



HAL
open science

A Tracking Model Predictive Control for Input-Output LPV Systems using Parameter Extrapolation

Marcelo Menezes Morato, Julio E Normey-Rico, Olivier Sename

► **To cite this version:**

Marcelo Menezes Morato, Julio E Normey-Rico, Olivier Sename. A Tracking Model Predictive Control for Input-Output LPV Systems using Parameter Extrapolation. LPVS 2022 - 5th IFAC Workshop on Linear Parameter Varying Systems, Sep 2022, Montreal, Canada. pp.55-60, 10.1016/j.ifacol.2022.11.290 . hal-03714308

HAL Id: hal-03714308

<https://hal.science/hal-03714308v1>

Submitted on 5 Jul 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

A Tracking Model Predictive Control for Input-Output LPV Systems using Parameter Extrapolation ^{*}

Marcelo M. Morato ^{*,**} Julio E. Normey-Rico ^{*}
Olivier Sename ^{**}

^{*} *Dept. de Automação e Sis., Univ. Fed. de Santa Catarina, Brazil.*
^{**} *Univ. Grenoble Alpes, CNRS, Grenoble INP[†], GIPSA-lab, 38000 Grenoble, France. [†]Institute of Engineering Univ. Grenoble Alpes. (marcelomnzm@gmail.com)*

Abstract: This paper presents a novel Model Predictive Control (MPC) algorithm for Linear Parameter Varying (LPV) systems represented in the Input-Output (IO) form. The proposed MPC is derived using estimates for the future scheduling parameter trajectory, made viable through a recursive Taylor-based extrapolation law. The method also includes explicit integral action, which, coupled with quadratic terminal ingredients, enables offset-free reference tracking and asymptotic IO stability. A numeric benchmark example is used to illustrate the advantages of the proposed method, as well as its real-time capabilities.

Keywords: Model Predictive Control, Linear Parameter Varying Systems, Input-Output Representation, LMI and Convex Optimisation, Stability and Performance.

1. INTRODUCTION

Model Predictive Control (MPC) is a very well established control method, based on a receding-horizon optimisation problem (Camacho and Bordons, 2013). MPC schemes using state-space (SS) process models have had considerable research focus over the last years, with a wide variety of results (Limon et al., 2018). Moreover, exploiting the capacity of Linear Parameter Varying (LPV) SS models to describe nonlinear and time-varying dynamics, LPV MPC schemes have been developed for a wide range of applications, e.g. (Cisneros and Werner, 2017; Morato et al., 2021); refer also to the survey (Morato et al., 2020) and references therein.

Yet with great theoretical value, the standard SS MPC design requires the availability of state measurements in real time. In turn, state variables are often difficult to measure or estimate with precision. Moreover, state estimation schemes can significantly deteriorate closed-loop performances of MPC in the presence of disturbances and constraints. Due to these issues, SS realisations fall short of industrial expectations, which are seldom anchored in input-output (IO) process descriptions. We also highlight that powerful LPV IO identification tools nowadays exist for a great variety of applications and system classes (Bachnas et al., 2014; den Boef et al., 2021).

For the previous reasons, we focus henceforth on IO MPC schemes for LPV systems. Obtaining reliable SS descriptions of an LPV IO model is numerically very

tough¹, since the dynamic-dependency problem hinders such translation (Tóth et al., 2011a). Therefore, it is desirable to design MPC schemes directly using IO LPV models, without the need for any IO-SS conversion. Up to the Authors' best knowledge, there are only a few papers which propose MPC schemes for LPV systems described in the IO form: (Abbas et al., 2015, 2016, 2018). Although these works provide closed-loop stability guarantees, they assume that the future scheduling behaviour is known, which is false for the vast majority of applications, or resort to the worst-case solution, robustifying the MPC by considering the scheduling variables as bounded uncertainties, which often leads to excessive conservativeness.

Motivated by the fact that recent works have proposed (numerically-cheap) recursive linear schemes that are able to extrapolate the future scheduling trajectories with accurateness, e.g. (Morato et al., 2019, 2021), we formulate herein a novel robust MPC scheme for IO LPV systems using such parameter extrapolation laws. With their inclusion, we are able to enhance the previous IO LPV MPC schemes from (Abbas et al., 2015, 2016, 2018), while benefiting from their stability frameworks. Specifically, the contributions of this paper are as follows:

- Using an explicit integral action scheme, we propose an IO LPV MPC formulation (Sec. 2), which enables offset-free output tracking of piece-wise constant reference trajectories. The scheme includes the future scheduling parameter estimation scheme from (Morato et al., 2021, 2022), which uses a Taylor argument to conceive a recursive extrapolation law.

^{*} This work has been supported by *CNPq* (304032/2019 – 0), *CAPES* (001) and *ITEA3* European project (15016) *EMPHYSIS*.

¹ This issue is exploited in details in recent key papers on the topic, e.g. (Tóth et al., 2011a,b).

- Benefiting from the stability framework from Mayne et al. (2000) as expanded to the LPV IO setting in (Abbas et al., 2016), we generate terminal ingredients that ensure a recursively feasible optimisation, and an asymptotically stable closed-loop (Sec. 3). These ingredients are enabled through a sufficient Linear Matrix Inequality (LMI) constraint.
- A numeric benchmark example is used to demonstrate the effectiveness of the proposed method, as well as its advantages and capabilities of the algorithm for real-time nonlinear applications (Sec. 4).

Notation. The index set $\mathbb{N}_{[a,b]}$ represents $\{i \in \mathbb{N} \mid a \leq i \leq b\}$, with $0 \leq a \leq b$. The identity matrix of size j is denoted as I_j ; $I_{j,\{i\}}$ denotes the i -th row of I_j ; \overrightarrow{I}_j^k denotes I_j with k columns shifted to the right. $\mathbf{1}_n$ denotes the stacked vector of n unitary entries. $\text{col}\{\cdot\}$ denotes the vectorisation of the entries and $\text{diag}\{v\}$ denotes the diagonal matrix generated with the line vector v . The predicted value of a given variable $v(k)$ at time instant $k+i$, computed based on the information available at instant k , is denoted as $v(k+i|k)$. \mathcal{K} refers to the class of positive and strictly increasing scalar functions that pass through the origin. A \mathcal{C}^1 function $f: \mathbb{R}^m \rightarrow \mathbb{R}^n$ is such that it is differentiable with continuous derivatives. $\|\cdot\|$ denotes the 2-norm. A' denotes the transpose of A , while (\star) denotes corresponding symmetrical transposes in LMIs.

