Data enabled Predictive Control of LPV systems

Taleb Bou Hamdan, Patrick Coirault, Guillaume Mercère, Thibault Dairay

Control Engineering Practice 149 (2024) 105969

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Data enabled Predictive Control of LPV systems

Taleb Bou Hamdan [a,b], Patrick Coirault [a,*], Guillaume Mercère [a], Thibault Dairay [b,c]

[a] Laboratoire d'Automatique et d'Informatique pour les Systèmes, Poitiers University, Poitiers, France
[b] Manufacture Française des Pneumatiques Michelin, Clermont-Ferrand, France
[c] Centre Borelli, CNRS, Université Paris Saclay, ENS Paris Saclay, Gif-sur-Yvette, France

ARTICLE INFO

Keywords:
Data enabled predictive control
Linear Parameter Varying Systems
Willem's lemma

ABSTRACT

This paper introduces a data-enabled predictive control methodology designed for Linear Parameter Varying (LPV) systems. By leveraging a polytopic representation of the LPV system, we formulate Willem's lemma to accommodate parameter variations. The system trajectory is predicted over a finite horizon using specific trajectories generated offline. A notable advantage of this approach is its independence from *a priori* knowledge of parametric variations. Simulation studies conducted on both a numerical example and a simulator of a calendering process governed by partial differential equations substantiate the effectiveness of this approach.

1. Introduction

The effectiveness of Model Predictive Control (MPC) in industrial system control is well-established, attributed to its favorable balance between performance and ease of tuning. Notably, MPC adeptly handles Multi-Input Multi-Output (MIMO) systems, addressing both equality and inequality constraints on control signal and output. Despite its known computational complexity, the diminishing costs of computational resources, especially on-board computers, have further bolstered the adoption of MPC. A significant challenge in deploying MPC lies in obtaining a representative model. This model enables MPC to predict system outputs over a finite horizon based on control variables. Typically, this model is derived through an identification step reliant on input/output measurements from a properly excited system (Van Overschee & De Moor, 1994, 1996). The execution of this identification step demands proficiency in signal processing, statistics and optimization, making its accurate implementation potentially complex. Consequently, the ability to represent system dynamics without the prerequisite of the model identification step holds substantial value in this context.

Addressing this demand, recent years have witnessed the emergence of Data Driven methods. Utilizing Willem's fundamental lemma, these methods offer a solution to the predictive control problem by relying solely on input/output sequences (Willems, 2007; Willems, Rapisard, Markovsky, & De Moor, 2005), with extensions found in Dinkla, Mulders, van Wingerden, and Oomen (2023), Markovsky and Dörfler (2021) and Van Waarde, De Persis, Camlibel, and Tesi (2020). The key concept revolves around the notion that a trajectory (an input/output sequence) from a Linear Time-Invariant (LTI) system, obtained through sufficiently rich excitation, inherently contains all relevant information on the system dynamics *a priori*. This trajectory can effectively serve as a parametrization tool for any other trajectory of the system. The extensive literature on this approach, known as Data Driven Predictive Control or Data Enabled Predictive Control (DeePC), has predominantly focused on discrete-time LTI systems, particularly exploring its robustness in the presence of noise (Breschi, Chiuso, & Formentin, 2023a; Markovsky & Dörfler, 2021). This article principal contribution lies in extending this approach to Linear Parameter Varying (LPV) systems.

Linear Parameter Varying (LPV) systems are linear dynamic systems with a state–space representation dependent on nonstationary exogenous parameters (Morato, Normey-Rico, & Sename, 2020; Toth, 2010). Conceptually, an LPV system can be visualized as a collection of Linear Time-Invariant (LTI) systems interconnected by these exogenous parameters, resulting in nonstationary dynamics. This framework enables the modeling of system nonlinearities through a series of LTI models (Gidon et al., 2021; Kapsalis, Sename, Milanes, & Molina, 2022; Li, Nguyen, Guerra, & Kruszewski, 2023). Leveraging this approach, control methods originally designed for linear systems can be applied to manage nonlinear counterparts effectively. Given the close proximity of LPV modeling to LTI modeling, it is logical to explore the conditions under which the DeePC approach, initially developed for LTI systems, can be extended to LPV systems. Recent studies have delved into the application of Willem's fundamental Lemma to LPV systems (Verhoek, Abbas, Toth, & Haesaert, 2021). However, a notable limitation arises in these investigations as the exogenous parameters must be known and measurable at every moment, posing a significant drawback to LPV modeling.

* Correspondence to: Bâtiment B25, 2 rue Pierre Brousse, TSA 41105, 86073 Poitiers cedex 9, France.
E-mail address: patrick.coirault@univ-poitiers.fr (P. Coirault).

https://doi.org/10.1016/j.conengprac.2024.105969
Received 19 December 2023; Received in revised form 5 April 2024; Accepted 6 May 2024
Available online 17 May 2024

The main contribution of this paper is the introduction of an extension of DeePC to systems represented by an LPV model, obviating the requirement to measure exogenous parameters. We consider in this paper the noise free case only. In achieving this objective, output prediction over a finite horizon is conducted based on a set of specific trajectories acquired offline. Key distinctions from the LTI case include the number of trajectories needed to capture system dynamics and the methodology used for generating these trajectories. Although the optimization problem remains quadratic, its size expands due to the increased number of decision variables corresponding to the higher number of trajectories.

The subsequent sections of this paper are organized as follows: Section 2 provides an overview of notations and the model structure. Section 3 initially revisits Data Enabled Predictive Control for Linear Time-Invariant systems and then introduces a novel methodology for applying DeePC to Linear Parameter Varying systems without prior knowledge of the varying parameter trajectories. Section 4 evaluates the applicability and efficiency of the introduced methodology through two numerical examples, including an LPV system with two varying parameters and a calendering process. The paper concludes with a summary of remarks in Section 5.

2. Preliminaries

This section presents the notations used in the paper, the model structure and useful definitions.

2.1. Notations and definitions

Let $\mathbb{N}$ and $\mathbb{R}$ be the sets of positive integers and real numbers, respectively. $\mathbb{N}^*$ denotes the set of positive non-zero integers. Let $\mathcal{I}_{k,s} = \{k, k+1, \dots, k+s-1\}$. The set of real column vectors of dimension $n \in \mathbb{N}^*$ is denoted by $\mathbb{R}^n$ and the set of real matrices of $n \in \mathbb{N}^*$ rows and $m \in \mathbb{N}^*$ columns is denoted by $\mathbb{R}^{n\times m}$. Given a rectangular matrix $\boldsymbol{A} \in \mathbb{R}^{n\times m}$, its transpose is denoted by $\boldsymbol{A}^\top \in \mathbb{R}^{m\times n}$, $\boldsymbol{A}^{(i)} \in \mathbb{R}^{1\times m}$ represents its ith row. For a vector $\boldsymbol{x}(k) \in \mathbb{R}^{n_x}$, $\Delta\boldsymbol{x}(k) = \boldsymbol{x}(k) - \boldsymbol{x}(k-1)$. For any vector $\boldsymbol{x}(k) \in \mathbb{R}^{n_x}$, with $k \in \mathbb{N}$, the finite vector over a specific window of size ρ steps ($\rho \in \mathbb{N}^*$) starting from a specified instant $k \in \mathbb{N}$ is denoted as

$$\boldsymbol{X}_{k,\rho,1} = \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{x}(k+1) \\ \vdots \\ \boldsymbol{x}(k+\rho-1) \end{bmatrix} \in \mathbb{R}^{n_x\rho}. \tag{1}$$

Accordingly, the block Hankel matrix containing the available data starting from instant $k \in \mathbb{N}$ distributed over $\rho \in \mathbb{N}^*$ rows and $M \in \mathbb{N}^*$ columns is denoted as

$$\boldsymbol{X}_{k,\rho,M} = \begin{bmatrix} \boldsymbol{X}_{k,\rho,1} & \boldsymbol{X}_{k+1,\rho,1} & \cdots & \boldsymbol{X}_{k+M-1,\rho,1} \end{bmatrix} \in \mathbb{R}^{n_x\rho\times M}. \tag{2}$$

Definition 1. An input sequence $\{\boldsymbol{u}(k)\}_{k=0}^{N-1}$, $\boldsymbol{u}(k) \in \mathbb{R}^{n_u}$, is persistently exciting of order ρ if the matrix $\boldsymbol{U}_{0,\rho,N}$ is of rank $n_u\rho \le N \in \mathbb{N}$ (see Verhaegen and Verdult (2007, chapter 10)).

The norm of the vector $\|\boldsymbol{X}_{k,\rho,1}\|_{\boldsymbol{Q}}^2$ denotes the quadratic form $\boldsymbol{X}_{k,\rho,1}^\top \boldsymbol{Q} \boldsymbol{X}_{k,\rho,1}$ where $\boldsymbol{Q} \in \mathbb{R}^{n_x\rho\times n_x\rho}$ is a symmetric and strictly positive matrix. The following matrices are defined by

$$\boldsymbol{S}_{\ell,n} = \begin{bmatrix} \mathbb{I}_{n\times n} & \mathbb{O}_{n\times n} & \cdots & \cdots \\ \mathbb{I}_{n\times n} & \mathbb{I}_{n\times n} & \mathbb{O}_{n\times n} & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{I}_{n\times n} & \mathbb{I}_{n\times n} & \mathbb{I}_{n\times n} & \mathbb{I}_{n\times n} \end{bmatrix} \in \mathbb{R}^{\ell n\times \ell n}, \tag{3}$$

$$\mathbb{1}_{\ell,n} = \begin{bmatrix} \mathbb{I}_{n\times n} \\ \vdots \\ \mathbb{I}_{n\times n} \end{bmatrix} \in \mathbb{R}^{\ell n\times n}. \tag{4}$$

2.2. Model structure

This section introduces a behavioral framework for LPV systems (see Toth (2010) for more details).

Definition 2. A discrete-time parameter varying dynamical system $\boldsymbol{S}$ is defined as a quadruple

$$\boldsymbol{S} = (\mathbb{T}, \mathbb{P}, \mathbb{W}, \mathcal{B}) \tag{5}$$

with $\mathbb{T} = \mathbb{Z}$ the discrete time axis, $\mathbb{P} \subseteq \mathbb{R}^{n_p}$ the scheduling space, $\mathbb{W} = \mathbb{R}^{n_W}$ the signal space and $\mathcal{B} \subseteq (\mathbb{W}\times\mathbb{P})^{\mathbb{T}}$ the behavior ($X^{\mathbb{T}}$ is the standard notation for the collection of all maps from $\mathbb{T}$ to X)

$\boldsymbol{p} \in \mathbb{P}^{\mathbb{T}}$ is the scheduling variable, an external free signal of the system that governs the dynamical behavior. In the sequel, we assume that $\boldsymbol{p}$ evolves in a polytope of dimension $N_p = 2^{n_p}$. Inside the polytope, the admissible trajectories of $\boldsymbol{p}$ are restricted to a subset of $\mathbb{P}^{\mathbb{T}}$, the set of admissible scheduling trajectories. This set, denoted by $\mathcal{B}_{\mathbb{P}}$, is described as follows

$$\mathcal{B}_{\mathbb{P}} = \left\{ \boldsymbol{p} \in \mathbb{P}^{\mathbb{T}} \,\middle|\, \exists\, \boldsymbol{w} \in \mathbb{W}^{\mathbb{T}} \text{ s.t. } (\boldsymbol{w}, \boldsymbol{p}) \in \mathcal{B} \right\}. \tag{6}$$

Additionally, for a given trajectory $\boldsymbol{p} \in \mathcal{B}_{\mathbb{P}}$, the projected behavior

$$\mathcal{B}_{\boldsymbol{p}} = \left\{ \boldsymbol{w} \in \mathbb{W}^{\mathbb{T}} \,|(\boldsymbol{w}, \boldsymbol{p}) \in \mathcal{B} \right\} \tag{7}$$

defines all the signal trajectories compatible with the fixed scheduling trajectory $\boldsymbol{p}$, In case of a constant scheduling trajectory $\boldsymbol{p} \in \mathcal{B}_{\boldsymbol{P}}$, $\boldsymbol{p}(k) = \boldsymbol{p}_f\ \forall k \in \mathbb{Z}$, the projection behavior $\mathcal{B}_{\boldsymbol{p}_f}$ is called a frozen behavior, i.e.,

$$\mathcal{B}_{\boldsymbol{p}_f} = \left\{ \boldsymbol{w} \in \mathbb{W}^{\mathbb{T}} \,\middle|\, (\boldsymbol{w}, \boldsymbol{p}) \in \mathcal{B} \text{ with } \boldsymbol{p}(k) = \boldsymbol{p}_f\ \forall k \in \mathbb{Z} \right\}. \tag{8}$$

In this case, $\mathcal{F}_{\boldsymbol{p}_f} = \left(\mathbb{T}, \mathbb{W}, \mathcal{B}_{\boldsymbol{p}_f}\right)$ defines a frozen system of $\boldsymbol{S}$ for each $\boldsymbol{p}_f \in \mathbb{P}$. $\boldsymbol{S}$ is called Linear Parameter Varying (LPV) if the following conditions are satisfied:

- $\mathbb{W}$ is a vector space and $\mathcal{B}_{\boldsymbol{p}}$ is linear subspace of $\mathbb{W}^{\mathbb{T}}\ \forall\ \boldsymbol{p} \in \mathcal{B}_{\mathbb{P}}$,
- $\mathbb{T}$ is closed under addition,
- for any $(\boldsymbol{w}, \boldsymbol{p}) \in \mathcal{B}$ and any $k \in \mathbb{Z}$, it holds that $(\boldsymbol{w}(.+k), \boldsymbol{p}(.+k)) \in \mathcal{B}$.

