Abstract: In standard linear fractional representation (LFR)-based linear parameter-varying (LPV) modeling the size of the (diagonal) scheduling block depends on the number of scheduling parameters and their repetitions, which in turn influences both the complexity of synthesis conditions and the computational load during online implementation of LPV controllers. A modeling framework motivated by, but not limited to, mechanical systems is proposed, where the size of the scheduling block depends on the system’s physical degrees-of-freedom. The scheduling block then turns out block-diagonal and can be parameterized in an affine or rational manner. This parameterization yields less complex LFRs when considering the example of a three degrees-of-freedom robotic manipulator, for which then full-block multipliers are tractable and also necessary in synthesis. Synthesis and both simulation and experimental implementation results indicate that the novel rational LPV controller provides improved performance at both reduced implementation and synthesis complexity as compared to an affine LPV controller.

1. INTRODUCTION

Linear parameter-varying (LPV) (Rugh and Shamma, 2000) systems are linear systems which depend on time-varying parameters referred to as scheduling parameters. They are capable of representing many nonlinear and time-varying systems via the notion of quasi-LPV systems in which the scheduling parameters are functions of states, inputs and/or outputs. Linear matrix inequality (LMI)-based linear time-invariant (LTI) control techniques have been extended to such systems. Linear fractional transformation (LFT)-based synthesis techniques employing the full-block $S$-procedure (Scherer, 2000) provide means to trade conservatism against synthesis complexity via structural constraints on multipliers. LFT-based techniques allow for a rational parameter dependence, which can reduce or avoid overbounding the parameter range Kisielowski and Werner (2008). Furthermore, the LFT framework in conjunction with full-block multipliers allows for non-diagonal scheduling blocks, a potential already stated back in Scherer (2000), but—to the best of the authors’ knowledge—overlooked since. In LFT LPV synthesis, even if the least amount of conservatism is desired, only the number of LMI constraints on the multipliers grows exponentially with the number of scheduling parameters. Hence, the smaller the multiplier (and consequently the plant’s scheduling function), the more LMI constraints are tractable. In addition, parameter-dependent invariance in mechanical systems increases the number of parameter repetitions in standard LFT representations using diagonal scheduling blocks due to the involved rational dependency.

In this paper, we propose an explicit modeling framework for systems resembling differential equations common in mechanical systems. The inversion of the inertia matrix is considered via the LFT framework, which results in a block-diagonal scheduling block. The size of the scheduling block depends on the physical degrees of freedom and is therefore independent from the LPV parameterization. For illustration, a three degrees-of-freedom robotic manipulator is considered, for which full-block multiplier-based synthesis is now tractable. Furthermore, the proposed modeling approach yields less complex rational models with diagonal scheduling blocks than what has been achieved previously despite employing available LFR reduction tools from Matlab. Even when using the well-known $D/G$-scalings with these latter models, the new modeling approach yields a controller that is computationally less expensive during online implementation. Additionally, a two-stage approach to the application of the full-block $S$-procedure can trade LMI constraints versus decision variables and promises the ability to tackle problems of even higher scheduling complexity, as well as selecting parameterizations of the scheduling block other than affine ones for reduced overbounding.

In Section 2, notation is given and LFT-based LPV controller synthesis is reviewed. In Section 3.1, the novel modeling approach is presented. Extensions to the evaluation of multiplier conditions are discussed in Section 3.2. The ideas are applied to a 3-DOF robotic manipulator and discussed in Section 4. Conclusions are drawn in Section 5.

2. PRELIMINARIES

Notation: An upper LFT is denoted by $\Delta = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$, whereas the lower LFT is given by $\Delta = M_{11} + M_{12} \Delta (I - M_{22} \Delta)^{-1} M_{21}$. The symmetric completion of a matrix is denoted by $\bullet$. Time dependence is regularly dropped, e.g. $\theta = \theta(t)$. The nullspace of some matrix $M$ is denoted $\ker(M)$. For a (real-rational proper) transfer matrix $G : \mathbb{R} \to \mathbb{C}^{z \times w}$, define $G^\dagger(s) = G^\top(-s)$. Therefore, $G = \begin{bmatrix} A \\ C \end{bmatrix}$.
and $-G^* = \begin{bmatrix} \begin{bmatrix} A^T \\ B^T \\ C^T \\ D^T \end{bmatrix} \end{bmatrix}$. Let $G_{ss} = (A, B, C, D)$ collect the state space model matrices. For compact notation, we follow Scherer (2012) and define

$$
\mathcal{L}(X, \Pi, G_{ss}) := \begin{bmatrix} \begin{array}{c} 0 \\ A \\ B \\ C \\ D \end{array} \end{bmatrix}^T \begin{bmatrix} X \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} I \end{bmatrix}.
$$