2. SETUP

2.1 Preliminaries: IO LPV Model

We consider the following discrete-time multi-input multi-output (MIMO) LPV system given in an IO form:

$$\left(I_{n_y} + \sum_{i=1}^{n_a} a_i(\rho(k))z^{-i} \right) y(k) = \sum_{j=1}^{n_b} b_j(\rho(k))z^{-j}u(k) \quad (1)$$

Moreover, we include an implicit integral action, taking the vector of control inputs as follows: $u(k) := \delta(k) + u(k-1) \in \mathbb{R}^{n_u}$, where $\delta(k)$ stands for the corresponding control increments. In (1), $y(k) \in \mathbb{R}^{n_y}$ is the process outputs vector, z^{-1} is the one-sample backward shift operator, $n_a, n_b \geq 0$, while $a_i \in \mathbb{R}^{n_y \times n_u}$ and $b_i \in \mathbb{R}^{n_y \times n_u}$ are coefficient functions.

This LPV system is scheduled by the vector of time-varying parameters $\rho(k) \in \mathbb{R}^{n_\rho}$. Specifically, we assume that these parameters are output-dependent, bounded and measured online, this is: $\rho(k) = f_\rho(y(k-1)) \in \mathcal{P} \subseteq \mathbb{R}^{n_\rho}$, being \mathcal{P} a known set. Moreover, we consider that the nonlinear scheduling map $f_\rho(y)$ is algebraic, class \mathcal{C}^1 for all y ; $\mathcal{P} := \{\rho \in \mathbb{R}^{n_\rho} : \underline{\rho}_j \leq \rho_j \leq \bar{\rho}_j, j \in \mathbb{N}_{[1, n_\rho]}\}$ defines a compact, convex scheduling set. By definition, the future scheduling parameters $\rho(k+j)$, $\forall j \in \mathbb{N}_{[1, \infty]}$ are unknown at time instant k .

Remark 1. Nonlinear systems can be embedded to a LPV realisations such as Eq. (1), as long as a differential inclusion property is satisfied. This topic is further explained in (Morato et al., 2020), with experimental examples given in (Hoffmann and Werner, 2014).

2.2 Impulse Response

The IO realisation from Eq. (1) has an equivalent infinite impulse response form, which is given by:

$$y(k) = \sum_{i=0}^{+\infty} h_i(\rho(k), \dots, \rho(k-i))u(k-i), \quad (2)$$

where $h_i(\cdot) \in \mathbb{R}^{n_y \times n_u}$ are known as the Markov coefficients of the LPV systems. These coefficients can be computed recursively as follows, where $h_i(k) := h_i(\rho(k), \dots, \rho(k-i))$:

$$h_i(k) = \begin{cases} b_i(\rho(k)) - \sum_{j=1}^{\min(i, n_a)} a_j(\rho(k))h_{i-j}(k-j), & i \leq n_b, \\ - \sum_{j=1}^{\min(i, n_a)} a_j(\rho(k))h_{i-j}(k-j), & \text{else.} \end{cases}$$

2.3 SS Representation for Analyses

Now, we give a non-minimal, non-unique SS representation of Eq. (1), taking $\delta(k)$ as input. This SS model, used for the stability analyses presented in Sec. 4, is as follows:

$$\begin{aligned} x(k+1) &= A(\rho(k))x(k) + B(\rho(k))\delta(k), \\ y(k) &= C(\rho(k))x(k) + D(\rho(k))\delta(k), \end{aligned} \quad (3)$$

where $x(k) = [y(k-1)', \dots, y(k-n_a)', u(k-1)', \dots, u(k-n_b)'] \in \mathbb{R}^{n_x}$ defines the state vector with $n_x = n_a n_y + n_b n_u$. Note that in this SS representation, all states are known variables (current and past values of the inputs and outputs). The corresponding matrices $\begin{bmatrix} A(\cdot) & B(\cdot) \\ C(\cdot) & D(\cdot) \end{bmatrix}$ are:

$$\left[\begin{array}{cccc|cc} -a_1(\cdot) & \dots & -a_{n_a}(\cdot) & (b_0(\cdot) + b_1(\cdot)) & \dots & b_{n_b}(\cdot) & b_0 \\ I_{n_y} & \dots & 0 & 0 & \dots & 0 & 0 \\ & & \vdots & & & \vdots & \vdots \\ & & 0 & I_{n_u} & \dots & 0 & I_{n_u} \\ & & 0 & I_{n_u} & \dots & 0 & 0 \\ & & \vdots & & & \vdots & \vdots \\ \hline -a_1(\cdot) & \dots & -a_{n_a}(\cdot) & (b_0(\cdot) + b_1(\cdot)) & \dots & b_{n_b}(\cdot) & b_0(\cdot) \end{array} \right].$$

2.4 Full-horizon IO Predictor

The IO LPV model from Eq. (1) is used by an MPC algorithm and, thus, a full-horizon prediction equation is required to compute control law at each sample. Consider that $P(k|k) = \text{col}\{\rho(k+j|k)'\}'$, $\forall j \in \mathbb{N}_{[0, N_p-1]}$ is known (we will discuss how this vector can be generated later on), being N_p the prediction horizon. Thus, it follows that:

$$Y(k|k) = H(P(k|k))\Delta(k|k) + \Theta(P(k|k))x(k), \quad (4)$$

where $Y(k) = \text{col}\{y(k+j|k)'\}'$, $\forall j \in \mathbb{N}_{[0, N_p-1]} \in \mathbb{R}^{n_y N_p}$ and $\Delta(k) = \text{col}\{\delta(k+j|k)'\}'$, $\forall j \in \mathbb{N}_{[0, N_p-1]} \in \mathbb{R}^{n_u N_p}$. In MPC theory $H(P(k|k))\Delta(k)$ is usually named the “forced response” of the system, giving the dynamics implied by the control effort, where $\Theta(P(k|k))x(k)$ is called the “free response”, which gives the behaviour in the absence of future control inputs. The prediction matrices are:

$$\begin{aligned} H(P(k|k)) &= \begin{bmatrix} h_0(k) & \dots & 0 \\ \vdots & \ddots & \vdots \\ \sum_{j=0}^{N_p-1} h_i(k+N_p-1) & \dots & h_0(k+N_p-1) \end{bmatrix}, \\ \Theta(P(k|k)) &= [\tilde{\theta}(k+1)' \dots \tilde{\theta}(k+N_p)']'. \end{aligned}$$

The elements of $\Theta(P(k|k))$ are found using $\theta(k+j) = -\sum_{i=1}^{\min(j, n_a)} a_i(\rho(k+j|k))\theta(k+j-1) + \overrightarrow{I}^j \check{\theta}(k+j)$, $\forall j \in \mathbb{N}_{[1, N_p]}$, with $\check{\theta}(k+j) = [-a_1(\rho(k+j|k)) \dots - a_{n_a}(\rho(k+j|k))b_1(\rho(k+j|k)) \dots b_{n_b}(\rho(k+j|k))]'$ and $\overrightarrow{I}^j = \text{diag}\{\overrightarrow{I}_{n_a}^j, \overrightarrow{I}_{n_b}^j\}$. Finally, each $\tilde{\theta}(k+j)$ is given by $\theta(k+j)$ with its (n_a+1) -th element added to a correction bias $\sum_{i=1}^j h_i(k+j)$.

