INTERNATIONAL JOURNAL OF ROBUST AND NONLINEAR CONTROL
Int. J. Robust. Nonlinear Control 2016; **00**:1–19
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/rnc

LPV Modelling with Affine Dependence: A Local $\mathcal{H}_\infty$ Approach

S. Taamallah[1*], X. Bombois[2], R. Tóth[3], Paul M.J. Van den Hof[3]

[1]*National Aerospace Laboratory (NLR), Anthony Fokkerweg 2, 1059 CM, Amsterdam, The Netherlands*
[2]*Laboratoire Ampère UMR CNRS 5005, Ecole Centrale de Lyon, 36 avenue Guy de Collongue, 69134 Ecully Cedex, France*
[3]*Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*

SUMMARY

We present a Linear Parameter-Varying (LPV) modeling approach that delivers a state-space model with affine dependency, suitable for controller design, from an already existing Non-Linear (NL) model of the plant. Specifically, the method is based on the following steps: 1) local linearization of the NL system model in order to obtain a set of local Linear Time-Invariant (LTI) models in state-space form and affine remainder terms; 2) assuming that a data set capturing the transient behavior of the system between these local linearization points is also present, Singular Value Decomposition (SVD) of this data is computed to derive linear maps characterizing the coefficient dependence of the to be obtained LPV model; 3) synthesis of the scheduling variations of the model w.r.t. the obtained linear maps by minimizing the $\mathcal{H}_\infty$ distance between the local linearizations and the corresponding frozen scheduling based aspects of the LPV model together with the ℓ_2 approximation of change of the affine remainder terms; and 4) use of a Neural Networks (NN) based approach to relate the synthesized scheduling variables to on-line measurable signals in the system. Although our focus is primarily set upon obtaining LPV models useful for control synthesis, we provide extensive analysis of, both, open- and closed-loop simulation results to illustrate the practicality of the method.

Received . . .

KEY WORDS: LPV modeling, $\mathcal{H}_\infty$-based approach, LPV control, robust control

1. INTRODUCTION

The *Linear Parameter-Varying* (LPV) framework has become popular recently as it represents an attractive intermediate system class between *Linear Time-Invariant* (LTI), and *Non-Linear* (NL) or *Time-Varying* (TV) structures [1, 2]. In particular, LPV systems allow to embed NL behaviors into the solution set of linear representations (see, [1, 2]) for which powerful LPV control methods can be applied. These methods can be seen as an extension of the standard $\mathcal{H}_2$ and $\mathcal{H}_\infty$ LTI synthesis techniques (e.g., [3–6]). The LPV approach amends also the main drawbacks of classical *Gain Scheduling* (GS) [7]: i) by eliminating the need for repeated designs/simulations in order to handle the global control problem; and ii) by guaranteeing both stability and performance along all possible parameter variations, i.e., scheduling trajectories. In addition, LPV control design problems are efficiently solved, by first expressing the problems as *Linear Matrix Inequalities* (LMIs) based optimization problems, subsequently formulated as *Semi-Definite Programs* (SDPs). This has resulted in a growing number of applications of the LPV framework in aerospace [8–10], wind turbines [11], wafer steppers [12, 13] and CD players [14] to name a few.

*Correspondence to: National Aerospace Laboratory (NLR), Anthony Fokkerweg 2, 1059 CM, Amsterdam, The Netherlands. Email: staamall@nlr.nl

Besides of all of these attractive features, the LPV control framework typically takes the existence of an accurate, but low complexity LPV model of the plant, as a starting point. Despite of many available LPV modeling methods, transformation of a given NL system model into a suitable (quasi-)LPV[†] model is a difficult and non-systematic exercise [2]. There exists two main modeling avenues to transform or approximate a NL representation into an LPV one, namely the so-called *local* and *global* approaches [2, 15]. The local approach consists of applying linearization theories on the NL model to obtain local LTI models of the plant behavior, and subsequently interpolating these models to derive an LPV approximation. Within this framework several methods have been developed, based upon, e.g., extended linearization [16], Jacobian linearization [17], multiple-model design procedure [18], $\mathcal{H}_2/\mathcal{H}_\infty$ norm minimization [19, 20] and multivariable polynomial fitting [21]. On the other hand, the global approaches generate LPV models which preserve the dynamic behavior of the NL system. This can either be done by using a range of mathematical manipulations, e.g.: state transformation [22], velocity-based formulation [23], function substitution [24], normal form based conversion [25] and automated LPV model generation [2, 26], or alternatively by using data-driven modeling, i.e., global identification approaches, like *prediction error methods* [27].

Often it is important that the global behavior of the LPV model be similar to the global behavior of the NL system. This is typically the case when the LPV model is used for prediction/simulation in open-loop [28], model predictive control or optimal control. On the other hand, it is sometimes desirable that the local (frozen) behavior of the LPV model, i.e., for constant scheduling, be representative of the local behavior of the NL system, i.e., local linearizations of the NL system. For such cases, a local approach would be recommended[‡]. This is particularly the case when the LPV model is used for gain scheduled controller design, where controllers are synthesized on the basis of local models. In this paper, and inspired by the work done in [18], we present a novel local modeling approach, which approximates a known NL model by a quasi-LPV one with affine dependence, resulting in a model valid for both open- and closed-loop applications.

Like in [18], our method uses a set of simulation data obtained from the original NL model. This set of data is assumed to sufficiently capture the transient behavior of the NL system between the operating points. Like in [18], local linearization along the trajectory described by the simulation data leads to a set of local LTI models in *State-Space* (SS) form together with affine remainder terms. A singular value decomposition of a matrix containing the coefficients of these linearized models is here also used to determine the basis functions that characterize the coefficient dependence of the to-be-obtained LPV model. However, for the subsequent LPV modeling steps, our method is different from the one in [18]. The scheduling dependencies are indeed not determined in such a way that the LPV model is best able to reproduce the time-domain simulation data, but in such a way to minimize the H_∞ distance between the local linearizations and the transfer functions obtained when freezing the scheduling variable in the LPV model. This approach is preferred since our objective is here to approximate as good as possible the local behavior of the system for control purpose. Another important difference with [18] is that the remainder terms in the linearized models are also modeled using the LPV setting leading to a true LPV representation of the original NL model. Unlike the one in [18], this true LPV representation supports LPV control design over the complete operating regime. Our approach can also be seen as an extension of Jacobian linearization based gain-scheduling [17], since, even if our modeling method is based upon interpolation of local linearizations, it does not rely upon local deviation signals.

The paper is organized as follows. In Section 2, the general LPV modeling setting and the to-be-addressed optimization problems are defined. In Section 3 through 6, the steps of the proposed modeling approach are described and solutions to the optimization problems are derived. In Section 7, open- and closed-loop simulation results are analyzed using a nominal $\mathcal{H}_\infty$, a robust μ, and two LPV controllers. Finally, conclusions and future directions are presented in Section 8.

[†]The *quasi-* prefix is used to indicate LPV system models in which the scheduling variables are chosen to be endogenous, i.e., dependent of the inputs, outputs or additional latent variables of the modeled physical system [1].

[‡]Note that global embedding of the behavior of a nonlinear system into an LPV representation often does not imply that the frozen aspects of the LPV models will have anything in common with the local linearizations of the NL system [29].

