Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Complexity of Implementation and Synthesis in Linear Parameter-Varying Control

Christian Hoffmann * **Herbert Werner** *

* *Hamburg University of Technology, Hamburg, 21073 Germany (e-mail: {christian.hoffmann, h.werner}@ tuhh.de).*

Abstract: In this paper an analysis of the complexity involved in the implementation and synthesis of linear parameter-varying (LPV) controllers is presented. Its purpose is to provide guidance in the selection of a synthesis approach for practical LPV control problems and reveal directions for further research with respect to complexity issues in LPV control. Standard methods are classified into polytopic, linear fractional transformation and gridding-based techniques with an emphasis on output-feedback synthesis. Carried out as a convex optimization problem via finitely many linear matrix inequalities (LMIs) for both parameter-independent and parameter-dependent Lyapunov functions (PiDLF/PDLF), the complexity of LPV controller existence conditions is assessed in terms of the number of decision variables and total size of the LMI. The implementation complexity is assessed in terms of the number of arithmetic operations required to compute the parameter-varying state space matrices of the controller, as well as the memory requirements to store associated variables. The results are applied to the LPV controller synthesis for a three-degrees-of-freedom robotic manipulator and the charge control of a spark-ignited engine, for which multiple models, as well as associated synthesis results have been reported in the literature.

1. INTRODUCTION

The field of nonlinear control using the quasi-linear parameter-varying (quasi-LPV) modeling and control paradigm has matured since its introduction in the seminal work of Shamma and Athans (1990). LPV controller synthesis is attractive since linear time-invariant (LTI) control methodologies, such as sensitivity shaping and modeling tools, have been extended to and are available for the LPV framework. Early synthesis methods were limited to slow parameter variations Shamma and Athans (1991), but over the years methods have been derived, capable of allowing for arbitrarily fast or rate limited parameter variations (Apkarian et al., 1995; Apkarian and Gahinet, 1995; Wu et al., 1996; Scorletti and Ghaoui, 1998; Apkarian and Adams, 1998; Scherer, 2000, 2001; Wu, 2001). Incorporating knowledge on known bounds on the parameters' rate of variation is known to reduce conservatism and has been explored, e.g., in Apkarian and Adams (1998); Wu and Dong (2006).

Most controller synthesis approaches involve the formulation of matrix inequalities, whose efficient solution requires certain convexification techniques. The applicable techniques strongly depend on the type of LPV model that is used. Common LPV representations are generally, rationally or affinely parameterized state space representations. The suitability of one of the individual LPV modeling frameworks is influenced by the nonlinearities found in the considered systems. Complex nonlinear functions of states, inputs and outputs, look-up tables in conjunction with a limited number of measured scheduling signals can be handled well within a general LPV framework and a gridding approach for analysis and synthesis. Linear fractional transformation (LFT)-based or polytopic techniques require to mask nonlinearities in scheduling parameters. This can introduce conservatism through overbounding (Kwiatkowski and Werner, 2008). In turn, this may also allow for more complex systems, when, e.g., in the LFT-based approach, multiplier constraints are employed to reduce synthesis and implementation complexity. Apart from these fundamental differences, assessing the suitability of a synthesis approach via its complexity in synthesis and implementation is not always straight-forward. Therefore, this paper aims at providing a tool to estimate the costs *a priori*—provided the respective models are available.

Outline: Section 2 briefly reviews different LPV model descriptions and associated synthesis techniques. The complexity during online implementation associated with the synthesis approaches is investigated in Section 3. Synthesis complexity is discussed in Section 4 and a tabular summary is presented in Section 5. The results are applied to a three-degrees-of-freedom robotic manipulator, as well as a spark-ignited engine model, for both of which several modeling approaches are motivated and referenced. Section 7 draws conclusions.

2. PRELIMINARIES

Notation: An (upper) LFT is denoted by $\Delta \star \left[\begin{array}{c|c} M_{11} & M_{12} \\ \hline M_{21} & M_{22} \end{array}\right] = M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12}$. The symmetric completion of a matrix is denoted by $\bullet$. Time dependence is regularly dropped, e.g. $\theta = \theta(t)$. Also, ker$A$ denotes the kernel of A. Four types of complexities are considered: The number of arithmetic operations to compute a value A, which at some point is referenced to be computed by some formula $A = f(x)$, is denoted $\mathsf{a}(A)$. The multiplications, additions, divisons, etc., involved are assumed clear from the context by the explicit formula $f(x)$. Similarly, the number of scalar values (memory) required to store the variables, from which A can be computed, is denoted by $\mathsf{m}(A)$. Furthermore, the size of a matrix inequality $\mathcal{L}$ is written as $\mathsf{s}(\mathcal{L})$ (provided only in terms of one dimension, since LMIs are square), whereas the associated number of decision variables is given by $\mathsf{d}(\mathcal{L})$. If, e.g., $\mathcal{L}$ solely contains the matrix variables X, Y, we also write $\mathsf{d}(\mathcal{L}) = \mathsf{d}(X) + \mathsf{d}(Y)$. We use the short-hand notation $\partial X(\rho(t)) := \sum_{i=1}^{n_\rho} \frac{\partial X}{\partial \rho_i(t)} \dot\rho_i(t)$, $\rho(t) : \mathbb{R} \mapsto \mathbb{R}^{n_\rho}$.

2.1 LPV Model Representations

Consider a general LPV plant of the form

$$\mathcal{P}^\rho : \left\{ \begin{bmatrix} \dot x \\ \hdashline z_\mathrm{p} \\ y \end{bmatrix} = \left[\begin{array}{c:cc} \mathscr{A}(\rho) & \mathscr{B}_\mathrm{p}(\rho) & \mathscr{B}_u(\rho) \\ \hdashline \mathscr{C}_\mathrm{p}(\rho) & \mathscr{D}_\mathrm{pp}(\rho) & \mathscr{D}_\mathrm{pu}(\rho) \\ \mathscr{C}_y(\rho) & \mathscr{D}_{y\mathrm{p}}(\rho) & \mathscr{D}_{yu}(\rho) \end{array}\right] \begin{bmatrix} x \\ \hdashline w_\mathrm{p} \\ u \end{bmatrix} \right., \tag{1}$$

where $x\in\mathbb{R}^{n_x}$, $u\in\mathbb{R}^{n_u}$, $y\in\mathbb{R}^{n_y}$, $w_\mathrm{p}\in\mathbb{R}^{n_w}$, $z_\mathrm{p}\in\mathbb{R}^{n_z}$, are the state, input, output and performance signal vectors of the system, respectively. We denote as *scheduling signals* $\rho = [\rho_1(t)\ \rho_2(t)\ \dots\ \rho_{n_\rho}(t)] \in \boldsymbol{\rho}$ measurable quantities that range in some admissible compact set $\boldsymbol{\rho}$, as well as associated bounded rates of variation $\dot\rho =: \sigma =\in \boldsymbol{\sigma}$. For the purpose of rewriting a general LPV system (1), in order to have affine or a rational dependence on a set of *scheduling parameters*, we introduce mappings

$$f^{\rho\to\delta} :\ \mathbb{R}^{n_\rho} \to \mathbb{R}^{n_\delta},\ \rho(t) \mapsto f^{\rho\to\delta}\big(\rho(t)\big) := \delta(t) \text{ and}$$
$$f^{\delta\to\theta} :\ \mathbb{R}^{n_\delta} \to \mathbb{R}^{n_\theta},\ \delta(t) \mapsto f^{\delta\to\theta}\big(\delta(t)\big) := \theta(t).$$

The former is used to denote a transformation into a linear fractional dependence on the parameters δ, whereas the latter is used for mapping the LFT parameters into parameters θ that allow for an affine representation. The linear fractional representation (LFR) of the LPV plant is given by

$$\mathcal{P}^\delta : \left\{ \begin{array}{l} \begin{bmatrix} \dot x \\ \hdashline z_\Delta \\ z_\mathrm{p} \\ y \end{bmatrix} = \left[\begin{array}{c:ccc} A & B_\Delta & B_\mathrm{p} & B_u \\ \hdashline C_\Delta & D_{\Delta\Delta} & D_{\Delta\mathrm{p}} & D_{\Delta u} \\ C_\mathrm{p} & D_{\mathrm{p}\Delta} & D_\mathrm{pp} & D_\mathrm{pu} \\ C_y & D_{y\Delta} & D_{y\mathrm{p}} & D_{yu} \end{array}\right] \begin{bmatrix} x \\ \hdashline w_\Delta \\ w_\mathrm{p} \\ u \end{bmatrix}, \\ w_\Delta = \Delta z_\Delta, \end{array} \right. \tag{2}$$

where $w_\Delta\in\mathbb{R}^{n_\Delta}$, $z_\Delta\in\mathbb{R}^{n_\Delta}$ denote the scheduling channel of the system. We denote the state space model matrices P^δ, such that $\mathcal{P}^\delta = \operatorname{diag}\big(s^{-1}I, \Delta\big) \star P^\delta$. Assume that the LFR is well-posed, i.e., $(I - D_{\Delta\Delta}\Delta)$ is invertible for all admissible parameter values. The vector $\delta(t) = [\delta_1(t)\ \delta_2(t)\ \dots\ \delta_{n_\delta}(t)]$ collects all scheduling parameters assumed to be contained in a compact set $\boldsymbol{\delta}$. The time derivatives are also bounded, such that $\dot\delta(t) =: \upsilon(t) = [\upsilon_1(t)\ \upsilon_2(t)\ \dots\ \upsilon_{n_\delta}(t)]$, with $\upsilon \in \boldsymbol{\upsilon}$ For an LFT-LPV plant Δ is often assumed to have block-diagonal structure $\Delta(t) = \operatorname*{diag}_{i=1}^{n_\delta}\Big(\delta_i I_{r_i^\delta}\Big)$, where $\sum_{i=1}^{n_\delta} r_i^\delta = n_\Delta$. However, it may also be possible to find full block-matrix structured scheduling blocks Δ, (Hoffmann and Werner, 2014).

Using $\delta(t) = f^{\rho\to\delta}\big(\rho(t)\big)$, the general LPV form is recovered by

$$S^\delta(\delta) = \begin{bmatrix} \mathscr{A}(\rho) & \mathscr{B}_\mathrm{p}(\rho) & \mathscr{B}_u(\rho) \\ \mathscr{C}_\mathrm{p}(\rho) & \mathscr{D}_\mathrm{pp}(\rho) & \mathscr{D}_\mathrm{pu}(\rho) \\ \mathscr{C}_y(\rho) & \mathscr{D}_{y\mathrm{p}}(\rho) & \mathscr{D}_{yu}(\rho) \end{bmatrix} = \begin{bmatrix} A & B_\mathrm{p} & B_u \\ C_\mathrm{p} & D_\mathrm{pp} & D_\mathrm{pu} \\ C_y & D_{y\mathrm{p}} & D_{yu} \end{bmatrix}$$
$$+ \begin{bmatrix} B_\Delta \\ D_{\mathrm{p}\Delta} \\ D_{y\Delta} \end{bmatrix} \Delta(I - D_{\Delta\Delta}\Delta)^{-1} [C_\Delta\ D_{\Delta\mathrm{p}}\ D_{\Delta u}]. \tag{3}$$

A plant affine in its parameters can be written as

$$\mathcal{P}^\theta : \left\{ \begin{array}{l} \begin{bmatrix} \dot x \\ \hdashline z_\Theta \\ z_\mathrm{p} \\ y \end{bmatrix} = \left[\begin{array}{c:ccc} A & B_\Theta & B_\mathrm{p} & B_u \\ \hdashline C_\Theta & 0 & D_{\Theta\mathrm{p}} & D_{\Theta u} \\ C_\mathrm{p} & D_{\mathrm{p}\Theta} & D_\mathrm{pp} & D_\mathrm{pu} \\ C_y & D_{y\Theta} & D_{y\mathrm{p}} & D_{yu} \end{array}\right] \begin{bmatrix} x \\ \hdashline w_\Theta \\ w_\mathrm{p} \\ u \end{bmatrix}, \\ w_\Theta = \Theta z_\Theta, \end{array} \right. \tag{4}$$

where $w_\Theta\in\mathbb{R}^{n_\Theta}$, $z_\Theta\in\mathbb{R}^{n_\Theta}$. Again, diagonal scheduling blocks $\Theta(t) = \operatorname*{diag}_{i=1}^{n_\theta}\Big(\theta_i I_{r_i^\theta}\Big)$, with $\sum_{i=1}^{n_\theta} r_i^\theta = n_\Theta$ are common, but other forms can be found (Hoffmann and Werner, 2014). We assume also for the affine LPV parameters and their rates of change to be contained in compact sets, s.t. $\theta \in \boldsymbol{\theta}$ and $\dot\theta(t) =: \nu(t) = [\nu_1(t)\ \nu_2(t)\ \dots\ \upsilon_{n_\theta}(t)]$, with $\nu \in \boldsymbol{\nu}$.