A discrete time state–space representation of an LPV system, denoted as $\boldsymbol{S} = (\boldsymbol{p}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$, is given by

$$\boldsymbol{x}(k+1) = \boldsymbol{A}(\boldsymbol{p}(k))\boldsymbol{x}(k) + \boldsymbol{B}(\boldsymbol{p}(k))\boldsymbol{u}(k), \tag{9a}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}(\boldsymbol{p}(k))\boldsymbol{x}(k) + \boldsymbol{D}(\boldsymbol{p}(k))\boldsymbol{u}(k), \tag{9b}$$

where $\boldsymbol{x}(k) \in \mathbb{R}^{n_x}$ is the state vector of the system at each instant $k \in \{0, \dots, n_t\}$ (n_t is the concerned time domain and k stands for the discrete time step), $\boldsymbol{u}(k) \in \mathbb{R}^{n_u}$ and $\boldsymbol{y}(k) \in \mathbb{R}^{n_y}$ are the input and output vectors of the system, respectively. $\boldsymbol{A} \in \mathbb{R}^{n_x\times n_x}$, $\boldsymbol{B} \in \mathbb{R}^{n_x\times n_u}$, $\boldsymbol{C} \in \mathbb{R}^{n_y\times n_x}$ and $\boldsymbol{D} \in \mathbb{R}^{n_y\times n_u}$ are the system, input, output and feedthrough matrices, respectively. $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ are matrix functions with static dependence on $\boldsymbol{p} \in \mathbb{R}^{n_p}$.

As $\boldsymbol{p}$ lies in a polytope, each component p_h of $\boldsymbol{p}$ is bounded, i.e.,

$$\underline{p}_h \le p_h \le \overline{p}_h \qquad \forall h \in \{1, n_p\}.$$

For LPV systems, unlike Linear Time Varying (LTV) systems, the variation of the system dynamics is not directly linked to time but is instead associated with the variation of the parameter vector $\boldsymbol{p}$.

The class of LTI systems can be derived from the LPV ones through the following approach.

Definition 3. Let $\boldsymbol{S} = (\mathbb{T}, \mathbb{P}, \mathbb{W}, \mathcal{B})$ be an LPV system. The set of LTI systems

$$\mathcal{F}_f = \left\{ \mathcal{F} = \left(\mathbb{T}, \mathbb{W}, \mathcal{B}'\right) \,\middle|\, \exists\ \boldsymbol{p}_f \in \mathbb{P} \text{ s.t. } \mathcal{B}' = \mathcal{B}_{\boldsymbol{P}_f} \right\}$$

is called the frozen set of $\boldsymbol{S}$.

It follows that the structure of an element of $\mathcal{F}_s$ is given by

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k), \tag{10a}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k). \tag{10b}$$

This LTI system is denoted by $\boldsymbol{S}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$. The following assumptions are considered for each element of $\mathcal{F}_s$, *i.e.* an LTI system $\boldsymbol{S}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$.

Assumption 1. The pair $(\boldsymbol{A}, \boldsymbol{B})$ is controllable.

Assumption 2. The pair $(\boldsymbol{A}, \boldsymbol{C})$ is observable.

3. Data enabled predictive control

This section first recalls the DeePC problem for deterministic LTI systems. It begins with a review of Willem's fundamental lemma and its application to predictive control. Subsequently, an extension to deterministic LPV systems is introduced. Finally, a systematic approach to embed an integral action into data driven predictive control is presented. These two last aspects are the primary contributions of this paper.

3.1. Trajectory based representation for LTI systems

To implement Predictive Control, it is necessary to be able to calculate the derivatives of a constraint cost function with respect to the optimization variables over a predictive horizon. This involves expressing the output and/or input as a function of these optimization variables (Toth, 2010). The classic approach is to use a model. This model can be deduced from a transfer function, a state–space realization or from a subspace approach (Favoreel, Moor, & Gevers, 1999). Obtaining this model requires an identification step which can be complex to implement. In contrast, data driven approaches do away with the model and the identification step, starting from the idea that all the information on the dynamics of the system is contained in the input–output data. This is the main idea of Willem's fundamental lemma which is reviewed in Lemma 1.

The notion of system trajectory plays a key role in this framework. An output–input sequence $\{\boldsymbol{y}(k), \boldsymbol{u}(k)\}_{k=0}^{N-1}$ for $N \in \mathbb{N}^*$ is a trajectory of the LTI system $\boldsymbol{S}$ if it satisfies Definition 4.

Definition 4. An output–input sequence $\{\boldsymbol{y}(k), \boldsymbol{u}(k)\}_{k=0}^{N-1}$ for $N \in \mathbb{N}^*$ (represented by $\{\boldsymbol{Y}_{0,N,1}, \boldsymbol{U}_{0,N,1}\}$) is a trajectory of the LTI system $\boldsymbol{S}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ if an initial condition $\boldsymbol{x}(0) \in \mathbb{R}^{n_x}$ and a state sequence $\{\boldsymbol{x}(k)\}_{k=0}^{N}$ exist such that the sequence $\{\boldsymbol{x}(k)\}_{k=0}^{N}$ satisfies the state space equations of the system given in Eq. (10).

Starting with the LTI system (10) and from Definition 4, Lemma 1 gives a trajectory based representation which is known as Willem's Lemma (Willems, 2007), or the Fundamental lemma as established in Berberich and Allgöwer (2020).

Lemma 1. *Consider an LTI system $\boldsymbol{S}(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$. Assume that the input signal used to build the Hankel matrix $\boldsymbol{U}_{0,s,N}$ is persistently exciting of order s with $s \in \mathbb{N}^*$. Then, for $N \in \mathbb{N}^*$, the system output–input trajectory at each instant $k \in \mathbb{N}$ over s steps $\begin{bmatrix}\boldsymbol{Y}_{k,s,1}^\top & \boldsymbol{U}_{k,s,1}^\top\end{bmatrix}^\top$ belongs to the column space of the matrix $\begin{bmatrix}\boldsymbol{Y}_{0,s,N}^\top & \boldsymbol{U}_{0,s,N}^\top\end{bmatrix}^\top$.*

In other words, there exists a vector $\boldsymbol{\alpha}(k) \in \mathbb{R}^N$ such that

$$\begin{bmatrix}\boldsymbol{Y}_{k,s,1}\\ \boldsymbol{U}_{k,s,1}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Y}_{0,s,N}\\ \boldsymbol{U}_{0,s,N}\end{bmatrix}\boldsymbol{\alpha}(k). \tag{11}$$

The key result of Lemma 1 is that any output–input trajectory of the LTI system $\boldsymbol{S}$ can be embedded into a linear combination of sufficiently excited past output–input trajectory.

3.2. Data driven predictive control for LTI systems

The main objective behind the predictive control approach is to find, at each time k, the optimal trajectory $\begin{bmatrix}\boldsymbol{Y}_{k,L,1}\\ \boldsymbol{U}_{k,L,1}\end{bmatrix}$ over the prediction horizon $L \in \mathbb{N}^*$ which minimizes the quadratic cost function

$$J(\boldsymbol{Y}_{k,L,1}, \boldsymbol{U}_{k,L,1}) = \|\boldsymbol{Y}_{k,L,1} - \boldsymbol{Y}_r\|_{\boldsymbol{Q}}^2 + \|\boldsymbol{U}_{k,L,1}\|_{\boldsymbol{R}}^2, \tag{12a}$$

$$\text{s.t. } \boldsymbol{u}(i) \in \mathcal{U}, \tag{12b}$$

$$\boldsymbol{y}(i) \in \mathcal{Y}, \tag{12c}$$

$\forall i \in \mathcal{I}_{k,L}$, where $\boldsymbol{Q} \in \mathbb{R}^{n_y L \times n_y L}$ and $\boldsymbol{R} \in \mathbb{R}^{n_u L \times n_u L}$ are user-defined output and input error penalizing positive definite matrices. They are tuned based on a trade-off between the degree of importance of each of the outputs and inputs terms (Rawlings, Mayne, & Diehl, 2017). $\mathcal{Y}$ and $\mathcal{U}$ are polytopes defining the applicable lower and upper boundaries of the system output and input respectively. $\boldsymbol{Y}_r \in \mathbb{R}^{n_y L}$ stands for the reference trajectory over the prediction horizon such that

$$\boldsymbol{Y}_r = \begin{bmatrix}\boldsymbol{y}_r(k)\\ \vdots\\ \boldsymbol{y}_r(k+L-1).\end{bmatrix} \tag{13}$$

Splitting up the trajectories between past and future trajectories as well as the Hankel matrices as shown in Fig. 1 and assuming that $\text{rank}\left(\boldsymbol{U}_{0,\rho+L,N}\right) = \rho + L$, a vector $\boldsymbol{\alpha}(k) \in \mathbb{R}^N$ exists such that

$$\begin{bmatrix}\boldsymbol{Y}_{k-\rho,\rho,1}\\ \boldsymbol{Y}_{k,L,1}\\ \boldsymbol{U}_{k-\rho,\rho,1}\\ \boldsymbol{U}_{k,L,1}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Y}_{0,\rho,N}\\ \boldsymbol{Y}_{\rho,L,N}\\ \boldsymbol{U}_{0,\rho,N}\\ \boldsymbol{U}_{\rho,L,N}\end{bmatrix}\boldsymbol{\alpha}(k). \tag{14}$$

Referring to Eq. (12), the decision variables are the future output and input vectors that can be written in terms of $\boldsymbol{\alpha}(k)$ based on Eq. (14) such that

$$\boldsymbol{Y}_{k,L,1} = \boldsymbol{Y}_{\rho,L,N}\boldsymbol{\alpha}(k), \tag{15a}$$

$$\boldsymbol{U}_{k,L,1} = \boldsymbol{U}_{\rho,L,N}\boldsymbol{\alpha}(k). \tag{15b}$$

Using the result of Eq. (15), the cost function (12) can be reformulated with $\boldsymbol{\alpha}(k)$ as the sole decision variable. The DeePC problem boils down to finding $\boldsymbol{\alpha}(k)$ such that

$$\arg\min_{\boldsymbol{\alpha}(k)} \left(\|\boldsymbol{\alpha}(k)\|_{\boldsymbol{W}}^2 + 2\boldsymbol{C}_p^\top\boldsymbol{\alpha}(k)\right), \tag{16}$$

subject to

$$\begin{bmatrix}\boldsymbol{Y}_{k-\rho,\rho,1}\\ \boldsymbol{U}_{k-\rho,\rho,1}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Y}_{0,\rho,N}\\ \boldsymbol{U}_{0,\rho,N}\end{bmatrix}\boldsymbol{\alpha}(k), \tag{17a}$$

$$\boldsymbol{Y}_{L,\rho,N}\boldsymbol{\alpha}(k) = \begin{bmatrix}\boldsymbol{y}_r(k+L-\rho)\\ \vdots\\ \boldsymbol{y}_r(k+L-1)\end{bmatrix}, \tag{17b}$$

$$\boldsymbol{U}_{\rho,L,N}^{(i)}\boldsymbol{\alpha}(k) \in \mathcal{U}, \tag{17c}$$

$$\boldsymbol{Y}_{\rho,L,N}^{(i)}\boldsymbol{\alpha}(k) \in \mathcal{Y}, \tag{17d}$$

where $i \in \mathcal{I}_{1,L}$. The terminal equality constraint is added to guarantee closed loop stability (Berberich, Köhler, Müller, & Allgöwer, 2021). $\boldsymbol{W}$ and $\boldsymbol{C}_p$ are given as

$$\boldsymbol{W} = \boldsymbol{Y}_{\rho,L,N}^\top\boldsymbol{Q}\boldsymbol{Y}_{\rho,L,N} + \boldsymbol{U}_{\rho,L,N}^\top\boldsymbol{R}\boldsymbol{U}_{\rho,L,N}, \tag{18a}$$

$$\boldsymbol{C}_p = -\boldsymbol{Y}_{\rho,L,N}^\top\boldsymbol{Q}\boldsymbol{Y}_r. \tag{18b}$$

The resulting predictive control algorithm is summed up in Algorithm 1.

