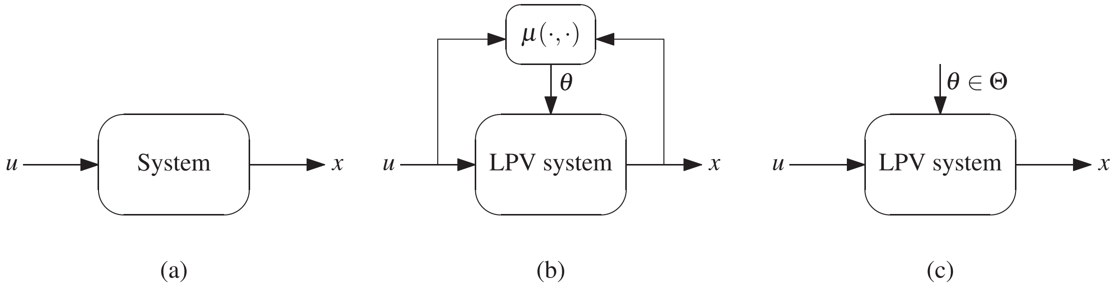

FIGURE 1 The construction of an LPV embedding. (a) The original non-linear system. (b) The non-linear system equivalently represented. (c) The scheduling variable is 'disconnected' and becomes a free signal: the LPV embedding is obtained

From the above definition, it can be concluded that any state trajectory—within the operating domain $\mathcal{X}$—of the non-linear system (1) is equal to a state trajectory of the embedding for *a particular* trajectory of the scheduling signal defined by the scheduling map $\mu(\cdot,\cdot)$. However, when a controller is designed for an LPV embedding, it has to guarantee closed-loop stability with respect to *all possible* scheduling trajectories inside the scheduling set Θ. This leads to the following important result, which the authors will exploit in this article.

Lemma 1 (See [17]). *Suppose that there exists a controller that asymptotically stabilizes the origin of an LPV embedding of a non-linear system, for all possible scheduling trajectories inside the bounded scheduling set* Θ. *Then, the same controller asymptotically stabilizes the origin of the original non-linear system.*

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 analysis of non-linear systems or systems with time delay [18, 19]. 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, for example in [20] for flight control design.

In the use of LPV embeddings, there is a trade-off between (computational) simplicity and achievable control performance. On the one hand, using an 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.

In the literature, a large number of methods are available to generate embeddings for certain classes of non-linear systems [2, Section 7]. These can be classified into three important categories: automated transformation approaches [21–26], [2, Section 7.4], substitution-based approaches [3, 27, 28], and local approaches [29–31]. The definitions of the scheduling variable $\theta(\cdot)$ and the corresponding scheduling map $\mu(\cdot,\cdot)$ depend on the particular approach that is followed to construct the required embedding of the given non-linear system.

The MPC approach presented herein does not assume one particular method for constructing the LPV embedding of the non-linear system (1), as long as the resulting embedding (2) satisfies Definition 1. However, note that velocity-based embedding approaches cannot be applied directly, as these yield an integral form of the embedding defined in terms of $\delta x(k)$ and $\delta u(k)$.

2.3 | Problem setting

This paper considers constrained non-linear systems (1) that can be embedded in a constrained LPV-SS representation of the form

$$x(k+1) = A\big(\theta(k)\big)x(k) + Bu(k), \tag{4a}$$

$$\theta(k) = \mu(x(k)), \tag{4b}$$

where the matrix $A(\cdot)$ is an affine function given as

$$A(\theta) = A_0 + \sum_{i=1}^{n_\theta} A_i$$

and where $\{A_i\}_{i=1}^{n_\theta}$ are real matrices of compatible dimensions. The affine dependence of $A(\cdot)$ on θ is necessary to enable the derivation of a computationally tractable MPC algorithm later. Note that an embedding with affine scheduling dependence can always be constructed by appropriate choice of the scheduling map $\mu(\cdot)$ [26]. The following assumptions on the non-linear system (1) are made.

Assumption 1. The system (1) satisfies the following properties:

(i) The function $f : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}^{n_x}$ is continuously differentiable, and it is bounded on $\mathbb{X} \times \mathbb{U}$.
(ii) The origin is an equilibrium of (1), that is $f(0,0) = 0$.
(iii) The value of the state variable $x(k)$ can be measured for all times $k \in \mathbb{N}$.
(iv) The state and input constraint sets $(\mathbb{X}, \mathbb{U})$ are PC-sets.

Knowledge of the scheduling map $\mu(\cdot)$ can be exploited to obtain sets of all possible scheduling trajectories based on the possible future state evolution; this is what will be done in the MPC developed herein. Note that by knowing $\mu(\cdot)$, Assumption 1.(iii) implies that the value $\theta(k)$ in (4) is known for all time $k \in \mathbb{N}$ as well.

The class of LPV embeddings described by (4) is somewhat restricted, because the map $\mu(\cdot)$ does not depend on u and because the input matrix B does not depend on θ. Some remarks on possible future extensions of the results of this paper to a more general setting are provided in the conclusion of Section 7. A class of systems that can always be embedded in the required form of (4) are, for instance, Lur'e systems of the form

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

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 (4) can be constructed for an arbitrary continuous-time non-linear system can be found in [3, 32]. The discrete-time case is still a topic of ongoing research.

The control problem considered here is to design a controller $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ that renders the origin of the closed-loop representation

$$\begin{aligned} x(k+1) &= A(\theta(k))x(k) + BK_{\mathrm{mpc}}(x(k),\theta(k),k), \\ \theta(k) &= \mu(x(k)), \end{aligned} \tag{5}$$

regionally asymptotically stable in the following sense:

Definition 2. Let $\mathbf{x}(k|x_0)$ denote the solution $x(k)$ of (5) for the initial state $x(0) = x_0$. The origin is said to be a regionally asymptotically stable equilibrium of (5), if there exists a $\mathcal{KL}$-function β and a PC-set $\mathcal{X} \subseteq \mathbb{X} \subset \mathbb{R}^{n_x}$ such that $\|\mathbf{x}(k|x_0)\| \leq \beta(\|x_0\|, k)$ for all $x_0 \in \mathcal{X}$, and for all $k \in \mathbb{N}$.

From the basic properties of embeddings, it follows that if the developed MPC stabilizes the LPV representation (4), then it stabilizes the original non-linear system (1) as well (Lemma 1). It is emphasized that the output of $K_{\mathrm{mpc}}(\cdot,\cdot,\cdot)$ is the result of an optimization problem that is solved online at every sampling instant.

3 | THE MPC ALGORITHM

In this section, the proposed MPC algorithm for LPV embeddings is presented.

3.1 | Fundamentals: tubes for embeddings

Before introducing the concept of a tube, the so-called $\mathcal{CH}_1$-property and the controlled image must be defined as mathematical preliminaries.

Definition 3. A controller $K : X \times \Theta \to \mathbb{U}$ is $\mathcal{CH}_1$ if it is (i) continuous and (ii) positively homogeneous of degree one in the sense $\forall \alpha \in \mathbb{R}_+ : K(\alpha x, \theta) = \alpha K(x, \theta)$.

Definition 4. The controlled image of a set for a constrained LPV embedding represented by the 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 $\mathcal{I}(X, \Theta|K) = \big\{A(\theta)x + B(\theta)K(x,\theta) \mid x \in X, \theta \in \Theta\big\}$.

The type of tubes we employed can now be defined:

Definition 5. Let $\Theta \subseteq \Theta^N$ be a sequence of scheduling sets. A tube of length N is a pair $\mathbf{T} = \big(\mathbf{X}, \mathbf{K}\big) = \big(\{X_i\}_{i=0}^{N}, \{K_i\}_{i=0}^{N-1}\big)$ 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 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 the tubes that were used in previous work [15, 16, 33], our approach has the following two important differences:

- The state constraints are allowed to be time-dependent sequences $\vec{\mathbb{X}} = \{\mathbb{X}_0, \dots, \mathbb{X}_{N-1}\}$.
- 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 paper, 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 dynamics (4).

Note that if all involved sets are polytopic, all the conditions appearing in Definition 5 can be expressed in terms of linear inequalities. The role of the scheduling tube $\Theta \subseteq \Theta^N$ is to bound all possible behaviours of θ in the embedding (4). To this end, the sequence of time-varying state constraints allowed by Definition 5

$$\vec{\mathbb{X}} = \{\mathbb{X}_0, \dots, \mathbb{X}_{N-1}\} \subseteq \mathbb{X}^N \tag{6}$$

is constructed at each time instant k, and an associated scheduling sequence Θ is generated by applying the scheduling map $\mu(\cdot)$. So, the signal θ is still free, but all its possible future trajectories are known to belong to the scheduling tube $\boldsymbol{T}heta$ in a way that is consistent with all possible future state trajectories allowed by the constraints (6). This setup, that allows constraint sequences consisting of arbitrary sets, is more general than the bounds on the state ROV assumed in [6].

For notational and computational simplicity, this paper considers homothetic tubes [33–35]:

Definition 6. A tube $\mathbf{T}$ satisfying to Definition 5 is called a *homothetic tube* if

(i) For all $i \in [0..N]$ the cross sections are parameterized as $X_i = z_i \oplus \alpha_i S$ with (z_i, α_i) being the *centre* and *scaling*, respectively, with S being a fixed polytopic shape set;

(ii) For all $i \in [0..N-1]$ the controllers $K_i : X_i \times \Theta_i \to \mathbb{U}$ are parameterized as gain-scheduled vertex controllers defined on the corresponding sets X_i.

It is, however, possible to use the approach from this paper with the heterogeneously parameterized tubes of [16] or the periodically parameterized tubes of [15] as well.