(1)

2.1 LPV Model Representations

Consider a plant with rational parameter-dependence

$$
\begin{bmatrix} P_1^\delta \\ P_2^\delta \end{bmatrix} = \begin{bmatrix} A & B_a & B_p & B_u \\ C_a & D_{aa} & D_{ap} & D_{au} \\ C_p & D_{pa} & D_{pp} & D_{pu} \\ C_y & D_{ya} & D_{yp} & D_{yu} \end{bmatrix} = \begin{bmatrix} p_{h0}^\delta & p_{h1}^\delta & p_{h2}^\delta & p_{h3}^\delta \\ p_{g0}^\delta & p_{g1}^\delta & p_{g2}^\delta & p_{g3}^\delta \\ p_{w0}^\delta & p_{w1}^\delta & p_{w2}^\delta & p_{w3}^\delta \\ p_{y0}^\delta & p_{y1}^\delta & p_{y2}^\delta & p_{y3}^\delta \end{bmatrix},
$$

(2)

$$
P_3^\delta = \Delta(\delta(t)) \ast P_0^\delta, \quad \Delta \in \mathbb{R}^{n \times n}. 
$$

(3)

The respective channels indicated by subscripts $q, h, w, y, u, y$ are illustrated in Fig. 1(a). The vector $\delta(t) = [\delta_1(t) \delta_2(t) \ldots \delta_n(t)]$ collects all scheduling parameters, whose values are confined by a compact set $\delta$. Assume that the LFR is well-posed, i.e., $(I - D_{aa}\Delta)$ is invertible for all $\delta \in \Delta$. We explicitly do not assume $\Delta(\delta(t))$ to have diagonal structure. Furthermore, we consider scheduling parameters possibly nonlinear functions of measurable scheduling signals $\rho$, that range in some compact set $\rho$. These might, for example in robotics, comprise joint angles and their derivatives, whereas the scheduling parameters are functions involving sine and cosine terms. We let the mapping $f^{\rho^\delta} : \mathbb{R}^{n_\rho} \rightarrow \mathbb{R}^{n_\rho}, \rho(t) \mapsto f^{\rho^\delta}(\rho(t)) := \delta(t)$, denote the nonlinear function with which the LFT parameters $\delta(t)$ in $\Delta$ can be computed from the measurable signals $\rho$. State space matrices of the plant are related to the LFR by

$$
S^P(\delta) = \begin{bmatrix} \begin{bmatrix} A & B_p & B_u \\ C_p & D_{pp} & D_{pu} \\ C_y & D_{yp} & D_{yu} \end{bmatrix} \end{bmatrix} \begin{bmatrix} I \\ \end{bmatrix} (I - D_{aa}\Delta)^{-1}[C_a D_{ap} D_{au}],
$$

(4)

2.2 Gain-Scheduled LFT LPV Controller Synthesis

A standard LFT LPV gain-scheduling synthesis result (Scherer, 2000) provides a condition for the existence of a gain-scheduled controller.