DOI: 10.1002/rnc

The nomenclature is fairly standard. $M^\top$, M^*, $M^\dagger$ denote the transpose, the complex-conjugate transpose, and the Moore-Penrose inverse of a real or complex matrix M, whereas $\mathrm{He}(M) = M + M^*$. We use $\star$ as an ellipsis for terms that are induced by symmetry and $\mathrm{vec}(\cdot)$ stands for the vectorization of a matrix. Matrix inequalities are considered in the sense of *Löwner*. Furthermore, $\lambda(M)$ denotes the zeros of the characteristic polynomial $\det(sI - M) = 0$. $\mathcal{L}_\infty$ is the *Banach* space of complex matrix-valued functions that are essentially bounded on $j\mathbb{R}$ with norm $\|G\|_\infty := \operatorname{ess\,sup}_{\omega\in\mathbb{R}} \bar{\sigma}(G(j\omega)) < \infty$ with $\bar{\sigma}$ denoting the largest singular value of a matrix. Similarly, $\mathcal{H}_\infty$ is a closed subspace of $\mathcal{L}_\infty$ with functions that are analytic on right-half plane $\mathbb{C}^+$. $\mathcal{RL}_\infty$ (resp. $\mathcal{RH}_\infty$) represent the subspace of real, rational and proper *Transfer Functions* (TFs) in $\mathcal{L}_\infty$ (resp. $\mathcal{H}_\infty$). For a $\mathcal{W} \subset \mathbb{R}_0^+$, we use $\|G\|_\infty^{\mathcal{W}} := \operatorname{ess\,sup}_{\omega\in\mathcal{W}} \bar{\sigma}(G(j\omega))$. For appropriately dimensioned matrices K and M, where the latter is partitioned as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},$$

the lower *Linear Fractional Transformation* (LFT) is defined as $F_\mathrm{l}(M, K) = M_{11} + M_{12}K(I - M_{22}K)^{-1}M_{21}$, and the upper LFT is defined as $F_\mathrm{u}(M, K) = M_{22} + M_{21}K(I - M_{11}K)^{-1}M_{12}$ under the assumption that the inverses exist. For $M \in \mathbb{C}^{q\times p}$, the structured singular value $\mu_\Delta(M)$ of M, with respect to an uncertainty set $\mathbf{\Delta} \subset \mathbb{C}^{p\times q}$, is defined as $\mu_\Delta^{-1}(M) := \min_{\Delta\in\mathbf{\Delta}}\{\bar{\sigma}(\Delta) \mid \det(I - M\Delta) = 0\}$.

2. PROBLEM STATEMENT

We suppose that the system which is to be modeled in the LPV setting can be described by a known, *Continuous-Time* (CT), NL dynamical model in the following state-space form

$$\dot{x}(t) = f(x(t), u(t)), \qquad y(t) = g(x(t), u(t)), \tag{1}$$

$\forall t \geq 0$ with f, g being partially differentiable smooth real functions, $x(t) \in \mathcal{P}_\mathrm{x} \subset \mathbb{R}^{n_\mathrm{x}}$ the plant state, $y(t) \in \mathcal{P}_\mathrm{y} \subset \mathbb{R}^{n_\mathrm{y}}$ the plant output, $u(t) \in \mathcal{P}_\mathrm{u} \subset \mathbb{R}^{n_\mathrm{u}}$ the control input, while t is the time variable and $\mathcal{P}_\mathrm{x}, \mathcal{P}_\mathrm{y}, \mathcal{P}_\mathrm{u}$ are some non-empty compact sets. In this simulation model, the simulated data is not perturbed by noise. Furthermore, we assume that the simulation model (1) perfectly describes the behavior of the NL system. However, as mentioned earlier, this model is deemed to be too complex for control design. Hence, our goal is to approximate (1) by a quasi-LPV representation, suitable for μ or LPV control design. To simplify the discussion, we consider in this paper the approximation of the state-equation only, however the procedure can be extended to approximate the output equation, i.e., g as well. Hence, from now on, we consider the case

$$\dot{x}(t) = f(x(t), u(t)), \qquad y(t) = x(t). \tag{2}$$

Our procedure uses simulation data to synthesize a quasi-LPV model of the NL model characterized by $f(\cdot)$. For this purpose, we apply to the simulation model (2) a typical input signal and store equidistant samples of the corresponding response with the sampling period $T_\mathrm{s} > 0$. This yields the following sampled *Input-Output* (IO) sequence $\mathcal{Z}^N := \{u_i, x_i\}_{i=1}^N$ where $x_i = x(iT_\mathrm{s})$ and $u_i = u(iT_\mathrm{s})$. We also assume that this sequence is informative enough for the synthesis of the quasi-LPV model, i.e., the mapping f has been excited over the entire operating regime $\mathcal{P}_\mathrm{x} \times \mathcal{P}_\mathrm{u}$. Note that according to the classical results of system theory, if the free CT signals (i.e., inputs) can be assumed to be piecewise constant on a sampling period, i.e., T_s is sufficiently small, then the CT output trajectory may be completely reconstructed from its sampled observations.

Remark 1

We will encompass our discussion within the CT framework, since stability and performance requirements for controller synthesis are generally conveniently expressed in this setting. If a *Discrete-Time* (DT) LPV approximation is needed, then it can be achieved by either discretizing the obtained CT LPV model through one of the methods presented in [30], or alternatively, by using the equivalent DT formulation of the machinery outlined in this paper.

We intend to capture (2) in terms of the LPV-SS model

$$\dot{x}(t) = A_0 x(t) + B_0 u(t) + \sum_{r=1}^{n_r} p_r(t)\Big(A_r x(t) + B_r u(t)\Big), \tag{3}$$

where the scalar signals $p_r(t) = \phi_r(x(t), u(t))$ are the so called *scheduling variables* that are on-line measurable via the *scheduling map* $\phi = [\ \phi_1 \quad \cdots \quad \phi_{n_r}\ ]^\top : \mathcal{P}_x \times \mathcal{P}_u \to \mathcal{P}_\phi$ where $\mathcal{P}_\phi$, known as the *scheduling "space"*, is a compact set and $\{A_r, B_r\}_{r=0}^{n_r}$ are matrices of appropriate sizes. Furthermore, we also choose to enclose our analysis within the affine LPV setting with static scheduling-parameter dependence, as dynamic dependence may lead to difficulties in terms of controller design and implementation.

Non-stationary linearizations of the NL model, along a given trajectory, as suggested for the GS modeling framework (see, e.g., [23]) have often been used to extend the validity of GS controllers to operating regions far from equilibrium points. When combined with a sufficiently small sampling period T_s, such an approach may allow to better capture the transient behavior of the NL model. Accordingly, we also choose to base our LPV modeling methodology upon such linearizations. The latter may be computed via first-order Taylor-series expansions, or via classical numerical perturbation methods. By using the data points $\mathcal{Z}^N := \{u_i, x_i\}_{i=1}^N$ as the linearization points, we can compute the set $\mathcal{D}^N := \{\bar{A}_i, \bar{B}_i, \bar{d}_i\}_{i=1}^N$ where

$$\begin{aligned} \bar{A}_i &= \left.\frac{\partial f(x,u)}{\partial x}\right|_{(x_i,u_i)} \qquad \bar{B}_i = \left.\frac{\partial f(x,u)}{\partial u}\right|_{(x_i,u_i)} \\ \bar{d}_i &= f(x_i, u_i) - \bar{A}_i x_i - \bar{B}_i u_i \end{aligned} \tag{4}$$

with $\bar{d}_i$ the so-called affine remainder term. For each $i \in \{1, \ldots, N\}$, let us also define the corresponding local transfer function $\bar{G}_i(s) = (Is - \bar{A}_i)^{-1}\bar{B}_i$. Now, for each operating point (x_i, u_i), we can approximate the NL model (2), in a local neighborhood of (x_i, u_i), as

$$\dot{x}(t) = f(x(t), u(t)) \approx \bar{A}_i x(t) + \bar{B}_i u(t) + \bar{d}_i. \tag{5}$$

Note that the two sets $\mathcal{Z}^N$ and $\mathcal{D}^N$ describe the behavior of the NL system (2) from a global and local perspective, respectively, and will be used to synthesize an LPV approximation of (2) as outlined in Section 1.

