A method for reducing implementation complexity in 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, "A method for reducing implementation complexity in linear parameter-varying controllers", *Automatica*, vol. 146, pp. 110588, 2022.

Final version available at:

URL: https://www.sciencedirect.com/science/article/pii/S0005109822004502

DOI: 10.1016/j.automatica.2022.110588

BibTex:

```
@Article{Bianchi2022,
    Title   = {A method for reducing implementation complexity in linear
       parameter-varying controllers},
    Author  = {Fernando D. Bianchi, and Ricardo S. Sánchez-Peña},
    Journal = {Automatica},
    Year    = {2022},
    Number  = {},
    Pages   = {110588},
    Volume  = {146},
    Doi     = {10.1016/j.automatica.2022.110588}
}
```

A method for reducing implementation complexity in 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

Gain-scheduling is a popular control technique to deal with nonlinear and time varying systems. The linear parameter-varying (LPV) system approach offers systematic tools to design gain-scheduled controllers. However, the implementation of these controllers might demand complex mathematical operations to be performed in real-time. This limits the hardware and the applications in which LPV controllers can be used. In this article, we analyze these limitations and propose a design methodology to reduce the implementation complexity of gain-scheduled LPV controllers. The methodology is illustrated with a nonlinear bicycle model used in electric vehicle control.

Key words: Linear parameter varying (LPV) systems, gain scheduling, implementation complexity.

1 Introduction

Linear parameter-varying (LPV) systems have become a popular framework for modelling nonlinear and time-varying systems. In this framework, the system dynamics is described as a linear model depending on a set of time-varying parameters. The popularity can be explained by the extension of well-known linear optimal tools to systematically design gain-scheduled controllers (Becker & Packard, 1994; Apkarian, Gahinet & Becker, 1995). Throughout the years, several improvements have been proposed, mainly focused on enhancing performance and considering more general LPV models (Wu, Yang, Packard & Becker, 1996; Apkarian & Adams, 1998; Scherer, 2001; Mohammadpour & Scherer, 2012). Unfortunately, these improvements also lead to more complex controller implementations.

In general, implementation complexity of LPV controllers is given by the particular synthesis procedure, the LPV plant and the control specifications. In the implementation of LPV controllers, a set of scheduling parameters must be measured in real-time and used to update the controller coefficients. Once these coefficients are updated, the output is obtained similarly to any linear time-invariant (LTI) controller. As in LTI controllers, the order increases the implementation complexity due to the number of variables and mathematical operations needed to compute the control action. Using systematic design tools, the controller order can be large and is given mainly by the plant order and the specifications. To solve this issue, several model order reduction techniques have been extended to LPV models (Wood, Goddard & Glover, 1996; Beck, 2006).

Another source of implementation complexity is the update of the controller coefficients. The complexity of these computations depends on the controller design procedure and the particular parameter dependence of the plant. For instance, synthesis procedures based on parameter dependent Lyapunov functions have been proposed to improve closed-loop performance (Apkarian & Adams, 1998; Wu et al., 1996). However, the resulting controllers require online matrix operations that could be highly demanding for low computation-power systems.

Two approaches can be identified to simplify the update of the controller coefficients. As the mathematical expression of the controller coefficients is in general inherited from the plant, one approach consists in simplifying the plant before designing the controller. In this line, Kwiatkowski & Werner (2008); Hoffmann, Hashemi, Abbas & Werner (2014) propose the use of principal com-

⋆ 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).

ponent analysis (PCA) to reduce the parameter space in affine and linear fractional LPV plants. This approach is not always the best strategy with regard to closed-loop performance. Sometimes, controllers with a parameter dependence much simpler than the plant or even an LTI control can achieve satisfactory performance (Sánchez Peña & Bianchi, 2012). In these cases, it might be more suitable the second approach in which the controller is directly designed with a given parameter dependence. In this line, de Hillerin, Scorletti & Fromion (2011) pose the search of the best parameter dependence as a structure simplification of a linear fractional representation. Other two works with this approach are (Petersson & Lofberg, 2014; Gahinet & Apkarian, 2013) in which a controller with a pre-established structure is designed from a set of linear approximations of the plant using nonlinear optimization.

In this article, we propose an implementation-oriented design based on the classical synthesis procedure introduced in (Apkarian & Adams, 1998). The aim is to find a controller parameter dependence that reaches a compromise between performance and implementation complexity. Imposing a parameter dependence reduces the set of stabilizing controllers and might affect the closed-loop performance. Therefore, the proposed methodology not only provides design tools to ease the controller implementation but also guidelines to estimate the simplification effects on the closed-loop performance. Unlike available controller simplifications, the proposed approach can be applied to rather general LPV plants without model approximations.

This article is organized as follows. Next section provides some necessary background on LPV control, Section 3 presents the main contribution, and Section 4 illustrates it with an example. Finally, some concluding remarks are drawn in Section 5.

Notation: The following notation will be used

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

For a symmetric matrix $Q \in \mathbb{R}^{n\times n}$, $\lambda_i(Q)$ stands for the i-th eigenvalue of Q, ordered as $\underline{\lambda} = \lambda_1 \leq \cdots \leq \lambda_n = \bar{\lambda}$. For a matrix $Q \in \mathbb{R}^{n\times m}$, $\sigma_i(Q)$ denotes the i-th singular value of Q and $\sigma_1(Q) = \bar{\sigma}(Q)$ the maximum. 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 Background: LPV control design

Consider the following LPV system

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

where $x \in \mathbb{R}^{n_s}$ 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 ρ is assumed taking values in a compact set $\mathcal{P} \subset \mathbb{R}^{n_p}$ and its derivative $\dot{\rho}$ in $\mathcal{P}_d \subset \mathbb{R}^{n_p}$. It is common to take these sets as hypercubes corresponding to the variation ranges of ρ and $\dot{\rho}$.

The parameter dependent matrices in (1) are

$$\begin{bmatrix} A(\rho) & B_1(\rho) \\ C_1(\rho) & D_{11}(\rho) \end{bmatrix} = \begin{bmatrix} A_0 & B_{1,0} \\ C_{1,0} & D_{11,0} \end{bmatrix} + \sum_{i=1}^{n_f} f_i(\rho) \begin{bmatrix} A_i & B_{1,i} \\ C_{1,i} & D_{11,i} \end{bmatrix}, \quad (2)$$

where the parameter functions $f_i : \mathbb{R}^{n_p} \to \mathbb{R}^{n_f}$ are known and bounded for all $\rho \in \mathcal{P}$. Notice that a large number of nonlinear and time-varying systems can be described by the LPV representation (1)-(2), see *e.g.* (Belikov, Kotta & Tõnso, 2014; Tóth, 2010).