To obtain a scheduling tube consistent with the state constraints (6), the map $\mu(\cdot)$ has to be applied to set-valued arguments $X \subseteq \mathbb{X}$ according to

$$\mu(X) := \{\mu(x) \mid x \in X\}. \tag{7}$$

Because $\mu(\cdot)$ is a non-linear function in general, Equation 7 is not straightforward to compute directly. If X is a polytope, then there is no guarantee that $\mu(X)$ is also a polytope, or even convex. Therefore, in the sequel, it is assumed that there exists a so-called convex outer-approximator $\hat{\mu}(\cdot)$ of $\mu(\cdot)$.

Definition 7. The function $\hat{\mu} : 2^{\mathbb{X}} \to 2^{\Theta}$ is a convex outer-approximator of $\mu : 2^{\mathbb{X}} \to 2^{\Theta}$ if

(i) For a singleton set $X = \{x\}$, $\hat{\mu}(X) = \hat{\mu}(\{x\}) = \mu(x)$.

(ii) For all convex and compact sets $X \subseteq \mathbb{X}$, $\hat{\mu}(X)$ is a convex set and $\mu(X) \subseteq \hat{\mu}(X)$.

(iii) For all convex and compact sets $X_0 \subseteq X_1 \subseteq \mathbb{X}$, $\hat{\mu}(X_0) \subseteq \hat{\mu}(X_1)$.

(iv) If $X \subseteq \mathbb{X}$ is a polytope, then $\hat{\mu}(X)$ 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 5.

To derive an MPC approach for LPV embeddings, for a measured initial state x and a given constraint sequence $\overset{\rightharpoonup}{\mathbb{X}}$, a set of feasible tubes which depends on the time-varying state constraints can be defined as

$$\begin{aligned}\mathcal{T}_N(x|\overset{\rightharpoonup}{\mathbb{X}}) = \{\mathbf{T} \mid \mathbf{T} \text{ satisfies Def. 6 with } X_0 = \{x\},\\ X_N \subseteq X_{\mathrm{f}}, \text{and } \forall i \in [0..N-1] : \Theta_i = \hat{\mu}(\mathbb{X}_i)\}\end{aligned} \tag{8}$$

where X_{f} is a terminal set, whose construction is discussed later in Section 3.3. For control, it is useful to select from the set (8) a tube that optimizes a specified performance criterion. This is done by solving the tube synthesis problem

$$V(x \mid \overset{\rightharpoonup}{\mathbb{X}}) = \min_{\mathbf{T}} J_N(\mathbf{T}, \overset{\rightharpoonup}{\mathbb{X}}) \text{ subject to } \mathbf{T} \in \mathcal{T}_N(x \mid \overset{\rightharpoonup}{\mathbb{X}}) \tag{9}$$

where $J_N(\cdot,\cdot)$ is the cost function, which will be specified later. The function $V(\cdot|\cdot)$ is called the value function, and an optimizer of (9) is denoted by $\mathbf{T}^\star \in \mathcal{T}_N(x|\overset{\rightharpoonup}{\mathbb{X}})$. Observe that (8)–(9) are not dependent on $\boldsymbol{T}heta$, as one may expect from Definition 5, but on $\overset{\rightharpoonup}{\mathbb{X}}$. This is because in the proposed approach, the scheduling sets in the sequence $\boldsymbol{T}heta \subseteq \Theta^N$ are generated by applying the outer-approximator $\hat{\mu}(\cdot)$ to the given sequence $\overset{\rightharpoonup}{\mathbb{X}} \subseteq \mathbb{X}^N$.

This construction is also the reason why the state constraints in Definition 5 have to be of the form $X_i \subseteq \mathbb{X}_i$ instead of the more relaxed form $\mathcal{I}(X_i, \Theta_i|K_i) \subseteq \mathbb{X}$. The tube transition constraints are of the form $\mathcal{I}(X_i, \Theta_i|K_i) \subseteq X_{i+1}$ where, according to (8), $\Theta_i = \hat{\mu}(\mathbb{X}_i)$. If X_i would not be a subset of $\mathbb{X}_i$, there could exist $x \in X_i$ such that $\mu(x) \notin \Theta_i$, and therefore the transition constraint $\mathcal{I}(X_i, \Theta_i|K_i) \subseteq X_{i+1}$ would not be consistent with the dynamics (4). Therefore the condition $X_i \subseteq \mathbb{X}_i$ is necessary.

To realize (9), the cost function is chosen in the form

$$J_N(\mathbf{T}, \overset{\rightharpoonup}{\mathbb{X}}) = \sum_{i=0}^{N} \ell(X_i, K_i, \hat{\mu}(\mathbb{X}_i)) + F(X_N), \tag{10}$$

with $\ell(\cdot,\cdot,\cdot)$ being the stage cost and $F(\cdot)$ being the terminal cost. The design of the stage cost, which specifies a performance target, is discussed next in Section 3.2. The design of the terminal set and cost, essential for proving recursive feasibility and closed-loop stability, will be addressed in Section 3.3.

Details on how the tube synthesis problem (9) can be formulated as a single LP or QP are given in the Appendix. The exact implementation is dependent on the selected stage cost function, which is addressed next in Section 3.2.

3.2 | Stage cost design

This section presents two possible designs for the stage cost $\ell(\cdot,\cdot,\cdot)$ in (10): a worst-case cost used in previous work [15, 16, 33], and a centre-based cost that is introduced here. Whether to use the worst-case or the centre-based cost is a choice that can be made by the control designer according to the needs of a particular application.

The worst-case cost is defined as

$$\ell^{\mathrm{wc}}(X, K, \Theta) = \max_{(x,\theta)\in X\times\Theta} \left(\|Qx\|^c + \|RK(x,\theta)\|^c\right), \tag{11}$$

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 used to specify performance objectives. Some properties of (11) will be important later to guarantee stability of the MPC algorithm we presented. These are summarized in the following proposition.

Lemma 2 ([16, Proposition 17] or [36, Proposition 3.2]). *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* $\ell^{\mathrm{wc}}(X', K, \Theta') \leq \ell^{\mathrm{wc}}(X, K, \Theta)$.

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

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

It can also be of interest to optimize an average- or expected cost instead of the worst-case one. Hence we can also introduce an alternative centre-based stage cost that has properties similar to those of (11). Define $r(2) := 2$ and $r(\infty) := 1$. This alternative centre-based stage cost function, for sets of the form $X = z \oplus \alpha S$ with $S = \mathrm{convh}\{\bar{s}^1, \ldots, \bar{s}^{q_s}\}$, is then formulated as

$$\ell_p^{\mathrm{cb}}(X, K, \Theta) = \|Qz\|_p^{r(p)} + P|\alpha|^{r(p)} + \frac{1}{q_\Theta q_s} \sum_{i=1}^{q_s} \sum_{j=1}^{q_\Theta} \left\| RK\left(\bar{x}^i, \bar{\theta}^j\right) \right\|_p^{r(p)} \tag{12}$$

where $p \in \{2, \infty\}$, and where the matrices $(Q, R) \in \mathbb{R}^{n_{\mathrm{q}} \times n_{\mathrm{x}}} \times \mathbb{R}^{n_{\mathrm{r}} \times n_{\mathrm{u}}}$ and the scalar $P > 0$ are tuning parameters of full column rank. In (12), the vectors $\bar{x}^i$ and $\bar{\theta}^j$ represent the vertices of the sets X and Θ that are being passed as arguments of $\ell^{\mathrm{cb}}(\cdot, \cdot, \cdot)$. As the centre-based cost (12) is defined for sets represented as $X = z \oplus \alpha S$, it is less general than the worst-case cost (11) which can be applied to arbitrary sets. Nonetheless it has some useful properties:

Lemma 3 ([36, Proposition 8.1]). *In what follows, let* $K : X \times \Theta \to \mathbb{U}$ *be a* $\mathcal{CH}_1$*-controller. The cost function (12) satisfies the following properties:*

(i) For $p \in \{2, \infty\}$, *there exists a* $\mathcal{K}_\infty$*-function* $\underline{\ell}$ *such that* $\underline{\ell}\left(d_H^0(X)\right) \leq \ell_p^{\mathrm{cb}}((X, K, \Theta)$.

(ii) For $p \in \{2, \infty\}$, *the centre-based cost is homogeneous of degree* $r(p)$ *in the sense that for all* $\alpha \in \mathbb{R}_+$, $\ell_p^{\mathrm{cb}}(\alpha X, K, \Theta) = \alpha^{r(p)} \ell_p^{\mathrm{cb}}(X, K, \Theta)$.

Remark 1. It is important to note that the centre-based cost (12) does not satisfy a subset inclusion property like ℓ^{wc} in Lemma 2.(i). This has a consequence for the proof of closed-loop stability: in particular, if the cost (12) is used, it is necessary to relax the initial condition constraint in the set of feasible tubes (8) from $X_0 = \{x\}$ to $x \in X_0$. The full proof of stability, where this point is taken into account, will be presented later for Theorem 1.

3.3 | Terminal set and cost design

To guarantee recursive feasibility of (9), a terminal set constraint $X_N \subseteq X_{\mathrm{f}}$ is included in (8). The set X_{f} must be designed to be controlled λ-contractive for (4). In the case of an LPV embedding, this can be defined in the following sense.

Definition 8. 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-SS embedding (4), if there exists a local $\mathcal{CH}_1$-controller $K : X \times \hat{\mu}(\breve{\mathbb{X}}) \to \mathbb{U}$ such that $\lambda = \inf\left\{\gamma \geq 0 \mid \mathcal{I}(X, \hat{\mu}(\breve{\mathbb{X}}) | K) \subseteq \gamma X\right\} < 1$.

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 available for LPV models (e.g. the algorithm of [37]). 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 is repeated. Observe that X_{f} is always bounded as $X_{\mathrm{f}} \subseteq \breve{\mathbb{X}}$. However, a large $\breve{\mathbb{X}}$ also results in a large scheduling set $\hat{\mu}(\breve{\mathbb{X}})$ and therefore can also lead to a smaller set X_{f}.

