Int. J. Appl. Math. Comput. Sci., 2019, Vol. 29, No. 4, 725–737
DOI: 10.2478/amcs-2019-0054

AN LMI–BASED HEURISTIC ALGORITHM FOR VERTEX REDUCTION IN LPV SYSTEMS

ADRIÁN SANJUAN [a,*], DAMIANO ROTONDO [a,b], FATIHA NEJJARI [a], RAMON SARRATE [a]

[a]Research Center for Supervision, Safety and Automatic Control (CS2AC)
Polytechnic University of Catalonia (UPC), Rambla Sant Nebridi 10, 08222 Terrassa, Spain
e-mail: {adrian.sanjuan,damiano.rotondo}@upc.edu,
{fatiha.nejjari,ramon.sarrate}@upc.edu

[b]Institute of Robotics and Industrial Informatics (CSIC-UPC)
Llorens i Artigas 4-6, 08028 Barcelona, Spain

The linear parameter varying (LPV) approach has proved to be suitable for controlling many non-linear systems. However, for those which are highly non-linear and complex, the number of scheduling variables increases rapidly. This fact makes the LPV controller implementation not feasible for many real systems due to memory constraints and computational burden. This paper considers the problem of reducing the total number of LPV controller gains by determining a heuristic methodology that combines two vertices of a polytopic LPV model such that the same gain can be used in both vertices. The proposed algorithm, based on the use of the Gershgorin circles, provides a combinability ranking for the different vertex pairs, which helps in solving the reduction problem in fewer attempts. Simulation examples are provided in order to illustrate the main characteristics of the proposed approach.

Keywords: linear parameter varying (LPV) paradigm, linear matrix inequality (LMI), Gershgorin circles, gain scheduling, controller design.

1. Introduction

The necessity for systematic analysis and design tools for non-linear systems has led to the development of gain-scheduled control techniques (Rugh and Shamma, 2000; Rotondo, 2017). In this sense, one of the most successful paradigms is the linear parameter varying (LPV) one (Mohammadpour and Scherer, 2012), which has proved to be suitable for controlling non-linear systems by embedding the non-linearities in the varying parameters (Kwiatkowski *et al.*, 2006; Rotondo *et al.*, 2015b), which often depends on some endogenous signals, e.g., states, inputs or outputs. In this case, the system is referred to as *quasi*-LPV, to distinguish it from a *pure* LPV system, in which the varying parameters depend only on exogenous signals (Marcos and Balas, 2004). The LPV paradigm has evolved quickly in the last two decades and has been successfully exploited in many applications (Hoffmann and Werner, 2015), e.g., wind turbines (Inthamoussou *et al.*, 2014), vehicles (Fergani *et al.*, 2017), robots (Rotondo *et al.*, 2015a) and aeromechanical systems (Rotondo *et al.*, 2013).

Since the introduction of the LPV paradigm, a lot of research has concerned the development of design techniques for LPV systems with a linear fractional transformation (LFT) form (Packard, 1994). However, this approach took into account complex varying parameters that did not appear in real plants, thus introducing a strong source of conservativeness (White *et al.*, 2013). For this reason, Lyapunov-based approaches were developed, allowing taking into account constraints on the rate of variation of the varying parameters (Gahinet *et al.*, 1996). A high number of related results published in the last few years shows that the development of efficient approaches to the design of gain-scheduled controllers for LPV systems (Zhao and Nagamune, 2017; Nguyen *et al.*, 2018; Chitraganti *et al.*, 2017) and for fault diagnosis (López-Estrada *et al.*, 2015; Zhou *et al.*, 2018) is still a hot research topic.

*Corresponding author

The design problem in the case of a gain-scheduled controller for LPV systems takes the form of a set of parameter-dependent linear matrix inequalities (LMIs), which correspond to an infinite number of constraints and are thus computationally intractable. For this reason, polytopic LPV models for which the state space parameter-varying matrices can be expressed as convex combinations of constant matrices, usually referred to as *vertex matrices*, are widespread (Rotondo *et al.*, 2014) (the LTI systems obtained from these matrices are referred to as *vertex systems*). However, when a system is highly non-linear and complex, the number of scheduling variables required to represent the system in an LPV (or quasi-LPV) structure increases. Consequently, when methods for obtaining a polytopic representation are applied, the number of vertices grows even faster. For example, for the most popular method, which is the *bounding box* technique (Sun and Postlethwaite, 1998), the relation between the number of vertices n_v and the number of scheduling variables n_p is given by the exponential expression

$$n_v = 2^{n_p}. \tag{1}$$

Even for a relatively small and simple laboratory setup, such as the twin rotor MIMO system (Rahideh and Shaheed, 2007), the direct application of the bounding box technique would lead to 2048 vertex systems (Rotondo *et al.*, 2013). Ideally, the LPV controller for this system would be obtained as a convex combination of 2048 vertex controller gains, i.e., a controller gain for each vertex system. Thus, for many real systems, the real-time implementation of an LPV controller in a microcontroller might be infeasible. This fact is due to many reasons, some of which are:

- Memory constraints: for each vertex, a different controller gain should be stored in the embedded system. For a system with 6 states, 4 inputs and 15 scheduling variables, each controller gain will involve 24 floating-point values, so 3.15 MB will be required to store the controller. For a system with the same number of states and inputs, and 20 scheduling variables, 100.7 MB would be required.
- Computational burden: the time required by the embedded system to compute the current controller gain as a convex combination of the vertex gains is directly related to the number of vertices. Therefore, the computational load can be a critical issue when the main dynamics of the controlled system are fast.

In order to ease these problems, several researchers have investigated methods that help in reducing the controller complexity. These methods can be summarized and classified as follows:

- PCA-based approaches: they aim at determining operational trajectories of the plant and reshape the hyper-box representing the parameter range, such that it matches the given operating points as closely as possible. This is done by means of a procedure based on principal component analysis (PCA) (Kwiatkowski and Werner, 2008; Rizvi *et al.*, 2016; Jabali and Kazemi, 2017).
- Gap metric-based approaches: they use the gap metric (El-Sakkary, 1985) and the ν-gap metric (Vinnicombe, 1996) to obtain an indication of how much performance/robustness is lost when a controller synthesized for a linear system P_1 is applied to a linear system P_2. In this way, an algorithm for finding an appropriate gridding for the scheduling parameter space can be devised (Fleischmann *et al.*, 2016; Theodoulis and Duc, 2009; Zribi *et al.*, 2016).
- Model order reduction approaches: their goal is to reduce the state dimension and, accordingly, the complexity of the underlying mathematical model, with a minimal change in the overall input-output behaviour (Luspay *et al.*, 2018; Theis *et al.*, 2017).

In this paper, we would like to tackle the problem of reducing the total number of controller gains from a different point of view than the aforementioned approaches. This reduction will be made analyzing how the LMIs, which represent a set of specifications, are modified when two vertices have the same controller.

In particular, the goal of this paper is to provide a heuristic methodology to combine two vertices of an LPV polytopic model such that the same vertex controller gain can be used in both vertices. In this way, the number of vertex controllers will be reduced from n_v to $n_v - 1$, which is a first step in order to develop an iterative algorithm that leads to the design of an efficient and implementable controller.

Note that a possible way of solving the reduction problem is through a brute-force search that considers all possible vertex combinations and tests the feasibility of suitable LMI-based design when groups of vertices are combined together by assigning them a common controller gain. However, this approach is not efficient in practice, since one should check the feasibility for a number of cases equal to the Bell number (Pemmaraju and Skiena, 2003):

$$B_{n_v} = \sum_{k=0}^{n_v-1} \binom{n_v-1}{k} B_k. \tag{2}$$

For example, in the case of a system with 6 vertices, 203 possible combinations should be checked, while for a system with 15 vertices, we would need to check 1.38×10^9 combinations, which is computationally untreatable. Similarly, a brute-force search for reducing the number of vertex controllers from n_v to $n_v - 1$ would require checking LMI feasibility for $n_v(n_v - 1)/2$ possible pairs of vertices. The heuristic algorithm proposed in this paper provides a combinability ranking for the different vertex pairs that helps in solving the reduction problem in fewer attempts. The proposed heuristic algorithm is based on the use of the Gershgorin circles, an algebraic tool which has already found several applications in systems and control theories (Curran, 2009; Ho *et al.*, 2000).