It is important to point out that regarding this approximation, simultaneous optimization of the linear maps $\{A_r, B_r\}_{r=0}^{n_r}$, and the possibly nonlinear scheduling maps $\phi(x(t), u(t))$, in Eq. (3), is a non-trivial, non-unique problem as it contains excessive degrees of freedom, giving rise to an ill-conditioned optimization problem. Previous attempts towards such simultaneous approximation problems have used nonlinear optimization methods [31, 32]. Another approach to mitigate the excess of degrees of freedom in such ill-conditioned optimization problems requires the inclusion of additional constraints or regularization [28]. Alternatively, one can divide the problem to separate optimization problems, e.g., synthesize first the linear maps followed by a synthesis of the scheduling maps. We opt here for such a philosophy, i.e., by following an extension of the three-step methodology introduced in [18]. Our method is further summarized in terms of Algorithm 1. Our contribution w.r.t. [18] lies in formulating these steps by absorbing the affine remainder terms $\bar{d}_i$

Algorithm 1 Our LPV model conversion

1: Compute the best LTI approximation of (2) in terms of the $\mathcal{H}_\infty$ distance of the central model (A_0, B_0) w.r.t. $\mathcal{D}^N$.
2: Based on (A_0, B_0) and $\mathcal{D}^N$, synthesize the linear maps $\{A_r, B_r\}_{r=1}^{n_r}$ by a rank, i.e., n_r revealing transformation.
3: Synthesize the scheduling variables $\{p_r(t)\}_{r=0}^{n_r}$ using $\{A_r, B_r\}_{r=0}^{n_r}$ and $\mathcal{D}^N$.
4: Based on $\mathcal{Z}_N$, synthesize the scheduling maps $p_r(t) = \phi_r(x(t), u(t))$ for $r = 0, \ldots, n_r$.

into a coherent LPV state-space representation that is valid also at off-equilibria points, and further

DOI: 10.1002/rnc

by having the local behavior of the NL model at the linearization points being approximated in an $\mathcal{H}_\infty$ optimal sense, which is in line with the commonly used performance objectives of LPV control design (see Section 1). In the sequel we discuss, in more detail, the four-step methodology outlined above.

3. STEP 1: THE CENTRAL MODEL

The central model (A_0, B_0) is chosen within all models present in the set $\{\bar{A}_i, \bar{B}_i\}_{i=1}^N$. A natural approach is to find the model which may be defined as the most *central* one, within the models spanning the set of local behaviors over the entire operating regime. In this paper, we will base this model selection within the $\mathcal{H}_\infty$ framework, since our primary focus is on modeling for control. In addition, for controller synthesis, design specifications are typically generated on various compact frequency ranges of interest, like $\mathcal{W} = [\omega_1, \omega_2]$, with $0 \leq \omega_1 < \omega_2$, which led us to use the $\mathcal{H}_\infty$ norm on a frequency range of relevance: $\|G\|_\infty^{\mathcal{W}}$ (see Appendix A for further details on this frequency restricted norm).

The central model (A_0, B_0), i.e., $G_0(s) = (sI - A_0)^{-1}B_0$, defined on its appropriate region of convergence, is chosen as follows: based on $\mathcal{D}^N$, compute for each $\bar{G}_i(s) = (sI - \bar{A}_i)^{-1}\bar{B}_i$, the following mean m_i and standard-deviation h_i of the $\mathcal{H}_\infty$ distance measure

$$m_i = \frac{1}{N}\sum_{j=1}^{N} \|\bar{G}_i(s) - \bar{G}_j(s)\|_\infty^{\mathcal{W}}, \tag{6a}$$

$$h_i = \sqrt{\frac{1}{N}\sum_{j=1}^{N} \left(\|\bar{G}_i(s) - \bar{G}_j(s)\|_\infty^{\mathcal{W}} - m_i\right)^2}, \tag{6b}$$

where $\|\cdot\|_\infty^{\mathcal{W}}$ for example can be obtained by minimizing the bound γ subject to the *Linear Matrix Inequality* (LMI) (27) (see Appendix A). Let $\bar{m}$ and $\bar{h}$ be the maximum and $\underline{m}$ and $\underline{h}$ be the minimum of $\{m_i\}_{i=1}^N$ and $\{h_i\}_{i=1}^N$. Then, G_0 is chosen as $G_{\hat{k}}$ where $\hat{k}$ is computed as

$$\hat{k} = \arg\min_{k\in\{1,\ldots,N\}} \left(\rho\left([m_k - \underline{m}]/[\bar{m} - \underline{m}]\right)^2 + \left([h_k - \underline{h}]/[\bar{h} - \underline{h}]\right)^2\right) \tag{7}$$

with $\rho \geq 0$ a user-defined weighting parameter.

4. STEP 2: CALCULATION OF THE LINEAR MAPS

Whereas the role of the central model (A_0, B_0) is to capture the most significant linear behavior of the NL system, the linear maps $(A_r, B_r)_{r=1}^{n_r}$ (together with the scheduling maps) are responsible for capturing the NL behavior of the system. As a philosophical starting point of a procedure to synthesize these maps, we can recall that the NL system model can be approximated, in a local neighborhood of (x_i, u_i), in terms of (5). Hence, the gap between the NL behavior and the central model behavior may be characterized, in a local neighborhood of (x_i, u_i), as follows

$$\delta\dot{x}(t) \approx (\bar{A}_i - A_0)x(t) + (\bar{B}_i - B_0)u(t) + \bar{d}_i. \tag{8}$$

Now from Eq. (8), we intend to find the following factorizations

DOI: 10.1002/rnc

$$\bar{A}_i - A_0 = \sum_{j=1}^{n_g} \eta_j(i) L_j, \tag{9a}$$

$$\bar{B}_i - B_0 = \sum_{j=1}^{n_g} \eta_j(i) R_j, \tag{9b}$$

$$\bar{d}_i = \sum_{j=1}^{n_d} \zeta_j(i)(T_j x_i + Z_j u_i), \tag{9c}$$

where $n_g > 0$ and $n_d > 0$ are minimal and (L_j, R_j) with (T_j, Z_j) can be seen as the common normalized basis of the involved linear operators and $\eta_j(i)$ with $\zeta_j(i)$ are the linear combination weights of the basis associated with each operating point (x_i, u_i) in $\mathcal{Z}^N$. This provides that the matrices $\{A_r, B_r\}_{r=1}^{n_r}$ with $n_r = n_g + n_d$ according to (3) are realized as

$$\left[\begin{array}{c:c} A_r & B_r \end{array}\right] = \begin{cases} \left[\begin{array}{cc} L_r & R_r \end{array}\right] & \text{if } r \le n_g; \\ \left[\begin{array}{cc} T_{r-n_g} & Z_{r-n_g} \end{array}\right] & \text{if } r > n_g. \end{cases}$$

One possible approach to accomplish such a factorization follows through a *Singular Value Decomposition* (SVD). Let

$$\Phi = \left[\begin{array}{ccc} \mathrm{vec}(\bar{A}_1 - A_0) & \cdots & \mathrm{vec}(\bar{A}_N - A_0) \\ \mathrm{vec}(\bar{B}_1 - B_0) & \cdots & \mathrm{vec}(\bar{B}_N - B_0) \end{array}\right],$$

where $\otimes$ denotes the Kronecker product. Similarly, we can develop

$$\Lambda_i = \bar{d}_i \left[\begin{array}{c} x_i \\ u_i \end{array}\right]^{\dagger}, \quad \Psi = \left[\begin{array}{ccc} \mathrm{vec}(\Lambda_1) & \cdots & \mathrm{vec}(\Lambda_N) \end{array}\right].$$

Next, we can obtain a proper orthogonal decomposition of Φ and Ψ which gives the principal directions in the space of the coefficients of $\{L_r, R_r\}_{r=1}^{n_r}$ and $\{T_r, Z_r\}_{r=1}^{n_r}$. This is done by obtaining the following economical SVD decompositions

$$\Phi = U_1 \Sigma_1 V_1^{\top}, \quad \Psi = U_2 \Sigma_2 V_2^{\top}, \tag{10}$$

where Σ_1 and Σ_2 are full rank, square, diagonal matrices with dimension $n_{s,1}, n_{s,2} \le \min(n_x(n_x + n_u), N)$. Then, each basis pair (L_j, R_j) and (T_j, Z_j) are simply recovered from the matricization[§] of each column of U_1 and U_2. This results in $n_g = n_{s,1}$ and $n_d = n_{s,2}$, i.e., $n_r = n_{s,1} + n_{s,2}$. In practice, it is often desired to capture the nonlinear behavior with a low number of scheduling variables as the complexity and often the conservativeness of the LPV control design procedure is directly characterized by n_r. Hence, the relative magnitude of the singular values in Σ can be used to find a reduced factorization by taking only the $n_r < n_{s,1} + n_{s,2}$ most significant singular values associated with the columns of U_1 and U_2 in the matricization. Furthermore, it is also possible to compute the factorizations (9) jointly with the use of a single SVD. However, here for reasons to be revealed in the next section, we chose to keep the factorizations separate.