The synthesis problem consists in finding an LPV controller

$$K(\rho)\begin{cases} \dot{x}_c = A_c(\rho)x_c + B_c(\rho)y, \\ u = C_c(\rho)x_c + D_c(\rho)y, \end{cases} \quad (3)$$

that ensures closed-loop quadratic stability and

$$\|z\|_2 \leq \gamma\|w\|_2, \quad \forall\, \rho \in \mathcal{P}, \text{ and } \dot{\rho} \in \mathcal{P}_d. \quad (4)$$

Using the linearizing change of variables introduced in (Scherer, Gahinet & Chilali, 1997), this design problem can be cast as solving a convex optimization problem that minimizes γ subject to the following linear matrix inequalities (LMIs)

$$\Pi(\mathbf{X}(\rho), \mathbf{Y}(\rho), \hat{\mathbf{A}}(\rho), \hat{\mathbf{B}}(\rho), \hat{\mathbf{C}}(\rho), \hat{\mathbf{D}}(\rho), \gamma) < 0, \quad (5)$$

$$\begin{bmatrix} \mathbf{X}(\rho) & I_{n_s} \\ I_{n_s} & \mathbf{Y}(\rho) \end{bmatrix} > 0, \quad (6)$$

where $\Pi(\cdot)$ is given in (7), and the matrix functions $\mathbf{X}(\rho) = \mathbf{X}(\rho)^T$, $\mathbf{Y}(\rho) = \mathbf{Y}(\rho)^T$, $\hat{\mathbf{A}}(\rho)$, $\hat{\mathbf{B}}(\rho)$, $\hat{\mathbf{C}}(\rho)$, $\hat{\mathbf{D}}(\rho)$ and the scalar $\gamma > 0$ are decision variables to be found (Apkarian & Adams, 1998).

$$\Pi(\mathbf{X}(\rho),\mathbf{Y}(\rho),\hat{\mathbf{A}}(\rho),\hat{\mathbf{B}}(\rho),\hat{\mathbf{C}}(\rho),\hat{\mathbf{D}}(\rho),\gamma) =$$
$$\begin{bmatrix} -\dot{\mathbf{Y}}(\rho,\dot\rho) + A(\rho)\mathbf{Y}(\rho) + B_2\hat{\mathbf{C}}(\rho) + (\star) & \star & \star & \star \\ \hat{\mathbf{A}}(\rho) + (A(\rho) + B_2\hat{\mathbf{D}}(\rho)C_2)^T & \dot{\mathbf{X}}(\rho,\dot\rho) + \mathbf{X}(\rho)A(\rho) + \hat{\mathbf{B}}(\rho)C_2 + (\star) & \star & \star \\ (B_1(\rho) + B_2\hat{\mathbf{D}}(\rho)D_{21})^T & (\mathbf{X}(\rho)B_1(\rho) + \hat{\mathbf{B}}(\rho)D_{21})^T & -\gamma I_{n_w} & \star \\ C_1(\rho)\mathbf{Y}(\rho) + D_{12}\hat{\mathbf{C}}(\rho) & C_1(\rho) + D_{12}\hat{\mathbf{D}}(\rho)C_2 & D_{11}(\rho) + D_{12}\hat{\mathbf{D}}(\rho)D_{21} & -\gamma I_{n_z} \end{bmatrix} \quad (7)$$

In order to reduce the optimization problem to a finite-dimensional one, the auxiliary functions are taken as

$$\begin{bmatrix} \hat{\mathbf{A}}(\rho) & \hat{\mathbf{B}}(\rho) \\ \hat{\mathbf{C}}(\rho) & \hat{\mathbf{D}}(\rho) \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{A}}_0 & \hat{\mathbf{B}}_0 \\ \hat{\mathbf{C}}_0 & \hat{\mathbf{D}}_0 \end{bmatrix} + \sum_{i=1}^{n_g} g_i(\rho) \begin{bmatrix} \hat{\mathbf{A}}_i & \hat{\mathbf{B}}_i \\ \hat{\mathbf{C}}_i & \hat{\mathbf{D}}_i \end{bmatrix}, \quad (8)$$

and the Lyapunov matrix functions as

$$\mathbf{X}(\rho) = \mathbf{X}_0 + \sum_{i=1}^{n_h} h_i(\rho)\mathbf{X}_i, \quad \mathbf{Y}(\rho) = \mathbf{Y}_0 + \sum_{i=1}^{n_l} \ell_i(\rho)\mathbf{Y}_i,$$

where $g_i : \mathbb{R}^{n_p} \to \mathbb{R}^{n_g}$, $h_i : \mathbb{R}^{n_p} \to \mathbb{R}^{n_h}$, and $\ell_i : \mathbb{R}^{n_p} \to \mathbb{R}^{n_l}$ are basis functions (Apkarian & Adams, 1998). There is no rule to select the basis functions, but it is common to copy the functions $f_i(\cdot)$. Nevertheless, notice that the choice of these functions might affect the closed-loop performance as this limits the set in which the Lyapunov functions and the controller matrices are sought.

The optimization problem still consists of an infinite number of LMIs unless $f_i(\rho) = g_i(\rho) = h_i(\rho) = \ell_i(\rho) = \rho_i$ for all $i = 1, \ldots, n_p$ (affine case). In this case, it is sufficient to check the LMIs (5) and (6) at the vertices of $\mathcal{P}$ and $\mathcal{P}_d$ (Apkarian et al., 1995). A common practice for the rest of cases consists in selecting a grid of points in the parameter set $\mathcal{P}$, solving the optimization problem with the LMIs evaluated at these points and then checking if $\mathbf{X}(\rho)$ and $\mathbf{Y}(\rho)$ and the auxiliary variables (8) satisfy the LMIs (5) and (6) evaluated in a denser grid (Apkarian & Adams, 1998; Wu et al., 1996).

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

$$A_c(\rho) = N^{-1}(\rho)(\hat{\mathbf{A}}(\rho) - \mathbf{X}(\rho)(A(\rho) - B_2\hat{\mathbf{D}}(\rho)C_2)\mathbf{Y}(\rho) - \hat{\mathbf{B}}(\rho)C_2\mathbf{Y}(\rho) - \mathbf{X}(\rho)B_2\hat{\mathbf{C}}(\rho))M^{-T}(\rho), \quad (9)$$
$$B_c(\rho) = N^{-1}(\rho)(\hat{\mathbf{B}}(\rho) - \mathbf{X}(\rho)B_2\hat{\mathbf{D}}(\rho)), \quad (10)$$
$$C_c(\rho) = (\hat{\mathbf{C}}(\rho) - \hat{\mathbf{D}}(\rho)C_2\mathbf{Y}(\rho))M^{-T}(\rho), \quad (11)$$
$$D_c(\rho) = \hat{\mathbf{D}}(\rho), \quad (12)$$

where $M(\rho)$ and $N(\rho)$ are selected to satisfy

$$I - \mathbf{X}(\rho)\mathbf{Y}(\rho) = N(\rho)M^T(\rho), \quad \forall\, \rho \in \mathcal{P}. \quad (13)$$

In order to prevent $A_c(\rho)$ from depending on $\dot\rho$ in those cases in which $\dot\rho \neq 0$, it is assumed that one of the matrix functions $\mathbf{X}(\rho)$ or $\mathbf{Y}(\rho)$ is parameter independent. Please see more details in (Apkarian & Adams, 1998).