The remainder of the paper is structured as follows. Section 2 presents the controller design for an LPV system, while the vertex reduction is analyzed in Section 3. The Gershgorin circles-based algorithm is provided in Section 4. Section 5 deals with the pole placement case study and some simulation results are presented in Section 6. Finally, Section 7 outlines the main conclusions.

2. Feedback controller design for an LPV model

Consider the following continuous time LPV system:

$$\dot{\mathbf{x}}(t) = \mathbf{A}(\boldsymbol{\theta}(t))\mathbf{x}(t) + \mathbf{B}(\boldsymbol{\theta}(t))\mathbf{u}(t), \tag{3}$$

where $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ and $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ are the state and input vectors, respectively, and $\mathbf{A}(\boldsymbol{\theta}(t)) \in \mathbb{R}^{n_x \times n_x}$ and $\mathbf{B}(\boldsymbol{\theta}(t)) \in \mathbb{R}^{n_x \times n_u}$ are parameter-dependent matrices. Here $\boldsymbol{\theta}(t) \in \Theta \subset \mathbb{R}^{n_p}$ is the vector of time-varying parameters.

A standard assumption in the literature is that the matrix $\mathbf{A}$ depends on $\boldsymbol{\theta}(t)$ in a polytopic way, i.e., it can be expressed as a convex combination of constant matrices through non-negative coefficients which depend on $\boldsymbol{\theta}(t)$ (Apkarian *et al.*, 1995). Moreover, the matrix $\mathbf{B}$ is often assumed to be constant, which means that the LPV system (3) can be equivalently represented as follows:

$$\dot{\mathbf{x}}(t) = \sum_{v=1}^{n_v} \alpha_v(\boldsymbol{\theta}(t))\mathbf{A}_v\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{4}$$

where $\mathbf{A}_v$, $v = 1, \dots, n_v$, defines the so-called vertex systems, n_v is the number of vertices and α_v are the coefficients of the polytopic decomposition, such that

$$\begin{cases} \sum\limits_{v=1}^{n_v} \alpha_v(\boldsymbol{\theta}(t)) = 1, \\ \alpha_v(\boldsymbol{\theta}(t)) \geq 0, \quad \forall v = 1, \dots, n_v, \ \forall \boldsymbol{\theta}(t) \in \Theta. \end{cases} \tag{5}$$

Note that LPV systems for which the matrix $\mathbf{B}$ is not constant can be brought to the form (4) by prefiltering the input $\mathbf{u}(t)$ (Apkarian *et al.*, 1995) or, alternatively, they can be dealt with by adding some complexity to the design conditions (Montagner *et al.*, 2005).

Hereafter, we will consider the problem of designing an LPV state-feedback controller which achieves some desired closed-loop specifications:

$$\mathbf{u}(t) = \mathbf{K}(\boldsymbol{\theta}(t))\mathbf{x}(t), \tag{6}$$

where $\mathbf{K}(\boldsymbol{\theta}(t)) \in \mathbb{R}^{n_u \times n_x}$ is the parameter-dependent controller gain. In their simplest version, the design conditions for different types of specifications, e.g., stability or guaranteed $\mathcal{H}_\infty$ performance, take the form of parameterized LMIs with unknown variables $\mathbf{X} \succ \mathbf{0}$ (Lyapunov matrix) and $\boldsymbol{\Gamma}(\boldsymbol{\theta}(t)) = \mathbf{K}(\boldsymbol{\theta}(t))\mathbf{X}$ (Apkarian and Tuan, 2000) such that

$$\mathbf{F}\left(\mathbf{A}(\boldsymbol{\theta}(t)), \mathbf{B}, \mathbf{X}, \boldsymbol{\Gamma}(\boldsymbol{\theta}(t))\right) \prec 0, \quad \forall \theta \in \Theta. \tag{7}$$

However, (7) involves an infinite number of LMIs, and needs to be reduced to a finite number in order to be computationally tractable. To this end, the LPV controller is chosen to be polytopic, which means that (6) becomes

$$\mathbf{u}(t) = \sum_{v=1}^{n_v} \alpha_v(\boldsymbol{\theta}(t))\mathbf{K}_v\mathbf{x}(t), \tag{8}$$

where $\mathbf{K}_v \in \mathbb{R}^{n_u \times n_x}$ are the vertex controller gains. Under this assumption, (7) can be rewritten at the vertices as:

$$\mathbf{F}\left(\mathbf{A}_v, \mathbf{B}, \mathbf{X}, \boldsymbol{\Gamma}_v\right) \prec 0, \quad \forall v = 1, \dots, n_v \tag{9}$$

and, once a feasible solution of (9) has been obtained, the vertex controller gains can be determined as

$$\mathbf{K}_v = \boldsymbol{\Gamma}_v\mathbf{X}^{-1}. \tag{10}$$

As stated in the introduction, we are interested in decreasing the number of controller gains from n_v to $n_v - 1$, which means that for two indices $i, j \in \{1, \dots, n_v\}$ with $i \neq j$, $\mathbf{K}_i = \mathbf{K}_j$ will hold. However, depending on how i and j are chosen, this reduction could potentially lead to infeasibility of the LMIs (9). In the following sections, we will describe a heuristic algorithm, based on the use of the Gershgorin circles, which will provide a combinability ranking for the different vertex pairs, so that if vertices i and j are combined, the feasibility of the LMI (9) when $\mathbf{K}_i = \mathbf{K}_j$ still holds.

3. Analysis of vertex reduction

The complexity of the LMI problem (9) implies that an explicit solution cannot be determined. Hence, numerical solvers such as interior point methods (Boyd *et al.*, 1994) must be used to obtain a solution. For this reason, predicting when two or more vertices can be combined is a complex task. Let us start by introducing a formal definition of *combinable vertices*.

Definition 1. (*Combinable vertices*) Two vertices i and j are said to be *combinable* for the set of LMIs (9) if there exist matrices $\mathbf{\Gamma}_v \in \mathbb{R}^{n_u \times n_x}$, $v \in \{1,\dots,n_v\} \setminus \{i,j\}$ and a matrix $\mathbf{\Gamma}_w \in \mathbb{R}^{n_u \times n_x}$ such that

$$\begin{cases} \mathbf{F}\left(\mathbf{A}_v, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}_v\right) \prec 0, & \forall v \in \{1,\dots,n_v\} \setminus \{i,j\}, \\ \mathbf{F}\left(\mathbf{A}_v, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}_w\right) \prec 0, & \forall v \in \{i,j\}. \end{cases} \tag{11}$$

Given this definition, we present a heuristic methodology for determining the likelihood that two vertices are combinable, based on quantifying how much the LMIs are perturbed when a common controller gain is used.

Considering two vertices of the system i and j, the state matrix at the vertex i ($\mathbf{A}_i$) can be seen as a perturbation in the state matrix $\mathbf{A}_j$ such that

$$\mathbf{A}_i = \mathbf{A}_j + \Delta\mathbf{A}^{\{i,j\}}, \tag{12}$$

where $\Delta\mathbf{A}^{\{i,j\}}$ is the perturbation state matrix.

Furthermore, when vertices i and j are combinable, this means that a common controller gain can be designed for these vertices. This means that $\mathbf{K}_i = \mathbf{K}_j$ and consequently $\mathbf{\Gamma}_i = \mathbf{\Gamma}_j = \mathbf{\Gamma}^{\{i,j\}}$. Then, the objective is to determine the effect of $\Delta\mathbf{A}^{\{i,j\}}$ on the LMI (9):

$$\begin{aligned} &\begin{cases} \mathbf{F}\left(\mathbf{A}_i, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}^{\{i,j\}}\right) \prec 0, \\ \mathbf{F}\left(\mathbf{A}_j, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}^{\{i,j\}}\right) \prec 0, \end{cases} \\ \Leftrightarrow &\begin{cases} \mathbf{F}\left(\mathbf{A}_i, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}^{\{i,j\}}\right) \prec 0, \\ \mathbf{F}\left(\mathbf{A}_i, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}^{\{i,j\}}\right) - \Delta\mathbf{F}\left(\Delta\mathbf{A}^{\{i,j\}}, \mathbf{X}\right) \prec 0, \end{cases} \\ \Leftrightarrow &\begin{cases} \mathbf{F}\left(\mathbf{A}_j, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}^{\{i,j\}}\right) + \Delta\mathbf{F}\left(\Delta\mathbf{A}^{\{i,j\}}, \mathbf{X}\right) \prec 0, \\ \mathbf{F}\left(\mathbf{A}_j, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}^{\{i,j\}}\right) \prec 0, \end{cases} \end{aligned} \tag{13}$$

where $\Delta\mathbf{F}\left(\Delta\mathbf{A}^{\{i,j\}}, \mathbf{X}\right)$ is the perturbation matrix that represents the effect of $\Delta\mathbf{A}^{\{i,j\}}$ on the LMIs of vertex j. Note that the notation $\Delta\mathbf{F}\left(\Delta\mathbf{A}^{\{i,j\}}, \mathbf{X}\right) := \Delta\mathbf{F}^{\{i,j\}}$, $\mathbf{F}\left(\mathbf{A}_i, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}^{\{i,j\}}\right) := \mathbf{F}_i$ and $\mathbf{F}\left(\mathbf{A}_j, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}^{\{i,j\}}\right) := \mathbf{F}_j$ will be used in the following sections.

