

# Model Predictive Control Design for Linear Parameter Varying Systems: A Survey

Marcelo M. Morato<sup>a</sup>, Julio E. Normey-Rico<sup>a</sup>, Olivier Sename<sup>b</sup>

<sup>a</sup>*Renewable Energy Research Group (GPER), Departamento de Automação e Sistemas (DAS),  
Universidade Federal de Santa Catarina, Florianópolis, Brazil.*

<sup>b</sup>*Grenoble INP<sup>†</sup>, CNRS, GIPSA-lab, 38000 Grenoble, France.*

<sup>†</sup>*Institute of Engineering, Université Grenoble-Alpes.*

---

## Abstract

Motivated by the fact that many nonlinear plants can be represented through Linear Parameter Varying (LPV) embedding, and being this framework very popular for control design, this paper investigates the available Model Predictive Control (MPC) policies that can be applied for such systems. This paper reviews the available works considering LPV MPC design, ranging from the sub-optimal, simplified, yet Quadratic Programming (QP) algorithms, the tube-based tools, the set-constrained procedures, the Nonlinear Programming procedures and the robust ones; the main features of the recent research body on this topic are examined. A simulation example is given comparing some of the important techniques. Finally, some suggestions are given for future investigation threads, seeking further applicability of these methods.

*Keywords:* Model Predictive Control, Linear Parameter Varying Systems, Optimization, Quadratic Programming, Survey.

---

## 1. Introduction

Control System literature has significantly grown over the course of the last decades. Reference tracking, output regulation, disturbance rejection, noise attenuation and other topics have been studied for various systems, with diverse kinds of methodologies.

### 5 1.1. Model Predictive Control

In such a way, Model Predictive Control (*MPC*) has become a very well-established and widespread technique (Camacho & Bordons, 2013), with a wide range of industrial applications, counting more than 5800 successful results in many different areas (Alamir, 2013), including processes in chemical, mechanical and thermic systems (Feng et al., 2007). A survey of some of these industrial applications is presented in  
10 (Qin & Badgwell, 2003).

Sometimes named moving/sliding horizon control, the *MPC* concept is a straightforward method for the optimal control of processes subject to constraints (Zhou et al., 1996; Normey-Rico & Camacho, 2007). The basic *MPC* algorithm computes an optimal control action  $u$  at each discrete instant  $k$  by solving an optimization problem, which is written in terms of a process prediction model<sup>1</sup>. The optimization cost  
15 function is used to consider performance goals, such as reference tracking and disturbance rejection. This framework allows to explicitly consider the effect of input, output and state constraints in the control design procedure, which is rather convenient. The remaining advantages of *MPC* design are: that it can be applied

---

*Email addresses:* marcelomnzm@gmail.com (Marcelo M. Morato), julio.normey@ufsc.br (Julio E. Normey-Rico), olivier.sename@grenoble-inp.fr (Olivier Sename)

<sup>1</sup>Of course, nowadays *MPC* has also been generalized for model-free paradigms, which are often referred to as “data-driven” or “learning” algorithms. These are not under the scope of this paper.

to multivariable (MIMO), coupled dynamical processes; that optimality concerns are directly implied by the design procedure itself; and that it has a systematic design methodology.

The core principle of *MPC* design is that a process model is needed so that the future outputs can be predicted. Therefore, although being indisputably useful, the original *MPC* algorithms (such as DMC<sup>2</sup> and GPC<sup>3</sup> algorithms) were linked to the context of processes with linear time-invariant (*LTI*) models, using impulse-response or state-space formulations.

The design of *MPC* for *LTI* plants is very well established. These algorithms are today available through many alternative formulations and include diverse variations (feedforward capabilities, economic purposes, logic constraints, etc.). In (Mayne, 2014), the future promise of *LTI* predictive control and some of these recent developments are discussed. Some other *MPC* surveys may also interest the readers: (Scattolini, 2009) discussed application architectures, (Garriga & Soroush, 2010) surveyed the different tuning methodologies and (Yu-Geng et al., 2013) debated the status and challenges of this control framework.

To verify that these *LTI MPC* algorithms ensure closed-loop asymptotic stability properties (and recursive feasibility of the optimization procedure), literature points out to two main paths, detailed in the sequel.

The first is to rely on the use of so-called “terminal ingredients” of the optimization procedure: a terminal offset cost  $V(\cdot)$  is included to the optimization cost function  $J = V(\cdot) + \sum_{i=0}^{N_p} \ell(\cdot)$ , weighting the terminal state variable, and a terminal constraint, defining a stable feasibility region for this terminal variable (which implies in a stabilizing terminal feedback controller). The idea of these terminal ingredients coupled to the optimization dates 1988 (Keerthi & Gilbert, 1988), but was formally generalized to MPC in 1993 (Michalska & Mayne, 1993). The terminal constraints often require long prediction horizons, which may borderline increase the computational complexity of these algorithms. The assumptions required by these terminal ingredients are somewhat simple, they refer to a  $\mathcal{K}$ -class lower bound on  $\ell(\cdot)$ , a  $\mathcal{K}$ -class upper bound on  $V(\cdot)$  and to a Lyapunov-decreasing characteristic for  $V(\cdot)$ , which must decay along the horizon. In (Mayne et al., 2000), a concrete overview of many results using these terminal ingredients is presented. In (Löfberg, 2012), different tests are proposed, through these terminal ingredients, to verify recursive feasibility properties.

The second option is to use dissipativity arguments (Seiler et al., 2010). In (Heath et al., 2005), the use of Integral Quadratic Constraints (IQC) is applied to *LTI MPC*. The IQCs are demonstrated under some sector boundary conditions. For *LTI* systems with uncertain parameters, the IQCs can be extended trivially (Heath et al., 2006), while the resulting conservativeness can be addressed with Zames-Falb multipliers, see (Heath & Wills, 2005; Heath & Li, 2010). The main characteristic of this second stability-verification path is that they are Linear Matrix Inequality (*LMI*) formulations. No terminal ingredients are necessary, which means that online complexity and feasibility are not affected by such proofs. For the *LTI* case, the proofs are provided *a priori* to implementation, which is rather convenient.

The complete standard linear *MPC* algorithm is presented in the sequel. The linear *MPC* design procedure resides in solving the following<sup>4</sup> Quadratic Programming Problem (*QP*):

**Problem 1.**

$$\min_U J = \min_U \sum_{i=0}^{N_p} \overbrace{\ell(x(k+i|k), y(k+i|k), u(k+i-1|k))}^{\text{MPC Cost}} \quad (1)$$

$$\text{s.t.} \quad \underbrace{x(k+i) = Ax(k) + Bu(k), \quad y(k) = Cx(k) + Du(k)}_{\text{Linear System Model}}, \quad (2)$$

$$u(k+i-1|k) \in \mathcal{U}, \quad (3)$$

$$x(k+i|k) \in \mathcal{X}, \quad (4)$$

$$y(k+i|k) \in \mathcal{Y}. \quad (5)$$

<sup>2</sup>Dynamic Matrix Control, as proposed by Cutler & Ramaker (1980).

<sup>3</sup>Generalized Predictive Control, as proposed by Clarke et al. (1987).

<sup>4</sup>Notation  $(k+i|k)$  is used to represent a predicted value for instant  $k+i$ , computed at instant  $k$ . For now, the presence of disturbances is suppressed, for simplicity.

It is implied that  $U = \text{col}\{u(k|k), \dots, u(k + N_p - 1|k)\}$  is the sequence of actions inside the (sliding) prediction horizon  $N_p$ . Often, a different sliding horizon is chosen for the control signal (control horizon  $N_c$ , with  $N_c < N_p$ ). As previously explained, sometimes a terminal stage cost is also minimized; the optimization may as well include terminal constraints, i.e.  $x(k + N_p|k) \in \mathbb{X}_{N_p}$ . Throughout this work, take  $x \in \mathbb{R}^x$ ,  $u \in \mathbb{R}^u$  and  $y \in \mathbb{R}^y$ , with  $\mathcal{U}$ ,  $\mathcal{X}$  and  $\mathcal{Y}$  as the set constraints that define feasibility.

### 1.2. Nonlinear Model Predictive Control

It seems evident that when nonlinear systems are controlled over larger operating conditions or when the process responses heavily depend on external parameters, the original *LTI MPC* design method of Problem 1 must be adapted, because an *LTI* model cannot truly represent the controlled process.

The concept of *MPC* itself is not restricted to those systems with linear time-invariant models, and has been progressively extended for systems with nonlinear dynamics. Unfortunately, including nonlinear model predictions and nonlinear constraints to an optimization problem is not trivial and much increases its complexity (Allgöwer & Zheng, 2012).

Up until the late 10's, only a few efficient nonlinear *MPC* (denoted hereafter as *NMPC*) algorithms were available, as reviewed in (Cannon, 2004): the majority of these algorithms were not able to be computed in real-time for embedded applications and require excessive computational capacities. This issue is clearly stated by all reviews in *NMPC* design from then, see (Alamir, 2006; Camacho & Bordons, 2007).

To overlap the complexity of these “full-blown” *NMPC* procedures, many sub-optimal solutions have been presented, such as: *a*) approaches that overcome the nonlinearities via linearization along the prediction trajectory (Ayala et al., 2011); *b*) methods that manipulate the nature of the nonlinearities to represent them in some specific manner (as bilinearities or sector conditions, for instance) which may reduce complexity (Nguyen et al., 2016); and *c*) algorithms that parametrize the control inputs in finite discrete possibilities, solving a search algorithm to find the smallest  $J$  instead of the actual optimization procedure (Rathai et al., 2018; Alamir, 2012). A brief survey on these works has been recently performed by Zhang et al. (2019).

In spite of these sub-optimal *NMPC* strategies, some recent solver-bases solutions have been presented to fasten the operation of *NMPC* loops (Bock et al., 2007), which must be highlighted: *a*) the real-time iteration methods, such as ACADO (Houska et al., 2011; Quirynen et al., 2015), convert the *NMPC* problem into a multiple-shooting discretization optimization accompanied by a generalized Gauss-Newton problem and are able to solve some *NMPC* problems within a microsecond range, as detailed in (Gros et al., 2020); *b*) gradient-based methods, such as GRAMPC (Richter et al., 2011; Käpernick & Graichen, 2014; Englert et al., 2019), which convert the *NMPC* through augmented Lagrangians, being also quite fast; and *c*) some other tools, such as CasADi (Andersson et al., 2019) and FPGA-based *NMPC*s (Xu et al., 2015), but these are not so widespread.

**Remark 1.** Regarding stability, the same procedures of *LTI MPC* can be followed, with some adequate modifications to include the nonlinearities (Magni & Scattolini, 2004): the terminal ingredients are computed with respect to some equilibria of the nonlinear process (reference tracking steady-state) (Lazar & Spinu, 2015; Köehler et al., 2019), while the dissipativity formulations, such as the IQCs, are computed with a restrain on the nonlinearities (such as to be sector bounded, or structurally exploited, including homotopy) (Cisneros & Werner, 2018).

### 1.3. Linear Parameter Varying Systems

In the last decade, in parallel to the growth and establishment of predictive control applications, the Linear Parameter Varying (*LPV*) system framework has become very popular to model processes with complex dynamics (Tóth, 2010; Tóth et al., 2011b; Hoffmann & Werner, 2014).

One can say that *LPV* embedding is somewhere in between the nonlinear and the *LTI* formalisms. These systems are linear in the state space, but nonlinear in the parameter space. Differently than in the *LTI* case, their linear state transition map depends on endogenous/exogenous scheduling variables, denoted by  $\rho$ . The scheduling parameters are bounded and known (their current values  $\rho(k)$  can be measured online or

estimated) - note that their future behaviour is generally not known. Theoretical analysis of *LPV* system properties, such as stability, observability, controllability, often falls into the framework of linear time-varying systems or of nonlinear systems, which usually presents more difficulty compared to the classical *LTI* framework (Blanchini & Miani, 2003; Bokor & Balas, 2005; Szabó & Bokor, 2018).

Many nonlinear processes can be described under an *LPV* formalism, as long as Linear Differential Inclusion (LDI) is respected, see (Boyd et al., 1994; Abbas et al., 2014). Consider the generic nonlinear system below, with states  $x \in \mathbb{R}^{n_x}$ , measured outputs  $y \in \mathbb{R}^{n_y}$  and control signal  $u \in \mathbb{R}^{n_u}$ :

$$\begin{aligned} x(k+1) &= f_x(x(k), u(k)), \\ y(k) &= f_y(x(k), u(k)). \end{aligned} \quad (6)$$

Then, the LDI property enables one to generate either state-space (SS) or Input/Output (I/O) LPV formulations for this nonlinear system. This property is as follows: for each  $x$ ,  $y$  and  $u$  and every instant  $k$ , there exists a matrix  $G(x, u, k) \in \mathcal{G}$  such that:

$$\begin{bmatrix} f_x(x(k), u(k)) \\ f_y(x(k), u(k)) \end{bmatrix} = G(x, u, k) \begin{bmatrix} x(k) \\ u(k) \end{bmatrix}, \quad (7)$$

where  $\mathcal{G} \subseteq \mathbb{R}^{(n_x+n_y) \times (n_x+n_u)}$ .

For nonlinear systems that the LDI property holds (i.e.  $G(x, u, k)$  indeed exists), the SS and I/O LPV formulations can be found as detailed next. Regarding a SS formulation, Eq. (7) can be directly re-written as:

$$\begin{aligned} x(k+1) &= A(\rho(k))x(k) + B(\rho(k))u(k), \\ y(k) &= C(\rho(k))x(k) + D(\rho(k))u(k), \\ \rho(k) &\in \mathcal{P}, \end{aligned} \quad (8)$$

where the scheduling parameters  $\rho$  and their respective set  $\mathcal{P}$  derive from  $G(x, u, k)$ . In the I/O representation case, Eq. (7) is re-casted as the following difference equation:

$$\mathcal{S} : \left( \mathbb{I}_{n_y} + \sum_{i=1}^{n_a} a_i(\rho(k))z^{-i} \right) y(k) = \sum_{j=0}^{n_b} b_j(\rho(k))z^{-j}u(k), \quad (9)$$

where  $z^{-1}$  the time-shift operator and matrices  $a_i(\cdot)$  and  $b_i(\cdot)$  derive from  $G(x, u, k)$ . In this formulation, the states variables are no longer of importance, being embedded directly into the I/O structure itself. An example of how a nonlinear system, respecting LDI, can be cast into SS or I/O formulations is given in Appendix A. Note that this formulation is similar to the *LTI* one for MIMO *GPC* algorithms, when sorted with respect to time-domain.