Remark 1. The former results as well as Algorithm 1 are valid for noise free data only. If the available data are noisy, a regularization term

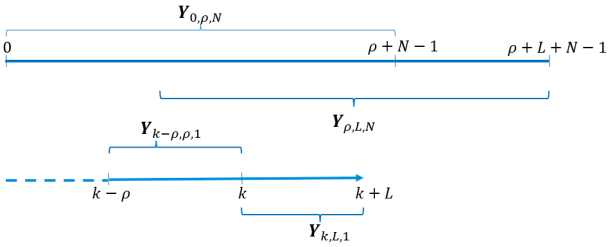

Fig. 1. Timeline showing the collected data for the Hankel matrices $\boldsymbol{Y}_{0,\rho,N}$ and $\boldsymbol{Y}_{\rho,L,N}$ (top) and the past and future vectors at each instant k, $\boldsymbol{Y}_{k-\rho,\rho,1}$ and $\boldsymbol{Y}_{k,L,1}$, respectively (bottom).

should be added to the optimization problem (16) in order to mitigate the noise effect. For instance, inspired by Verheijen, Breschi, and Lazar (2023), the following problem can be suggested

$$\arg\min_{\boldsymbol{\alpha}(k)} \left(\|\boldsymbol{\alpha}(k)\|^2_{\boldsymbol{W}} + 2\boldsymbol{C}_{\boldsymbol{p}}^{\top}\boldsymbol{\alpha}(k) + \lambda \boldsymbol{h}(\boldsymbol{\alpha}) \right), \tag{19}$$

where $\lambda \in \mathbb{R}^+$ and $\boldsymbol{h}(.)$ a nonnegative function. The regularization term is often chosen as a squared two-norm $\|\boldsymbol{\alpha}(k)\|_2^2$ but this choice modifies the initial problem and does not leads to the exact solution. To overcome this problem, the following alternative is introduced in Dörfler, Coulson, and Markovsky (2023)

$$\boldsymbol{h}(\boldsymbol{\alpha}) = \|(\mathbb{I} - \boldsymbol{\Pi})\boldsymbol{\alpha}(k)\|_2^2, \quad \text{where } \boldsymbol{\Pi} = \begin{bmatrix} \boldsymbol{U}_{0,\rho,N} \\ \boldsymbol{Y}_{0,\rho,N} \\ \boldsymbol{U}_{\rho,L,N} \end{bmatrix}^{\dagger} \begin{bmatrix} \boldsymbol{U}_{0,\rho,N} \\ \boldsymbol{Y}_{0,\rho,N} \\ \boldsymbol{U}_{\rho,L,N} \end{bmatrix}, \tag{20}$$

but this specific choice of $\boldsymbol{h}(.)$ increases the time load of the corresponding DeePC.

Algorithm 1 Data Enabled Predictive Control

Phase 1 – Before closing the control loop:

1: Choose $\boldsymbol{Q}$, $\boldsymbol{R}$, ρ, L, $\mathcal{Y}$, $\mathcal{U}$, and $\boldsymbol{Y}_r$ suitably
2: Collect the open loop data sets $\boldsymbol{Y}_{0,\rho+L,N}$, $\boldsymbol{U}_{0,\rho+L,N}$
3: Compute $\boldsymbol{W}$ and $\boldsymbol{C}_p$ from (18)

Phase 2 – During the control loop

4: **for** $k = 1, 2, \cdots$ **do**
5: Measure $\boldsymbol{y}(k)$
6: Minimize the optimization function in Eq. (16) for $\boldsymbol{\alpha}(k)$
7: Compute $\boldsymbol{U}_{k,L,1} = \boldsymbol{U}_{\rho,L,N}\boldsymbol{\alpha}(k)$
8: Apply the input $\boldsymbol{u}(k) = \boldsymbol{U}_{k,1,1}$ to the system
9: Update $\boldsymbol{C}_p$ if $\boldsymbol{Y}_r$ changes
10: Update the past vectors $\boldsymbol{U}_{k-\rho,\rho,1}$ and $\boldsymbol{Y}_{k-\rho,\rho,1}$
11: **end for**

3.3. Extension of data enabled predictive control for LPV systems

LPV systems are highly beneficial for modeling parametric uncertainties, nonlinearities, or variations in operating points (Toth, 2010). In state representation, the matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ are contingent upon varying parameters. For the analysis, and more particularly for robustness concerning parametric uncertainties, it is essential to be aware of the variation intervals of these parameters. This requirement holds even when defining a robust control law that encompasses the entire parameter variation domain. Nevertheless, this approach tends to exhibit a conservative nature when applied to the real-time adaptation of control in response to parametric variations. In analysis, particularly in the study of robustness concerning parametric uncertainties, it is imperative to have knowledge of the variation intervals of the parameters. This requirement holds true even when a robust control law is established to span the entire domain of parameter variation.

This section aims at addressing the challenge of real-time awareness of parameter variations while maintaining an adaptive control law aligned with the system dynamics at each moment. To achieve this, we present an extension of the Fundamental Lemma 1 tailored for LPV systems.

The considered model structure is noted as $\boldsymbol{S} = (\boldsymbol{p}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ and defined by Eq. (9). Let matrix $\boldsymbol{E}$ be defined by

$$\boldsymbol{E} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right].$$

Without loss of generality, it is assumed that $\boldsymbol{E}$ has affine dependence, *i.e.*,

$$\boldsymbol{E} = \boldsymbol{E}_0 + \sum_{h=1}^{n_p} p_h \boldsymbol{E}_h, \tag{21}$$

where $\boldsymbol{E}_0$ is the nominal term. Let us define

$$m_h = \frac{\underline{p}_h + \bar{p}_h}{2}, \quad w_h = \frac{\bar{p}_h - \underline{p}_h}{2}, \quad \forall h \in \{1, \dots, n_p\}.$$

Then Eq. (21) can be written as

$$\begin{aligned} \boldsymbol{E} &= \boldsymbol{E}_0 + \sum_{h=1}^{n_p} \left(\frac{p_h - m_h}{w_h} w_h \boldsymbol{E}_h \right) + \sum_{h=1}^{n_p} m_h \boldsymbol{E}_h, \\ &= \boldsymbol{Z}_0 + \boldsymbol{Z}, \end{aligned}$$

with

$$\boldsymbol{Z} = \sum_{h=1}^{n_p} \theta_h \boldsymbol{Z}_h, \qquad \theta_h = \frac{p_h - m_h}{w_h},$$

$$\boldsymbol{Z}_h = w_h \boldsymbol{E}_h, \qquad \boldsymbol{Z}_0 = \boldsymbol{E}_0 + \sum_{h=1}^{n_p} m_h \boldsymbol{E}_h.$$

Note that θ_h are normalized parameter variation such that

$$-1 \le \theta_h \le 1, \qquad \forall h \in \{1, \dots, n_p\}.$$

Consider the $\mathbb{R}^{n_p}$-vectors $\boldsymbol{\phi}_k$, $k \in \mathbb{N}^*$, with components equaling either 0 or 1, associated with the binary numbers, and then define

$$\boldsymbol{\psi}_k = \mathbb{1}_{n_p,1} - 2\boldsymbol{\phi}_k, \qquad \forall k \in \mathbb{N}^*.$$

Define $N_p = 2^{n_p}$ and note $\boldsymbol{\Psi} = \begin{bmatrix} \boldsymbol{\psi}_1 & \cdots & \boldsymbol{\psi}_{N_p} \end{bmatrix}$. Let $\boldsymbol{G}$ be defined by $\boldsymbol{G} = \begin{bmatrix} \boldsymbol{Z}_1^T & \cdots & \boldsymbol{Z}_{n_p}^T \end{bmatrix}^T$. When every p_h varies in its range, $\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \cdots & \theta_{n_p} \end{bmatrix}^T$ describes an hypercube. As a consequence, $\boldsymbol{G}$ describes an orthotope of matrices whose vertices, denoted by $\boldsymbol{M}_i$, $i = 1, \dots, N_p$, correspond to extreme values of various p_h and are given by

$$\boldsymbol{M}_i = \boldsymbol{\psi}_i^T \boldsymbol{G}, \quad \forall i \in \{1, N_p\}.$$

In other words, $\boldsymbol{Z}$ can be written as a convex combination of vertices $\boldsymbol{M}_i$, *i.e.*,

$$\boldsymbol{Z} = \sum_{i=1}^{N_p} \beta_i \boldsymbol{M}_i,$$

with

$$\beta_i \ge 0 \quad \forall i \in \{0, \dots, N_p\}, \quad \sum_{i=1}^{N_p} \beta_i = 1,$$

or equivalently,

$$\boldsymbol{E} = \sum_{i=1}^{N_p} \beta_i \left(\boldsymbol{E}_0 + \sum_{h=1}^{n_p} m_h \boldsymbol{E}_h + \boldsymbol{M}_i \right) \tag{22a}$$

$$= \sum_{i=1}^{N_p} \beta_i \boldsymbol{E}^{[i]}, \tag{22b}$$

where $\boldsymbol{E}$ describes an orthotope of matrices with its N_p vertices

$$\boldsymbol{E}^{[i]} = \left[\begin{array}{c|c} \boldsymbol{A}^{[i]} & \boldsymbol{B}^{[i]} \\ \hline \boldsymbol{C}^{[i]} & \boldsymbol{D}^{[i]} \end{array}\right].$$

Let us denote $\boldsymbol{p}^{[i]} \in \mathbb{R}^{n_p}$ as the fixed value representing the scheduling parameter $\boldsymbol{p}$ at the ith vertex. Each component of $\boldsymbol{p}^{[i]}$ corresponds to one of the N_p combinations of $\overline{p}_h$ and $\underline{p}_h$, where h ranges from 1 to N_p. $\boldsymbol{E}^{[i]}$ is derived by evaluating $\boldsymbol{E}$ at the ith vertex with $\boldsymbol{p} = \boldsymbol{p}^{[i]}$. Note that $\boldsymbol{E}$ is entirely described by a convex combination of its vertices. The coefficients β_i are the parameters of the convex combination and are function of $\boldsymbol{p}(k)$. Indeed, it can be shown that

$$\beta_i = \frac{1}{N_p}\left(\prod_{h=1}^{n_p}\left(\psi_{i_h}\theta_h + 1\right)\right) \quad \forall i \in \{1, \dots, N_p\}, \tag{23}$$

where ψ_{i_h} is the h^{ith} component of $\boldsymbol{\psi}_i$. To simplify the notation, this dependence is omitted. In the paper, it is denoted β_i instead of $\beta_i(\boldsymbol{p}(k))$.

In prior LPV control methods documented in the literature (refer, for instance, to Verhoek et al. (2021)), the trajectory of $\boldsymbol{p}$ was presumed to be known, or equivalently, the coefficients β_i were assumed to be known at each time. In contrast, this work considers the trajectory of the parameter vector $\boldsymbol{p}$ as unknown. The achievement of data driven predictive control for LPV systems with unknown parameter variations involves extending the fundamental lemma to encompass LPV systems, as shown by Proposition 1.

Proposition 1. *Consider the LPV system (9) where the parameter $\boldsymbol{p} \in \mathbb{R}^{n_p}$. Then, any output–input trajectory $\begin{bmatrix}\boldsymbol{Y}_{k,s,1}^\top & \boldsymbol{U}_{k,s,1}^\top\end{bmatrix}^\top$ of (9), for $k \in \mathbb{N}$, belongs to the column space of the matrix*

$$\boldsymbol{H}_{yu} = \begin{bmatrix} \boldsymbol{Y}_{0,s,N}^{[1]} & \dots & \boldsymbol{Y}_{0,s,N}^{[N_p]} \\ \boldsymbol{U}_{0,s,N}^{[1]} & \dots & \boldsymbol{U}_{0,s,N}^{[N_p]} \end{bmatrix} \in \mathbb{R}^{s(n_u+ny)\times N * N_p}, \tag{24}$$

where $\boldsymbol{Y}_{0,s,N}^{[i]}$ and $\boldsymbol{U}_{0,s,N}^{[i]}$ are the Hankel matrices of the frozen system $\boldsymbol{S}^{[i]} = (\boldsymbol{p}^{[i]}, \boldsymbol{E}^{[i]})$, assuming that $\operatorname{rank}(\boldsymbol{U}_{0,s,N}^{[i]}) = sn_u$ for $i = 1, \dots, N_p$.

Proof. Using Eqs. (22), (9) can be equivalently rewritten in the polytopic form

$$\boldsymbol{x}(k+1) = \sum_{i=1}^{N_p}\beta_i\boldsymbol{A}^{[i]}\boldsymbol{x}(k) + \sum_{i=1}^{N_p}\beta_i\boldsymbol{B}^{[i]}\boldsymbol{u}(k), \tag{25a}$$

$$\boldsymbol{y}(k) = \sum_{i=1}^{N_p}\beta_i\boldsymbol{C}^{[i]}\boldsymbol{x}(k) + \sum_{i=1}^{N_p}\beta_i\boldsymbol{D}^{[i]}\boldsymbol{u}(k). \tag{25b}$$

Let us define

$$\boldsymbol{x}^{[i]}(k) = \beta_i\boldsymbol{x}(k), \quad \boldsymbol{u}^{[i]}(k) = \beta_i\boldsymbol{u}(k), \quad \boldsymbol{y}^{[i]}(k) = \beta_i\boldsymbol{y}(k). \tag{26}$$

As $\sum_{i=1}^{N_p}\beta_i = 1$, it is straightforward to write

$$\boldsymbol{x}(k) = \sum_{i=1}^{N_p}\boldsymbol{x}^{[i]}(k), \qquad \boldsymbol{u}(k) = \sum_{i=1}^{N_p}\boldsymbol{u}^{[i]}(k), \qquad \boldsymbol{y}(k) = \sum_{i=1}^{N_p}\boldsymbol{y}^{[i]}(k). \tag{27}$$

The vectors $\boldsymbol{x}^{[i]}(k)$, $\boldsymbol{u}^{[i]}(k)$ and $\boldsymbol{y}^{[i]}(k)$ correspond to the state, input and output vectors of a frozen model at instant $k \in \mathbb{N}$ and corresponds to a vertex $\boldsymbol{S}^{[i]} = (\boldsymbol{A}^{[i]}, \boldsymbol{B}^{[i]}, \boldsymbol{C}^{[i]}, \boldsymbol{D}^{[i]})$ of the orthotope $\boldsymbol{S}$. This frozen model can be written in the state space form

$$\boldsymbol{x}^{[i]}(k+1) = \boldsymbol{A}^{[i]}\boldsymbol{x}^{[i]}(k) + \boldsymbol{B}^{[i]}\boldsymbol{u}^{[i]}(k), \tag{28a}$$

$$\boldsymbol{y}^{[i]}(k) = \boldsymbol{C}^{[i]}\boldsymbol{x}^{[i]}(k) + \boldsymbol{D}^{[i]}\boldsymbol{u}^{[i]}(k). \tag{28b}$$

Consider now an excitation signal persistently exciting of order $s \in \mathbb{N}^*$. Then, based on Definition 4, the output–input sequence $\{\boldsymbol{y}^{[i]}(k), \boldsymbol{u}^{[i]}(k)\}_{k=0}^{N-1}$ is a trajectory of the frozen model $\boldsymbol{S}^{[i]}$.