Analyzing (13), it can be concluded that a lower perturbation generates a lower variation in the eigenvalues of (9) and this means that it is more likely that the matrix on the left-hand side of the LMI will keep negative definiteness by using the same controller gain. A graphical method, referred to as Gershgorin circles, can be used to predict the variation in the eigenvalues, allowing determining the likelihood that two vertices can be combined successfully.

4. Gershgorin circles-based algorithm

Gershgorin circles (GCs) (Quarteroni *et al.*, 2010) provide an *a priori* bound to the eigenvalues of the matrix $\mathbf{E}$:

$$\forall \lambda \in \sigma(\mathbf{E}), \qquad \lambda \in \mathcal{S}_{\mathcal{R}} \cap \mathcal{S}_{\mathcal{C}}, \tag{14}$$

where

$$\mathcal{S}_{\mathcal{R}} = \bigcup_{i=1}^{n_e} \mathcal{R}_i, \quad \mathcal{R}_i = \Big\{ z \in \mathbb{C} : |z - e_{ii}| \le \sum_{\substack{j=1 \\ j \ne i}}^{n_e} |e_{ij}| \Big\}, \tag{15}$$

$$\mathcal{S}_{\mathcal{C}} = \bigcup_{j=1}^{n_e} \mathcal{C}_j, \quad \mathcal{C}_j = \Big\{ z \in \mathbb{C} : |z - e_{jj}| \le \sum_{\substack{i=1 \\ i \ne j}}^{n_e} |e_{ij}| \Big\}, \tag{16}$$

The sets $\mathcal{R}_i$ and $\mathcal{C}_j$ are called row and column Gershgorin circles, respectively. Here e_{ij} is the element of matrix $\mathbf{E} \in \mathbb{R}^{n_e \times n_e}$ located in row i and column j and $\sigma(\mathbf{E})$ is the spectrum of matrix $\mathbf{E}$. It must be highlighted that $\mathbf{F}_i$, $\mathbf{F}_j$ and $\Delta\mathbf{F}^{\{i,j\}}$ (13) are symmetric matrices. Therefore, their eigenvalues will be real and $\mathcal{S}_{\mathcal{R}} = \mathcal{S}_{\mathcal{C}}$ will hold.

The perturbation $\Delta\mathbf{F}^{\{i,j\}}$ will move the centres and modify the radii of the Gershgorin circles of the matrices $\mathbf{F}_i$ and $\mathbf{F}_j$, because they are symmetric. Thus, in the worst situation, the perturbation matrix $\Delta\mathbf{F}^{\{i,j\}}$ will produce a variation in the eigenvalues of $\mathbf{F}_i$ and $\mathbf{F}_j$ equal to the largest distance of the Gershgorin circles of $\Delta\mathbf{F}^{\{i,j\}}$:

$$\Delta r_{\max}^{\{i,j\}} = \max_k \Big(|\Delta f_{kk}^{\{i,j\}}| + \sum_{\substack{l=1 \\ l \ne k}}^{n} |\Delta f_{kl}^{\{i,j\}}| \Big), \tag{17}$$

where $\Delta f_{kl}^{\{i,j\}}$ is the element of matrix $\Delta\mathbf{F}^{\{i,j\}}$ located in row k and column l. Note that only the absolute values of $\Delta\mathbf{F}^{\{i,j\}}$ are involved. Therefore, $\Delta r_{\max}^{\{i,j\}} = \Delta r_{\max}^{\{j,i\}}$.

The algorithm for determining the sequence will be based on sorting increasingly the values $\Delta r_{\max}^{\{i,j\}}$, as detailed in Algorithm 1. The algorithm will stop when a feasible solution is found or the first n_k vertex pairs of the sequence (s_{GC}) are analyzed, where $n_k \in \{1, n_v(n_v - 1)/2\}$ defines a trade-off between the probability to find combinable vertices and the computational burden (a lower value of n_k means that it is harder to find a solution but the computational load would be lower). If the second condition is fulfilled, this means that *probably* two combinable vertices for the required design specifications do not exist. Note that certainty about this statement can be obtained only by checking feasibility for all the possible vertex pairs.

Algorithm 1. Vertex reduction ($n_v \to n_v - 1$): GC sequence.

Solve (9) to obtain a solution of $\mathbf{X}$.
Calculate $\Delta\mathbf{A}^{\{i,j\}}$ using (12) and determine $\Delta\mathbf{F}^{\{i,j\}}$ for each pair of vertices.
Compute $\Delta r_{\max}$ (17) for each $\Delta\mathbf{F}^{\{i,j\}}$.
Sort $\Delta r_{\max}^{\{i,j\}}$ increasingly, generating the Gershgorin circles-based sequence (s_{GC}).
for $k = 1$ **to** n_k **do**
    Combine the vertices $\{i, j\}$ in the k-th position of s_{GC}.
    Evaluate the LMIs (9).
    if $\{i, j\}$ provides a feasible solution **then**
        return $\{i, j\}$
    end if
end for
return No feasible solution found

Example 1. (*Computation of* $\Delta r_{\max}$) Consider an unknown matrix $\mathbf{M}$ that is perturbed by $\mathbf{N}$ as

$$\mathbf{M} + \mathbf{N} = \begin{bmatrix} m_a & m_c \\ m_c & m_b \end{bmatrix} + \begin{bmatrix} n_a & n_c \\ n_c & n_b \end{bmatrix}. \tag{18}$$

In the worst case, the Gershgorin circles are

$$\begin{aligned} \mathcal{R}_1 = \mathcal{C}_1 = \{z : |z - (m_a + |n_a|)| \le |m_c| + |n_c|\}, \\ \mathcal{R}_2 = \mathcal{C}_2 = \{z : |z - (m_b + |n_b|)| \le |m_c| + |n_c|\}. \end{aligned} \tag{19}$$

Then, analyzing (19), it can be concluded that $\Delta r_{\max}$ (17) of $\mathbf{N}$ is

$$\Delta r_{\max} = \max(|n_a| + |n_c|, |n_b| + |n_c|). \tag{20}$$

♦

Example 2. (*Sequence generation*) Consider a system with three vertices $v = [1, 2, 3]$ such that the values of $\Delta r_{\max}$ corresponding to the combinations $\{1, 2\}$, $\{1, 3\}$ and $\{2, 3\}$ are 7, 2 and 6, respectively. Hence, the sequence that should be followed to reduce the vertices of the system is as follows:

Vertex	$\Delta r_{\max}^{\{i,j\}}$
$\{1, 2\}$	7
$\{1, 3\}$	2
$\{2, 3\}$	6

$\to$

s_{GC}	$\Delta r_{\max}^{\{i,j\}}$
$\{1, 3\}$	2
$\{2, 3\}$	6
$\{1, 2\}$	7

♦

It must be noted that, for a given Lyapunov matrix $\mathbf{X}$, different sets of possible $\mathbf{\Gamma}_v$ which provide the feasibility of the LMIs (9) are obtained. As a consequence, determining whether or not two vertices can be combined will depend, in general, on the value of $\mathbf{X}$. Algorithm 1 assumes that a feasible solution of (9) has been obtained (with non-combined vertices), and uses this specific solution $\mathbf{X}$ to compute the Gershgorin circles and generate a sequence which ranks the vertex pairs based on their susceptibility to be combinable.

5. Case study: Pole-placement constraints

In this paper, the pole-placement controller design (Chilali and Gahinet, 1996) has been selected to apply and illustrate the proposed methodology.