For LPV systems affine in their parameters, representations other than the LFT form (4) are also common. From

$$S^\theta(\theta) = \begin{bmatrix} \mathscr{A}(\rho) & \mathscr{B}_\mathrm{p}(\rho) & \mathscr{B}_u(\rho) \\ \mathscr{C}_\mathrm{p}(\rho) & \mathscr{D}_\mathrm{pp}(\rho) & \mathscr{D}_\mathrm{pu}(\rho) \\ \mathscr{C}_y(\rho) & \mathscr{D}_{y\mathrm{p}}(\rho) & \mathscr{D}_{yu}(\rho) \end{bmatrix} = \tag{5}$$
$$+ \begin{bmatrix} A & B_\mathrm{p} & B_u \\ C_\mathrm{p} & D_\mathrm{pp} & D_\mathrm{pu} \\ C_y & D_{y\mathrm{p}} & D_{yu} \end{bmatrix} + \begin{bmatrix} B_\Theta \\ D_{\mathrm{p}\Theta} \\ D_{y\Theta} \end{bmatrix} \Theta [C_\Theta\ D_{\Theta\mathrm{p}}\ D_{\Theta u}]$$
$$= S_0^\theta + \sum_{i=1}^{n_\theta} \theta_i S_i^\theta = S_0^\theta + \sum_{i=1}^{n_\theta} f_i^{\delta\to\theta}\left(f^{\rho\to\delta}\big(\rho(t)\big)\right) S_i^\theta, \tag{6}$$

one may also obtain a representation in barycentric coordinates α_i, where $\theta_{\mathrm{v},i},\ i = 1, \dots, n_\mathrm{v}$ denote scheduling parameter vectors in the vertices of a convex hull that encapsulates the admissible parameter range $\boldsymbol{\theta}$:

$$S^\theta(\theta) = \sum_{i=1}^{n_\mathrm{v}} \alpha_i \left.\begin{bmatrix} \mathscr{A}(\rho) & \mathscr{B}_\mathrm{p}(\rho) & \mathscr{B}_u(\rho) \\ \mathscr{C}_\mathrm{p}(\rho) & \mathscr{D}_\mathrm{pp}(\rho) & \mathscr{D}_\mathrm{pu}(\rho) \\ \mathscr{C}_y(\rho) & \mathscr{D}_{y\mathrm{p}}(\rho) & \mathscr{D}_{yu}(\rho) \end{bmatrix}\right|_{\rho_{\mathrm{v},i}} \quad \begin{array}{l} \sum_{i=1}^{n_\mathrm{v}} \alpha_i = 1, \\ \alpha_i \geqslant 0. \end{array}$$

Analogously, we have $\theta = \sum_{i=1}^{n_\mathrm{v}} \alpha_i \theta_{\mathrm{v},i}$. If the parameters θ are assumed to vary inside a hyperbox, we have $n_\mathrm{v} = 2^{n_\theta}$.

A general formula for determining the barycentric coordinates for a scheduling parameter vector $\theta(t)$ ranging in a simple polytope is given in (Warren et al., 2007).

If the parameters reside in a hyperbox, the derivation of the barycentric coordinates as implemented in the Matlab function `polydec` is performed iteratively. Initialize a vector $c_0 = 1$. Then for $k = 1, \dots, n_\theta$, compute

$$t_k = \frac{\theta_k(t) - \underline{\theta}_k}{\overline{\theta}_k - \underline{\theta}_k}, \quad c_k = [c_{k-1}(1 - t_k),\ c_{k-1}t_k],$$

where $\underline{\theta}_k \leqslant \theta_k \leqslant \overline{\theta}_k$. The vector $c_{n_\theta} = [\alpha_1,\ \alpha_2,\ \cdots,\ \alpha_{n_\theta}]$ then contains the barycentric coordinates.

2.2 State Space Based LPV Output-Feedback Control

In the following, we will review methods to synthesize LPV output-feedback controllers. Consider a gain-scheduled controller of the form

$$\mathcal{K}^\rho : \left\{ \begin{bmatrix} \dot{x}^K \\ \hdashline u \end{bmatrix} = \left[\begin{array}{c:c} \mathscr{A}^K(\rho,\sigma) & \mathscr{B}_y^K(\rho) \\ \hdashline \mathscr{C}_u^K(\rho) & \mathscr{D}_{uy}^K(\rho) \end{array}\right] \begin{bmatrix} x^K \\ \hdashline y \end{bmatrix} \right., \tag{7}$$

where $x^K \in \mathbb{R}^{n_x^K}$. An LFR of the controller may be written as

$$\mathcal{K}^\delta : \left\{ \begin{array}{l} \begin{bmatrix} \dot{x}^K \\ \hdashline z_\Delta^K \\ u \end{bmatrix} = \left[\begin{array}{c:cc} A^K & B_\Delta^K & B_y^K \\ \hdashline C_\Delta^K & D_{\Delta\Delta}^K & D_{\Delta y}^K \\ C_u^K & D_{u\Delta}^K & D_{uy}^K \end{array}\right] \begin{bmatrix} x^K \\ \hdashline w_\Delta^K \\ y \end{bmatrix}, \\ \quad w_\Delta^K = \Delta^K z_\Delta^K, \end{array} \right. \tag{8}$$

where Δ^K contains the scheduling parameters δ_i, as well as the rate of variations υ_i, $i = 1, \ldots, n_\delta$, on the diagonal. As above, we use the notation

$$S^{K^\rho}(\rho) = \begin{bmatrix} \mathscr{A}^K(\rho,\sigma) & \mathscr{B}_y^K(\rho) \\ \mathscr{C}_u^K(\rho) & \mathscr{D}_{uy}^K(\rho) \end{bmatrix},\ S^{K^\delta}(\delta) = \begin{bmatrix} \mathscr{A}^K(\delta,\upsilon) & \mathscr{B}_y^K(\delta) \\ \mathscr{C}_u^K(\delta) & \mathscr{D}_{uy}^K(\delta) \end{bmatrix},$$

such that, e.g., $\mathcal{K}^\delta = \operatorname{diag}\big(s^{-1}I, \Delta^K\big) \star K^\delta = s^{-1}I \star S^{K^\delta}(\delta)$.

The interconnection $\mathcal{T}^\rho = \mathcal{P}^\rho \star \mathcal{K}^\rho$ denotes the closed-loop system

$$\mathcal{T}^\rho : \left\{ \begin{bmatrix} \dot{\xi} \\ \hdashline z_\mathrm{p} \end{bmatrix} = \left[\begin{array}{c:c} \mathcal{A}\big(\rho,\sigma\big) & \mathcal{B}_\mathrm{p}\big(\rho\big) \\ \hdashline \mathcal{C}_\mathrm{p}\big(\rho\big) & \mathcal{D}_\mathrm{pp}\big(\rho\big) \end{array}\right] \begin{bmatrix} \xi \\ \hdashline w_\mathrm{p} \end{bmatrix} \right., \tag{9}$$

where $\xi \in \mathbb{R}^{n_x + n_x^K}$.

Many LPV controller synthesis techniques are based on first formulating a sufficient analysis condition in terms of an infinite-dimensional parameter-dependent matrix inequality, e.g., Wu (1995); Apkarian and Adams (1998); Scherer (2001). In order to arrive at a convex synthesis condition, the matrix inequalities need to be rendered linear in the decision variables. Furthermore, parameter-dependent linear matrix inequalities (PLMIs) require relaxation techniques, in order to be solved by finitely many LMIs. In the following, parameter-dependency is suppressed in notation for the sake of brevity. We restrict the scope to controller synthesis optimal in the sense of the $\mathcal{L}_2$-gain and define $\Gamma = \operatorname{diag}(-\gamma I, 1/\gamma I)$. For brevity, define $\tilde{\Gamma} = \Gamma^{-1}$.

Theorem 1. (LPV System Analysis). The system $\mathcal{T}^\rho$ is stable and achieves an $\mathcal{L}_2$-gain $\gamma > 0$ on the channel $w_\mathrm{p} \to z_\mathrm{p}$, if there exists $\mathcal{X}\big(\rho\big) = \mathcal{X}^\top\big(\rho\big) > 0$, that satisfies

$$\begin{bmatrix} \bullet \\ \hdashline \bullet \end{bmatrix}^\top \left[\begin{array}{cc:c} \partial\mathcal{X} & \mathcal{X} & \\ \mathcal{X} & 0 & \\ \hdashline & & \tilde{\Gamma} \end{array}\right] \begin{bmatrix} I & 0 \\ \mathcal{A} & \mathcal{B}_\mathrm{p} \\ \hdashline 0 & I \\ \mathcal{C}_\mathrm{p} & \mathcal{D}_\mathrm{pp} \end{bmatrix} < 0,\ \forall(\rho,\sigma) \in \boldsymbol{\rho} \times \boldsymbol{\sigma}. \tag{10}$$

The bilinearity of the matrix inequality is usually tackled by a linearizing change of variables or via an elimination of the controller variables (Apkarian and Adams, 1998; Scherer, 2000, 2001), the latter leading to the following existence condition.

Theorem 2. (LPV Controller Existence). There exists a controller $\mathcal{K}^\rho$, that renders the system $\mathcal{T}^\rho$ stable and achieves an $\mathcal{L}_2$-gain $\gamma > 0$ on the channel $w_\mathrm{p} \to z_\mathrm{p}$, if there exist $\mathcal{R}\big(\rho\big) = \mathcal{R}^\top\big(\rho\big) > 0$ and $\mathcal{S}\big(\rho\big) = \mathcal{S}^\top\big(\rho\big) > 0$, that satisfy

$$\mathcal{L}_R = \mathcal{N}_R^\top \begin{bmatrix} \bullet \\ \hdashline \bullet \end{bmatrix}^\top \left[\begin{array}{cc:c} \partial\mathcal{R} & \mathcal{R} & \\ \mathcal{R} & 0 & \\ \hdashline & & \tilde{\Gamma} \end{array}\right] \begin{bmatrix} I & 0 \\ \mathscr{A} & \mathscr{B}_\mathrm{p} \\ \hdashline 0 & I \\ \mathscr{C}_\mathrm{p} & \mathscr{D}_\mathrm{pp} \end{bmatrix} \mathcal{N}_R < 0, \tag{11}$$

$$\mathcal{L}_S = \mathcal{N}_S^\top \begin{bmatrix} \bullet \\ \hdashline \bullet \end{bmatrix}^\top \left[\begin{array}{cc:c} 0 & \mathcal{S} & \\ \mathcal{S} & -\partial\mathcal{S} & \\ \hdashline & & \tilde{\Gamma} \end{array}\right] \begin{bmatrix} -\mathscr{A}^\top & -\mathscr{C}_\mathrm{p}^\top \\ I & 0 \\ \hdashline -\mathscr{B}_\mathrm{p}^\top & -\mathscr{D}_\mathrm{pp}^\top \\ 0 & I \end{bmatrix} \mathcal{N}_S > 0, \tag{12}$$

$$\mathcal{L}_{RS} = \begin{bmatrix} \mathcal{R} & I \\ I & \mathcal{S} \end{bmatrix} > 0, \quad \forall(\rho,\sigma) \in \boldsymbol{\rho} \times \boldsymbol{\sigma}, \tag{13}$$

where $\mathcal{N}_R = \ker\left(\mathscr{C}_y\ \mathscr{D}_{y\mathrm{p}}\right)$, $\mathcal{N}_S = \ker\left(\mathscr{B}_u^\top\ \mathscr{D}_{\mathrm{p}u}^\top\right)$. (14)

Assuming full row rank of $[\mathscr{C}_y\ \mathscr{D}_{y\mathrm{p}}]$ and $\left[\mathscr{B}_u^\top\ \mathscr{D}_{\mathrm{p}u}^\top\right]$ guarantees the existence of a stabilizing output-feedback LPV controller. Note that the matrix inequalities (11) and (12) are sigmonial inequalities in γ. A Schur argument renders these linear in γ. If the variables fulfilling these existence LMIs have been found, explicit formulae can be used to calculate the controller matrices Wu (1995); Wu et al. (1996); Apkarian and Adams (1998); Wu (2001); Wu and Dong (2006). Note that they have been derived under the assumptions $\mathscr{D}_{yu} = 0$ and $\mathscr{D}_\mathrm{pp} = 0$, which can be achieved by proper pre- and postfiltering. Compute the controller matrices via

$$\mathcal{M}\mathcal{N}^\top = I - \mathcal{R}\mathcal{S}$$

$$\mathcal{F} = -\left(\mathscr{D}_{\mathrm{p}u}^\top \mathscr{D}_{\mathrm{p}u}\right)^{-1} \left(\gamma \mathscr{B}_u^\top \mathcal{R}^{-1} + \mathscr{D}_{\mathrm{p}u}^\top \mathscr{C}_\mathrm{p}\right), \tag{15a}$$

$$\mathcal{L} = -\left(\gamma \mathcal{S}^{-1} \mathscr{C}_y^\top + \mathscr{B}_\mathrm{p} \mathscr{D}_{y\mathrm{p}}^\top\right) \left(\mathscr{D}_{y\mathrm{p}} \mathscr{D}_{y\mathrm{p}}^\top\right)^{-1}, \tag{15b}$$

$$\begin{aligned} \mathscr{A}^K = -\mathcal{N}^{-1}\Big(& -\mathcal{S}\dot{\mathcal{R}} - \mathcal{N}\dot{\mathcal{M}}^\top + \\ & + \mathscr{A}^\top + \mathcal{S}\big[\mathscr{A} + \mathscr{B}_u \mathcal{F} + \mathcal{L}\mathscr{C}_y\big]\mathcal{R} + \\ & + \gamma^{-1}\mathcal{S}\big[\mathscr{B}_\mathrm{p} + \mathcal{L}\mathscr{D}_{y\mathrm{p}}\big]\mathscr{B}_\mathrm{p}^\top + \\ & + \gamma^{-1}\mathscr{C}_\mathrm{p}^\top\big[\mathscr{C}_\mathrm{p} + \mathscr{D}_{\mathrm{p}u}\mathcal{F}\big]\mathcal{R}\Big)\mathcal{M}^{-\top}, \end{aligned} \tag{16a}$$

$$\mathscr{B}^K = \mathcal{N}^{-1}\mathcal{S}\mathcal{L}, \quad \mathscr{C}^K = \mathcal{F}\mathcal{R}\mathcal{M}^{-\top}, \quad \mathscr{D}^K = 0. \tag{16b}$$