5. STEP 3: SYNTHESIS OF THE SCHEDULING VARIABLES

As a first step in our synthesis, we will determine the set of linear combination weights $\eta(i) = \left[\begin{array}{ccc} \eta_1(i) & \cdots & \eta_{n_g}(i) \end{array}\right]^{\top}$ and $\zeta(i) = \left[\begin{array}{ccc} \zeta_1(i) & \cdots & \zeta_{n_d}(i) \end{array}\right]^{\top}$ in (9). If $n_r = n_{s,1} + n_{s,2}$, one can easily find weights [18] that ensure a perfect equality in (9). If $n_r < n_{s,1} + n_{s,2}$, this is no longer the case and an approximation is required. Since our focus is mainly on modeling for control, we choose to

[§]The operation that turns a vector into a matrix.

DOI: 10.1002/rnc

approximate the local behavior of the NL system (2) by matching it with the frozen aspects of the LPV model (i.e., the LTI model obtained when freezing the scheduling variable in the LPV model).

Partly, this can be seen as approximating the LTI transfer functions $\bar{G}_i(s) = (sI - \bar{A}_i)^{-1}\bar{B}_i$ with the transfer function $G_i(s)$ of the frozen-time LPV models:

$$G_i(s) = \left(sI - A_0 - \sum_{j=1}^{n_g} \eta_j(i) L_j\right)^{-1} \left(B_0 + \sum_{j=1}^{n_g} \eta_j(i) R_j\right).$$

for $i = 1, \ldots, N$. Additionally, due to the affine terms, we also need to guarantee that

$$\bar{d}_i \approx d_i := \sum_{j=1}^{n_d} \zeta_j(i)(T_j x_i + Z_j u_i).$$

These objectives can be formulated as optimization problems as follows: for a given user defined frequency range $\mathcal{W} = [\omega_1, \omega_2]$, find, for each i, the optimal parameters $\{\hat{\eta}(i)\}_{i=1}^N$ and $\{\hat{\zeta}(i)\}_{i=1}^N$ that minimize

$$J_{g,i}(\eta(i)) := \|\bar{G}_i(s) - G_i(s)\|_\infty^{\mathcal{W}}, \tag{11a}$$

$$J_{d,i}(\zeta(i)) := \|\bar{d}_i - d_i\|_2. \tag{11b}$$

While minimizing (11b) corresponds to a least-squares problem with an analytic solution, minimizing (11a) is equivalent to minimizing a scalar variable, subject to the LMI (27) or (28). The (A, B, C, D) matrices in these equations are

$$\bar{G}_i(s) - G_i(s) = \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right] = \left[\begin{array}{cc|c} \bar{A}_i & 0 & \bar{B}_i \\ 0 & A_0 + \sum_{j=1}^{n_g} \eta_j(i) L_r & B_0 + \sum_{j=1}^{n_g} \eta_j(i) R_j \\ \hline I & -I & 0 \end{array}\right] \tag{12}$$

using the standard notation in robust control. In case the underlying system description is parametrized as (9), these LMIs contain cross products of the decision variables with $\eta(i)$ resulting in a bilinear relationship. In such situations, the projection lemma has often been used to provide convex reformulation of the original problem. In our case, unfortunately, a straightforward application of the projection lemma is not achievable, due to the structured nature of our problem (see [33] for additional details). Hence, we choose to use an iterative approach to solve (11a). The procedure has a two-stage modus operandi: an initialization stage, followed by a repeated refinement stage. The first stage computes a reasonable guess value for $\hat{\eta}(i)$. The idea used here is to approximate the maximum gain of the LTI matrices, $\bar{A}_i$ and $\bar{B}_i$, in the following way

$$\hat{\eta}(i) = \arg\min_{\eta} \|X_{A,i}(\eta)\|_2 + \|X_{B,i}(\eta)\|_2, \tag{13}$$

where $X_{A,i}(\eta) = \bar{A}_i - (A_0 + \sum_{j=1}^{n_g} \eta_j L_j)$ and $X_{B,i}(\eta) = \bar{B}_i - (B_0 + \sum_{j=1}^{n_g} \eta_j R_j)$. This is readily recast into the sum minimization of the $\mathcal{L}_2$-induced gains of two static operators

$$\begin{array}{ll} \underset{\eta(i), \gamma_{A,i}, \gamma_{B,i}}{\text{minimize}} & \gamma_{A,i} + \gamma_{B,i} \\ \text{subject to} & \gamma_{A,i} > 0, \quad \gamma_{B,i} > 0 \\ & \left[\begin{array}{cc} \gamma_{A,i} I & \star \\ X_{A,i}(\eta(i)) & I \end{array}\right] > 0 \\ & \left[\begin{array}{cc} \gamma_{B,i} I & \star \\ X_{B,i}(\eta(i)) & I \end{array}\right] > 0 \end{array} \tag{14}$$

Next, the second stage uses the initial values found in (14) in order to solve Eq. (11a) through an iterative approach. Here $\|\cdot\|_\infty^{\mathcal{W}}$ is computed via (28) since the latter is convex in either the (F, K) or (A, B) matrices, resulting in a DK-iteration like optimization scheme.

DOI: 10.1002/rnc

Finally, we can realize that as (x_i, u_i) corresponds to the time moment t_i in $\mathcal{Z}_N$, hence the scheduling trajectory of $p(t_i) = p(iT_s) = [\ \eta^\top(i) \quad \zeta^\top(i)\ ]^\top$ would lead to the required scheduling to match the change of the local behavior of the NL system along the observed data set $\mathcal{Z}_N$. Yet, we require a final step to generalize this observed relation of $p(t)$ for arbitrary trajectories of $(x(t), u(t))$ which is discussed in the next section in terms of a synthesis of a scheduling map ϕ.

Remark 2
DK iteration appears to work well in practice, such as in model order reduction [34] and LPV controller synthesis with parameter-dependent scalings [35]. Convergence of these methods towards a global optimum or even a local one is not guaranteed [36], however, in practice, convergence is commonly achieved within a few iterations.

Remark 3
In (11b), we have based the approximation of the affine remainder terms on the ℓ_2 norm as it is computationally very cheap. An alternative approach would be to consider the ℓ_∞ norm to be consistent with the $\mathcal{H}_\infty$ objective.

Remark 4
In case a common factorization of the matrices and the affine remainder terms is chosen, (11) becomes a joint optimization problem where someone needs to specify the trade-off between the approximation of the local transfer functions versus fitting the variation of the remainder terms. This is why a separate factorization in (9) is chosen to avoid ad-hoc choices in such a trade off. However, the price to be paid for such a separation of the approximation problems is a possibly larger number of scheduling variables.

6. STEP 4: SYNTHESIS OF THE SCHEDULING MAPS

Finally, with the previously synthesized scheduling variation w.r.t. the observed data set $\mathcal{Z}_N$, the aim is here to find a suitable representation of a smooth continuous-time mapping $\phi(\cdot)$, that satisfies $[\ \eta^\top(i) \quad \zeta^\top(i)\ ]^\top \approx \phi(x_i, u_i)$ for all $i \in \{1, \ldots, N\}$ and hence will give $p(t) = \phi(x(t), u(t))$ for arbitrary trajectories of $x(t)$ and $u(t)$.

Now, for physically-intuitive plants, one may select the required states and inputs in $\phi(x(t), u(t))$, based upon engineering judgment, and derive these mappings through popular curve-fitting methods. However, in case such insight is missing, one may consider formal/systematic tools such as: orthogonal basis functions, principal component analysis, statistical analysis, fuzzy tools, support vector machines, or *Neural Networks* (NN). Regarding NN, it is well-known that, under mild assumptions on continuity and boundedness, a network of two layers, the first being hidden sigmoid and the second linear, can be trained to approximate any functional relationship arbitrarily well, provided there are enough neurons in the hidden layer [37]. Hence, NN have found a wide range of applications in control theory. We also choose here to base the $\phi(\cdot)$ modeling on a two-layer feedforward NN, the first being sigmoid and the second linear, with l neurons (l large enough), such that

$$p(t) = \phi(x(t), u(t)) = V\xi(t), \tag{15}$$

with

$$\xi(t) = \kappa(W_{\mathrm{x}}x(t) + W_{\mathrm{u}}u(t) + W_{\mathrm{b}}), \tag{16}$$

and $V \in \mathbb{R}^{n_{\mathrm{r}} \times l}$, $W_{\mathrm{x}} \in \mathbb{R}^{l \times n_{\mathrm{x}}}$, $W_{\mathrm{u}} \in \mathbb{R}^{l \times n_{\mathrm{u}}}$ contain the output and hidden layer weights, while $W_{\mathrm{b}} \in \mathbb{R}^l$ is the bias in the hidden layer, and $\kappa(\cdot)$ is the activation function, taken as a continuous, diagonal, differentiable, and bounded static sigmoid nonlinearity. Hence, we consider NN models in the classical feedforward network with the hyperbolic tangent activation transfer function in the hidden layer and back-propagation training for the weights and biases. The training of these models is accomplished here via the NN MATLAB toolbox by using $\{(x_i, u_i)\}_{i=1}^N$ as the input data set and $\{\eta(i), \zeta(i)\}_{i=1}^N$ as the intended response.