5.1. Pole-placement constraints: LMI formulation.

The system (4) is quadratically $\mathcal{D}$-stabilizable in the region $S(\alpha, r, \gamma)$ (Chilali and Gahinet, 1996) (see Fig. 1) if there exist a symmetric positive definite matrix $\mathbf{X} > 0$ and $\mathbf{\Gamma}_v$ $\forall\ v = 1, \dots, n_v$ such that

$$\boldsymbol{\mathcal{D}}_v := \begin{pmatrix} \boldsymbol{\mathcal{D}}_{\alpha_v} & 0 & 0 \\ 0 & \boldsymbol{\mathcal{D}}_{r_v} & 0 \\ 0 & 0 & \boldsymbol{\mathcal{D}}_{\gamma_v} \end{pmatrix} < 0, \quad \forall\ v = 1, \dots, n_v, \tag{21}$$

where $\boldsymbol{\mathcal{D}}_v = \mathbf{F}(\mathbf{A}_v, \mathbf{B}, \mathbf{X}, \mathbf{\Gamma}_v)$ is the LMI corresponding to the $\mathcal{D}$-stable region $S(\alpha, r, \gamma)$, created by the intersection of three regions:

- Region α ($\boldsymbol{\mathcal{D}}_{\alpha_v}$): represents the α-stability region (a left half-plane with abscissa α),

$$\boldsymbol{\mathcal{D}}_{\alpha_v} = \mathbf{z}_v + \mathbf{z}_v^T + 2\alpha\mathbf{X} \tag{22}$$

  with

$$\mathbf{z}_v = \mathbf{A}_v\mathbf{X} + \mathbf{B}\mathbf{\Gamma}_v; \tag{23}$$

- Region r ($\boldsymbol{\mathcal{D}}_{r_v}$): represents the disc with radius r and centre $(-q, 0)$,

$$\boldsymbol{\mathcal{D}}_{r_v} = \begin{pmatrix} -r\mathbf{X} & q\mathbf{X} + \mathbf{z}_v \\ q\mathbf{X} + \mathbf{z}_v^T & -r\mathbf{X} \end{pmatrix} \tag{24}$$

  with $\mathbf{z}_v$ defined as (23);

- Region γ ($\boldsymbol{\mathcal{D}}_{\gamma_v}$): represents the conic sector with angle γ,

$$\boldsymbol{\mathcal{D}}_{\gamma_v} = \begin{pmatrix} \sin(\gamma)[\mathbf{z}_v + \mathbf{z}_v^T] & \cos(\gamma)[\mathbf{z}_v - \mathbf{z}_v^T] \\ \cos(\gamma)[\mathbf{z}_v^T - \mathbf{z}_v] & \sin(\gamma)[\mathbf{z}_v + \mathbf{z}_v^T] \end{pmatrix} \tag{25}$$

  with $\mathbf{z}_v$ defined as (23).

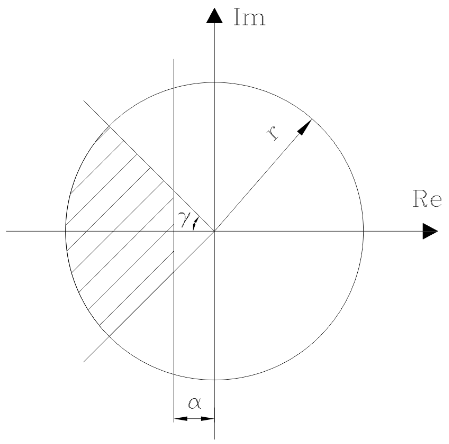

Fig. 1. $S(\alpha, r, \gamma)$ LMI region.

5.2. Generating sequences based on Gershgorin circles. The effect of the perturbation matrix $\Delta\mathbf{A}^{\{i,j\}}$ on the $\mathcal{D}$-stability region $S(\alpha, r, \gamma)$, taking into account (13), is described by

$$\begin{cases}\boldsymbol{\mathcal{D}}_i < 0,\\ \boldsymbol{\mathcal{D}}_j < 0,\end{cases} \Leftrightarrow \begin{cases}\boldsymbol{\mathcal{D}}_i < 0,\\ \boldsymbol{\mathcal{D}}_i - \Delta\boldsymbol{\mathcal{D}}^{\{i,j\}} < 0,\end{cases} \Leftrightarrow \begin{cases}\boldsymbol{\mathcal{D}}_j + \Delta\boldsymbol{\mathcal{D}}^{\{i,j\}} < 0,\\ \boldsymbol{\mathcal{D}}_j < 0,\end{cases} \tag{26}$$

where $\Delta\boldsymbol{\mathcal{D}}^{\{i,j\}} = \Delta\mathbf{F}\left(\Delta\mathbf{A}^{\{i,j\}}, \mathbf{X}\right)$ is the perturbation matrix which represents the effect of $\Delta\mathbf{A}^{\{i,j\}}$ on the LMI of vertex j.

$\Delta\boldsymbol{\mathcal{D}}^{\{i,j\}}$ can be computed by replacing (12) in (23) and setting $\boldsymbol{\Gamma}_i = \boldsymbol{\Gamma}_j = \boldsymbol{\Gamma}^{\{i,j\}}$:

$$\begin{aligned}\mathbf{z}_i = \mathbf{A}_i\mathbf{X} + \mathbf{B}\boldsymbol{\Gamma}^{\{i,j\}} &= (\mathbf{A}_j + \Delta\mathbf{A}^{\{i,j\}})\mathbf{X} + \mathbf{B}\boldsymbol{\Gamma}^{\{i,j\}}\\ &= \mathbf{z}_j + \Delta\mathbf{A}^{\{i,j\}}\mathbf{X}.\end{aligned} \tag{27}$$

Taking into account that

$$\Delta\mathbf{z}^{\{i,j\}} = \Delta\mathbf{A}^{\{i,j\}}\mathbf{X} \tag{28}$$

and substituting (27) in the LMIs (22), (24) and (25), we obtain

- Region α: stability region,

$$\boldsymbol{\mathcal{D}}_{\alpha_j} = \boldsymbol{\mathcal{D}}_{\alpha_i} + \Delta\boldsymbol{\mathcal{D}}_{\alpha}^{\{i,j\}}, \tag{29}$$

where

$$\Delta\boldsymbol{\mathcal{D}}_{\alpha}^{\{i,j\}} = \Delta\mathbf{z}^{\{i,j\}} + (\Delta\mathbf{z}^{\{i,j\}})^T; \tag{30}$$

- Region r: disc region,

$$\boldsymbol{\mathcal{D}}_{r_j} = \boldsymbol{\mathcal{D}}_{r_i} + \Delta\boldsymbol{\mathcal{D}}_{r}^{\{i,j\}}, \tag{31}$$

where

$$\Delta\boldsymbol{\mathcal{D}}_{r}^{\{i,j\}} = \begin{pmatrix} 0 & \Delta\mathbf{z}^{\{i,j\}}\\ (\Delta\mathbf{z}^{\{i,j\}})^T & 0\end{pmatrix}; \tag{32}$$

- Region γ: conic region,

$$\boldsymbol{\mathcal{D}}_{\gamma_j} = \boldsymbol{\mathcal{D}}_{\gamma_i} + \Delta\boldsymbol{\mathcal{D}}_{\gamma}^{\{i,j\}}, \tag{33}$$

where

$$\Delta\boldsymbol{\mathcal{D}}_{\gamma}^{\{i,j\}} = \begin{pmatrix}\sin(\gamma)[\Delta\mathbf{z}^{\{i,j\}} + (\Delta\mathbf{z}^{\{i,j\}})^T] & \cos(\gamma)[\Delta\mathbf{z}^{\{i,j\}} - (\Delta\mathbf{z}^{\{i,j\}})^T]\\ \cos(\gamma)[(\Delta\mathbf{z}^{\{i,j\}})^T - \Delta\mathbf{z}^{\{i,j\}}] & \sin(\gamma)[\Delta\mathbf{z}^{\{i,j\}} + (\Delta\mathbf{z}^{\{i,j\}})^T]\end{pmatrix}. \tag{34}$$

Thus, the perturbation matrix is

$$\begin{aligned}\Delta\mathbf{F}^{\{i,j\}} &= \Delta\boldsymbol{\mathcal{D}}^{\{i,j\}}\\ &= \mathtt{diag}(\Delta\boldsymbol{\mathcal{D}}_{\alpha}^{\{i,j\}}, \Delta\boldsymbol{\mathcal{D}}_{r}^{\{i,j\}}, \Delta\boldsymbol{\mathcal{D}}_{\gamma}^{\{i,j\}}).\end{aligned} \tag{35}$$

Finally, to generate the sequence of vertex pairs based on the Gershgorin circles, it is required to compute $\Delta r_{\max}^{\{i,j\}}$ (17) for each vertex pair.