Theorem 1. There exists a controller $K^\delta$, such that the closed-loop system $P^\delta \times K^\delta$ is internally stable and achieves an $\mathcal{L}_2$-gain of $\gamma > 0 \forall \delta \in \Delta$, if there exist $X = Y^T > 0$ and $M = M^T, N = N^T$ that satisfy

$$
X \mathcal{L}(X, \Pi, G_{ss}) \begin{bmatrix} \begin{bmatrix} I \\ \end{bmatrix} \end{bmatrix} V_X < 0,
$$

(5)

$$
Y \mathcal{L}(Y, \Pi, G_{ss}) \begin{bmatrix} \begin{bmatrix} I \\ \end{bmatrix} \end{bmatrix} V_Y > 0,
$$

(6)

$$
\begin{bmatrix} \begin{bmatrix} I \\ \end{bmatrix} \end{bmatrix} M \begin{bmatrix} \begin{bmatrix} I \\ \end{bmatrix} \end{bmatrix} N \begin{bmatrix} \begin{bmatrix} I \\ \end{bmatrix} \end{bmatrix} < 0, \quad \forall \delta \in \Delta
$$

(7)

where $V_X = \ker [C_y D_{yy} D_{yu}], V_Y = \ker [B_y^T D_{yy} D_{yu}]$.
\[\alpha(S^K) \leq 2\Delta(n_x + n_y) + \alpha(\Psi) + \ldots \] 
(11) 
\[\ldots + \Delta(n_x + n_u) \in O(n_x^3), \] 
(12) 
with \(\alpha(\Psi) \leq \Delta(n_x^2/n_y^2 - 1) \in O(n_x^3).\) 
(13)

3. MAIN RESULTS

3.1 Low Complexity LFRs

Consider the nonlinear differential equation
\[M(q, t) \dot{q} + k(\dot{q}, q, t) = u.\] 
(14)

No input nonlinearity is assumed for simplicity. This can often be achieved by considering transformed inputs \(u = T(q, q, t)\). Models of many physical systems from different disciplines can be represented in this way by using first principles modeling approaches. Motivated by mechanical structures, we refer to \(q \in \mathbb{R}^{n_q}\) as the (angular or translational) position vector and \(M(q)\) as the inertia matrix. The input is denoted \(u \in \mathbb{R}^{n_u}\). In mechanical models the nonlinear vector \(k(\dot{q}, \dot{q})\) contains stiffness and damping terms, as well as e.g. gyroscopic effects. In the following, dependence on time of the matrices—as already done with the signals—will be dropped for brevity.

It is often possible to rewrite (14) equivalently as
\[M(q)\dot{q} + D(\dot{q}, q) + K(q)q = u.\] 
(15)

For simplicity, we assume that \(n_u = n_q\). Note, that rewriting \(k(\dot{q}, q) = D(\dot{q}, q) + K(q)q\) is not unique. The question of how to choose the matrix-vector products \(D(\dot{q}, q)q\) and \(K(q)q\) is closely related to the non-uniqueness of LPV representations. In essence, the question of which degree-of-freedom is to be pulled into the vector or a matrix entry determines which coupling effects are linearly visible, i.e., if the parameter-dependent matrices are frozen in some operating point. On the other hand, the choice influences the complexity of the LPV parameterization of the matrices used later on and can therefore lead to a trade-off between model/synthesis complexity and achievable control performance.

Suppose that for the inertia, damping, stiffness and input matrices one can find LFRs
\[M(q) = M_T(q) \times \begin{bmatrix} 0 & W_M \\ V_M & M_0 \end{bmatrix}, \quad K(q) = K_T(q) \times \begin{bmatrix} 0 & W_K \\ V_K & K_0 \end{bmatrix}, \quad D_T(\dot{q}, \dot{q}) = D_T(\dot{q}, \dot{q}) \times \begin{bmatrix} 0 & W_D \\ V_D & D_0 \end{bmatrix}.\]

Note the representations are affine and contain constant shifts, such that \(M_0\) is invertible and \(M_T(q), D_T(\dot{q}, \dot{q})\) and \(K_T(q)\) all contain a zero matrix over the set of admissible trajectories. The matrices \(W_M, V_M, W_K, V_K\) and \(W_D, V_D\) can be chosen, such that only the parameter-dependent part of the respective matrices is contained in \(Q_T,\) for all \(Q \in \{M, D, K\}.\) LFRs with diagonal blocks \(Q_T,\) for all \(Q \in \{M, D, K\},\) can be constructed with available standard tools from MATLAB. However, using the physical insight from (15) one can easily construct full blocks, whose dimensions can often turn out smaller than the diagonal ones.