2.5 Process Constraints

We consider that the LPV system in Eq. (1) is subject to hard compact polyhedral constraints on outputs and inputs, which define an admissible operation. Specifically, we use $y(k) \in \mathcal{Y}$ and $u(k) \in \mathcal{U}$, $\forall k \geq 0$, where:

$$\begin{aligned} \mathcal{Y} &:= \{y \in \mathbb{R}^{n_y} : |y_j| \leq \bar{y}_j, \forall j \in \mathbb{N}_{[1, n_y]}\}, \\ \mathcal{U} &:= \{u \in \mathbb{R}^{n_u} : |u_j| \leq \bar{u}_j, \forall j \in \mathbb{N}_{[1, n_u]}\}. \end{aligned} \quad (5)$$

Due to the explicit integral description of the control input, the following constraint is also used: $\delta(k) \in \mathcal{D} := \{\delta \in \mathbb{R}^{n_u} : |\delta_j| \leq \bar{\delta}_j, \forall j \in \mathbb{N}_{[1, n_u]}\} \forall k \geq 0$.

Through the sequel, we assume that the scheduling proxy is locally Lipschitz whenever these constraints are satisfied:

Assumption 1. The nonlinear scheduling parameter map $f_\rho : \mathcal{Y} \rightarrow \mathcal{P}$ agrees to a local Lipschitz condition around any arbitrary point $y \in \mathcal{Y}$, i.e. $\|f_\rho(y) - f_\rho(\hat{y})\|_\infty \leq \gamma_\rho \|y - \hat{y}\|_\infty$, $\forall y, \hat{y} \in \mathcal{Y}$, where the smallest constant γ_ρ is known as the Lipschitz constant for $f_\rho(\cdot)$.

2.6 Scheduling Parameter Extrapolation

The concept of MPC is based on spanning a prediction of the process variables along a future horizon window. For such, we use the IO LPV prediction Eq. (4). As previously stated, the future scheduling sequence $P(k|k)$ is required.

For this, we benefit from the framework from (Morato et al., 2022), which is based on a first-order Taylor expansion of the scheduling proxy $f_\rho(y(k-1))$. The main advantage of this extrapolation procedure is that it guarantees a convergent guess with a small, bounded residual error, while only resorting to linear operators.

Denote $\delta y(k) = (y(k) - y(k-1))$ as the incremental output deviation. By definition, δy is bounded to a compact and convex box-type set $\delta \mathcal{Y} := \{\delta y \in \mathbb{R}^{n_y} : |\delta y_j| \leq \bar{\delta y}_j, \forall j \in \mathbb{N}_{[1, n_y]}\}$.

The method is as follows: consider that the static scheduling map $f_\rho(y)$ can be approximated, at the first derivative order, by $f_\rho(y) = f_\rho(y)|_{\check{y}} + \frac{\partial f_\rho(y)}{\partial y} \Big|_{\check{y}} (y - \check{y}) + \xi_\rho$, being \check{y} the expansion point and ξ_ρ a residual which inherits the discrepancy between the real static map and its approximate. Since $f_\rho(y)$ is assumed class \mathcal{C}^1 , it is direct that the partial derivatives $f_\rho^\partial(k) := \frac{\partial f_\rho(y)}{\partial y} \Big|_{y(k)}$ are ultimately bounded for all $y(k) \in \mathcal{Y}$. From this development, we obtain:

$$\begin{aligned} \rho(k+1) &= \rho(k) + f_\rho^\partial(k)\delta y(k) + \xi_\rho(k), \\ &\vdots \\ \rho(k+N_p-1) &= \rho(k+N_p-2) \\ &\quad + f_\rho^\partial(k+N_p-2)\delta y(k+N_p-2) \\ &\quad + \xi_\rho(k+N_p-2). \end{aligned}$$

Note that $\rho(k)$ and $\delta y(k)$ are known variables at each instant k , whereas $f_\rho^\partial(k)$ can be numerically evaluated. In practice, $f_\rho^\partial(k+j)$ is unknown for $j \in \mathbb{N}_{[1, N-2]}$, but we can assume that it shows an approximately uniform variation rate, i.e. $f_\rho^\partial(k+j) = f_\rho^\partial$, $\forall j \in \mathbb{N}_{[1, N-2]}$, where f_ρ^∂ denotes the partial derivative evaluated at instant k (refer to the discussion in (Morato et al., 2022)). By doing so, it is implied that $\rho(k+j) \approx \rho(k+j-1) + f_\rho^\partial \delta y(k+j-1|k)$.

Therefore, the estimate for the future scheduling variables can be written as the sum of the estimate from the previous sample corrected with an adjustment term $f_\rho^\partial \delta y(k+j-1)$. Accordingly, we can write the vector-wise extrapolation in a recursive fashion:

$$P(k|k) = P(k-1|k-1) + f_\rho^\partial \delta Y(k|k), \quad (6)$$

where the sequence of output increments is given by $\delta Y(k) = \text{col}\{\delta y(k+j|k)\}'$, $\forall j \in \mathbb{N}_{[0, N_p-2]}$.

Lemma 1. Assume that $f_\rho(\cdot)$ is class \mathcal{C}^1 and that f_ρ^∂ is ultimately bounded. Assume that (1) is stable in closed-loop. Then, the recursive extrapolation algorithm in Eq. (6) converges.

Proof 1. This proof is reduced for brevity. Complementary discussions are available in (Morato et al., 2022). Consider that the residual term $\xi_\rho(k+j)$ should turn null. Thus, use $\lim_{k \rightarrow \infty} y(k) = y_e$ holds (stability) and take $\xi_\rho(k) = f_\rho(y(k-1)) - f_\rho(y(k-2)) - f_\rho^\partial \delta y(k)$. Due to the stabilisation, it directly follows that $\lim_{k \rightarrow \infty} f_\rho(y(k-1)) = \lim_{k \rightarrow \infty} f_\rho(y_e)$ and $\lim_{k \rightarrow \infty} \delta y(k) = 0$, which implies in $\lim_{k \rightarrow \infty} \xi_\rho(k) = -\lim_{k \rightarrow \infty} f_\rho^\partial \delta y(k) \rightarrow 0$. This concludes the proof. \square

Lemma 2. The estimation error (residual) is ultimately bounded, i.e. $\|\xi_\rho(k)\|_\infty \leq (\gamma_\rho + \overline{f_\rho^\partial}) \bar{\delta y}$, $\forall k \geq 0$.