6. Simulation results

The results obtained in the following examples are based on the methodology explained in Sections 3 and 4 when the pole-placement specification is considered, as detailed in Section 5.

Example 3. (*Academic system*) The methodology for finding two combinable vertices is detailed for an academic example. Consider a system that has 5 vertices, 3 states and 3 inputs. Assume that the state matrices $\mathbf{A}_v$ are

$$\mathbf{A}_1 = \begin{bmatrix} 3.79 & 2.07 & -9.41\\ 8.81 & 5.01 & -8.70\\ -0.06 & -0.20 & 7.76\end{bmatrix},$$

$$\mathbf{A}_2 = \begin{bmatrix} -2.80 & 5.17 & -4.85\\ -2.56 & -6.78 & -0.36\\ -7.50 & -3.08 & 2.87\end{bmatrix},$$

$$\mathbf{A}_3 = \begin{bmatrix} -0.56 & 2.27 & -2.05\\ 8.63 & 4.52 & 6.46\\ -4.71 & -5.85 & 2.24\end{bmatrix},$$

$$\mathbf{A}_4 = \begin{bmatrix} 0.10 & 6.81 & 8.92\\ 0.87 & -1.45 & -3.55\\ -0.16 & 1.83 & -7.48\end{bmatrix},$$

$$\mathbf{A}_5 = \begin{bmatrix} 2.26 & -6.29 & 5.55\\ 2.68 & -6.49 & 2.12\\ -2.17 & 6.26 & -3.80\end{bmatrix},$$

and the input matrix $\mathbf{B}$ is

$$\mathbf{B} = \begin{bmatrix} 0.23 & 0.26 & -0.09 \\ -0.10 & 0.50 & 0.92 \\ 0.21 & -0.91 & -0.19 \end{bmatrix}$$

According to Algorithm 1, the steps used to find two combinable vertices for this system are described below.

1. The LMIs (21) have been modelled using YALMIP (Lofberg, 2004) and solved with SeDuMi (Sturm, 1999) for the $\mathcal{D}$-stable region $S(\alpha, r, \gamma) = S(1, 18.75, 0.6)$. The matrix $\mathbf{X}$ and the controller gains $\mathbf{K}_v$ obtained as a feasible solution of (21) are

$$\mathbf{X} = \begin{bmatrix} 24.2 & -3.4 & 3.5 \\ -3.4 & 5.4 & -0.2 \\ 3.5 & -0.2 & 2.3 \end{bmatrix} \cdot 10^{-4},$$

$$\mathbf{K}_1 = \begin{bmatrix} -65.9 & -79.3 & -16.3 \\ -11.6 & -14.1 & 9.8 \\ -11.8 & -10.1 & 1.0 \end{bmatrix},$$

$$\mathbf{K}_2 = \begin{bmatrix} -12.6 & -37.8 & 69.5 \\ -12.7 & -13.7 & 24.4 \\ 6.5 & 6.8 & -2.4 \end{bmatrix},$$

$$\mathbf{K}_3 = \begin{bmatrix} -35.7 & -63.8 & -44.9 \\ -12.5 & -18.3 & 8.3 \\ -7.3 & -9.2 & -19.2 \end{bmatrix},$$

$$\mathbf{K}_4 = \begin{bmatrix} -33.7 & -45.9 & 43.4 \\ -8.3 & -9.2 & 7.9 \\ -1.6 & -2.3 & 7.1 \end{bmatrix},$$

$$\mathbf{K}_5 = \begin{bmatrix} -49.2 & -72.7 & 24.3 \\ -10.4 & -3.5 & 3.2 \\ -6.1 & -8.8 & 5.7 \end{bmatrix}.$$

2. $\Delta\boldsymbol{\mathcal{D}}^{\{i,j\}}$ has been built for each vertex pair:

$$\Delta\boldsymbol{\mathcal{D}}^{\{i,j\}} = \mathtt{diag}(\Delta\boldsymbol{\mathcal{D}}_\alpha^{\{i,j\}}, \Delta\boldsymbol{\mathcal{D}}_r^{\{i,j\}}, \Delta\boldsymbol{\mathcal{D}}_\gamma^{\{i,j\}}),$$

where $\Delta\boldsymbol{\mathcal{D}}_\alpha^{\{i,j\}}$, $\Delta\boldsymbol{\mathcal{D}}_r^{\{i,j\}}$ and $\Delta\boldsymbol{\mathcal{D}}_\gamma^{\{i,j\}}$ are defined in (30), (32) and (34), respectively. The number of vertex pairs is $n_{2vtx} = (n_v - 1)n_v/2 = 10$. For example, to compute $\Delta\boldsymbol{\mathcal{D}}^{\{1,2\}}$, it is necessary to determine $\Delta\mathbf{A}^{\{1,2\}}$ using (12):

$$\Delta\mathbf{A}^{\{1,2\}} = \begin{bmatrix} 6.6 & -3.1 & -4.6 \\ 11.4 & 11.8 & -8.3 \\ 7.4 & 2.8 & 4.9 \end{bmatrix}.$$

Then $\Delta\mathbf{z}^{\{1,2\}}$ is generated using (28):

$$\Delta\mathbf{z}^{\{1,2\}} = \begin{bmatrix} 15.4 & -3.8 & 1.3 \\ 20.6 & 2.7 & 1.7 \\ 18.7 & -1.1 & 3.6 \end{bmatrix} \cdot 10^{-3}.$$

Finally, $\Delta\boldsymbol{\mathcal{D}}^{\{1,2\}}$ is built using (30), (32) and (34).

3. $\Delta r_{\max}^{\{i,j\}}$ (17) has been calculated for each matrix $\Delta\boldsymbol{\mathcal{D}}^{\{i,j\}}$.

4. Finally, as shown previously in Example 2, the values $\Delta r_{\max}^{\{i,j\}}$ have been sorted to produce the Gershgorin circles-based sequence given in Table 1 . Note that only a feasible combination of vertices for the selected region exists, which corresponds to $4-5$, highlighted in boldface in Table 1.

For this system, the solution generated by the solver, when the same controller gain is used for vertices 4 and 5, is

$$\mathbf{X} = \begin{bmatrix} 170.6 & 9.5 & 27.1 \\ 9.5 & 9.9 & -2.2 \\ 27.1 & -2.2 & 10.6 \end{bmatrix} \cdot 10^{-3},$$

$$\mathbf{K}_1 = \begin{bmatrix} -38.6 & -165.6 & -135.1 \\ -6.5 & -28.1 & -12.0 \\ -10.7 & -15.1 & -5.0 \end{bmatrix},$$

$$\mathbf{K}_2 = \begin{bmatrix} -15.6 & -124.5 & 113.0 \\ -12.6 & -29.7 & 29.9 \\ 5.4 & 0.2 & 5.2 \end{bmatrix},$$

$$\mathbf{K}_3 = \begin{bmatrix} 0.2 & -207.2 & -188.8 \\ -4.4 & -37.6 & -18.5 \\ -5.0 & -22.8 & -17.4 \end{bmatrix},$$

$$\mathbf{K}_4 = \mathbf{K}_5 = \begin{bmatrix} -47.8 & -104.9 & 73.4 \\ -10.7 & -16.7 & 12.1 \\ -2.1 & -9.0 & 6.2 \end{bmatrix}.$$

Hence, only two iterations will be required to find the combinable vertices using the sequence determined by the Gershgorin circles. By contrast, the mean number of iterations needed when random sequences are used is 5.5. A random sample sequence that would require five iterations is also shown in Table 1. ♦

Example 4. (*Set of academic systems*) A set of academic systems has been generated to investigate the efficiency

Table 1. Sequence to combine vertex pairs in Example 3.

Gershgorin circles		Random
Sequence (s_{GC})	$\Delta r_{\max}^{\{i,j\}}$ [$\cdot 10^{-2}$]	sequence (s_R)
$1-3$	2.8	$2-4$
$\mathbf{4-5}$	3.0	$1-2$
$1-5$	3.1	$2-5$
$1-4$	3.6	$1-5$
$3-4$	3.7	$\mathbf{4-5}$
$3-5$	3.9	$2-3$
$2-4$	4.4	$3-4$
$2-5$	5.7	$1-4$
$2-3$	5.8	$3-5$
$1-2$	7.2	$1-3$

of the proposed algorithm. The procedure that has been followed is described below:

- n_{sys} artificial systems with n_v vertices have been generated, using the parameters detailed in Table 2. The chosen $\mathcal{D}$-stability region is $S(\alpha, r, \gamma) = S(1, r_{\min}, 0.6)$ (21), where $r_{\min}$ depends on the specific system and allows restricting the number of existing combinable vertex pairs n_m.