From Lemma 1, the system output–input sequence over s steps $\begin{bmatrix}\boldsymbol{Y}_{k,s,1}^{[i]\top} & \boldsymbol{U}_{k,s,1}^{[i]\top}\end{bmatrix}^\top$ belongs to the column space of the matrix $\begin{bmatrix}\boldsymbol{Y}_{0,s,N}^{[i]\top} & \boldsymbol{U}_{0,s,N}^{[i]\top}\end{bmatrix}^\top$. In other words, each trajectory of the frozen model $\boldsymbol{S}^{[i]}$ is a linear combination of the column space of the Hankel matrices $\boldsymbol{Y}_{0,s,N}^{[i]}$ and $\boldsymbol{U}_{0,s,N}^{[i]}$.

As a result of Lemma 1, for each trajectory and each instant k, over $s \in \mathbb{N}^*$ steps of $\boldsymbol{S}^{[i]}$ for $i = 1, \dots, 2^{n_p}$, there exists a vector $\boldsymbol{\alpha}^{[i]}(k) \in \mathbb{R}^N$ such that

$$\begin{bmatrix}\boldsymbol{Y}_{k,s,1}^{[i]} \\ \boldsymbol{U}_{k,s,1}^{[i]}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Y}_{0,s,N}^{[i]} \\ \boldsymbol{U}_{0,s,N}^{[i]}\end{bmatrix}\boldsymbol{\alpha}^{[i]}(k). \tag{29}$$

Substituting Eq. (29) into Eq. (27) leads to write the trajectory of the LPV system at instant $k \in \mathbb{N}$ over $s \in \mathbb{N}^*$ steps as

$$\begin{bmatrix}\boldsymbol{Y}_{k,s,1} \\ \boldsymbol{U}_{k,s,1}\end{bmatrix} = \sum_{i=1}^{N_p}\begin{bmatrix}\boldsymbol{Y}_{k,s,1}^{[i]} \\ \boldsymbol{U}_{k,s,1}^{[i]}\end{bmatrix} \tag{30a}$$

$$= \sum_{i=1}^{N_p}\begin{bmatrix}\boldsymbol{Y}_{0,s,N}^{[i]} \\ \boldsymbol{U}_{0,s,N}^{[i]}\end{bmatrix}\boldsymbol{\alpha}^{[i]}(k). \tag{30b}$$

Eq. (30) can be summed up in a general form as

$$\begin{bmatrix}\boldsymbol{Y}_{k,s,1} \\ \boldsymbol{U}_{k,s,1}\end{bmatrix} = \boldsymbol{H}_{yu}\boldsymbol{\alpha}(k), \tag{31}$$

where $\boldsymbol{\alpha}(k) \in \mathbb{R}^{N_p N}$. This proves that any trajectory of the LPV system $\boldsymbol{S}$ belongs to the column space of $\boldsymbol{H}_{yu}$. □

Remark 2. The coefficients β_i do not appear explicitly in Eq. (31). However, they are implicitly present in the vectors $\boldsymbol{\alpha}^{[i]}(k)$. As in the DeePC algorithm, $\boldsymbol{\alpha}(k)$ is recalculated at each time k. This means that at each time a new convex combination is calculated to position matrices $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ in the orthotope $\boldsymbol{S}$ without an explicit measurement of $\boldsymbol{p}(k)$.

Using this result, Proposition 2 can be established to implement an Data Enabled Predictive Control to the LPV systems with no prior knowledge of the parameter trajectory.

Proposition 2. *The Data Enabled Predictive Control problem for the LPV system (9) where the parameter vector $\boldsymbol{p}$ is unknown is given by*

$$\arg\min_{\boldsymbol{\alpha}(k)} \|\boldsymbol{\alpha}(k)\|_{\boldsymbol{W}}^2 + 2\boldsymbol{C}_p^\top\boldsymbol{\alpha}(k), \tag{32}$$

subject to

$$\begin{bmatrix}\boldsymbol{Y}_{k-\rho,\rho,1} \\ \boldsymbol{U}_{k-\rho,\rho,1}\end{bmatrix} = \begin{bmatrix}\boldsymbol{Y}_{0,\rho,N}^{[1]} & \dots & \boldsymbol{Y}_{0,\rho,N}^{[N_p]} \\ \boldsymbol{U}_{0,\rho,N}^{[1]} & \dots & \boldsymbol{U}_{0,\rho,N}^{[N_p]}\end{bmatrix}\boldsymbol{\alpha}(k), \tag{33a}$$

$$\begin{bmatrix}\boldsymbol{Y}_{L,\rho,N}^{[1]} & \dots & \boldsymbol{Y}_{L,\rho,N}^{[N_p]}\end{bmatrix}\boldsymbol{\alpha}(k) = \begin{bmatrix}\boldsymbol{y}_r(k+L-\rho) \\ \vdots \\ \boldsymbol{y}_r(k+L-1)\end{bmatrix}, \tag{33b}$$

$$\begin{bmatrix}\boldsymbol{U}_{\rho,L,N}^{[1]} & \dots & \boldsymbol{U}_{\rho,L,N}^{[N_p]}\end{bmatrix}^{(i)}\boldsymbol{\alpha}(k) \in \mathcal{U}, \tag{33c}$$

$$\begin{bmatrix}\boldsymbol{Y}_{\rho,L,N}^{[1]} & \dots & \boldsymbol{Y}_{\rho,L,N}^{[N_p]}\end{bmatrix}^{(i)}\boldsymbol{\alpha}(k) \in \mathcal{Y}, \tag{33d}$$

where $\mathcal{U}$ and $\mathcal{Y}$ are the polytopes governing the input and output trajectories respectively and $i \in \mathcal{I}_{1,L}$. $\boldsymbol{W}$ and $\boldsymbol{C}_p$ are now given as

$$\boldsymbol{W} = \left\|\begin{bmatrix}\boldsymbol{Y}_{\rho,L,N}^{[1]} & \dots & \boldsymbol{Y}_{\rho,L,N}^{[N_p]}\end{bmatrix}\right\|_{\boldsymbol{Q}}^2 + \left\|\begin{bmatrix}\boldsymbol{U}_{\rho,L,N}^{[1]} & \dots & \boldsymbol{U}_{\rho,L,N}^{[N_p]}\end{bmatrix}\right\|_{\boldsymbol{R}}^2, \tag{34a}$$

$$\boldsymbol{C}_p = -\begin{bmatrix}\boldsymbol{Y}_{\rho,L,N}^{[1]} & \dots & \boldsymbol{Y}_{\rho,L,N}^{[N_p]}\end{bmatrix}^\top\boldsymbol{Q}\boldsymbol{Y}_r, \tag{34b}$$

where $\boldsymbol{Q} \in \mathbb{R}^{n_y L}$ and $\boldsymbol{R} \in \mathbb{R}^{n_u L}$ are the positive definite diagonal penalizing matrices depending on the trade off between the input and the output of the LPV system.

Proof. The predictive control problem is given by minimizing Eq. (12) where the relation between the system output and input trajectory, $[\boldsymbol{Y}_{k,L,1}^\top, \boldsymbol{U}_{k,L,1}^\top]^\top$ is dependent on the system to be controlled. For an LPV system $\boldsymbol{S} = (\boldsymbol{p}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$, any output–input trajectory of the system can be written as a linear combination of the columns of $\boldsymbol{H}_{yu}$ as illustrated in Proposition 1. As a result, Eq. (31) can be reformulated

with $s = \rho + L$ corresponding to $\rho \in \mathbb{N}^*$ steps before the instant $k \in \mathbb{N}^*$ and $L \in \mathbb{N}^*$ steps ahead of the instant k such that

$$\begin{bmatrix} \boldsymbol{Y}_{k-\rho,\rho,1} \\ \boldsymbol{Y}_{k,L,1} \\ \boldsymbol{U}_{k-\rho,\rho,1} \\ \boldsymbol{U}_{k,L,1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}^{[1]}_{0,\rho,N} & \cdots & \boldsymbol{Y}^{[N_p]}_{0,\rho,N} \\ \boldsymbol{Y}^{[1]}_{\rho,L,N} & \cdots & \boldsymbol{Y}^{[N_p]}_{\rho,L,N} \\ \boldsymbol{U}^{[1]}_{0,\rho,N} & \cdots & \boldsymbol{U}^{[N_p]}_{0,\rho,N} \\ \boldsymbol{U}^{[1]}_{\rho,L,N} & \cdots & \boldsymbol{U}^{[N_p]}_{\rho,L,N} \end{bmatrix} \boldsymbol{\alpha}(k), \tag{35}$$

where $\boldsymbol{\alpha}(k) \in \mathbb{R}^{NN_p}$. This shows that the future output and input trajectories can be written as a function of $\boldsymbol{\alpha}(k)$ such that

$$\boldsymbol{Y}_{k,L,1} = \begin{bmatrix} \boldsymbol{Y}^{[1]}_{\rho,L,N} & \cdots & \boldsymbol{Y}^{[N_p]}_{\rho,L,N} \end{bmatrix} \boldsymbol{\alpha}(k), \tag{36a}$$

$$\boldsymbol{U}_{k,L,1} = \begin{bmatrix} \boldsymbol{U}^{[1]}_{\rho,L,N} & \cdots & \boldsymbol{U}^{[N_p]}_{\rho,L,N} \end{bmatrix} \boldsymbol{\alpha}(k). \tag{36b}$$

Based on Eq. (36), the minimization problem in Eq. (12) is reformulated as (32) subject to (33a). □

3.4. Integral action for DeePC

In practical applications, addressing deterministic disturbances necessitates the incorporation of an integrator for offset-free tracking. The optimal strategy for rejecting a deterministic disturbance involves integrating its model into the design model. Consequently, to eliminate a static error, an integral action must be integrated into the design model. Unfortunately, in DeePC, there is no predefined design model available *a priori*. To bypass this challenge, let us consider the innovation form of a state space representation of an LTI system

$$\boldsymbol{x}(k+1) = \boldsymbol{A}\boldsymbol{x}(k) + \boldsymbol{B}\boldsymbol{u}(k) + \boldsymbol{K}\boldsymbol{a}(k), \tag{37a}$$

$$\boldsymbol{y}(k) = \boldsymbol{C}\boldsymbol{x}(k) + \boldsymbol{D}\boldsymbol{u}(k) + \boldsymbol{a}(k), \tag{37b}$$

where $\boldsymbol{K}$ is the Kalman gain and $\boldsymbol{a}$ is a random walk, *i.e.*, $\boldsymbol{a}(k+1) = \boldsymbol{a}(k) + \boldsymbol{e}(k)$ with $\boldsymbol{e}$ a zero mean white noise. It follows that

$$\boldsymbol{\Delta x}(k+1) = \boldsymbol{A}\boldsymbol{\Delta x}(k) + \boldsymbol{B}\boldsymbol{\Delta u}(k) + \boldsymbol{K}\boldsymbol{e}(k), \tag{38a}$$

$$\boldsymbol{\Delta y}(k) = \boldsymbol{C}\boldsymbol{\Delta x}(k) + \boldsymbol{D}\boldsymbol{\Delta u}(k) + \boldsymbol{e}(k), \tag{38b}$$

where $\boldsymbol{\Delta}$ is the differentiation operator, *i.e.*, $\boldsymbol{\Delta z}(k+1) = \boldsymbol{z}(k+1) - \boldsymbol{z}(k)$. Let us consider an augmented state

$$\boldsymbol{x_a}(k) = \begin{bmatrix} \boldsymbol{\Delta x}(k) \\ \boldsymbol{y}(k) \end{bmatrix}. \tag{39}$$

The corresponding augmented model is given by

$$\boldsymbol{x_a}(k+1) = \boldsymbol{A_a}\boldsymbol{x_a}(k) + \boldsymbol{B_a}\boldsymbol{\Delta u}(k) + \boldsymbol{K_a}\boldsymbol{e}(k), \tag{40a}$$

$$\boldsymbol{y}(k) = \boldsymbol{C_a}\boldsymbol{x_a}(k), \tag{40b}$$

with

$$\boldsymbol{A_a} = \begin{bmatrix} \boldsymbol{A} & \mathbb{O}_{n_x,n_y} \\ \boldsymbol{CA} & \mathbb{I}_{n_y,n_y} \end{bmatrix}, \qquad \boldsymbol{B_a} = \begin{bmatrix} \boldsymbol{B} \\ \boldsymbol{CB} \end{bmatrix}, \tag{41a}$$

$$\boldsymbol{C_a} = \begin{bmatrix} \mathbb{O}_{n_y,n_x} & \mathbb{I}_{n_y,n_y} \end{bmatrix}, \qquad \boldsymbol{K_a} = \begin{bmatrix} \boldsymbol{K} \\ \boldsymbol{CK} \end{bmatrix}. \tag{41b}$$