To reduce the online computational load by avoiding an online singular value decomposition, the trivial factorizations $\mathcal{M} = I - \mathcal{R}\mathcal{S}$, $\mathcal{N} = I$ or $\mathcal{M} = I$, $\mathcal{N} = I - \mathcal{S}\mathcal{R}$ can be chosen (Apkarian and Adams, 1998). Furthermore, if in the first case $\mathcal{R} := \mathcal{R}(\rho)$, $\mathcal{S} := \mathcal{S}_0$ (or vice versa for the second case) is chosen, the controller will not depend on the parameters' rate of variation (Apkarian and Adams, 1998). For example, if $\mathcal{N} = I, \mathcal{S} := \mathcal{S}_0$, we have

$$\mathcal{M} = I - \mathcal{R}\mathcal{S}_0$$

$$\mathcal{F} = -\left(\mathscr{D}_{\mathrm{p}u}^\top \mathscr{D}_{\mathrm{p}u}\right)^{-1} \left(\gamma \mathscr{B}_u^\top \mathcal{R}^{-1} + \mathscr{D}_{\mathrm{p}u}^\top \mathscr{C}_\mathrm{p}\right), \tag{17a}$$

$$\mathcal{L} = -\left(\gamma \mathcal{S}_0^{-1} \mathscr{C}_y^\top + \mathscr{B}_\mathrm{p} \mathscr{D}_{y\mathrm{p}}^\top\right) \left(\mathscr{D}_{y\mathrm{p}} \mathscr{D}_{y\mathrm{p}}^\top\right)^{-1}, \tag{17b}$$

$$\begin{aligned} \mathscr{A}^K = -\Big(& \mathscr{A}^\top + \mathcal{S}_0\big[\mathscr{A} + \mathscr{B}_u \mathcal{F} + \mathcal{L}\mathscr{C}_y\big]\mathcal{R} + \gamma^{-1}\mathcal{S}_0\big[\mathscr{B}_\mathrm{p} + \mathcal{L}\mathscr{D}_{y\mathrm{p}}\big]\mathscr{B}_\mathrm{p}^\top \\ & + \gamma^{-1}\mathscr{C}_\mathrm{p}^\top\big[\mathscr{C}_\mathrm{p} + \mathscr{D}_{\mathrm{p}u}\mathcal{F}\big]\mathcal{R}\Big)\mathcal{M}^{-\top}, \end{aligned} \tag{18a}$$

$$\mathscr{B}^K = \mathcal{S}_0\mathcal{L}, \quad \mathscr{C}^K = \mathcal{F}\mathcal{R}\mathcal{M}^{-\top}, \quad \mathscr{D}^K = 0. \tag{18b}$$

For constant Lyapunov functions, there exist approaches for both affine and rational parameter-dependencies that allow to search for the controller variables via LMIs. In this case, further optimizations—e.g. w.r.t. the spectral radius of the controller's state matrix—can be performed.

In the following, we will distinguish the three most common approaches for turning the infinite set of LMIs into a finite set of LMIs that can be solved via semi-definite programming:

a) Polytopic LPV synthesis,
b) LFT-based LPV synthesis with multipliers,
c) Gridding-based LPV synthesis.

Polytopic LPV Synthesis: If the system (1) admits an affine/polytopic LPV representation with $\mathscr{C}_y$, $\mathscr{D}_{y\mathrm{p}}$, $\mathscr{B}_u$, $\mathscr{D}_{\mathrm{p}u}$ being constant matrices and the parameter-dependence of $\mathcal{R}(\rho)$ and $\mathcal{S}(\rho)$ is dropped at the expense of conservatism, the existence conditions (11)–(13) can be solved in the vertices $\theta_{\mathrm{v},i},\ i=1,\dots,n_{\mathrm{v}},\ n_{\mathrm{v}}=2^{n_\theta}$, where the convex hull of this set of vertices $\operatorname{conv}(\boldsymbol{\theta}_{\mathrm{v}})$ includes the parameter set $\boldsymbol{\theta}$ (Apkarian et al., 1995). If $\mathcal{R}(\rho)$ and $\mathcal{S}(\rho)$ are assumed to depend affinely on the parameters θ, a multi-convexity approach (Gahinet et al., 1996) can be used to introduce additional constraints

$$\frac{\partial^2}{\partial\theta_i^2}\mathcal{L}_R \geqslant 0, \quad \frac{\partial^2}{\partial\theta_i^2}\mathcal{L}_S \leqslant 0, \quad i=1,\dots,n_\theta \tag{19}$$

which allow to still solve the inequalities on a finite set of vertices Gahinet et al. (1996). If the performance channel, or more specifically matrices $\mathscr{B}_{\mathrm{p}}$ and $\mathscr{C}_{\mathrm{p}}$ are parameter-independent, the multi-convexity constraints can be reduced to

$$\frac{\partial^2}{\partial\theta_i^2}\left(\dot{\mathcal{R}}+\mathcal{R}\mathscr{A}+\mathscr{A}^\top\mathcal{R}\right)\geqslant 0,\ -\frac{\partial^2}{\partial\theta_i^2}\left(\dot{\mathcal{S}}+\mathcal{S}\mathscr{A}^\top+\mathscr{A}\mathcal{S}\right)\leqslant 0,$$

for all $i=1,\dots,n_\theta$. Matrix inequalities (11)–(12) will be affine in ν and assuming that $n_\nu \leqslant n_\theta$ of the parameters have a non-zero rate of change and/or are considered in the parameter-dependent Lyapunov function (PDLF), the number of vertices increases to $n_{\mathrm{v}}=2^{n_\theta+n_\nu}$.

The controller is computed online as a weighted sum of the vertex controllers, which may be obtained explicitly. So-called overbounding may occur, i.e., guarantees are provided for portions of the scheduling signal range, that are not physically admissible. In many applications, the parameter polytope can be optimized to either cover the parameter set more closely and/or use less vertices than incurred by naively considering a hyperbox.

Multiplier-Based LFT LPV Synthesis: The analysis condition (10) can be turned into a finite-dimensional inequality on the system matrices by application of the Full-Block $\mathcal{S}$-Procedure (FBSP) (Scherer, 2001). An additional inequality on the multiplier quadratic in the LFT scheduling block is introduced, which can again be evaluated in the vertices of the parameter range if constrained by multi-convexity conditions. Existence conditions linear in all variables are derived via controller elimination and controller construction is either performed by closed-form formulae (15)–(16), or by first reconstructing the extended closed-loop multiplier and Lyapunov variable (Scherer, 2001) and then solving an LMI problem. Both approaches will lead to potentially complex, yet less conservative, controller scheduling policies, while by the introduction of some more conservative structural constraints on the multipliers—such as D/G-scalings (Scorletti and Ghaoui, 1998)—, the conditions are rendered trivially fulfilled and the controller is allowed to receive a copy of the plant's scheduling block. This reduces implementation complexity but increases conservatism.

PDLF based synthesis (Wu and Dong, 2006) requires the application of the FBSP on (11)–(13). For this purpose, LFRs of the null-spaces $\mathcal{N}_R$ and $\mathcal{N}_S$ are required, if they are parameter-dependent, or pre- and postfiltering can be applied to render them parameter-independent. To the best of the authors' knowledge, the controller can then only be constructed by closed-form formulae (15)–(16) or (17)–(18), respectively.

For illustration consider quadratic Lyapunov parameterizations $\mathcal{R}=\mathcal{T}_R^\top R\mathcal{T}_R$, $\mathcal{S}=\mathcal{T}_S^\top S\mathcal{T}_S$. LFRs of the outer factors

$$\Delta_R\star\left[\begin{array}{c:c} G_R^{11} & G_R^{12} \\ \hdashline G_R^{21} & G_R^{22}\end{array}\right]=\left[\begin{array}{cc:cc} \mathcal{T}_R & 0 & & \\ \dot{\mathcal{T}}_R & \mathcal{T}_R & & \\ \hdashline & & I & 0 \\ & & 0 & I\end{array}\right]\left[\begin{array}{cc} I & 0 \\ \mathscr{A} & \mathscr{B}_{\mathrm{p}} \\ \hdashline 0 & I \\ \mathscr{C}_{\mathrm{p}} & \mathscr{D}_{\mathrm{pp}}\end{array}\right]\mathcal{N}_R, \tag{20}$$

$$\Delta_S\star\left[\begin{array}{c:c} G_S^{11} & G_S^{12} \\ \hdashline G_S^{21} & G_S^{22}\end{array}\right]=\left[\begin{array}{cc:cc} \mathcal{T}_S & -\dot{\mathcal{T}}_S & & \\ 0 & \mathcal{T}_S & & \\ \hdashline & & I & 0 \\ & & 0 & I\end{array}\right]\left[\begin{array}{cc} -\mathscr{A}^\top & -\mathscr{C}_{\mathrm{p}}^\top \\ I & 0 \\ \hdashline -\mathscr{B}_{\mathrm{p}}^\top & -\mathscr{D}_{\mathrm{pp}}^\top \\ 0 & I\end{array}\right]\mathcal{N}_S, \tag{21}$$

$$\Delta_{RS}\star\left[\begin{array}{c:c} G_{RS}^{11} & G_{RS}^{12} \\ \hdashline G_{RS}^{21} & G_{RS}^{22}\end{array}\right]=\begin{bmatrix} \mathcal{T}_R & 0 \\ 0 & \mathcal{T}_S \\ I & 0 \\ 0 & I\end{bmatrix} \tag{22}$$

allow to write (11)–(13) as

$$\mathcal{L}_R=\left[\begin{array}{c}\bullet \\ \hdashline \bullet\end{array}\right]^\top\left[\begin{array}{c:cc:c} M & & & \\ \hdashline & 0 & R & \\ & R & 0 & \\ \hdashline & & & \Gamma\end{array}\right]\left[\begin{array}{cc} G_R^{11} & G_R^{12} \\ I & 0 \\ \hdashline G_R^{21} & G_R^{22}\end{array}\right]<0, \tag{23}$$

$$\mathcal{L}_S=\left[\begin{array}{c}\bullet \\ \hdashline \bullet\end{array}\right]^\top\left[\begin{array}{c:cc:c} N & & & \\ \hdashline & 0 & S & \\ & S & 0 & \\ \hdashline & & & \tilde{\Gamma}\end{array}\right]\left[\begin{array}{cc} G_S^{11} & G_S^{12} \\ I & 0 \\ \hdashline G_S^{21} & G_S^{22}\end{array}\right]>0, \tag{24}$$

$$\mathcal{L}_{RS}=\left[\begin{array}{c}\bullet \\ \hdashline \bullet\end{array}\right]^\top\left[\begin{array}{c:cc:cc} P & & & & \\ \hdashline & R & 0 & & \\ & 0 & S & & \\ \hdashline & & & 0 & I \\ & & & I & 0\end{array}\right]\left[\begin{array}{cc} G_{RS}^{11} & G_{RS}^{12} \\ I & 0 \\ \hdashline G_{RS}^{21} & G_{RS}^{22}\end{array}\right]>0 \tag{25}$$

and $\forall(\rho,\sigma)\in\boldsymbol{\rho}\times\boldsymbol{\sigma}$

$$\mathcal{L}_M=\left[\begin{array}{c}\bullet \\ \hdashline \bullet\end{array}\right]^\top M\left[\begin{array}{c} I \\ \hdashline \Delta_R\end{array}\right]>0,\ \mathcal{L}_N=\left[\begin{array}{c}\bullet \\ \hdashline \bullet\end{array}\right]^\top N\left[\begin{array}{c} I \\ \hdashline \Delta_S\end{array}\right]<0, \tag{26}$$

$$\mathcal{L}_P=\left[\begin{array}{c}\bullet \\ \hdashline \bullet\end{array}\right]^\top P\left[\begin{array}{c} I \\ \hdashline \Delta_{RS}\end{array}\right]<0. \tag{27}$$

Note that in the case of parameter-independent Lyapunov functions, $\mathcal{T}_R$ and $\mathcal{T}_S$ degenerate to identities, rendering the multiplier P obsolete. The results from (Scherer, 2000, 2001) are recovered if in contrast to the derivation of (11)–(12) the multiplier is introduced before the controller is eliminated. To arrive at the LMIs for the existence conditions for parameter-independent Lyapunov function based synthesis, set the LFRs of the outer factors (20)–(21) to

$$\Delta\star\left[\begin{array}{c:c} G_R^{11} & G_R^{12} \\ \hdashline G_R^{21} & G_R^{22}\end{array}\right]=\left[\begin{array}{cc} I & 0 \\ \mathscr{A} & \mathscr{B}_{\mathrm{p}} \\ \hdashline 0 & I \\ \mathscr{C}_{\mathrm{p}} & \mathscr{D}_{\mathrm{pp}}\end{array}\right],\ \Delta\star\left[\begin{array}{c:c} G_S^{11} & G_S^{12} \\ \hdashline G_S^{21} & G_S^{22}\end{array}\right]=\left[\begin{array}{cc} -\mathscr{A}^\top & -\mathscr{C}_{\mathrm{p}}^\top \\ I & 0 \\ \hdashline -\mathscr{B}_{\mathrm{p}}^\top & -\mathscr{D}_{\mathrm{pp}}^\top \\ 0 & I\end{array}\right],$$

and use

$$\tilde{\mathcal{L}}_R = \tilde{\mathcal{N}}_R^\top \mathcal{L}_R \tilde{\mathcal{N}}_R < 0, \quad \tilde{\mathcal{L}}_S = \tilde{\mathcal{N}}_S^\top \mathcal{L}_S \tilde{\mathcal{N}}_S > 0, \tag{28}$$

$$\tilde{\mathcal{L}}_{RS} = \begin{bmatrix} R & 0 \\ 0 & S \end{bmatrix} > 0, \tag{29}$$

$$\tilde{\mathcal{L}}_M = \left[\begin{array}{c} \bullet \\ \hline \bullet \end{array}\right]^\top M \left[\begin{array}{c} I \\ \hline \Delta \end{array}\right] > 0, \; \tilde{\mathcal{L}}_N = \left[\begin{array}{c} \bullet \\ \hline \bullet \end{array}\right]^\top N \left[\begin{array}{c} I \\ \hline \Delta \end{array}\right] < 0, \tag{30}$$

with $\tilde{\mathcal{N}}_R = \ker\left(C_y \; D_{y\Delta} \; D_{y\mathrm{p}}\right)$, $\tilde{\mathcal{N}}_S = \ker\left(B_u^\top \; D_{\Delta u}^\top \; D_{\mathrm{p}u}^\top\right)$. Controller construction can then be handled by reconstructing an extended multiplier M_{cl} and a closed-loop Lyapunov matrix $\mathcal{X}$, which—inserted in the analysis matrix inequality—allow to solve for the controller LTI matrices K^δ explicitly instead of using (15) and (16).