DOI: 10.1002/rnc

Table I. Results of the SVD decomposition in Section 4.

	$n_g = 3$	$n_g = 2$	$n_g = 1$	$n_d = 2$	$n_d = 1$
Captured Energy of U_1	100 %	79 %	51 %	-	-
Captured Energy of U_2	-	-	-	100 %	53 %

7. EXAMPLE: A MODIFIED POINT-MASS PENDULUM

The simulation results, presented next, are based on a simple example of a point-mass pendulum. In this section, both *Open-Loop* (OL) and *Closed-Loop* (CL) analysis of our LPV modeling framework will be discussed. The rotational motion of a point-mass pendulum is given by

$$\frac{d}{dt}\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -bx_2(t) - a^2 \sin x_1(t) \end{bmatrix} + \begin{bmatrix} 0 \\ c \sin u(t) \end{bmatrix} \tag{17}$$

with $[x_1 \ x_2]^\top = [\theta \ \dot{\theta}]^\top$ the states, θ the rotation angle, u the input torque, $a = \sqrt{g/L}$ the angular frequency, $g = 9.8$ the acceleration due to gravity, $L = 3$ the pendulum length, $b = 2$ damping and $c \sin u(t)$ a fictional nonlinearity (with coefficient $c = 4$) for making the system more nonlinear. Obviously, (17) can exactly be recast into a quasi-LPV form using a global approach, i.e., by choosing two scheduling parameters p_1 and p_2, such that $p_1(t) = \sin x_1(t)/x_1(t)$ and $p_2(t) = \sin u(t)/u(t)$. We have intentionally chosen a simple example to better illustrate the practicality of our modeling method, which will be used to derive multiple LPV models.

7.1. LPV model synthesis

To gather data for LPV modeling, we excite the pendulum model, from its rest position, with a 20 sec long sine-sweep $u(t) = \sin(2\pi f t)$, with frequency f in the range 0.001–1 Hz, sampled with $T_s = 0.05$ sec, resulting in $N = 401$ data points. The purpose is also to illustrate the applicability of our modeling method in a conservative context, i.e., in the case where the control input signal-richness (used for data generation) is rather limited, as is the case with this single sine-sweep chosen here, and for the case of a relatively large sampling period, resulting in few data points. Furthermore, we also use a frequency range of interest defined as $\mathcal{W} = [\omega_1, \omega_2] = [0, 10]$ Hz.

First, the *central* model $\hat{G}_0(s)$ is obtained according to Eq. (7) with $\rho = 100$. The resulting model is nr. 185, i.e., $G_{185}(s)$. Table I is given to provide an overview of the SVD results of Step 2 described in Section 4 to derive the linear maps, where the *captured energy* refers to the percentage ratio between the sum of the retained singular values to the sum of all singular values. To analyze our modeling framework, we will use three LPV models, the first two to evaluate the OL response, whereas the third one will be used for dynamic output feedback control design[¶]. The first two models assume full-information, whereas the third corresponds to the case where only state x_1 is measured:

1. **Model M1**. Generated with $n_g = 3$, $n_d = 2$ and a 10-neurons network with $p(t) = \phi(x_1(t), x_2(t))$; $n_r = 5$.
2. **Model M2**. Generated with $n_g = 1$, $n_d = 1$ and a 10-neurons network with $p(t) = \phi(x_1(t), x_2(t))$; $n_r = 2$.
3. **Model M3**. Generated with $n_g = 1$, $n_d = 1$ and a 10-neurons network with $p(t) = \phi(x_1(t))$; $n_r = 2$.

Note that mapping $\phi(\cdot)$ is here function of the states only, rather than both states and inputs, since better validation results were obtained this way when exciting the LPV models with fresh inputs (i.e. inputs not used during the modeling process). Next, to compare the effectiveness of the proposed LPV models, we define the following two cost functions:

[¶]In most practical situations, when designing control systems, one does not have access to the full state-vector. In the case of the pendulum, often only the rotation angle θ is being measured.

DOI: 10.1002/rnc

Table II. Quality of the LPV approximation in terms of J_∞ and J_2.

Model	J_∞ (init. by (13))	J_∞	J_2
M1	0	N.A.	100
M2=M3	0.34	0.32	74

1. **Cost C1**. Mean of the local TF deviation in terms of $J_\infty := \frac{1}{N}\sum_{i=1}^{N} \|\bar{G}_i(s) - G_i(s)\|_\infty^{\mathcal{W}}$;
2. **Cost C2**. Fit of the affine remainder terms by

$$J_2 := \frac{1}{n_x}\sum_{k=1}^{n_x} \mathrm{BFR}\Big(\{[\bar{d}_i]_k\}_{i=1}^{N}, \{[d_i]_k\}_{i=1}^{N}\Big),$$

where $d_i = \sum_{r=1}^{n_r} \eta_r(i)(T_r x_i + Z_r u_i)$, $[\cdot]_k$ stand for referring to the k^{th} element of a vector and for two signals $z(i)$ and $\hat{z}(i)$ the *Best Fit Rate* (BFR) is defined as

$$\mathrm{BFR}\Big(z(i), \hat{z}(i)\Big) := 100\% \cdot \max\left(1 - \frac{\left\|z(i) - \hat{z}(i)\right\|_{\ell_2}}{\left\|z(i) - \mathrm{mean}(z(i))\right\|_{\ell_2}}, 0\right).$$

Note that costs C1 and C2 evaluate the modeling concept before the synthesis of the NN component.

The results are given in Table II, where all LMIs, used to compute cost C1, are solved using YALMIP [38] with solver SeDuMi [39]. For model M1, since we kept the full linear map, the cost functions J_∞ and J_2 reveal a perfect match in both aspects. On the other hand, models M2 and M3 use only the most dominant singular values associated linear map. M2 and M3 are equivalent in terms of J_∞ and J_2, since they differ only in their respective NN representation of the scheduling map. We can observe that J_2 is relatively large for these models (which is good), and that the simple approach Eq. (13), to compute an initialization point, gives a reasonably low value for J_∞. For this example, the refinement steps to solve the bilinear optimization problem do not provide significant improvement, although on a different example [40] it did provide substantial improvements. This preliminary evaluation of the model accuracy shows that models M2 and M3, although based on only two scheduling signals, may potentially provide good model fidelity in OL. In the sequel, additional evaluations of the OL and the CL behaviors is given.

7.2. Open-loop analysis

To better compare the effectiveness of the proposed LPV models, we define the following additional cost functions:

1. **Cost C3**. For a comparison of time-domain responses on $[0, \infty)$, we use fresh data sets, namely step-inputs, and sine-inputs at varying amplitudes and frequencies and compute the $\mathrm{BFR}(\{y(iT_s)\}_{i=1}^{\bar{N}}, \{\hat{y}(iT_s)\}_{i=1}^{\bar{N}})$ between the sampled NL model response y and the response $\hat{y}$ provided by the LPV model.
2. **Cost C4**. For similar purposes we also compute the *Variance-Accounted-For* (VAF)

$$\mathrm{VAF} := 100\% \frac{1}{n_x}\sum_{k=1}^{n_x} \max\Big(1 - \frac{\mathrm{var}(y(iT_s) - \hat{y}(iT_s))}{\mathrm{var}(y(iT_s))}, 0\Big).$$

In this section, we will test the resulting CT LPV models with the NN synthesized scheduling maps. Note that all models become now quasi-LPV models. We will compare next the behavior of these LPV models with that of the NL system. We excite the LPV models with data sets not used during the modeling phase. First, we use sine-inputs, for several fixed amplitudes and fixed frequencies, and present the respective BFR and VAF for each model in Table III through V. Overall, all three models exhibit good to excellent fit with the NL model, using these error measures, for

DOI: 10.1002/rnc

Table III. Time Response for M1. Left value is BFR (%), Right value is VAF (%).