The eigenvalues of the augmented model (40) comprise both the eigenvalues of the plant model (10) and an additional n_y eigenvalues located on the unit circle. This implies the presence of n_y integrators incorporated into the augmented design model. Although the augmented system and the initial system share the same outputs, their inputs differ. To introduce an integral action in DeePC, it suffices to consider Hankel matrices using incremental inputs, as detailed in Lazar and Verheijen (2022). In the LPV case, this involves defining matrices at each vertex as follows:

$$\begin{bmatrix} \boldsymbol{Y}^{[i]}_{0,\rho,N} \\ \boldsymbol{Y}^{[i]}_{\rho,L,N} \\ \boldsymbol{\Delta U}^{[i]}_{0,\rho,N} \\ \boldsymbol{\Delta U}^{[i]}_{\rho,L,N} \end{bmatrix}. \tag{42}$$

The DeePC problem with embedded integrator is recasted as

$$\arg\min_{\boldsymbol{\alpha}(k)} \left(\|\boldsymbol{\alpha}(k)\|^2_{\boldsymbol{W_I}} + 2\boldsymbol{C}^\top_{\boldsymbol{p_I}}\boldsymbol{\alpha}(k) \right), \tag{43}$$

subject to

$$\boldsymbol{Y}_{k-\rho,\rho,1} = \begin{bmatrix} \boldsymbol{Y}^{[1]}_{0,\rho,N} & \cdots & \boldsymbol{Y}^{[N_p]}_{0,\rho,N} \end{bmatrix} \boldsymbol{\alpha}(k),$$

$$\boldsymbol{\Delta U}_{k-\rho,\rho,1} = \begin{bmatrix} \boldsymbol{\Delta U}^{[1]}_{0,\rho,N} & \cdots & \boldsymbol{\Delta U}^{[N_p]}_{0,\rho,N} \end{bmatrix} \boldsymbol{\alpha}(k),$$

$$\begin{bmatrix} \boldsymbol{y}_r(k+L-\rho) \\ \vdots \\ \boldsymbol{y}_r(k+L-1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{Y}^{[1]}_{L,\rho,N} & \cdots & \boldsymbol{Y}^{[N_p]}_{L,\rho,N} \end{bmatrix} \boldsymbol{\alpha}(k),$$

$$\left(\boldsymbol{S}_{L,n_u} \begin{bmatrix} \boldsymbol{\Delta U}^{[1]}_{0,\rho,N} & \cdots & \boldsymbol{\Delta U}^{[N_p]}_{0,\rho,N} \end{bmatrix} \boldsymbol{\alpha}(k) + \mathbb{1}_{L,n_u}\boldsymbol{u}(k-1) \right)^{(i)} \in \mathcal{U},$$

$$\left(\begin{bmatrix} \boldsymbol{Y}^{[1]}_{0,\rho,N} & \cdots & \boldsymbol{Y}^{[N_p]}_{0,\rho,N} \end{bmatrix} \boldsymbol{\alpha}(k) \right)^{(i)} \in \mathcal{Y},$$

where $i \in \mathcal{I}_{1,L}$. $\boldsymbol{W_I}$ and $\boldsymbol{C_{p_I}}$ are given by

$$\begin{aligned} \boldsymbol{W_I} = &\begin{bmatrix} \boldsymbol{Y}^{[1]}_{\rho,L,N} & \cdots & \boldsymbol{Y}^{[N_p]}_{\rho,L,N} \end{bmatrix}^\top \boldsymbol{Q} \begin{bmatrix} \boldsymbol{Y}^{[1]}_{\rho,L,N} & \cdots & \boldsymbol{Y}^{[N_p]}_{\rho,L,N} \end{bmatrix}, \\ &+ \begin{bmatrix} \boldsymbol{\Delta U}^{[1]}_{\rho,L,N} & \cdots & \boldsymbol{\Delta U}^{[N_p]}_{\rho,L,N} \end{bmatrix}^\top \boldsymbol{R} \begin{bmatrix} \boldsymbol{\Delta U}^{[1]}_{\rho,L,N} & \cdots & \boldsymbol{\Delta U}^{[N_p]}_{\rho,L,N} \end{bmatrix}, \\ \boldsymbol{C_{p_I}} = &-\boldsymbol{Y}_r^\top \boldsymbol{Q} \begin{bmatrix} \boldsymbol{Y}^{[1]}_{\rho,L,N} & \cdots & \boldsymbol{Y}^{[N_p]}_{\rho,L,N} \end{bmatrix}. \end{aligned}$$

The resulting algorithm is summed up in Algorithm 3.

Algorithm 2 LPV-Data Enabled Predictive Control

Phase 1 – Before closing the control loop:

1: Choose $\boldsymbol{Q}$, $\boldsymbol{R}$, ρ, L, $\mathcal{Y}$, $\mathcal{U}$, and $\boldsymbol{Y}_r$ suitably
2: **for** $i = 1, \ldots, N_p$ **do**
3: Collect the open loop data sets $\boldsymbol{Y}^{[i]}_{0,\rho+L,N}$, $\boldsymbol{U}^{[i]}_{0,\rho+L,N}$
4: **end for**
5: Build $\boldsymbol{H}_{yu}$ as in Eq. (24)
6: Compute $\boldsymbol{W_I}$ and $\boldsymbol{C_{p_I}}$ from Eq. (18)

Phase 2 – During the control loop

7: **for** $k = 1, 2, \cdots$ **do**
8: Measure $\boldsymbol{y}(k)$
9: Minimize the optimization function in Eq. (43) for $\boldsymbol{\alpha}(k)$
10: Compute $\boldsymbol{\Delta U}_{k,L,1} = \boldsymbol{\Delta U}_{\rho,L,N}\boldsymbol{\alpha}(k)$
11: Apply the input $\boldsymbol{u}(k) = \boldsymbol{\Delta U}_{k,1,1} + \boldsymbol{u}(k-1)$ to the system
12: Update $\boldsymbol{C_{p_I}}$ if $\boldsymbol{Y}_r$ changes
13: Update the past vectors $\boldsymbol{\Delta U}_{k-\rho,\rho,1}$ and $\boldsymbol{Y}_{k-\rho,\rho,1}$
14: **end for**

3.5. Computational complexity

The formulation of the LPV system in the polytopic form (25) needs $N_p = 2^{n_p}$ vertices. This implies that for trajectory definition, we require $\boldsymbol{\alpha}(k) \in \mathbb{R}^{N*N_p}$. The exponential increase in the number of vertices N_p as the number of scheduling variables $\boldsymbol{p}$ increase can lead to large optimization problem and computational challenges. To

overcome this issue, one way is to consider the γ-DDPC algorithm first introduced in Breschi, Chiuso, and Formentin (2023b) and Breschi, Fabris, Formentin and Chiuso (2023) to parametrize the solution in terms of a lower dimensional parameter vector $\boldsymbol{\alpha}(k)$. Let us define

$$\boldsymbol{Z}_{ini}(k) = \begin{bmatrix} \boldsymbol{Y}_{k-\rho,\rho,1} \\ \boldsymbol{\Delta U}_{k-\rho,\rho,1} \end{bmatrix}, \quad \boldsymbol{Z}_p^{[i]} = \begin{bmatrix} \boldsymbol{Y}_{0,\rho,N}^{[i]} \\ \boldsymbol{\Delta U}_{0,\rho,N}^{[i]} \end{bmatrix},$$

and

$$\begin{aligned} \boldsymbol{Z}_p &= \begin{bmatrix} \boldsymbol{Z}^{[1]} & \dots & \boldsymbol{Z}^{[N_p]} \end{bmatrix}, \\ \boldsymbol{Y}^f &= \begin{bmatrix} \boldsymbol{Y}_{\rho,L,N}^{[1]} & \dots & \boldsymbol{Y}_{\rho,L,N}^{[N_p]} \end{bmatrix}, \\ \boldsymbol{\Delta U}^f &= \begin{bmatrix} \boldsymbol{\Delta U}_{\rho,L,N}^{[1]} & \dots & \boldsymbol{\Delta U}_{\rho,L,N}^{[N_p]} \end{bmatrix}. \end{aligned}$$

Consider the LQ decomposition

$$\begin{bmatrix} \boldsymbol{Z}_p \\ \boldsymbol{Y}^f \\ \boldsymbol{\Delta U}^f \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_{11} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{L}_{21} & \boldsymbol{L}_{22} & \boldsymbol{0} \\ \boldsymbol{L}_{31} & \boldsymbol{L}_{32} & \boldsymbol{L}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{Q}_1 \\ \boldsymbol{Q}_2 \\ \boldsymbol{Q}_3 \end{bmatrix}. \tag{44}$$

By multiplying both sides of Eq. (44) by $\boldsymbol{\alpha}(k)$, we have

$$\begin{bmatrix} \boldsymbol{Z}_{ini}(k) \\ \boldsymbol{Y}_{k,L,1} \\ \boldsymbol{\Delta U}_{k,L,1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}_{11} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{L}_{21} & \boldsymbol{L}_{22} & \boldsymbol{0} \\ \boldsymbol{L}_{31} & \boldsymbol{L}_{32} & \boldsymbol{L}_{33} \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma}_1(k) \\ \boldsymbol{\gamma}_2(k) \\ \boldsymbol{\gamma}_3(k) \end{bmatrix}, \tag{45}$$

where $\boldsymbol{\gamma}_i(k) = \boldsymbol{Q}_i\boldsymbol{\alpha}(k)$ for $i = 1, 2, 3$. The dimensions of $\boldsymbol{\gamma}_i(k)$ are independent of N and N_p. The optimization problem becomes computationally tractable by using $\boldsymbol{\gamma}_i(k)$ as optimization variables instead of $\boldsymbol{\alpha}(k)$, whatever the number of vertices of the polytope. Moreover, since $\boldsymbol{Z}_{ini}(k)$ is known at time k, $\boldsymbol{\gamma}_1(k)$ can be obtained from

$$\boldsymbol{\gamma}_1(k) = \boldsymbol{L}_{11}^{\dagger}\boldsymbol{Z}_{ini}(k). \tag{46}$$

By denoting

$$\boldsymbol{\gamma}(k) = \begin{bmatrix} \boldsymbol{\gamma}_2(k) \\ \boldsymbol{\gamma}_3(k) \end{bmatrix},$$

problem (43) thus reduces to

$$\arg\min_{\boldsymbol{\gamma}(k)} \left(\|\boldsymbol{\gamma}(k)\|_{\boldsymbol{W}_\gamma}^2 + 2\boldsymbol{C}_{p_\gamma}^{\top}\boldsymbol{\gamma}(k) \right), \tag{47}$$

subject to

$$\begin{bmatrix} \boldsymbol{y}_r(k+L-\rho) \\ \vdots \\ \boldsymbol{y}_r(k+L-1) \end{bmatrix} = \left[\boldsymbol{A}_1\boldsymbol{\gamma}(k) + \boldsymbol{L}_{21}\boldsymbol{\gamma}_1(k) \right]_{(L-\rho+1:L)},$$

$$\left(\boldsymbol{S}_{L,n_u} \left(\boldsymbol{A}_2\boldsymbol{\gamma}(k) + \boldsymbol{L}_{31}\boldsymbol{\gamma}_1(k) \right) + \mathbb{1}_{L,n_u}\boldsymbol{u}(k-1) \right)^{(i)} \in \mathcal{U},$$

$$\left(\boldsymbol{A}_1\boldsymbol{\gamma}(k) + \boldsymbol{L}_{21}\boldsymbol{\gamma}_1(k) \right)^{(i)} \in \mathcal{Y},$$

where $i \in \mathcal{I}_{1,L}$, $[.]_{(n:m)}$ represents the rows n to m of [.], whereas

$$\boldsymbol{A}_1 = \begin{bmatrix} \boldsymbol{L}_{22} & \boldsymbol{0} \end{bmatrix}, \tag{48}$$

$$\boldsymbol{A}_2 = \begin{bmatrix} \boldsymbol{L}_{32} & \boldsymbol{L}_{33} \end{bmatrix}, \tag{49}$$

$$\boldsymbol{W}_\gamma = \boldsymbol{A}_1^{\top}\boldsymbol{Q}\boldsymbol{A}_1 + \boldsymbol{A}_2^{\top}\boldsymbol{R}\boldsymbol{A}_2, \tag{50}$$

$$\boldsymbol{C}_{p_\gamma} = \boldsymbol{A}_1^{\top}\boldsymbol{Q}\left(\boldsymbol{L}_{21}\boldsymbol{\gamma}_1(k) - \boldsymbol{Y}_r \right) + \boldsymbol{A}_2^{\top}\boldsymbol{R}\boldsymbol{L}_{31}\boldsymbol{\gamma}_1(k). \tag{51}$$

The resulting algorithm is given in Algorithm 3.