Indeed, in the vast majority of industrial applications regarding LPV systems, the LDI property is used to recast a nonlinear model into an (SS or I/O) LPV formulation (Hoffmann & Werner, 2014). In the majority of cases, the scheduling variable is written as an endogenous function of the internal variables of the system, such as:

$$\rho(k) = f_\rho(x(k), u(k)) \in \mathcal{P}, \quad (10)$$

where  $f_\rho(\cdot) : \mathcal{X} \times \mathcal{U} \rightarrow \mathcal{P}$  is possibly nonlinear and whose form and class derives from  $G(x, u, k)$ . Eventually, the scheduling variable is not taken as a function of the process variables, willingly treated as a completely exogenous variables, independent from the result of the signals (Shamma, 2012). In such case,  $G(\cdot)$  is written as a function of these exogenous signals, which yields a model that is also linear in the parameter space with respect to the process variables, at the expense of some linearity conservativeness of the embedding.

135 **Remark 2.** Some authors differentiate these *LPV* models with respect to two classes, see (Shamma, 2012):  
a) quasi-*LPV* models, for which the scheduling parameters are taken as a function of endogenous variables,  
as in Eq. (10); and b) those for which the scheduling variables are purely exogenous. In this paper,  
this kind of classification is not used, since, regarding the application of MPC, the *LPV* embedding of  
the nonlinear system (whether performed with endogenous or exogenous scheduling parameters) is treated  
140 equally. Anyhow, these differences are typically important for stability analysis, which is rendered as a  
nonlinear program for the first class while verified through a robust procedure for the latter category. Note  
that stability analysis of nonlinear and parameter dependent *MPC* algorithms (which these both *LPV*  
categories fall under) can be verified through dissipativity formulations, as proposed in (Cisneros & Werner,  
2018).

#### 145 1.4. MPC for *LPV* Systems

The design of *LPV* control and observation is nowadays standard using *LFTs* (Casella & Lovera, 2008;  
Scherer, 2001), for  $H_2$  (Wu et al., 1997) and  $H_\infty$  (Mohammadpour & Scherer, 2012; Sename et al., 2013)  
regulation performance objectives, as well as for tracking and rejection (Scorletti et al., 2015). But the case  
is not true for predictive control applications.

150 Since many nonlinear systems can be represented with the *LPV* formalism, the study of *MPC* policies  
for these processes has been formally deployed by the Control Systems community in the beginning of the  
00's. Anyhow, there are still some open literature gaps to be properly researched (Bachnas et al., 2014).

The *LPV* framework certainly Hence the introduction has to be sharpened as it gives a picture which  
might have been true 20-15 years ago, but not today

155 Since *LPV* systems retain the linearity property from inputs to outputs, it is possible to formulate  
computationally efficient design procedures for such systems (such as polytopic approaches). This means  
that the drawbacks of full-blown *NMPC* algorithms can be avoided.

For *MPC* framework, it is imperiously necessary for one to be able to describe the response of the system  
in the future ( $N_p$  steps ahead, from the view of instant  $k$ ). To use an *LPV* model of the system to predict  
160 its future outputs  $y(k+j|k)$  is a procedure that depends not solely on the future inputs  $u(k+j)$ , but also on  
the future scheduling parameters  $\rho(k+j)$ . Because of the unavailability of the future scheduling values, this  
complicates the control design procedure, essentially because recursive feasibility and closed-loop stability  
of this moving horizon strategy requires an *MPC* which is robust against all possible future scheduling  
variations, as discussed Hanema et al. (2017a).

165 Consider that an *MPC* policy as in Problem 1 is applied to a process with a *LPV* model as in Eq. (8),  
with the vector of scheduling parameters evolving as:

$$\Gamma_k = \text{col}\{\rho(k+1), \rho(k+2), \dots, \rho(k+N_p-1)\}. \quad (11)$$

For this case, with initial condition  $x(k) = x_k$ , the optimizer has to internally elaborate the model constraint  
(2) based on Eq. (8), which exhibits nonlinearities from the second iteration onward:

$$\begin{cases} x(k+1|k) &= A(\rho(k))x_k + B(\rho(k))u(k|k), \\ y(k|k) &= A(\rho(k))x_k + D(\rho(k))u(k|k). \end{cases} \quad (12)$$

$$\begin{cases} x(k+2|k) &= A(\rho(k+1))A(\rho(k))x_k + A(\rho(k+1))B(\rho(k))u(k|k), \\ &\quad + B(\rho(k+1))u(k+1|k), \\ y(k+1|k) &= C(\rho(k+1))A(\rho(k))x_k + C(\rho(k+1))B(\rho(k))u(k|k) \\ &\quad + D(\rho(k+1))u(k+1|k), \end{cases} \quad (13)$$

and so forth, up to the  $N_p$ -th prediction. This results, therefore, in *NP* version of Problem 1, which has to  
170 be addressed by different algorithms than those for regular *LTI* plants, i.e. the *MPC* algorithm has to be  
designed specifically for *LPV* systems.

### 1.5. Motivation, Contributions and Organization

As evidenced, the *LPV* toolkit certainly appears as an interesting solution to *NMPC* design, especially in terms of using the proxy linear representation of the scheduling parameters to solve the control problem. This paper is motivated by the fact the nonlinear predictive control algorithms are sometimes too complex or result in rather conservative performances. Therefore, this work is devoted to review and discuss the available *MPC* algorithms that were specifically conceived for *LPV* systems, taking into account the advantages of this *LTI*-like framework. The available *LPV MPC* algorithms are mainly populated with robust or sub-optimal methods, and some recent developments include fast *NP* solutions to the problem, which are roughly equivalent to modern *NMPC* solvers such as ACADO in terms of computational performances. This paper contributes by analysing each of these studies and pointing out possible some open investigation threads to enrich *LPV MPC* literature.

This survey is organized in a progressive fashion: the more conservative and robust design methods are recalled first, ranging on to the less robust and even sub-optimal frameworks:

- The available robust *MPC* algorithms for *LPV* systems are reviewed in Section 2. These methods usually consider worst-case performances, taking into account an upper bound on the *MPC* cost function due to all possible future variations of the scheduling parameter vector.
- Section 3 surveys the works that make use of robust “trajectory tubes”, which are planned offline and guarantee convergence without having to compute the online scheduling parameter variations.
- The methods that take into account bounded scheduling parameter rates to simplify the computation of the worst-case performances are recalled in Section 4. Usually, these approaches yield less conservative results than those recalled in Section 2.
- The novel Nonlinear Programming *MPC* methods that exploit the linear proxy representation of *LPV* systems with scheduling variables taken as static functions of endogenous variables are discussed in Section 4.
- The available sub-optimal methods for the *LPV MPC* context are presented in Section 6. Generally speaking, these methods use an *LTI* guess for the behaviour of the system within the next  $N_p$  steps, considering a frozen value for the scheduling parameters. They have major drawbacks (since recursivity and feasibility is not guaranteed in many cases), while being much simpler to implement and run in real-time. Recent advances using set-membership design to overcome the sub-optimality drawbacks of these works are also reviewed in this Section.
- In Section 7, a numerical example of an *LPV* system controlled with some of the different surveyed *MPC* algorithms is presented, for illustration purposes.
- Based on the discussion raised throughout the paper, Section 8 narrows the available investigation gaps in this field of research.

This paper ends with general conclusions concerning *MPC* design for *LPV* systems, in Section 9.

**Remark 3.** The purpose of this review paper resides not only in that fact that recent *LPV MPC* methods pose a competitive framework to fast *NMPC* algorithms (such as ACADO and GRAMPC), but because great interest has been given to *LPV MPC* development over the years, which lead to a considerable amount of papers, with different approaches and formulations. Many of the surveyed techniques are not necessarily (numerical or performance-wise) “better” than some of these *NMPC* frameworks, but they represent altogether a different category that has grown in the literature and deserves a proper survey and analysis.

### 1.6. Definitions

Before formally presenting the surveyed papers, the formal definitions of Quadratic and Nonlinear Programming Problems are presented:

#### 220 **Definition 1.** Nonlinear Programming Problem

Consider an arbitrary real-valued nonlinear function  $f_c(x_c)$ . A Nonlinear Programming Problem (*NP*) determines the vector  $x_c$  that minimizes  $f_c(x_c)$  subject to  $g_i(x_c) \leq 0$ ,  $h_j(x_c) = 0$  and  $x_c \in \mathcal{X}_c$ , where  $g_i$  and  $h_j$  are also nonlinear.

#### 225 **Definition 2.** Quadratic Programming Problem

A Quadratic Programming Problem (or simply Quadratic Problem, *QP*) is a linearly constrained mathematical optimization problem of a quadratic function. A *QP* is a particular type of *NP*. The quadratic function may be defined with respect to several variables, all of which may be subject to linear constraints. Considering a  $c \in \mathbb{R}^{n_c}$  vector, a symmetric matrix  $Q_c \in \mathbb{R}^{n_c \times n_c}$ , a real matrix  $A_{ineq} \in \mathbb{R}^{m_c \times n_c}$ , a real matrix  $A_{eq} \in \mathbb{R}^{m_c \times n_c}$ , a vector  $b_{ineq} \in \mathbb{R}^{m_c}$  and another vector  $b_{eq} \in \mathbb{R}^{m_c}$ , the goal of a *QP* is to determine the vector  $x_c \in \mathbb{R}^{n_c}$  that minimizes a regular quadratic function of form  $\frac{1}{2}(x_c^T Q_c x_c + c^T x_c)$  subject to constraints  $A_{ineq} x_c \leq b_{ineq}$  and  $A_{eq} x_c = b_{eq}$ . The solution  $x_c$  to this kind of problem is found by many solvers seen in the literature, based on Interior Point algorithms, quadratic search, etc.

## 2. Robust (Worst-Case) Methods

235 For a start, this paper analyses the robust (worst-case) *MPC* methods for *LPV* systems. These approaches derived from the available robust and stochastic *MPC*s originally designed for dynamical systems or *LTI* ones under bounded disturbances; (Mayne, 2016) discusses whether literature is going in the right direction with these robust methods and what are the barriers to overcome.

Generally speaking, robust control policies are those able to steer the system to a specified target despite uncertainties. Robust *MPC* (*RMPC*) methods applied for the *LPV* case are usually named “min-max” algorithms, since they ensure stability for the minimal and maximal bounds of the system trajectories induced by the bounded scheduling parameters. Notice that if bounded parameter variations are considered, these bounds could also be taken into account for the design procedure; methods that take advantage of scheduling parameter variation rates are discussed in the sequel (Section 4) - for now, only works that solve the classical min-max problem are reviewed.

245 *RMPC* is rather consolidated; literature shows a range of works that vary according to how the optimization problem is set up, how the uncertainty set is described and how the feedback is incorporated to the optimization horizon (Vesely et al., 2010; Mayne et al., 2005). Again, it must be re-affirmed that the aim of this paper is not to review all *RMPC* methods considered for the *LTI* case that could be applied for the *LPV* one (considering the nonlinearities in the uncertain parameter space), but those that were conceived for *LPV* systems and evaluate the evolution of the scheduling parameters as an uncertainty constraints.

Therefore, for the case of *LPV RMPC* methods, the *LPV* system in Eq. (8) is elegantly rewritten as:

$$x(k+1) = A(\rho(k))x(k) + B(\rho(k))u(k) + \xi(k), \quad (14)$$

with  $\rho \in \mathcal{P}$  as the scheduling parameters and  $\xi(k) \in \mathcal{E}$  as an uncertainty variable that grows along the prediction horizon, whose uniform bounds  $\mathcal{E}$  depend on the value of  $\rho(k)$  and on the feasibility sets  $\mathcal{X}$ ,  $\mathcal{U}$  and  $\mathcal{P}$ .

Given that the knowledge of  $\rho(k)$  is available online, *LPV RMPC* methods are those that use it to compute the control matrices  $[A(\rho(k)) B(\rho(k))]$  and the bounds on the uncertainty  $\xi(k)$  at each sampling instant  $k$ . For instance, these uncertainties can be given with respect to some nominal/average  $\rho^*$ :

$$\xi(k+j) \in \mathcal{E} = [(A(\underline{\rho}) - A(\rho^*))\underline{x} + (B(\underline{\rho}) - B(\rho^*))\underline{u}, (A(\bar{\rho}) - A(\rho^*))\bar{x} + (B(\bar{\rho}) - B(\rho^*))\bar{u}],$$

where notation  $\underline{f}$  and  $\bar{f}$  represents the minimal and maximal bounds on an arbitrary variable  $f \in \mathcal{F}$ .

Essentially,  $\xi(k)$  represents the model-process mismatch, which increases along the horizon. Then, as discussed in (Gesser et al., 2018), *RMPC* are those that guarantee feasibility and regulation for whichever value of  $\xi(k+j)$ . This means that the whole uncertainty set  $\mathcal{E}$  must be taken into account at each optimization procedure, which is set to find the best possible sequence of control inputs  $U$  that minimizes  $J$  for all  $\xi(k+j) \in \mathcal{E}$ . By doing so, deteriorated/conservative performances are usually yielded, because the actual value for  $\xi(k+j)$  may stay very far from its edges (that were taken into account for the computation of  $U$ ) - this is the reason why *RMPC* is linked to a worst-case condition (a.k.a min/max).

Essentially, *RMPC* methods conceived for the *LPV* case are those that find, online with respect to the knowledge of  $\rho(k)$ , the bounds of the uncertainty set  $\mathcal{E}$  and thus perform the optimization procedure. Considering these methods, nowadays there are three possible application routes: *i*) those that find all possible uncertainty sets offline (using, for instance, *LMI* formulations); *ii*) those that use (polyhedral, ellipsoidal) target set techniques to parametrize  $\mathcal{E}$  and *iii*) those that use dynamic output-feedback (and input-output formulations) of the *LPV* model. These papers are discussed in the sequel, by category.

### 2.1. Algorithms with Offline Preparations

The first idea to solve the so-called robust min-max problem for nonlinear systems under *LPV* embedding is to solve some semidefinite problems under *LMIs* prior to the actual online *MPC* algorithm to find the possible uncertainty sets and their respective worst-case upper bound and control sequence  $U$ .