Input Amplitude	Input Frequency (Hz) 0.25	0.5	0.75	1
0.25	93 99	94 100	96 100	97 100
0.5	97 100	91 99	94 100	94 100
0.75	93 100	90 99	91 99	92 100
1	94 100	91 99	90 99	90 99
1.5	78 97	81 97	79 97	73 95
1.75	0 0	54 88	69 95	61 92

Table IV. Time Response for M2. Left value is BFR (%), Right value is VAF (%).

Input Amplitude	Input Frequency (Hz) 0.25	0.5	0.75	1
0.25	93 100	87 98	90 99	91 99
0.5	94 100	87 98	88 99	89 99
0.75	96 100	90 99	90 99	90 99
1	94 100	94 100	93 100	91 99
1.5	83 97	86 98	80 98	77 97
1.75	80 96	77 96	68 95	63 94

Table V. Time Response for M3. Left value is BFR (%), Right value is VAF (%).

Input Amplitude	Input Frequency (Hz) 0.25	0.5	0.75	1
0.25	91 99	87 98	90 99	90 99
0.5	93 99	86 97	88 98	88 99
0.75	95 100	88 98	87 98	89 99
1	97 100	91 99	88 98	81 97
1.5	73 95	85 98	81 97	62 94
1.75	55 90	74 96	70 94	53 91

input amplitudes below one (i.e., the value used during model synthesis which defines the operating regime). The accuracy of these LPV models diminishes when the input amplitude is increased above one, even though model M2 still retains a relatively good fit. We also note that model M2, even though based on fewer scheduling variables than M1, is roughly at least as good as model M1. This may be explained by the fact that the NN models were trained with local data sets only with a few number of samples while these responses incorporate transient dynamics as well. Good training data sets may be two orders of magnitude bigger, in the tens of thousands of points rather than a few hundreds [41]. Hence, a model with fewer to-be-estimated parameters, like M2, may provide, in this case, a higher quality model, but this is unpredictable as the effect of transients is not visible from the point of view of local linearizations. Next, the fit for model M3 is slightly worse than that of M2, e.g., for input amplitudes above one. This may be explained by the fact that the training of M3 was based on state x_1 only. Finally, we also compare the model responses to a step input of amplitude $A = 0.5$, with the outcomes given in Table VI, and Fig. 1 through 3, where again the respective high model quality is being confirmed. In summary, model M2 provides good model fidelity in OL, coupled with slightly better computational efficiency than model M1 (since having fewer scheduling variables and NN models to evaluate), and may thus be used for OL prediction, whereas model M3 has also shown to be a suitable candidate for subsequent controller design, in a dynamic output feedback framework based upon the measurement of x_1 only.

DOI: 10.1002/rnc

Table VI. Time Response to Step Input $u(t)$ of Amplitude $A = 0.5$.

Model	BFR (%)	VAF (%)
M1	70	97
M2	72	98
M3	55	96

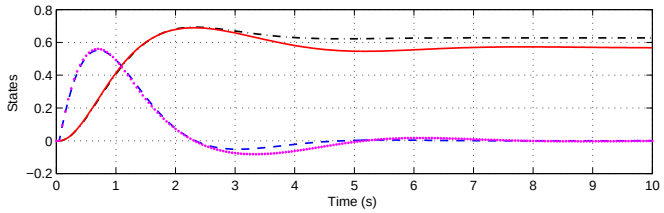

Figure 1. M1 response for step input $u(t)$ of Amplitude $A = 0.5$ (legend: '–.' NL x_1; '– –' NL x_2; '–' LPV x_1; '.' LPV x_2).

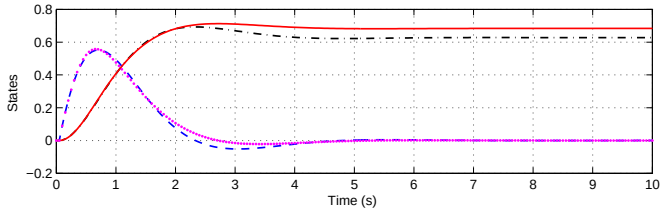

Figure 2. M2 response for step input $u(t)$ of Amplitude $A = 0.5$ (legend: '–.' NL x_1; '– –' NL x_2; '–' qLPV x_1; '.' qLPV x_2).

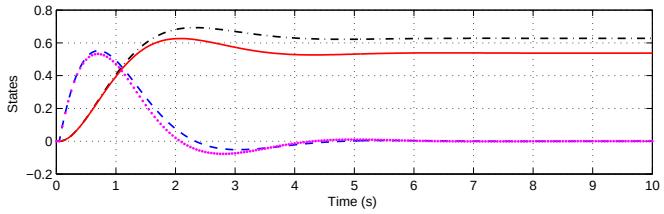

Figure 3. M3 response for step input $u(t)$ of Amplitude $A = 0.5$ (legend: '–.' NL x_1; '– –' NL x_2; '–' qLPV x_1; '.' qLPV x_2).

7.3. Closed-loop analysis

The objective of this section is to evaluate the quasi-LPV model M3 in a CL setting. To this end, we define a generalized plant $\breve{P}$ which maps the generalized disturbances $w(t) = [r^\top(t)\ n^\top(t)]^\top$ and control inputs $u(t)$, to controlled outputs, i.e., generalized performance channels, $z(t) = [z_u^\top(t)\ z_p^\top(t)]^\top$ and measured outputs $v(t) = [r^\top(t)\ e^\top(t)]^\top$, see Fig. 4. The signals consist of $r(t)$ the reference signal, $n(t)$ the sensors noise, $e(t)$ the tracking errors, $z_u(t)$ the actuators performance signal (to limit actuation magnitudes and rates), and $z_p(t)$ the desired performance in terms of closed-loop signal responses. As shaping filters, the performance weight $W_p(s)$, the actuator weight $W_u(s)$, and the sensor noise weight $W_n(s)$, all given in Appendix B, are used. The shaped generalized plant $\breve{P}$ is further given by

$$\begin{bmatrix} z_u \\ z_p \\ r \\ e \end{bmatrix} = \begin{bmatrix} 0 & 0 & W_u \\ W_p & 0 & -W_p P \\ I & 0 & 0 \\ I & -W_n & -P \end{bmatrix} \begin{bmatrix} r \\ n \\ u \end{bmatrix}, \tag{18}$$

DOI: 10.1002/rnc

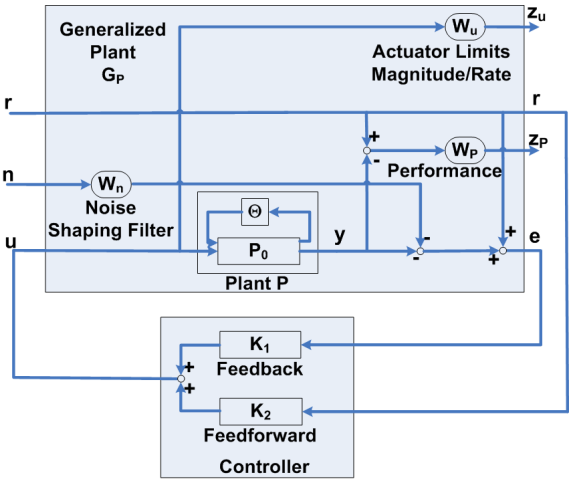

Figure 4. Control synthesis: two degrees-of-freedom control structure.

where P denotes the dynamic operator defined by the LPV model. The goal of the control synthesis is to find a two-degrees-of-freedom dynamic controller $K = [K_1 \;\; K_2]$ that establishes closed-loop stability, while guaranteeing an induced $\mathcal{L}_2$-gain bound γ from the exogenous inputs w to the controlled outputs z, such that

$$\int_0^T z^\top(t)z(t)\,\mathrm{d}t \;\leq\; \gamma^2 \int_0^T w^\top(t)w(t)\,\mathrm{d}t \quad \forall T \geq 0. \tag{19}$$