Remark 1. The proofs shown in Scherer (2000) depend on the LFT scheduling block containing the origin. Therefore, in an LPV parameterization of, e.g., \(M_T(q)\), it might be necessary to enhance the admissible LPV parameter range or to even define new parameters, such that this is possible.

Omitting parameter-dependency for brevity, a general LPV state space model with \(q = [q^T \ q^T]^T\) reads as
\[G^p : \begin{bmatrix} \dot{q} \\ \dot{\dot{q}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -A^{-1}K_0 -A^{-1}D -A^{-1}M_{pp} -A^{-1}M_{wq} -A^{-1}M_{wv} -A^{-1}M_{yv} \end{bmatrix} \begin{bmatrix} \dot{q} \\ \dot{\dot{q}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}.\] 
(16)

Now, from simple inversion of an LFT (Zhou et al., 1996) we have
\[-(M_0 + V_M M_T W_M)^{-1} = M_T * \begin{bmatrix} -W_{Mq}^{-1} V_{Mq} & -W_{Mq}^{-1} V_{Mq} \\ -W_{Mq}^{-1}V_{Mq} & -W_{Mq}^{-1} V_{Mq} \end{bmatrix},\] 
(17)
and the respective LFRs
\[\begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} -W_{M0}^{-1} V_{M0} & 0 & 0 \\ 0 & -M_0^{-1} & 0 \end{bmatrix} = M_T * \begin{bmatrix} K_T \ I \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},\] 
(18)
we obtain the physical model representation in structured LFT form (21). In consequence, we may obtain the LPV representation of the generalized plant
\[P^V = \begin{bmatrix} A & B_T & B_p & B_u \\ C_T & D_{Tq} & D_{Tq} & D_{Tu} \\ C_T & D_{Tq} & D_{Tq} & D_{Tu} \\ C_T & D_{Tq} & D_{Tq} & D_{Tu} \end{bmatrix} = \begin{bmatrix} P^{Wq} & P^{Wy} & P^{Wy} \\ P^{Wq} & P^{Wy} & P^{Wy} \\ P^{Wq} & P^{Wy} & P^{Wy} \end{bmatrix},\] 
(19)
where
\[P^V = \Upsilon * P^v, \quad \Upsilon \in \mathbb{R}^{n_T \times n_T^V}.\] 
(20)

Note that the proposed representation maintains generality for the cases if any of the matrices \(M_T, K_T\) or \(D_T\) is parameter-independent by simply considering zero dimensions. We therefore choose to present the general form and leave the special cases to the interested reader and our example. In addition, an identity output gain and parameter-independent performance channel \((D_{Tq} = 0, D_{Tu} = 0)\) are assumed for simplicity. Extensions, however, are straightforward. Consequently, we arrive at an LFR, whose size of the scheduling block \(\Upsilon\) is smaller or equal than \(3n_q \times 3n_q\). If conventional techniques result in a smaller size block, they should be used. In fact, the representation proposed above can also be used to obtain a mixed block-diagonal/diagonal \(\Upsilon,\) e.g. by affinely parameterizing \(K(q)\) and using a diagonal \(K_T.\) This can be useful if the number of affine parameters in \(K_T\) and/or \(D_T\) is exceptionally high. Then for these, a diagonal block in conjunction with \(D/G\)-scalings can avoid an evaluation of the multiplier conditions on the vertices of a convex hull, which might be prohibitive.

Rational and Affine Parameterization: The scheduling block \(\Upsilon\) can be written as an LFT in terms of both parameters \(\delta, \nu,\) which provide a rational or affine...
parameterization of $\Upsilon$ with diagonal blocks $\Delta(\delta)$ and $\hat{\Upsilon}(\nu)$, respectively.

\[
\Upsilon(\delta) = \Delta \ast \begin{bmatrix} W_{11}^h & W_{12}^h \\ W_{21}^w & W_{22}^w \end{bmatrix},
\]

$\Delta = \text{diag}(\delta_i I_{_i})$, \quad $i = 1, 2, \ldots, n_\delta$.

\[
\Upsilon(\nu) = \hat{\Upsilon} \ast W^v,
\]

$\hat{\Upsilon} = \text{diag}(\nu_i I_{_i})$.

This is illustrated in Figs. 1(b) and 1(c).

