Post-print version:

Complexity reduction of piecewise affine linear parameter varying controllers

F. D. Bianchi, and R. S. Sánchez-Peña

This work has been published in **Automatica**:

F. D. Bianchi, and R. S. Sánchez-Peña, "Complexity reduction of piecewise affine linear parameter varying controllers", *Automatica*, vol. 167, pp. 111783, 2024.

Final version available at:

URL: https://www.sciencedirect.com/science/article/abs/pii/S0005109824002772

DOI: 10.1016/j.automatica.2024.111783

BibTex:

```
@Article{Bianchi2024,
    Title   = {Complexity reduction of piecewise affine linear parameter varying
       controllers},
    Author  = {Fernando D. Bianchi, and Ricardo S. Sánchez-Peña},
    Journal = {Automatica},
    Year    = {2024},
    Number  = {},
    Pages   = {111783},
    Volume  = {167},
    Doi     = {10.1016/j.automatica.2024.111783}
}
```

Complexity reduction of piecewise affine linear parameter varying controllers

Fernando D. Bianchi [a], Ricardo S. Sánchez-Peña [a]

[a] *Instituto Tecnológico Buenos Aires (ITBA) and Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Ciudad Autónoma de Buenos Aires, Argentina*

Abstract

This work analyzes the design and implementation of piecewise affine (PWA) Linear Parameter Varying (LPV) controllers. PWA-LPV models can be employed to describe complex nonlinear or time-varying systems with the aim of control design. The complexity of a PWA-LPV controller depends on the number of local models used to describe the plant. Here we propose a methodology to reduce this number and thus simplify the design and implementation complexity of the resulting controller without a significant performance degradation. The methodology is illustrated by an academic example and a more realistic wind turbine control application.

Key words: Piecewise affine (PWA), linear parameter varying (LPV), gain scheduling (GS), implementation complexity.

1 Introduction

Gain scheduling (GS) is a popular tool to address the control of nonlinear and time-varying systems. A GS strategy consists of a set of linear controllers parameterized by the operating conditions (Åström & Wittenmark, 2008). The linear parameter varying (LPV) framework provides more adequate and systematic design tools (Apkarian, Gahinet & Becker, 1995; Becker & Packard, 1994; Wu, Yang, Packard & Becker, 1996; Apkarian & Adams, 1998; Scherer, 2001). In particular, the design in the GS-LPV approach is formulated as a convex optimization problem in which robust control concepts can be included naturally in order to consider model approximation errors. GS-LPV techniques have been used in a large number of applications. An extensive list can be found in (Hoffmann & Werner, 2015) and some more recent applications in *e.g.* (Morera-Torres, Ocampo-Martinez & Bianchi, 2022; Häberle, Fisher, Prieto-Araujo & Dörfler, 2022), among others.

In GS-LPV control design, the plant is described as a linear model parameterized by exogenous variables (scheduling variables) that take values in a known parameter space. Several approaches are available to fulfill this task depending on the particular plant. For instance, an LPV model can be obtained from nonlinear dynamics expressions by a change of variables or Jacobian linearization (Mohammadpour & Scherer, 2012; Rotondo, 2018). Another modeling strategy is the piecewise affine (PWA) LPV representation introduced by Lim & How (2002, 2003). A PWA-LPV model is obtained from a set of linear models corresponding to a grid of frozen parameter values or operating points, and an affine interpolation for intermediate points. PWA-LPV modeling offers a systematic methodology and may be suitable to embed highly complex nonlinear plants in LPV descriptions, especially when analytic expressions are too complex or the model is based on look-up-tables. In general, the model approximation improves with the number of linear models.

The particular parameter dependence of the LPV model determines the computational complexity of the control design. In case of affine and PWA parameter dependent plants, the design reduces to solving a convex optimization problem with a finite set of decision variables and constraints (Apkarian et al., 1995). The number of these decision variables and constraints is related to the number of vertices in a convex hull covering the parameter space or to the number of points in the grid in case of PWA-LPV models. The complexity of controller implementation, *i.e.* the mathematical operations and information storage necessary for computing the control law in real-time, is also connected to the number of these vertices or points.

In case of LPV plants with a general parameter dependence, the parameter space is sampled in order to obtain a problem with a finite number of variables and constraints (Wu

⋆ This paper was not presented at any IFAC meeting. Corresponding author F. Bianchi.

Email addresses: febianchi@itba.edu.ar (Fernando D. Bianchi), rsanchez@itba.edu.ar (Ricardo S. Sánchez-Peña).

et al., 1996; Apkarian & Adams, 1998). In this approach, known as gridding, the closed-loop stability and performance is only checked at a discrete set of parameter values. Hence, the denser the grid of points, the more accurate is the stability and performance estimation. Unfortunately, the burden of the controller design also increases with the grid density. As an alternative to avoid gridding, a general LPV plant can be covered with an affine LPV model at the expense of a conservative design (Kwiatkowski & Werner, 2008). A number of methodologies have been proposed to improve the model fitting by adding auxiliary parameters and tightening the parameter space to a convex hull with additional vertices (Hoffmann, Hashemi, Abbas & Werner, 2014; Sadeghzadeh, Sharif & Tóth, 2020; Sadeghzadeh & Tóth, 2023; Abbas, Tóth, Petreczky, Meskin, Mohammadpour Velni & Koelewijn, 2021; Kapsalis, Sename, Milanes & Molina, 2022). Although these methods also aim to determine models with low number of auxiliary parameters and vertices, the resulting representations still might lead to controllers with a significant number of variables, and difficult to design and/or implement.

As the complexity is in some degree inherited from the plant dynamics representation, the application of LPV techniques in high-order nonlinear and time-varying systems is still hard. This may limit the plants in which LPV techniques can be successfully used to design GS controllers. Nevertheless, in some cases, a dynamic model with apparently complex nonlinear expressions can be successfully controlled with relatively simple controllers (Sánchez-Peña & Bianchi, 2012; Belikov, Kotta & Tõnso, 2014). Considering this fact, it appears interesting to seek simpler LPV models leading to low complexity control design and implementation without a significant performance degradation. In this vein, Bianchi & Sánchez-Peña (2022) proposed a methodology aimed at simplifying the LPV model expressions and thus the controller design and implementation. The gridding is still necessary but the implementation complexity does not depend on the density of the grid. The present article focuses on the GS control design for PWA-LPV plants. The objective is to find a methodology for reducing the number of linear models used in the system description and thus simplifying the control design and implementation complexity. The proposed methodology analyzes the effects on the closed-loop stability and performance to determine a less dense grid of points to design a reduced-complexity PWA-LPV controller.

The article is organized as follows. Section 2 presents a brief summary on PWA-LPV modeling and control design. Section 3 introduces the main result, a methodology to reduce the complexity of the design and implementation of PWA-LPV controllers. Then, the effectiveness of the proposed methodology is analyzed with two examples: an academic one in which the model simplification can be easily shown (Section 4), and a practical application in the control of a wind turbine, a highly complex nonlinear system (Section 5). Finally, some conclusions are drawn in Section 6.

Notation: The following notation will be used:

$$\begin{bmatrix} P+P^T & R^T \\ R & S \end{bmatrix} = \begin{bmatrix} P+(\star) & \star \\ R & S \end{bmatrix}.$$

For a matrix $Q \in \mathbb{R}^{n\times m}$, $\bar{\sigma}(Q)$ denotes its maximum singular value. For a real symmetric matrix Q, $Q > 0$ and $Q \geq 0$ stand for positive definite and positive semi-definite, respectively, and $Q < 0$ and $Q \leq 0$ for negative definite and negative semi-definite, respectively. The identity matrix of dimension $n \times n$ is denoted as I_n.