In the sequel, we will synthesize four controllers—one $\mathcal{H}_\infty$ LTI, one robust μ LTI, and two LPV controllers—and compare their reference tracking performance to step reference inputs. As already mentioned, the controller synthesis is based upon a two-degrees-of-freedom control structure, see Fig. 4. Besides a feedback action K_1, a feedforward action K_2 is indeed added to improve tracking performance. For optimality, both the feedback and feedforward parts of the controller K will be designed in one step. The four controllers are designed as follows:

7.3.1. Controller 1: LTI $\mathcal{H}_\infty$ method The purpose is here to synthesize an LTI controller which is not based on model M3, but rather based upon a nominal LTI model P_{nom}. This P_{nom} is obtained from a single linearization, at the equilibrium point $[x_1 \;\; x_2]^\top = [0 \;\; 0]^\top$, of the NL model (17), see Appendix B. Furthermore, for this $\mathcal{H}_\infty$ LTI controller, the control synthesis does not include any robustness with respect to some uncertainty Θ, hence in Fig. 4 we have

$$P = P_0 = P_{\text{nom}}. \tag{20}$$

Now, with Eq. (20) in mind, our control synthesis problem is equivalent to the minimization of the $\mathcal{H}_\infty$ norm of a standard, weighted, mixed-sensitivity S/KS criterion. Here, the dynamic controller K is computed such that [42]

$$K = \arg\inf_K \|F_l(\check{P}, K)\|_\infty. \tag{21}$$

We will consider this controller as the nominal controller.

7.3.2. Controller 2: Robust μ method The synthesized affine LPV model M3, as defined in Eq. (3), is first re-scaled, in terms of normalization of $p(t)$ and given by

$$P(\alpha(t)) := \left\{ \begin{array}{l} \dot{x}(t) = \tilde{A}_0 x(t) + \tilde{B}_0 u(t) \\ \qquad + \sum_{r=1}^{2} \alpha_r(t)\Big(\tilde{A}_r x(t) + \tilde{B}_r u(t)\Big) \end{array} \right. \tag{22}$$

such that $\alpha = [\alpha_1 \;\; \alpha_2]^\top$ and $|\alpha_r(t)| \leq 1$. Here, the generalized plant $\check{P}$ has a linear fractional dependence on the normalized scheduling variable $\alpha(t)$. This plant can be represented by the upper

DOI: 10.1002/rnc

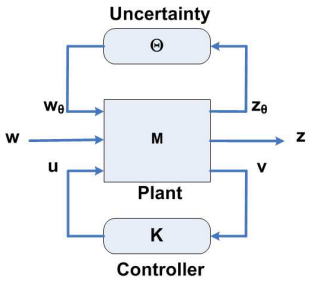

Figure 5. Standard $M - \Theta - K$ robust control setting.

LFT interconnection

$$\begin{pmatrix} z \\ v \end{pmatrix} = \check{P} \begin{pmatrix} w \\ u \end{pmatrix} = F_{\mathrm{u}}(M, \Theta) \begin{pmatrix} w \\ u \end{pmatrix}, \tag{23}$$

where $M(s)$ is a known LTI plant, see Fig. 5. Furthermore, $\Theta := \mathrm{blockdiag}\{\alpha_1(t) I_{c_1}, \alpha_2(t) I_{c_2}\}$ specifies how the scheduling variables enter to the plant dynamics, whereas I_{c_r} is an identity matrix whose size c_r corresponds to the minimal *Linear Fractional Realization* (LFR) of (22), i.e., $c_r = \mathrm{rank}([\tilde{A}_r \;\; \tilde{B}_r])$. Next, the feedback structure associated with the LFT interconnection Eq. (23) is given by

$$\begin{pmatrix} z_\theta \\ z \\ v \end{pmatrix} = M \begin{pmatrix} w_\theta \\ w \\ u \end{pmatrix}, \tag{24a}$$

$$w_\theta = \Theta z_\theta, \tag{24b}$$

with $z_\theta(t)$, and $w_\theta(t)$, the inputs and outputs of operator Θ, shown in Fig. 5. According to the robust control framework, we further proceed by treating the scheduling parameter variations, i.e., given in Θ, as real static uncertainties (not measured on-line). Again, the goal of the controller synthesis is to minimize the $\mathcal{L}_2$-gain bound γ from the exogenous inputs $w(t)$ to the controlled outputs $z(t)$, despite the uncertainty Θ with $\bar{\sigma}(\Theta) \leq 1$. This is achieved by synthesizing an LTI K through the D-K iteration [43]:

$$K = \arg \inf_{K} \inf_{D_{\mathrm{s}}} \| D_{\mathrm{s}} F_{\mathrm{l}}(M, K) D_{\mathrm{s}}^{-1} \|_\infty, \tag{25}$$

with $D_{\mathrm{s}}(s)$ a stable and minimum-phase scaling matrix, chosen such that $D_{\mathrm{s}}(s)\Theta = \Theta D_{\mathrm{s}}(s)$. By the `dksyn` implementation in MATLAB, after five iterations a 12^{th} order controller based on a 8^{th} order $D(s)$-scalings is obtained. The controller is further reduced to 5^{th} order after balanced Hankel-norm model reduction without any significant effect on CL robustness/performance. In summary, we have obtained here a single robust LTI controller, for a family of LTI plants. In the next section, we will see how to design an LPV controller that exploits that $p(t)$ is measurable in the system.

7.3.3. Controller 3 & 4: LPV methods LPV control has received much attention over the past 20 years resulting in a plethora of control methods. Although, a full review of LPV control methods is beyond the scope of this paper most methods can be classified to either the set of *polytopic* approaches, e.g., [4], or the set of LFR or *norm-bounded* approaches [3, 6]. Alternatively, the methods can also be categorized as *Parameter-Independent Lyapunov Function* (PILF) techniques (such as the methods listed above) versus the so-called *Parameter-Dependent Lyapunov Function* (PDLF) approaches, also known as gridding methods, [5, 35]. While PILF methods enjoy simplicity and numerical tractability, PDLF methods can improve performance, i.e., decrease conservatism, in case variation bounds of the scheduling variable are known. Here, we investigate control synthesis with a polytopic PILF [4] and a scaled small-gain PDLF [35] method.

Controller 3: Polytopic PILF. In the LPV model $P(\alpha(t))$, given by Eq. (22), the scheduling parameter $\alpha(t)$ is defined on a compact set $\mathcal{P}_\alpha$, represented by a unit hypercube of dimension 2 with vertices $\{w_j\}_{j=1}^{2^{n_r}}$ corresponding to the extremal values of α. The considered polytopic PILF method,

DOI: 10.1002/rnc

implemented as `hinfgs`, follows the lines of classical $\mathcal{H}_\infty$ synthesis, with the difference that it is based upon the $\mathcal{H}_\infty$ quadratic stability and performance concept (since both plant and controller are time-varying). The global LPV controller $K(\alpha(t))$ is obtained through interpolation of local controllers, the latter being synthesized at each vertex point $P(w_j)$ [4]. Since the method requires the control-matrix to be independent of the time-varying scheduling parameter, we pre-filtered the LPV model with the low-pass filter defined at the beginning of Section 7.1. A gain $\gamma = 0.92$, in Eq. (19), was achieved with the weights defined in Appendix B. Although the synthesized controller $K(\alpha(t))$ is time-varying—and hence represents an improvement compared to the previous LTI μ controller—the quadratic stability and performance concept assumes arbitrarily fast varying scheduling parameters $\alpha(t)$. Obviously this may result in some conservatism, in case the scheduling parameters have a bounded rate of variation.

Controller 4: Small-gain PDLF. This last controller is also referred in the sequel as the LPV-LFT controller. Both plant and controller are now dependent on the time-varying scheduling parameter $\Theta(t)$ according to the LFR formulation in Section 7.3.2. Now the main difference w.r.t. the robust control design is that the time-variation of $\Theta(t)$ will be taken explicitly into account. The to-be-designed LPV controller $K(\Theta(t))$ is obtained by minimizing the induced $\mathcal{L}_2$-norm of the closed-loop operator in terms of (19). Moreover, the controller synthesis method also takes parameter time-derivative into account, implying a dependence on both $\Theta(t)$ and its derivative $\dot{\Theta}(t)$. This results in an infinite-dimensional LMI problem which, in our case, was tackled by using a small grid, containing only the extrema of $\Theta(t)$ and $\dot{\Theta}(t)$ [35]. Since the method [35] is an iterative method, good starting values for the scalings were obtained by performing a robust μ synthesis, with constant scalings, on the $(\tilde{A}_0, \tilde{B}_0)$ plant (this plant is defined by (22)). A gain $\gamma = 0.51$, in Eq. (19), was achieved with the weights defined in Appendix B, after ten iterations.