Algorithm 3 γ-LPV-Data Enabled Predictive Control

Phase 1 – Before closing the control loop:

Choose $\boldsymbol{Q}$, $\boldsymbol{R}$, ρ, L, $\mathcal{Y}$, $\mathcal{U}$, and $\boldsymbol{Y}_r$ suitably
for $i = 1, \dots, N_p$ **do**
  Collect the open loop data sets $\boldsymbol{Y}_{0,\rho+L,N}^{[i]}$, $\boldsymbol{U}_{0,\rho+L,N}^{[i]}$
end for
Build $\boldsymbol{H}_{yu}$ as in Eq. (24)
Compute the LQ decomposition as in Eq. (44)
Compute $\boldsymbol{W}_\gamma$ and $\boldsymbol{C}_{p_\gamma}$ from Eq. (18)

Phase 2 – During the control loop

for $k = 1, 2, \cdots$ **do**
  Measure $\boldsymbol{y}(k)$
  Compute $\gamma_1(k)$ from Eq. (46)
  Minimize the optimization function in Eq. (43) for $\boldsymbol{\gamma}(k)$
  Compute $\boldsymbol{\Delta U}_{k,L,1} = \boldsymbol{L}_{31}\gamma_1(k) + \boldsymbol{A}_2\boldsymbol{\gamma}(k)$
  Apply the input $\boldsymbol{u}(k) = \boldsymbol{\Delta U}_{k,1,1} + \boldsymbol{u}(k-1)$ to the system
  Update $\boldsymbol{C}_{p_\gamma}$ and the past vectors $\boldsymbol{Z}_{ini}(k)$
end for

4. Numerical simulations

To demonstrate the effectiveness of this novel method, two examples are presented in this section. Firstly, the proposed control method undergoes testing on a numerical LPV system, as detailed in Section 4.1. Subsequently, the effectiveness of the proposed control strategy is evaluated using simulated data generated from a high-fidelity simulator replicating a calendering process.

4.1. Numerical example

Consider the LPV system $\boldsymbol{S} = (\boldsymbol{p}, \boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \boldsymbol{D})$ where

$$\boldsymbol{A} = \begin{bmatrix} 1-0.001p_1(k) & 0.01 & 0.02p_2(k) & 0 \\ 0 & 1-0.01025p_1(k) & 0 & 0.033 \\ 0.01p_2(k) & 0 & 1-0.0095p_1(k) & 0 \\ 0 & 0.05p_2(k) & 0.0375p_1(k) & -0.053 \end{bmatrix},$$

$$\boldsymbol{B} = \begin{bmatrix} 0.017 \\ 0.001 \\ 0.5 \\ 0.072 \end{bmatrix}, \quad \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}, \quad \boldsymbol{D} = \boldsymbol{0}.$$

All the possible trajectories of the scheduling parameter $\boldsymbol{p}(k) = \begin{bmatrix} p_1(k) \\ p_2(k) \end{bmatrix}$ evolve inside a convex polytope such that

$$\begin{bmatrix} 8 \\ 0.1 \end{bmatrix} \leq \boldsymbol{p}(k) \leq \begin{bmatrix} 16 \\ 2 \end{bmatrix}.$$

From Eq. (22), an orthotope $\boldsymbol{E}$ can be constructed with four vertices. Each vertex $\boldsymbol{E}^{[i]}$, $i = 1, \dots, 4$ defines an LTI system which is excited with an input signal $\boldsymbol{u}_d^{[i]}$ chosen as a Pseudo Random Binary Signal (PRBS) of length $N + \rho + L$. The data length is $N = 300$ time steps and the prediction horizon L is set as 40 time steps. This choice ensures the persistency of excitation of the input signal. The resulting Hankel matrices of the N_p LTI systems are horizontally stacked as per Eq. (30). The output and input penalty weights are chosen as $Q = 10$ and $R = 1$. The control objective is to track a reference trajectory $\boldsymbol{y_r}(k)$ while the scheduling parameter $\boldsymbol{p}(k)$ follows a trajectory demonstrated in Fig. 2. The input u is bounded between $[-20, 20]$.

The control input and output results from the closed-loop implementation of the control algorithm with the LPV system over 6000 time steps are presented in Figs. 3 and 4 respectively. As evident in Fig. 4, the DeePC algorithm (43)–(41) adeptly guides the system output (depicted by the blue line) toward the reference trajectory (indicated by the red line), effectively handling variations in the scheduling parameters. For comparison, the LPV system is also controlled using a DeePC controller tuned for an LTI system, considering the nominal LTI model of the studied system. The nominal model, representing the barycenter of $\boldsymbol{E}$, corresponds to the frozen model defined for $\boldsymbol{p} = [12, 1.05]$. The resulting output is depicted by the black line in Fig. 4. Notably, incorporating the

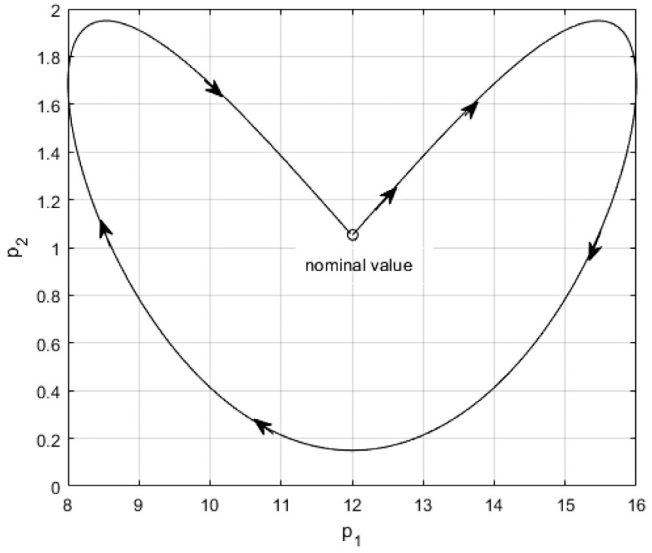

Fig. 2. Trajectory of the scheduling parameter $p(k)$. The nominal value $p = [12, 1.05]$ corresponds to the starting point ($k = 1$).

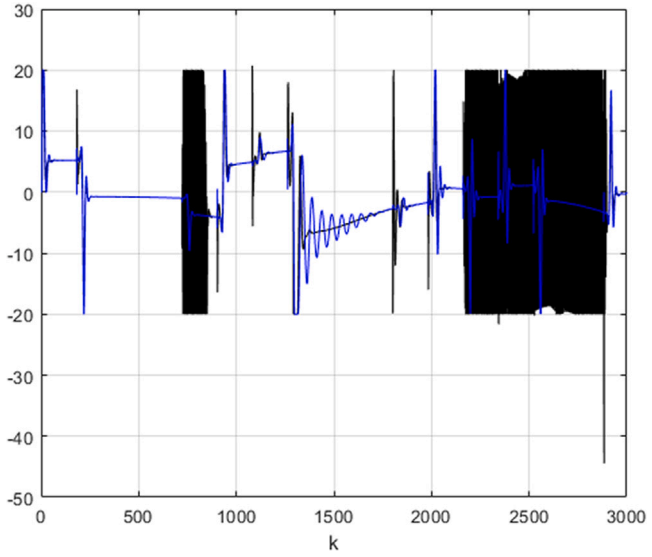

Fig. 3. Control inputs with DeePC for LPV system (blue line) and DeePC for nominal LTI system (black line). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

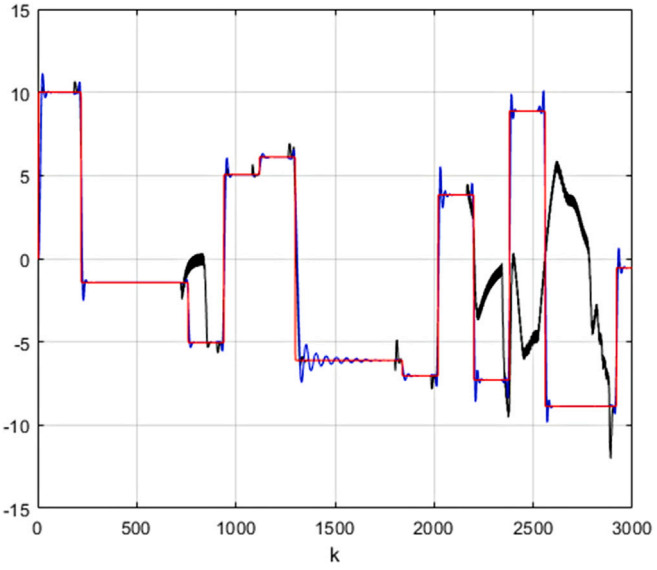

Fig. 4. Controlled outputs with DeePC for LPV system (blue line) and DeePC for nominal LTI system (black line). Red line corresponds to reference trajectory. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

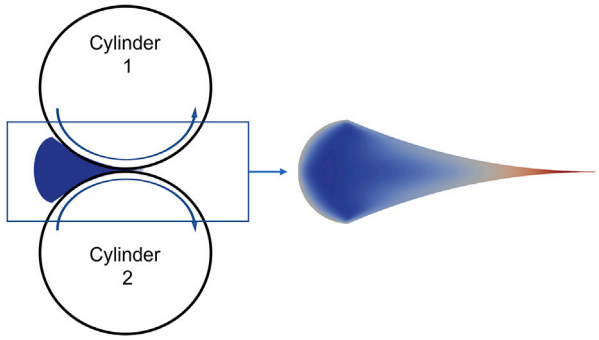

Fig. 5. A two dimensional scheme showing the calendering process with the spatial variation of the temperature at a given calendering velocity. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

LTI models on the vertices of the orthotope $\boldsymbol{E}$ surpasses the standard solution with the nominal LTI model. In the latter case, the controller fails to guide the system output to the required trajectory. It is important to note that the algorithm does not always find a solution to the optimization problem and may yield a saturated control signal, as shown in Fig. 3 (black line). In contrast, the DeePC algorithm for LPV systems yields an unsaturated control signal, facilitating the system output to precisely track its reference trajectory, as illustrated by the blue line in Fig. 3.

4.2. Calendering process control

4.2.1. Problem formulation

Through the tire manufacturing process, a pivotal calendering stage is employed to create thin rubber layers for subsequent processing. This crucial step entails compressing the rubber material through a series of rotating cylinders. Its importance is underscored by its role in preparing thin and seamless layers, ultimately shaping the overall quality of the manufactured tires. A two-dimensional scheme of the calendering rolls is illustrated in Fig. 5 depicting the spatial variation of the temperature within the rubber at a specific calendering velocity. Lower temperatures are visually represented in blue and higher temperatures are depicted in red. Within the calendering process, the viscoelastic properties of rubber can give rise to the phenomenon known as viscous heating. This heat generation is a function of the production rate, namely the rotational speed of the calendering rolls. Notably, substantial temperature fluctuations occur, particularly at the inlet, where the rubber mixture accumulates to form a rubber bed, and at the outlet, where the rubber exits the gap between the two cylindrical wheels. These specific regions experience the most significant temperature changes throughout the calendering process.

Maintaining the rubber temperature within a specific setpoint is crucial to preserve the quality of the rubber, underscoring the critical role of efficient control temperature dynamics in optimizing the tire manufacturing process. Despite the significant impact of this mechanical process on overall production, achieving full automation proves challenging due to the high nonlinearities inherent in the system. In this particular scenario, the objective is to regulate the rubber temperature at the process outlet while adhering to the constraints imposed by

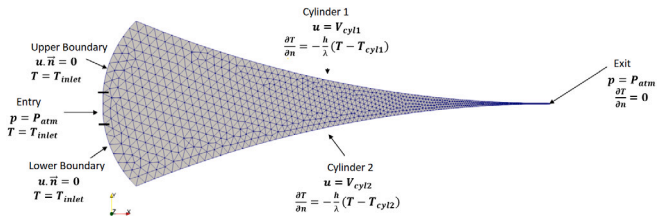

Fig. 6. Scheme of studied domain and the boundary conditions used for calculations.

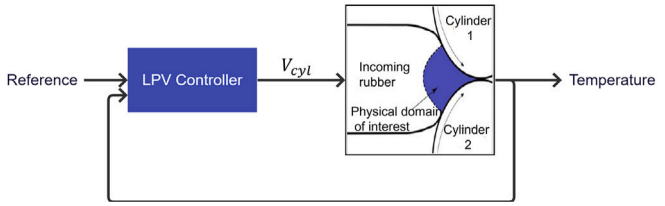

Fig. 7. Diagram showing the controller implementation in closed loop with the process simulator.

the maximum temperature the mixture can endure before breaking down. This control is implemented using a high-fidelity simulator of the calendering process. It is important to note that temperature variations at the inlet, influenced by the rubber feeding system, are not addressed in this example as they fall beyond its scope.