2 PWA-LPV modeling and control design

Consider the following LPV system

$$G(\theta): \begin{cases} \dot{x} = A(\theta)x + B_1(\theta)w + B_2u, \\ z = C_1(\theta)x + D_{11}(\theta)w + D_{12}u, \\ y = C_2x + D_{21}w, \end{cases} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ is the state vector, $w \in \mathbb{R}^{n_w}$ is a disturbance, $u \in \mathbb{R}^{n_u}$ is the control input, $z \in \mathbb{R}^{n_z}$ is an output related to performance specifications, and $y \in \mathbb{R}^{n_y}$ is the measured output. The time-varying parameter θ is assumed to be measured in real-time and taking values in a compact set $\mathcal{P} \subset \mathbb{R}^{n_p}$.

The parameter dependent matrices in (1) are assumed to be PWA functions with the following expression

$$\begin{bmatrix} A(\theta) & B_1(\theta) \\ C_1(\theta) & D_{11}(\theta) \end{bmatrix} = \sum_{i=1}^{n_v} \alpha_i(\theta) \begin{bmatrix} A_i & B_{1,i} \\ C_{1,i} & D_{11,i} \end{bmatrix}, \tag{2}$$

where A_i, $B_{1,i}$, $C_{1,i}$ and $D_{11,i}$ $(i = 1, \ldots, n_v)$ are constant matrices.

Let $\mathcal{P}_g = \{\theta^1, \ldots, \theta^{n_v}\}$ be a grid of points with $\mathcal{P}$ its convex hull. Using triangulation, the set $\mathcal{P}$ is split into a set of simplices or subdivisions $\mathcal{S}_k$ $(k = 1, \ldots, m_v)$. Hence, any point $\theta \in \mathcal{S}_k$ can be expressed as a convex combination of its nearest points or vertices of $\mathcal{S}_k$, denoted as $\mathcal{V}(\mathcal{S}_k)$. That is,

$$\theta = \sum_{i=1}^{n_v} \alpha_i\theta^i, \quad \sum_{i=1}^{n_v} \alpha_i = 1, \quad \text{and} \quad 0 \leq \alpha_i \leq 1, \tag{3}$$

where $\alpha_i = 0$ for all $\theta^i \notin \mathcal{V}(\mathcal{S}_k)$. The multipliers $\alpha_i(\theta)$'s in (2) are the weights of the convex decomposition (3) for the particular value θ. The matrices A_i, $B_{1,i}$, $C_{1,i}$ and $D_{11,i}$ correspond to the values of the functions (2) evaluated at $\theta^i \in \mathcal{P}_g$. Figure 1 illustrates the PWA representation of one element $[m_{ij}]$ of matrix (2) for $\theta \in \mathbb{R}$.

The synthesis problem consists in finding an LPV controller

$$K(\theta): \begin{cases} \dot{x}_c = A_c(\theta)x_c + B_c(\theta)y, \\ u = C_c(\theta)x_c + D_c(\theta)y, \end{cases} \tag{4}$$

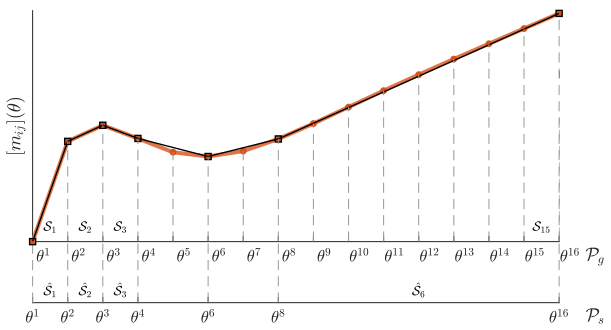

Figure 1. Illustrative example of one element of the matrix (2), the grid of points $\mathcal{P}_g$ and simplices $\mathcal{S}$ for a case of $\theta \in \mathbb{R}$. The red line corresponds to the PWA interpolation based on the grid $\mathcal{P}_g$ and the black line based on a grid $\mathcal{P}_s$ reduced according to the affine parameter dependence of the system matrices.

that ensures closed-loop quadratic stability and

$$\|z\|_2 \leq \gamma \|w\|_2, \quad \forall\, \theta \in \mathcal{P}. \tag{5}$$

In the LPV framework, $G(\theta)$ in (1) is known as the generalized or augmented plant, which includes the weighting functions used to establish the control specifications in the synthesis procedure (Wu et al., 1996; Apkarian & Adams, 1998). The measured parameter θ is the scheduling variable of the GS controller $K(\theta)$.

Following the procedure introduced in (Scherer, Gahinet & Chilali, 1997), this design problem can be cast as a convex optimization problem that minimizes γ subject to the following linear matrix inequalities (LMIs):

$$\Pi(\mathbf{X}, \mathbf{Y}, \hat{\mathbf{A}}(\theta), \hat{\mathbf{B}}(\theta), \hat{\mathbf{C}}(\theta), \hat{\mathbf{D}}(\theta), \gamma) < 0, \tag{6}$$

$$\begin{bmatrix} \mathbf{X} & I_{n_s} \\ I_{n_s} & \mathbf{Y} \end{bmatrix} > 0, \tag{7}$$

where $\Pi(\cdot)$ is given in (8), and the matrices $\mathbf{X} = \mathbf{X}^T$, $\mathbf{Y} = \mathbf{Y}^T$, auxiliary matrix functions

$$\begin{bmatrix} \hat{\mathbf{A}}(\theta) & \hat{\mathbf{B}}(\theta) \\ \hat{\mathbf{C}}(\theta) & \hat{\mathbf{D}}(\theta) \end{bmatrix} = \sum_{i=1}^{n_v} \alpha_i(\theta) \begin{bmatrix} \hat{\mathbf{A}}_i & \hat{\mathbf{B}}_i \\ \hat{\mathbf{C}}_i & \hat{\mathbf{D}}_i \end{bmatrix}, \tag{9}$$

and the scalar $\gamma > 0$ are decision variables to be found (Apkarian & Adams, 1998). As the model and auxiliary matrix functions are PWA functions of θ, the testing of LMIs (6) and (7) at the points $\theta^i \in \mathcal{P}_g$ holds for all $\theta \in \mathcal{P}$ (Apkarian et al., 1995). Therefore, the controller design reduces to finding matrices $\mathbf{X}$, $\mathbf{Y}$, $\hat{\mathbf{A}}_i$, $\hat{\mathbf{B}}_i$, $\hat{\mathbf{C}}_i$, $\hat{\mathbf{D}}_i$ that minimize γ subject to

$$\Pi(\mathbf{X}, \mathbf{Y}, \hat{\mathbf{A}}_i, \hat{\mathbf{B}}_i, \hat{\mathbf{C}}_i, \hat{\mathbf{D}}_i, \gamma) < 0, \quad i = 1, \ldots, n_v, \tag{10}$$

$$\begin{bmatrix} \mathbf{X} & I_{n_s} \\ I_{n_s} & \mathbf{Y} \end{bmatrix} > 0. \tag{11}$$

Once the optimization problem is solved, the controller matrices are computed from

$$A_c(\theta) = N^{-1}(\hat{\mathbf{A}}(\theta) - \mathbf{X}(A(\theta) - B_2\hat{\mathbf{D}}(\theta)C_2)\mathbf{Y} - \hat{\mathbf{B}}(\theta)C_2\mathbf{Y} - \mathbf{X}B_2\hat{\mathbf{C}}(\theta))M^{-T}, \tag{12}$$

$$B_c(\theta) = N^{-1}(\hat{\mathbf{B}}(\theta) - \mathbf{X}B_2\hat{\mathbf{D}}(\theta)), \tag{13}$$

$$C_c(\theta) = (\hat{\mathbf{C}}(\theta) - \hat{\mathbf{D}}(\theta)C_2\mathbf{Y})M^{-T}, \tag{14}$$

$$D_c(\theta) = \hat{\mathbf{D}}(\theta), \tag{15}$$

where M and N are selected to satisfy $I - \mathbf{X}\mathbf{Y} = NM^T$.