Proof 2. The residual term in the extrapolation law is given by $\xi_\rho(k+j+1) = f_\rho(y(k+j+1)) - f_\rho(y(k+j)) - f_\rho^\partial \delta y(k+j)$. Using a triangular inequality, we obtain $\|\xi_\rho(k+j+1)\|_\infty \leq \|f_\rho(y(k+j+1)) - f_\rho(y(k+j))\|_\infty + \|f_\rho^\partial \delta y(k+j)\|_\infty$. Finally, due to Assumption 1, we state that $\|\xi_\rho(k+j+1)\|_\infty \leq \gamma_\rho \|\delta y(k+j)\|_\infty + \|f_\rho^\partial \delta y(k+j)\|_\infty$. Since f_ρ^∂ is ultimately bounded, it follows that: $\|\xi_\rho\|_\infty \leq (\gamma_\rho + \overline{f_\rho^\partial}) \bar{\delta y}$. This concludes the proof. \square

Remark 2. The extrapolation procedure presented in this section is completely detailed in (Morato et al., 2022). The Hessian-based estimates from (Hanema et al., 2021) and the iteratively refined estimated from (Cisneros and Werner, 2017) are comparable methods to the one considered herein. Nevertheless, the main advantage of the proposed scheme is that converging estimates with reduced error bounds, formulated by the means of linear laws. Thereof, the corresponding model-process uncertainties

that are derived when applying MPC became very reduced, enabling less conservative control synthesis.

2.7 The MPC Design

As previously stated, we consider an MPC design for LPV processes with IO descriptions. Accordingly, we use the recursive extrapolation procedure in Eq. (6) to generate the scheduling sequence $P(k|k)$ that is used to compute the future output predictions $Y(k)$ through Eq. (4). In this paper, we assume that the residual errors ξ_ρ are null. The MPC terminal ingredients are presented later, in Sec. 3.

Consider the following cost function:

$$J = \sum_{i=0}^{N_p-1} \ell(e(k+i), v(k+i)) + V(x(k+N_p) - x_r),$$

where the stage cost $\ell(e, v) := \|e\|_Q + \|v\|_R$ is given with respect to the output tracking error $e(k) = r(k) - y(k)$, under the assumption that $r(k)$ is piece-wise constant s.t. $r(k+j) = r(k), \forall j \in \mathbb{N}_{[0, N_p-1]}$. The terminal cost $V(x - x_r)$ requires x , which is the state description given through Eq. (3), and x_r , the envisioned state reference target, defined in terms of $r(k)$; this cost is used to penalise the distance of non-minimal states at the end of the prediction horizon to a given target, whereas $\ell(\cdot, \cdot)$ weights the performance along the horizon. Consider Q and R as positive definite weighting matrices, used to imply the envisioned trade-off between control effort and output reference tracking.

Taking J into account, the proposed MPC resides in solving the following optimisation, at each instant k :

$$\begin{aligned} \min_{\Delta(k)} & \sum_{i=0}^{N_p-1} \ell(e(k+i), v(k+i)) + V(x(k+N_p) - x_r), \\ \text{s.t. : } & Y(k|k) = H(P(k|k))\Delta(k) + \Theta(P(k|k))x(k), \quad (7) \\ & y(k+j|k) \in \mathcal{Y}, \\ & u(k+j|k) \in \mathcal{U}, \\ & \delta(k+j|k) \in \mathcal{D}, \\ & (x(k+N_p) - x_r) \in \mathbf{X}_f, \end{aligned}$$

where \mathbf{X}_f is a terminal invariant set for the controlled IO LPV system. Let $J^*(x(0), r(k), P(k|k))$ be the optimal solution of the optimisation in Eq. (7), from which $\Delta(k)^*$ is the optimal sequence of control inputs. Then, the MPC law at time instant k considers in apply the first entry of $\Delta(k)^*$, i.e. $\delta^*(k|k)$, to the process using $u(k) = u(k-1) + \delta^*(k)$.

Definition 1. Positive Invariant Set

Assume that there exists a terminal set \mathbf{X}_f . \mathbf{X}_f is a positively invariant set for stability-related SS LPV model from Eq. (3) iff, for any $x \in \mathbf{X}_f$ and $\rho \in \mathcal{P}$, it follows that $x^+ \in \mathbf{X}_f$, where the successor state x^+ is given by $A(\rho)x + B(\rho)\delta$.

3. CLOSED-LOOP STABILITY AND RECURSIVE FEASIBILITY

Next, we discuss how to generate the MPC terminal ingredients $V(\cdot)$ and \mathbf{X}_f in order to render an asymptotically

stable closed-loop, as well as a recursively feasible optimisation.

First, let us define the admissible steady-state targets for the IO LPV system in Eq. (1). Consider r as a constant output reference target. We say that $r \in \mathcal{R}$ is an admissible target if and only if there exists a control input $u_r \in \mathcal{U}$ such that $(I_{n_y} + \sum_{i=1}^{n_a} a_i(f_\rho(r)))r = (\sum_{i=1}^{n_b} b_i(f_\rho(r)))u_r$. Accordingly, we introduce the set of all admissible terminal state-input targets (x_r, u_r) , that is: $\mathcal{X}_r := \{x_r \in \mathbb{R}^{n_x} \mid x_r = [1_{n_a}r, 1_{n_b}u_r], \forall (r, u_r) \in \mathcal{R} \times \mathcal{U}\}$. Due to the box-type constraints over y and u , \mathcal{X}_r can be equivalently described as $\{x \in \mathbb{R}^{n_x} \mid |x_j| \leq \bar{x}_j, \forall j \in \mathbb{N}_{[1, n_x]}\}$.

Assumption 2. The reference trajectory $R(k) = \text{col}\{r(k+j)\}, \forall j \in \mathbb{N}_{[0, N_p-1]}$ is known, piece-wise constant, and admissible at each sampling instant k , i.e. $r(k) \in \mathcal{R}$.

Assumption 3. The scheduling parameters take a constant value ρ_r in steady-state, i.e. $\rho = \rho_r, \forall (x - x_r) \in \mathbf{X}_f$.