Gridding-Based LPV Synthesis: In gridding-based LPV synthesis approaches Wu (1995); Apkarian and Adams (1998), the inequalities (11)–(13) are solved on a set of points covering the admissible parameter signals. This technique is applicable to LPV plants with general parameter-dependency, requiring neither polytopic nor LFRs. Accordingly, the Lyapunov variable can be parameterized by general parameter-dependency. Since this approach does not provide any rigorous guarantees for closed-loop stability and performance, the analysis inequality (10) is usually checked *a posteriori* on a much denser grid. Note that the gridding approach can easily accommodate non-convex signal ranges.

The implementation scheme may consist in an interpolation or a switching between local pre-computed controllers, requiring a high amount of memory. Closed-form controller formulae can be applied instead, shifting the complexity to online computing. Especially in the light of microprocessors limited in precision and/or computing power, the gridded look-up table based implementation is attractive, but applies to the other synthesis methods as well. I.e. also controller matrices S^{K^θ} or S^{K^δ} derived from polytopic and LFT-based controller representations can be gridded in terms of the scheduling signals ρ and implemented in look-up tables.

If observer-based state-feedback synthesis is performed, parameter-dependent state-feedback matrices of both observer and controller can be computed online each by a single matrix inversion without the need to store multiple controllers in memory (Saupe and Pfifer, 2011, 2012). Via a loop-shaping approach frequency-dependent characteristics as in the output-feedback case can be achieved, while observer and state-feedback gains can be synthesized sequentially via projection or a linearizing change of variables each at the cost of approximately the analysis problem (10) instead of the two projected LMIs. This can—in some sense—alleviate the relatively high complexity of the gridding approach.

3. IMPLEMENTATION COMPLEXITY

The choice of the modeling framework and associated synthesis techniques affect both the complexities encountered during implementation and synthesis. In this section the involved complexity is analyzed, in order to generate numerical indicators for benefits and drawbacks of the individual approaches *a priori*.

Table 1. Complexity of matrix operations.

Operation		Sizes		$\mathsf{a}(A)$
Multiplication	$A = BC$	$B \in \mathbb{R}^{n\times m}$,	$C \in \mathbb{R}^{m\times p}$	$n(2m-1)p$
Scaling	$A = \operatorname{diag}_{i=1}^{n}\left(b_i\right) C$	$b_i \in \mathbb{R}$,	$C \in \mathbb{R}^{n\times m}$	nm
Addition	$A = B + C$	$B \in \mathbb{R}^{n\times m}$,	$C \in \mathbb{R}^{n\times m}$	nm
Inversion*	$A = B^{-1}$	$B \in \mathbb{R}^{n\times n}$,		$\frac{2}{3}n^3$

* *Gauss elimination provides an upper bound for the cost.*

Table 2. Memory requirements of matrix types.

Matrix structure		Sizes	$\mathsf{m}(A)$
Full	A	$A \in \mathbb{R}^{n\times m}$,	nm
Symmetric	$A = A^\top$	$A \in \mathbb{R}^{n\times n}$,	$\sum_{k=1}^{n+1} k = n(n+1)/2$
Skew-sym.	$A = -A^\top$	$A \in \mathbb{R}^{n\times n}$,	$\sum_{k=1}^{n-1} k = n(n-1)/2$

For the sake of simplicity, we consider the case, when signal and parameter values are taken to range inside a hyperbox, which in most cases will mark an upper bound on the complexity. Furthermore, we will—as in the preceding discussions—for the most part consider the synthesis and implementation of state space output-feedback LPV controllers (7) of full order ($n_x^K = n_x$).

Tab. 1 shows the number of arithmetic computations, denoted by $\mathsf{a}(\cdot)$, for elementary matrix operations and the number of scalar variables to be stored $\mathsf{m}(\cdot)$, which is used as a measure for the memory requirements.

3.1 General Complexity

Updating the states and calculating the outputs is assumed to consume the same amount of arithmetic operations for each output-feedback-based method, which amounts to

$$\mathsf{a}\left(\left[\dot{x}^{K\top}, u^\top\right]^\top\right) \leqslant \left(n_x + n_u\right)\left(2(n_x + n_y) - 1\right) \tag{31}$$

arithmetic operations once the state space matrices of the controller at a particular time instant are available. In contrast, a state-feedback controller requires

$$\mathsf{a}(u) \leqslant n_u(2n_x - 1). \tag{32}$$

once the state-feedback gain matrix $\mathcal{F}$, as from (15a) with $\mathscr{D}_{\mathrm{p}u} = I$, has been computed.

If the projection approach considered in Theorem 2 is applied, formulae (15)–(18) have to be used for the implementation of PDLF-based controllers, independent of the LPV framework (polytopic, LFT-based, gridding) considered. The reason for this resides in the fact that the construction of the closed-loop Lyapunov matrix $\mathcal{X}$—even if carried out symbolically—results in a rational parameter-dependence and a convex search for the controller variables is only possible by again solving a gridded LMI problem based on (10). For online implementation via (15)–(18), the plant matrices (1) should therefore be available at any given time instant via their polytopic, affine, look-up table or LFR, respectively. We consider the case where the controller depends on rates of change practically undesirable and therefore restrict the analysis to Eqs. (17)–(18). Computing (17) and (18) then amounts to

$$\mathsf{a}\left(S^{K^\rho}\right) \leqslant \mathsf{a}\left(S^\rho\right) + \mathsf{a}(\mathcal{F}) + \mathsf{a}(\mathcal{L}) + \mathsf{a}(\mathcal{M}) + \mathsf{a}\left(\mathscr{A}^K\right) + \mathsf{a}\left(\mathscr{B}^K\right) + \mathsf{a}\left(\mathscr{C}^K\right).$$

For implementation, the generalized plant matrices in (1) are required and we assume that their evaluation can be performed efficiently enough, such that the cost in arithmetic operations is negligible. The actual memory and evaluation costs then depend on the parameterization of $\mathcal{R}$. Throughout we will consider the intuitive heuristic that a Lyapunov function basis is chosen that aims at reflecting the parameter-dependency of the plant. Therefore $\mathsf{a}(\mathcal{R})$ will depend on the framework, the plant is modeled in. Once $\mathcal{R}$ is constructed online, however, its inversion requires

$$\mathsf{a}\big(\mathcal{M}^{-\top}\big) \approx \mathsf{a}\big(\mathcal{R}^{-1}\big) \leqslant {}^2\!/_3 n_x^3 \tag{33}$$

operations. Note that this complexity is absorbed in $\mathsf{a}(\mathcal{F})$. We further have $\mathsf{a}(\mathcal{M}) \leqslant n_x + 2n_x^3$. In the following enumeration of complexities care of an economic sequence of operations has been taken, such that, e.g., the inversion of $\mathcal{M}$ is considered in $\mathsf{a}\big(\mathscr{A}^K\big)$, but not in $\mathsf{a}\big(\mathscr{C}^K\big)$, as an efficient implementation will store the result of the inversion for multiple uses.

$$\begin{aligned} \mathsf{a}(\mathcal{F}) &\leqslant 2n_u^2 n_z + {}^2\!/_3(n_u^3 + n_x^3) + n_u(2n_x - 1)n_x \\ &\quad + 2n_u n_x + n_u(2n_z - 1)n_x + n_u(2n_u - 1)n_x && (34) \\ \mathsf{a}(\mathcal{L}) &\leqslant 2n_x^2 n_y + n_x(2n_w - 1)n_y + n_x n_y + 2n_y^2 n_w + {}^2\!/_3 n_y^3 && (35) \\ \mathsf{a}\big(\mathscr{A}^K\big) &\leqslant 7n_x^2 + n_x n_w + n_x n_z + n_x(2n_y - 1)(n_x + n_w) \\ &\quad + (n_x + n_z)(2n_u - 1)n_x + n_x(2n_x - 1)(5n_x + n_z) \\ &\quad + n_x^2(2n_w - 1) + {}^2\!/_3 n_x^3 && (36) \\ \mathsf{a}\big(\mathscr{B}^K\big) &\leqslant n_x(2n_x - 1)n_u && (37) \\ \mathsf{a}\big(\mathscr{C}^K\big) &\leqslant n_x(2n_x - 1)(n_y + n_x) && (38) \end{aligned}$$

In addition to evaluating the Lyapunov variable online, which costs $\mathsf{a}(\mathcal{R})$, this results in a total number of arithmetic operations to evaluate the controller's state space matrices from (17)–(18)

$$\begin{aligned} \mathsf{a}(S^{K^\rho}) &\leqslant {}^{46}\!/_3 n_x^3 + (6m_{uy} + 2m_{wz} - 2)n_x^2 \\ &\quad + (n_w + m_{uz}(2n_u - 1) + 2n_u n_z - m_{uyw} + 4n_y n_w + 1)n_x \\ &\quad + 2n_z n_u^2 + 2n_w n_y^2 + {}^2\!/_3 n_u^3 + {}^2\!/_3 n_y^3. \end{aligned} \tag{39}$$

In many cases the performance channel related matrices will be parameter-independent. If in addition, $\mathscr{C}_y$ is also parameter-independent, $\mathcal{L} \in \mathbb{R}^{n_x \times n_y}$ can be computed offline and the number of required computational steps reduces. The same applies to the alternative practical case, where $\mathcal{R}$ instead of $\mathcal{S}$ is chosen constant and performance channel and input matrices are parameter-independent.

The memory requirements to store the plant matrices of general dependency on the scheduling signals are approximated by ($\mathscr{D}_{\mathrm{pp}} = 0, \mathscr{D}_{yu} = 0$)

$$\mathsf{m}(S^\rho) \approx (n_x + n_z + n_y)(n_x + n_w + n_u) - n_z n_w - n_y n_u. \tag{40}$$

We further have $\mathsf{m}(\mathcal{S}_0) = \mathsf{m}\big(\mathcal{S}_0^{-1}\big) = n_x(n_x + 1)/2$. Throughout the paper, we will neglect the memory requirement $\mathsf{m}(\gamma)$. Thus the total required scalar variables to be stored amount to

$$\begin{aligned} \mathsf{m}(S^{K^\rho}) &\approx \mathsf{m}(S^\rho) + 2\mathsf{m}(\mathcal{S}_0) + \mathsf{m}(\mathcal{R}) \\ &\approx n_x(n_x + 1) + (n_x + n_z + n_y)(n_x + n_w + n_u) \\ &\quad - n_z n_w - n_y n_u + \mathsf{m}(\mathcal{R}) \end{aligned} \tag{41}$$

A particularly efficient implementation can be performed for state-feedback LPV controllers. It is possible to evaluate the state-feedback gain $\mathcal{F} = -\big(\gamma \mathscr{B}_u^\top \mathcal{R}^{-1} + \mathscr{C}_{\mathrm{p}}\big)$ by

$$\mathsf{a}(\mathcal{F}) \leqslant \mathsf{a}(\mathcal{R}) + {}^2\!/_3 n_x^3 + n_u(2n_x - 1)n_x + n_u n_x.$$

Here, only two plant matrices and a single inversion need to be calculated, which makes up the main computational load (Saupe and Pfifer, 2012).

3.2 Polytopic LPV Controllers

Polytopic LPV controllers synthesized based on parameter-independent Lyapunov functions can be implemented by the interpolation of the state space matrices of the LTI vertex controllers. Therefore, we have:

$$S^{K^\theta}(\theta) = \begin{bmatrix} \mathscr{A}^K(\rho) & \mathscr{B}_y^K(\rho) \\ \mathscr{C}_u^K(\rho) & \mathscr{D}_{uy}^K(\rho) \end{bmatrix} = \sum_{i=1}^{n_{\mathrm{v}}} \alpha_i S^{K^\theta}(\theta_{\mathrm{v},i}). \tag{42}$$

The associated number of arithmetic operations is

$$\mathsf{a}(S^{K^\theta}) \leqslant (2^{n_\theta + 1} - 1)(n_x + n_u)(n_x + n_y), \tag{43}$$

which results from scaling each of the $n_{\mathrm{v}} = 2^{n_\theta}$ vertex controllers by the respective α_i and then calculating the controller as a weighted sum by $2^{n_\theta} - 1$ matrix additions. In addition, the algorithm given in (Warren et al., 2007) to compute the barycentric coordinates α from the affine parameters θ requires approximately

$$\mathsf{a}(\alpha) \leqslant n_{\mathrm{v}} \mathsf{a}(\alpha_i) = n_{\mathrm{v}} \left(O(n_\theta^3) + n_\theta^2 + n_\theta - 1\right).$$

When the parameters range in a hyperbox, the computation of the involved determinants is always one and we have

$$\mathsf{a}(\alpha) \leqslant 2^{n_\theta} \left(n_\theta^2 + n_\theta - 1\right).$$

In contrast the Matlab implementation of the command `polydec` requires

$$\begin{aligned} \mathsf{a}(\alpha) &\leqslant \sum_{k=1}^{n_\theta} \left(\mathsf{a}(t_k) + \mathsf{a}(c_k)\right) \\ &= 3n_\theta + 2\frac{1 - 2^{n_\theta + 1}}{1 - 2} = 2^{n_\theta + 2} + 3n_\theta - 2, \end{aligned}$$

with $\mathsf{a}(t_k) = 3$, $\mathsf{a}(c_k) = 2^{k+1}$ and by using the geometric series $\sum_{k=0}^n a^k = \frac{1 - a^{n+1}}{1 - a}$. Note that this is only valid for parameters ranging in a hyperbox, but also that it is always less costly than the algorithm proposed by (Warren et al., 2007).