Notice that the controller matrix functions (12)-(15) are also PWA functions of the parameter θ and therefore they can also be expressed as

$$\begin{bmatrix} A_c(\theta) & B_c(\theta) \\ C_c(\theta) & D_c(\theta) \end{bmatrix} = \sum_{i=1}^{n_v} \alpha_i(\theta) \begin{bmatrix} A_{c,i} & B_{c,i} \\ C_{c,i} & D_{c,i} \end{bmatrix}, \tag{16}$$

and α_i according to (3).

The system description (1) can be seen as a particular case of PWA-LPV systems introduced by Lim & How (2002). Here, the plant and controller matrix functions are assumed continuous and the Lyapunov functions constant [1] .

In case the PWA-LPV model is used to embed a nonlinear plant in an LPV description, the matrices A_i, $B_{1,i}$, $C_{1,i}$, and $D_{11,i}$ correspond to linearizations of the nonlinear plant at

[1] These assumptions might limit the closed-loop performance in same cases but ease the presentation of the proposed methodology. The use of parameter dependent Lyapunov functions makes the design and implementation more complex, but does not alter the complexity reduction criterion introduced in the next section.

$$\Pi(\mathbf{X}, \mathbf{Y}, \hat{\mathbf{A}}(\theta), \hat{\mathbf{B}}(\theta), \hat{\mathbf{C}}(\theta), \hat{\mathbf{D}}(\theta), \gamma) = \begin{bmatrix} A(\theta)\mathbf{Y} + B_2\hat{\mathbf{C}}(\theta) + (\star) & \star & \star & \star \\ \hat{\mathbf{A}}(\theta) + (A(\theta) + B_2\hat{\mathbf{D}}(\theta)C_2)^T & \mathbf{X}A(\theta) + \hat{\mathbf{B}}(\theta)C_2 + (\star) & \star & \star \\ (B_1(\theta) + B_2\hat{\mathbf{D}}(\theta)D_{21})^T & (\mathbf{X}B_1(\theta) + \hat{\mathbf{B}}(\theta)D_{21})^T & -\gamma I_{n_w} & \star \\ C_1(\theta)\mathbf{Y} + D_{12}\hat{\mathbf{C}}(\theta) & C_1(\theta) + D_{12}\hat{\mathbf{D}}(\theta)C_2 & D_{11}(\theta) + D_{12}\hat{\mathbf{D}}(\theta)D_{21} & -\gamma I_{n_z} \end{bmatrix} \tag{8}$$

points $\theta^i \in \mathcal{P}_g$. For instance, the set of LTI models may come from numerical linearizations of well-validated and high-fidelity simulation models. In these cases, it is difficult to predict in which areas of the entire operating envelope the system dynamics exhibits the most significant changes. As a result, $\mathcal{P}_g$ is taken as a sufficiently dense and equally spaced set of points in order to have a good approximation of the nonlinear model for all $\theta \in \mathcal{P}$.

3 Low complexity LPV control design

The online computation of the control action u requires storing the set of n_v matrices in (16), solving the convex decomposition (3), and determining u from (4). Clearly, the implementation complexity of the LPV controller given by (16) depends on the number of points n_v in the grid $\mathcal{P}_g$. The higher n_v, the larger is the number of variables to be stored and the more complex is the computation of the α_i's. A large number of points in $\mathcal{P}_g$ also implies a higher computational burden for the controller design as it increases the number of decision variables and LMI constraints.

If the matrix function (2) exhibits an almost affine parameter dependence in some areas of $\mathcal{P}$, the PWA-LPV model can be described with a smaller grid $\mathcal{P}_s$ resulting in a controller with a simpler implementation. For instance, in Figure 1 the black line is a PWA interpolation based on a reduced grid $\mathcal{P}_s$, which introduces only a small error in the values of $[m_{ij}]$ as compared with the full grid interpolation based on $\mathcal{P}_g$ (red line). For any point θ between θ^8 and θ^{16}, $[m_{ij}]$ can be described by only one affine function with domain given by the simplex $\hat{\mathcal{S}}_6$ with vertices θ^8 and θ^{16}, instead of the original PWA interpolation based on 8 affine functions corresponding to a grid of 9 points. Considering this fact, the rest of this section presents a criteria to find a smaller grid of points and a controller design to simplify the implementation.

3.1 Reduction of the parameter grid

The criterion for reducing the number of points in the parameter grid can be based on the linear dependency of the matrices A_i, $B_{1,i}$, $C_{1,i}$, and $D_{11,i}$. However, this criterion does not clearly show the effect of using a smaller grid on the closed-loop stability and performance as defined in (5). For this reason, we will use the ideas introduced by Bianchi & Sánchez-Peña (2022) to formulate such a criterion.

To this end, let us assume that the grid $\mathcal{P}_g$ is partitioned into two sets $\mathcal{P}_s$ and $\mathcal{P}_r$ such that

$$\mathcal{P}_s \cap \mathcal{P}_r = \emptyset, \qquad \mathcal{P}_s \cup \mathcal{P}_r = \mathcal{P}_g. \tag{17}$$

The smaller set $\mathcal{P}_s$ contains the vertices of new simplices $\hat{\mathcal{S}}_h$ ($h = 1, \ldots, m_s$ and $m_s < m_v$), in which the matrix function (2) presents an almost affine parameter dependence. On the other hand, the set $\mathcal{P}_r$ includes the remaining points that can be included in one of the new simplices $\hat{\mathcal{S}}_h$'s. The aim is to find a controller described as a convex combination of matrices $A_{c,j}$, $B_{c,j}$, $C_{c,j}$, and $D_{c,j}$ ($j = 1, \ldots, n_s$ and $n_s < n_v$) only associated to the points in $\mathcal{P}_s$. Following the example in Figure 1, $\mathcal{P}_s = \{\theta^1, \theta^2, \theta^3, \theta^4, \theta^6, \theta^8, \theta^{16}\}$ and $\mathcal{P}_r = \{\theta^5, \theta^7, \theta^9, \theta^{10}, \theta^{11}, \theta^{12}, \theta^{13}, \theta^{14}, \theta^{15}\}$.

In order to select the points in $\mathcal{P}_s$, we need to take into account the effect on the closed-loop stability and performance, when modifying the controller interpolation based on the smaller grid $\mathcal{P}_s$. That is, constraints (6)-(7) must hold for all $\theta \in \mathcal{P}$ with the new expressions of the controller matrices:

$$\begin{bmatrix} A_c(\theta) & B_c(\theta) \\ C_c(\theta) & D_c(\theta) \end{bmatrix} = \sum_{j=1}^{n_s} \eta_j(\theta) \begin{bmatrix} A_{c,j} & B_{c,j} \\ C_{c,j} & D_{c,j} \end{bmatrix}, \tag{18}$$

and

$$\theta = \sum_{j=1}^{n_s} \eta_j \theta^{\mathcal{I}(j)}, \quad \sum_{j=1}^{n_s} \eta_j = 1, \quad \text{and} \quad 0 \leq \eta_j \leq 1, \tag{19}$$

where $\eta_j = 0$ for all $\theta^{\mathcal{I}(j)} \notin \mathcal{V}(\hat{\mathcal{S}}_h)$, when $\theta \in \hat{\mathcal{S}}_h$. The symbol $\mathcal{I}(\cdot)$ denotes the mapping from the indices of $\mathcal{P}_s$ to the indices of $\mathcal{P}_g$, for example, in Figure 1, $\mathcal{I}(5)$ will return the index 6 as the fifth element in $\mathcal{P}_s$ is θ^6.