3.2 Two-Stage Full-Block $\Sigma$-Procedure

The use of full-block multipliers in conjunction with an affine parameterized scheduling block requires to solve multiplier conditions on a possibly large number of vertices. Furthermore, it may be possible to find a rational parameter set with tighter bounds—i.e., with less overbounding (Kwiatkowski and Werner, 2008)—on the admissible trajectories via the map $f^\phi(v)$. Evaluating (6) on the vertices of the convex hull spanned by the admissible parameter range in terms of $\phi$ is resulting in a non-convex region in terms of $\theta$ in general. However, a further application of the full-block $\Sigma$-procedure on (6) introduces secondary multipliers and therefore further decision variables, but in turn allows to evaluate the primary multiplier condition convexly on the tightened parameter set. Note that this does not compromise the small size of the primary multiplier, which decides the size of the controller’s scheduling block.

**Proposition 1.** With the LFT parameterization (22) and Theorem 1 applied to $\mathbb{T}^\nu$ from (20), the conditions

\[
\begin{bmatrix} I \\ \nu \end{bmatrix} M^{\ast} \begin{bmatrix} I \\ \Upsilon \end{bmatrix} > 0, \quad \begin{bmatrix} \nu \end{bmatrix} N^{\ast} \begin{bmatrix} \nu \end{bmatrix} < 0, \quad \forall \nu \in \mathbb{V}
\]

(analogous to (6)) are equivalent to

\[
\begin{bmatrix} \nu \end{bmatrix} R \begin{bmatrix} \nu \end{bmatrix} > 0, \quad \begin{bmatrix} \nu \end{bmatrix} \begin{bmatrix} \nu \end{bmatrix} > 0, \quad \forall \phi \in \delta.
\]

\[
\begin{bmatrix} \nu \end{bmatrix} S \begin{bmatrix} \nu \end{bmatrix} < 0, \quad \forall \phi \in \delta.
\]

**Proof 1.** The proof follows by straightforward application of the full-block $\Sigma$-procedure on (24).

Remark 2. Proposition 1 can be similarly formulated based on the parameterization $\Upsilon = \hat{\Upsilon} \ast W^v$.

No additional synthesis complexity is introduced in the controller construction problem, as the new multipliers $R$ and $S$ are not required for the construction of the extended multiplier $M_3$ and in the LMI-based controller variable construction step.

4. APPLICATION TO A 3-DOF ROBOT

4.1 Modeling

Three degrees-of-freedom of an industrial manipulator of type Thermo CRS A465 are considered including the first, second and third joints as shown in Fig. 2(a). The joint limits are listed in Tab. 3(a).

![Fig. 2. Robot schematics and generalized plant.](image)

Table 2. Signal ranges.

<table>
<thead>
<tr>
<th>Angle</th>
<th>Range $[\text{rad}]$</th>
<th>Angular Velocity</th>
<th>Range $[\text{rad} \text{s}^{-1}]$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$q_1$</td>
<td>$[-180, \ldots, 180]$</td>
<td>$\dot{q}_1$</td>
<td>$[-100, \ldots, 100]$</td>
</tr>
<tr>
<td>$q_2$</td>
<td>$[-90, \ldots, 90]$</td>
<td>$\dot{q}_2$</td>
<td>$[-80, \ldots, 80]$</td>
</tr>
<tr>
<td>$q_3$</td>
<td>$[-45, \ldots, 135]$</td>
<td>$\dot{q}_3$</td>
<td>$[-125, \ldots, 125]$</td>
</tr>
</tbody>
</table>

Table 3. Scheduling signals and parameters.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Value</th>
<th>(b) LFT scheduling parameters.</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho_1$</td>
<td>$q_1$</td>
<td>$\delta_1$ $\sin(q_1)$ $\delta_5$ $\sin(q_2)$</td>
</tr>
<tr>
<td>$\rho_2$</td>
<td>$q_2$</td>
<td>$\delta_2$ $\sin(q_2)$ $\delta_7$ $\rho_3$</td>
</tr>
<tr>
<td>$\rho_3$</td>
<td>$q_3$</td>
<td>$\delta_3$ $\cos(q_1)$ $\delta_8$ $\rho_4$</td>
</tr>
<tr>
<td>$\rho_4$</td>
<td>$q_3$</td>
<td>$\delta_4$ $\cos(q_2)$ $\delta_9$ $\rho_5$</td>
</tr>
<tr>
<td>$\rho_5$</td>
<td>$q_3$</td>
<td>$\delta_5$ $\sin(q_3)$</td>
</tr>
</tbody>
</table>