Suppose that X_{f} is λ-contractive according to Definition 8. Then a suitable terminal cost $F(\cdot)$ to be used in (10) can be computed with the same basic structure as the terminal cost used in [15, 16]. As a preliminary, define a so-called set-gauge function as follows:

Definition 9 ([15]). The set-gauge function $\Psi_S : 2^{\mathbb{R}^n} \to \mathbb{R}_+$ corresponding to a PC-set $S \subset \mathbb{R}^n$ is $\Psi_S(X) = \sup_{x \in X} \psi_S(x) = \inf\{\gamma \geq 0 \mid X \subseteq \gamma S\}$. For any PC-set S, there exist $\mathcal{K}_\infty$-functions $\underline{\psi}, \overline{\psi}$ such that $\underline{\psi}((d_H^0(X)) \leq \Psi_S(X) \leq \overline{\psi}(d_H^0(X))$.

Next, define the terminal cost as

$$\begin{aligned} F(X) &= \frac{\Psi_{X_{\mathrm{f}}}(X)}{1-\lambda} \left(\max_{\tilde{X} \subseteq X_{\mathrm{f}}, \tilde{\Theta} \subseteq \hat{\mu}(\breve{\mathbb{X}})} \ell\left(\tilde{X}, K_{\mathrm{f}}, \tilde{\Theta}\right) \right) \\ &= \frac{\Psi_{X_{\mathrm{f}}}(X)}{1-\lambda} \bar{\ell}, \end{aligned} \tag{13}$$

where the maximization is done with respect to sets $X = z \oplus \alpha S$ to be in agreement with Definition 6. The function $K_{\mathrm{f}} : X_{\mathrm{f}} \times \hat{\mu}(\breve{\mathbb{X}}) \to \mathbb{U}$ is a local set-induced controller that renders X_{f} λ-contractive in the sense of Definition 8. Because $\ell(\cdot, \cdot, \cdot)$ is convex, (13) reduces to a finite-dimensional optimization problem provided that $\hat{\mu}(\breve{\mathbb{X}})$ is a polytope.

Both possible choices of stage cost (11) or (12) and the terminal cost (13) satisfy conditions that guarantee the existence of $\mathcal{K}_\infty$-bounds on the value function of (9):

Lemma 4 ([16, Proposition 19] or [36, Proposition 3.2]). *There exist* $\mathcal{K}_\infty$*-functions* $\underline{v}, \overline{v}$ *such that for all* $x \in \mathbb{X}$ *and* $\vec{\mathbb{X}} \subseteq \mathbb{X}^N$ *for which (9) is feasible, it holds that* $\underline{v}\left(\|x\|\right) \leq V\left(x | \vec{\mathbb{X}}\right) \leq \overline{v}\left(\|x\|\right)$.

3.4 | Main result: The algorithm

The proposed MPC approach is summarized in Algorithm 1, and Theorem 1 gives the main result concerning its properties.

ALGORITHM 1 The MPC algorithm for LPV embeddings

```
Require: $N \in [1..\infty)$
  1: Compute $\vec{\mathbb{X}}(0)$ based on the initial state $x(0)$
Assume: Step 4 is feasible at $k = 0$, that is, $\mathcal{T}_N(x(0)|\vec{\mathbb{X}}_0) \neq \emptyset$.
  2: $k \leftarrow 0$
  3: loop
  4:     Solve (9) to obtain $\mathbf{T}^\star \in \mathcal{T}_N(x(k)|\vec{\mathbb{X}}(k))$
  5:     Apply $u(k) = K_0^\star(x(k), \theta(k)) = u_0^\star$ to system (1)
  6:     $\forall i \in \mathbb{N}_{[0,N-1]}$ :
             $\mathbb{X}_i(k+1) \leftarrow \begin{cases} \mathbb{X}_{i+1}(k), & i < N-1, \\ \breve{\mathbb{X}}, & i = N-1. \end{cases}$
  7:     $k \leftarrow k + 1$
  8: end loop
```

Theorem 1. *Suppose that $\hat{\mu}(\cdot)$ is a convex outer-approximator of the scheduling map $\mu(\cdot)$ according to Definition 7. Let X_f satisfy Definition 8, let $F(\cdot)$ be as in (13), and in Definition 6 set $S = X_\text{f}$. Furthermore let $\vec{\mathbb{X}} = \{\mathbb{X}_0, \dots, \mathbb{X}_{N-1}\}$ and let $\vec{\mathbb{X}}^+ \subseteq \mathbb{X}^N$ be related to $\vec{\mathbb{X}}$ as $\vec{\mathbb{X}}^+ = \{\mathbb{X}_1, \dots, \mathbb{X}_{N-1}, \breve{\mathbb{X}}\}$ according to Step 6 of Algorithm 1. Then, the algorithm satisfies the following properties:*

- *(i) It is recursively feasible, that is if $\exists \mathbf{T} \in \mathcal{T}_N(x|\vec{\mathbb{X}})$, then $\exists \mathbf{T}^+ \in \mathcal{T}_N(x^+|\vec{\mathbb{X}}^+)$ where $x^+ = A(\theta)x + BK_0(x, \theta) = A(\mu(x))x + BK_0(x, \mu(x))$.*
- *(ii) The state of the controlled system reaches $\breve{\mathbb{X}}$ in N steps or less, that is $\exists k_\star \in \mathbb{N}_{[0,N]}$ such that $\forall k \geq k_\star$: $x(k) \in \breve{\mathbb{X}}$.*
- *(iii) The origin of (1) is regionally asymptotically stabilized in the sense of Definition 2.*

Proof of (i). In what follows, let

$$\Theta = \{\Theta_0, \Theta_1, \dots, \Theta_{N-1}\} = \{\{\theta\}, \hat{\mu}(\mathbb{X}_1), \dots, \hat{\mu}(\mathbb{X}_{N-1})\}$$

be the sequence of scheduling sets generated by applying the convex outer-approximator $\hat{\mu}(\cdot)$ to the sequence $\vec{\mathbb{X}}$. By construction of $\vec{\mathbb{X}}^+$, at the next time instant, the scheduling set sequence becomes

$$\Theta^+ = \{\Theta_0^+, \Theta_1^+, \dots, \Theta_{N-1}^+\} = \{\hat{\mu}(\mathbb{X}_0^+), \hat{\mu}(\mathbb{X}_1^+), \dots, \hat{\mu}(\mathbb{X}_{N-1}^+)\}$$
$$= \{\Theta_1, \Theta_2, \dots, \Theta_{N-1}, \breve{\Theta}\}$$

where $\breve{\Theta} = \hat{\mu}(\breve{\mathbb{X}})$ according to Definition 8. (Note that when actually executing Algorithm 1 we additionally have $\Theta_0^+ = \{\theta^+\} \subseteq \Theta_1$, but for all following proofs the definition of Θ^+ given above is sufficient and more convenient notationally.)

Next, define $\gamma = \Psi_{X_\text{f}}(X_N)$ according to (13). It is easily observed that the relationship $\Theta_i^+ \subseteq \Theta_{i+1}$, $i \in [0..N-2]$ holds. Therefore, if the worst-case stage cost of (11) is employed, a feasible candidate tube $\mathbf{T}^+ \in \mathcal{T}_N(x^+|\vec{\mathbb{X}}^+)$ can be constructed in the same way as in [15, Proposition 8] as

$$\mathbf{T}^+ = (\{\{x^+\}, X_2, \dots, X_{N-1}, \gamma X_\text{f}, \gamma\lambda X_\text{f}\},$$
$$\{K_1, \dots, K_{N-1}, K_\text{f}\}).$$

Observe that if the tube $\mathbf{T}$ satisfies the state constraints of Definition 5, then $\mathbf{T}^+$ satisfies the state constraints $\vec{\mathbb{X}}^+$.

If the centre-based stage cost (12) is used, the set of feasible tubes has a different initial condition constraint (see Remark 1), and it is given by

$$\mathcal{T}_N(x|\vec{\mathbb{X}}) = \{\mathbf{T} \mid \mathbf{T} \text{ satisfies Def. 6 with } x \in X_0,$$
$$X_N \subseteq X_\text{f}, \text{and } \Theta_i = \hat{\mu}(\mathbb{X}_i)\}.$$

The candidate tube $\mathbf{T}^+ \in \mathcal{T}_N(x^+|\vec{\mathbb{X}}^+)$ consequently becomes

$$\mathbf{T}^+ = (\{X_1, X_2, \dots, X_{N-1}, \gamma X_\text{f}, \gamma\lambda X_\text{f}\},$$
$$\{K_1, \dots, K_{N-1}, K_\text{f}\}).$$

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 1.