This kind of *LPV RMPC* algorithm was firstly seen in figure (Wada et al., 2004), which adapted the *LTI* algorithms originally based on quadratic stability constraints (Boyd et al., 1994; Kothare et al., 1996) to the *LPV* case by using parameter-dependent Lyapunov function to find the *LMI* solutions from (Daafouz & Bernussou, 2001). These *LMIs* have also been employed in (Cao & Lin, 2005), whose result was celebrated by Montagner et al. (2007) in terms of necessary and sufficient *LMI* conditions to compute quadratically stabilizing controllers. Anyhow, while the parameter-dependent Lyapunov provided relaxed stability condition, an increased number of *LMIs* and decision variables appeared. Moreover, these original min-max algorithms yielded quite conservative closed-loop results with rather poor performances.

In paralel, the works (Casavola et al., 2003, 2006) adapt the prior *LMI* conditions specifically for the polytopic class of *LPV* systems, which improves the formulation and reduces the number of *LMI* variables. These papers have been recently extended in (Ileš et al., 2015), which considers some flexibility to the formulation of the set-membership constraints.

Finally, instead of using parameter-dependent Lyapunov functions, papers (Jungers et al., 2009; Bumroongsri, 2014) also consider the polytopic *LPV* case and use multiaffine Lyapunov function (which are also called path-dependent), improving the analysis of stability and the accuracy of the upper bound to the quadratic performance cost  $J$ . These results have efficient online performances and can yield reasonable results, but their major drawback is that only state-feedback is considered and the offline preparations (through *LMIs*) can be quite tough to perform and set up. For industrial applications, this is usually not very well received (Qin & Badgwell, 2003).

### 2.2. Algorithms with Target Sets

Another way to solve the *RMPC* problem for *LPV* systems is to consider target sets. This kind of target set technique is based on predicting the future behaviours of the system states  $x(k+j)$  along the prediction horizon  $N_p$  and bounding them to a sequence of sets of similar form (polyhedral or ellipsoidal), which are computed with respect to the worst-case/min-max condition implied by the bounds on  $\xi(k+j)$ , i.e.  $\mathcal{E}$ . Bounding the future state behaviour guarantees regulation, but the sequence of sets must be computed beforehand for each possible value of the time-varying uncertainty.

The advantage of these methods is that they can easily guarantee robust stability - only some simple assumptions regarding the formulation of the sets are required, such as dissipativity or IQC remarks. Their major disadvantage is that, being robust and taking into account the mix-max performances, their online computation is quite heavy, since computing sets is not trivial and the numerical complexity increases with the order of the system and the size of the horizon (for the polyhedral case).

Individually, the recent works are detailed:



- Paper (Pluymers et al., 2005b) uses polyhedral invariant sets based on the possible values of  $\xi(k+j)$  and interpolates, online, a set of different *LTI MPCs* to solve the problem for the *LPV* case, considering the available knowledge of  $\rho(k)$ ;
- More recently, in (Bumroongsri & Kheawhom, 2012b) an adapted computation of these same invariant polyhedral sets is seen, where an offline preparation procedure is incorporated to the design, and thus reducing the online computational effort of this *LPV RMPC* algorithm;
- Based on the *LTI* results from (Brooms et al., 2001; Cannon et al., 2001) that have polynomial complexity, Angeli et al. (2008) use online ellipsoidal target sets, carrying out most of the computations offline and making use of the closed-loop predictions to improve the control performance and adapt the sets online. This method presents reduced computational burdens and not so much conservativeness as the other ones with polyhedral sets. The major downside is that the definition of the appropriate parameter values for the ellipsoids are not simple.

Note that other algorithms that use set-membership tools, but are not robust (that do not take into account the whole  $\mathcal{E}$  online), will be discussed in Section 6.

### 2.3. Dynamic Output-feedback Algorithms

There are three main reasons to investigate dynamic output-feedback *RMPC* algorithms: *i*) state-feedback *RMPC* for *LPV* systems is rather mature (see the two previous Sections); *ii*) handling state estimation can be a separate, complex problem, since *LPV* observers are needed; and *iii*) *RMPC* will only be well-received by the industry (and thus accepted for wide applications) if dynamic output-feedback tools are available.

To assume that the system states are available during the control implementation is somewhat unreasonable for many kinds of systems. Moreover, the need for *LPV* observers to estimate the states may harshly deteriorate closed-loop performances with respect to input disturbance rejection for active input constraints, as discussed by Wang & Young (2006).

Considering dynamic output-feedback *RMPC* policies for *LPV* systems, two sets of works are available, as detailed in the sequel: *i*) those that use input/output formulations of the *LPV* plant and *ii*) those that use state-space models and add a state estimation error constraint to the optimization problem.

#### 2.3.1. Input/Output Formulation

Note that there are quite a few practical techniques to yield I/O *LPV* models via identification procedures from both open-loop and closed-loop data sets (Abbas & Werner, 2009; Tóth et al., 2011a). This is supported with several powerful identification approaches and successful applications, refer to the review by Bachnas et al. (2014).

The results considering *MPC* for *LPV* systems under I/O formulation are very few. Since state variables do not appear in I/O formulations, the *MPC* algorithms must ensure bounded input to bounded output (BIBO) stabilization (instead of input-to-state). A first, preliminary results was presented in (Abbas et al., 2015), which lacked proofs for stability for any  $\xi$ . Then, later on, the method was “robustified” to solve the optimization procedure considering the whole  $\mathcal{E}$  set in (Abbas et al., 2016, 2018). This recent paper offers a very practical solution for *LPV MPC* design and should be well incorporated for practical industrial applications. The core design is based on the use of full block multipliers and *LMI* conditions borrowed from Scherer (2001). The scheduling parameters are assumed to be constant along the prediction horizon  $N_p$ , which leads to fixed values for the impulse response parameters  $a_i$  and  $b_i$  and, then, the uncertainty set is included into the resulting *QP*. In (Liu et al., 2016), a distributed *MPC* formulation for spatially-invariant interconnected I/O *LPV* systems.

The majority of *LPV* controller synthesis techniques and real-life applications have been originally conceived for state-space frameworks, as argues (Hoffmann et al., 2014). Therefore, there are no major drawbacks in these papers apart from defining the system in I/O formulation of Eq. (9), which is, at least for now, rather unusual. Their *RMPC* strategy is designed to counteract the worst-case possible

uncertainties of the scheduling variable, i.e. solving the problem at each instant based on the bounded of  $\mathcal{E}$  as a hard constraint.

**Remark 4.** The stability analysis for I/O LPV models can be based on implicit image representation and through Finsler’s Lemma, as done in (Wollnack & Werner, 2016; Wollnack et al., 2017)

### 2.3.2. State-Space + Estimation Error Constraint

LPV MPC algorithms concerning the I/O formulation are very novel and a promising investigation field. Anyhow, the majority of available output-feedback RMPC works consider state-space formulation and the inclusion of an estimation error constraint to the QP when observers are used.

Notice that, while the output is fed back to the main controller, it must choose an optimal control policy  $u(k) = \kappa(y(k), \rho(k))$  such that the (internally) estimated states do not deviate from a bounded limit. The optimization procedure, since they are RMPC methods, continues to be a min-max procedure, but now with the worst-case bounds on  $y$  instead of  $x$ .

Take  $\check{x}(k)$  as the difference between the actual state  $x(k)$  and the estimated state  $\hat{x}(k)$ , i.e.  $\check{x}(k) = x(k) - \hat{x}(k) \in \check{\mathcal{X}}$ . Problem 1 is adapted to include  $\check{x}(k) \in \check{\mathcal{X}}$  as a constraint such that the estimation error also becomes bounded.

This kind of algorithm with dynamic output-feedback and bounds on  $\check{x}(k)$  are included into the algorithm was firstly presented in (Lee & Park, 2007). In this paper, the MPC cost function is taken as a parameter-dependent Lyapunov cost that decreases along the horizon to induce I/O stability. Notice that the downside of this original work is that the resulting algorithm was solved online using an LMI at each sampling instant, which is numerically costly.

Considering LTI processes, (Baocang et al., 2009) splits this kind of RMPC problem into two separate procedures, with one step dedicated solely for handling the estimation error constraint via a feasible optimization problem, being solved after the main optimization is performed. Then, the recursive feasibility of the main optimization is found in a much easier fashion, without having to include the estimation error constraint. This result was extended to the LPV case in (Ding et al., 2013) and for the polytopic case with parametric uncertainty and bounded disturbances in (Ding & Pan, 2016). Both these latter papers present formal stability and recursive feasibility proofs based on the terminal ingredient axioms by Mayne et al. (2000).

Other interesting works with similar results are:

- (Park et al., 2011; De Caigny et al., 2012), that yield quasi-min-max algorithms by simplifying some of the constraints, using a robust state observers and repeatedly solving a convex optimization problem based on the LMI conditions from previous literature to minimize ta worst-case upper bound of the cost function;
- Yu et al. (2012) discuss the variation of bounds on the estimation error for the case of possibly asymmetric constraints and symmetric box-type constraints.

**Remark 5.** Note that the issue of explicit MPC for LPV systems is also to be recalled. Such methods were were firstly introduced by Besselmann et al. (2008) and later on investigated in terms of stability and optimality in (Besselmann et al., 2012). Some perspectives on these algorithms are discussed in (Pistikopoulos, 2009). Their major downside is that, since the future values of the scheduling parameters are unknown, the explicit algorithm ensures the constraints are satisfied for all possible system trajectories, which leads to (numerical-wise) high-demanding QPs. Some times, non-QP procedures are found, as multiparametric programming optimization problems.

## 3. Tube-Based Methods

There also exist a different kind of RMPC framework that deserves special attention, due to its applicability, advantages and number of works: these are the tube-based RMPC algorithms.

As thoroughly discussed in the prequel, predictive control under uncertainty leads to a feedback min-max optimization problem, that can be tackled through dynamic programming. Due to the inherent complexity and conservativeness of these robust min-max algorithms, literature shows a series of papers that try to provide computationally tractable approximations of this problem for practical applications (Raković, 2015).

Tube-based *MPC* (*TMPC*) design is a paradigm originally conceived to reduce the numerical complexity of the *QP* with respect to the min-max *RMPC* framework. Thereby, its main advantage is that the numerical toughness grows well-behaved (linear behaviour, in many cases) with the size of the prediction horizon  $N_p$ . This framework was originally conceived for the *LTI* case and soon generalized for *LPV* systems. *LPV TMPC* papers are reviewed in the sequel.

The *TMPC* technique firstly grew to address the control problem of constrained linear systems subject to additive norm-bounded disturbances, as seen in (Langson et al., 2004; Rakovic et al., 2012). Note that there are different types of possible tube parametrizations (rigid, homothetic, elastic) seen in the literature.

In an *LTI* setting, when applying *MPC* for systems subject to bounded uncertainties, the controller should include an accurate prediction of the system variables (states or states plus dynamic outputs) along the prediction horizon. The mix-max procedures recalled in Section 2 assumed worst-case predictions for these variables, bounding them with respect to the uncertainty bounds. The *TMPC* algorithm does not compute the worst-case predictions, but predicts a nominal system trajectory without uncertainties and guarantees that the deviation that the real system has from this nominal trajectory lies inside a “tube” (which is exploited by using the bounds of the uncertainties). The trade-offs and advantages of *TMPC* with respect to min-max *RMPC* techniques are reviewed and quantified in Gesser et al. (2018).

Soon after the original *LTI TMPC* papers, *TMPC* was extended to nonlinear cases, as in (Mayne et al., 2011; Santos et al., 2018), and to time-varying processes, as done in (Gonzalez et al., 2011). These developments opened doors to a *LPV TMPC* formulation.

The notion of *TMPC* applied to the *LPV* is slightly different than the prior. In the aforementioned papers (for the *LTI*, *LTV* and nonlinear cases), persistent additive disturbances were assumed to be present and, thus, the stabilization of the process to the origin could not be fulfilled, but some convergence to a boundary set centered around the origin. For the *LPV* case, this condition can be relaxed because it should be possible to stabilize the process to the origin since the uncertainty arises due to the scheduling parameters, that enter in the model multiplicatively, as discusses Hanema et al. (2017a).

*LPV TMPC* algorithms have an essentially similar idea of the classical *TMPC* papers, of building a tube for the real trajectories (which, in the *LPV* case depend on future unknown scheduling variables), and computing the predictions with a nominal trajectories (assuming the scheduling variable would remain constant along the horizon). This is further explained: the admissible values of the scheduling variables  $\rho(k+j|k)$  should be known for a “perfect” *MPC* algorithm; anyhow, a very simple mechanism can be used to bound these values: taking a sequence of subsets of the scheduling set  $\mathcal{P}$  along the horizon, namely  $\{\mathcal{P}\}_k = \{\{\mathcal{P}\}_{0|k}, \dots, \{\mathcal{P}\}_{N_p-1|k}\}$ ; this sequence of sets is understood as forming a tube around a “nominal” future trajectory taken with  $\rho(k+j) = \rho(k) - \{\mathcal{P}\}_k$  is referred to, in the literature, as the “scheduling tube” at time instant  $k$ . For the *LPV TMPC* procedure, at each instant  $k$  this scheduling tube is constructed based on the actual bounds  $\bar{\rho}$  and  $\underline{\rho}$ .

Works (Cannon et al., 2011; Fleming et al., 2014; Muñoz-Carpintero et al., 2015; Hariprasad & Bhartiya, 2014) proposed *TMPC* design specifically for multiplicative uncertain processes. In these papers, the scheduling parameters are not properly defined (neither measured online), which means that the context study is more robust rather than *LPV*. Anyhow, the application of **stabilizing** *TMPC* for *LPV* systems have appeared consequently:

- A first paper on *LPV TMPC* was proposed by Su et al. (2012), where an output feedback algorithm is proposed, whose results is slightly superior to that of a min-max technique, thereby named quasi-min-max;
- A first framework for the development of stabilizing *TMPC* was presented in (Brunner et al., 2013), where polytopic systems are considered under assigned initial condition sets. This work ensures that all process trajectories that depart from the initial sets are contained inside a tube which ends in a  $\lambda$ -contractive terminal set.