Storing the controller matrices in the $n_{\mathrm{v}} = 2^{n_\theta}$ vertices requires

$$\mathsf{m}(S^{K^\theta}) \approx 2^{n_\theta}(n_x + n_u)(n_x + n_y). \tag{44}$$

If either an offline preprocessing can be applied, which converts the convex coordinates back into the affine LPV parameter coordinates, or the synthesis for an affine LPV plant is carried out using multiplier-based LFT methods with additional constraints (Dettori and Scherer, 2001; Hoffmann et al., 2013a, 2014a, 2013b), the exponential growth can be reduced to linear growth and convex coordinates need no longer be computed online:

$$\mathsf{a}(S^{K^\theta}) \leqslant 2n_\theta(n_x + n_u)(n_x + n_y), \tag{45}$$

$$\mathsf{m}(S^{K^\theta}) \approx (n_\theta + 1)(n_x + n_u)(n_x + n_y). \tag{46}$$

In the subsequent summary, we will assume that the implementation of affine controllers is carried out in this more efficient way.

Using PDLFs in conjunction with the multi-convexity approach results in a controller which is no longer affine in the parameters θ, but rational. Therefore, the explicit formulae (15)–(16) or (17)–(18), respectively, have to be used, the complexity of which has already been discussed. Assuming the Lyapunov matrix has been parameterized as

$$\mathcal{R} = \mathcal{R}_0 + \sum_{i=r}^{s} \theta_i \mathcal{R}_i, \quad n_\theta^{\mathcal{R}} = s - r. \tag{47}$$

its online construction requires

$$\mathsf{a}(\mathcal{R}) \leqslant 2n_\theta^{\mathcal{R}} n_x^2, \quad \mathsf{m}(\mathcal{R}) \approx {}^1\!/_2(n_\theta^{\mathcal{R}} + 1)n_x(n_x + 1)$$

operations and stored scalars, respectively.

3.3 LFT-Based LPV Controllers

The computation of LFT-based controllers is of polynomial order:

$$\begin{aligned}\mathsf{a}(S^{K^\delta}) \leqslant & 2n_\Delta(n_x + n_u)(n_x + n_y) + \mathsf{a}(\Psi^K) + \ldots \\ & \ldots + n_\Delta(n_x + n_u) + \mathsf{a}(\Delta^K) \in O(n_\Delta^3), \\ & \text{with } \mathsf{a}(\Psi^K) \leqslant n_\Delta \left({}^2\!/_3 n_\Delta^2 + 2n_\Delta + 1\right),\end{aligned} \tag{48}$$

where $\Psi^K(t) = \Delta^K(I - D_{\Delta\Delta}^K \Delta^K)^{-1}$, $\Delta^K \in \mathbb{R}^{n_\Delta \times n_\Delta}$. Memory requirements amount to

$$\mathsf{m}(S^{K^\delta}) \leqslant (n_x + n_\Delta + n_u)(n_x + n_\Delta + n_y) + \mathsf{m}(\Delta^K). \tag{49}$$

The terms $\mathsf{a}(\Delta^K)$ and $\mathsf{m}(\Delta^K)$ are due to the scheduling block $\Delta^K(\Delta)$. For LFT-based synthesis with PiDLFs, Scherer (2000) provides details on the conditions and explicit construction of the scheduling function $\Delta^K(\Delta)$ as an LFT in $\begin{bmatrix} 0 & \Delta^\top \\ \Delta & 0 \end{bmatrix}$, which reads as

$$\Delta^K(\Delta) = -W_{22} + [W_{21} \ V_{21}] \begin{bmatrix} U_{11} & \bullet \\ W_{11} + \Delta & V_{11} \end{bmatrix}^{-1} \begin{bmatrix} U_{12} \\ W_{12} \end{bmatrix} \tag{50}$$

with matrices V_{ij}, W_{ij}, U_{ij}, $i, j = 1, 2$ being elements of $\mathbb{R}^{n_\Delta \times n_\Delta}$ constructed from a certain block partitioned symmetric extended multiplier M_{cl} via $V = -M_{\text{cl},22}^{-1}$, $W = -VM_{\text{cl},12}$, $U = M_{\text{cl},11} + M_{\text{cl},12}^\top W$. It can be observed that both V and U are symmetric.

The computation of the controller's scheduling function therefore requires

$$\mathsf{a}(\Delta^K) \leqslant {}^2\!/_3(2n_\Delta)^3 + 2n_\Delta^2 + 2n_\Delta^2(4n_\Delta - 1) + n_\Delta^2(4n_\Delta - 1) \tag{51}$$

with summands from left to right denoting the inversion, additions performed before the inverse, the right hand and left hand side multiplications, respectively. Due to the symmetry, the inversion can possibly be performed more efficiently, which has been neglected here. Taking into account the symmetry of U and V, however, the memory requirements amount to

$$\mathsf{m}(\Delta^K) = 7n_\Delta(n_\Delta + 1). \tag{52}$$

Via additional constraints on multipliers and at the price of increased conservatism (Dettori and Scherer, 2001), the choice $\Delta^K = \Delta$ can be made admissible. We neglect the cost of evaluating Δ and therefore consider both memory requirements and arithmetic operations negligible.

In the case of PDLFs and LFT-based synthesis methods the Lyapunov variable can be parameterized in a multitude of ways and an affine parameterization is most likely not the best choice. Therefore, we consider an Ansatz which—in a sense—mimics the rational parameter-dependence of the plant (Iwasaki and Shibata, 2001):

$$\mathcal{R} = \mathcal{T}_R^\top R \mathcal{T}_R = \begin{bmatrix} \bullet \\ \bullet \end{bmatrix}^\top \begin{bmatrix} R_0 & R_\Delta \\ R_\Delta^\top & 0 \end{bmatrix} \begin{bmatrix} I \\ \Delta(I - D_{\Delta\Delta}\Delta)^{-1}C_\Delta \end{bmatrix}, \tag{53}$$

with $\mathsf{s}(R) = n_x + n_\Delta$. For controller implementation, evaluating (17)–(18) is required and the complexity again follows (39) with

$$\begin{aligned}\mathsf{a}(\mathcal{R}) \leqslant & n_\Delta(2n_\Delta - 1)n_x + (n_x + n_\Delta)(2(n_x + n_\Delta) - 1)n_x \\ & + n_x(2(n_x + n_\Delta) - 1)n_x + \mathsf{a}(\Psi) \\ & \text{with } \mathsf{a}(\Psi) \leqslant n_\Delta \left({}^2\!/_3 n_\Delta^2 + 2n_\Delta + 1\right).\end{aligned}$$

Storing the Lyapunov variables requires

$$\mathsf{m}(\mathcal{R}) = {}^1\!/_2 n_x(n_x + 1) + n_\Delta^2. \tag{54}$$

However, the evaluation of S^{K^δ} can possibly be performed more efficiently via first evaluating (17)–(18) offline symbolically (Gonzalez et al., 2013). It is put into LFT form by tools available in the Control System Toolbox of Matlab or the LFR-toolbox available from the German Aerospace Center (DLR) (Hecker et al., 2004). In this case, it is difficult to predict the size of the scheduling block Δ^K, which will no longer match the size of the block Δ. Then again both (48) and (49) apply, but with n_Δ replaced by n_Δ^K.

3.4 Gridding-Based LPV Controllers

Gridding-based LPV controllers can be implemented online by the formulae (15)–(16) or (17)–(18). For PiDLFs the computation simplifies drastically, as, e.g., the factorization problem and many multiplications can be performed offline.

Apart from using the explicit formulae, it is also possible to store precomputed controllers on some parameter grid, which does not necessarily need to match the one used to solve the synthesis LMIs. If an evenly spaced grid of n_{g} points per parameter dimension is assumed, the required memory amounts to

$$\mathsf{m}(S^{K^\rho}) = n_{\text{g}}^{n_\rho}(n_x + n_u)(n_x + n_y).$$

It is clear, that an interpolation for intermediate grid points requires a number of arithmetic operations in the same order as in the polytopic case:

$$\mathsf{a}(S^{K^\rho}) \leqslant (2^{n_\rho + 1} - 1)(n_x + n_u)(n_x + n_y).$$

This approach, which resembles the complexity of classical gain-scheduling techniques, can therefore quickly become intractable and control engineers might opt for switching between controller parameters or the above mentioned closed-form formulae instead.

4. SYNTHESIS COMPLEXITY

In the following both the total size of the LMI resulting from the diagonal concatenation of multiple LMI conditions and the number of decision variables are assessed or estimated.

4.1 Polytopic LPV Synthesis

Parameter-Independent Lyapunov Functions: The size of the LMIs (11) and (12) is determined by

$$\mathsf{s}(\mathcal{N}_R) + n_z = (n_x + n_w - n_y) + n_z \quad \text{and}$$
$$\mathsf{s}(\mathcal{N}_S) + n_w = (n_x + n_z - n_u) + n_w.$$

Note that in order to solve the LMIs a Schur complement with respect to $\frac{1}{\gamma}$ has to be taken each, which accounts for the additional terms n_z and n_w. The dimensions of the basis forming the null-spaces $\mathcal{N}_R$ and $\mathcal{N}_S$ are due to the assumptions on the full row rank, which means that it is derived by the number of columns minus the number of rows, respectively.

Furthermore, if evaluated in a hyperbox the number of LMIs grows with $O(2^{n_\theta})$. Together with the Lyapunov variable coupling condition of size $2n_x$ the total size of the LMI amounts to

$$2^{n_\theta}\left(2(n_x + n_w + n_z) - (n_y + n_u)\right) + 2n_x. \qquad (55)$$

The associated number of decision variables of the existence conditions are limited to the Lyapunov variables $\mathcal{R}$ and $\mathcal{S}$ and amount to $n_x(n_x+1)$. When solving for the controller in the vertices, we again obtain $\mathsf{m}(S^{K^\theta})$ from (44), although closed-form formulae (15)–(16) can also be used, as performed in the MATLAB implementation "`hinfgs`".

Parameter-Dependent Lyapunov Functions: The parameterization of the Lyapunov functions has a strong impact on the synthesis complexity. Assume again that while $\mathcal{S} = \mathcal{S}_0$ is chosen constant, $\mathcal{R}$ is parameterized as (47). The number of decision variables therefore increases to $n_x(n_x + 1)(1 + {}^1\!/_2 n_\theta^{\mathcal{R}})$. Furthermore, the LMI (11) has to be evaluated on $2^{n_\theta + n_\theta^{\mathcal{R}}}$ vertices when considering the extremal values of $(\theta, \nu) \in \boldsymbol{\theta} \times \boldsymbol{\nu}$. The multi-convexity approach further introduces $n_\theta^{\mathcal{R}}$ additional LMI constraints of size $\mathsf{s}(\mathcal{N}_R) + n_z$. The second multi-convexity constraint is not required if only $\mathcal{R}$ is parameter-dependent. Furthermore, as above, we will assume that $\mathscr{B}_\mathrm{p}$ is parameter-independent, such that only LMIs of size n_x are introduced.

Additionally the coupling (13) needs to be verified on the $2^{n_\theta^{\mathcal{R}}}$ vertices, as well. In conclusion the total size of the LMI is

$$\begin{aligned}&(2^{n_\theta + n_\theta^{\mathcal{R}}})\,(n_x + n_w + n_z - n_u)\\ &\quad + 2^{n_\theta}\,(n_x + n_w + n_z - n_y) + n_\theta^{\mathcal{R}} n_x + 2^{n_\theta^{\mathcal{R}}+1} n_x.\end{aligned}$$

4.2 LFT-Based LPV Synthesis

A core advantage in the synthesis of LPV controllers based on the LFT paradigm and the Full-Block $\mathcal{S}$-Procedure consists in decoupling parameter-dependent from parameter-independent LMIs. In addition, the multiplier conditions are quadratic in the parameters and therefore easily convexified by inertia hypotheses (multi-convexity) even in the face of rational parameter-dependence of the plant.