3 Low complexity LPV control design

In this section, we propose two procedures to design LPV controllers with a simpler parameter dependence. As indicated previously, the additional implementation complexity of an LPV controller, with respect to LTI controllers, is in computing matrices (9)-(12) at each step time for a measured ρ. Therefore, the objective here is to simplify these expressions in order to make them more suitable for real-time implementation.

Observing the controller matrix expressions (9)-(12), the most complex mathematical operations to be performed at each step time are the matrix inversions of $M(\rho)$ and $N(\rho)$ and the factorization (13). All these operations can be computed off-line if $\mathbf{X}(\cdot)$ and $\mathbf{Y}(\cdot)$ are taken constant. Parameter dependent Lyapunov functions allows considering bounds on $\dot\rho$ and thus reduce conservatism, but this also limits the hardware in which these controllers can be implemented. Next, $\mathbf{X}(\cdot)$ and $\mathbf{Y}(\cdot)$ are considered as constant matrices in order to present the simplest controller implementation. Nevertheless, the proposed ideas can be extended to the more general case. Comments on this regard can be found in Remark 1 at the end of this section.

Using $\mathbf{X}(\rho) = \mathbf{X}_0$ and $\mathbf{Y}(\rho) = \mathbf{Y}_0$ and the auxiliary variables (8) in (9)-(12), the controller matrices to be updated at each step time can be written as

$$\begin{bmatrix} A_c(\rho) & B_c(\rho) \\ C_c(\rho) & D_c(\rho) \end{bmatrix} = \begin{bmatrix} A_{c,0} & B_{c,0} \\ C_{c,0} & D_{c,0} \end{bmatrix} + \sum_{i=1}^{n_g} g_i(\rho) \begin{bmatrix} A_{c,i} & B_{c,i} \\ C_{c,i} & D_{c,i} \end{bmatrix} - \sum_{i=1}^{n_f} f_i(\rho) \begin{bmatrix} N^{-1}\mathbf{X}_0 A_i \mathbf{Y}_0 M^{-T} & 0 \\ 0 & 0 \end{bmatrix}, \quad (14)$$

where N and M are now also constant matrices. This is a linear combination of constant pre-computed matrices with parameter dependent weights.

As mentioned previously, it is common to take $g_i(\cdot) = f_i(\cdot)$ $(i = 1, \ldots, n_f)$. From the implementation perspective, it might be interesting to limit the $g_i(\cdot)$'s to a subset of the $f_i(\cdot)$'s. This will reduce the number of auxiliary matrices in (8), which implies less decision variables to be found during the design. Less $g_i(\cdot)$ will also reduce the number of terms in (14) and thus the matrices to be stored in the embedded system and the number of mathematical operations to be performed on-line. Moreover, including only those functions $f_i(\cdot)$'s depending on a subset of parameters, the number of sensors may be reduced. However, it can be observed in (14) that independently of the selected $g_i(\cdot)$'s, the controller matrix $A_c(\cdot)$ includes terms with $f_i(\cdot)$'s. Therefore, to ensure a lower complexity parameter dependence, the last term in (14) should be removed.

To this end, let the system matrix be decomposed as

$$A(\rho) = A_{\mathrm{s}}(\rho) + A_{\mathrm{r}}(\rho) \tag{15}$$

where

$$A_{\mathrm{s}}(\rho) = A_0 + \sum_{i\in\mathcal{I}} f_i(\rho)A_i, \quad A_{\mathrm{r}}(\rho) = \sum_{i\in\mathcal{J}} f_i(\rho)A_i.$$

The index set is split into $\mathcal{I}$ and $\mathcal{J}$ such that

$$\mathcal{I} \cap \mathcal{J} = \emptyset, \qquad \mathcal{I} \cup \mathcal{J} = \{1, 2, \ldots, n_f\},$$

and the controller basis functions are taken as

$$g_j(\cdot) = f_{i_j}(\cdot), \quad i_j \in \mathcal{I} = \{i_1, \ldots, i_{n_g}\}. \tag{16}$$

The substitution of (15) in (9) allows us to express the auxiliary function $\hat{\mathbf{A}}(\cdot)$ as

$$\hat{\mathbf{A}}(\rho) = \check{\mathbf{A}}(\rho) + \mathbf{X}_0 A_{\mathrm{r}}(\rho)\mathbf{Y}_0,$$

with

$$\begin{aligned}\check{\mathbf{A}}(\rho) = NA_c(\rho)M^T + \mathbf{X}_0(A_{\mathrm{s}}(\rho) + B_2D_c(\rho)C_2)\mathbf{Y}_0 + \\ \mathbf{X}_0B_2C_c(\rho)M^T + NB_c(\rho)C_2\mathbf{Y}_0.\end{aligned}$$

Thus, $\check{\mathbf{A}}(\rho)$ is the matrix function to be found instead of $\hat{\mathbf{A}}(\rho)$, and the auxiliary functions (8) are substituted by

$$\begin{bmatrix}\check{\mathbf{A}}(\rho) & \hat{\mathbf{B}}(\rho)\\ \hat{\mathbf{C}}(\rho) & \hat{\mathbf{D}}(\rho)\end{bmatrix} = \begin{bmatrix}\check{\mathbf{A}}_0 & \hat{\mathbf{B}}_0\\ \hat{\mathbf{C}}_0 & \hat{\mathbf{D}}_0\end{bmatrix} + \sum_{i=1}^{n_g} g_i(\rho)\begin{bmatrix}\check{\mathbf{A}}_i & \hat{\mathbf{B}}_i\\ \hat{\mathbf{C}}_i & \hat{\mathbf{D}}_i\end{bmatrix}, \tag{17}$$

with the new decision variables $\check{\mathbf{A}}_i$ $(i = 0, \ldots, n_g)$. Then, the inequality (5) can be rewritten as

$$\begin{aligned}&\Pi(\mathbf{X}_0, \mathbf{Y}_0, \check{\mathbf{A}}(\rho), \hat{\mathbf{B}}(\rho), \hat{\mathbf{C}}(\rho), \hat{\mathbf{D}}(\rho), \gamma) + \\ &+ \underbrace{\begin{bmatrix} 0 & \star & \star & \star \\ \mathbf{X}_0 A_{\mathrm{r}}(\rho)\mathbf{Y}_0 & 0 & \star & \star \\ 0 & 0 & 0 & \star \\ 0 & 0 & 0 & 0\end{bmatrix}}_{\Gamma(\mathbf{X}_0,\mathbf{Y}_0)} < 0.\end{aligned} \tag{18}$$