- Based on the theoretical developments of Brunner et al. (2013), the recent paper (Hanema et al., 2016) designs a novel *LPV TMPC*, wherein the homothetic tubes guarantee the convergence of the system trajectories to a  $\lambda$ -contractive terminal set; the online *QP* for the feedback policy is conducted in the *LPV* vertices (the system is polytopic);
- Then, (Hanema et al., 2017a) elucidated how to properly build finite-step terminal set and stage cost ingredients for *LPV TMPC* design. In this paper, it is shown that, under certain assumptions on the parametrization of the tube, recursive feasibility can be obtained.
- Very recently, papers (Abbas et al., 2019; Hanema et al., 2020) deploy *TMPCs* for *LPV* systems with bounded rates of parameter variations. The first paper considers the maximal and minimal evolution sequence for the scheduling variables for the next  $N_p$  steps, considering the rates of variations, and thus computes the tube that encompasses all these possible trajectories. The latter reference uses heterogeneously parameterized tubes.

It must be remarked, from the above literature survey, that tube-based *MPC* design can be quite promising for the *LPV* case. The use of tubes abruptly reduces numerical complexity of the *QP*, but, if these tubes are equivocally designed, the control performances may turn out excessively conservative. It was discussed how *TMPC* for the *LPV* can indeed be stabilizing, and bring the process to the origin. Anyhow, rather few works have shown practical applications and further theoretical advances are lacking. The topic is definitely rather unexplored (or under-investigated).

### 3.1. Discussion

Tube-based *MPC* design can be formally understood as a subset of *RMPC* tools, although in this paper these algorithms were analysed as distinct/different categories. Anyhow, some words must still be spent on how to choose whether to apply *TMPC* or *RMPC* methods:

- *RMPC* methods with offline preparations (*LMIs*) are the simplest to implement; deep theoretical knowledge is not necessary to develop the online algorithm or to tune the controller, even for the case of systems with many states (i.e. large  $n_x$ ). The offline *LMI* procedures are also quite standard. In terms of scalability, it must be remarked that the average computational time for the offline procedures increases considerably with respect to the number of states, inputs and prediction horizon, as discusses Gesser et al. (2018).
- Online min-max *RMPC* (as with target sets) as well as *TMPC* algorithms, are more complex to code: the computation of the robust invariant sets and the scheduling tubes is not trivial; the tuning of the *MPC* parameters may also not be explicit. Their computational cost is simpler than computing *LMIs*, although totally done online, given that state inequalities such as  $A_{in}x(k+j) \leq b_{in}$  are plugged into the resulting *QP*.
- The decision of which kind of robust *MPC* method is better is related to the system model (number of states, inputs) and the size of the prediction horizon. *LMI*-based *RMCP* methods are the best solution for smaller systems, due to implementation simplicity. Online min-max *RMPCs* and *TMPCs* are interesting options for faster systems because of the low computation time for the *QPs*, despite being harder to tune and implement. Computing the robust control invariant set might be a challenge for the target-based methods, while *TMPCs* can achieve better performances if an accurate description for the future scheduling tube is available (such as when considering bounds on the variations of the scheduling variables).

## 4. Bounded Parameter Variation Rates

This Section surveys the papers that take into account bounds on the variations of the scheduling parameters to simplify the predictive controller synthesis.

Note that, for the robust *MPC* case discussed in the previous Sections, the surveyed algorithms had the goal to minimize the *MPC* worst-case upper bound of the cost function (or the trajectory inside a scheduling tube), defined by the boundaries of  $\mathcal{E}$  over the prediction horizon, assuming that the scheduling parameters could vary arbitrarily inside the description domain  $\mathcal{P}$ .

Nonetheless, for the great majority of real world applications, the dynamics of the scheduling parameters vary within a given (known) rate limit; this issue is debated in (Amato & Mattei, 2001; Amato et al., 2005). Mathematically speaking, this means that the time-varying scheduling parameters  $\rho$  also present bounds on their rates of variations:

$$\partial\rho(k) = [\rho(k) - \rho(k-1)] \in [\underline{\partial\rho}, \overline{\partial\rho}] = \dot{\mathcal{P}}. \quad (15)$$

It is reasonable to understand that if Eq. (15) is taken into account for the control design procedure, the bounds on the future scheduling trajectory are very easy to be found: taking a receding prediction horizon of  $N_p$  discrete-time steps, departing from  $k_0$ , the two (upper and lower) trajectories for the evolutions of the scheduling parameters are:

$$\Gamma_{k_0}^{min} = \text{col}\{\rho(k_0) + \underline{\partial\rho}, \rho(k_0) + 2\underline{\partial\rho} \dots \rho(k_0) + (N_p - 1)\underline{\partial\rho}\}, \quad (16)$$

$$\Gamma_{k_0}^{max} = \text{col}\{\rho(k_0) + \overline{\partial\rho}, \rho(k_0) + 2\overline{\partial\rho} \dots \rho(k_0) + (N_p - 1)\overline{\partial\rho}\}. \quad (17)$$

This simple consideration greatly simplifies the *MPC* algorithms (specially for the robust offline procedures), as verified in the sequel:

- In the case of *RMPC*, the paper by Jungers et al. (2011) must be highlighted, given that it uses *LMIs* for offline preparations that take into account these bounds on  $\partial\rho(k)$ .
- Casavola et al. (2002); Park & Jeong (2004) develop state-feedback min-max *RMPC* algorithm for the case of *LPV* models with bounded rates of parameter variations. Their online computational stress is smaller than papers (Pluymers et al., 2005b; Bumroongsri & Kheawhom, 2012b), for instance, because the upper bound of the *MPC* cost function  $J$  becomes easier to compute than when the parameters are assumed to vary arbitrarily;
- Considering *RMPC* methods with target sets, Suzukia & Sugie (2006) use ellipsoidal target sets also taking these bounds on  $\partial\rho$  to simplify the online computation of the sets. This work, unfortunately, involves heavy numerical procedures, even for smaller systems. Papers (Casavola et al., 2012; Bumroongsri & Kheawhom, 2012a) present a major advance in simplifying the computation of the ellipsoidal sets online and achieving quite fast *MPC* algorithms when a polytopic structure of *LPV* model is assumed. Yet, these works need some complementary offline procedures which are not necessarily simple to perform;
- An extension of the explicit *LPV MPC* algorithm has been provided using bounded  $\partial\rho$  constraints in (Besselmann et al., 2009a,b). These methods are based on dynamic programming, Pólya's Theorem and Delaunay's triangulation of a polytope. An alternative path for explicit *MPC* design is proposed in (Li & Xi, 2010), where an iterative method is used to refine the polytope describing the possible output, state dynamics as functions of the discrete-time instants  $k$  and the scheduling parameter variation rate  $\partial\rho$ .

There is also another group of works that take into account the variation rate of the scheduling parameters for the design of predictive controllers - the papers that adapt search tree *MPC* algorithms for the context of *LPV* systems. These methods are often referred to as "scenario"-based *MPC* algorithms. Some of these robust *MPC* are reviewed in (Saltik et al., 2018).

The essential idea behind these methods is to compute all possible scheduling trajectories from the discrete instant  $k_0$ . Using search algorithms to find all possible system trajectories, the *MPC* design is based on minimizing the upper bound of objective function for all the possible trajectories found via the search procedure. Using an orthogonal search tree, this control framework is fully detailed for a nonlinear case in (Johansen & Grancharova, 2003).

540 Specifically designed for the *LPV* case, (Calafiore & Fagiano, 2013) shows the application of this scenario optimization algorithm under stochastic scheduling conditions.

Note that this control framework using scenario optimization is essentially robust (*RMPC*), as argued by Calafiore & Fagiano (2012). The overview and perspectives for future research on this topic are extensively discussed by Mesbah (2016). The downside of these methods is that the search algorithm adds a difficulty, 545 tree-based search is rather computationally costly. Therefore, the application of these algorithms is still not optimized for real-time applications.

## 5. Nonlinear Programming Methods

Very recently, a new branch of *LPV MPC* algorithms has appeared in the literature as an interesting 550 alternative to modern solver-based *NMPC* tools, such as ACADO, GRAMPC and CasADi. These algorithms are rapidly improving and their core is to embed nonlinear systems to an *LPV* representation with an endogenous  $\rho(k) = f_\rho(x(k), u(k))$  proxy, and then manipulate this proxy to make the predictions along the horizon.

Differently than dealing with the scheduling parameters via robust solutions, this *NP* algorithms were 555 firstly seen in (Cisneros et al., 2016), where an efficient formulation was presented. In this first paper, the scheduling parameters were iteratively guessed along the horizon, through a mechanism that iteratively refined the predictions for the scheduling parameters through the nonlinear model, coming close to a sequential *QP*-based solution of *NMPC*. In fact, the solution for the *NP* comes close to that of a Second-order Cone Program.

Such method was extended and further formalized in (Cisneros & Werner, 2017a), where the reference tracking problem was embedded to the *NP* formulation. In (Cisneros & Werner, 2017b), parameter-dependent stability conditions for these algorithms were presented. Finally, an alternative stabilizing tube-based formulation of such algorithm was seen in (Hanema et al., 2017b).

It must be remarked that this kind of algorithm represents a numerically-efficient approach to *LPV MPC* 565 through *NPs*, for which the nonlinear constraints are handled in an *LPV* fashion. The numerical results, in terms of average computational stress, is compared to the fast real-time iterations *NMPC* algorithms. It has been recently shown (Cisneros & Werner, 2017a, 2019) that such *NP LPV MPC* technique is able to outperform or achieve similar performances to both ACADO and CasADi solver-based *NMPC*s. In (Cisneros et al., 2019), for the case of a pendubot stabilization, the average computational time needed to solve the 570 optimal nonlinear tracking control problem is of 0.265 ms with such *LPV MPC*, while 0.290 ms and 11.82 ms with the other two tools, respectively. These results are quite impressive.

Since these formulations are quite new, further guarantees in terms of recursive feasibility and stability are still lacking. Some insights regarding a dissipativity formulation have been given in (Cisneros & Werner, 2018), but the results are not yet conclusive.

575 **Remark 6.** Some papers that also provide fast solutions to the *LPV MPC* problem without resorting to plain sub-optimality or robust tools are: a) those that use velocity-based linearization to yield the *LPV* model, instead of *LDI*, as in (Cisneros et al., 2018); and b) Data-driven algorithms based on velocity linearization and truncated Koopman operators (Korda & Mezić, 2018; Arbabi et al., 2018) - such operators 580 can recursively updated linear model to make the predictions, solving just one *QP* per sampling instant  $k$ .

## 6. Sub-Optimal Methods

Differently from the previously described methods, sub-optimal *LPV MPC* frameworks disregard the possible values for the real evolution of the scheduling parameters inside the prediction, namely  $\Gamma_k$ , and opt for a prediction guess  $\hat{\Gamma}_k$  to simplify the complex optimization problem, using simplified (*LTI*) models for the prediction of the future controlled outputs inside the horizon. This Section focuses in analysing these 585 works that use such simplification approach.

Sub-optimal *LPV MPC* methods can be those that schedule the nonlinear system into multiple local *LTI* models, as done in (Lazar et al., 2006; Bravo & Normey-Rico, 2009), and develop the predictive controller based on the *LPV* scheduling of these models (gain scheduling). This kind of technique was applied for practical purposes (such as the control of solar and desalination plants) in (Torrico et al., 2010; Ayala et al., 2011). A recent survey of near-optimal control of nonlinear processes was delivered by Zhang et al. (2019).

An one-step receding-horizon *MPC*-like strategy is proposed for discrete-time *LPV* systems in (Nguyen et al., 2013); for this case, as  $N_p = 1$ ,  $\Gamma_k$  is known and, thus, a parameter-dependent Lyapunov function can be used to find, at each instant, the optimal solution by solving a linear *QP*. Anyhow, when the prediction horizon increases, this no longer becomes feasible, since  $\Gamma_k$  is no longer known.

To address this, a scheduling parameter trajectory guess can be used, which translates Problem 1 into a *QP*. This has been done in (Morato et al., 2018), for the case of vehicle suspension systems, taken a frozen guess  $\hat{\rho}(k+j) = \rho(k)$  for  $j = 1, \dots, N_p$ . Considering the fault-tolerant control of renewable microgrids, the same kind of frozen guess *LPV MPC* algorithm has been recently applied in (Morato et al., 2020). The same methodology is has also been recently used by Alcalá et al. (2019).

In (Calderón et al., 2019), a benchmark comparison is given for frozen scheduling guess approaches considering both state-space and input/output formulations. Paper (Morato et al., 2019a) uses a fixed *MPC*, developed upon a fixed nominal *LTI* model, coupled to a feedback filter that adapts the control policy according to the evolution of the scheduling parameters. However, the issue that resides with all these previous works is that the results may be (very) sub-optimal (or even unstable) since the system trajectory might not be inside the region of attraction of the *MPC*, resulting in infeasibility. This means that the recursive feasibility property of the main optimization problem is not guaranteed, and the receding horizon implementation is only viable if the previous control solution is maintained whenever an infeasible point occurs.

Recursive feasibility, as thoroughly discussed by Löfberg (2012), is a very important property of predictive control loops: the optimization problem must be feasible for all future time instants  $k > k_0$  if it is feasible for the initial starting instant  $k_0$ . Albeit sub-optimal approaches having possible worse performances, these more relevant papers are those that guarantee recursive feasibility despite model-process mismatches due to the model simplifications by considering a fixed evolution guess for the scheduling variables.

### 6.1. Set-Membership Methods

The Set-membership *MPC* methods were firstly conceived based on the set-theoretical definitions developed by Blanchini (1999) (originally defined for *LTI* models but then extended to nonlinear and *LPV* plants) of Positively Invariant Control Sets and  $N_r$ -Steps Invariant Set Sequences.

With these set-theoretical foundations, the original paper by Mayne et al. (2000) uses terminal sets (and costs), as well as constrained system performance, and demonstrated how these tools may be used to guarantee recursive feasibility and stabilization to *MPC* algorithms. Therefore, these tools have become extremely important for the control of uncertain systems (or nonlinear ones with linear models, *etc*), since they allow controlled system to abide as close to the performance objectives (such as reference tracking, disturbance rejection, *etc*) as possible, in the sense that these are obeyed whilst stability and feasibility are maintained, as discussed in the Introduction. They have been, therefore, combined to enlarge the domain of attraction of *MPC* algorithms, finding more options of stable closed-loop equilibrium points (Limon et al., 2005).