- Algorithm 1 has been applied to every system by computing s_{GC} for each vertex pair, along with a randomly generated sequence s_R. Note that the matrix $\mathbf{X}$ used is the solution obtained when a different controller is designed for each vertex.

- Combinable vertices have been searched by checking whether a feasible solution of (21) exists when vertices are combined following the sequences s_{GC} and s_R previously determined. The number of attempts required to find combinable vertices has been recorded.

Figure 2 shows the results obtained for a set of 250 systems with 3 states, 3 inputs and 5 vertices, and provides the cumulative probability of finding combinable vertices at each iteration. For instance, for 68.8% of the systems a maximum of 2 iterations was needed in order to find combinable vertices using the Gershgorin circles-based sequence. On the other hand, for only 29.6% of the systems, combinable vertices could be found using the same number of iterations, when applying the random sequence.

The results show that the probability of finding combinable vertices at each iteration using the sequence generated by the proposed methodology is higher than following a random sequence. ♦

Example 5. (*Different system structures*) The objective of this example is to demonstrate that the efficiency of the proposed methodology is independent of the system parameters. For this reason, different system structures have been considered.

Table 2. Parameters of Example 4.

Parameter	Symbol	Value
No. of systems	n_{sys}	250
No. of states	n_x	3
No. of inputs	n_u	3
No. of vertices	n_v	5
No. of combinations (max)	n_m	2
Elements of $\mathbf{A}_v$	a_{kl_v}	$[-10, 10]$
Elements of $\mathbf{B}$	b_{km}	$[-1, 1]$

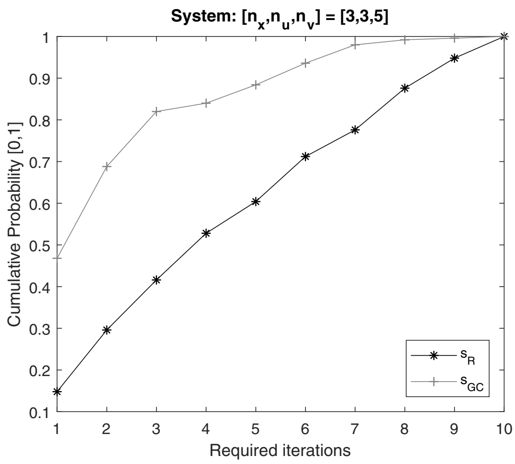

Fig. 2. Behaviour of the proposed approach (Example 4). Systems with 3 states, 3 inputs and 5 vertices.

Four scenarios have been generated (see Table 3), each composed by three configurations which represent an under-actuated, a fully-actuated and an over-actuated system. Each configuration consists of 250 academic systems with a predefined number of states, inputs and vertices. The approach based on the Gershgorin circles has been applied to combine two vertices of each system. The range of values for the matrices $\mathbf{A}_v$ and $\mathbf{B}$ is the same as in Example 4. The maximum number of combinable vertices n_m is 2 and 5 for the systems with 5 and 10 vertices, respectively. The same procedure explained in Example 4 has been applied and the results obtained are presented in Figs. 3–6.

In all scenarios and configurations, the Gershgorin circles-based sequence ($s_{GC} := n_s \in \{1 - n_m\}$) has a higher probability of finding combinable vertices than the random sequence s_R. For about 40% of the systems, combinable vertices were found at the first iteration using s_{GC} when the system has 3 states (see Figs. 3 and 5). On the other hand, for only about 12% of the systems, combinable vertices were found at the first iteration when a random sequence s_R was used.

Hence, according to the results, the main advantage of the proposed methodology is that fewer iterations are required to find combinable vertices. Figures 3 and 4 show that for a system with 5 vertices, only 4 or 5 iterations

Table 3. Characteristics of the scenarios in Example 5.

Scenario	n_x	n_u	n_v
1	3	{2,3,5}	5
2	6	{5,6,8}	5
3	3	{2,3,5}	10
4	6	{5,6,8}	10

are required in order to reach an 80% of probability of finding combinable vertices, and less than 10 iterations are necessary when 10 vertices are considered (see Figs. 5 and 6).

Note that the results denoted by $n_s \in \{\phi\}$ show the behaviour of the algorithm when the systems have exactly ϕ combinable vertex pairs. For instance, in Fig. 3 $n_s \in \{1\}$ denotes the results obtained for systems which have exactly one pair of combinable vertices. As expected, when there exist multiple combinable vertices that are feasible, the probability of finding them becomes higher. However, the worst case of $n_s \in \{1\}$ is still better than when using a random sequence (s_R).

It can be concluded that the performance of the proposed algorithm does not depend on the number of states and inputs, whereas the number of vertices determines the problem complexity. The sequence generated by the proposed methodology increases the likelihood of finding combinable vertices at each iteration. It must be highlighted that the time to execute Algorithm 1 is shorter by fewer orders of magnitude than the one for the brute-force procedure.

♦

Example 6. (*Two-link robot*) The developed algorithm has been applied to the two-link robot described by Tseng *et al*. (2001) as a Takagi–Sugeno fuzzy system with nine rules. Therefore, nine controllers should be designed to regulate the robot motion. In this case, the matrix $\mathbf{B}$ is not constant, consequently, a prefiltering of the inputs is required to obtain a constant input matrix $\bar{\mathbf{B}}$ (Apkarian *et al*., 1995):

$$\dot{\mathbf{x}}_u(t) = -\omega\mathbf{x}_u(t) + \omega\mathbf{u}(t), \tag{36}$$

where $\mathbf{u}(t)$ is the input vector, $\mathbf{x}_u(t)$ is a state vector which represents the filtered input and ω allows us to define the filter bandwidth. The selected ω is 45 s^{-1} because the filter bandwidth must be chosen larger than the desired system bandwidth (Apkarian *et al*., 1995).

The selected $\mathcal{D}$-stable region (21) is $S(\alpha, r, \gamma) = S(1, 34.2, 0.6)$ and the first ten elements of the Gershgorin circles-based sequence (s_{GC}), when Algorithm 1 is applied, are presented in Table 4.

The results show that all the combinable vertex pairs are situated in the first sequence positions. It is worth noting that, by repeatedly combining vertex pairs following the obtained sequence, it is possible to reduce the number of controller gains not only from nine to eight, but also from nine to five, since by combining vertices 3–7, 1–9, 2–8 and 4–6, the feasibility of the designed LMIs is kept. It must be highlighted that the obtained solution might not have the minimal number of combinable tuples of vertices. Nevertheless, the proposed algorithm allows achieving a substantial reduction in the number of controller gains through a faster search of combinable vertices. ♦

7. Conclusions

In this paper, a heuristic algorithm to decrease the number of controller gains when the design of an LPV state-feedback controller takes the form of parametrized LMIs has been developed. The proposed technique is based on the use of Gershgorin circles to determine how much the LMIs are perturbed when a common controller gain is chosen. This quantification allows us to produce a sequence which provides information on the likelihood that two vertices are combinable. The approach is not limited to reduction in controller gains, since it can be used for any LMI-based design, e.g., for state observers or fault estimators.

The application examples have shown that the efficiency of the algorithm does not depend on the number of states and inputs, and the complexity of the problem is linked to the number of vertices of the system. The results show that, by means of the proposed approach, fewer iterations are required to find combinable vertices. Furthermore, the proposed methodology is not only limited to reduce from n_v to $n_v - 1$, since a higher reduction can be achieved, although the obtained solution might not yield the minimal number of combinable tuples of vertices.