Proof of (iii). This proof is essentially identical to the proof of [15, Theorem 13], but slightly adapted to connect more directly to the setting of the current paper. Substitute the solutions $\mathbf{T}$ and $\mathbf{T}^+$ (see the proof of (i)) in the cost function of the tube synthesis problem and compute the difference between the value functions at time k and time $k+1$ to obtain

$$\Delta V_k = V(x^+|\vec{\mathbb{X}}^+) - V(x|\vec{\mathbb{X}})$$
$$\leq \ell(X_0^+, K_1, \Theta_1) + \gamma\ell(X_\text{f}, K_\text{f}, \breve{\Theta}) + \gamma\lambda F(X_\text{f}) - F(X_N)$$
$$+ \sum_{i=2}^{N-1} \ell(X_i, K_i, \Theta_i) - \sum_{i=0}^{N-1} \ell(X_i, K_i, \Theta_i),$$

where either $X_0^+ = \{x^+\}$ or $X_0^+ = X_1$ depending on the applicable definition of $\mathbf{T}^+$ (see the proof of (i), also for the definition of γ, Θ_i, and $\breve{\Theta}$). In both cases, observe that $\ell(X_0^+, K_1, \Theta_1) \leq \ell(X_1, K_1, \Theta_1)$ and therefore

$$\Delta V_k \leq \sum_{i=1}^{N-1} \ell(X_i, K_i, \Theta_i) - \sum_{i=0}^{N-1} \ell(X_i, K_i, \Theta_i) + \gamma\ell(X_\text{f}, K_\text{f}, \breve{\Theta})$$
$$+ \gamma\lambda F(X_\text{f}) - F(X_N)$$
$$= -\ell(X_0, K_0, \Theta_0) + \gamma\ell(X_\text{f}, K_\text{f}, \breve{\Theta}) + \gamma\lambda F(X_\text{f}) - F(X_N)$$
$$\leq -\ell(X_0, K_0, \Theta_0) + \gamma\bar{\ell} + \gamma\lambda F(X_\text{f}) - F(X_N)$$

where the last inequality follows from the definition of $\bar{\ell}$ in (13). Since $X_N \subseteq \gamma X_\mathrm{f}$,

$$F(X_N) = \frac{\bar{\ell}}{1-\lambda}\Psi_{X_\mathrm{f}}(X_N) \leq \gamma \frac{\bar{\ell}}{1-\lambda}\Psi_{X_\mathrm{f}}(X_\mathrm{f}) = \gamma F(X_\mathrm{f})$$

and therefore

$$\begin{aligned}\Delta V_k &\leq -\ell(X_0, K_0, \Theta_0) + \gamma\big(\bar{\ell} + \lambda F(X_\mathrm{f}) - F(X_\mathrm{f})\big)\\ &= -\ell(X_0, K_0, \Theta_0)\\ &\leq -\underline{\ell}(\|x\|)\end{aligned}$$

where the last inequality follows from Lemma 2 or Lemma 3. The fact that $V(x|\mathbb{X})$ is monotonically decreasing with rate $\underline{\ell}(\|x\|)$ is, in conjunction with the bounds of Lemma 4, sufficient to conclude that $V(\cdot|\cdot)$ is a (time-varying) Lyapunov function. Regional asymptotic closed-loop stability of the LPV embedding in the sense of Definition 2 follows [38, Theorem 2]. The stability of the origin of the non-linear system (1), which was to be proven, subsequently follows by Lemma 1.

In Step 1 of Algorithm 1, before starting the control loop, an initial state constraint sequence $\vec{\mathbb{X}}(0)$ is computed based on the measured initial state $x(0)$. Possible ways to implement this computation are discussed next in Section 4.

4 | INITIALIZATION APPROACHES

This section presents two possible methods for determining the initial state constraint sequence $\vec{\mathbb{X}}(0)$ that is required in Algorithm 1. The first method of Section 4.2 constructs the sets around a given initially feasible 'seed' trajectory, whereas the second method of Section 4.1 is based on restricting the ROV of the state variable.

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 $\vec{\mathbb{X}}(0)$.

4.1 | Initial feasible trajectory

Suppose that a feasible initial state and input trajectory

$$\tilde{\mathbf{T}}_0 = (\{\tilde{x}_0, \tilde{x}_1, \dots, \tilde{x}_N\}, \{\tilde{u}_0, \dots, \tilde{u}_{N-1}\})$$

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 $\tilde{x}_{i+1} = A\big(\mu(\tilde{x}_i)\big)\tilde{x}_i + B\tilde{u}_i$. Then, by defining some tolerance $\delta \in \mathbb{R}_+^{n_\mathrm{x}}$, the initial state constraint sets in $\vec{\mathbb{X}}(0)$ can be computed according to the relationship

$$\forall i \in \mathbb{N}_{[0,N]} \,:\, \mathbb{X}_i(0) = \begin{cases}\{x_0\}, & i = 0,\\ \big(\tilde{x}_i(0) \oplus \Delta(\delta)\big) \cap \mathbb{X}, & i > 0,\end{cases} \tag{14}$$

where

$$\Delta(\delta) = \big\{x \in \mathbb{R}^{n_\mathrm{x}} \mid \forall i \in \mathbb{N}_{[1,n_\mathrm{x}]} \,:\, |x_i| \leq \delta_i\big\}. \tag{15}$$

In comparison to the bounded ROV initialization discussed next in Section 4.2, this approach is generally more likely to yield a feasible solution to (9), but the assumed availability of the initially feasible trajectory $\tilde{\mathbf{T}}_0$ can be considered a disadvantage (note, however, that linearization-based approaches [39, 40] also presume the knowledge of such a trajectory).

If $\vec{\mathbb{X}}(0)$ is constructed according to (14) with $\delta = 0$, then (9) is always feasible at $k = 0$. Due to the constraint update Step 6 of Algorithm 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. Therefore, 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 [10]. The method [10] is not guaranteed to converge to a solution, but if it converges, the result can be used to initialize the MPC we proposed which subsequently guarantees recursive feasibility and stability.

4.2 | Bounded rate of variation

Let $\delta \in \mathbb{R}_+^{n_\mathrm{x}}$ denote a bound on the ROV of the state variable. Then, given a measurement of the initial system state $x(0) = x_0$, the initial constraint sequence $\vec{\mathbb{X}}(0)$ can be computed as

$$\forall i \in \mathbb{N}_{[0,N]} \,:\, \mathbb{X}_i(0) = \begin{cases}\{x_0\}, & i = 0,\\ \big(\mathbb{X}_{i-1}(0) \oplus \Delta(\delta)\big) \cap \mathbb{X}, & i > 0,\end{cases} \tag{16}$$

with $\Delta(\delta)$ as in (15). The bounded ROV-construction is similar to what was used in [6]. Similarly to the situation in Section 4.1 a value of δ needs to be found which is not too restrictive in terms of the allowed state variation, but which also does not lead to too large scheduling sets.

Note that if there exists a partitioning of the state vector $x = (x_1, x_2)$ such that $\mu(x) = \mu(x_1)$, then it is sufficient to impose a bounded ROV on the x_1-component only.

5 | BOUNDING THE SCHEDULING TUBE

In this section, methods for constructing outer-approximations $\hat{\mu}(X)$ are proposed. The easy special case of an affine scheduling map is presented first in Section 5.1. For general non-linear scheduling maps, some methods to construct the approximations are discussed in Section 5.2. These methods include straightforward computation of bounds using global optimization or gridding, and a more advanced approach based on Taylor expansion and interval analysis. The latter approach was used previously—in a slightly different form—for the purpose of reachability analysis of non-linear systems [41].

Throughout this section, for notational simplicity, all results are given with respect to a general set $X \subseteq \mathbb{X}$ that is assumed to be convex and compact. When applying these results, use should be made of the state constraint sequence $\overset{\rightharpoonup}{\mathbb{X}}(k)$ in order to obtain tight outer-approximations of the scheduling sets that are valid for all possible state trajectories of the closed-loop system.

5.1 | Affine scheduling map

Suppose that $\mu : \mathbb{X} \to \Theta$ is affine, that is of the form

$$\mu(x) = \mathcal{T}x + b,$$

where $\mathcal{T} \in \mathbb{R}^{n_\theta \times n_x}$ and $b \in \mathbb{R}^{n_\theta}$. An affine scheduling map will typically result when constructing an embedding of a bilinear system (1). Furthermore suppose that the vertex representation of X is available as

$$X = \left\{\bar{x}^1, \dots, \bar{x}^{q_x}\right\}. \tag{17}$$

Then, the outer-approximation $\hat{\mu}(X)$ is exact and is given by

$$\hat{\mu}(X) = \mu(X) = \operatorname{convh}\left\{\mathcal{T}\bar{x}^1 + b, \dots, \mathcal{T}\bar{x}^{q_x} + b\right\}. \tag{18}$$

The assumption that the vertex representation (17) is available is not restrictive, because hyperplane and vertex representations of the sets in the sequence $\overset{\rightharpoonup}{\mathbb{X}}$ can be constructed simultaneously when using one of the initialization approaches from Section 4. Hence, no computationally expensive conversion between the two representations has to be performed.

5.2 | General scheduling map

In the general case, the scheduling map $\mu : \mathbb{X} \to \Theta$ is a non-linear function of the form

$$\mu(x) = \begin{bmatrix} \mu_1(x) \\ \vdots \\ \mu_{n_\theta}(x) \end{bmatrix}, \tag{19}$$

where for each $i \in [1..n_\theta]$, $\mu_i : \mathbb{X} \to \mathbb{R}$. For a possibly non-linear and non-convex function $\mu(\cdot)$, obtaining a convex outer-approximator $\hat{\mu}(\cdot)$ is not always straightforward.

Perhaps the simplest way to get an outer approximation $\hat{\mu}(X)$ of $\mu(X)$ 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,

$$\hat{\mu}(X) = \operatorname{convh}\left\{\mu\left(\hat{x}^1\right), \dots, \mu\left(\hat{x}^M\right)\right\}, \tag{20}$$

where the computation of the convex hull only requires the elimination of redundant vertices [42]. In low dimensions, gridding can work efficiently. The main caveat is that the grid has to be dense enough in order to guarantee that $\hat{\mu}(\cdot)$ is indeed a valid outer-approximation in the sense of Definition 7.

Another conceptually simple way of constructing the convex outer-approximation $\hat{\mu}(X)$ is to compute the interval hypercube

$$\hat{\mu}(X) = \left[\min_{x\in X} \mu_1(x), \max_{x\in X} \mu_1(x)\right] \times \cdots \times \left[\min_{x\in X} \mu_{n_\theta}(x), \max_{x\in X} \mu_{n_\theta}(x)\right]. \tag{21}$$

Because the functions $\mu_i(\cdot)$ are non-convex in general, computing the minima and maxima in (21) requires the use of an optimization algorithm that is guaranteed to find global optima, for example [43]. This optimization has to be done online, 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. This conservatism may be reduced by also considering rotations, by compressing the scheduling space [44], or via the method of [26].

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 [41]. Therein, the idea is to construct outer approximations of non-linear functions applied to sets using a combination of Taylor expansion and interval analysis [45]. To apply this approach to the present setting, the following assumption is necessary.