Using the new auxiliary function $\check{\mathbf{A}}(\rho)$ instead of $\hat{\mathbf{A}}(\rho)$ as a decision variable in the expression (9)-(12), the controller matrices are

$$\begin{bmatrix}A_c(\rho) & B_c(\rho)\\ C_c(\rho) & D_c(\rho)\end{bmatrix} = \begin{bmatrix}A_{c,0} & B_{c,0}\\ C_{c,0} & D_{c,0}\end{bmatrix} + \sum_{i=1}^{n_g} g_i(\rho)\begin{bmatrix}A_{c,i} & B_{c,i}\\ C_{c,i} & D_{c,i}\end{bmatrix}, \tag{19}$$

hence, eliminating the last term in (14). Unfortunately (18) is not an LMI and the synthesis problem becomes nonconvex. Two procedures are proposed to approximately solve the controller design using convex optimization.

Using the fact (Horn & Johnson, 2013) that for a matrix

$$P = \begin{bmatrix}0 & Q\\ Q^T & 0\end{bmatrix}, \text{ with } Q \in \mathbb{R}^{n\times n},$$

the eigenvalues are $\lambda_j(Q) = -\sigma_j(Q)$ and $\lambda_{2n-j+1}(Q) = \sigma_j(Q)$, $j = 1, \ldots, n$, the following holds

$$\begin{aligned}\underline{\lambda}(\Gamma(\mathbf{X}_0, \mathbf{Y}_0)) &= -\bar{\sigma}(\mathbf{X}_0 A_{\mathrm{r}}(\rho)\mathbf{Y}_0),\\ \bar{\lambda}(\Gamma(\mathbf{X}_0, \mathbf{Y}_0)) &= \bar{\sigma}(\mathbf{X}_0 A_{\mathrm{r}}(\rho)\mathbf{Y}_0),\end{aligned}$$

and thus (Horn & Johnson, 2013)

$$\begin{aligned}\bar{\lambda}(\Pi) - \bar{\sigma}(\mathbf{X}_0 A_{\mathrm{r}}(\rho)\mathbf{Y}_0) &\le \bar{\lambda}(\Pi + \Gamma)\\ &\le \bar{\lambda}(\Pi) + \bar{\sigma}(\mathbf{X}_0 A_{\mathrm{r}}(\rho)\mathbf{Y}_0),\end{aligned}$$

where the arguments of Π and Γ have been dropped to ease the notation. The matrix $A_{\mathrm{r}}(\rho)$ is usually sparse, with rank $n_a \le n_s$. Therefore, applying the singular value decomposition

$$A_{\mathrm{r}}(\rho) = \begin{bmatrix}U_1(\rho) & U_2(\rho)\end{bmatrix}\begin{bmatrix}\Sigma(\rho) & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}V_1^T(\rho)\\ V_2^T(\rho)\end{bmatrix} = E(\rho)F^T(\rho),$$

where $E(\rho) = U_1(\rho)\Sigma(\rho)^{1/2}$, $F(\rho) = V_1(\rho)\Sigma(\rho)^{1/2}$, and $\Sigma(\rho) \in \mathbb{R}^{n_a}$, the following inequality holds

$$\bar{\sigma}(\mathbf{X}_0 A_{\mathrm{r}}(\rho)\mathbf{Y}_0) \le \bar{\sigma}(\mathbf{X}_0 E(\rho))\bar{\sigma}(\mathbf{Y}_0 F(\rho)).$$

With the previous results, the following design procedure can be stated. The aim is to find a controller that minimizes ε_x and ε_y, where

$$\bar{\sigma}(\mathbf{X}_0 E(\rho)) \leq \varepsilon_x, \qquad \bar{\sigma}(\mathbf{Y}_0 F(\rho)) \leq \varepsilon_y$$

and satisfies (6) and (18).

Procedure 1 *Select a desired index set* $\mathcal{J}$ *and basis functions* $g_i(\cdot)$, $i = 1, \ldots, n_g$, *according to* (16). *Then, find symmetric matrices* $\mathbf{X}_0$ *and* $\mathbf{Y}_0$, *matrices* $\check{\mathbf{A}}_i$, $\hat{\mathbf{B}}_i$, $\hat{\mathbf{C}}_i$ *and* $\hat{\mathbf{D}}_i$ *according to* (17), *and positive scalars* γ, ε_x ε_y *that minimize*

$$L_1 = \gamma + q(\varepsilon_x + \varepsilon_y)$$

subject to:

$$\Pi(\mathbf{X}_0, \mathbf{Y}_0, \check{\mathbf{A}}(\rho), \hat{\mathbf{B}}(\rho), \hat{\mathbf{C}}(\rho), \hat{\mathbf{D}}(\rho), \gamma) < 0,$$

$$\begin{bmatrix} I_{n_s} & \mathbf{X}_0 E(\rho) \\ E^T(\rho)\mathbf{X}_0 & I_{n_a}\varepsilon_x \end{bmatrix} > 0, \quad \begin{bmatrix} I_{n_s} & \mathbf{Y}_0 F(\rho) \\ F^T(\rho)\mathbf{Y}_0 & I_{n_a}\varepsilon_y \end{bmatrix} > 0,$$

$$\begin{bmatrix} \mathbf{X}_0 & I_{n_s} \\ I_{n_s} & \mathbf{Y}_0 \end{bmatrix} > 0,$$

for all ρ *in a grid of points* $\mathcal{P}_g \subset \mathcal{P}$ *and a given weight* $q > 0$.
Then, check if there exists a γ *such that, with the obtained variables* $\mathbf{X}_0$, $\mathbf{Y}_0$, $\check{\mathbf{A}}_i$, $\hat{\mathbf{B}}_i$, $\hat{\mathbf{C}}_i$ *and* $\hat{\mathbf{D}}_i$, *the conditions* (6) *and* (18) *are satisfied, otherwise increase* q *and repeat the design.*

Procedure 1 seeks to exploit the fact that $A_r(\rho)$ can be sparse. Thus, with a sufficiently large q, $\bar{\sigma}(\mathbf{X}_0 A_r(\rho)\mathbf{Y}_0)$ will be small and $\bar{\lambda}(\Pi)$ will be close to $\bar{\lambda}(\Pi + \Gamma)$. However, technically $\bar{\lambda}(\Pi) \approx \bar{\lambda}(\Pi + \Gamma)$ does not imply that (18) holds. For this reason, as indicated in the last part of Procedure 1, it is necessary to check if the computed matrices satisfy the inequalities and repeat the optimization with a larger penalization over ε_x and ε_y, if needed. Even if the found matrices satisfy (6) and (18), the performance level γ will be probably lower that the one obtained during the last verification. Hence, it can be assumed that Procedure 1 will produce an optimistic design.