Assumption 4. The stage cost function is positive definite and uniformly continuous such that $\ell(e, \delta) \geq \alpha_\ell(\|e\|)$ and $|\ell(e_1, \delta_1) - \ell(e_2, \delta_2)| \leq \lambda_e(\|e_1 - e_2\|) + \lambda_\delta(\|\delta_1 - \delta_2\|)$, where α_ℓ, λ_e and λ_δ are \mathcal{K} -functions. It is implied that $\ell(0, 0) = 0$.

Assumption 5. (1) There exists an admissible terminal feedback law $u(k-1) + \kappa_t(x(k), r(k), P(k|k)) \in \mathcal{U}$.

(2) The terminal set \mathbf{X}_f is closed, contains the origin, and represents admissible positive invariant set.

(3) The terminal cost $V(x - x_r)$ is continuous and positive for all $x - x_r \in \mathcal{X}_r$. Moreover $V(\cdot)$ represents a control Lyapunov function for the unconstrained LPV system in Eq. (1), meaning that there exist constants $b > 0$ and $\sigma > 1$ such that $V(x - x_r) \leq b|x - x_r|^\sigma$. This implies that $V(A(f_\rho(r))(x - x_r) + B(\rho)\kappa_t(\cdot)) - V(x - x_r) \leq \ell(y - r, \kappa_t(\cdot))$, for all $r \in \mathcal{R}$ and $V(x_1 - x_r) - V(x_2 - x_r) \leq \alpha_r(\|x_1 - x_2\|)$ (i.e. $V(\cdot)$ is a \mathcal{K} function).

The following Theorems ensure that the error dynamics $e(k) = r(k) - y(k)$ converge to the origin, which conversely means that offset-free reference tracking is ensured.

Theorem 1. Stability and Recursive Feasibility

Consider that Assumptions 2-5 hold. Suppose that there exists a terminal control law $\delta = \kappa_t(x, r, P)$. Consider that the IO LPV system in Eq. (1) is controlled by the MPC, as rendered through Eq. (7). Then, asymptotic output stability is ensured if the following conditions hold $\forall \rho \in \mathcal{P}$:

(C1) The origin $x(k) - x_r(k) = 0$ lies in the interior of \mathbf{X}_f ;

(C2) \mathbf{X}_f is positively invariant under the terminal feedback controller $\kappa_t(\cdot)$;

(C3) The discrete Lyapunov equation is verified within this invariant set, this is, $\forall x - x_r \in \mathbf{X}_f$ and $\forall \rho \in \mathcal{P}$: $V(x^+ - x_r) - V(x - x_r) \leq -\ell(e, \kappa_t(\cdot))$, where $x^+ = A(\rho)x + B(\rho)\kappa_t(\cdot)$;

(C4) The image of the terminal control is admissible, i.e. $\kappa_t(\cdot) + u \in \mathcal{U}, \forall \rho \in \mathcal{P}$;

(C5) The terminal set \mathbf{X}_f is a subset of \mathcal{X}_r .

Assuming that the initial solution of the MPC problem $\Delta^*(k)$ is feasible, then, the MPC is recursively feasible, steering $e(k) = r(k) - y(k)$ to the origin.

Proof 3. This proof is standard and follows (Mayne et al., 2000; Morato et al., 2020).

In order to satisfy the conditions required by Theorem 1, we choose the following quadratic terminal cost function $V(x - x_r) = (x - x_r)'T(x - x_r)$, where $T = T'$ is a positive definite weight. Accordingly, the terminal set \mathbf{X}_f is taken as a sub-level set of the terminal cost, i.e.: $\mathbf{X}_f := \{x \in \mathbb{R}^{n_x} \mid x'Tx \leq 1\}$.

By definition, \mathbf{X}_f is an ellipsoidal set constraint, which should be positively invariant for the terminal feedback $\kappa_t(\cdot)$. Thus, the following Theorem gives a numerically solvable sufficient solution that can be used to generate the terminal ingredients.

Theorem 2. Terminal Ingredients

Consider that Assumptions 2-5 hold. Then, conditions (C1)-(C5) from Theorem 1 are satisfied if there exist a symmetric positive definite matrix $T \in \mathbb{R}^{n_x \times n_x}$ and a rectangular matrix $W \in \mathbb{R}^{n_u \times n_x}$ such that $Y = T^{-1} > 0$, $W = KY$ and that LMIs (8)-(10) hold under the minimisation of $\log \det\{Y\}$ for all $\rho \in \mathcal{P}$. The terminal feedback is then given by $\kappa_t(\cdot) = K(x - x_r)$.

$$\begin{bmatrix} Y & \star & \star \\ (A(\rho)Y + B(\rho)W) & Y & \star \\ Y & 0 & \tilde{Q}^{-1} \\ W & 0 & R^{-1} \end{bmatrix} \geq 0, \quad (8)$$

$$\left[\frac{\bar{x}_j^2}{I_{\{j\}}Y'} \middle| \frac{I_{\{j\}}Y}{Y} \right] \geq 0, j \in \mathbb{N}_{[1, n_x]} \quad (9)$$

$$\left[\frac{\bar{\delta}_i^2}{\star} \middle| \frac{I_{\{i\}}W}{Y} \right] \geq 0, i \in \mathbb{N}_{[1, n_u]} \quad (10)$$

Proof 4. Consider $r(k)$ is piece-wise constant, thus satisfying Assumption 2. Note that $\ell(e, v) = \|e\|_Q + \|v\|_R$ is equivalent to $\|x - x_r\|_{\tilde{Q}} + \|v\|_R$, using $\tilde{Q} = \text{diag}\{QI_{n_y n_a}, 0_{n_u n_b}\}$. Consider $P(k|k)$ is known due to the extrapolation procedure from Eq. (6). Since $\rho(k) = f_\rho(y(k-1))$, it follows that $\rho(k) = \rho_r$ in steady-state, iff $\lim_{k \rightarrow +\infty} y(k-1) = r$. Then, by applying a Schur complement to LMI (8), we obtain condition (C3), which suffices for (C2). By definition, an ellipsoid ensures (C1). (C4) and (C5) are respectively satisfied by applying Schur complements to LMI (9) and (10). In turn, the terminal feedback $\kappa_t(\cdot) = K(x - x_r)$ ensures that the SS representation in Eq. (3) is asymptotically stable, which conversely implies in the output tracking of $r(k)$. This concludes the proof. \square

Remark 3. The terminal ingredients provided through Theorem 2 ensure recursive feasibility and asymptotic stability of the tracking error trajectories (refer to Propositions 1-2). Note that LMI (8) is infinite-dimensional, having to hold $\forall \rho \in \mathcal{P}$. In practice, the solution can be found by enforcing the inequalities over a sufficiently dense grid of points (ρ) along \mathcal{P} , then verifying it for a denser grid.