Assumption 2. The function $\mu(\cdot) : \mathbb{X} \to \Theta$ in (19) is twice continuously differentiable.

Let $Df(z)$ denote the gradient of a function $f : \mathbb{R}^n \to \mathbb{R}$ evaluated at the point z with $z \in \mathbb{R}^n$, and let $D^2 f(z)$ denote its Hessian matrix evaluated at z. The functions $\mu_i(\cdot)$ can be approximated by a first-order Taylor series plus a Lagrange remainder term as

$$\mu_i(x) \in \mu_i\left(x^\star\right) + \left(x - x^\star\right)^\top D\mu_i\left(x^\star\right) \oplus \mathcal{L}_i\left(x, x^\star\right) \tag{22}$$

where the linearization point $x^\star \in X$ is arbitrary and where the set of possible values of the Lagrange remainder is given by

$$\begin{aligned}&\mathcal{L}_i\left(x, x^\star\right)\\&=\left\{\frac{1}{2}\left(x-x^\star\right)^\top D^2\mu_i\left(x^\star+\alpha\left(x-x^\star\right)\right)\left(x-x^\star\right)\right.\\&\qquad\left.\mid \alpha \in [0,1]\right\},\end{aligned} \tag{23}$$

see [46]. 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,

$$\mu_i(x) \in \mu_i\left(x^\star\right) + \left(x - x^\star\right)^\top D\mu_i\left(x^\star\right) \oplus \bar{\mathcal{L}}_i.$$

An expression for the set $\bar{\mathcal{L}}_i$ can be derived following the same basic approach as in [41, Propositions 1–2]. Due to the different problem setting we adopted, the result is summarized for convenience in the following proposition.

Lemma 5 ([36, Proposition 8.2]). *An overbound $\bar{\mathcal{L}}_i$ for the Lagrange remainder $\mathcal{L}(x, x^\star)$ is*

$$\begin{aligned}\forall (x, x^\star) \in X \times X : \mathcal{L}_i\left(x, x^\star\right) \in \bar{\mathcal{L}}_i\\ = \left[-\frac{1}{2}e^\top(x^\star)\max_{x\in X}|D^2\mu_i(x)|e(x^\star),\right.\\ \left.\frac{1}{2}e^\top(x^\star)\max_{x\in X}|D^2\mu_i(x)|e(x^\star)\right]\end{aligned}$$

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

$$x^\star = \frac{1}{q_\mathrm{x}}\sum_{i=1}^{q_\mathrm{x}} \bar{x}^i \tag{24}$$

is 'optimal' in the sense that it gives the smallest possible interval $\bar{\mathcal{L}}_i$.

In Lemma 5, evaluation of the intervals $\bar{\mathcal{L}}_i$ still requires computing the maximum $\max_{x\in X}|D^2\mu_i(x)|$ where $D^2\mu_i(\cdot)$ is the second derivative of $\mu_i(\cdot)$. In [41], it is proposed to compute this bound using interval analysis [45]. 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

$$f\left([a,b]\right) = \left[\min_{x\in[a,b]} f(x), \max_{x\in[a,b]} f(x)\right]$$

in a computationally efficient manner, that is without using global optimization (the max- and min-operations in the above equation are taken element-wise). A list of how interval approximation for a number of common functions can be computed is given in [47], and a commercially available toolbox implementing computationally efficient interval methods is [48].

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 Lemma 5 is small. In [41] it is shown that, if necessary, the linearization error bound can be reduced by splitting the sets X. This approach is not pursued further 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 (19), using, for example the implementation of [48]. This yields an interval hypercube of the form (21), but depending on the function, it may result in a coarse over-approximation.
- Consider the first-order approximation (22). Instead of the interval result in Lemma 5, another method (such as gridding or sampling) to bound the Lagrange remainder could be used.

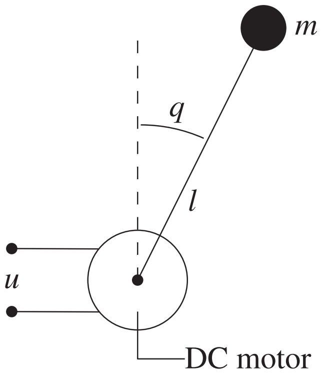

FIGURE 2 The electrically driven inverted pendulum

6 | NUMERICAL EXAMPLES

In this section, the capabilities of the MPC approach we proposed are demonstrated on two numerical examples.

6.1 | 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 2.

TABLE 1 Model parameters for Example 2

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}

Define the state vector

$$x = \begin{bmatrix} q \\ \dot{q} \\ i \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \tag{25}$$

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 1.Introduce a scheduling variable

$$\theta(k) = \mu(x(k)) = \mathrm{sinc}\big(x_1(k)\big),$$

where

$$\mathrm{sinc}\,(x) := \begin{cases} 1, & x = 0, \\ \frac{\sin\,(x)}{x}, & x \neq 0. \end{cases}$$

The dynamics (25) can now be written in the equivalent LPV embedding form as

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 & 0 \\ \frac{mgl}{J}\theta(t) & -\frac{b}{J} & \frac{k}{J} \\ 0 & -\frac{k}{L} & \frac{-R}{L} \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L} \end{bmatrix} u(t), \tag{26}$$
$$\theta(t) = \mathrm{sinc}\big(x_1(t)\big).$$

For control design, (26) is discretized using a second-order polynomial approximation [2, Chapter 6]. This polynomial approximation is more accurate than the first-order (Euler) approximation that is frequently used. Due to the particular structure of (26), 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:

$$x(k+1) = \left(\begin{bmatrix} 1.00 & 0.04 & 0.07 \\ 0 & 0.90 & 2.64 \\ 0 & -0.02 & 0.64 \end{bmatrix} + \begin{bmatrix} 0.08 & 0 & 0 \\ 3.81 & 0.08 & 0 \\ -0.05 & 0 & 0 \end{bmatrix} \theta(k) \right) x(k) + \begin{bmatrix} 0 \\ 0.69 \\ 0.32 \end{bmatrix} u(k), \tag{27}$$
$$\theta(k) = \mathrm{sinc}\,\big(x_1(k)\big)\,.$$

The system is subject to the constraints

$$\mathbb{X} = \left\{ x \in \mathbb{R}^3 \mid |x_1| \leq 2\pi, |x_2| \leq 18, |x_3| \leq 6 \right\},$$
$$\mathbb{U} = \left\{ u \in \mathbb{R} \mid |u| \leq 8 \right\}.$$

For the simulation, the following settings are used:

- The prediction horizon and tuning parameters are set to $N = 6$, $Q = \mathrm{diag}\{100, 1, 1\}$, and $R = 10$. An infinity-norm based worst-case cost (11) is used.
- A 0.98-contractive terminal set X_f is computed using the algorithm of [37], with $\breve{\mathbb{X}} = \left\{ x \in \mathbb{R}^3 \mid |x_1| \leq \frac{1}{2}\pi, |x_2| \leq 18, |x_3| \leq 2 \right\}$. The set X_f is contractive under a robustly stabilizing linear state feedback $u = Kx$, and has 24 vertices.
- The initial state was $x_0 = \begin{bmatrix} \pi & 0 & 0 \end{bmatrix}^\top$ (stable equilibrium).
- The initial state constraints sequence and the corresponding scheduling sets are generated using the bounded ROV method of Section 4.2 with $\delta = \begin{bmatrix} \frac{1}{4}\pi & \infty & \infty \end{bmatrix}^\top$. Because in (27) the scheduling map $\mu(\cdot)$ is only dependent on x_1, it is only necessary to bound the ROV of x_1.
- The non-linear continuous-time model (25) is used as the real controlled system.

The following controllers are compared:

TMPC: The controller from this paper with the infinity-norm cost function (11).

qLPV-MPC: The iterative approach of [10].

NMPC: A non-linear MPC optimizing a quadratic stage cost, but using the same terminal cost and terminal constraint as the 'TMPC' controller.

TMPC Heterogeneous: The tube-based controller from [16] with the same infinity-norm cost function (11). In [16], there is no mechanism to take into account the relation between the scheduling and state variables. For that reason the static scheduling set $\Theta_\mathrm{full} = [\min_x \mu(x),\ \max_x \mu(x)] = [-0.22,\ 1]$ was used to guar-

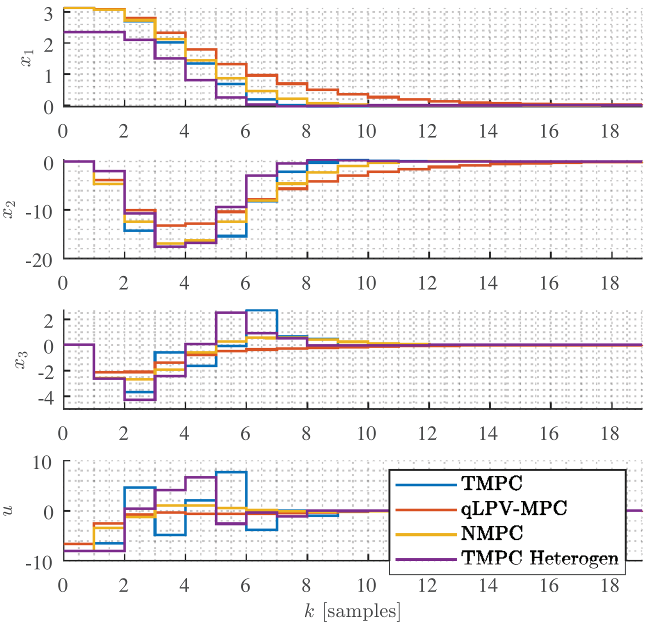

FIGURE 3 Simulation result for the electrically driven inverted pendulum

TABLE 2 Performance summary for the pendulum example. Note that 'TMPC Heterogeneous' used a different initial state

	RMS			Total variance
Method	x_1	x_2	x_2	u
TMPC	1.29	7.51	1.26	52.3
qLPV-MPC	1.41	5.88	0.78	7.13
NMPC	1.31	6.98	0.97	10.5
TMPC Heterogeneous	0.96	6.35	1.39	28.0

TABLE 3 Summary of solver computation times per sample

Method	Average (ms)	Max (ms)
TMPC	63	95
qLPV-MPC	3.3	7.4
NMPC	263	930
TMPC Heterogeneous	65	85

antee recursive feasibility. With this scheduling set, no feasible solution was found for the initial state x_0, so a different initial state $\tilde{x}_0 = 0.75x_0$ was used. Constructing the sequence of scheduling sets based on a bounded ROV was not possible with [16], because this would lead to infeasibility after a few samples due to the neglected relation between scheduling and state variables. The framework of [16] allows considerable freedom in choosing the parameterization of the synthesized state tube. For this example, a 'scenario tube' parameterization was employed for the first 3 prediction time instances, with the subsequent prediction instances using the homothetic tube parameterization that is also presented in the current paper.