Parameter-Independent Lyapunov Functions: After a Schur argument, the nominal LMIs (28) are of the size

$$\mathsf{s}(\tilde{\mathcal{N}}_R) + n_z = (n_x + n_w + n_\Delta - n_y) + n_z \quad \text{and}$$
$$\mathsf{s}(\tilde{\mathcal{N}}_S) + n_w = (n_x + n_z + n_\Delta - n_u) + n_w,$$

which is again derived from the dimensions of the null-spaces as explained above. The Lyapunov variable coupling condition is again of size $2n_x$ and the multiplier conditions (30) are both of size n_Δ, where in general M and N take the form

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^\top & M_{22}\end{bmatrix}, \quad N = \begin{bmatrix} N_{11} & N_{12} \\ N_{12}^\top & N_{22}\end{bmatrix}, \quad \begin{matrix} M_{11} > 0, \ M_{22} < 0 \\ N_{11} < 0, \ \ N_{22} > 0 \end{matrix}.$$

With the multi-convexity constraints above, the multiplier conditions (30) have to be evaluated at vertices of the convex hull of the parameter range. Assuming a hyperbox, we have 2^{n_δ} LMI constraints on each multiplier and a total size of the concatenated LMIs of

$$2(2n_x + n_w + n_z + n_\Delta) - (n_y + n_u) + 2^{n_\delta + 1} n_\Delta. \qquad (56)$$

As before, the Lyapunov variables require $n_x(n_x+1)$ decision variables and the major increase is due to the size of the multipliers, which can be structurally constrained. For full-block multipliers M and N both require $n_\Delta(2n_\Delta + 1)$ decision variables each. So-called D/G-scalings can be used, which require all blocks of M and N to commute with Δ and

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{12}^\top & -M_{11}\end{bmatrix}, \quad N = \begin{bmatrix} N_{11} & N_{12} \\ N_{12}^\top & -N_{11}\end{bmatrix}, \quad \begin{matrix} M_{12} = -M_{12}^\top \\ N_{12} = -N_{12}^\top \end{matrix}.$$

The commutativity requirement essentially reduces the number of decision variables to the case, where several multiplier conditions involving only a single parameter are solved simultaneously and the individual multiplier block sizes are inferred from the parameter's repetitions, leading to a total of

$$\sum_{i=1}^{n_\delta} r_i^\delta(r_i^\delta + 1) + r_i^\delta(r_i^\delta - 1) = 2\sum_{i=1}^{n_\delta} {r_i^\delta}^2.$$

decision variables for M and N. Without a priori knowledge of the number of repetitions, we consider the limiting cases: Take $n_\delta \to n_\Delta$, which leads to $r_i^\delta = 1, \ i = 1, \ldots, n_\delta$ and therefore the number of decision variables collapses to $2n_\delta$. If $n_\delta \to 1$, D/G-scalings are lossless (Meinsma et al., 1997) and full-block multipliers are not required. We then have $2n_\Delta^2$ decision variables for both multipliers. Note that D/G-scalings render the multiplier conditions trivially fulfilled, such that the total size of the LMI reduces to

$$2(2n_x + n_w + n_z + n_\Delta) - (n_y + n_u). \qquad (57)$$

When solving for the controller variables, the number of decision variables adheres to $\mathsf{m}(S^{K^\delta})$ from (49). Again, closed-form formulae (15)–(16) can also be used.

Parameter-Dependent Lyapunov Functions: Consider again the approach to mimic the plant's parameter-dependence in the Lyapunov variable (53), which introduces $\mathsf{m}(\mathcal{R}) = {}^1\!/_2 n_x(n_x + 1) + n_\Delta^2$ decision variables, as seen from (54). When non-constant null-spaces $\mathcal{N}_R$ and $\mathcal{N}_S$ are considered, the sizes of the resulting LFT blocks in (20) and (21) are upper bounded by

$$\begin{aligned} n_{\Delta_R} &= \mathsf{s}(\Delta_R) = \mathsf{s}(\mathrm{diag}(\dot{\Delta}, \Delta, \Delta, \Delta, \Delta)) = 5n_\Delta,\\ n_{\Delta_S} &= \mathsf{s}(\Delta_S) = \mathsf{s}(\mathrm{diag}(\Delta, \Delta)) = 2n_\Delta,\\ n_{\Delta_{RS}} &= \mathsf{s}(\Delta_{RS}) = \mathsf{s}(\mathrm{diag}(\Delta, \Delta)) = n_\Delta.\end{aligned}$$

At the cost of an increased number of states n_x, in this approach it will usually be beneficial to pre- and postfilter, reducing the problem to the sizes $\mathsf{s}(\Delta_R) = 4n_\Delta$, $\mathsf{s}(\Delta_S) = n_\Delta$. For this case, LMIs (23)–(25) jointly have the size

$$\begin{aligned}&2(n_x + n_w + n_z) - (n_y + n_u) + n_{\Delta_R} + n_{\Delta_S} + (n_{\Delta_{RS}} + 2n_x)\\ &\quad = 2(n_x + n_w + n_z) - (n_y + n_u) + 6n_\Delta + 2n_x\end{aligned}$$

In order to evaluate the multiplier conditions (29) via full-block multipliers, also the rates of change have to be taken into account for conditions on M, which requires the formulation of 2^{2n_δ} LMI constraints. In total the multiplier conditions form an LMI of size

$$2^{n_\delta}(2^{n_\delta}4n_\Delta + 2n_\Delta) = 2^{n_\delta+1}n_\Delta(2^{n_\delta+1}+1),$$

containing

$$1/2\,(4n_\Delta(4n_\Delta+1) + 4n_\Delta(2n_\Delta+1)) = 4n_\Delta(3n_\Delta+1)$$

decision variables. Again, using D/G-scalings the multiplier conditions are trivially fulfilled and the number of multiplier related decision variables reduces to

$$3\sum_{i=1}^{n_\delta} r_i^{\delta 2} + \sum_{i=1}^{n_\delta}(3r_i^\delta)^2 = 12\sum_{i=1}^{n_\delta} r_i^{\delta 2},$$

when treating rates of change independently and regarding the repeated Δ-block structure as a single block with three times the repetitions for each parameter δ_i.

4.3 Gridding-Based LPV Synthesis

Parameter-Independent Lyapunov Functions: For n_g equidistant grid points between the minimum and maximum value of a scheduling signal, the number of LMI constraints grow with $O(n_\mathrm{g}^{n_\rho})$. The size of the LMIs (11) and (12) is identical to the polytopic case, which leads to a total size of the LMI of

$$n_\mathrm{g}^{n_\rho}\,(2(n_x+n_w+n_z) - (n_y+n_u)) + 2n_x. \tag{58}$$

As before, the only decision variables of the existence conditions are the Lyapunov variables $\mathcal{R}$ and $\mathcal{S}$ and amount to $n_x(n_x+1)$. When solving for the controller in the grid points, one needs to solve for $n_\mathrm{g}^{n_\rho}\cdot\mathsf{m}(S^{K^\rho})$ variables as obtained from (55). More typically the closed-form formulae (17)–(18) are used, which further reduce in online complexity for constant Lyapunov functions.

Parameter-Dependent Lyapunov Functions: As in the previous approaches, the parameterization of the Lyapunov functions has a strong impact on the synthesis complexity. Assume again that while $\mathcal{S} = \mathcal{S}_0$ is chosen constant, $\mathcal{R}$ chosen parameter-dependent. Following the heuristic to mimic the plant's parameter-dependence, it appears a natural choice to consider the parameterization (47), which leads to $n_x(n_x+1)(1+1/2n_\theta^\mathcal{R})$ decision variables. However, the Lyapunov matrix can also be chosen to depend on the scheduling signals ρ directly. In any case, the rates of change ν or σ do not have to be gridded, since they enter the matrix inequality in an affine manner. Therefore, LMI (11) has to be evaluated on $2^{n_\theta^\mathcal{R}}n_\mathrm{g}^{n_\rho}$ grid points, whereas LMI (12) is still only considered in $n_\mathrm{g}^{n_\rho}$ grid points. For affine parameterizations of the Lyapunov variable, the coupling (13) needs to be verified on $2^{n_\theta^\mathcal{R}}$ vertices, whereas—perhaps more typically—it is gridded over the $n_\mathrm{g}^{n_\rho}$ grid points. In conclusion the total size of the LMI is

$$2^{n_\theta^\mathcal{R}}n_\mathrm{g}^{n_\rho}\,(2(n_x+n_w+n_z) - (n_y+n_u)) + 2^{n_\theta^\mathcal{R}+1}n_x. \tag{59}$$

5. SUMMARY

Tab. 3 summarizes the above discussion by collecting the complexities in implementation for the different synthesis methods. Tab. 4 in turn collects the complexity in synthesis focussing on the LMI sizes and decision variables for solving the existence conditions with respect to the different synthesis methods. Note that in both tables the number of arithmetic operations, memory requirements, etc. are provided as the sum of individual components. These components are given in the headerless columns to the right, respectively. The formulae are provided online at `www.tuhh.de/~rtsch/HoWe13b` for easy adaptation and use. The notational shortcut $m_{abcd} = n_a + n_b + n_c + n_d$, with $a, b, c, d \in \{x, u, y, w, z, \Delta, \{\}, \dots\}$ is used, such that, e.g., $m_{xu} = n_x + n_u$ or $m_{xw\Delta u} = n_x + n_w + n_\Delta + n_u$.

6. NUMERICAL EXAMPLES

6.1 Robotic Manipulator

In the following, the example of a three-degrees-of-freedom (3-DOF) robotic manipulator is considered and the involved complexity for each synthesis method will be analyzed. The problem has been extensively studied and detailed model representations for both 2-DOF and 3-DOF control problems can be found in (Hashemi et al., 2009, 2012; Hoffmann et al., 2013a; Hoffmann and Werner, 2014), as well as online, including Matlab files at `www.tuhh.de/~rtsch/HoHaAbWe13`. A compact LFR using a block-diagonal scheduling block, which is called Υ to distinguish it from the diagonal block Δ, is proposed in Hoffmann and Werner (2014). It is derived based on the standard formulation of differential equations for mechanical systems

$$M(q)\ddot{q} + D(\dot{q}, q)\dot{q} + K(q)q = u,$$

which essentially leads to $\Upsilon = \mathrm{diag}(M, D, K)$, except for one zero row and column in K. The matrix Υ can be affinely parameterized by ten scheduling parameters denoted υ_i, $i = 1, \dots, 10$. The block-diagonal structure prohibits the use of D/G-scalings in LFT-based LPV controller synthesis due to an excessive amount of conservatism. Instead full-block multipliers are used. Additionally a second multiplier stage can be incorporated to avoid the evaluation in the vertices, introducing additional decision variables. The benefits are discussed in Hoffmann and Werner (2014) and omitted here for lack of space. Using standard Matlab tools, the LFR can be converted into a form, which uses a diagonal scheduling block $\hat{\Upsilon}(\upsilon) = \operatorname*{diag}_{i=1}^{n_\upsilon=10}\left(\upsilon_i I_{r_i^\upsilon}\right)$. The parameters υ in turn are rational functions of the parameters δ, which are mostly defined as angular velocities and sine and cosine terms of joint angles, see Hoffmann and Werner (2014) for details. Using this relationship, an LFR with scheduling block $\Delta(\delta) = \operatorname*{diag}_{i=1}^{n_\delta=9}\left(\delta_i I_{r_i^\delta}\right)$ can be generated. The general LPV model is revealed by defining scheduling signals ρ as joint angles two and three, as well as angular velocities of joints one to three. In these the model is no longer rational due to the trigonometric functions and the gridding approach is therefore applied. It is also possible to derive an affine LPV model for the robot incorporating 16 scheduling parameters θ_i, $i = 1, \dots, 16$. Naturally, the affine model can also be represented by an LFT-based model. The technique of parameter set mapping introduced in Kwiatkowski and Werner (2008) is taken up here, to heuristically reduce the number of affine parameters based on a principal component analysis of experimental data or parameter values

Table 3. Implementation complexity of LPV controllers vs. synthesis technique.