Remark 4. An alternative formulation to the prior can be used in order to drop the parameter-dependency of LMI (8): the full-block \mathcal{S} -procedure over a corresponding Linear Fractional Transformation (LFT) of the LPV system (Scherer, 2001), leading to similar LMIs to those presented in (Abbas et al., 2018).

Proposition 1. (Recursive Feasibility). Consider Y as a solution to Theorem 2. Then, given any $x \in \mathbf{X}_f$, $x_r \in \mathcal{X}_r$, $r \in \mathcal{R}$ and $\delta = \kappa_t(x, r, \cdot)$, we have $x^+ = Ax + BKx \in \mathbf{X}_f$. Con-

sider an optimal sequence $\Delta^* = (\delta_0^*, \delta_1^*, \dots, \delta_{N_p-1}^*)$ and an reference target r . Then $\hat{\delta}^c = (\delta_1^*, \dots, \delta_{N_p-1}^*, \kappa_t(x, r, \cdot))$ define feasible (candidate) solution of the MPC problem in Eq. (7) for any $r \in \mathcal{R}$, which means that the optimisation is recursively feasible.

Proof 5. Let Assumptions 2-5 hold. Consider there exists a solution Y to Theorem 2. Then, from conditions (C1), (C2), (C4), and (C5) from Theorem 1, we can infer that the generated control signal provides recursively feasible solutions to the MPC optimisation in (7). Take the feedback² $\kappa_t(\cdot) = K(x - x_r)$ and $\rho_r = f_\rho(r)$. Then, the if the initial condition $x(0)$ generates a feasible input sequence Δ^* , all future iterations of the optimisation will also be feasible: the generated control law control is admissible (condition (C4)) and all state variables $x \in \mathbf{X}_f$ generate successor state variables x^+ which are also inside \mathbf{X}_f (condition (C2)), which contains the origin (condition (C1), terminal condition for $(x - x_r)$, and \mathbf{X}_f being sub-set of \mathcal{X}_r (condition (C5), which ensures that x, x^+ are admissible). This concludes the proof. \square

Proposition 2. (Asymptotic Stability). Consider Y as a solution to Theorem 2. Then, the LPV system (1) in closed loop with the MPC input $\kappa_t(\cdot)$ exhibits an asymptotic stable tracking error dynamics. That is, for any feasible initial condition x_0 and constant set-point $r \in \mathcal{R}$, it is implied that $\|x(k) - x_r\| \leq \beta(\|x(0)\|, k)$, where β is a \mathcal{K} -function which passes through the origin.

Proof 6. Let there be a terminal stage cost $V(\cdot)$ such that Assumption 4 holds. Let Assumption 5 also hold and Proposition 1 be verified. Note that since $\ell(\cdot, \cdot)$ is a quadratic stage cost, α_ℓ , λ_e and λ_u indeed exists. Consider there exists a solution Y to Theorem 2. Then, the SS closed-loop is stable due to (C3) of Theorem 1, which conversely ensures that $V(x(k) - x_r) - V(x(k-1) - x_r) \leq -\|y(k) - r\|_Q$. Thus, we obtain $\|y(k) - r\|_Q \leq \beta(\|x(0) - x_r\|, k)$. Since $Q > 0$ (positive definite), we have $\|y(k) - r(k)\|_Q \geq \|x(k) - x_r\|$ and thus asymptotic stability is established. This concludes the proof. \square

4. NUMERIC BENCHMARK

In this Section, we demonstrate the effectiveness of the proposed method for controlling LPV system in the IO form, without any state measure or estimation procedure involved. For such, we consider an adapted version of the unstable second order system from (Abbas et al., 2015), with additional nonlinearities, given as:

² Note that this not a traditional state-feedback, since x relates to past input and output measurements only (refer to Eq. (3)).

$$\begin{aligned}
y(k) &= \overbrace{\left(0.2 - 0.7 \frac{y(k-1)^2}{10}\right)}^{-a_1(\rho(k))} y(k-1) \\
&+ \overbrace{\left(-0.7 - 0.4 \frac{y(k-1)^2}{10}\right)}^{-a_2(\rho(k))} y(k-2) \\
&+ \overbrace{\left(3.4 - 1.2 \frac{y(k-1)^2}{10}\right)}^{b_1(\rho(k))} u(k-1) \\
&+ \overbrace{\left(1.6 - 2.8 \frac{y(k-1)^2}{10}\right)}^{b_2(\rho(k))} u(k-2),
\end{aligned} \tag{11}$$

where the scheduling parameter is endogenous and given by $\rho(k) = \frac{y(k-1)^2}{10}$. This system should be controlled such that the output trajectory $y(k)$ tracks a given piecewise reference signal, whilst the following constraints are respected: $u(k) \in \mathcal{U} := [-1, 1]$, $\delta(k) \in \mathcal{D} := [-0.04, 0.04]$, $y(k) \in \mathcal{Y} := [0, 5]$, $\forall k \geq 0$. We note that these constraints imply that $\rho(k) \in \mathcal{P} := [0, 2.5]$, $\forall k \geq 0$. This system is assumed to operate under a sampling rate of 40 Hz. The proposed MPC is tuned with a prediction horizon of $N_p = 10$ discrete-time steps, thus previewing a window of 250 ms.

First of all, we show that there indeed exists a solution to Theorem 2, given by:

$$T^{-1} = Y = \begin{bmatrix} 15.9262 & 1.2843 & 1.6841 & 2.5095 \\ \star & 14.9439 & 2.3116 & 1.4652 \\ \star & \star & 1 & 0.4505 \\ \star & \star & \star & 1 \end{bmatrix}, \\
W = [0.1042 \ 0.0991 \ 0.0299 \ 0.0299].$$

Therefore, we can affirm that $\kappa_t(x - x_r) = K(x - x_r)$, with $K = WT$, $V(x - x_r) = (x - x_r)'T(x - x_r)$, and $\mathbf{X}_f := \{x - x_r \in \mathbb{R}^{n_x} \mid (x - x_r)'T(x - x_r) \leq 1\}$ are suitable terminal ingredients for the MPC optimisation in Eq. (7), ensuring recursive feasibility and asymptotic stability (and, in turn, offset-free reference tracking). We stress that these ingredients are generated offline, with respect to the tuning parameters $Q = 0.8I_{n_y}$ and $R = 0.2I_{n_u}$ and the known sets \mathcal{U} , \mathcal{D} , \mathcal{Y} , and \mathcal{P} .