As mentioned in Section 2, due to the fact that the system (1) is assumed described by a PWA-LPV model, the constraint (6) only needs to be imposed at the points $\theta^i \in \mathcal{P}_g$. For all points $\theta^i \in \mathcal{P}_s$, the inequality (6) is the same as those presented in Section 2, since these points are vertices of both interpolations (the original (16) and the new (18)). On the other hand, for all points $\theta^i \in \mathcal{P}_r$, this constraint changes as a result of the new interpolation. At each $\theta^i \in \mathcal{P}_r$, the controller matrices are given by

$$\begin{bmatrix} A_c(\theta^i) & B_c(\theta^i) \\ C_c(\theta^i) & D_c(\theta^i) \end{bmatrix} = \sum_{j=1}^{n_s} \eta_j(\theta^i) \begin{bmatrix} A_{c,j} & B_{c,j} \\ C_{c,j} & D_{c,j} \end{bmatrix} \tag{20}$$

and the closed-loop matrices with these new controller matrices result:

$$\left[\begin{array}{c|c} \mathcal{A}(\theta^i) & \mathcal{B}(\theta^i) \\ \hline \mathcal{C}(\theta^i) & \mathcal{D}(\theta^i) \end{array}\right] = \left[\begin{array}{cc|c} A_i & 0 & B_{1,i} \\ 0 & 0 & 0 \\ \hline C_{1,i} & 0 & D_{11,i} \end{array}\right] + \left[\begin{array}{cc|c} B_2 D_c(\theta^i) C_2 & B_2 C_c(\theta^i) & B_2 D_c(\theta^i) D_{21} \\ B_c(\theta^i) C_2 & A_c(\theta^i) & B_c(\theta^i) D_{21} \\ \hline D_{12} D_c(\theta^i) C_2 & D_{21} C_c(\theta^i) & D_{12} D_c(\theta^i) D_{21} \end{array}\right]. \tag{21}$$

Then, applying the change of variable proposed by Scherer et al. (1997), and using the same interpolation (18) for the

auxiliary matrices (9), *i.e.*,

$$\begin{bmatrix} \hat{\mathbf{A}}(\theta) & \hat{\mathbf{B}}(\theta) \\ \hat{\mathbf{C}}(\theta) & \hat{\mathbf{D}}(\theta) \end{bmatrix} = \sum_{j=1}^{n_s} \eta_j(\theta) \begin{bmatrix} \hat{\mathbf{A}}_j & \hat{\mathbf{B}}_j \\ \hat{\mathbf{C}}_j & \hat{\mathbf{D}}_j \end{bmatrix}, \tag{23}$$

inequality (6) becomes

$$\Pi(\mathbf{X}, \mathbf{Y}, \hat{\mathbf{A}}(\theta^i), \hat{\mathbf{B}}(\theta^i), \hat{\mathbf{C}}(\theta^i), \hat{\mathbf{D}}(\theta^i), \gamma) + \underbrace{\begin{bmatrix} 0 & \star & \star & \star \\ \mathbf{X}A_{\mathrm{r},i}\mathbf{Y} & 0 & \star & \star \\ 0 & 0 & 0 & \star \\ 0 & 0 & 0 & 0 \end{bmatrix}}_{\Gamma(\mathbf{X},\mathbf{Y})} < 0, \tag{24}$$

for all $\theta^i \in \mathcal{P}_r$. The matrix $A_{\mathrm{r},i}$ is defined as

$$A_{\mathrm{r},i} = A_i - A(\theta^i), \tag{25}$$

where

$$A(\theta^i) = \sum_{j=1}^{n_s} \eta_j(\theta^i) A_{\mathcal{I}(j)} \tag{26}$$

and the $\eta_j(\theta^i)$'s are computed according to (19). The matrix $A_{\mathrm{r},i}$ represents the error between the matrix A_i and the approximation $A(\theta^i)$ using the interpolation (26).

Comparing (10) and (24), it can be seen that the term $\Gamma(\mathbf{X}, \mathbf{Y})$ is the additional restriction the new controller must satisfy to ensure closed-loop stability and performance. Then, considering the fact:

$$\bar{\sigma}(\mathbf{X}A_{\mathrm{r},i}\mathbf{Y}) \leq \bar{\sigma}(\mathbf{X})\bar{\sigma}(A_{\mathrm{r},i})\bar{\sigma}(\mathbf{Y}),$$

the effect of the new approximated interpolation can be connected with the maximum singular value of the matrices $A_{\mathrm{r},i}$. If they are close to zero, the interpolation will not introduce a significant effect on the closed-loop behavior. Taken this fact into account, the following value

$$\nu(A_{\mathrm{r},i}, A_i) = \frac{\bar{\sigma}(A_{\mathrm{r},i})}{\bar{\sigma}(A_i)}$$

can be proposed as a measure of the effect on stability and performance introduced by the simplified interpolation (18).

The measure $\nu(A_{\mathrm{r},i}, A_i)$ can be used to formulate an algorithm to determine the set partition (17). Algorithm 1 is an implementation of such a partition strategy for $\theta \in \mathbb{R}$. The algorithm starts with the first and third points in $\mathcal{P}_g$ and checks if $\nu(A_{\mathrm{r},2}, A_2)$ is lower than a tolerance ϵ. If so, θ^2 is excluded from $\mathcal{P}_s$ and the test is repeated using the vertices θ^1 and θ^4. Otherwise, θ^2 is included in $\mathcal{P}_s$ and the next test is performed using θ^2 and θ^4. The algorithm continues until the last point in $\mathcal{P}_g$ is tested. For higher parameter dimensions, the nonuniform grid can be obtained with partition algorithms as those used to find nonuniform meshes (de Berg, Cheong, van Kreveld & Overmars, 2008). The partition of the parameter space is performed starting with the largest simplices corresponding to the convex hull of $\mathcal{P}_g$ and iteratively splitting them into smaller ones, if the condition $\nu(A_{\mathrm{r},i}, A_i) < \epsilon$ is not satisfied.

Algorithm 1 Reduction of parameter set

```
P_s ← {θ^1}, j ← 1
for i = 2 to n_v − 1 do
  Compute φ such that θ^i = φ θ^j + (1 − φ)θ^{i+1}
  A_{r,i} = A_i − (φA_j + (1 − φ)A_{i+1})
  Compute δ = ν(A_{r,i}, A_i)
  if δ > ε then
    j ← i
    P_s ← P_s ∪ {θ^i}
  end if
end for
```

3.2 Control design based on a smaller grid

Once the smaller grid $\mathcal{P}_s$ is determined using the measure $\nu(A_{\mathrm{r},i}, A_i)$, two design options are available:

- *Method 1:* The controller is designed using a standard gridding strategy (Apkarian & Adams, 1998; Wu et al., 1996) as presented in Section 2. That is, the controller is obtained solving the optimization problem with the LMIs (6) and (7) over the smaller set $\mathcal{P}_s$, ignoring the constraints associated to the points in $\mathcal{P}_r$. With the resulting controller, closed-loop stability and performance is checked in the denser grid $\mathcal{P}_g$. If the error introduced by the new interpolation is small, this controller in general will satisfy these conditions.
- *Method 2:* The controller is designed adapting the procedures proposed by Bianchi & Sánchez-Peña (2022) in order to consider the interpolation errors as it will be shown next.