Technique	PDLF		Implementation complexity: Arith. ops. $\mathsf{a}(\cdot)$		Mem. req. $\mathsf{m}(\cdot)$	
Poly.	×		Polytopic: $\mathsf{a}(S^{K^\theta}) + \mathsf{a}(\alpha)$	$\mathsf{a}(\alpha) = 2^{n_\theta+2} + 3n_\theta - 2$ $\mathsf{a}(S^{K^\theta}) = (2^{n_\theta+1} - 1)m_{xu}m_{xy}$	Polytopic: $\mathsf{m}(S^{K^\theta})$	$\mathsf{m}(S^{K^\theta}) = 2^{n_\theta} m_{xu}m_{xy}$
			Affine: $\mathsf{a}(S^{K^\theta})$	$\mathsf{a}(S^{K^\theta}) = 2n_\theta m_{xu}m_{xy}$	Affine: $\mathsf{m}(S^{K^\theta})$	$\mathsf{m}(S^{K^\theta}) = (n_\theta + 1)m_{xu}m_{xy}$
	✓		$\mathsf{a}(S^{K^\rho}) + \mathsf{a}(\mathcal{R})$	$\mathsf{a}(\mathcal{R}) = 2n_\theta^{\mathcal{R}} n_x^2$	$\mathsf{m}(S^\rho) + \mathsf{m}(\mathcal{R}) + 2\mathsf{m}(\mathcal{S}_0)$	$\mathsf{m}(\mathcal{R}) = {}^1\!/_2 n_x(n_x + 1)(n_\theta^{\mathcal{R}} + 1)$ $\mathsf{m}(\mathcal{S}_0) = {}^1\!/_2 n_x(n_x + 1)$
LFT	×	FBM D/G D/G w/ Θ	$\mathsf{a}(S^{K^\delta}) + \mathsf{a}(\Psi) + \mathsf{a}(\Delta^K(\Delta))$ $\mathsf{a}(S^{K^\delta}) + \mathsf{a}(\Psi)$ $\mathsf{a}(S^{K^\theta})$	$\mathsf{a}(\Psi) = n_\Delta\left({}^2\!/_3 n_\Delta{}^2 + 2n_\Delta + 1\right)$ $\mathsf{a}(\Delta^K(\Delta)) = n_\Delta^2\left({}^{52}\!/_3 n_\Delta - 1\right)$ $\mathsf{a}(S^{K^\delta}) = n_\Delta m_{xu}\left(2m_{xy} + 1\right)$ $\mathsf{a}(S^{K^\theta}) = n_\Theta m_{xu}\left(2m_{xy} + 1\right)$	$\mathsf{m}(S^{K^\delta}) + \mathsf{m}(\Delta^K(\Delta))$ $\mathsf{m}(S^{K^\delta})$ $\mathsf{m}(S^{K^\theta})$	$\mathsf{m}(\Delta^K(\Delta)) = 7n_\Delta(n_\Delta + 1)$ $\mathsf{m}(S^{K^\delta}) = m_{x\Delta u}m_{x\Delta y}$ $\mathsf{m}(S^{K^\theta}) = m_{x\Theta u}m_{x\Theta y} - n_\Theta^2$
	✓		$\mathsf{a}(S^{K^\rho}) + \mathsf{a}(\mathcal{R})$	$\mathsf{a}(\mathcal{R}) = 2n_x\left(3n_x n_\Delta + 2n_\Delta^2 + 2n_x^2 - m_{x\Delta}\right) + \mathsf{a}(\Psi)$	$\mathsf{m}(S^\rho) + \mathsf{m}(\mathcal{R}) + 2\mathsf{m}(\mathcal{S}_0)$	$\mathsf{m}(\mathcal{R}) = {}^1\!/_2 n_x(n_x + 1) + n_\Delta^2$ $\mathsf{m}(\mathcal{S}_0) = {}^1\!/_2 n_x(n_x + 1)$
Grid.	×		$\mathsf{a}(S^{K^\rho}) - \mathsf{a}(\mathcal{M}^{-\top}) - \mathsf{a}(\mathcal{R}^{-1})$	$\mathsf{a}(\mathcal{M}^{-\top}) \approx \mathsf{a}(\mathcal{R}^{-1}) \leqslant {}^2\!/_3 n_x^3$	$\mathsf{m}(S^\rho) + 4\mathsf{m}(\mathcal{R}_0) + \mathsf{m}(\mathcal{M}^{-\top})$	$\mathsf{m}(\mathcal{R}_0) = {}^1\!/_2 n_x(n_x + 1)$ $\mathsf{m}(\mathcal{M}^{-\top}) = n_x^2$
	✓		$\mathsf{a}(S^{K^\rho}) + \mathsf{a}(\mathcal{R})$	$\mathsf{a}(\mathcal{R}) = 2n_\theta^{\mathcal{R}} n_x^2$	$\mathsf{m}(S^\rho) + \mathsf{m}(\mathcal{R}) + 2\mathsf{m}(\mathcal{S}_0)$	$\mathsf{m}(\mathcal{R}) = {}^1\!/_2 n_x(n_x + 1)(n_\theta^{\mathcal{R}} + 1)$ $\mathsf{m}(\mathcal{S}_0) = {}^1\!/_2 n_x(n_x + 1)$

$\mathsf{a}(S^{K^\rho}) = {}^{46}\!/_3 n_x^3 + (6m_{uy} + 2m_{wz} - 2)n_x^2 + (n_w + m_{uz}(2n_u - 1) + 2n_u n_z - m_{uyw} + 4n_y n_w + 1)n_x + 2n_z n_u^2 + 2n_w n_y^2 + {}^2\!/_3 n_u^3 + {}^2\!/_3 n_y^3$
$\mathsf{m}(S^\rho) = m_{xzy}m_{xwu} - n_z n_w - n_y n_u$

Table 4. Synthesis complexity of LPV controllers vs. synthesis technique.

Technique	PDLF		Synthesis complexity (existence conditions): Size of LMI $\mathsf{s}(\cdot)$		No. of dec. vars. $\mathsf{d}(\cdot)$	
Poly.	×		$2^{n_\theta}(\mathsf{s}(\mathcal{L}_R) + \mathsf{s}(\mathcal{L}_S)) + \mathsf{s}(\mathcal{L}_{RS})$	$\mathsf{s}(\mathcal{L}_R) = m_{xzw} - n_y$ $\mathsf{s}(\mathcal{L}_S) = m_{xzw} - n_u$, $\mathsf{s}(\mathcal{L}_{RS}) = 2n_x$	$\mathsf{d}(\mathcal{R}_0) + \mathsf{d}(\mathcal{S}_0)$	$\mathsf{d}(\mathcal{R}_0) = \mathsf{d}(\mathcal{S}_0) = {}^1\!/_2 n_x(n_x + 1)$
	✓		$2^{n_\theta + n_\theta^{\mathcal{R}}}\mathsf{s}(\mathcal{L}_R) + n_\theta^{\mathcal{R}}\mathsf{s}(\frac{\partial^2}{\partial\theta_i^2}\mathcal{L}_R)$ $+2^{n_\theta}\mathsf{s}(\mathcal{L}_S) + 2^{n_\theta^{\mathcal{R}}}\mathsf{s}(\mathcal{L}_{RS})$	$\mathsf{s}(\frac{\partial^2}{\partial\theta_i^2}\mathcal{L}_R) = n_x$	$\mathsf{d}(\mathcal{R}) + \mathsf{d}(\mathcal{S}_0)$	$\mathsf{d}(\mathcal{R}) = {}^1\!/_2 n_x(n_x + 1)(n_\theta^{\mathcal{R}} + 1)$
LFT	×	FBM	$\mathsf{s}(\tilde{\mathcal{L}}_R) + \mathsf{s}(\tilde{\mathcal{L}}_S) + \mathsf{s}(\tilde{\mathcal{L}}_{RS})$ $+2^{n_\delta}(\mathsf{s}(\tilde{\mathcal{L}}_M) + \mathsf{s}(\tilde{\mathcal{L}}_N))$	$\mathsf{s}(\tilde{\mathcal{L}}_R) = m_{x\Delta zw} - n_y,\ \mathsf{s}(\tilde{\mathcal{L}}_M) = n_\Delta$ $\mathsf{s}(\tilde{\mathcal{L}}_S) = m_{x\Delta zw} - n_u,\ \mathsf{s}(\tilde{\mathcal{L}}_N) = n_\Delta$ $\mathsf{s}(\tilde{\mathcal{L}}_{RS}) = 2n_x$	$\mathsf{d}(\mathcal{R}_0) + \mathsf{d}(\mathcal{S}_0) + \mathsf{d}(M) + \mathsf{d}(N)$	$\mathsf{d}(\mathcal{R}_0) = \mathsf{d}(\mathcal{S}_0) = {}^1\!/_2 n_x(n_x + 1)$ $\mathsf{d}(M) = \mathsf{d}(N) = n_\Delta(2n_\Delta + 1)$
		D/G	$\mathsf{s}(\tilde{\mathcal{L}}_R) + \mathsf{s}(\tilde{\mathcal{L}}_S) + \mathsf{s}(\tilde{\mathcal{L}}_{RS})$	$\mathsf{s}(\tilde{\mathcal{L}}_R) = m_{x\Delta zw} - n_y,\ \mathsf{s}(\tilde{\mathcal{L}}_{RS}) = 2n_x$ $\mathsf{s}(\tilde{\mathcal{L}}_S) = m_{x\Delta zw} - n_u$	$\mathsf{d}(\mathcal{R}_0) + \mathsf{d}(\mathcal{S}_0) + \mathsf{d}(M) + \mathsf{d}(N)$	$\mathsf{d}(\mathcal{R}_0) = \mathsf{d}(\mathcal{S}_0) = {}^1\!/_2 n_x(n_x + 1)$ $\mathsf{d}(M) = \mathsf{d}(N) = \sum_{i=1}^{n_\delta} r_i^{\delta^2}$ $2n_\delta \leqslant \mathsf{d}(M) \leqslant 2n_\Delta^2$
	✓	FBM	$\mathsf{s}(\mathcal{L}_R) + \mathsf{s}(\mathcal{L}_S) + \mathsf{s}(\mathcal{L}_{RS})$ $+2^{2n_\delta}\mathsf{s}(\mathcal{L}_M) + 2^{n_\delta}(\mathsf{s}(\mathcal{L}_N) + \mathsf{s}(\mathcal{L}_P))$	$\mathsf{s}(\mathcal{L}_R) = m_{xzw} + 4n_\Delta - n_y$ $\mathsf{s}(\mathcal{L}_S) = m_{xzw} + n_\Delta - n_u,\ \mathsf{s}(\mathcal{L}_M) = 4n_\Delta$ $\mathsf{s}(\mathcal{L}_{RS}) = 2n_x + n_\Delta,\ \mathsf{s}(\mathcal{L}_N) = \mathsf{s}(\mathcal{L}_P) = n_\Delta$	$\mathsf{d}(\mathcal{R}) + \mathsf{d}(\mathcal{S}_0)$ $+\mathsf{d}(M) + \mathsf{d}(N) + \mathsf{d}(P)$	$\mathsf{d}(\mathcal{R}) = {}^1\!/_2 n_x(n_x + 1) + n_\Delta^2$ $\mathsf{d}(\mathcal{S}_0) = {}^1\!/_2 n_x(n_x + 1)$ $\mathsf{d}(M) = 4n_\Delta(8n_\Delta + 1)$ $\mathsf{d}(N) = \mathsf{d}(P) = n_\Delta(2n_\Delta + 1)$
		D/G	$\mathsf{s}(\mathcal{L}_R) + \mathsf{s}(\mathcal{L}_S) + \mathsf{s}(\mathcal{L}_{RS})$	$\mathsf{s}(\mathcal{L}_R) = m_{xzw} + 4n_\Delta - n_y$, $\mathsf{s}(\mathcal{L}_S) = m_{xzw} + n_\Delta - n_u$, $\mathsf{s}(\mathcal{L}_{RS}) = 2n_x + n_\Delta$	$\mathsf{d}(\mathcal{R}) + \mathsf{d}(\mathcal{S}_0)$ $+\mathsf{d}(M) + \mathsf{d}(N) + \mathsf{d}(P)$	$\mathsf{d}(\mathcal{R}) = {}^1\!/_2 n_x(n_x + 1) + n_\Delta^2$ $\mathsf{d}(\mathcal{S}_0) = {}^1\!/_2 n_x(n_x + 1)$ $\mathsf{d}(M) = 10\sum_{i=1}^{n_\delta} r_i^{\delta^2}$ $\mathsf{d}(N) = \mathsf{d}(P) = \sum_{i=1}^{n_\delta} r_i^{\delta^2}$
Grid.	×		$n_{\mathrm{g}}^{n_\rho}(\mathsf{s}(\mathcal{L}_R) + \mathsf{s}(\mathcal{L}_S)) + \mathsf{s}(\mathcal{L}_{RS})$	$\mathsf{s}(\mathcal{L}_R) = m_{xzw} - n_y$ $\mathsf{s}(\mathcal{L}_S) = m_{xzw} - n_u$, $\mathsf{s}(\mathcal{L}_{RS}) = 2n_x$	$\mathsf{d}(\mathcal{R}_0) + \mathsf{d}(\mathcal{S}_0)$	$\mathsf{d}(\mathcal{R}_0) = \mathsf{d}(\mathcal{S}_0) = {}^1\!/_2 n_x(n_x + 1)$
	✓		$n_{\mathrm{g}}^{n_\rho}(2^{n_\theta^{\mathcal{R}}}\mathsf{s}(\mathcal{L}_R) + \mathsf{s}(\mathcal{L}_{RS}) + \mathsf{s}(\mathcal{L}_S))$	$\mathsf{s}(\mathcal{L}_R) = m_{xzw} - n_y$ $\mathsf{s}(\mathcal{L}_S) = m_{xzw} - n_u$, $\mathsf{s}(\mathcal{L}_{RS}) = 2n_x$	$\mathsf{d}(\mathcal{R}) + \mathsf{d}(\mathcal{S}_0)$	$\mathsf{d}(\mathcal{R}) = {}^1\!/_2 n_x(n_x + 1)(n_\theta^{\mathcal{R}} + 1)$

gridded over the scheduling signals ρ. In this course, the affine parameters θ are approximated by the parameters $\phi_i,\ i = 1, \ldots, 4$ (Hoffmann et al., 2013a). Note that this technique can also be applied to the parameters υ in which the scheduling block Υ is affine, which is omitted here for brevity.

The numbers of scheduling signals and parameters associated with each modeling framework are summarized in Tab. 5(a). The scheduling block sizes and associated repetitions for the LFRs are given in Tab. 5(c) and signal dimensions are provided in Tab. 5(b).

Tab. 6 lists all complexity indicators computed *a priori* based on the model data. The last column indicates an estimation of whether synthesis is considered tractable or no definitive statement can be made. Values in red text color are considered critical and either prevent successful implementation/synthesis or at least endanger it. Reasons for the latter include numerical difficulties on common PC hardware. The colored bars behind these values are not to scale. Instead their length is relative to the highest value written in black text color of the respective column. Note that even though the figures do not appear extraordinarily large, synthesis approaches associated with values marked in red with respect to the full-block multiplier based LFT approach have been tried and discussed in Hoffmann and Werner (2014) and were found intractable.

Tab. 6 indicates a limited applicability of polytopic and general intractability of gridding methods for the considered problem.

Table 5. Problem sizes

(a) No. of sched. signals/params.

Sched. params.	No. Robot	No. SI Eng.
n_ρ	5	3
n_δ	9	3
n_θ	16	3
n_ϕ	4	2
n_υ	10	—

(b) Signal sizes.