Taking into account that stability is ensured by design (and, thus, Lemma 1 holds), we now demonstrate how the scheduling parameter extrapolation procedure operates. During the implementation, the recursive estimates are generated through the linear operator given in Eq. (6). As shows Fig. 1, the generated estimates $P(k|k)$ are very accurate, thus passing to the MPC precise information regarding the future behaviour of ρ . In this Fig., we can also see that the estimation error ξ_ρ is indeed small (refer to Lemma 2) and converges rapidly, which means that it can be neglected for simplicity (treating this uncertainty robustly directly in the MPC synthesis step is a topic for future works).

Next, we show the obtained performances with the proposed method. Fig. 2 shows the system trajectories being steered to the terminal set \mathbf{X}_f , as ensured by the MPC. The left-side sub-figure gives a three-dimensional cut of \mathbf{X}_f , while the right-side shows a two-dimensional projec-

tion. The system trajectories shown are the tracking error $e(k) = r(k) - y(k)$ and the control input $u(k)$. As one can see, the tracking error dynamics are repeatedly steered to the origin (for each new reference goal). Fig. 3 gives the control input trajectories (and the control increment), altogether with the piecewise constant reference target signal $r(k)$ and the output. Evidently, the integral-embedded MPC ensures offset-free tracking, which is significant.

The MPC requires, in average, 6.5 ms to solve the optimisation procedure, while the Taylor-based scheduling extrapolation takes only 0.07 ms, in average. This clearly indicates the relieved numeric burden of the proposed technique (6.57 ms, in average), which is ready for real-time embedded applications. We recall, once again, that no state measures or observers are necessary, making the proposed method coherent with industrial practices.

5. CONCLUSIONS

In this paper, a novel MPC algorithm for IO LPV systems is proposed. The future LPV scheduling parameters are extrapolated using a recursive Taylor expansion law, which generates the MPC prediction matrices at each sampling period. Terminal ingredients are offered through an LMI-solvable remedy, which ensures Lyapunov properties of the closed-loop. The method is able to ensure asymptotic offset-free reference tracking, also thanks to an explicit integral action and to these optimisation ingredients. In order to demonstrate the effectiveness of the method, it is to a numeric benchmark system, exhibiting good performances.

The proposed IO MPC method exhibits real-time capabilities, since its online implementation only requires a linear operation (scheduling parameter extrapolation) and the solution of a single quadratic optimisation problem. Also, the method is directly applicable to nonlinear systems, as long as if LPV embedding is satisfied. Moreover, the method does not require any additional reference tracking tool (such as the artificial reference variables).

For future works, we plan on generalising the proposed framework for data-driven predictive control synthesis, replacing the IO LPV description of the system by a data-based IO realisation.

REFERENCES

- Abbas, H.S., Hanema, J., Tóth, R., Mohammadpour, J., and Meskin, N. (2018). A new approach to robust MPC design for LPV systems in input-output form. *IFAC-PapersOnLine*, 51(26), 112–117.
- Abbas, H.S., Tóth, R., Meskin, N., Mohammadpour, J., and Hanema, J. (2015). An MPC approach for LPV systems in input-output form. In *2015 54th IEEE Conference on Decision and Control (CDC)*, 91–96. IEEE.
- Abbas, H.S., Toth, R., Meskin, N., Mohammadpour, J., and Hanema, J. (2016). A robust MPC for input-output LPV models. *IEEE Transactions on Automatic Control*, 61(12), 4183–4188.
- Bachnas, A., Tóth, R., Ludlage, J., and Mesbah, A. (2014). A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation

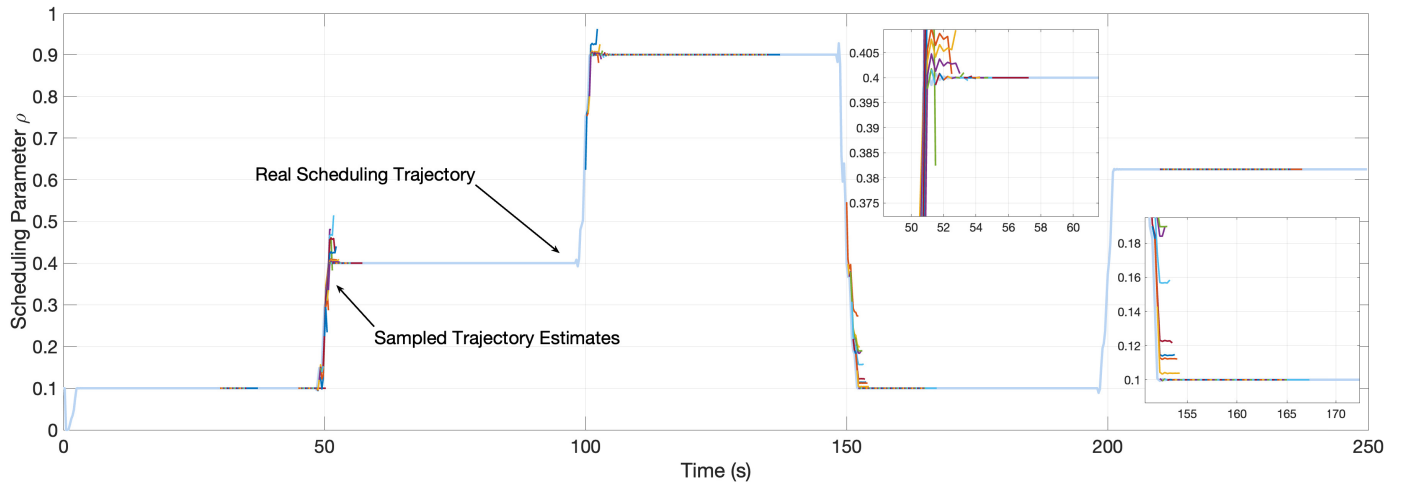


Fig. 1. Scheduling trajectory and Taylor-based extrapolation estimated (at different samples).