Both resulting controllers will be similar in terms of implementation complexity. However, Method 1 involves fewer LMI constraints but might require to recompute the partition (17). On the other hand, the second option produces a controller that ensures stability and performance in all $\theta^i \in \mathcal{P}_g$ for a given interpolation partition, at the expense of a more complex design optimization problem with more constraints and a possible conservative design.

More concretely, Method 2 aims to find the controller considering the additional term $\Gamma(\mathbf{X}, \mathbf{Y})$ in (24) arisen from the new interpolation. However, this term $\Gamma(\mathbf{X}, \mathbf{Y})$ is nonlinear and the matrix inequality (24) is not an LMI. As a consequence, the control design is a non-convex optimization problem. In (Bianchi & Sánchez-Peña, 2022), two procedures are proposed to approximately solve this problem. Here, we adapt Procedure 1 to the present synthesis problem, the other procedure can be adapted using similar arguments. The main idea consists in considering the fact that the matrices $A_{\mathrm{r},i}$'s are usually sparse with rank $n_a \leq n_x$

and they can be decomposed using singular values as:

$$A_{\mathrm{r},i} = \begin{bmatrix} U_{1,i} & U_{2,i} \end{bmatrix} \begin{bmatrix} \Sigma_i & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_{1,i}^T \\ V_{2,i}^T \end{bmatrix} = E_i F_i^T,$$

where $E_i = U_{1,i}\Sigma_i^{1/2}$, $F_i = V_{1,i}\Sigma_i^{1/2}$, and $\Sigma_i \in \mathbb{R}^{n_a}$. Then, the following inequality holds

$$\bar{\sigma}(\mathbf{X}A_{\mathrm{r},i}\mathbf{Y}) \leq \bar{\sigma}(\mathbf{X}E_i)\bar{\sigma}(\mathbf{Y}F_i).$$

Let us introduce the bounds ε_x and ε_y, such that

$$\bar{\sigma}(\mathbf{X}E_i)^2 \leq \varepsilon_x, \qquad \bar{\sigma}(\mathbf{Y}F_i)^2 \leq \varepsilon_y.$$

With these definitions, the following procedure can be used to find a simpler PWA-LPV controller for the model (1) that ensures quadratic stability and the performance criterion (5).

Procedure 1 *Given the partition* (17)*, find symmetric matrices* $\mathbf{X}$ *and* $\mathbf{Y}$*, matrices* $\hat{\mathbf{A}}_j$*,* $\hat{\mathbf{B}}_j$*,* $\hat{\mathbf{C}}_j$ *and* $\hat{\mathbf{D}}_j$ *(*$j = 1, \ldots, n_s$*) according to* (9)*, and positive scalars* γ*,* ε_x ε_y *that minimize*

$$\gamma + q(\varepsilon_x + \varepsilon_y)$$

subject to:

$$\Pi(\mathbf{X}, \mathbf{Y}, \hat{\mathbf{A}}(\theta^i), \hat{\mathbf{B}}(\theta^i), \hat{\mathbf{C}}(\theta^i), \hat{\mathbf{D}}(\theta^i), \gamma) < 0, \quad i = 1, \ldots, n_v$$

$$\begin{bmatrix} I_{n_s} & \mathbf{X}E_{\mathcal{J}(\ell)} \\ E_{\mathcal{J}(\ell)}^T\mathbf{X} & I_{n_a}\varepsilon_x \end{bmatrix} > 0, \quad \ell = 1, \ldots, n_r$$

$$\begin{bmatrix} I_{n_s} & \mathbf{Y}F_{\mathcal{J}(\ell)} \\ F_{\mathcal{J}(\ell)}^T\mathbf{Y} & I_{n_a}\varepsilon_y \end{bmatrix} > 0,$$

$$\begin{bmatrix} \mathbf{X} & I_{n_s} \\ I_{n_s} & \mathbf{Y} \end{bmatrix} > 0,$$

for a given weight $q > 0$*, where* $\mathcal{J}(\cdot)$ *denotes the mapping from the indices of* $\mathcal{P}_r$ *to the indices of* $\mathcal{P}_g$*, and* $n_r = n_v - n_s$*. Check if there exists a* γ *such that, with the obtained variables* $\mathbf{X}$*,* $\mathbf{Y}$*,* $\hat{\mathbf{A}}_j$*,* $\hat{\mathbf{B}}_j$*,* $\hat{\mathbf{C}}_j$ *and* $\hat{\mathbf{D}}_j$*, the condition* (24) *is satisfied, otherwise increase* q *and repeat the design.* ■

3.3 Selection of tolerance ϵ

The tolerance ϵ used in Algorithm 1 to find a smaller grid is a result of a trade-off between implementation complexity, and stability and performance degradation. As a starting point, the tolerance can be set high enough to achieve a significant reduction in the number of points. Then, if the stability and performance degradation results unacceptable, the tolerance should be decreased and a new reduced grid determined in order to find a more suitable controller.

4 Academic Example

In order to illustrate the procedure proposed in the previous section, consider the LPV system (1) with matrices:

$$A(\theta) = \begin{bmatrix} -50 & 0 & 0 & 0 & 0 \\ 50\,k(\theta) & -50 & 0 & 0 & 0 \\ 0 & p(\theta) & -p(\theta) & 0 & 0 \\ 0 & 0 & -40 & -5 & 0 \\ 0 & 0 & 0 & 0 & -5000 \end{bmatrix},$$

$$B_1 = \begin{bmatrix} 0 & 0 & 0 & 8 & 0 \end{bmatrix}^T, \quad B_2 = \begin{bmatrix} 5 & 0 & 0 & 0 & 64 \end{bmatrix}^T,$$

$$C_1 = \begin{bmatrix} 0 & 0 & -1 & 12.3750 & 0 \\ 0 & 0 & 0 & 0 & -77.3438 \end{bmatrix},$$

$$C_2 = \begin{bmatrix} 0 & 0 & -5 & 0 & 64 \end{bmatrix},$$

$$D_{11} = \begin{bmatrix} 0.2 & 0 \end{bmatrix}^T, \quad D_{12} = \begin{bmatrix} 0 & 1 \end{bmatrix}^T, \quad D_{21} = 1,$$

$p(\theta) = 50\tanh(20(\theta - 0.02))$, and $k(\theta) = 1 + 20\,(\exp(-20\,\theta) - \exp(-55\,\theta))$. The parameter θ takes values in $\mathcal{P} = \{\theta \in \mathbb{R} : 0 \leq \theta \leq 1\}$. Let us assume that only a set of $n_v = 41$ system matrices are available for the controller design. These matrices correspond to $A(\theta)$ evaluated at the points in the grid $\mathcal{P}_g = \{0, 0.025, 0.05, \ldots, 1\}$. Figure 2 shows the parameter functions $p(\theta)$ and $k(\theta)$ and three PWA interpolations corresponding to three grids. It can be seen that a dense grid is necessary to capture the change in $p(\theta)$ and $k(\theta)$ if an equally spaced set of points is used. The top-plot in Figure 2 shows the interpolation obtained with a sufficiently dense grid $\mathcal{P}_g$. On the other hand, the mid-plot presents an interpolation using a less dense and equally spaced grid $\mathcal{P}_d = \{0, 0.1, 0.2, \ldots, 1\}$, with 11 points. Clearly, part of the changes in the parameter functions are missed. Finally, the bottom-plot presents the interpolation using a less dense but not equally spaced grid $\mathcal{P}_s$ obtained using Algorithm 1 presented in Section 3. This is a more sensible grid of points.