Signal	Size Robot	Size SI Eng.
n_x	21	4
n_u	3	1
n_y	6	1
n_w	3	1
n_z	6	2

(c) LFT block size and no. of repetitions.

LFT block	Size Robot	Size SI Eng.	Repetitions r_i Robot	Repetitions r_i SI Eng.
Δ	37	6	$r^\delta = [8\ 8\ 8\ 6\ 1\ 1\ 3\ 1\ 1]$	$r^\delta = [3\ 2\ 1]$
Θ	16	3	$r^\theta_i = 1,\ i = 1, \ldots, 16$	$r^\theta_i = 1,\ i = 1, \ldots, 3$
Φ	10	4	$r^\phi = [3\ 3\ 1\ 3]$	$r^\phi = [2\ 2]$
Υ	8	—	—	—
$\hat{\Upsilon}$	15	—	$r^\upsilon = [1\ 2\ 2\ 2\ 2\ 2\ 1\ 1\ 1\ 1]$	—

Table 6. 3-DOF robot example: Implementation and synthesis complexity.

Technique	PDLF			Impl. complexity: Arith. ops.	Impl. complexity: Mem.	Synth. complexity: LMI size	Synth. complexity: Dec. vars.	Tract.
Poly.	×	PSM	θ	20,736	11,016	$\approx 3.3 \cdot 10^6$	462	×
	×	PSM	ϕ	5,184	3,240	858	462	✓
	✓	PSM	θ	190,515	5,343	$\approx 102.3 \cdot 10^9$	4,158	×
	✓	PSM	ϕ	179,931	2,571	7,332	4,158	?
LFT	×	FBM	Δ	962,000	13,746	38,055	6,012	×
	×	FBM	Υ	19,848	1,624	16,493	734	✓
	×	FBM	$\hat{\Upsilon}$	80,790	3,318	30,843	1,392	×
	×	FBM	Θ	95,120	3,624	$\approx 2.0 \cdot 10^6$	1,518	×
	×	FBM	Φ	31,310	2,028	433	882	✓
	×	D/G	Δ	85,384	3,904	167	944	✓
	×	D/G	$\hat{\Upsilon}$	22,515	1,638	123	512	✓
	×	D/G	Θ	21,120	1,464	125	494	✓
	×	D/G	Φ	5,280	1,158	113	518	✓
	✓	FBM	Δ	460,450	3,016	$\approx 38.8 \cdot 10^6$	51,337	×
	✓	FBM	Υ	239,250	1,711	$\approx 33.6 \cdot 10^6$	2,878	×
	✓	FBM	$\hat{\Upsilon}$	273,240	1,872	$\approx 62.9 \cdot 10^6$	8,877	×
	✓	FBM	Θ	278,990	1,903	$\approx 270.0 \cdot 10^9$	10,030	×
	✓	FBM	Φ	247,880	1,747	10,713	4,222	?
	✓	D/G	Δ	460,450	3,016	315	4,723	?
	✓	D/G	$\hat{\Upsilon}$	273,240	1,872	183	987	✓
	✓	D/G	Θ	278,990	1,903	189	910	✓
	✓	D/G	Φ	247,880	1,747	153	898	✓
Grid.	×	$n_g = 4$				52,266		?
	×	$n_g = 8$		164,055	2,319	$\approx 1.7 \cdot 10^6$	462	×
	×	$n_g = 12$				$\approx 12.6 \cdot 10^6$		×
	✓	$n_g = 4$				$\approx 1.6 \cdot 10^9$		×
	✓	$n_g = 8$		190,515	5,343	$\approx 52.0 \cdot 10^9$	4158	×
	✓	$n_g = 12$				$\approx 390.0 \cdot 10^9$		×

6.2 *Charge Control of a Spark-Ignited Engine*

Next, the charge control of a spark-ignited (SI) engine is considered as an example of more moderate complexity. Details on the model can be found in (Kwiatkowski et al., 2006; Hoffmann et al., 2014b). The numbers of scheduling signals, parameters, LFT block sizes and repetitions associated with each modeling framework are again summarized in Tab. 5(a), 5(b) and 5(c), respectively. The block-diagonal LFT-based modeling approach from Hoffmann and Werner (2014) is not pursued here. Tab. 7 lists the *a priori* complexity indicators in the same fashion as before. Levels of complexity deemed intractable are altered with respect to implementation, as in automotive applications usually electronic control units (ECUs) impose limits on controller complexity (Kwiatkowski et al., 2009).

Table 7. SI Engine example: Implementation and synthesis complexity.

Technique	PDLF			Impl. complexity: Arith. ops.	Impl. complexity: Mem.	Synth. complexity: LMI size	Synth. complexity: Dec. vars.	Tract.
Poly.	×	PSM	θ	150	100	104	20	✓
	×	PSM	ϕ	100	75	56	20	✓
	✓	PSM	θ	1,380	99	508	50	✓
	✓	PSM	ϕ	1,349	89	160	50	✓
LFT	×	FBM	Δ	4,260	415	128	176	✓
	×	FBM	Θ	663	148	74	62	✓
	×	FBM	Φ	1,392	221	60	92	✓
	×	D/G	Δ	552	121	32	48	✓
	×	D/G	Θ	165	55	26	26	✓
	×	D/G	Φ	110	65	28	36	✓
	✓	FBM	Δ	2,835	105	1,688	1,388	✓
	✓	FBM	Θ	1,958	78	854	371	✓
	✓	FBM	Φ	2,195	85	332	636	✓
	✓	D/G	Δ	2,835	105	56	224	✓
	✓	D/G	Θ	1,958	78	38	65	✓
	✓	D/G	Φ	2,195	85	44	132	✓
Grid.	×	$n_g = 4$				776		✓
	×	$n_g = 8$		1,199	95	6,152	20	?
	×	$n_g = 12$				20,744		×
	✓	$n_g = 4$				3,968		?
	✓	$n_g = 8$		1,380	99	31,744	50	×
	✓	$n_g = 12$				107,136		×

6.3 *Discussion*

In the first example, gridding approaches are clearly ruled out due to their high synthesis complexity even for low grid densities. This situation changes, when a problem of fewer scheduling signals is considered, as in the SI engine case. With non-evenly spaced grids, the gridding approach can maintain tractable LMI sizes, while PDLF-based synthesis provides improved performance at little additional cost during implementation. In fact, implementation can even appear less costly than non-PDLF-based synthesis using LFT techniqes and FBM. In contrast, the gap in implementation complexity between parameter-independent and parameter-dependent Lyapunov functions appears much larger in both the LFT- and the polytopic approach. Taking into consideration that the gridding-based synthesis can accommodate nonconvex parameter ranges, the resulting controller may typically incur little conservatism. An interesting aspect can be seen in the implementation complexity of non-PDLF LFT-based synthesis with rational parameters and FBM. The matrix inversions are very costly due to the large size of the scheduling block Δ. This shows that unless a compact LFR can be found, as in Hoffmann and Werner (2014), the reduced conservatism of FBMs come at the price of a greatly increased implementation complexity. Since even PDLF-based controllers can be less complex to implement, the use of closed-form formulae (17)–(18) appears advantageous.

7. CONCLUSION

In this paper, a survey of implementation and synthesis complexity associated with polytopic, LFT- and gridding-based LPV output-feedback controller synthesis is pursued. The complexity is assessed in terms of the number of arithmetic operations and memory requirements, as well as the size of the LMIs of the existence conditions and the associated number of decision variables. Formulae are made available online for adaptation and immediate application to *a priori* assessments. A three-degrees-of-freedom robotic manipulator with a relatively high number of both scheduling signals and parameters, as well as a more moderate size problem are considered as numerical examples. The examples are analyzed with respect to the LPV modeling frameworks and synthesis methods, some of which have already been successfully applied in previous literature.

REFERENCES

Apkarian, P. and Adams, R.J. (1998). Advanced gain-scheduling techniques for uncertain systems. *IEEE Trans. Contr. Syst. Technol.*, 6(1), 21–32.

Apkarian, P. and Gahinet, P. (1995). A Convex Characterization of Gain-Scheduled H∞ Controllers. *IEEE Trans. Automat. Contr.*, 853–864.

Apkarian, P., Gahinet, P., and Becker, G. (1995). Self-Scheduled H∞ Control of Linear Parameter-varying Systems: a Design Example. *Automatica*, 31(9), 1251–1261.

Dettori, M. and Scherer, C.W. (2001). LPV design for a CD player: an experimental evaluation of performance. In *Proc. 40th IEEE Conf. Decision Contr.*

Gahinet, P., Apkarian, P., and Chilali, M. (1996). Affine parameter-dependent Lyapunov functions and real parametric uncertainty. *IEEE Trans. Automat. Contr.*, 41(3), 436–442.

Gonzalez, A.M., Hoffmann, C., Radisch, C., and Werner, H. (2013). LPV Observation and Damping Control of Container Crane Load Swing. In *Europ. Control Conf.*

Hashemi, S.M., Abbas, H.S., and Werner, H. (2009). LPV modelling and control of a 2-DOF robotic manipulator using PCA-based parameter set mapping. In *Proc. 48th IEEE Conf. Decision Control*.

Hashemi, S.M., Abbas, H.S., and Werner, H. (2012). Low-complexity linear parameter-varying modeling and control of a robotic manipulator. *Contr. Eng. Prac.*, 20(3), 248–257.

Hecker, S., Varga, A., and Magni, J.F. (2004). Enhanced LFR-toolbox for MATLAB and LFT-based gain scheduling. In *ONERA-DLR Aerospace Symposium*.

Hoffmann, C., Hashemi, S.M., Abbas, H.S., and Werner, H. (2013a). Benchmark Problem - Nonlinear Control of a 3-DOF Robotic Manipulator. In *Proc. 52nd IEEE Conf. Decision Control*.

Hoffmann, C., Hashemi, S.M., Abbas, H.S., and Werner, H. (2013b). Synthesis of LPV Controllers with Reduced Implementation Complexity. In *Proc. 52nd IEEE Conf. Decision Control*.

Hoffmann, C., Hashemi, S.M., Abbas, H.S., and Werner, H. (2014a). Synthesis of LPV Controllers with Low Implementation Complexity Based on a Reduced Parameter Set. *IEEE Trans. Contr. Syst. Technol.*

Hoffmann, C., Hashemi, S.M., Abbas, H.S., and Werner, H. (2014b). Synthesis of LPV Controllers with Reduced Implementation Complexity. In *Proc. Amer. Control Conf.*

Hoffmann, C. and Werner, H. (2014). Linear Parameter-Varying Control of Complex Mechanical Systems. In *Proc. 19th IFAC World Congr.*

Iwasaki, T. and Shibata, G. (2001). LPV System Analysis via Quadratic Separator for Uncertain Implicit Systems. *IEEE Trans. Automat. Contr.*, 46(8), 1195–1208.

Kwiatkowski, A., Blath, J.P., Werner, H., and Schultalbers, M. (2006). Application of LPV gain scheduling to charge control of a SI engine. In *Proc. Conf. Comput. Aided Control Syst. Des.*

Kwiatkowski, A. and Werner, H. (2008). PCA-Based Parameter Set Mappings for LPV Models With Fewer Parameters and Less Overbounding. *IEEE Trans. Contr. Syst. Technol.*, 16(4), 781–788.

Kwiatkowski, A., Werner, H., Blath, J.P., Ali, A., and Schultalbers, M. (2009). Linear Parameter Varying PID Controller Design for Charge Control of a Spark-Ignited Engine. *Contr. Eng. Prac.*, 17(11), 1307–1317.

Meinsma, G., Shrivastava, Y., and Fu, M. (1997). A dual formulation of mixed μ and on the losslessness of (D, G) scaling. *IEEE Trans. Automat. Contr.*, 42(7), 1032–1036.

Saupe, F. and Pfifer, H. (2011). Applied LPV control exploiting the separation principle for the single axis positioning of an industrial manipulator. In *Proc. Int. Conf. Control Applicat.*

Saupe, F. and Pfifer, H. (2012). An Observer Based State Feedback LFT LPV Controller for an Industrial Manipulator. In *7th IFAC Symp. Robust Control Des.*

Scherer, C.W. (2000). Robust Mixed Control and LPV Control with Full Block Scalings. In L.E. Ghaoui and S.I. Niculescu (eds.), *Advances in linear matrix inequality methods in control*. SIAM.

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

Scorletti, G. and Ghaoui, L.E. (1998). Improved LMI Conditions for Gain Scheduling and Related Control Problems. *Int. J. Robust Nonlinear Contr.*, 8, 845–877.

Shamma, J.S. and Athans, M. (1990). Analysis of gain scheduled control for nonlinear plants. *IEEE Trans. Automat. Contr.*, 35(8), 898–907.

Shamma, J.S. and Athans, M. (1991). Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27(3), 559–564.

Warren, J., Schaefer, S., Hirani, A.N., and Desbrun, M. (2007). Barycentric coordinates for convex sets. *Adv. Comput. Math.*, 27(3), 319–338.

Wu, F. (1995). *Control of Linear Parameter Varying Systems*. Ph.D. thesis, University of Berkeley, California and USA. Dissertation.

Wu, F. (2001). A Generalized LPV System Analysis and Control Synthesis Framework. *Int. J. Contr.*, 74(7), 745–759.

Wu, F. and Dong, K. (2006). Gain-scheduling control of LFT systems using parameter-dependent Lyapunov functions. *Automatica*, 42(1), 39–50.

Wu, F., Yang, X., Packard, A.K., and Becker, G. (1996). Induced L2-Norm Control for LPV Systems with Bounded Parameter Variation Rates. *Int. J. Robust Nonlinear Contr.*, 6(9-10), 983–998.