In order to avoid the last iterative step at the expense of introducing certain conservatism, consider the matrix $\Psi(\mathbf{X}(\rho), \mathbf{Y}(\rho), \check{\mathbf{A}}(\rho), \hat{\mathbf{B}}(\rho), \hat{\mathbf{C}}(\rho), \hat{\mathbf{D}}(\rho), \gamma)$ given in (20), which is obtained by adding the positive semi-definite matrix

$$\Lambda(\mathbf{X}_0, \mathbf{Y}_0) = \begin{bmatrix} \mathbf{Y}_0 F(\rho) F^T(\rho) \mathbf{Y}_0 & \star & \star & \star \\ 0 & \mathbf{X}_0 E(\rho) E^T(\rho) \mathbf{X}_0 & \star & \star \\ 0 & 0 & 0 & \star \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

to $\Pi + \Gamma$ and taking into account that $\Pi + \Gamma + \Lambda$ is the Schur complement of I_{n_a} in Ψ. Then, considering the following eigenvalue inequalities

$$\bar{\lambda}(\Pi + \Gamma) + \bar{\lambda}(\Lambda) \leq \bar{\lambda}(\Psi), \text{ and } \bar{\lambda}(\Lambda) \geq 0,$$

the second design procedure can be stated.

Procedure 2 *Select a desired index set* $\mathcal{J}$ *and basis functions* $g_i(\cdot)$, $i = 1, \ldots, n_g$, *according to* (16). *Then, find symmetric matrices* $\mathbf{X}_0$ *and* $\mathbf{Y}_0$, *matrices* $\check{\mathbf{A}}_i$, $\hat{\mathbf{B}}_i$, $\hat{\mathbf{C}}_i$ *and* $\hat{\mathbf{D}}_i$ *according to* (17), *and positive scalar* γ *that minimize*

$$L_2 = \gamma$$

subject to:

$$\Psi(\mathbf{X}_0, \mathbf{Y}_0, \check{\mathbf{A}}(\rho), \hat{\mathbf{B}}(\rho), \hat{\mathbf{C}}(\rho), \hat{\mathbf{D}}(\rho), \gamma) < 0,$$

$$\begin{bmatrix} \mathbf{X}_0 & I_{n_s} \\ I_{n_s} & \mathbf{Y}_0 \end{bmatrix} > 0,$$

for all ρ *in a grid of points* $\mathcal{P}_g \subset \mathcal{P}$.

In order to determine the most suitable decomposition (15), the following fact is used

$$\bar{\sigma}(\mathbf{X}_0 A_{\mathrm{r}}(\rho)\mathbf{Y}_0) \leq \bar{\sigma}(\mathbf{X}_0)\bar{\sigma}(A_{\mathrm{r}}(\rho))\bar{\sigma}(\mathbf{Y}_0)),$$

and, from the expression of $A_{\mathrm{r}}(\rho)$, that

$$\bar{\sigma}(A_{\mathrm{r}}(\rho)) \leq \sum_{i \in \mathcal{J}} |f_i(\rho)| \bar{\sigma}(A_i), \quad \forall \rho \in \mathcal{P}.$$

Hence, if

$$\bar{\sigma}(A_0) + \sum_{i \in \mathcal{I}} |f_i(\rho)| \bar{\sigma}(A_i) \gg \sum_{i \in \mathcal{J}} |f_i(\rho)| \bar{\sigma}(A_i), \qquad (21)$$

for all $\rho \in \mathcal{P}$, then it can be assumed that $\mathbf{X}_0 A(\rho)\mathbf{Y}_0 \approx \mathbf{X}_0 A_{\mathrm{s}}(\rho)\mathbf{Y}_0$ and the effect on the closed-loop performance, due to the simplification of the controller parameter dependence, may be negligible.

Remark 1 *In case of using parameter dependent Lyapunov functions, the proposed procedures are still useful to simplify the controller expressions,* e.g. *to reduce the number of measured parameters. The controller matrices*

$$\Psi(\mathbf{X}_0, \mathbf{Y}_0, \check{\mathbf{A}}(\rho), \hat{\mathbf{B}}(\rho), \hat{\mathbf{C}}(\rho), \hat{\mathbf{D}}(\rho), \gamma)$$
$$= \begin{bmatrix} A(\rho)\mathbf{Y}_0 + B_2\hat{\mathbf{C}}(\rho) + (\star) & \star & \star & \star & \star \\ \check{\mathbf{A}}(\rho) + (A(\rho) + B_2\hat{\mathbf{D}}(\rho)C_2)^T & \mathbf{X}_0 A(\rho) + \hat{\mathbf{B}}(\rho)C_2 + (\star) & \star & \star & \star \\ (B_1(\rho) + B_2\hat{\mathbf{D}}(\rho)D_{21})^T & (\mathbf{X}_0 B_1(\rho) + \hat{\mathbf{B}}(\rho)D_{21})^T & -\gamma I_{n_w} & \star & \star \\ C_1(\rho)\mathbf{Y}_0 + D_{12}\hat{\mathbf{C}}(\rho) & C_1(\rho) + D_{12}\hat{\mathbf{D}}(\rho)C_2 & D_{11}(\rho) + D_{12}\hat{\mathbf{D}}(\rho)D_{21} & -\gamma I_{n_z} & \star \\ F^T(\rho)\mathbf{Y}_0 & E^T(\rho)\mathbf{X}_0 & 0 & 0 & -I_{n_a} \end{bmatrix} \quad (20)$$

---

will consist of more terms, as a result of cross terms like $g_i(\rho)g_j(\rho)$ and those associated to $\mathbf{X}(\rho)$ or $\mathbf{Y}(\rho)$. However, in spite of these additional terms, the controller parameter dependence can be imposed with a suitable selection of the basis functions $g_i(\cdot)$, $h_i(\cdot)$ and $\ell_i(\cdot)$.