4.2.2. Solution framework

Within the scope of this example, the governing equations that model the dynamics of the rubber flow through the calendering process are formulated based on the following assumptions: (*i*) two dimensional flow, (*ii*) unsteady flow, (*iii*) the elastic property of the rubber is neglected thus eventually, the rubber is treated as an incompressible non Newtonian viscous fluid flow. Consequently, the equations characterizing the rubber behavior encompass the generalized Stokes equation for momentum conservation, a divergence-free velocity equation for mass conservation, and a convection–diffusion equation that incorporates a term for viscous heating for energy conservation. The constitutive law applied across all scenarios presented is a power-law coupled with an Arrhenius law, effectively accounting for temperature-dependent viscosity. (For the comprehensive set of physical equations, please refer to Appendix). To achieve precise High-Fidelity (HF) solutions for temperature profiles, we utilize an in-house finite element solver called MEF++ (developed collaboratively by Michelin and GIREF Laboratory Laval & GIREF, 2021). This simulator effectively solves the discretized physical equations that govern the calendering process with a sampling time of 1 s. The discretized domain and the associated boundary conditions utilized for the finite elements simulations are detailed in Fig. 6. As previously mentioned, the rotational velocities of the cylinders, denoted by V_{cyl1} for the upper cylinder and V_{cyl2} for the lower cylinder in Fig. 6, serve as the inputs of the system and they are to be adjusted to control the temperature of the rubber. Both of these cylinders in reality rotate together with a constant ratio, expressed as

$$\frac{V_{cyl1}}{V_{cyl2}} = fr \tag{52}$$

where $0 < fr \leq 1$ represents the frictional constant between the two calendering wheels. In the scope of this example, the frictional constant is chosen unitary. As a result, the rotational speeds are the same, leading to $V_{cyl1} = V_{cyl2}$. This leads to a unified input for the system, denoted as V_{cyl}. Based on this, the LPV data driven controller can be implemented as shown in Fig. 7.

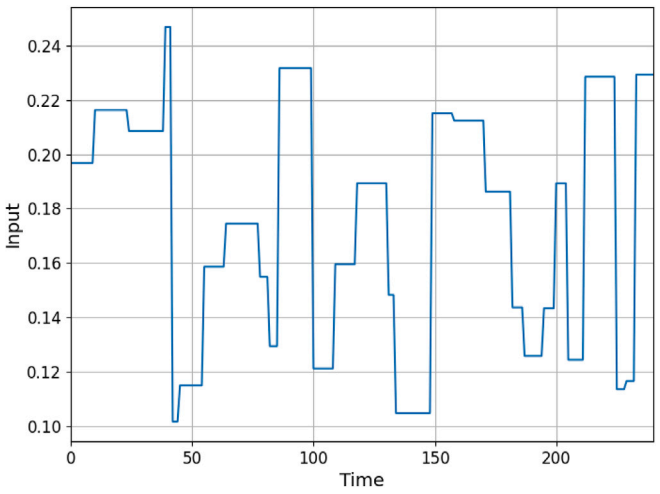

Fig. 8. Random cylinder rolls velocity signal used to excite the system dynamics.

4.2.3. Control implementation and results

The dynamic viscosity of the rubber, denoted as η, is a function of both the rubber temperature and the calendering velocity in the physical domain, as explained in Appendix. In order to find the bounds of this varying parameter, a multi step signal is used to excite the system dynamics. Specifically, for the considered application, the rotational speeds of the calendering wheels range between 0.01 m/s and 0.7 m/s. Accordingly, the viscosity of the rubber (expressed in Pa.s) is found to fall within the interval of $[30, 600]$. The established viscosity bounds result in a Reynolds number, denoted as Re, that is less than 1 aligning with the physical assumptions inherent in the equations presented in Appendix. Based on this, the calendering system can be modeled as an LPV system where the scheduling parameter is the dynamic viscosity η. All the feasible trajectories of η evolve in the convex polytopic set, η^v defined by $N_p = 2$ vertices such that

$$\eta^v = \{30, 600\}. \tag{53}$$

At each bound of the dynamic viscosity range, a constant dynamic viscosity is assumed. From a physical standpoint, this can be achieved by employing two different materials, each characterized by a constant viscosity. Subsequently, the resulting LTI systems are simulated using the signal presented in Fig. 8.

On the one hand, a prediction horizon of 15 time steps is adopted, $L = 15$. On the other hand, the number of past time steps required must be at least equal to the number of states corresponding to the system. The number of actual states is equal to the number of nodes based on the spatial discretization of the physical domain. In this example, the discretized domain in Fig. 6 contains 8387 space nodes. In order to determine the number of states of this system, the simulator is excited over 200 time steps and a snapshot of the complete system states (*i.e*, temperature field) is collected. A singular value decomposition of the temperature snapshot matrix shows that only 9 singular values are sufficient to capture 99.99% of the temperature variance. Consequently, the system order can be reduced to $n_x = 9$ states, (for further details about model order reduction, the reader is referred to Schilders, van der Vorst, and Rommes (2008)). Based on this, the past data length is set as 10 time steps.

In addressing the minimization problem for the controller, the weighting matrices Q and R are selected as 10 and 0.1 respectively. The upper and lower limits for the calendering velocity are established at 0.01 m/s and 0.7 m/s, aligning with the physical constraints of the calendering process. Our new solution performance is benchmarked against DeePC for LTI systems, considering the system behavior as linear time-invariant (LTI), and DeePC for LTI systems based on a

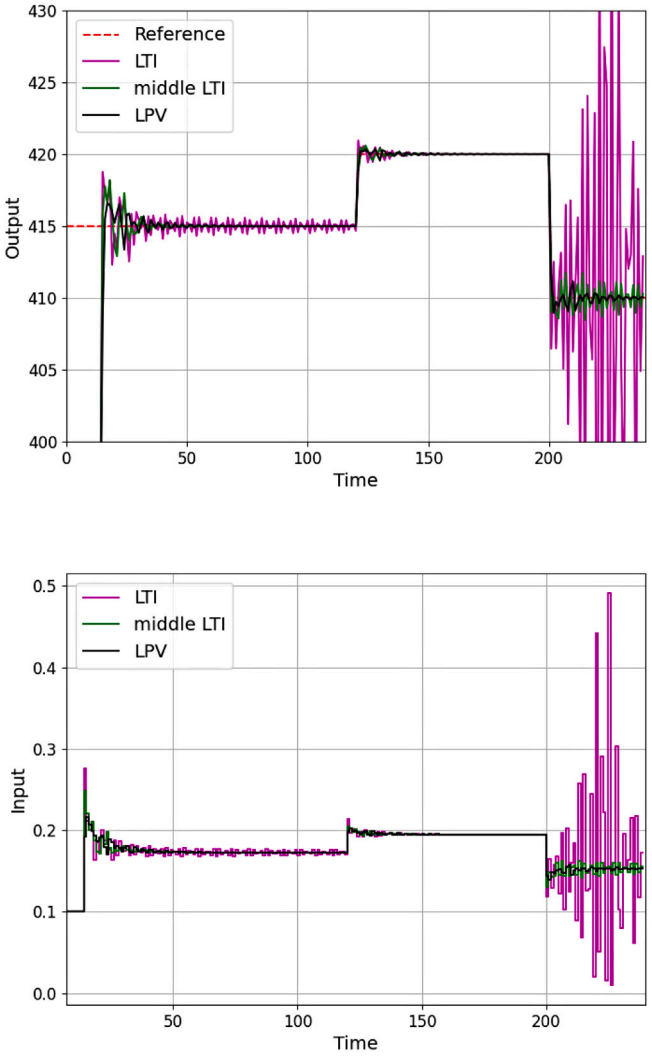

Fig. 9. Temporal variation of the output temperature (K) (top) and the calendering speed as input (m/s) (bottom) after the implementation of three different controllers for a given mixture of rubber.

nominal system. The latter assumes a fixed viscosity value at $\eta = 450$ Pa.s, selected to ensure that the temperature variation aligns with that of the considered rubber mixture. The outcomes of applying the three controllers are illustrated in Fig. 9 for a given rubber mix. Fig. 9 illustrates the effective application of the proposed method, showcasing its ability to adjust the system input and drive the output to converge accurately toward various reference values with minimal error. Notably, the DeePC based on the nominal system successfully controls the system for the first two reference values but encounters difficulty with the third. Conversely, the controller assuming the system as a Linear Time Invariant (LTI) model fails to regulate the system when the reference is set at $T_r = 410$ K. These outcomes, stemming from the assumption of the system as an LTI model, are expected due to the inherent limitations of LTI models in capturing system nonlinearities, contrasting with the high efficiency of modeling the system using an LPV model.

In real-world applications, the viscosity of rubber material varies across different mixtures. To assess the efficacy of our proposed method, the controller undergoes testing with two distinct rubber mixtures. The viscosity variations of the two considered mixtures is shown in Fig. 10 due to the excitation of the calendering system using the calendering velocity signal depicted in Fig. 8.

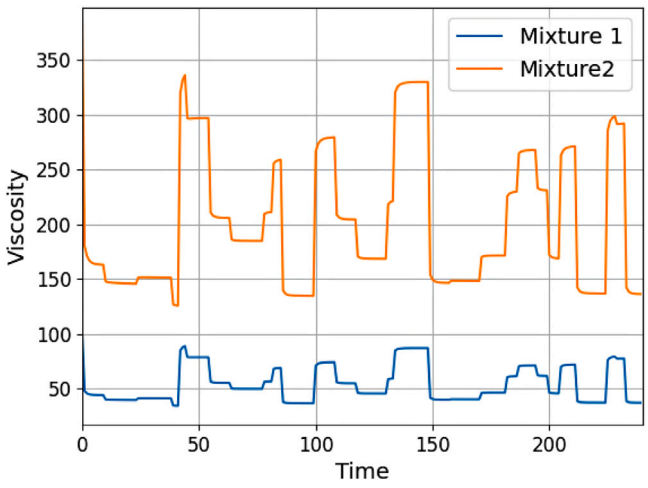

Fig. 10. Temporal variation of the viscosity of two different rubber mixtures due to the excitation of the calendering process using the calendering speed in Fig. 8.

It is crucial to emphasize that the temporal variation of viscosity for both materials around the output occurs within the polytope η^v, staying within the prescribed excitation limits. The outcomes of implementing the introduced control method on the two materials are illustrated in Figs. 11 and 12 accordingly. Figs. 11 and 12 depict the system output converging to the setpoint with slight oscillations around it for the two types of rubber mixtures undergoing calendering. Notably, when the required calendering rolls velocity reaches 0.275 m/s, exceeding the maximum velocity used during excitation (0.25 m/s, as shown in Fig. 8), the controller adeptly manages system dynamics variations, showcasing its robustness. This underscores the effectiveness of the proposed approach across different rubber types without prior knowledge of the corresponding viscosity temporal variation. Additionally, the controller exhibits inherent versatility, enabling its direct application to various materials without requiring an initialization procedure. As anticipated, controllers utilizing Hankel matrices from system excitation with a base material or assuming the system is an LTI system fail to regulate the system effectively, struggling to steer the output toward the predefined setpoint.

5. Conclusion

This paper confronts two core challenges in predictive control for LPV systems: the identification of a system model and the real-time measurement of time-varying parameters. Through a polytopic formulation of the model, a novel derivation of Willem's lemma is introduced. The proposed approach involves measuring a system trajectory at each vertex of the polytope, necessitating the freezing of parameters at all combinations of their extreme values. However, the notable advantage lies in eliminating the need to know the scheduling parameter's value at each moment for effective system control. This methodology preserves the advantages of Data Driven Predictive Control approaches, such as the absence of an identification stage, inherent consideration of multivariable systems, an output feedback control law without the need for designing an observer, and the incorporation of equality and/or inequality constraints on inputs/outputs.

The primary drawback of this method is its dependency on the ability to generate a trajectory on each vertex with persistent excitation. This requires accessibility and tuning of the components of $\boldsymbol{p}$, which may not always be feasible or practical. Another limitation lies in the requirement for $\boldsymbol{p}$ to evolve within a polytope, stemming from the necessity of modeling the LPV system in polytopic form. However, the challenge of exponential computational complexity growth with the increasing number of parameters can be solved by the way of an LQ

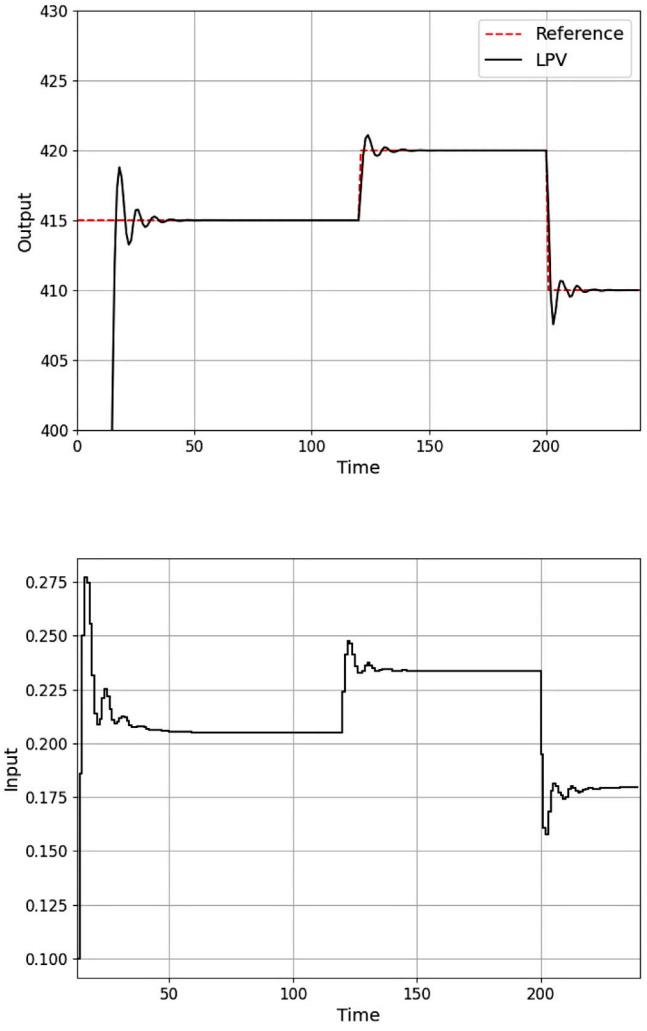

Fig. 11. Figure showing the variation of the output temperature and the adjusted input after the implementation of the LPV controller using the first mixture.