In (Limón et al., 2008), these tools are generalized for the case of reference tracking of *LTI* plants. The framework proposed therein was then extended in (Ferramosca et al., 2008; Raimondo et al., 2007; Ferramosca et al., 2009) and guarantees performance and feasibility of the *MPC* policies when applied to nonlinear systems. These tools are used in the *LPV MPC* context in many of the works surveyed in this paper, such as (Brunner et al., 2013; Hanema et al., 2017a; Morato et al., 2019b; Mate et al., 2019). The major innovation of these methods is to use contractive sets to make sure the *LPV* does not “steer” towards an unwanted region.

To guarantee that within  $N_r$  steps from the initial instant  $k_0$  the controlled system reaches a control invariant set  $\Upsilon$  which contains the target equilibrium, the following contractive terminal set constraint is

included to the *MPC* design:

$$x(k_0 + N_r) \in \Upsilon_j, j = \max\{N_r - k, 0\}, \quad (18)$$

assuming  $\mathcal{S}_{N_r}$  is available from a computed sequence of control invariant sets. Note that this terminal set  $\Upsilon_j$  is equal to the larger  $\Upsilon_0$  at the initial instant  $k_0$  being shrunk subsequently until, at  $k_0 + N_r$ , it becomes the smallest set  $\Upsilon_{N_r}$ . Moreover, note that  $N$  is a sliding-horizon, but  $N_r$  is not, it is fixed and, gets closer to  $k$  as  $k$  increases. This constraint makes the *MPC* method intrinsically time-varying, since, at least for the first  $N_r$  samples, the sets are contracting. Usually,  $N_r \geq N_p$ .

In (Morato et al., 2019b), an online, iterative Least-Squares identified scheduling parameter model is used to compute  $\hat{\Gamma}_k$ . In this paper, a Lyapunov-decreasing terminal cost and a contractive terminal set is used to make sure that the *LPV* system converges, despite the model-process mismatch and the issue of possible infeasibility. In (Mate et al., 2019), proofs are provided on how to develop the proper terminal ingredients and Lyapunov-decreasing stage costs to ensure stabilization of *LPV* systems. Thereby, it becomes clear that the recent advances to overcome the sub-optimality drawbacks of the previous works are ensured by the use of set-membership design.

**Remark 7.** In terms of the computation of the set sequences, (Pluymers et al., 2005a) proposes an efficient method for polyhedral invariant sets and polytopic systems and (Bravo et al., 2005) offers their computation for constrained nonlinear systems based on an interval arithmetic approach.

## 6.2. Multiple Model MPC

There is also another (final) category of works that must be commented on. These papers may take into account the variation rate of the scheduling parameters for the design of predictive controllers (but also can neglect this factor). These works are those that adapt Multiple Model *MPC* for the context of *LPV* systems. This is a rather unexplored field, with promising results, but rather few works.

Multiple Model *MPC*, as originally seen in (Aufderheide & Bequette, 2003), where the *DMC* algorithm is extended for multiple models of a same nonlinear plant, resides in including in the *QP* optimization procedure a model choosing variable  $\mu_j$  for the prediction of the controlled variables within the next  $N_p$  steps.

To synthetically describe these methods, take, for instance, a generic polytopic<sup>5</sup> *LPV* system in Eq. (14), that lays within the polytope  $\Omega$ . Even though the exact model of the process is unknown for the next steps, due to the unavailability of  $\Gamma_k$  for *MPC* design, this system can be described, for all future instants  $k > k'$ , by a generic pair  $[A(\rho(k')), B(\rho(k'))]$  which also belongs to the polytope  $\Omega$ .

Then, for the polytopic case, any pair  $[A(\rho(k)), B(\rho(k))]$  can be represented as a convex combination of the polytope vertices such that:

$$A(\rho(k)) = \sum_{j=1}^L \mu_j A_j \quad \text{and} \quad B(\rho(k)) = \sum_{j=1}^L \mu_j B_j, \quad (19)$$

$$\text{with} \quad \sum_{j=1}^L \mu_j = 1 \quad \text{and} \quad 0 \leq \mu_j \leq 1. \quad (20)$$

For this representation,  $\mu_j$  is a weighting variable that determines how much does each vertex (*LTI* model) of the polytope represents the uncertain *LPV* model for the future  $N_p$  steps. Then, as done recently in (Pipino & Adam, 2019), a complementary *QP* can be included to the *MPC* design to match  $\mu_j$  to the

---

<sup>5</sup>This is just an example. Generalized formulations have been discussed for other *LPV* classes.



last  $N_b$  data instants (backward horizon, for which  $x(i)$  is known, i.e. measured data). This is:

$$\min_{\text{col}\{\mu_j\}} (x(i) - \hat{x}(i))^2 \quad (21)$$

$$\text{s.t. } \hat{x}(i) = A(\rho(i))x(i) + B(\rho(i))u(i) \quad i \in \mathbb{Z}_{k-N_b:N},$$

$$A(\rho(i)) = \sum_{j=1}^L \mu_j(i)A_j \quad , \quad (22)$$

$$B(\rho(i)) = \sum_{j=1}^L \mu_j(i)B_j \quad , \quad (23)$$

$$\sum_{j=1}^L \mu_j(i) = 1 \quad , \quad (24)$$

$$0 \leq \mu_j(i) \leq 1 \quad j \in \mathbb{Z}_{1:L}, \quad (25)$$

$$\hat{x}(i) \in \mathcal{X}, \quad \forall i \in \mathbb{Z}_{k-N_b:k}. \quad (26)$$

The above *QP* results in a collection of  $\mu_j$ , which are plugged into the prediction model for the original *MPC* optimization, which becomes a *QP*.

Very similar *LPV MPC* algorithms like this were firstly seen in (Arkun et al., 1998). More recently, they were re-discussed by Jin et al. (2011). This framework is also available for hybrid (Nandola & Bhartiya, 2008) and switched (Hariprasad et al., 2012) nonlinear systems and could be adapted for the (hybrid, switched) *LPV* case. Evidently, this topic is rather under-explored. Debates on whether its effectiveness are better or worst than the regular the previous sub-optimal tools are lacking (in terms of numerical burden, achieved performances, etc).

680

## 7. An Application Example

In this Section, a simple numerical example is presented to illustrate how of the surveyed *LPV MPC* tools can be applied. The aim is to synthetically present to the reader how to design and implement such algorithms and to discuss their performances and preparation steps.

For this purpose, consider the following application example of two cascaded tanks, for which the level of water must be regulated. The nonlinear model of this process is cast into an SS *LPV* formulation through LDI; futher details are given in Appendix A. The controller must maintain the level of the tanks  $x(k)$  as close to a steady-state reference target  $x_r = [0.45 \quad 0.45]^T$  m as possible. The tanks have base areas of  $A_b = 1 \text{ m}^2$  and the system operates with a sampling period of  $T_s = 20 \text{ ms}$ . State-feedback is available: both (levels) states are measurable. Remark that the system has a single control input, which is the input flow of water at the top of the first tank. A step-like input disturbance signal of  $0.5 \text{ m}^3/\text{s}$  is considered upon  $x_1$ , from  $t = 0.5$  to  $0.6 \text{ s}$ .

690

Regarding this problem, some of the main methods from the previous survey are implemented. These are:

695

- The online min/max *RMPC* algorithm from (Cao & Lin, 2005);
- The online min/max *RMPC* algorithm considering bounded rates of parameter variation from (Li & Xi, 2010);
- The tube-based robust algorithm (*TMPC*) from (Su et al., 2012);
- The sub-optimal procedure from (Morato et al., 2019b), denote SOMPC;
- The *NP* design method from (Cisneros & Werner, 2017a), denoted NLMPC;

700

All these control strategies consider the same prediction horizon of  $N_p = 15$  samples and a cost function as follows:

$$J = \|x(k + N_p|k) - x_r\|_P^2 + \left( \sum_{i=0}^{N_p} \|x(k + i|k) - x_r\|_Q^2 + \sum_{i=0}^{N_p-1} \|u(k + i)\|_R^2 \right), \quad (27)$$

for which the matricial weights are taken according to (Limón et al., 2008):  $Q = \mathbb{I}_2$ ,  $R = 1$  and

$$P = \begin{bmatrix} 16.98 & 5.52 \\ 5.52 & 11.51 \end{bmatrix}.$$

The algorithms are implemented with respect to the guidelines presented in (Löfberg, 2003), in order to achieve faster optimization. The implementation is performed in a 2.4 GHz Macintosh computer with Yalmip and Matlab softwares. The solver used for the robust techniques was MPT, while for sequential QPs and single QPs, QuadProg. Note that recursive feasibility is guaranteed for all methods, due to the forms of matrices  $P$ ,  $Q$  and  $R$  and the use of terminal ingredients, see (Löfberg, 2012). The achieved results, which comprise 2s of simulation, are celebrated below:

### 7.1. Offline Preparations

Firstly, it must be recalled that the *TMPC* design requires some offline preparations to compute the robust tube within which the state predictions must lie. These preparations, for the considered system, imply in the solution three *LMIs*, of somewhat large dimensions ( $16 \times 16$ ,  $16 \times 16$  and  $4 \times 4$ ). The solution of such problem is found within 1.84s using Yalmip and SDPT3 solver. The solution of the *LMIs* gives a robust set parametrized over the sampling instants  $k$ . The online algorithm embeds a simple additional inequality constraint of type  $A_{tube}(k + j)x(k + j|k) \leq b_{tube}(k + j)$ .

### 7.2. Online Computational Stress

Regarding the online part of these methods, the average time elapsed to solve their optimization procedures is recalled in Table 1. Note that the min/max *RMPC* algorithm considering bounded rates of parameter variations from (Li & Xi, 2010) is slightly faster than the one from (Cao & Lin, 2005), since the uncertainty set  $\mathcal{E}$  becomes smaller, computed on the basis of  $\rho(k + j) = \rho(k) \pm \partial\rho$ , with  $\partial\rho \in \hat{\mathcal{P}}$ .

As expected, the sub-optimal and the tube-based optimization procedures are the fastest, since they solve simple regular *QPs*. The *NP LPV MPC* is solved through sequential *QPs*, with competitive results.

Table 1: Online Computational Stress

Method	Average Elapsed Time (ms)
Min/max <i>RMPC</i> : (Cao & Lin, 2005)	16.754
Min/max <i>RMPC</i> : (Li & Xi, 2010)	7.973
Robust <i>TMPC</i> : (Su et al., 2012)	1.904
Sub-optimal <i>LPV MPC</i> : (Morato et al., 2019b)	1.804
<i>NP LPV MPC</i> : (Cisneros & Werner, 2017a)	2.908

### 7.3. Tracking Performances

Figure 1 shows the achieved results in terms of reference tracking. The results with the robust approaches ((Cao & Lin, 2005), (Li & Xi, 2010) and (Su et al., 2012)) are quite similar, ensuring stabilization and tracking. The sub-optimal method from (Morato et al., 2019b) is able to achieve more aggressive due to the least-squares estimation used for the predictions of the scheduling parameters; the method lets go of robustness but does not ensure, *a priori*, stabilization. The *NP LPV MPC* (Cisneros & Werner, 2017a)

730 shows very interesting performances, using nonlinear (real) predictions and guaranteing optimality (and thus stabilization) and aggressiveness.

In terms of reference tracking performances, a root-mean-square (*RMS*) index of the tracking error is used to compile the achieved results, as exhibited in Table 2. The robust approaches are clearly more conservative and take more time to stabilize the plant and, thus, the tracking error takes more time to converge to zero. The best results are obtained with the *NP LPV MPC*, while the sub-optimal approach yields similar performances.

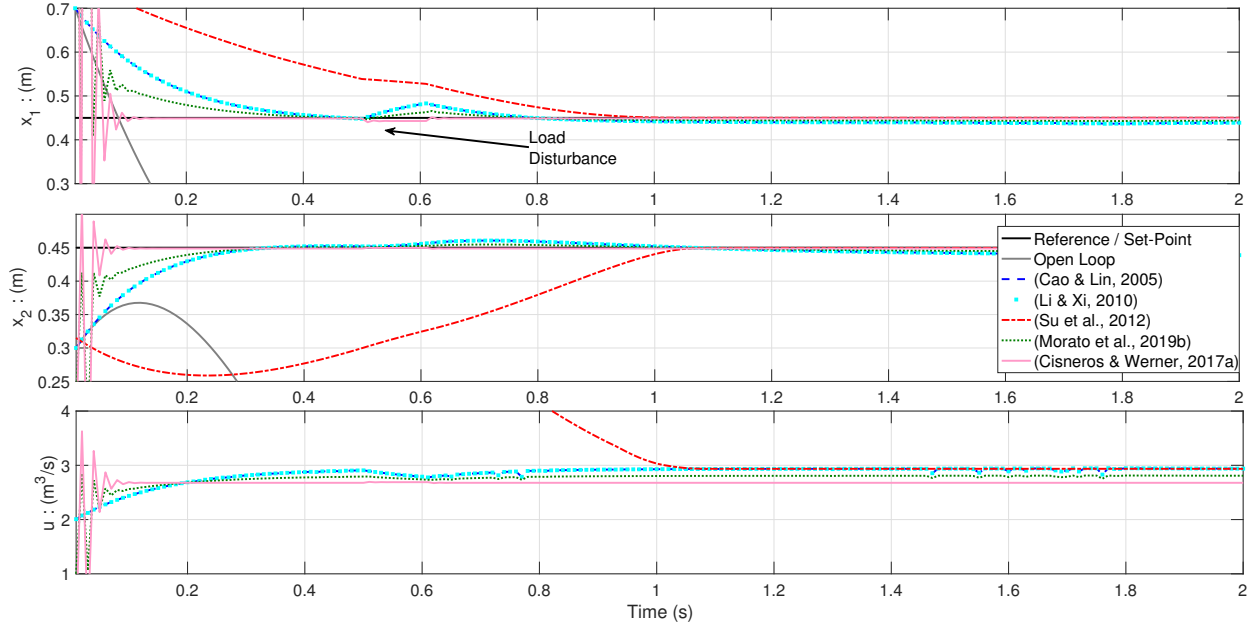


Figure 1: State Performances and Control Signal

Table 2: Tracking Results: *RMS* of Tracking Error

Method	Reference tracking for $x_1$	Reference tracking for $x_2$
Min/max <i>RMPC</i> : (Cao & Lin, 2005)	0.0489	0.0263
Min/max <i>RMPC</i> : (Li & Xi, 2010)	0.0367	0.0234
Robust <i>TMPC</i> : (Su et al., 2012)	0.1024	0.0989
Sub-optimal <i>LPV MPC</i> : (Morato et al., 2019b)	0.0258	0.0370
<i>NP LPV MPC</i> : (Cisneros & Werner, 2017a)	0.0229	0.0272

## 8. Available Investigation Gaps

740 Although there exist nowadays generalized formulations of *MPC* for nonlinear systems, the Linear Parameter Varying framework has become the most popular one for control purposes and, since the majority of nonlinear plants can be embedded into an *LPV* description, the previous Sections analyzed and discussed the available (robust and sub-optimal) predictive control methods for processes with Linear Parameter Varying models.