This approach cannot determine whether a feasible solution of (9) exists. However, the vertex pairs of the Gershgorin circles-based sequence increase their probability to become combinable vertices, when the specification constraints are relaxed, e.g., increasing r of the $\mathcal{D}$-stability region. It must also be pointed out that the metric based on Gershgorin circles is conservative

Table 4. Gershgorin circles-based sequence (s_{GC}) in Example 6.

s_{GC}	$\Delta r_{\max}^{\{i,j\}}$ [$\times 10^{-3}$]	Feasible?
3–7	1.0	Yes
1–9	1.2	Yes
2–8	2.7	Yes
4–6	4.7	Yes
6–8	13.7	Yes
4–8	15.6	No
2–6	15.7	No
2–4	18.0	No
5–9	48.5	No
1–5	48.5	No
⋮	⋮	⋮

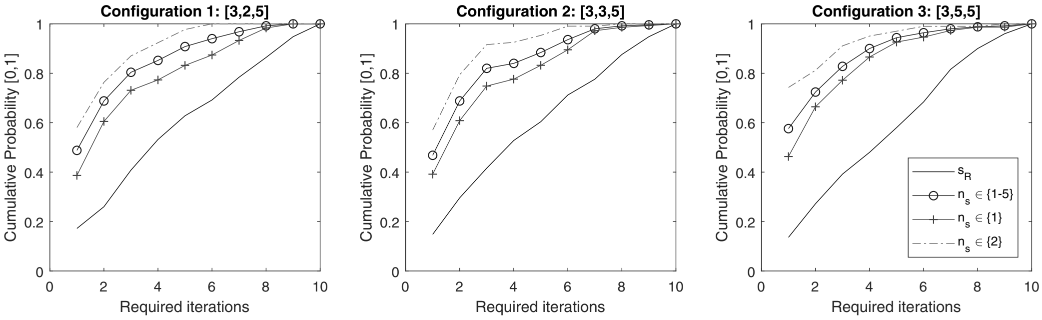

Fig. 3. Scenario 1 (Example 5): behaviour of the proposed approach. Configurations: 3 states, {2,3,5} inputs and 5 vertices.

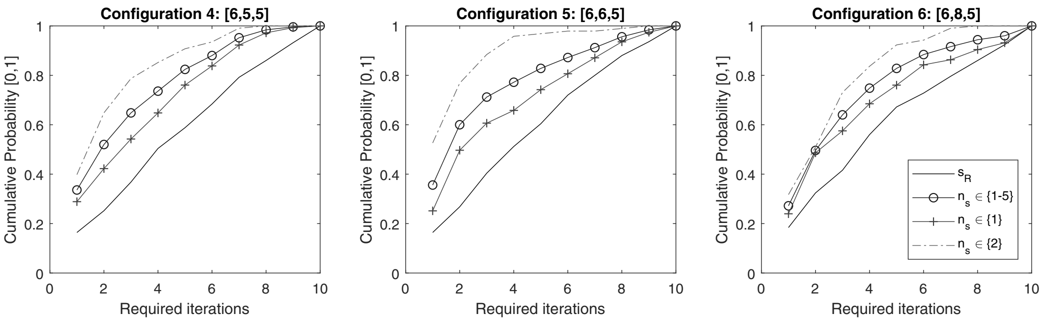

Fig. 4. Scenario 2 (Example 5): behaviour of the proposed approach. Configurations: 6 states, {5,6,8} inputs and 5 vertices.

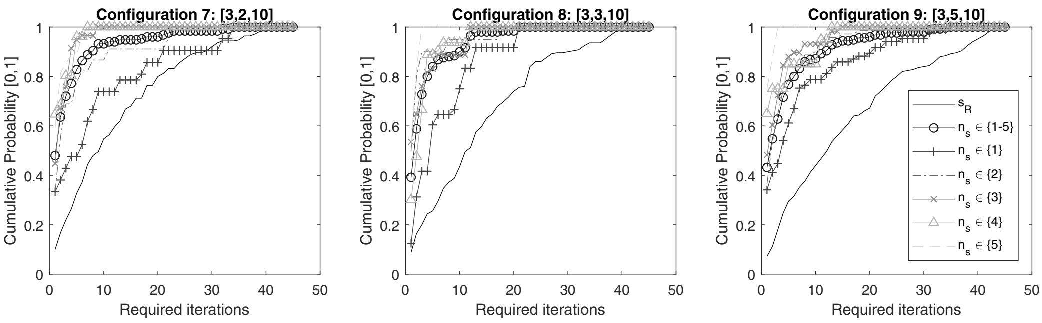

Fig. 5. Scenario 3 (Example 5): behaviour of the proposed approach. Configurations: 3 states, {2,3,5} inputs and 10 vertices.

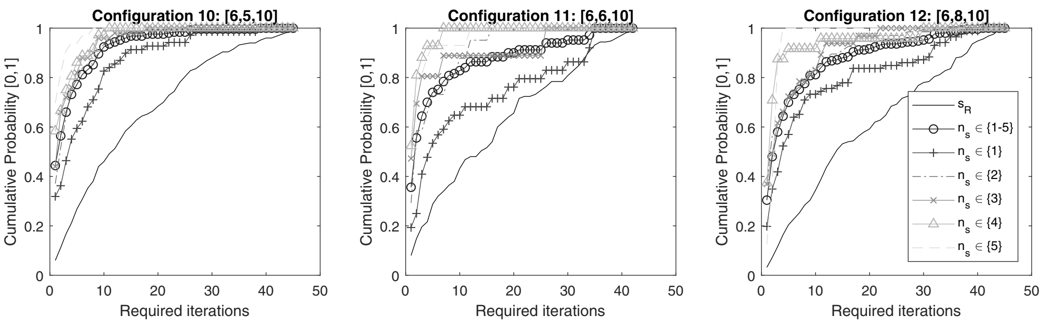

Fig. 6. Scenario 4 (Example 5): behaviour of the proposed approach. Configurations: 6 states, {5,6,8} inputs and 10 vertices.

because it considers the "worst-case scenario." Hence, it is necessary to develop new metrics in order to improve the sequence generated by Algorithm 1.

The proposed methodology cannot quantify how much the LMIs are perturbed when combinations of more than two vertices are involved. Hence, the maximum vertex reduction which can be achieved is $n_v/2$ and $(n_v + 1)/2$, when n_v is even and odd, respectively. All these are some open issues which deserve further investigation in future research, the main one being extension of the proposed approach to search for generic tuples of combinable vertices.

Acknowledgment

This work has been funded by the Spanish State Research Agency (AEI) and the European Regional Development Fund (ERFD) through the projects DEOCS (ref. MINECO DPI2016-76493) and SCAV (ref. MINECO DPI2017-88403-R), by AGAUR through the contract FI-AGAUR (ref. 2017FI-B00285) and by the DGR of Generalitat de Catalunya (SAC group Ref. 2017/SGR/482). This work has also been supported by the Spanish State Research Agency through the Maria de Maeztu Seal of Excellence to IRI (MDM-2016-0656) and the grant Juan de la Cierva—Formación (FJCI-2016-29019).

References

Apkarian, P., Gahinet, P. and Becker, G. (1995). Self-scheduled $\mathcal{H}_\infty$ control of linear parameter-varying systems: A design example, *Automatica* **31**(9): 1251–1261.

Apkarian, P. and Tuan, H.D. (2000). Parameterized LMIs in control theory, *SIAM Journal on Control and Optimization* **38**(4): 1241–1264.

Boyd, S., El Ghaoui, L., Feron, E. and Balakrishnan, V. (1994). *Linear Matrix Inequalities in System and Control Theory*, SIAM, Philadelphia, PA.

Chilali, M. and Gahinet, P. (1996). $\mathcal{H}_\infty$ design with pole placement constraints: An LMI approach, *IEEE Transactions on Automatic Control* **41**(3): 358–367.

Chitraganti, S., Tóth, R., Meskin, N. and Mohammadpour, J. (2017). Stochastic model predictive tracking of piecewise constant references for LPV systems, *IET Control Theory & Applications* **11**(12): 1862–1872.

Curran, P.F. (2009). On a variation of the Gershgorin circle theorem with applications to stability theory, *IET Irish Signals and Systems Conference (ISSC 2009), Dublin, Ireland*, pp. 1–5.

El-Sakkary, A. (1985). The gap metric: Robustness of stabilization of feedback systems, *IEEE Transactions on Automatic Control* **30**(3): 240–247.

Fergani, S., Menhour, L., Sename, O., Dugard, L. and D'Andréa-Novel, B. (2017). Integrated vehicle control through the coordination of longitudinal/lateral and vertical dynamics controllers: Flatness and LPV/$\mathcal{H}_\infty$-based design, *International Journal of Robust and Nonlinear Control* **27**(18): 4992–5007.

Fleischmann, S., Theodoulis, S., Laroche, E., Wallner, E. and Harcaut, J.-P. (2016). A systematic LPV/LFR modelling approach optimized for linearised gain scheduling control synthesis, *AIAA Modeling and Simulation Technologies Conference, San Diego, CA, USA*, p. 1921.