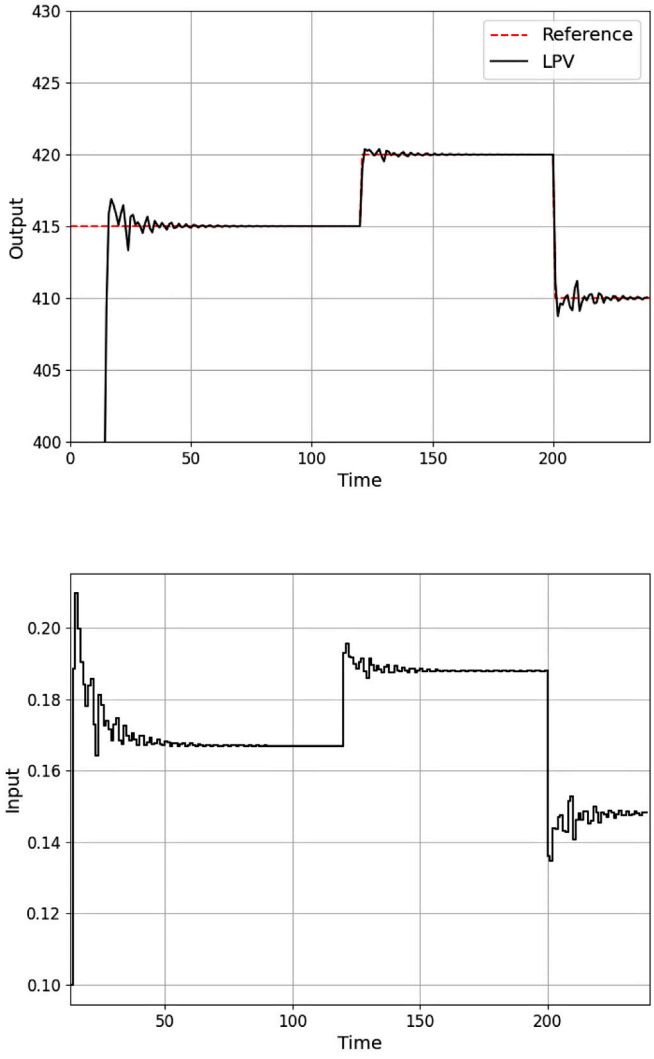

Fig. 12. Figure showing the variation of the output temperature and the adjusted input after the implementation of the LPV controller using the second mixture.

transformation conducted offline, i.e. prior to closing the control loop. The resulting computational complexity akin to the initial DeePC for LTI systems, irrespective of the scheduling parameter dimension.

The effectiveness of this approach is substantiated by results obtained from a high-fidelity, realistic process simulator modeling a rubber calendering process employed in tire manufacturing. The subsequent phase of this study involves implementing this novel method in an actual operational process and to evaluate its robustness in presence of noisy data.

CRediT authorship contribution statement

Taleb Bou Hamdan: Writing – original draft. **Patrick Coirault:** Supervision, Methodology, Conceptualization. **Guillaume Mercère:** Supervision, Methodology. **Thibault Dairay:** Supervision, Project administration.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Coirault Patrick reports was provided by University of Poitiers. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. Physical equations

Based on the assumptions presented in Section 4.2.2, the conservation equations can be formulated in a simplified manner. The conservation of momentum (Batchelor, 1967) can be written in the form

$$-\nabla.(2\eta(\mathbf{u})\dot{\gamma}(\mathbf{u})) - \nabla.p = 0, \tag{A.1}$$

where $\nabla.$ stands for the divergence operator, $\mathbf{u}$ is the two dimensional velocity vector, p is the pressure applied on the rubber, $\dot{\gamma}$ is the general deformation rate related to the shear rate, and η is the dynamic viscosity of the rubber. This equation is applicable for very low Reynolds number, denoted as Re (Lee, Lee, & Yeo, 2014). The Reynolds number is a dimensionless quantity used in fluid mechanics to characterize the flow of a fluid (liquid or gas). It provides information about the relative importance of inertial forces to viscous forces in the fluid flow. The definition of the Reynolds number is:

$$\mathrm{Re} = \frac{\rho V L}{\eta} \tag{A.2}$$

where L is a characteristic length. V is a reference velocity. ρ is the density of the fluid and η is the fluid dynamic viscosity. The dynamic viscosity of the rubber follows a power law (George & Qureshi, 2013)

with

$$\eta(\mathbf{u}) = \eta(\dot{\gamma}, T) = K(\tau\dot{\gamma})^{n-1} \exp(\alpha(\frac{1}{T} - \frac{1}{T_{ref}})), \tag{A.3}$$

where τ is a time characteristic constant, K is the fluid consistency, n is the exponent of the power law such that $n < 1$, α is the ratio of the material activation energy over the perfect gas constant, $\alpha = E_a/R$, T is the temperature of the rubber and T_{ref} is the reference temperature of the Arrhenius law. The mass conservation equation for an incompressible flow is given by

$$\nabla.\mathbf{u} = 0. \tag{A.4}$$

The energy conservation equation is given by (Fletcher, 1991):

$$\frac{\partial T}{\partial t} + \mathbf{u}.\nabla T = \frac{\lambda}{\rho C_p}\nabla^2 T + \frac{\eta(\mathbf{u})\dot{\gamma}^2(\mathbf{u})}{\rho C_p} \tag{A.5}$$

where λ is the rubber thermal conductivity, C_p is the specific heat capacity of the rubber. The additional term $\frac{\eta(\mathbf{u})\dot{\gamma}^2(\mathbf{u})}{\rho C_p}$ is the viscous heating source term. The dynamic viscosity $\eta(\mathbf{u})$ is calculated using the power law in Eq. (A.3). This induces the coupling between the conservation equations, *i.e.* Eq. (A.1), Eqs. (A.4) and (A.5) (thermo-mechanical coupling).

References

Batchelor, G. (1967). *An introduction to fluid dynamic: vol. 52*, Cambridge University Press.

Berberich, J., & Allgöwer, F. (2020). A trajectory-based framework for data-driven system analysis and control. In *European control conference* (pp. 1365–1370). http://dx.doi.org/10.23919/ECC51009.2020.9143608.

Berberich, J., Köhler, J., Müller, M. A., & Allgöwer, F. (2021). On the design of terminal ingredients for data-driven MPC. *IFAC-PapersOnLine*, *54*(6), 257–263. http://dx.doi.org/10.1016/j.ifacol.2021.08.554, 7th IFAC Conference on Nonlinear Model Predictive Control NMPC 2021.

Breschi, V., Chiuso, A., & Formentin, S. (2023a). Data-driven predictive control in a stochastic setting: a unified framework. *Automatica*, (152), 1–16. http://dx.doi.org/10.1016/j.automatica.2023.110961.

Breschi, V., Chiuso, A., & Formentin, S. (2023b). Data-driven predictive control in a stochastic setting: a unified framework. *Automatica*, *152*, Article 110961. http://dx.doi.org/10.1016/j.automatica.2023.110961.

Breschi, V., Fabris, M., Formentin, S., & Chiuso, A. (2023). Uncertainty-aware data-driven predictive control in a stochastic setting. *IFAC-PapersOnLine*, *56*(2), 10083–10088. http://dx.doi.org/10.1016/j.ifacol.2023.10.878, 22nd IFAC World Congress.

Dinkla, R., Mulders, S., van Wingerden, J.-W., & Oomen, T. (2023). Closed-loop aspects of data-enabled predictive control. In *IFAC 22st triennial world congress, Yokohama, Japan, 2023* (pp. 1388–1393). http://dx.doi.org/10.1016/j.ifacol.2023.10.1806.

Dörfler, F., Coulson, J., & Markovsky, I. (2023). Bridging direct and indirect data-driven control formulations via regularizations and relaxations. *IEEE Transactions on Automatic Control*, *68*(2), 883–897. http://dx.doi.org/10.1109/TAC.2022.3148374.

Favoreel, W., Moor, B. D., & Gevers, M. (1999). SPC: Subspace predictive control. *IFAC Proceedings Volumes*, *32*(2), 4004–4009. http://dx.doi.org/10.1016/S1474-6670(17)56683-5, 14th IFAC World Congress 1999, Beijing, Chia, 5-9 July.

Fletcher, C. A. J. (1991). Fluid dynamics: The governing equations. In *Computational techniques for fluid dynamics 2: specific techniques for different flow categories* (pp. 1–46). Springer Berlin, chapter 1.

George, H. F., & Qureshi, F. (2013). Newton's law of viscosity, Newtonian and non-Newtonian fluids. In *Encyclopedia of tribology* (pp. 2416–2420). Boston, MA: Springer.

Gidon, D., Abbas, H. S., Bonzanini, A. D., Graves, D. B., Mohammadpour Velni, J., & Mesbah, A. (2021). Data-driven LPV model predictive control of a cold atmospheric plasma jet for biomaterials processing. *Control Engineering Practice*, *109*, Article 104725. http://dx.doi.org/10.1016/j.conengprac.2021.104725.

Kapsalis, D., Sename, O., Milanes, V., & Molina, J. J. (2022). A reduced LPV polytopic look-ahead steering controller for autonomous vehicles. *Control Engineering Practice*, *129*, Article 105360. http://dx.doi.org/10.1016/j.conengprac.2022.105360.

Laval, U., & GIREF (2021). MEF++. URL: https://fr.wikipedia.org/wiki/MEF%2B%2B.

Lazar, M., & Verheijen, P. (2022). Offset-free data-driven predictive control. In *IEEE 61st confernce on decision and control* (pp. 1099–1104). Cancun, Mexico: http://dx.doi.org/10.1109/CDC51059.2022.9992453.

Lee, S. H., Lee, K.-K., & Yeo, I. (2014). Assessment of the validity of Stokes and Reynolds equations for fluid flow through a rough-walled fracture with flow imaging. *Geophysical Research Letters*, *41*, http://dx.doi.org/10.1002/2014GL060481.

Li, S., Nguyen, A.-T., Guerra, T.-M., & Kruszewski, A. (2023). Reduced-order model based dynamic tracking for soft manipulators: Data-driven LPV modeling, control design and experimental results. *Control Engineering Practice*, *138*, Article 105618. http://dx.doi.org/10.1016/j.conengprac.2023.105618.

Markovsky, I., & Dörfler, F. (2021). Behavioral systems theory in data-driven analysis, signal processing, and control. *Annual Reviews in Control*, *52*, 42–64. http://dx.doi.org/10.1016/j.arcontrol.2021.09.005.

Morato, M. M. M., Normey-Rico, J., & Sename, O. (2020). Model predictive control design for linear parameter varying systems: A survey. *Annual Reviews in Control*, (49), 64–80. http://dx.doi.org/10.1016/j.arcontrol.2020.04.016.

Rawlings, J., Mayne, D., & Diehl, M. (2017). *Model predictive control: Theory, computation, and design*. Nob Hill Publishing, URL: https://books.google.fr/books?id=MrJctAEACAAJ.

Schilders, W., van der Vorst, H., & Rommes, J. (2008). *Model order reduction: theory, research aspects and applications*. Springer.

Toth, R. (2010). *Modeling and identification of linear parameter varying systems*. Springer.

Van Overschee, P., & De Moor, B. (1994). N4SID: subspace algorithms for the identification of combined deterministic stochastic systems. *Automatica*, *30*, 75–93. http://dx.doi.org/10.1016/0005-1098(94)90230-5.

Van Overschee, P., & De Moor, B. (1996). *Subspace identification for linear systems. Theory, implementation, applications*. Kluwer Academic Publishers.

Van Waarde, H. J., De Persis, C., Camlibel, M. K., & Tesi, P. (2020). Willems' fundamental lemma for state-space systems and its extension to multiple datasets. *IEEE Control Systems Letters*, *4*(3), 602–607. http://dx.doi.org/10.1109/LCSYS.2020.2986991.

Verhaegen, M., & Verdult, V. (2007). *Filtering and system identification: a least squares approach*. Cambridge University Press.

Verheijen, P., Breschi, V., & Lazar, M. (2023). Handbook of linear data-driven predictive control: Theory, implementation and design. *Annual Reviews in Control*, *56*, Article 100914. http://dx.doi.org/10.1016/j.arcontrol.2023.100914.

Verhoek, C., Abbas, H., Toth, R., & Haesaert, S. (2021). Data-driven predictive control for linear parameter varying systems. *IFAC PapersOnline*, 101–108. http://dx.doi.org/10.1016/j.ifacol.2021.08.588.

Willems, J. (2007). System theoretic models for the analysis of physical systems. *Ricerche di Automatica*, *10*, 287–301.

Willems, J., Rapisard, P., Markovsky, I., & De Moor, B. (2005). A note on persistency of excitation. *Systems & Control Letters*, *54*, 325–329. http://dx.doi.org/10.1016/j.sysconle.2004.09.003.