Remark 2 *The griding procedure can be avoided by considering each $f_i(\rho)$ as independent parameters θ_i taking values in an hypercube Θ. Thus, model* (1) *is affine in θ and it is sufficient to evaluate* (5) *and* (6) *at the vertices of the new parameter set Θ. However, as each new parameter is assumed to vary independently from the others, this overbounding of the parameter space might result in significant performance degradation. Moreover, it is also common to express the controller as a polytopic LPV system,* i.e.,

$$\begin{bmatrix} A_c(\theta) & B_c(\theta) \\ C_c(\theta) & D_c(\theta) \end{bmatrix} = \sum_{i=1}^{2^{n_p}} \alpha_i(\theta) \begin{bmatrix} A_{c,i} & B_{c,i} \\ C_{c,i} & D_{c,i} \end{bmatrix},$$

where

$$0 \le \alpha_i \le 1, \; \sum_{i=1}^{2^{n_p}} \alpha_i = 1, \; and \; \theta = \sum_{i=1}^{2^{n_p}} \alpha_i \theta^i \quad (22)$$

with θ^i the vertices of Θ. The convex decomposition (22) *is an additional computation step to be performed online. Besides, compared with the LPV description* (14), *more matrices must be stored for the real-time implementation. In (Kwiatkowski & Werner, 2008), a procedure based on PCA is proposed to reduce the number of θ_i and find a tighter Θ. However, there is no guarantee that the new parameter space will be less conservative. Instead, in the scheme proposed here, the controller complexity is independent of the number of points in the parameter grid used to design it. Moreover, there is no approximation of the LPV plant, only a reduction of the set in which the controller is sought.*

4 Example

In order to illustrate the proposed implementation-oriented design, we consider the following nonlinear system inspired by the two degree-of-freedom bicycle model used in (Morera-Torres, Ocampo-Martinez & Bianchi, 2022) to control electric vehicles

$$\dot{x}_1 = -\frac{C_f(x_1) + C_r}{m\,v} x_1 - \left(1 + \frac{(C_f(x_1) - C_r)\,l}{m\,v^2}\right) x_2,$$
$$\dot{x}_2 = \frac{(C_r - C_f(x_1))\,l}{J} x_1 - \frac{(C_r + C_f(x_1))\,l^2}{J\,v} x_2 + \frac{10000}{J} u,$$

where $m = 300\,\mathrm{kg}$, $J = 100\,\mathrm{kgm}^2$, $l = 0.8\,\mathrm{m}$, $C_r = 36000\,\mathrm{N/rad}$, $C_f(x_1) = 14000/(1100\,x_1^2 + 1)$ [N/rad], v an external parameter and $y = x_2$ the controlled output. This is a practical application that will be considered here as an illustrative example of this technique.

This nonlinear system can be cast as a quasi-LPV model:

$$G_p(\rho) \begin{cases} \dot{x}_p = \left(A_{p,0} + \sum_{i=1}^{5} f_i(\rho) A_{p,i}\right) x_p + B_p u, \\ y = C_p x_p \end{cases}$$

taking $x_p = [x_1 \;\; x_2]^T$, $\rho = [\rho_1 \;\; \rho_2]^T \in \mathcal{P} = [-0.2 \;\; 0.2] \times [5 \;\; 28]$, $\rho_1 = x_1$, $\rho_2 = v$, functions

$$f_1(\rho) = C_f(\rho_1), \quad f_2(\rho) = C_f(\rho_1)/\rho_2, \quad f_3(\rho) = 1/\rho_2,$$
$$f_4(\rho) = C_f(\rho_1)/\rho_2^2, \quad f_5(\rho) = 1/\rho_2^2.$$

and

$$A_{p,0} = \begin{bmatrix} 0 & -1 \\ \frac{C_r l}{J} & 0 \end{bmatrix}, \quad A_{p,1} = \begin{bmatrix} 0 & 0 \\ -\frac{l}{J} & 0 \end{bmatrix}, \quad A_{p,2} = \begin{bmatrix} -\frac{1}{m} & 0 \\ 0 & -\frac{l^2}{J} \end{bmatrix},$$
$$A_{p,3} = \begin{bmatrix} -\frac{C_r}{m} & 0 \\ 0 & -\frac{C_r l^2}{J} \end{bmatrix}, \quad A_{p,4} = \begin{bmatrix} 0 & -\frac{l}{m} \\ 0 & 0 \end{bmatrix}, \quad A_{p,5} = \begin{bmatrix} 0 & \frac{C_r l}{m} \\ 0 & 0 \end{bmatrix},$$
$$B_p = [0 \;\; \tfrac{10000}{J}]^T, \quad C_p = [0 \;\; 1].$$

The control specifications are given by the control setup shown in Figure 1, where

$$W_e(s) = 20\,\frac{s/100 + 1}{s + 1}, \quad W_u(s) = 0.1\,\frac{s/2.5 + 1}{s/250 + 1}.$$

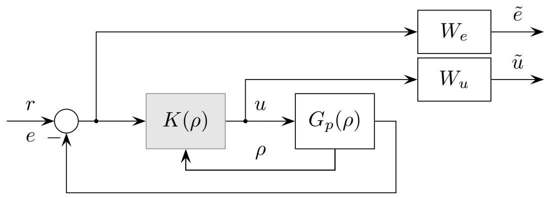

Figure 1. Control synthesis setup

The LPV model (1)-(2) is obtained interconnecting $G_p(\rho)$ with the previous weights according to Figure 1. For this resulting model

$$
\begin{aligned}
\bar{\sigma}(A_0) &= 288.00, & \max_{\rho\in\mathcal{P}} |f_1(\rho)|\bar{\sigma}(A_1) &= 109.51,\\
\max_{\rho\in\mathcal{P}} |f_2(\rho)|\bar{\sigma}(A_2) &= 17.85, & \max_{\rho\in\mathcal{P}} |f_3(\rho)|\bar{\sigma}(A_3) &= 37.85,\\
\max_{\rho\in\mathcal{P}} |f_4(\rho)|\bar{\sigma}(A_4) &= 1.49, & \max_{\rho\in\mathcal{P}} |f_5(\rho)|\bar{\sigma}(A_5) &= 3.71,
\end{aligned}
$$

which suggest that it would be reasonable to exclude the terms 4, 5 and even 2 from the controller expressions.

For sake of comparison, several LPV controllers were designed with a grid of 25 points using SeDuMi (Sturm, 1999) and YALMIP (Löfberg, 2004) to solve the optimization problems. The results are summarized in Table 1. The controller K_{f} was designed including all terms ($\mathcal{J} = \emptyset$) and the others taking $\mathcal{J}_1 = \{4,5\}$ and $\mathcal{J}_2 = \{2,4,5\}$. The controllers $K_{\mathrm{p1},j}$ ($j = 1,2$) were designed using Procedure 1 with $q = 0.0001$ and $K_{\mathrm{p2},j}$ using Procedure 2. The values γ_{real} were obtained checking the performance condition (4) for the closed-loop system with the previous controllers. It can be observed that although Procedure 1 is optimistic ($\gamma < \gamma_{\mathrm{real}}$) and Procedure 2 is conservative ($\gamma > \gamma_{\mathrm{real}}$), all controllers are stabilizing as both procedures produce controllers that satisfy (5) and (6). Clearly, a simpler controller excluding the terms in $\mathcal{J}_2$ results in higher γ's, *i.e.*, in lower performance levels.