Four controllers were designed using the three grid of parameter values shown in Figure 2. The controllers $K_f(\theta)$, $K_d(\theta)$ and $K_{s,1}(\theta)$ were designed with the synthesis procedure presented in Section 2 considering the grids $\mathcal{P}_g$, $\mathcal{P}_d$ and $\mathcal{P}_s$, respectively. The controller $K_{s,2}(\theta)$ was designed with Procedure 1 introduced in Section 3 and the grid $\mathcal{P}_s$. The optimization problems were solved using Yalmip (Löfberg, 2004) and Sedumi (Sturm, 1999). Table 1 lists the performance levels γ obtained when each controller is designed. The last column shows the analysis performance levels γ_a that result from a subsequent closed-loop performance evaluation performed over the complete grid of 41 points. Note that, in case of K_d, the performance level γ, computed in the design, is lower than the one obtained for the controller K_f. However, the analysis results infeasible as the controller K_d is not able to ensure quadratic stability in the complete

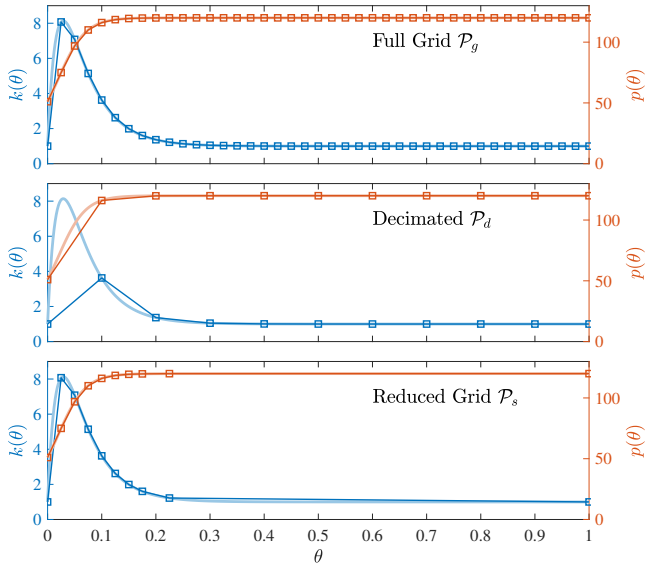

Figure 2. Academic example: Nonlinear functions $p(\theta)$ and $k(\theta)$ and three interpolations corresponding to three parameter grids $\mathcal{P}_g$, $\mathcal{P}_d$ and $\mathcal{P}_s$

Table 1
Academic example: Performance levels for each controller: γ obtained from synthesis and γ_a from analysis

Ctrl.	No. points	γ	γ_a
K_f	41	2.849	2.787
K_d	11	2.384	Infeasible
$K_{s,1}$	10	2.848	2.931
$K_{s,2}$	10	3.031	2.970

grid of points $\mathcal{P}_g$. This is because this controller ignores the changes in the plant dynamics at low values of θ. On the other hand, controllers $K_{s,1}$ and $K_{s,2}$ ensure stability and performance in all points in $\mathcal{P}_g$. It can be observed that in case of controller $K_{s,1}$, γ_a is slightly higher than γ because its design only considers the points in $\mathcal{P}_s$. However, the corresponding closed-loop system is stable and the performance level is acceptable, thanks to the proposed criterion for the grid reduction. On the other hand, controller $K_{s,2}$ results slightly more conservative compared to K_f (higher γ), but it provides a better closed-loop performance estimation ($\gamma > \gamma_a$).

Closed-loop simulations can be seen in Figure 3 for the parameter trajectory indicated in the bottom-plot and a square-wave as reference to be tracked (gray line). The tracking performance achieved by each controller is in accordance to the grid considered in each design. The baseline controller $K_f(\theta)$ (blue line) achieves a rather uniform response for all parameter range. Whereas the controllers $K_{s,1}(\theta)$ and $K_{s,2}(\theta)$ exhibit a slight degradation in the tracking for parameter values around 0.4 and 0.1. This can be associated to the small interpolation errors in this area of $\mathcal{P}$. However, the complexity in the design and implementation using set $\mathcal{P}_s$

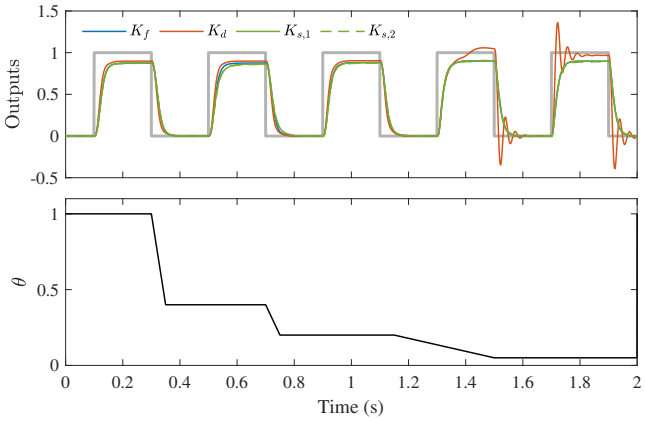

Figure 3. Academic example: Closed-loop simulations for a square-wave reference (gray line) and a parameter trajectory sweeping the entire set $\mathcal{P}$

is much lower. The benefits of the proposed gridding procedures is clear from the closed-loop response comparison with the design for set $\mathcal{P}_d$. It has significant errors in lower values of θ and the stability in this region is rather poor.

5 Wind turbine example

To illustrate the proposed procedure in a more realistic application, the methodology is used in the control of the 5 MW NREL baseline wind turbine (Jonkman, Butterfield, Musial & Scott, 2009). Wind turbines are highly complex and flexible mechanical structures that required proper control strategies to ensure a high efficient energy conversion and to limit mechanical loads to guarantee a long useful life. Several high fidelity simulators are available to evaluate the turbine dynamic behavior, which also include baseline controllers for a proper comparison with new proposals. (OpenFast 3.4, 2023) is one of these simulators, which is widely used to compare new strategies [2] .

The control scheme is shown in Figure 4, similar to that used in (Inthamoussou, Bianchi, De Battista & Mantz, 2014). The wind turbine has three inputs: the wind speed v (disturbance), the generator torque $T_{g,\text{ref}}$ and the pitch angle β (both used as control actions). As usual, the control strategy is split into two main operating regions. When the wind speed is below the rated value (11.4 m/s), the wind turbine is controlled by the $T_{g,\text{ref}}$ computed from a look-up-table (see (Jonkman et al., 2009)) while the pitch angle remains constant at $0°$. The aim is to maximize the energy capture. When the wind speed is above the rated value, the pitch controller regulates the generator speed Ω_g around the rated value $\Omega_{\text{ref}} = 1173.7$ rpm and $T_{g,\text{ref}}$ is kept at the rated value 43.09 kNm to maintain the generated power at the rated value of 5 MW. Additionally, the tower fore-after velocity $\dot{\chi}$ is also fed-back to help reduce the fore-after tower

[2] The wind turbine model and the baseline controller are available at `https://github.com/OpenFAST/openfast`

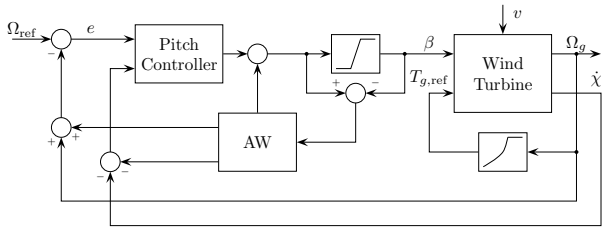

Figure 4. Control scheme used in the wind turbine example

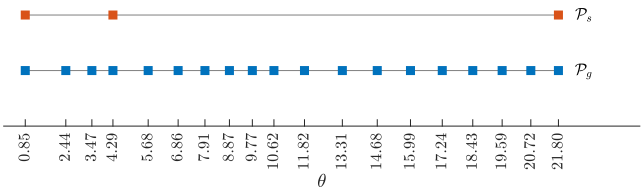

Figure 5. Grids of points used in the wind turbine PWA-LPV models

bending. The anti-windup (AW) scheme ensures a suitable transition from low- to high-speed operation (Inthamoussou et al., 2014).