Gahinet, P., Apkarian, P. and Chilali, M. (1996). Affine parameter-dependent Lyapunov functions and real parametric uncertainty, *IEEE Transactions on Automatic Control* **41**(3): 436–442.

Ho, W.K., Lee, T.H., Xu, W., Zhou, J.R. and Tay, E.B. (2000). The direct Nyquist array design of PID controllers, *IEEE Transactions on Industrial Electronics* **47**(1): 175–185.

Hoffmann, C. and 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**(2): 416–433.

Inthamoussou, F.A., Bianchi, F.D., De Battista, H. and Mantz, R.J. (2014). LPV wind turbine control with anti-windup features covering the complete wind speed range, *IEEE Transactions on Energy Conversion* **29**(1): 259–266.

Jabali, M. B.A. and Kazemi, M.H. (2017). A new LPV modeling approach using PCA-based parameter set mapping to design a PSS, *Journal of Advanced Research* **8**(1): 23–32.

Kwiatkowski, A., Boll, M.-T. and Werner, H. (2006). Automated generation and assessment of affine LPV models, *45th IEEE Conference on Decision and Control, San Diego, CA, USA*, pp. 6690–6695.

Kwiatkowski, A. and 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.

Lofberg, J. (2004). YALMIP: A toolbox for modeling and optimization in MATLAB, *IEEE International Symposium on Computer Aided Control Systems Design, New Orleans, LA, USA*, pp. 284–289.

López-Estrada, F.-R., Ponsart, J.-C., Astorga-Zaragoza, C.-M., Camas-Anzueto, J.-L. and Theilliol, D. (2015). Robust sensor fault estimation for descriptor-LPV systems with unmeasurable gain scheduling functions: Application to an anaerobic bioreactor, *International Journal of Applied Mathematics and Computer Science* **25**(2): 233–244, DOI: 10.1515/amcs-2015-0018.

Luspay, T., Péni, T., Gőzse, I., Szabó, Z. and Vanek, B. (2018). Model reduction for LPV systems based on approximate modal decomposition, *International Journal for Numerical Methods in Engineering* **113**(6): 891–909.

Marcos, A. and Balas, G.J. (2004). Development of linear-parameter-varying models for aircraft, *Journal of Guidance, Control, and Dynamics* **27**(2): 218–228.

Mohammadpour, J. and Scherer, C.W. (2012). *Control of Linear Parameter Varying Systems with Applications*, Springer, New York, NY.

Montagner, V., Oliveira, R., Leite, V.J. and Peres, P.L.D. (2005). LMI approach for $\mathcal{H}_\infty$ linear parameter-varying state feedback control, *IEE Proceedings: Control Theory and Applications* **152**(2): 195–201.

Nguyen, A.-T., Chevrel, P. and Claveau, F. (2018). Gain-scheduled static output feedback control for saturated LPV systems with bounded parameter variations, *Automatica* **89**: 420–424.

Packard, A. (1994). Gain scheduling via linear fractional transformations, *Systems & Control Letters* **22**(2): 79–92.

Pemmaraju, S. and Skiena, S. (2003). *Computational Discrete Mathematics: Combinatorics and Graph Theory with Mathematica*, Cambridge University Press, New York, NY.

Quarteroni, A., Sacco, R. and Saleri, F. (2010). *Numerical Mathematics*, Vol. 37, Springer Science & Business Media, New York, NY.

Rahideh, A. and Shaheed, M. (2007). Mathematical dynamic modelling of a twin-rotor multiple input-multiple output system, *Proceedings of the Institution of Mechanical Engineers I: Journal of Systems and Control Engineering* **221**(1): 89–101.

Rizvi, S. Z., Mohammadpour, J., Tóth, R. and Meskin, N. (2016). A kernel-based PCA approach to model reduction of linear parameter-varying systems, *IEEE Transactions on Control Systems Technology* **24**(5): 1883–1891.

Rotondo, D. (2017). *Advances in Gain-Scheduling and Fault Tolerant Control Techniques*, Springer, Cham.

Rotondo, D., Nejjari, F. and Puig, V. (2013). Quasi-LPV modeling, identification and control of a twin rotor MIMO system, *Control Engineering Practice* **21**(6): 829–846.

Rotondo, D., Nejjari, F. and Puig, V. (2014). Robust state-feedback control of uncertain LPV systems: An LMI-based approach, *Journal of the Franklin Institute* **351**(5): 2781–2803.

Rotondo, D., Puig, V., Nejjari, F. and Romera, J. (2015a). A fault-hiding approach for the switching quasi-LPV fault-tolerant control of a four-wheeled omnidirectional mobile robot, *IEEE Transactions on Industrial Electronics* **62**(6): 3932–3944.

Rotondo, D., Puig, V., Nejjari, F. and Witczak, M. (2015b). Automated generation and comparison of Takagi–Sugeno and polytopic quasi-LPV models, *Fuzzy Sets and Systems* **27**(C): 44–64.

Rugh, W.J. and Shamma, J.S. (2000). Research on gain scheduling, *Automatica* **36**(10): 1401–1425.

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.

Sun, X.-D. and Postlethwaite, I. (1998). Affine LPV modelling and its use in gain-scheduled helicopter control, *UKACC International Conference on Control '98, Swansea, UK*, Vol. 2, pp. 1504–1509.

Theis, J., Seiler, P. and Werner, H. (2017). LPV model order reduction by parameter-varying oblique projection, *IEEE Transactions on Control Systems Technology* **26**(3): 773–784.

Theodoulis, S. and Duc, G. (2009). Missile autopilot design: Gain-scheduling and the gap metric, *Journal of Guidance, Control, and Dynamics* **32**(3): 986–996.

Tseng, C.-S., Chen, B.-S. and Uang, H.-J. (2001). Fuzzy tracking control design for nonlinear dynamic systems via TS fuzzy model, *IEEE Transactions on Fuzzy Systems* **9**(3): 381–392.

Vinnicombe, G. (1996). The robustness of feedback systems with bounded complexity controllers, *IEEE Transactions on Automatic Control* **41**(6): 795–803.

White, A.P., Zhu, G. and Choi, J. (2013). *Linear Parameter-Varying Control for Engineering Applications*, Springer, London.

Zhao, P. and Nagamune, R. (2017). Switching LPV controller design under uncertain scheduling parameters, *Automatica* **76**: 243–250.

Zhou, M., Rodrigues, M., Shen, Y. and Theilliol, D. (2018). $\mathcal{H}_-/\mathcal{H}_\infty$ fault detection observer design for a polytopic LPV system using the relative degree, *International Journal of Applied Mathematics and Computer Science* **28**(1): 83–95, DOI: 10.2478/amcs-2018-0006.

Zribi, A., Chtourou, M. and Djemal, M. (2016). A systematic determination approach of model's base using gap metric for nonlinear systems, *Journal of Dynamic Systems, Measurement, and Control* **138**(3): 031008.

Adrián Sanjuan received the BS and MS degrees from the Polytechnic University of Catalonia (UPC), Spain. Since 2017, he has been a PhD student at the Research Center for Supervision, Safety and Automatic Control (CS2AC) at the UPC. His main research interests include artificial intelligence, identification, fault detection and isolation (FDI), and fault tolerant control (FTC).

Damiano Rotondo received the PhD degree from the Polytechnic University of Catalonia, Spain, in 2016. Since February 2018, he has been a Juan de la Cierva fellow at the Institute of Robotics and Industrial Informatics (IRI) in the Spanish National Research Council (CSIC). His main research interests include gain-scheduled control systems, fault detection and isolation (FDI), and fault tolerant control (FTC).

Fatiha Nejjari is an associate professor with the Department of Automatic Control, Polytechnic University of Catalonia (UPC). She is also a member of the Advanced Control Systems (SAC) research group of the Research Center for Supervision, Safety and Automatic Control (CS2AC) at the UPC. Her main research areas include LPV control, fault detection and isolation, and fault tolerant control of dynamic systems. She has published several papers in journals and international conferences, and has participated in several European projects and networks related to these topics.

Ramon Sarrate received the MS and PhD degrees in industrial engineering from the Polytechnic University of Catalonia (UPC), Terrassa, Spain, in 1994 and 2003, respectively. He is currently an assistant professor at the Department of Automatic Control, UPC. His present research interests include model-based fault diagnosis and hybrid systems. He has been involved in several national and European research projects, and has published several papers in scientific journals and international conference proceedings.

Received: 9 November 2018
Revised: 19 March 2019
Accepted: 30 May 2019