The obtained simulation result is shown in Figure 3. All controllers drive the pendulum to its upright unstable equilibrium position $x_1 = q = 0$ while respecting the constraints.

In terms of the obtained state response, the controllers behave similarly, taking note that the 'TMPC heterogeneous'-controller starts from a different initial state. The control inputs applied by both TMPC-based controllers are noticeably more aggressive than with the qLPV and NMPC. This difference is mainly because the tube-based controllers optimize an infinity-norm based cost, whereas the other controllers use a quadratic cost. An example comparing the difference between the infinity-norm and quadratic cost in the TMPC is given in Section 6.2.

The RMS value of the closed-loop state trajectories and the total variance [1] of the control input are summarized in Table 2 to give an idea of the relative closed-loop performance of the considered controllers. Note that a direct comparison of these numbers is not necessarily meaningful because all controllers aim to optimize a different cost function under different assumptions.

For the TMPC controller, 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 4–5 in this case is really necessary.

The 'TMPC heterogeneous' controller from [16] also reported infeasibility for the considered initial state x_0. When using a sequence of scheduling sets based on the assumption of a bounded ROV, this controller can give an initially feasible solution only to report infeasibility later when the state has evolved in a way that is no longer consistent with the assumed sequence of scheduling sets. This is the reason that with the 'TMPC heterogeneous' controller from [16], the system could only be stabilized for a smaller set of initial states. The undesirable situation where feasibility is lost after a few samples cannot occur with the TMPC controller developed in the current paper, because the relationship between scheduling and state variables is taken into account explicitly.

Furthermore, it was found that for some combinations of tuning parameters, for example $Q = \text{diag}\{400, 4, 4\}$ and $R = 1$, the qLPV-MPC diverges after a few samples and fails to stabilize the system. In contrast, the TMPC we proposed provides an a priori stability guarantee and is therefore not susceptible to this type of issue.

To get an idea of the relative computational complexity of the different approaches, the solver computation times per sample are summarized in Table 3. The results were obtained with a 2009 Intel Core i7 CPU at 2.8 GHz, with Gurobi 8.1 being the QP and LP solver, and `fmincon` from MATLAB 2018b the non-linear solver.

[1] The RMS value of a signal with a length of n samples is defined as $\text{RMS}(x) = \sqrt{\frac{1}{n}\sum_{k=1}^{n} x^2(k)}$ and the total variance is $\text{TV}(x) = \sum_{i=1}^{n-1} |x(k+1) - x(k)|$.

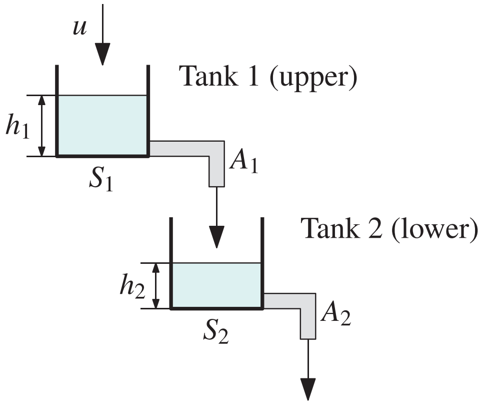

FIGURE 4 The two-tank system

TABLE 4 Parameters of the two-tank system, the same as used in [8]

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	kgcm^{-3}

6.2 | Two-tank system

The two-tank system from [8, 49] is depicted in Figure 4. 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 of 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

$$\begin{aligned} \rho S_1 \dot{h}_1 &= -\rho A_1 \sqrt{2gh_1} + u, \\ \rho S_2 \dot{h}_2 &= \rho A_1 \sqrt{2gh_1} - \rho A_2 \sqrt{2gh_2}, \end{aligned} \tag{28}$$

with the parameters shown in Table 4.

The input flow is subject to a constraint

$$\mathbb{U} = \{u \mid 0 \leq u \leq \overline{u}\}$$

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 5.

TABLE 5 Constraints of the two-tank system, the same as used in [8]

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

In [8], it is shown that the non-linear dynamics (28) can be embedded in a (continuous-time) LPV-SS representation by introducing the scheduling variables

$$\begin{aligned} \theta_1(h_1) &= \frac{1/\sqrt{h_1} - 1/\sqrt{\underline{h}_1}}{1/\sqrt{\overline{h}_1} - 1/\sqrt{\underline{h}_1}}, \\ \theta_2(h_2) &= \frac{1/\sqrt{h_2} - 1/\sqrt{\underline{h}_2}}{1/\sqrt{\overline{h}_2} - 1/\sqrt{\underline{h}_2}}, \end{aligned} \tag{29}$$

where the scheduling set corresponding to (29) is

$$\Theta = \left\{ \begin{bmatrix} \theta_1(h_1) \\ \theta_2(h_2) \end{bmatrix} \middle| \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \in \mathbb{X} \right\} = \left\{ \theta \in \mathbb{R}^2 \middle| 0 \leq \theta \leq 1 \right\}. \tag{30}$$

After discretizing the resulting continuous-time embedding using the Euler approach with a sampling time of $\tau = 0.9$ s, the discrete-time LPV embedding

$$\begin{aligned} x(k+1) = &\left(\begin{bmatrix} 0.86 & 0 \\ 0.22 & 0.97 \end{bmatrix} + \begin{bmatrix} 0.12 & 0 \\ -0.19 & 0 \end{bmatrix} \theta_1(k) + \right. \\ &\left. \begin{bmatrix} 0 & 0 \\ 0 & 0.02 \end{bmatrix} \theta_2(k) \right) x(k) + \begin{bmatrix} 0.36 \\ 0 \end{bmatrix} u(k), \\ \theta(k) = &\ \mu(x(k)), \end{aligned} \tag{31}$$

is obtained where $\mu(\cdot)$ is the scheduling map corresponding to (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 (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 [8], 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 are set to $Q = \text{diag}\{0, 2\}$, $R = 0.1$, and $P = 5$ for the centre-based stage cost (12);
- For the tube-based MPC, a controlled 0.995-contractive terminal set X_f with a fixed representation complexity 6 vertices is computed using the method in [36, Appendix A] with $\check{\mathbb{X}} = \mathbb{X}$;
- The tube-based MPC is initialized according to the bounded ROV approach of Section 4.2, with $\delta = \begin{bmatrix} 2 & 25 \end{bmatrix}^\top$;
- With the scheduling map $\mu(\cdot)$ from (31), even though it is not an affine function, the scheduling set sequence can be computed by evaluating $\mu(\cdot)$ at the vertices of the constraints sets as done in Section 5.1.

For the two-tank system, four different simulations are executed with the following corresponding MPC designs:

Casavola: The controller from [8], with $N = 2$ and $x_0 = \begin{bmatrix} 15 & 40 \end{bmatrix}^\top$.

TMPC 1: The tube-based MPC from this paper with infinity-norm cost function (11), and with $N = 4$ and $x_0 = \begin{bmatrix} 15 & 40 \end{bmatrix}^\top$. This simulation allows to compare the behaviour of the TMPC with an infinity-norm cost function to [8], which uses a quadratic cost.

TMPC 2: The tube-based MPC from this paper with infinity-norm cost function (11), and with $N = 4$ and $x_0 = \begin{bmatrix} 8 & 40 \end{bmatrix}^\top$. This represents an initial state for which [8] 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 paper with quadratic centre-based cost function, and with $N = 10$ and $x_0 = \begin{bmatrix} 15 & 40 \end{bmatrix}^\top$. This compares the behaviour of the TMPC with a quadratic cost function and a longer horizon to [8].

The suggested correction from [9] to the algorithm [8] 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 [8]. 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 [8] is run with an assumed arbitrarily fast scheduling ROV.

The simulation results are depicted in Figure 5. 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 [8], which uses a short horizon. Note that the computational complexity of [8] grows exponentially in the horizon length: therefore, its performance for longer horizons was not investigated.