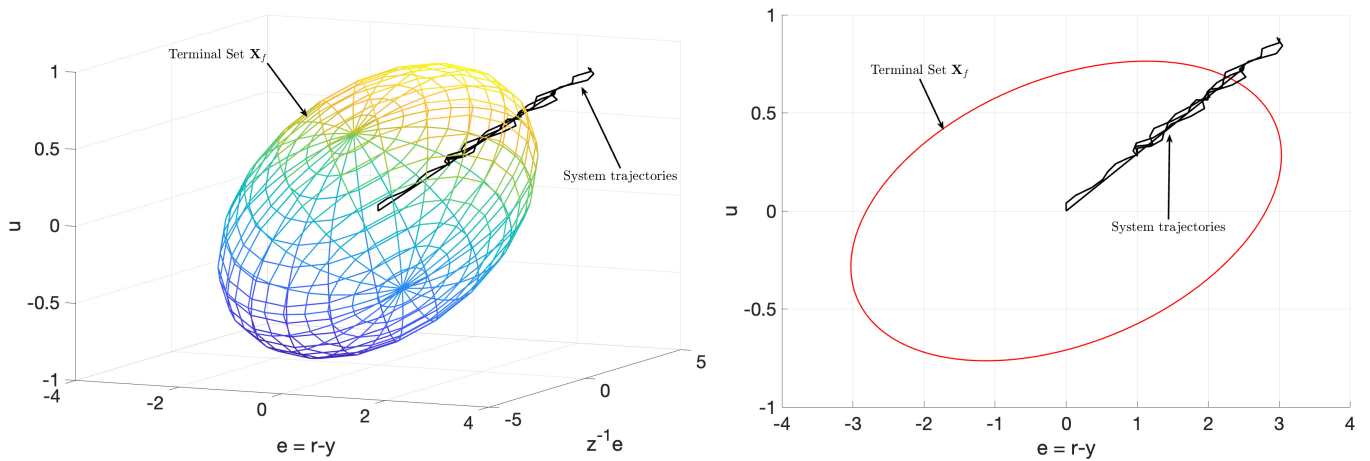


Fig. 2. Stability and terminal invariance. Terminal set X_f in two- (right) and three-dimensional (left) projections along with system trajectories $u(k), e(k) = r(k) - y(k)$.

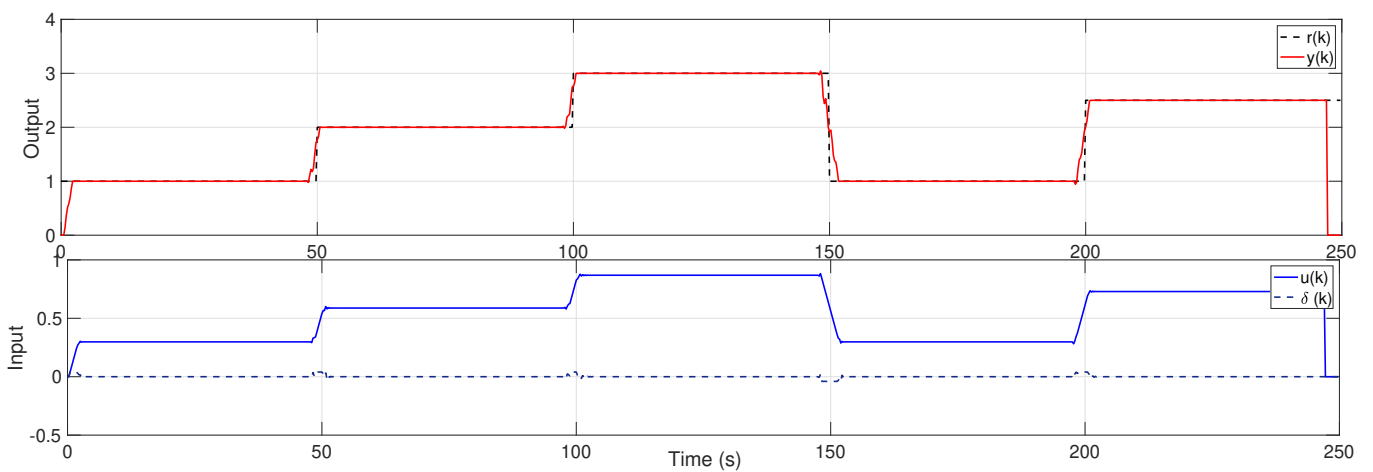


Fig. 3. System performances.

- column case study. *Journal of Process Control*, 24(4), 272–285.
- Camacho, E.F. and Bordons, C. (2013). *Model predictive control*. Springer Science & Business Media.
- Cisneros, P.G. and Werner, H. (2017). Fast nonlinear MPC for reference tracking subject to nonlinear constraints via quasi-LPV representations. *IFAC-PapersOnLine*, 50(1), 11601–11606.
- den Boef, P., Cox, P.B., and Tóth, R. (2021). LPVcore: MATLAB toolbox for LPV modelling, identification and control. *IFAC-PapersOnLine*, 54(7), 385–390.
- Hanema, J., Tóth, R., and Lazar, M. (2021). Stabilizing non-linear model predictive control using linear parameter-varying embeddings and tubes. *IET Control Theory & Applications*, 15(10), 1404–1421.
- Hoffmann, C. and 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(2), 416–433.
- Limon, D., Ferramosca, A., Alvarado, I., and Alamo, T. (2018). Nonlinear MPC for tracking piece-wise constant reference signals. *IEEE Transactions on Automatic Control*, 63(11), 3735–3750.
- Mayne, D.Q., Rawlings, J.B., Rao, C.V., and Sokaert, P.O. (2000). Constrained model predictive control: Stability and optimality. *Automatica*, 36(6), 789–814.
- Morato, M.M., Normey-Rico, J.E., and Sename, O. (2020). Model predictive control design for linear parameter varying systems: A survey. *Annual Reviews in Control*, 49, 64–80.
- Morato, M.M., Normey-Rico, J.E., and Sename, O. (2022). Sufficient conditions for convergent recursive extrapolation of qLPV scheduling parameters along a prediction horizon. URL <https://hal.archives-ouvertes.fr/hal-03536927>. Pre-print provisionally accepted at the IEEE Transactions on Automatic Control.
- Morato, M.M., Normey-Rico, J.E., and Sename, O. (2019). Novel qLPV MPC design with least-squares scheduling prediction. *IFAC-PapersOnLine*, 52(28), 158–163.
- Morato, M.M., Normey-Rico, J.E., and Sename, O. (2021). A fast dissipative robust nonlinear model predictive control procedure via quasi-linear parameter varying embedding and parameter extrapolation. *International Journal of Robust and Nonlinear Control*, 31(18), 9619–9651.
- Scherer, C.W. (2001). LPV control and full block multipliers. *Automatica*, 37(3), 361–375.
- Tóth, R., Abbas, H.S., and Werner, H. (2011a). On the state-space realization of LPV input-output models: Practical approaches. *IEEE transactions on control systems technology*, 20(1), 139–153.
- Tóth, R., Willems, J.C., Heuberger, P.S., and Van den Hof, P.M. (2011b). The behavioral approach to linear parameter-varying systems. *IEEE Transactions on Automatic Control*, 56(11), 2499–2514.