The proposed methodology is employed to design the pitch controller. The set of linear models were obtained from the linearization tool included in OpenFast considering only the degrees of freedom (DOF) involved in the input-output map corresponding to the control scheme, *i.e.*, drive-train rotational-flexibility, generator, and 1st fore-aft tower bending-mode. The model was linearized at 19 wind speeds v_{op} between 11.4 and 25 m/s, assuming the turbine works at rated conditions ($\Omega_g = 1173.7$ rpm and $T_{g,\text{ref}} = 43.09$ kNm). As the wind speed experienced by the turbine rotor is not available for the controller and there exists a one-to-one mapping between v_{op} and β_{op}, the models are parameterized by the latter variable. Hence, the resulting PWA-LPV controller uses β_{op} as scheduling parameter θ. The grid of points θ^i used in the definition of the PWA-LPV model (1) is shown in Figure 5 (blue square markers). Notice that in this case, the points in $\mathcal{P}_g$ are not equally spaced as they result from computing the operating point associated to each v_{op}.

The design setup defining the augmented plant $G(\theta)$ is shown in Figure 6, where

$$M(s) = \begin{bmatrix} 1/s & 0 \\ 0 & 5/(s^2 + 0.5s + 2.1^2) \end{bmatrix}, \quad W_e(s) = 0.01,$$

$$W_u(s) = 0.1\frac{s+1}{s/100+1}, \qquad W_z(s) = 1.$$

Using the multi-channel framework introduced in (Scherer et al., 1997), the specifications are expressed as minimizing

$$\gamma = \|T_1\|_2 + \|T_2\|_2, \tag{27}$$

where T_1 denotes the mapping from w to z_2, T_2 from Ω_{ref}

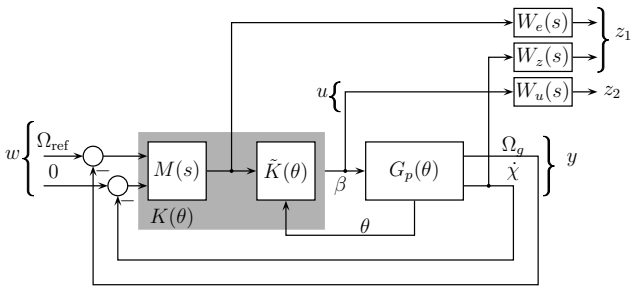

Figure 6. Control design setup for the wind turbine example

to z_1, and $\|\cdot\|_2$ the $\mathcal{L}_2$ induced norm. The first term in (27) aims to limit the pitch activity in high frequencies to increase robustness. The second term seeks to minimize the generator speed error in low frequencies and attenuate vibrations in the fore-after bending mode. The application of Algorithm 1 with a tolerance $\epsilon = 10^{-4}$ to the augmented plant indicates that the set $\mathcal{P}_g$ can be reduced to the set $\mathcal{P}_s$ shown in Figure 5 (red markers) with only 3 points. This is a significant complexity reduction that is difficult to observe directly from the set of linear models. With these grids, two controllers were designed: $K_f(\theta)$ based on the complete grid $\mathcal{P}_g$ and $K_s(\theta)$ based on the smaller grid $\mathcal{P}_s$.

The controllers were evaluated under two scenarios taken from the IEC 61400-1 standard using OpenFast and the model with all DOFs corresponding to the 5 MW NREL wind turbine. The first corresponds to a design load case (DLC) 1.1, the wind speed is a normal turbulent model profile created with TurbSim for a mean wind speed of 17 m/s and turbulence characteristics B. The simulation results are presented in Figure 7. The wind speed profile can be seen in the bottom plot. In the other plots, the generator speed Ω_g, the pitch angle β and the tower fore-aft displacement χ are presented. The gray lines correspond to the closed-loop responses with the gain-scheduled PI controller K_b introduced in (Jonkman et al., 2009), the blue lines to the controller K_f and the red lines to the controller K_s. It can be seen that the latter controller produces a quite similar response with a much simpler controller, both from the design and implementation standpoints.

The second scenario considered was a DLC 2.3 using an extreme operating gust (EOG) wind speed profile. The closed-loop responses for the three controllers can be observed in Figure 8. The wind speed in the x-direction is shown in the bottom plot. As in the previous figure, the gray lines corresponds to the responses for the baseline controller K_b, the blue lines for K_f and the red lines for K_s. Also in this case, both LPV controllers lead to similar behaviors, indicating that the proposed procedure is able to simplify the controller implementation and design without a significant performance degradation.

It is worthy to remark that the use of the proposed methodology does not require a control-oriented model for the controller design. This is performed directly from the models

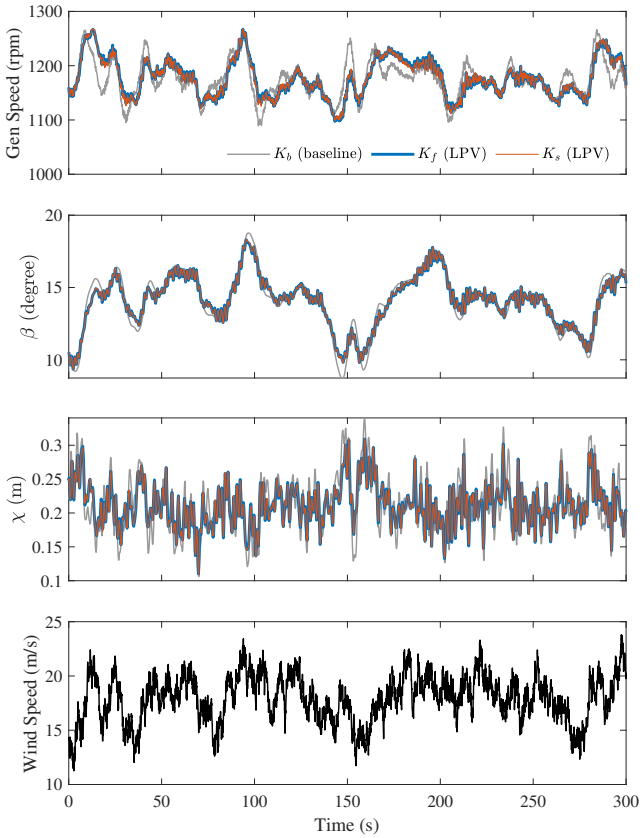

Figure 7. Closed-loop simulations for three controllers using a fully detailed 5 MW NREL onshore wind turbine model under a DLC 1.1 scenario.