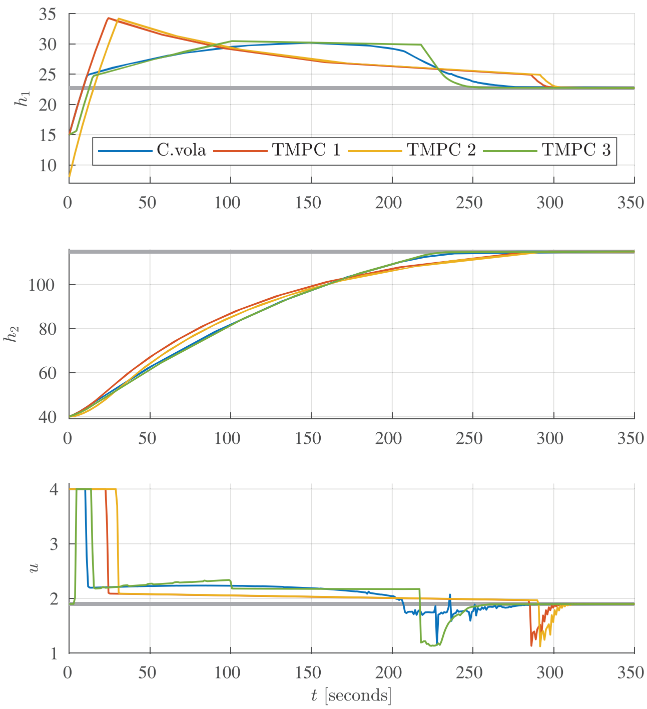

FIGURE 5 Closed-loop state and input trajectories in the two-tank example

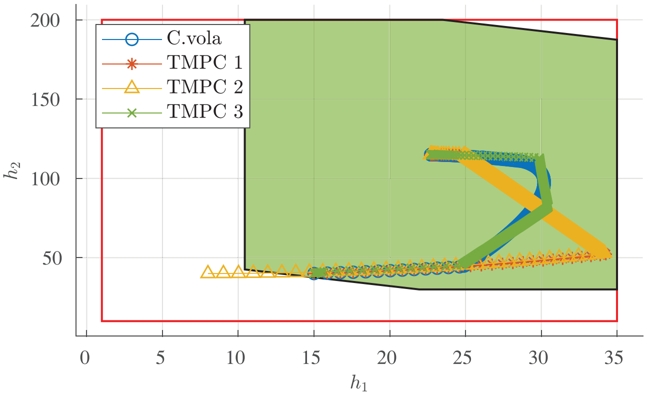

FIGURE 6 Closed-loop state trajectories for the two-tank example in a phase plot

For the purpose of further illustration, the realized closed-loop trajectories are shown in a phase plot in Figure 6. The initial tube for 'TMPC 2' is shown in Figure 7, together with the generated state constraint sets: as expected, every tube cross section X_i is contained inside of the corresponding constraint set $\mathbb{X}_i$. Similar to Section 6.1, a short summary of generic performance indicators of the closed-loop trajectories is given in Table 6.

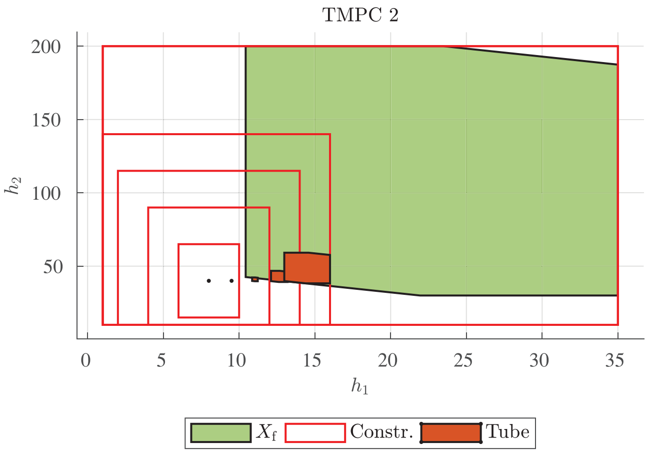

FIGURE 7 Tube $\mathbf{T}_0$ computed at $k = 0$ in the two-tank simulation 'TMPC 2'

TABLE 6 Performance summary for the two-tank example

	RMS		**Total variance**
Method	h_1	h_2	u
Casavola	26.3	104	2.86
TMPC 1	25.8	102	5.12
TMPC 2	26.7	96.8	5.16
TMPC 3	26.2	103	6.00

7 | CONCLUDING REMARKS

The authors presented a tube-based MPC strategy applicable to the control of non-linear systems embedded in an LPV-SS representation. 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 [5, 8], 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.

To enlarge the applicability of the approach, and to fully exploit the advantages of the proposed setting, the development of improved methods of selecting the initial state constraint sequences should be considered. Related to this, improved and computationally efficient ways for generating the associated scheduling sets should also be investigated. Extension of the method to LPV embeddings with input non-linearities is also of interest.

Lastly, an extension of the method to the tracking of piecewise-constant reference signals can be considered. This would require the computation of a new state constraint sequence and associated scheduling set sequence every time that the reference changes. Further research is necessary to investigate how this can be done efficiently and how convergence of the state of the controlled non-linear system to the reference can be guaranteed.

ACKNOWLEDGEMENTS

This work was supported by the Impulse 1 program of the Eindhoven University of Technology and ASML, and has also received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 714663). The Authors thank Shengling Shi for sharing his implementation of qLPV-MPC used in Example 1.

REFERENCES

1. Rugh, W.J., Shamma, J.S.: Research on gain scheduling. Automatica 36, 1401–1425 (2000)
2. Tóth, R.: Modeling and Identification of Linear Parameter-Varying Systems. Springer, New York (2010)
3. Abbas, H.S. et al.: Embedding of nonlinear systems in a linear parameter-varying representation. In: Proceedings of the 19th IFAC World Congress, Cape Town (2014)
4. Morato, M.M., Normey-Rico, J.E., Sename, O.: Model predictive control design for linear parameter varying systems: A survey. Annual Reviews in Control 49, 64–80 (2020)
5. Lu, Y., Arkun, Y.: A scheduling quasi-min-max model predictive control algorithm for nonlinear systems. J. Process Control 12, 589–604 (2002)
6. Chisci, L., Falugi, P., Zappa, G.: Gain-scheduling MPC of nonlinear systems. Int. J. Robust Nonlinear Control 13, 295–308 (2003)
7. Casavola, A., Famularo, D., Franzè, G.: A feedback min-max MPC algorithm for LPV systems subject to bounded rates of change of parameters. IEEE Trans. Autom. Control 47, 1147–1153 (2002)
8. Casavola, A., Famularo, D., Franzè, G.: Predictive control of constrained nonlinear systems via LPV linear embeddings. Int. J. Robust Nonlinear Control 13, 281–294 (2003)
9. Ding, B.-C., Huang, B.: Comments on a feedback min-max MPC algorithm for LPV systems subject to bounded rates of change of parameters. IEEE Trans. Autom. Control 52, 970–970 (2007)
10. Cisneros, P.S.G., Voss, S., Werner, H.: Efficient nonlinear model predictive control via quasi-LPV representation. In: Proceedings of the 55th IEEE Conference on Decision and Control, Las Vegas (2016)
11. Cisneros, P.S.G., Werner, H.: Parameter-dependent stability conditions for quasi-LPV model predictive control. In: Proceedings of the 2017 American Control Conference, Seattle (2017)
12. Martínez Calderón, H., Cisneros, P.S.G., Werner, H.: qLPV predictive control - a benchmark study on state space vs input-output approach. In: Proceedings of the 3rd IFAC Workshop on Linear Parameter-Varying Systems, Eindhoven (2020)
13. Morato, M.M., Normey-Rico, J.E., Sename, O.: Novel qLPV MPC design with least-squares scheduling prediction. in Proc. of the 3rd IFAC Workshop on Linear Parameter-Varying Systems, pp. 148–163.(2020)
14. Hanema, J., Tóth, R., Lazar, M.: Stabilizing non-linear MPC using linear parameter-varying representations. iIn: Proceedings of the 56th IEEE Conference on Decision and Control, Melbourne (2017)
15. Hanema, J., Lazar, M., Tóth, R.: Stabilizing tube-based model predictive control: Terminal set and cost construction for LPV systems. Automatica 85, 137–144 (2017)
16. Hanema, J., Lazar, M., Tóth, R.: Heterogeneously parameteized tube model predictive control for LPV systems. Automatica 111, 108622 (2020)