Considering the works mentioned in the presented literature review, their qualities and drawbacks, their applicability and their impact in the Control Systems community, it becomes evident that there is still some

vague space and available investigation threads in this context of *LPV MPC* design. These threads are mentioned in the sequel.

Firstly, it should be remarked that *LPV* formulation in the I/O form, although being supported by diverse data-driven identification procedures (Tóth et al., 2011a; Bachnas et al., 2014), is not at all common. The vast majority of available control methods for *LPV* systems (including predictive control design) are settled for state-space descriptions. Anyhow, industry is much more prone to accept I/O formulations, as discusses Froisy (2006); just to give an example, the original *GPC* algorithm (Clarke et al., 1987) has wide acceptance in many real-world applications due to its simple I/O prediction model, an issue that is discussed in (Darby & Nikolaou, 2012). Therefore, the bridge between theory and industrial applications for the case of *MPC* applied to *LPV* systems will be further sustained when a good body of research for I/O formulations is well-build. This research is fundamental and a very promising investigation area, with rather few papers and lacking assessments.

Nonlinear Programming versions of *LPV MPC* algorithms that use the  $\rho(k) = f_\rho(x(k), u(k))$  proxy to make the process predictions along the horizons have been recently shown to yield very fast results. As discussed, these algorithms have been shown to even outrank fast *NMPC* solvers, such as ACADO. Anyhow, the topic has only been studied only by a handful of papers and deserves further attention, especially regarding recursive feasibility and stability properties. It seems that their development can surely be established as a competitive category for nonlinear *MPC*.

Multiple Model *MPC* for the context of *LPV* systems is also a rather unexplored field. This type of algorithm may fall short of the robust methods and be equivalent to sub-optimal approaches, but comparison works to conclude on these characteristics are lacking. Nowadays, one cannot conclude if these algorithms are able to outperform search methods or *RMPC* design for the *LPV* case.

Robust *MPC* design translated to the *LPV* case is rather well-established (Li & Xi, 2010; Jungers et al., 2011; Abbas et al., 2018). These min-max (worst-case) algorithms can be synthesized via offline a-priori preparation (with *LMIs*), with terminal (ellipsoidal, polyhedral) target sets, for dynamic output-feedback and for the case of bounded parameter variation rates. The major drawback of the min-max optimization is that quite conservative results can be obtained with heavy numerical burden. To smooth this computational complexity, tube-based design shows itself quite promising for the *LPV* case. *LPV TMPC* is able to stabilize *LPV* systems to the origin (Hanema et al., 2017a) and provide good results with simpler *QPs*. The major drawback is that if the tubes are equivocally designed, the control performances may turn out excessively conservative. Rather few works have developed stabilizing *LPV TMPC* algorithms and further theoretical advances are lacking. This is an interesting research route specially concerning practical applications, which are yet unseen.

The prediction of the future scheduling parameter behaviour of *LPV* systems within the next sliding horizon of  $N_p$  steps also seems like an interesting route to follow. Possibly, one can also consider *MPC* applied for “nonlinear *PV* models”, i.e. *PV* models that keep some “easily handled” nonlinearities (such as local Lipschitz ones). These predictions, if coupled to the regular *MPC* algorithm, can translate the nonlinear programming procedure into a *QP*. But, this leads to sub-optimality of the control design and, therefore, feasibility and convergence guarantees need to be combined. Formal proofs on how contractive terminal sets and Lyapunov-decreasing stage costs for these kinds of algorithms are still to be generalized and formally conducted for *LPV* systems under *MPCs* with badly estimated future scheduling parameter behaviour.

The design of Fault-tolerant control (*FTC*) strategies (that can be derived from *MPC* loops) can be investigated, extending what was previously seen in (Rotondo et al., 2015, 2014). To tackle *FTC* problems<sup>6</sup>, as well as systems with delays is definitely under the research scope of *LPV MPC* procedures.

## 9. Conclusions

This paper surveyed the available predictive control techniques for the case of systems with Linear Parameter Varying models. The aim of this paper was to enhance the comprehension of the state-of-the-art

---

<sup>6</sup>Note that, in many situations - see (Morato et al., 2018, 2020) -, linear systems subject to faults become *LPV* ones.

on the topic of *LPV MPC* methods, concluding on the current situation, ongoing trends and open gaps that should be occupied. As evidenced, robust *MPC* methods have a wide applicability for the *LPV* context, but tube-based design, I/O formulations and sub-optimal methods with guaranteed recursive feasibility may yield similar control performances with less numerical burden. Furthermore, Nonlinear Programming formulations have been recently explored and achieved comparable numerical performances as ACADO, GRAMPC and CasADi *NMPC* solutions.

## Acknowledgments

This work has been supported by *CNPq* project 305785/2015 – 0 and *ITEA3* European project 15016 *EMPHYSIS*. We acknowledge the kind help by colleges Emanuel Bernardi and Hugo Pipino in structuring some simulations.

## References

- Abbas, H., & Werner, H. (2009). An instrumental variable technique for open-loop and closed-loop identification of input-output LPV models. In *2009 European Control Conference (ECC)* (pp. 2646–2651). IEEE.
- Abbas, H. S., Hanema, J., Tóth, R., Mohammadpour, J., & Meskin, N. (2018). A new approach to robust MPC design for LPV systems in input-output form. *IFAC-PapersOnLine*, *51*, 112–117.
- Abbas, H. S., Männel, G., né Hoffmann, C. H., & Rostalski, P. (2019). Tube-based model predictive control for linear parameter-varying systems with bounded rate of parameter variation. *Automatica*, *107*, 21–28.
- Abbas, H. S., Tóth, R., Meskin, N., Mohammadpour, J., & Hanema, J. (2015). An MPC approach for LPV systems in input-output form. In *IEEE Conference on Decision and Control* (pp. 91–96). IEEE.
- Abbas, H. S., Toth, R., Meskin, N., Mohammadpour, J., & Hanema, J. (2016). A robust MPC for input-output LPV models. *IEEE Transactions on Automatic Control*, *61*, 4183–4188.
- Abbas, H. S., Toth, R., Petreczky, M., Meskin, N., & Mohammadpour, J. (2014). Embedding of nonlinear systems in a linear parameter-varying representation. *IFAC Proceedings Volumes*, *47*, 6907–6913.
- Alamir, M. (2006). *Stabilization of nonlinear systems using receding-horizon control schemes*.
- Alamir, M. (2012). A framework for real-time implementation of low-dimensional parameterized NMPC. *Automatica*, *48*, 198–204.
- Alamir, M. (2013). *A pragmatic story of model predictive control: self-contained algorithms and case-studies*. CreateSpace Independent Publishing Platform.
- Alcalá, E., Puig, V., & Quevedo, J. (2019). LPV-MPC control for autonomous vehicles. In *Proceedings of the 3<sup>th</sup> IFAC Workshop on Linear Parameter Varying Systems, Eindhoven, The Netherlands, Nov. 4-6, 2019*.
- Allgöwer, F., & Zheng, A. (2012). *Nonlinear model predictive control* volume 26. Birkhäuser.
- Amato, F., & Mattei, M. (2001). Robust control of a plasma wind tunnel: An LPV discrete-time system depending on fast/slowly-varying parameters. *European journal of control*, *7*, 494–510.
- Amato, F., Mattei, M., & Pironti, A. (2005). Gain scheduled control for discrete-time systems depending on bounded rate parameters. *International Journal of Robust and Nonlinear Control*, *15*, 473–494.
- Andersson, J. A., Gillis, J., Horn, G., Rawlings, J. B., & Diehl, M. (2019). CasADi: a software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, *11*, 1–36.
- Angeli, D., Casavola, A., Franzè, G., & Mosca, E. (2008). An ellipsoidal off-line MPC scheme for uncertain polytopic discrete-time systems. *Automatica*, *44*, 3113–3119.
- Arbabi, H., Korda, M., & Mezić, I. (2018). A data-driven koopman model predictive control framework for nonlinear flows. *arXiv preprint arXiv:1804.05291*, .
- Arkun, Y., Banerjee, A., & Lakshmanan, N. (1998). Self-scheduling MPC using LPV models. In *Nonlinear Model Based Process Control* (pp. 59–84). Springer.
- Aufderheide, B., & Bequette, B. W. (2003). Extension of dynamic matrix control to multiple models. *Computers & chemical engineering*, *27*, 1079–1096.
- Ayala, C. O., Roca, L., Guzman, J. L., Normey-Rico, J. E., Berenguel, M., & Yebra, L. (2011). Local model predictive controller in a solar desalination plant collector field. *Renewable energy*, *36*, 3001–3012.
- Bachnas, A., Tóth, R., Ludlage, J., & Mesbah, A. (2014). A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study. *Journal of Process Control*, *24*, 272–285.
- Baocang, D., Lihua, X., & Xue, F. (2009). Improving robust model predictive control via dynamic output feedback. In *2009 Chinese Control and Decision Conference* (pp. 2116–2121). IEEE.
- Besselmann, T., Lofberg, J., & Morari, M. (2008). Explicit model predictive control for linear parameter-varying systems. In *IEEE Conference on Decision and Control* (pp. 3848–3853). IEEE.
- Besselmann, T., Löfberg, J., & Morari, M. (2009a). Constrained time-optimal control of linear parameter-varying systems. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference* (pp. 6923–6928). IEEE.

- Besselmann, T., Löfberg, J., & Morari, M. (2009b). Explicit LPV-MPC with bounded rate of parameter variation. *IFAC Proceedings Volumes*, *42*, 7–12.
- Besselmann, T., Lofberg, J., & Morari, M. (2012). Explicit MPC for LPV systems: Stability and optimality. *IEEE Transactions on Automatic Control*, *57*, 2322–2332.
- Blanchini, F. (1999). Set invariance in control. *Automatica*, *35*, 1747–1767.
- Blanchini, F., & Miani, S. (2003). Stabilization of LPV systems: state feedback, state estimation, and duality. *SIAM journal on control and optimization*, *42*, 76–97.
- Bock, H. G., Diehl, M., Kühl, P., Kostina, E., Schiöder, J. P., & Wirsching, L. (2007). Numerical methods for efficient and fast nonlinear model predictive control. In *Assessment and future directions of nonlinear model predictive control* (pp. 163–179). Springer.
- Bokor, J., & Balas, G. (2005). Linear parameter varying systems: A geometric theory and applications. *IFAC Proceedings Volumes*, *38*, 12–22.
- Boyd, S., El Ghaoui, L., Feron, E., & Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory* volume 15. Siam.
- Bravo, C. O. A., & Normey-Rico, J. E. (2009). Approach for non-linear predictive control based on the local model ideas. *SBA: Controle & Automação Sociedade Brasileira de Automatica*, *20*, 465–481.
- Bravo, J. M., Limón, D., Alamo, T., & Camacho, E. F. (2005). On the computation of invariant sets for constrained nonlinear systems: An interval arithmetic approach. *Automatica*, *41*, 1583–1589.
- Brooms, A., Kouvaritakis, B., & Lee, Y. I. (2001). Constrained MPC for uncertain linear systems with ellipsoidal target sets. *Systems & Control Letters*, *44*, 157–166.
- Brunner, F. D., Lazar, M., & Allgöwer, F. (2013). An explicit solution to constrained stabilization via polytopic tubes. In *52nd IEEE Conference on Decision and Control* (pp. 7721–7727). IEEE.
- Bumroongsri, P. (2014). An offline formulation of MPC for LPV systems using linear matrix inequalities. *Journal of Applied Mathematics*, *2014*.
- Bumroongsri, P., & Kheawhom, S. (2012a). An ellipsoidal off-line model predictive control strategy for linear parameter varying systems with applications in chemical processes. *Systems & Control Letters*, *61*, 435–442.
- Bumroongsri, P., & Kheawhom, S. (2012b). An off-line robust MPC algorithm for uncertain polytopic discrete-time systems using polyhedral invariant sets. *Journal of Process Control*, *22*, 975–983.
- Calafiore, G. C., & Fagiano, L. (2012). Robust model predictive control via scenario optimization. *IEEE Transactions on Automatic Control*, *58*, 219–224.
- Calafiore, G. C., & Fagiano, L. (2013). Stochastic model predictive control of LPV systems via scenario optimization. *Automatica*, *49*, 1861–1866.
- Calderón, H. M., Cisneros, P. S. G., & Werner, H. (2019). qLPV predictive control - a benchmark study on state space vs input-output approach. In *Proceedings of the 3<sup>th</sup> IFAC Workshop on Linear Parameter Varying Systems, Eindhoven, The Netherlands, Nov. 4-6, 2019*.
- Camacho, E. F., & Bordons, C. (2007). Nonlinear model predictive control: An introductory review. In *Assessment and future directions of nonlinear model predictive control* (pp. 1–16). Springer.
- Camacho, E. F., & Bordons, C. (2013). *Model predictive control*. Springer Science & Business Media.
- Cannon, M. (2004). Efficient nonlinear model predictive control algorithms. *Annual Reviews in Control*, *28*, 229–237.
- Cannon, M., Buerger, J., Kouvaritakis, B., & Rakovic, S. (2011). Robust tubes in nonlinear model predictive control. *IEEE Transactions on Automatic Control*, *56*, 1942–1947.
- Cannon, M., Kouvaritakis, B., Lee, Y., & Brooms, A. (2001). Efficient non-linear model based predictive control. *International Journal of Control*, *74*, 361–372.
- Cao, Y.-Y., & Lin, Z. (2005). Min-max MPC algorithm for LPV systems subject to input saturation. *IET Proceedings-Control Theory and Applications*, *152*, 266–272.
- Casavola, A., Famularo, D., & Franze, G. (2002). A feedback min-max MPC algorithm for LPV systems subject to bounded rates of change of parameters. *IEEE Transactions on Automatic Control*, *47*, 1147–1153.
- Casavola, A., Famularo, D., & Franzè, G. (2003). Predictive control of constrained nonlinear systems via LPV linear embeddings. *International Journal of Robust and Nonlinear Control*, *13*, 281–294.
- Casavola, A., Famularo, D., Franze, G., & Garone, E. (2006). An improved predictive control strategy for polytopic LPV linear systems. In *Proceedings of the 45<sup>th</sup> Conference on Decision and Control* (pp. 5820–5825). IEEE.
- Casavola, A., Famularo, D., Franzè, G., & Garone, E. (2012). A fast ellipsoidal MPC scheme for discrete-time polytopic linear parameter varying systems. *Automatica*, *48*, 2620–2626.
- Casella, F., & Lovera, M. (2008). LPV/LFT modelling and identification: overview, synergies and a case study. In *IEEE International Conference on Computer-Aided Control Systems* (pp. 852–857). IEEE.
- Cisneros, P. G., Voss, S., Datar, A., Martinez, H., Gottsch, P., & Werner, H. (2019). Plenary at the 3<sup>rd</sup> IFAC workshop on linear parameter varying systems: Fast Nonlinear MPC via quasi-LPV Models. URL: <https://lpvs2019.tue.nl/plenary-speakers/>.
- Cisneros, P. G., & Werner, H. (2017a). Fast nonlinear MPC for reference tracking subject to nonlinear constraints via quasi-LPV representations. *IFAC-PapersOnLine*, *50*, 11601–11606.
- Cisneros, P. S., Sridharan, A., & Werner, H. (2018). Constrained predictive control of a robotic manipulator using quasi-LPV representations. *IFAC-PapersOnLine*, *51*, 118–123.
- Cisneros, P. S., Voss, S., & Werner, H. (2016). Efficient nonlinear model predictive control via quasi-LPV representation. In *IEEE Conference on Decision and Control* (pp. 3216–3221). IEEE.
- Cisneros, P. S., & Werner, H. (2017b). Parameter-dependent stability conditions for quasi-LPV model predictive control. In