From the nonlinear differential equations (27) (Hoffmann et al., 2013), an LPV model is derived based on the novel proposed modeling scheme. Scheduling signals $\rho$ are defined in Tab. 3(a) and the parameter sets $\delta$ and $\nu$ are given in Tab. 3(b) and (28)–(30). The non-uniqueness of factoring $D(q, \dot{q})g$ is limited to the first row of $D(q, \dot{q})$. Here, products $q_1, q_2, i = 2, 3$ provide the option of choosing, e.g., the (1,2) and (1,3) entries of $D(q, \dot{q})$ as zero and gather all terms in the (1,1) entry. While this would allow to define only 9 parameters $\nu$, the coupling from...
By calculating the elementwise maxima \( Q(v) \) and minima \( \bar{Q}(v) \) over a grid covering \( \nu \), the matrices \( Q_0 \) and \( Q_{\text{rng}} \) are derived via \( Q_0 = (\bar{Q} + Q)/2 \) and \( Q_{\text{rng}} = \bar{Q} \), for \( Q \in \{ M, D, K \} \), respectively. With \( \Theta_{\text{rng}} = \text{diag}(M_{\text{rng}}, D_{\text{rng}}, K_{\text{rng}}) \), the model is normalized by

\[
\Theta^0 = \Theta_{\text{rng}}^{-1} \Theta = \begin{bmatrix} A & B \Delta \Theta_{\text{rng}} \end{bmatrix} \begin{bmatrix} B_{\Theta} \\ \Delta \Theta_{\text{rng}} \end{bmatrix}.
\]

**Comparison with Previous Modeling Approaches:**

As apparent from (28)–(30), the model’s scheduling block \( \Theta \) is of the size \( 8 \times 8 \). When diagonal affine and rational parameterizations of \( \Theta \) are considered according to (23) and (22), the sizes obtained are \( \bar{\Theta}(v) \in \mathbb{R}^{15 \times 15} \) and \( \Delta(\delta) \in \mathbb{R}^{37 \times 37} \). In comparison, the models derived from parameterizations detailed in Hoffmann et al. (2013) yield diagonal scheduling blocks \( \Theta(\theta) \in \mathbb{R}^{15 \times 15} \) and \( \Delta(\delta) \in \mathbb{R}^{38 \times 38} \), for an affine LPV model and a rational LPV model, respectively. The affine LPV parameters are denoted \( \theta \) and are rational functions of the parameters \( \delta \) defined in Tab. 3(b). In Hoffmann et al. (2013) the block \( \Delta(\delta) \) is derived by substituting the rational functions of \( \theta \) in terms of \( \delta \). Eventually, standard LFT reduction techniques of the Matlab Robust Control Toolbox are applied. This shows the attractiveness of the proposed approach for low scheduling order rational LPV models.

### 4.2 Controller Synthesis

Synthesis is performed in three categories, i.e. using 1) a single multiplier stage with full-block multipliers (FBM), 2) two multiplier stages with FBMs in the first and \( D/G \)-scalings in the second (FBM+D/G) and 3) a single \( D/G \)-multiplier stage. The latter approach would lead to diagonal multipliers for the block-diagonal scheduling block \( \Theta \) and is therefore not presented, since it results in excessive conservatism. The two-stage multiplier approach is performed both with respect to the affine parameterization \( \bar{\Theta}(v) \) and the rational parameterization \( \bar{\Delta}(\delta) \). Consequently, the second multiplier is then constrained with respect to the diagonal blocks \( \bar{\Theta}(v) \) and \( \bar{\Delta}(\delta) \) after normalization, respectively, cf. Fig. 1(c) and 1(b). A fourth category considers the affine LPV model with 16 LPV parameters (denoted by \( \theta \)) as detailed in Hoffmann et al. (2013). For all approaches, synthesis is performed with respect to the generalized plant configuration shown in Fig. 2(b) and the choice of shaping filters:

\[
W_{S1} = \frac{\bar{Q}_{\text{rng}}(M_{\text{rng}}, D_{\text{rng}}, K_{\text{rng}})}{\bar{Q}_{\text{rng}}} \quad W_{K1} = \frac{\bar{Q}_{\text{rng}}(M_{\text{rng}}, D_{\text{rng}}, K_{\text{rng}})}{\bar{Q}_{\text{rng}}} \quad \nu_i = \text{diag}(\nu_i) \quad i = 1, \ldots, 3
\]

The filter parameters are given in Tab. 4.

Tab. 5 shows the results for the respective approaches in terms of the decision variables and required solver time \(^1\) for (i) the existence condition (Theorem 1) and (ii) the LMI-based controller construction (Scherer, 2000). The root mean square tracking error (RMSE) with respect to each joint space is reported as well. Intractability is bested only when using the additional information overbounding. Interestingly, the two-stage approach 2) with \( \bar{\Theta}(v) \) shows the best performance, while also being solved very efficiently in only about half a minute. Using \( D/G \)-scalings directly on \( \bar{\Delta}(\delta) \) indicates a worse induced \( L_2 \)-gain \( \gamma \) but, expectedly, similar performance.

---

\(^1\) Intel® Core™ i7-2660 3.4 GHz, 8 GB RAM, 64-Bit Windows 7

---
Table 5. Synthesis results, complexity and tracking performance in simulation.

<table>
<thead>
<tr>
<th>Method</th>
<th>LMI Vars.</th>
<th>CPU [s]</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBM</td>
<td>(\Upsilon(v))</td>
<td>7.98 735 (1.122)</td>
<td>5.42 3 (15.5)</td>
</tr>
<tr>
<td>(\Theta(\delta))</td>
<td>6.013 (3.906)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FBM+</td>
<td>(\Upsilon(v))</td>
<td>4.26 763 (1.122)</td>
<td>6.8 (12.9)</td>
</tr>
<tr>
<td>D/G</td>
<td>(\Upsilon(\delta))</td>
<td>4.86 1,217 (1.122)</td>
<td>16.0 (14.7)</td>
</tr>
<tr>
<td>3) (D/G)</td>
<td>(\Upsilon(v))</td>
<td>7.63 945 (3.906)</td>
<td>15.2 (436.3)</td>
</tr>
<tr>
<td>4) (D/G)</td>
<td>(\Theta(\delta))</td>
<td>7.38 495 (1.466)</td>
<td>7.8 (46.6)</td>
</tr>
</tbody>
</table>

Table 6. Experimental performance and implementation complexity.

<table>
<thead>
<tr>
<th>Method</th>
<th>No. of Arith. Ops.</th>
<th>Sched. Matrices Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>FBM</td>
<td>(\Upsilon(v))</td>
<td>0.052 0.038 0.063</td>
</tr>
<tr>
<td>(\Theta(\delta))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FBM+</td>
<td>(\Upsilon(v))</td>
<td>0.063 0.060 0.097</td>
</tr>
<tr>
<td>D/G</td>
<td>(\Upsilon(\delta))</td>
<td>0.050 0.054 0.047</td>
</tr>
<tr>
<td>3) (D/G)</td>
<td>(\Upsilon(v))</td>
<td>0.049 0.061 0.138</td>
</tr>
<tr>
<td>(\Theta(\delta))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4) (D/G)</td>
<td>(\Theta(\delta))</td>
<td>0.048 0.059 0.119</td>
</tr>
</tbody>
</table>

The online implementation complexity is governed by both the size of scheduling blocks and whether or not \(D/G\)-scalings allow the controller to be scheduled by a mere copy of it. The additional effort to compute \(\Upsilon\) scalings prevent exponential growth in the LMI conditions, while maintaining a low implementation complexity. When applied to the nonlinear model of a 3-DOF robotic manipulator, full-block multipliers in the first stage improves performance while maintaining the lowest estimated implementation complexity of the compared controllers. The results are illustrated by real-time experiments.

REFERENCES