Figure 2 shows step responses obtained with controllers K_{f}, $K_{\mathrm{p1},j}$ and $K_{\mathrm{p2},j}$, $j = 1,2$. To ease the comparison, several characteristic values are provided: the rise time t_r, the settling time t_s, the overshoot M_p, and the steady state error e_{ss}. A slight performance degradation can be observed with controllers $K_{\mathrm{p1},1}$ and $K_{\mathrm{p2},1}$ as compared to K_{f}. In particular, the simpler controllers present less uniform responses indicating lower capability to adapt themselves to different working conditions. On the other hand, the performance degradation is more marked in the case of controllers $K_{\mathrm{p1},2}$ and $K_{\mathrm{p2},2}$, which corresponds with the estimation given by $|f_2(\rho)|\bar{\sigma}(A_2)$.

With the aim of comparison, a controller K_{pol} was designed by modelling the nonlinear system as a polytopic LPV model with parameters $\theta_i = f_i(\rho)$, $i = 1,\ldots,5$. The parameter set was defined as $\Theta = [\underline{f}_1\ \bar{f}_1] \times \cdots \times [\underline{f}_5\ \bar{f}_5]$,

Table 1
Synthesis results

	Procedure 1			Procedure 2		
$\mathcal{J}$	Ctrl	γ	γ_{real}	Ctrl	γ	γ_{real}
$\emptyset$	–	–	–	K_{f}	1.003	1.003
$\{4,5\}$	$K_{\mathrm{p1},1}$	1.422	1.865	$K_{\mathrm{p2},1}$	3.679	2.676
$\{2,4,5\}$	$K_{\mathrm{p1},2}$	1.779	1818	$K_{\mathrm{p2},2}$	11.770	5.542

Polytopic LPV			Reduced by PCA		
Ctrl	γ	Qty. param.	Ctrl	γ	Qty. param.
K_{pol}	2.49	5	K_{red}	20.14	3

Structured LPV		
Ctrl	γ	γ_{real}
K_{st}	0.99	∞

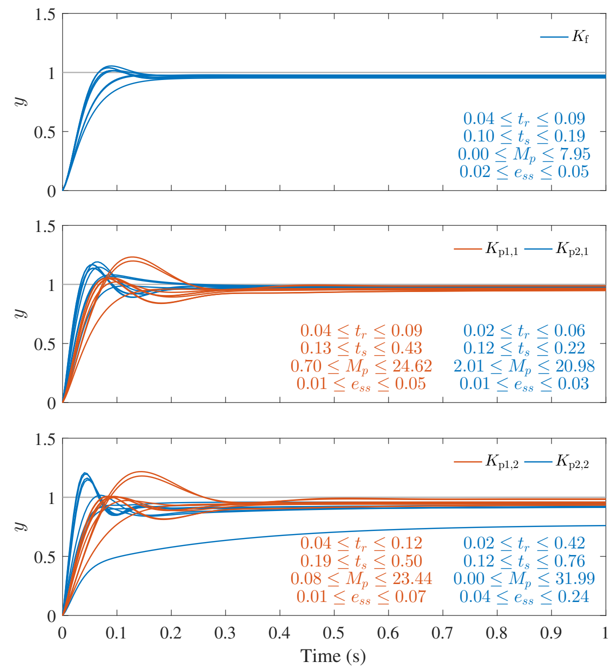

Figure 2. Step responses of the closed-loop system with the controllers listed in Table 1 at several frozen parameter values, where t_r is the rise time, t_s the settling time, M_p the overshoot, and e_{ss} the steady state error.

with $\underline{f}_i = \min_{\rho\in\mathcal{P}} f_i(\rho)$ and $\bar{f}_i = \max_{\rho\in\mathcal{P}} f_i(\rho)$. The performance level in this case was $\gamma = 2.49$. The resulting controller consists of 32 terms and the control action must be computed after solving a convex decomposition (22) of 32 vertices. Using the procedure proposed in (Kwiatkowski & Werner, 2008; Hoffmann et al., 2014), the parameter dimension was reduced to 3 in order to design the K_{red} controller. Unfortunately, in this particular example the performance level shows a significant degradation: $\gamma = 20.14$. These results are summarized in Table 1.

As an additional comparison, a structured gain scheduled controller K_{st} was designed using the methodology proposed in (Gahinet & Apkarian, 2013). The controller structure was selected similar to the controllers $K_{\mathrm{p1,1}}$ and $K_{\mathrm{p2,1}}$, *i.e.*, removing terms 4 and 5, and using the design setup in Figure 1. The performance level results $\gamma = 0.99$, but this controller cannot ensure quadratic stability as the γ_{real} is infinity. This implies that the closed-loop stability will deteriorate for fast parameter variations. The corresponding results are shown in Table 1.

In Figure 3, it can be seen simulation results corresponding to the closed-loop system using controllers K_{f}, $K_{\mathrm{p1,1}}$ and $K_{\mathrm{p2,1}}$ on the nonlinear model. The parameter trajectories are shown in the middle and bottom plots, respectively. Notice that ρ_1 is a state and its trajectory is given by the closed-loop behavior and the external parameter ρ_2. Nevertheless, the parameter ρ_1 remains inside the interval $[-0.2\ 0.2]$. The evolution of the corresponding controlled output y is in accordance to the results observed in Figure 2, although for the particular parameter trajectory the performance degradation is smaller. Figure 4 compares the closed-loop responses $K_{\mathrm{p2,1}}$, K_{red}, and K_{st} in the previously described scenario. It can be observed that the conservatism of K_{red} results in a poorer reference tracking. On the other hand, the lack of quadratic stability guarantees, in the case of K_{st}, yields more oscillatory responses when the parameters change.

The performance level in Table 1 and the closed-loop responses in Figures 2 and 3 allow us to evaluate possible implementations. The controller order is 4, therefore each matrix in (19) has dimension 5×5. Then the use of controllers $K_{\mathrm{p1,1}}$ and $K_{\mathrm{p2,1}}$ implies a reduction of 33.3% in the number of variables to be stored, and 50% in the case $K_{\mathrm{p1,2}}$ and $K_{\mathrm{p2,2}}$ are used. Besides there is a reduction in the number of mathematical operations. On the other hand, the reduced polytopic controller K_{red} requires the storage of 9 local controllers and the computation of a convex decomposition. Whereas, the structured gain-scheduled controller K_{st} presents a similar implementation complexity than $K_{\mathrm{p1,1}}$ and $K_{\mathrm{p2,1}}$, however it cannot guarantee quadratic stability. In any case, the proposed methodology also provides tools to evaluate the effects of the controller simplification.