- American Control Conference* (pp. 5032–5037). IEEE.
- 915 Cisneros, P. S., & Werner, H. (2018). A dissipativity formulation for stability analysis of nonlinear and parameter dependent MPC. In *Annual American Control Conference* (pp. 3894–3899). IEEE.
- Cisneros, P. S. G., & Werner, H. (2019). Wide range stabilization of a pendubot using quasi-LPV predictive control. In *Proceedings of the 3<sup>th</sup> IFAC Workshop on Linear Parameter Varying Systems, Eindhoven, The Netherlands, Nov. 4-6, 2019*.
- 920 Clarke, D. W., Mohtadi, C., & Tuffs, P. (1987). Generalized predictive control: Part i. the basic algorithm. *Automatica*, *23*, 137–148.
- Cutler, C. R., & Ramaker, B. L. (1980). Dynamic matrix control: A computer control algorithm. In *Joint Automatic Control Conference* 17 (p. 72).
- Daafouz, J., & Bernussou, J. (2001). Parameter dependent lyapunov functions for discrete time systems with time varying parametric uncertainties. *Systems & control letters*, *43*, 355–359.
- 925 Darby, M. L., & Nikolaou, M. (2012). MPC: Current practice and challenges. *Control Engineering Practice*, *20*, 328–342.
- De Caigny, J., Camino, J. F., Oliveira, R. C., Peres, P. L., & Swevers, J. (2012). Gain-scheduled dynamic output feedback control for discrete-time LPV systems. *International Journal of Robust and Nonlinear Control*, *22*, 535–558.
- Ding, B., & Pan, H. (2016). Output feedback robust MPC for LPV system with polytopic model parametric uncertainty and bounded disturbance. *International Journal of Control*, *89*, 1554–1571.
- 930 Ding, B., Ping, X., & Pan, H. (2013). On dynamic output feedback robust MPC for constrained quasi-LPV systems. *International Journal of Control*, *86*, 2215–2227.
- Englert, T., Völz, A., Mesmer, F., Rhein, S., & Graichen, K. (2019). A software framework for embedded nonlinear model predictive control using a gradient-based augmented lagrangian approach (GRAMPC). *Optimization and Engineering*, *20*, 769–809.
- 935 Feng, L., Wang, J. L., & Poh, E. K. (2007). Improved robust model predictive control with structured uncertainty. *Journal of Process Control*, *17*, 683–688.
- Ferramosca, A., Limón, D., Alvarado, I., Alamo, T., & Camacho, E. F. (2008). MPC for tracking with optimal closed-loop performance. In *IEEE Conference on Decision and Control* (pp. 4055–4060). IEEE.
- 940 Ferramosca, A., Limón, D., Alvarado, I., Alamo, T., & Camacho, E. F. (2009). MPC for tracking of constrained nonlinear systems. In *IEEE Conference on Decision and Control* (pp. 7978–7983). IEEE.
- Fleming, J., Kouvaritakis, B., & Cannon, M. (2014). Robust tube MPC for linear systems with multiplicative uncertainty. *IEEE Transactions on Automatic Control*, *60*, 1087–1092.
- Froisy, J. B. (2006). Model predictive control building a bridge between theory and practice. *Computers & chemical engineering*, *30*, 1426–1435.
- 945 Garriga, J. L., & Soroush, M. (2010). Model predictive control tuning methods: A review. *Industrial & Engineering Chemistry Research*, *49*, 3505–3515.
- Gesser, R. S., Lima, D. M., & Normey-Rico, J. E. (2018). Robust model predictive control: Implementation issues with comparative analysis. *IFAC-PapersOnLine*, *51*, 478–483.
- 950 Gonzalez, R., Fiacchini, M., Alamo, T., Guzmán, J. L., & Rodríguez, F. (2011). Online robust tube-based MPC for time-varying systems: A practical approach. *International Journal of Control*, *84*, 1157–1170.
- Gros, S., Zanon, M., Quirynen, R., Bemporad, A., & Diehl, M. (2020). From linear to nonlinear MPC: bridging the gap via the real-time iteration. *International Journal of Control*, *93*, 62–80.
- Hanema, J., Lazar, M., & Tóth, R. (2017a). Stabilizing tube-based model predictive control: Terminal set and cost construction for LPV systems. *Automatica*, *85*, 137–144.
- 955 Hanema, J., Lazar, M., & Tóth, R. (2020). Heterogeneously parameterized tube model predictive control for LPV systems. *Automatica*, *111*, 108622.
- Hanema, J., Tóth, R., & Lazar, M. (2016). Tube-based anticipative model predictive control for linear parameter-varying systems. In *55<sup>th</sup> Conference on Decision and Control* (pp. 1458–1463). IEEE.
- 960 Hanema, J., Tóth, R., & Lazar, M. (2017b). Stabilizing non-linear MPC using linear parameter-varying representations. In *56<sup>th</sup> Annual Conference on Decision and Control* (pp. 3582–3587). IEEE.
- Hariprasad, K., & Bhartiya, S. (2014). Adaptive robust model predictive control of nonlinear systems using tubes based on interval inclusions. In *53rd IEEE Conference on Decision and Control* (pp. 2032–2037). IEEE.
- Hariprasad, K., Bhartiya, S., & Gudi, R. D. (2012). A gap metric based multiple model approach for nonlinear switched systems. *Journal of process control*, *22*, 1743–1754.
- 965 Heath, W., Li, G., Wills, A., & Lennox, B. (2006). The robustness of input constrained model predictive control to infinity-norm bound model uncertainty. *IFAC Proceedings Volumes*, *39*, 495–500.
- Heath, W., Wills, A. G., & Akkermans, J. (2005). A sufficient condition for the stability of optimizing controllers with saturating actuators. *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, *15*, 515–529.
- 970 Heath, W. P., & Li, G. (2010). Multipliers for model predictive control with structured input constraints. *Automatica*, *46*, 562–568.
- Heath, W. P., & Wills, A. G. (2005). Zames-falb multipliers for quadratic programming. In *Proceedings of the 44<sup>th</sup> IEEE Conference on Decision and Control* (pp. 963–968). IEEE.
- Hoffmann, C., Hashemi, S. M., Abbas, H. S., & Werner, H. (2014). Synthesis of LPV controllers with low implementation complexity based on a reduced parameter set. *IEEE Transactions on Control Systems Technology*, *22*, 2393–2398.
- 975 Hoffmann, C., & Werner, H. (2014). A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on Control Systems Technology*, *23*, 416–433.
- Houska, B., Ferreau, H. J., & Diehl, M. (2011). An auto-generated real-time iteration algorithm for nonlinear MPC in the

- microsecond range. *Automatica*, *47*, 2279–2285.
- 980 Ileš, Š., Lazar, M., & Matuško, J. (2015). Stabilizing model predictive control based on flexible set-membership constraints. In *European Control Conference* (pp. 3358–3364). IEEE.
- Jin, X., Huang, B., & Shook, D. S. (2011). Multiple model LPV approach to nonlinear process identification with EM algorithm. *Journal of Process Control*, *21*, 182–193.
- Johansen, T. A., & Grancharova, A. (2003). Approximate explicit constrained linear model predictive control via orthogonal search tree. *IEEE Transactions on Automatic Control*, *48*, 810–815.
- 985 Jungers, M., Caun, R. P., Oliveira, R. C. L. F., & Peres, P. L. D. (2009). Model predictive control for linear parameter varying systems using path-dependent lyapunov functions. *IFAC Proceedings Volumes*, *42*, 97–102.
- Jungers, M., Oliveira, R. C., & Peres, P. L. (2011). MPC for LPV systems with bounded parameter variations. *International Journal of Control*, *84*, 24–36.
- 990 Käpernick, B., & Graichen, K. (2014). The gradient based nonlinear model predictive control software GRAMPC. In *European Control Conference* (pp. 1170–1175). IEEE.
- Keerthi, S. a., & Gilbert, E. G. (1988). Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems: Stability and moving-horizon approximations. *Journal of optimization theory and applications*, *57*, 265–293.
- Köhler, J., Müller, M. A., & Allgöwer, F. (2019). A nonlinear model predictive control framework using reference generic terminal ingredients. *IEEE Transactions on Automatic Control*, .
- 995 Korda, M., & Mezić, I. (2018). Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, *93*, 149–160.
- Kothare, M. V., Balakrishnan, V., & Morari, M. (1996). Robust constrained model predictive control using linear matrix inequalities. *Automatica*, *32*, 1361–1379.
- 1000 Langson, W., Chrysochoos, I., Raković, S., & Mayne, D. Q. (2004). Robust model predictive control using tubes. *Automatica*, *40*, 125–133.
- Lazar, M., Heemels, W. P. M. H., Weiland, S., & Bemporad, A. (2006). Stabilizing model predictive control of hybrid systems. *IEEE Transactions on Automatic Control*, *51*, 1813–1818.
- Lazar, M., & Spinu, V. (2015). Finite-step terminal ingredients for stabilizing model predictive control. *IFAC-PapersOnLine*, *48*, 9–15.
- 1005 Lee, S.-M., & Park, J. H. (2007). Output feedback model predictive control for LPV systems using parameter-dependent lyapunov function. *Applied Mathematics and Computation*, *190*, 671–676.
- Li, D., & Xi, Y. (2010). The feedback robust MPC for LPV systems with bounded rates of parameter changes. *IEEE Transactions on Automatic Control*, *55*, 503–507.
- 1010 Limon, D., Alamo, T., & Camacho, E. F. (2005). Enlarging the domain of attraction of MPC controllers. *Automatica*, *41*, 629–635.
- Limón, D., Alvarado, I., Alamo, T., & Camacho, E. F. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, *44*, 2382–2387.
- Liu, Q., Abbas, H. S., Mohammadpour, J., Wollnack, S., & Werner, H. (2016). Distributed model predictive control of constrained spatially-invariant interconnected systems in input-output form. In *American Control Conference* (pp. 3600–3605). IEEE.
- Löfberg, J. (2003). *Minimax approaches to robust model predictive control* volume 812. Linköping University Electronic Press.
- Löfberg, J. (2012). Oops! i cannot do it again: Testing for recursive feasibility in MPC. *Automatica*, *48*, 550–555.
- Magni, L., & Scattolini, R. (2004). Stabilizing model predictive control of nonlinear continuous time systems. *Annual Reviews in Control*, *28*, 1–11.
- 1020 Mate, S., Kodamana, H., Bhartiya, S., & Nataraj, P. S. V. (2019). A stabilizing sub-optimal model predictive control for quasi-linear parameter varying systems. *IEEE Control Systems Letters*, .
- Mayne, D. (2016). Robust and stochastic model predictive control: Are we going in the right direction? *Annual Reviews in Control*, *41*, 184–192.
- 1025 Mayne, D. Q. (2014). Model predictive control: Recent developments and future promise. *Automatica*, *50*, 2967–2986.
- Mayne, D. Q., Kerrigan, E. C., Van Wyk, E. J., & Falugi, P. (2011). Tube-based robust nonlinear model predictive control. *International Journal of Robust and Nonlinear Control*, *21*, 1341–1353.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V., & Sokaert, P. O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, *36*, 789–814.
- 1030 Mayne, D. Q., Seron, M. M., & Raković, S. V. (2005). Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, *41*, 219–224.
- Mesbah, A. (2016). Stochastic model predictive control: An overview and perspectives for future research. *IEEE Control Systems Magazine*, *36*, 30–44.
- Michalska, H., & Mayne, D. Q. (1993). Robust receding horizon control of constrained nonlinear systems. *IEEE transactions on automatic control*, *38*, 1623–1633.
- 1035 Mohammadpour, J., & Scherer, C. W. (2012). *Control of linear parameter varying systems with applications*. Springer Science & Business Media.
- Montagner, V. F., Oliveira, R. C., & Peres, P. L. (2007). Necessary and sufficient LMI conditions to compute quadratically stabilizing state feedback controllers for takagi-sugeno systems. In *2007 American Control Conference* (pp. 4059–4064). IEEE.
- 1040 Morato, M. M., Mendes, P. R. C., Normey-Rico, J. E., & Bordons, C. (2019a). LPV-filtered predictive control design for fault-tolerant energy managements. In *Proceedings of the 12<sup>th</sup> IFAC Symposium on Dynamics and Control of Process Systems, including Biosystems, Florianópolis, Brazil, Apr. 23-26, 2019*.