17. Koelewijn, P.J.W. et al.: Pitfalls of guaranteeing asymptotic stability in LPV control of nonlinear systems. In: Proceedings of the 2020 European Control Conference, St. Petersburg (2020)
18. Boyd, S. et al.: Linear Matrix Inequalities in System and Control Theory. Society for Industrial and Applied Mathematics, Philadelphia (1994)
19. Gielen, R.H., et al.: On polytopic inclusions as a modeling framework for systems with time-varying delays. Automatica 46, 615–619 (2010)
20. Scherer, C.W., Njio, R.G.E., Bennani, S.: Parametrically varying flight control system design with full block scalings. In: Proceedings of the 36th Conference on Decision and Control, San Diego (1997)
21. Kwiatkowski, A., Boll, M.T., Werner, H.: Automated generation and assessment of affine LPV models. In: Proceedings of the 45th IEEE Conference on Decision and Control, San Diego (2006)
22. Donida, F. et al.: Towards integrated modelling and parameter estimation: an LFT-Modelica approach. In: Proceedings of the 15th IFAC Symposium on System Identification, Saint-Malo (2009)
23. Hoffmann, C., Werner, H.: Linear parameter-varying control of complex mechanical systems. In: Proceedings of the 19th IFAC World Congress, Cape Town (2014)
24. Hoffmann, C., Werner, H.: Compact LFT-LPV modeling with automated parameterization for efficient LPV controller synthesis. In: Proceedings of the 2015 American Control Conference, Chicago (2015)
25. Hoffmann, C., Werner, H.: LFT-LPV modeling and control of a control moment gyroscope. In: Proceedings of the 54th IEEE Conference on Decision and Control, Osaka (2015)
26. Sadeghzadeh, A., Sharif, B., Tóth, R.: Affine linear parameter-varying embedding of nonlinear models with improved accuracy and minimal overbounding. IET Control Theory Appl. 14(20), 3363–3373 (2020)
27. Shamma, J.S., Cloutier, J.R.: Gain-scheduled missile autopilot design using linear parameter varying transformations. Journal of Guidance, Control, and Dynamics 16, 256–263 (1993)
28. Leith, D.J., Leithead, W.E.: Gain-scheduled and nonlinear systems: Dynamic analysis by velocity-based linearization families. Int. J. of Control 70, 289–317 (1998)
29. Petersson, D., Löfberg, J.: Optimization based LPV-approximation of multi-model systems. In: Proceedings of the European Control Conference, Budapest (2009)
30. Vízer, D., Mercère, G.: An H_∞-norm-based approach for operating point selection and LPV model identification from local experiments. Periodica Polytechnica Electrical Engineering and Computer Science 58, 121–131 (2014)
31. Vízer, D., Mercère, G., Laroche, E.: Gray-box LPV model identification of a 2-DoF surgical robotic manipulator by using an H_∞-norm-based local approach. In: Proceedings of the 1st IFAC Workshop on LPV Systems, Grenoble (2015)
32. Abbas, H.S. et al.: LPV modeling of nonlinear systems: A multi-path feedback linearization approach. Submitted to Int. J. Robust Nonlinear Control (2021)
33. Hanema, J., Tóth, R., Lazar, M.: Tube-based anticipative model predictive control for linear parameter-varying systems. In: Proceedings of the 55th IEEE Conference on Decision and Control, Las Vegas (2016)
34. Langson, W., et al.: Robust model predictive control using tubes. Automatica 40, 125–133 (2004)
35. Raković, S.V., et al.: Homothetic tube model predictive control. Automatica 48, 1631–1638 (2012)
36. Hanema, J.: Anticipative model predictive control for linear parameter-varying systems. Dissertation, Eindhoven University of Technology (2018)
37. Miani, S., Savorgnan, C.: MAXIS-G: A software package for computing polyhedral invariant sets for constrained LPV systems. In: Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, Seville (2005)
38. Aeyels, D., Peuteman, J.: A new asymptotic stability criterion for nonlinear time-variant differential equations. IEEE Trans. Autom. Control 43, 968–971 (1998)
39. Lee, Y.I., Kouvaritakis, B., Cannon, M.: Constrained receding horizon predictive control for nonlinear systems. Automatica 38, 2093–2102 (2002)
40. Cannon, M., et al.: Robust tubes in nonlinear model predictive control. IEEE Trans. Autom. Control 56, 1942–1947 (2011)
41. Althoff, M., Stursberg, O., Buss, M.: Reachability analysis of nonlinear systems with uncertain parameters using conservative linearization. In: Proceedings of the 47th IEEE Conference on Decision and Control, Cancun (2008)
42. Fukuda, K.: Frequently asked questions in polyhedral computation. https://www.cs.mcgill.ca/~fukuda/soft/polyfaq/polyfaq.html (2004). Accessed 25 March 2021.
43. Breiman, L., Cutler, A.: A deterministic algorithm for global optimization. University of California Berkeley, Technical Report, (1989)
44. Rizvi, S.Z., et al.: A Kernel-based PCA approach to model reduction of linear parameter-varying systems. IEEE Trans. Control Syst. Technol. 24, 1883–1891 (2016)
45. Jaulin, L. et al.: Applied Interval Analysis, 1st ed. Springer, New York (2001)
46. Berz, M., Hoffstätter, G.: Computation and application of taylor polynomials with interval remainder bounds. Reliable Computing 4, 83–97 (1998)
47. Althoff, M., Grebenyuk, D.: Implementation of Interval Arithmetic in CORA 2016. In: Proceedings of the 3rd International Workshop on Applied Verification for Continuous and Hybrid Systems, Vienna (2016)
48. Rump, S.M.: INTLAB - INTerval LABoratory. In: Csendes, T. (ed.) Developments in Reliable Computing, pp. 77–104.Kluwer Academic Publishers, Amsterdam (1999)
49. Angeli, D., Casavola, A., Mosca, E.: Constrained predictive control of nonlinear plants via polytopic linear system embedding. Int. J. Robust Nonlinear Control 10, 1091–1103 (2000)

How to cite this article: Hanema J, Tóth R, Lazar M. Stabilizing non-linear model predictive control using linear parameter-varying embeddings and tubes. IET Control Theory Appl. 2021;15:1404–1421. https://doi.org/10.1049/cth2.12131

APPENDIX: REALIZATION OF THE TUBE SYNTHESIS PROBLEM

In this appendix, the LP or QP realizing the tube synthesis problem (9) is constructed. Whether (9) is a LP or QP depends on the chosen stage cost.

Cross section parameterization. According to Definition 6, each tube cross section is characterized by a scaling $\alpha_i \in \mathbb{R}_+$ and a centre $z_i \in \mathbb{R}^{n_x}$ which are decision variables. The scalings are subject to

$$\forall i \in [0..N] : \ \alpha_i \geq 0. \tag{A.1}$$

Initial condition constraint. The constraint $X_0 = \{x\}$ in (8) is implemented by the equality constraints

$$z_0 = x, \ \alpha_0 = 0. \tag{A.2}$$

Tube dynamics constraints. The transition constraints $\mathcal{I}(X_i, \Theta_i | K_i) \subseteq X_{i+1}$ from Definition 5 are equivalent to

$$\forall i \in [0..N-1] : \ \forall (j,l) \in [1..q_\mathrm{s}] \times [1..q_{\Theta i}] :$$
$$H_\mathrm{s}\left(A(\bar{\theta}_i^l)\left(z_i + \alpha_i \bar{s}^j\right) + Bu_i^{(j,l)}\right) \leq 1 \tag{A.3}$$

where $u_i^{(j,l)} \in \mathbb{R}^{n_\mathrm{u}}$ are decision variables according to the vertex control parameterization of Definition 6. Vertex representations of the scheduling sets $\Theta_i = \operatorname{convh}\{\bar{\theta}^1, \dots, \bar{\theta}^{q_{\Theta i}}\}$ are assumed to be available. For the set S from Definition 6, both vertex and hyperplane representations $S = \operatorname{convh}\{\bar{s}^1, \dots, \bar{s}^{q_\mathrm{s}}\} = \{x \mid H_\mathrm{s} x \leq 1\}$ are used.

State constraints. The state constraints $X_i \subseteq \mathbb{X}_i$ from Definition 5 are equivalent to

$$\forall i \in [0..N-1] : \forall j \in [1..q_\mathrm{s}] : H_{\mathrm{x}i}\left(z_i + \alpha_i \bar{s}^j\right) \leq 1 \qquad \text{(A.4)}$$

if hyperplane representations $\mathbb{X}_i = \{x \mid H_{\mathrm{x}i} x \leq 1\}$ are available.

Input constraints. The input constraints are realized by

$$\begin{aligned} &\forall i \in [0..N-1] : \forall (j,l) \in [1..q_\mathrm{s}] \times [1..q_{\Theta i}] : \\ &H_\mathrm{u} u_i^{(j,l)} \leq 1. \end{aligned} \qquad \text{(A.5)}$$

if a hyperplane representation $\mathbb{U} = \{u \mid H_\mathrm{u} u \leq 1\}$ is available.

Terminal constraint. The terminal constraint $X_N \subseteq X_\mathrm{f}$ from (8) is equivalent to

$$\begin{aligned} \forall j \in [1..q_\mathrm{s}] : H_\mathrm{f}\left(z_N + \alpha_N \bar{s}^j\right) &\leq \gamma, \\ 0 \leq \gamma &\leq 1, \end{aligned} \qquad \text{(A.6)}$$

where $\gamma \in \mathbb{R}_+$ is a decision variable and a hyperplane representation $X_\mathrm{f} = \{x \mid H_\mathrm{f} \leq 1\}$ is used.

Terminal cost. The decision variable $\gamma \in \mathbb{R}_+$ introduced before corresponds to $\Psi_{X_\mathrm{f}}(X_N)$. Therefore, $F(X_N) = \frac{\gamma}{1-\lambda}\bar{\ell}$ with $\bar{\ell}$ the value corresponding to the maximization in (13).

Full LP with worst-case stage cost. We give the construction for $c = \infty$ in (11), as used in the numerical examples. The cost (11) can be implemented by introducing non-negative slack variables (μ, ν, ζ). Then, $\ell^{\mathrm{wc}}(X_i, K_i, \Theta_i) \leq \zeta_i$ is equivalent to $\forall i \in [0..N-1]$:

$$\begin{aligned} &\forall (j,l) \in [1..q_\mathrm{s}] \times [1..q_{\Theta i}] : -\nu_i^{(j,l)} \leq R u_i^{(j,l)} \leq \nu_i^{(j,l)},\ \nu_i^{(j,l)} \geq 0 \\ &\forall j \in [1..q_\mathrm{s}] : -\mu_i^j \leq Q\left(z_i + \alpha_i \bar{s}^j\right) \leq \mu_i^j,\ \mu_i^j \geq 0 \\ &\forall (j,l) \in [1..q_\mathrm{s}] \times [1..q_{\Theta i}] : \nu_i^{(j,l)} + \mu_i^j \leq \zeta_i. \end{aligned} \qquad \text{(A.7)}$$

The total LP realizing (9) becomes

$$\min_{\mathbf{d}} \frac{\gamma}{1-\lambda}\bar{\ell} + \sum_{i=0}^{N-1} \zeta_i$$

subject to (A.1)–(A.6), (A.7)

with $\mathbf{d}$ being the collection of all decision variables introduced above. By introducing appropriate slack variables and constraints, a similar construction can be made for the case $c = 1$.

Full QP with centre-based stage cost. We give the construction for $p = 2$ in (12), as used in the numerical examples. The total QP realizing (9) then becomes

$$\begin{aligned} \min_{\mathbf{d}} \frac{\gamma}{1-\lambda}\bar{\ell} + \sum_{i=0}^{N-1} \Bigg(& z_i^\top Q^\top Q z_i \\ & + P\alpha_i^2 + \frac{1}{q_{\Theta i} q_\mathrm{s}} \sum_{j=1}^{q_\mathrm{s}} \sum_{l=1}^{q_{\Theta i}} \left[u_i^{(j,l)}\right]^\top R^\top R u_i^{(j,l)} \Bigg) \end{aligned}$$

subject to (A.1)–(A.6)

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