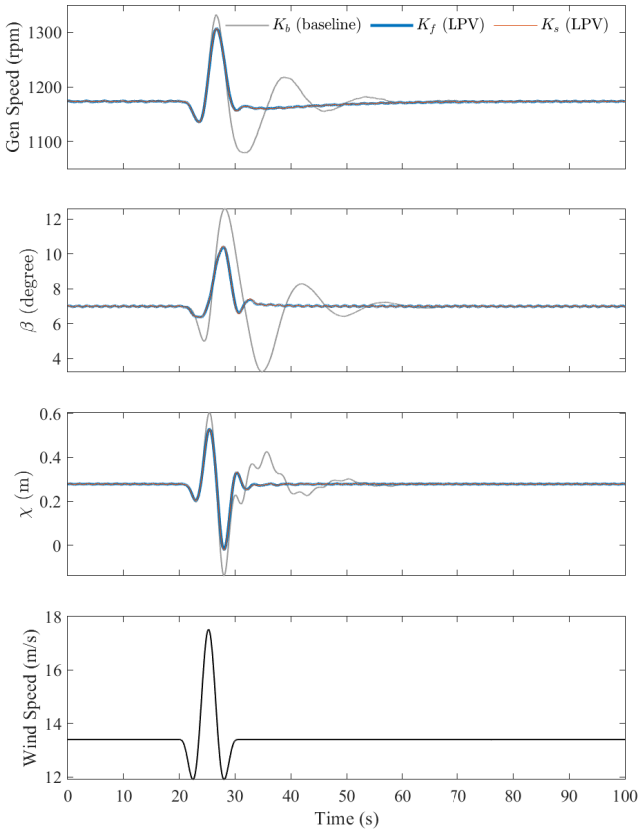

Figure 8. Closed-loop simulations for three controllers using a fully detailed 5 MW NREL onshore wind turbine model in under a DLC 2.3 scenario. The blue (K_f) and red (K_s) lines are practically coincident.

produced by OpenFast. This avoids a time consuming model validation needed in other standard control designs.

6 Conclusions

This article presents a novel methodology to reduce the design and implementation complexity of PWA-LPV controllers. This type of controllers is an attractive option to deal with the control of complex nonlinear systems, especially for those based on look-up-tables or too complex mathematical expressions. PWA-LPV systems are described by a set of local linear models and the number of these models affect the controller design and implementation. The proposed methodology allows us to find a smaller set of local models that ensures a suitable closed-loop performance. A controller based on a PWA-LPV approximation cannot guarantee closed-loop stability of the nonlinear model from which it originates. However, the proper selection of the local models along with robustness constraints, easily included in the design, allows us to find a controller with a reasonable implementation and design complexity that suitably works in practice. This point is clearly shown with the wind turbine control application presented in the last section, in which the methodology is able to find a much simpler GS controller achieving a similar closed-loop response to the one based on a larger set of local models.

Acknowledgements

The authors acknowledge the financial support of ITBA and CONICET.

References

Abbas, H. S., Tóth, R., Petreczky, M., Meskin, N., Mohammadpour Velni, J., & Koelewijn, P. J. (2021). LPV modeling of nonlinear systems: A multi-path feedback linearization approach. *International Journal of Robust and Nonlinear Control*, *31*, 9436–9465.

Apkarian, P., & Adams, R. (1998). Advanced gain-scheduling techniques for uncertain systems. *IEEE Transactions on Control Systems Technology*, *6*, 21–32.

Apkarian, P., Gahinet, P., & Becker, G. S. (1995). Self-scheduled H_∞ control of linear parameter-varying systems: a design example. *Automatica*, *31*, 1251–1261.

Åström, K. J., & Wittenmark, B. (2008). *Adaptive control*. (2nd ed.). Mineola, USA: Dover Publications.

Becker, G. S., & Packard, A. (1994). Robust performance of linear parametrically varying systems using

parametrically-dependent linear feedback. *Systems & Control Letters*, *23*, 205–215.

Belikov, J., Kotta, U., & Tõnso, M. (2014). Comparison of LPV an nonlinear system theory: a realization problem. *Systems & Control Letters*, *64*, 72–78.

de Berg, M., Cheong, O., van Kreveld, M., & Overmars, M. (2008). *Computational geometry: Algorithms and applications*. Heidelberg, Germany: Springer Berlin Heidelberg.

Bianchi, F. D., & Sánchez-Peña, R. S. (2022). A method for reducing implementation complexity in gain-scheduled LPV controllers. *Automatica*, *146*, 110588.

Häberle, V., Fisher, M. W., Prieto-Araujo, E., & Dörfler, F. (2022). Control design of dynamic virtual power plants: An adaptive divide-and-conquer approach. *IEEE Transactions on Power Systems*, *37*, 4040–4053.

Hoffmann, C., Hashemi, S. M., Abbas, H. S., & Werner, H. (2014). Synthesis of LPV controllers with low implementation complexity based on a reduced parameter set. *IEEE Transactions on Control Systems Technology*, *22*, 2393–2398.

Hoffmann, C., & Werner, H. (2015). A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations. *IEEE Transactions on Control Systems Technology*, *23*, 416–433.

Inthamoussou, F. A., Bianchi, F. D., De Battista, H., & Mantz, R. (2014). LPV wind turbine control with anti-windup features covering the complete wind speed range. *IEEE Transactions on Energy Conversion*, *29*, 256–266.

Jonkman, J., Butterfield, S. B., Musial, W., & Scott, G. (2009). *Definition of a 5-MW reference wind turbine for offshore system development*. Technical Report NREL/TP-500-38060 NREL Golden, Colorado, USA.

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*, 105360.

Kwiatkowski, A., & Werner, H. (2008). PCA-based parameter set mappings for LPV models with fewer parameters and less overbounding. *IEEE Transactions on Control Systems Technology*, *16*, 781–788.

Lim, S., & How, J. P. (2002). Analysis of linear parameter-varying systems using a non-smooth dissipative systems framework. *International Journal of Robust and Nonlinear Control*, *12*, 1067–1092.

Lim, S., & How, J. P. (2003). Modeling and H_∞ control for switched linear parameter-varying missile autopilot. *IEEE Transactions on Control Systems Technology*, *11*, 830–838.

Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proc. of the CACSD Conference*. Taipei, Taiwan.

Mohammadpour, J., & Scherer, C. (Eds.) (2012). *Control of linear parameter varying systems with applications*. New York, USA: Springer.

Morera-Torres, E., Ocampo-Martinez, C., & Bianchi, F. (2022). Experimental modelling and optimal torque vectoring control for 4WD vehicles. *IEEE Transactions on Vehicular Technology*, *71*, 4922–4932.

OpenFast 3.4 (2023). `https://github.com/OpenFAST/openfast` (accessed April 1, 2023).

Rotondo, D. (2018). *Advances in gain-scheduling and fault tolerant control techniques*. Cham, Switzerland: Springer International Publishing.

Sadeghzadeh, A., Sharif, B., & Tóth, R. (2020). Affine linear parameter-varying embedding of non-linear models with improved accuracy and minimal overbounding. *IET Control Theory & Applications*, *14*, 3363–3373.

Sadeghzadeh, A., & Tóth, R. (2023). Improved embedding of nonlinear systems in linear parameter-varying models with polynomial dependence. *IEEE Transactions on Control Systems Technology*, *31*, 70–82.

Sánchez-Peña, R. S., & Bianchi, F. D. (2012). Model selection: from LTI to switched-LPV. In *Proc. of the American Control Conference* (pp. 1561–1566). volume 1.

Scherer, C. W. (2001). LPV control and full block multipliers. *Automatica*, *37*, 361–375.

Scherer, C. W., Gahinet, P., & Chilali, M. (1997). Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, *42*, 896–911.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization Methods and Software*, *11*, 625–653.

Wu, F., Yang, X. H., Packard, A., & Becker, G. S. (1996). Induced l_2-norm control for LPV systems with bounded parameter variation rates. *International Journal of Robust and Nonlinear Control*, *6*, 983–998.