- 1045 Morato, M. M., Mendes, P. R. C., Normey-Rico, J. E., & Bordons, C. (2020). LPV-MPC fault-tolerant energy management strategy for renewable microgrids. *International Journal of Electrical Power & Energy Systems*, *117*, 105644. doi:<https://doi.org/10.1016/j.ijepes.2019.105644>.
- Morato, M. M., Normey-Rico, J. E., & Sename, O. (2019b). Novel qLPV MPC design with least-squares scheduling prediction. In *Proceedings of the 3<sup>th</sup> IFAC Workshop on Linear Parameter Varying Systems, Eindhoven, The Netherlands, Nov. 4-6, 2019*.
- 1050 Morato, M. M., Sename, O., & Dugard, L. (2018). LPV-MPC fault tolerant control of automotive suspension dampers. *IFAC-PapersOnLine*, *51*, 31–36.
- Muñoz-Carpintero, D., Cannon, M., & Kouvaritakis, B. (2015). Robust MPC strategy with optimized polytopic dynamics for linear systems with additive and multiplicative uncertainty. *Systems & Control Letters*, *81*, 34–41.
- Nandola, N. N., & Bhartiya, S. (2008). A multiple model approach for predictive control of nonlinear hybrid systems. *Journal of process control*, *18*, 131–148.
- 1055 Nguyen, H.-N., Oлару, S., & Gutman, P.-O. (2013). One-step receding horizon control for LPV systems in presence of constraints. In *American Control Conference* (pp. 6835–6840). IEEE.
- Nguyen, M.-Q., Canale, M., Sename, O., & Dugard, L. (2016). A model predictive control approach for semi-active suspension control problem of a full car. In *55<sup>th</sup> Conference on Decision and Control* (pp. 721–726). IEEE.
- 1060 Normey-Rico, J., & Camacho, E. (2007). *Control of dead-time processes*. Springer Science & Business Media.
- Park, J.-H., Kim, T.-H., & Sugie, T. (2011). Output feedback model predictive control for LPV systems based on quasi-min-max algorithm. *Automatica*, *47*, 2052–2058.
- Park, P., & Jeong, S. C. (2004). Constrained RHC for LPV systems with bounded rates of parameter variations. *Automatica*, *40*, 865–872.
- 1065 Pipino, H. A., & Adam, E. J. (2019). MPC for linear systems with parametric uncertainty. In *XVIII Workshop on Information Processing and Control* (pp. 42–47). IEEE.
- Pistikopoulos, E. (2009). Perspectives in multiparametric programming and explicit model predictive control. *AIChE journal*, *55*, 1918–1925.
- Pluymers, B., Rossiter, J., Suykens, J., & De Moor, B. (2005a). The efficient computation of polyhedral invariant sets for linear systems with polytopic uncertainty. In *Proceedings of the 2005, American Control Conference, 2005*. (pp. 804–809). IEEE.
- 1070 Pluymers, B., Rossiter, J. A., Suykens, J. A. K., & De Moor, B. (2005b). Interpolation-based MPC for LPV systems using polyhedral invariant sets. In *American Control Conference* (pp. 810–815). IEEE.
- Qin, S. J., & Badgwell, T. A. (2003). A survey of industrial model predictive control technology. *Control engineering practice*, *11*, 733–764.
- 1075 Quirynen, R., Vukov, M., Zanon, M., & Diehl, M. (2015). Autogenerating microsecond solvers for nonlinear MPC: a tutorial using ACADO integrators. *Optimal Control Applications and Methods*, *36*, 685–704.
- Raimondo, D. M., Alamo, T., Limón, D., & Camacho, E. F. (2007). Towards the practical implementation of min-max nonlinear model predictive control. In *IEEE Conference on Decision and Control* (pp. 1257–1262). IEEE.
- 1080 Raković, S. (2015). Robust model-predictive control. *Encyclopedia of Systems and Control*, (pp. 1225–1233).
- Rakovic, S. V., Kouvaritakis, B., Cannon, M., Panos, C., & Findeisen, R. (2012). Parameterized tube model predictive control. *IEEE Transactions on Automatic Control*, *57*, 2746–2761.
- Rathai, K. M. M., Alamir, M., Sename, O., & Tang, R. (2018). A parameterized NMPC scheme for embedded control of semi-active suspension system. *IFAC-PapersOnLine*, *51*, 301–306.
- 1085 Richter, S., Jones, C. N., & Morari, M. (2011). Computational complexity certification for real-time MPC with input constraints based on the fast gradient method. *IEEE Transactions on Automatic Control*, *57*, 1391–1403.
- Rotondo, D., Nejjari, F., & Puig, V. (2014). A virtual actuator and sensor approach for fault tolerant control of LPV systems. *Journal of Process Control*, *24*, 203–222.
- 1090 Rotondo, D., Nejjari, F., & Puig, V. (2015). Robust quasi-LPV model reference FTC of a quadrotor UAV subject to actuator faults. *International Journal of Applied Mathematics and Computer Science*, *25*, 7–22.
- Saltik, M. B., Özkan, L., Ludlage, J. H., Weiland, S., & Van den Hof, P. M. (2018). An outlook on robust model predictive control algorithms: Reflections on performance and computational aspects. *Journal of Process Control*, *61*, 77–102.
- Santos, M. A., Ferramosca, A., & Raffo, G. V. (2018). Tube-based MPC with nonlinear control for load transportation using a UAV. *IFAC-PapersOnLine*, *51*, 459–465.
- 1095 Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control—a review. *Journal of process control*, *19*, 723–731.
- Scherer, C. W. (2001). LPV control and full block multipliers. *Automatica*, *37*, 361–375.
- Scorletti, G., Fromion, V., & De Hillerin, S. (2015). Toward nonlinear tracking and rejection using LPV control. *IFAC-PapersOnLine*, *48*, 13–18.
- 1100 Seiler, P., Packard, A., & Balas, G. J. (2010). A dissipation inequality formulation for stability analysis with integral quadratic constraints. In *Proceedings of the 49<sup>th</sup> IEEE Conference on Decision and Control* (pp. 2304–2309). IEEE.
- Sename, O., Gaspar, P., & Bokor, J. (2013). *Robust control and linear parameter varying approaches: application to vehicle dynamics* volume 437. Springer.
- 1105 Shamma, J. S. (2012). An overview of LPV systems. In *Control of linear parameter varying systems with applications* (pp. 3–26). Springer.
- Su, Y., Tan, K. K., & Lee, T. H. (2012). Tube-based quasi-min-max output feedback MPC for LPV systems. In *Preprints of the 8<sup>th</sup> IFAC Symposium on advanced control of chemical processes* (pp. 10–13). IFAC.
- Suzukia, H., & Sugie, T. (2006). MPC for LPV systems with bounded parameter variation using ellipsoidal set prediction. In

- 2006 American Control Conference (pp. 6–pp). IEEE.
- 1110 Szabó, Z., & Bokor, J. (2018). Transformations for linear parameter varying systems. *IFAC-PapersOnLine*, 51, 87–93.
- Torrico, B. C., Roca, L., Normey-Rico, J. E., Guzman, J. L., & Yebra, L. (2010). Robust nonlinear predictive control applied to a solar collector field in a solar desalination plant. *IEEE Transactions on Control Systems Technology*, 18, 1430–1439.
- Tóth, R. (2010). *Modeling and identification of linear parameter-varying systems* volume 403. Springer.
- 1115 Tóth, R., Abbas, H. S., & Werner, H. (2011a). On the state-space realization of LPV input-output models: Practical approaches. *IEEE transactions on control systems technology*, 20, 139–153.
- Tóth, R., Willems, J. C., Heuberger, P. S., & Van den Hof, P. M. (2011b). The behavioral approach to linear parameter-varying systems. *IEEE Transactions on Automatic Control*, 56, 2499–2514.
- Vesely, V., Rosinová, D., & Foltin, M. (2010). Robust model predictive control design with input constraints. *ISA transactions*, 49, 114–120.
- 1120 Wada, N., Saito, K., & Saeki, M. (2004). Model predictive control for linear parameter varying systems using parameter-dependent lyapunov function. In *47th Midwest Symposium on Circuits and Systems* (pp. iii–133). IEEE volume 3.
- Wang, L., & Young, P. C. (2006). An improved structure for model predictive control using non-minimal state space realisation. *Journal of Process Control*, 16, 355–371.
- 1125 Wollnack, S., Abbas, H. S., Tóth, R., & Werner, H. (2017). Fixed-structure LPV-IO controllers: An implicit representation based approach. *Automatica*, 83, 282–289.
- Wollnack, S., & Werner, H. (2016). LPV-IO controller design: An LMI approach. In *American Control Conference* (pp. 4617–4622). IEEE.
- Wu, F., Yang, X. H., Packard, A., & Becker, G. (1997). Induced  $l_2$ -norm control for LPV systems with bounded parameter variation rates. In *IEEE conference on Decision and Control*.
- 1130 Xu, F., Chen, H., Gong, X., & Mei, Q. (2015). Fast nonlinear model predictive control on FPGA using particle swarm optimization. *IEEE Transactions on Industrial Electronics*, 63, 310–321.
- Yu, S., Böhm, C., Chen, H., & Allgöwer, F. (2012). Model predictive control of constrained LPV systems. *International Journal of Control*, 85, 671–683.
- Yu-Geng, X., De-Wei, L., & Shu, L. (2013). Model predictive control status and challenges. *Acta Automatica Sinica*, 39, 222–236.
- 1135 Zhang, Y., Li, S., & Liao, L. (2019). Near-optimal control of nonlinear dynamical systems: A brief survey. *Annual Reviews in Control*, .
- Zhou, K., Doyle, J. C., Glover, K. et al. (1996). *Robust and optimal control* volume 40. Prentice hall New Jersey.

## Appendix A. Linear Differential Inclusion

1140 In this Section, an example of how a nonlinear process can be represented in either SS or I/O LPV formulations is given, to facilitate the understanding of this paper.

Consider a cascade of two cylindrical water tank, with equal base areas  $A_b$ . The upper tank is flooded by an open-valve inlet flow at its top and deflated by a regular open hole. The second tank is flooded by the outlet of the first tank, while also having a regular open hole at its bottom. The inlet flow of water to 1145 the upper tank is given by  $u(t)$ , while its outlet flow is given by  $\sqrt{2gh_1(t)}$ , being  $h_1(t)$  the level of the water in this tank and  $g$  the gravitational constant. Equivalently, the outlet flow of the bottom tank is given by  $\sqrt{2gh_2(t)}$ , being  $h_2$  its water level. Assume that both levels are constantly measured; the controlled output is  $y = h_2$ . The dynamics of these cascaded tanks are directly found:

$$A_b \frac{dh_1(t)}{dt} = u(t) - \sqrt{2g} \sqrt{h_1(t)}, \quad (\text{A.1})$$

$$A_b \frac{dh_2(t)}{dt} = \sqrt{2g} \sqrt{h_1(t)} - \sqrt{2g} \sqrt{h_2(t)}, \quad (\text{A.2})$$

$$(\text{A.3})$$

1150 which, considering an Euler discretization with a  $T_s$  sampling period, being  $t = kT_s$ , can be described through:

$$\overbrace{\begin{bmatrix} h_1(k+1) \\ h_2(k+1) \end{bmatrix}}^{h(k+1)} = f_x(h(k)) = \begin{bmatrix} h_1(k) - \frac{T_s}{A_b} \sqrt{2g} \sqrt{h_1(k)} + \frac{T_s}{A_b} u(k) \\ h_2(k) - \frac{T_s}{A_b} \sqrt{2g} \sqrt{h_2(k)} + \frac{T_s}{A_b} \sqrt{2g} \sqrt{h_1(k)} \end{bmatrix}, \quad (\text{A.4})$$

$$y(k) = f_y(h(k)) = h_2(k). \quad (\text{A.5})$$

The LDI property is easily verified:

$$\begin{bmatrix} \frac{f_x(h(k))}{f_y(h(k))} \end{bmatrix} = \overbrace{\begin{bmatrix} (1 - \frac{T_s \sqrt{2g}}{A_b} \frac{\sqrt{h_1(k)}}{h_1(k)}) & 0 & \frac{T_s}{A_b} \\ (\frac{T_s \sqrt{2g}}{A_b} \frac{\sqrt{h_1(k)}}{h_1(k)}) & (1 - \frac{T_s \sqrt{2g}}{A_b} \frac{\sqrt{h_2(k)}}{h_2(k)}) & 0 \\ 0 & 1 & 0 \end{bmatrix}}^{G(h(k))} \begin{bmatrix} h(k) \\ u(k) \end{bmatrix}, \quad (\text{A.6})$$

where  $G(h(k))$  always exists, since the discontinuity  $h_1 = h_2 = 0$  is assumed to be avoided by the control action  $u$ .

For a SS LPV formulation, it follows that  $\rho(k) = \{\rho_1(k), \rho_2(k)\}$ , with  $\rho_1(k) = (\frac{T_s \sqrt{2g}}{A_b} \frac{\sqrt{h_1(k)}}{h_1(k)})$  and  $\rho_2(k) = (\frac{T_s \sqrt{2g}}{A_b} \frac{\sqrt{h_2(k)}}{h_2(k)})$ , which yields, for  $x(k) = h(k)$ :

$$x(k+1) = A(\rho(k))x(k) + Bu(k), \quad (\text{A.7})$$

$$A(\rho(k)) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix} \text{diag}\{\rho(k)\} \quad \text{and} \quad B = [\frac{T_s}{A} \quad 0]^T. \quad (\text{A.8})$$

Taking the same scheduling parameters, an I/O formulation is easily casted as:

$$(1 + a_1(\rho(k))z^{-1} + a_2(\rho(k)))y(k) = b_2(\rho(k))z^{-2}u(k), \quad (\text{A.9})$$

$$a_1(\rho(k)) = 2 - \rho_1(k) - \rho_2(k), \quad (\text{A.10})$$

$$a_2(\rho(k)) = 1 - \rho_1(k) - \rho_2(k) + \rho_1(k)\rho_2(k), \quad (\text{A.11})$$

$$b_2(\rho(k)) = \frac{T_s}{A_b} \rho_1(k). \quad (\text{A.12})$$

This concludes the example.