5 Conclusions

This article proposes a methodology to design LPV controllers with lower implementation complexity and to evaluate the effects of these simplifications on the closed-

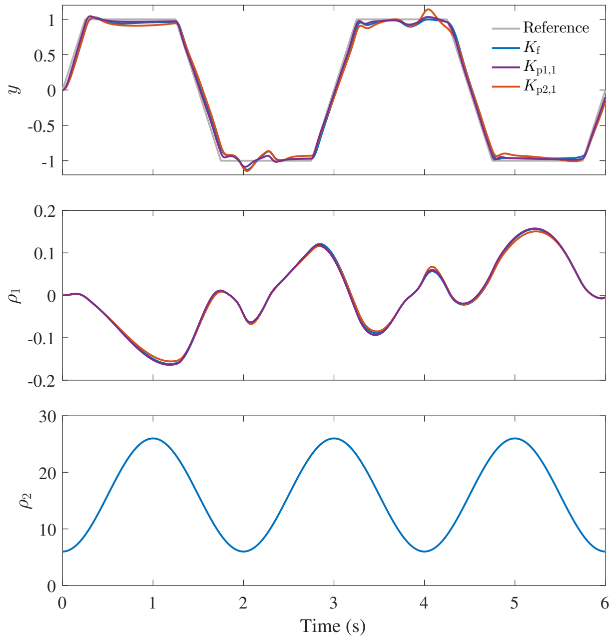

Figure 3. Comparison of the closed-loop responses using the full controller K_{f}, and the simplified $K_{\mathrm{p1,1}}$ and $K_{\mathrm{p2,1}}$ on the nonlinear bicycle model.

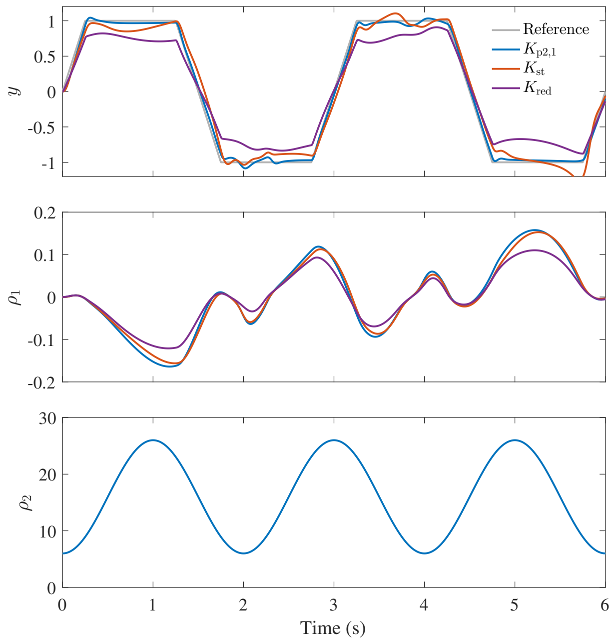

Figure 4. Comparison of the closed-loop responses using the controllers $K_{\mathrm{p2,1}}$, K_{st}, and K_{red} on the nonlinear bicycle model.

loop performance. The aim is to simplify the implementation by reducing the mathematical operations to be performed at each time step, the number of matrices to be stored and, depending on the particular plant, the number of measured parameters. This reduction in the implementation complexity may be useful to apply LPV controls in low computational-power systems and possibly reduce the number of sensors. Clearly, imposing a parameter dependence limits the set of stabilizing controllers and may affect the closed-loop performance. For this reason, guidelines to estimate the compromise between implementation complexity and performance degradation is proposed here.

Acknowledgements

The authors acknowledge the financial support of ITBA and CONICET.

References

Apkarian, P. & Adams, R. (1998). Advanced gain-scheduling techniques for uncertain systems. *IEEE Transactions on Control Systems Technology*, *6*(1), 21–32.

Apkarian, P., Gahinet, P., & Becker, G. S. (1995). Self-scheduled H_∞ control of linear parameter-varying systems: a design example. *Automatica*, *31*(9), 1251–1261.

Beck, C. (2006). Coprime factors reduction methods for linear parameter varying and uncertain systems. *Systems & Control Letters*, *55*(3), 199–213.

Becker, G. S. & Packard, A. (1994). Robust performance of linear parametrically varying systems using parametrically-dependent linear feedback. *Systems & Control Letters*, *23*(3), 205–215.

Belikov, J., Kotta, U., & Tõnso, M. (2014). Comparison of LPV an nonlinear system theory: a realization problem. *Systems & Control Letters*, *64*(2), 72–78.

de Hillerin, S., Scorletti, G., & Fromion, V. (2011). Reduced-complexity controllers for LPV systems: Towards incremental synthesis. In *Proc. 50th Conference on Decision and Control and European Control Conference*, (pp. 3404–3409).

Gahinet, P. & Apkarian, P. (2013). Automated tuning of gain-scheduled control systems. In *Proc. 52nd Conference on Decision and Control*, (pp. 2740–2745).

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*(6), 2393–2398.

Horn, R. A. & Johnson, C. R. (2013). *Matrix analysis* (2nd ed. ed.). New York, USA: Cambridge University Press.

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*(4), 781–788.

Löfberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB. In *In Proc. CACSD Conf.*, 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*(5), 4922–4932.

Petersson, D. & Lofberg, J. (2014). Optimisation-based modelling of LPV systems using an H_2-objective. *International Journal of Control*, *87*(8), 1536–1548.

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

Scherer, C. W. (2001). LPV control and full block multipliers. *Automatica*, *37*(3), 361–375.

Scherer, C. W., Gahinet, P., & Chilali, M. (1997). Multiobjective output-feedback control via LMI optimization. *IEEE Transactions on Automatic Control*, *42*(7), 896–911.

Sturm, J. F. (1999). Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. *Optimization methods and software*, *11*(1-4), 625–653.

Tóth, R. (2010). *Modeling and identification of linear parameter-varying systems*. Number 403 in Lecture notes in control and information sciences. Berlin, Germany: Springer.

Wood, G. D., Goddard, P. J., & Glover, K. (1996). Approximation of linear parameter-varying systems. In *Proc. 35th Conference on Decision and Control*, volume 1, (pp. 406–411).

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 on Robust and Nonlinear Control*, *6*(9-10), 983–998.