7.3.4. Discussion of results The evaluation of the performance of all controllers with the NL plant is done using step inputs on the reference signal, starting from a zero initial condition, i.e., pendulum at rest, see Fig. 6–Fig. 8. With respect to our LPV modeling method, we can arrive to the following observations:

- The nominal LTI H_∞ controller exhibits a steady-state error, which remains persistent despite several modifications of the performance weight $W_\mathrm{p}(s)$. All other controllers designed using our LPV modeling methodology, i.e., model M3, do not exhibit any steady-state error, and hence achieve much better reference tracking. This is achieved even though model M3 has been built with only two scheduling variables.
- As anticipated, the PDLF controller is less conservative than the PILF (compare the achieved γ values).
- The robust μ controller and the polytopic PILF LPV controller exhibit very similar tracking performance, although the control input of the latter one is much smoother. Comparison of robust μ control with several LPV control methods has been addressed in many papers, e.g., [44], in which it was reported that LPV methods are less conservative than a standard μ approach. Indeed, the distinct advantage of LPV control methods is based upon the on-line measurement of the scheduling parameters (and potentially its derivatives). However for LPV-LFT methods, this advantage needs to be put into perspective, since almost all LPV-LFT control methods (except [45]) have been based upon static scaling, whereas μ synthesis uses dynamic scaling.

Based on these observations, we can conclude that the LPV model M3 determined using the approach proposed in this paper is a good model for the control of the nonlinear system (17).

8. CONCLUSION

In this paper a comprehensive affine LPV modeling framework is presented which allows to derive models which are suitable for open-, and closed-loop applications such as robust and LPV controller

DOI: 10.1002/rnc

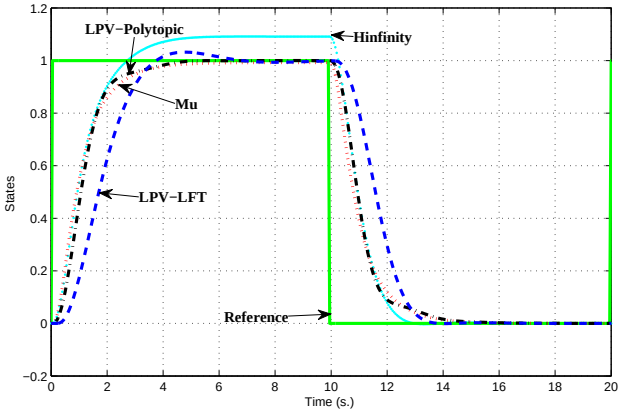

Figure 6. Closed-loop step response ($r \to x_1$) of the NL model with the synthesized controllers.

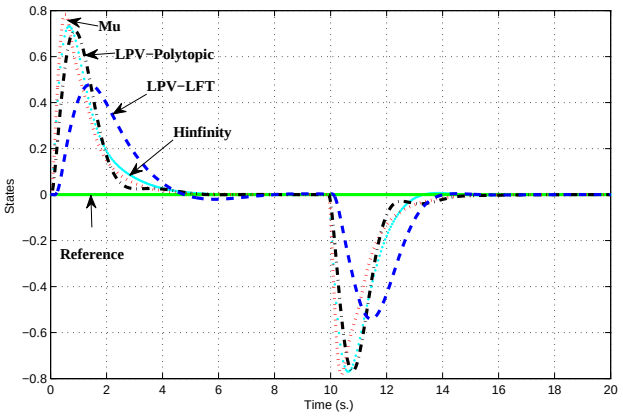

Figure 7. Closed-loop step response ($r \to x_2$) of the NL model with the synthesized controllers.

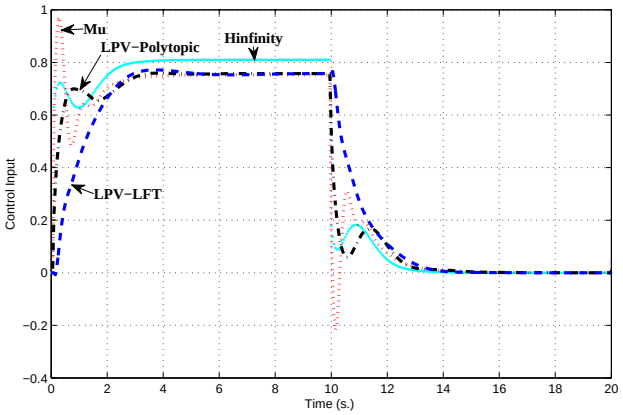

Figure 8. Closed-loop step response ($r \to u$) of the NL model with the synthesized controllers.

design, and has been demonstrated on a control design example of a point-mass pendulum. Since our LPV modeling approach does not incorporate any information on parameter time-derivatives, it is expected that significant enhancements could potentially be obtained in that respect.

DOI: 10.1002/rnc

ACKNOWLEDGEMENT

This work was supported by the Netherlands Organization for Scientific Research (NWO, grant no. 639.021.127).

APPENDIX A: KALMAN-YAKUBOVICH-POPOV LEMMA WITH SPECTRAL MASK CONSTRAINTS

We recall here how to compute the $\|\cdot\|_\infty^{\mathcal{W}}$ norm, i.e., the $\mathcal{H}_\infty$ norm with spectral mask constraints, through the use of the *Kalman-Yakubovich-Popov* (KYP) lemma with spectral constraints, see, e.g., [46].

Lemma 1
Let real scalars $\omega_1 \le \omega_2$, $\omega_c = (\omega_1 + \omega_2)/2$, and a transfer function $G(s) = C(sI - A)^{-1}B + D$ with minimal state-space realization (A, B, C, D) be given. The following statements are equivalent.

1. $\forall \gamma > 0,\ \lambda(A) \subset \mathbb{C}^- \cup \mathbb{C}^+,\ \|G\|_\infty^{\mathcal{W}} < \gamma;$ (26)
2. There exists matrices R and Q, of appropriate size, such that $R = R^*,\ Q > 0$, and $L(R, Q) + S(\gamma^2) < 0$, with
$$L(R,Q) = \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}^* \begin{bmatrix} -Q & R + j\omega_c Q \\ R - j\omega_c Q & -\omega_1\omega_2 Q \end{bmatrix} \begin{bmatrix} A & B \\ I & 0 \end{bmatrix}$$
$$S(\gamma^2) = \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}^* \begin{bmatrix} I & 0 \\ 0 & -\gamma^2 I \end{bmatrix} \begin{bmatrix} C & D \\ 0 & I \end{bmatrix}; \quad (27)$$
3. There exists matrices F and K, of appropriate size, such that $\forall l \in \{1, 2\}$, $M_l(F, K) + S(\gamma^2) < 0$
$$M_l(F,K) = \mathrm{He}\left(\begin{bmatrix} F \\ K \end{bmatrix} \begin{bmatrix} I & -j\omega_l I \end{bmatrix} \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \right). \quad (28)$$

Hence, the norm $(\|\cdot\|_\infty^{\mathcal{W}})^2$ is obtained by minimizing the bound γ^2 defined in (26), which is computationally done by minimizing γ^2 subject to the LMI in item 2) or 3). Let n be the number of decision variables and m the number of rows of LMIs, then comparing 2) and 3) shows that, while both have similar m, they differ in terms of n, i.e., $n_x^2 + n_x$ versus $n_x^2 + n_x n_u$, respectively. Since the asymptotic computational complexity, or flop cost, of SDP solvers is in $\mathcal{O}(n^2 m^{2.5} + m^{3.5})$ for SeDuMi [39], and in $\mathcal{O}(n^3 m)$ for MATLAB LMI-lab [47], the former approach is more efficient for large problems, and hence is the method we will use in this paper, however, the latter has the advantage that, for fixed F and K, it is also affine in the A and B matrices, and hence can be used in a bi-convex framework.

APPENDIX B: PROBLEM DATA

The nominal model, corresponding to a linearization of the pendulum NL model at $[x_1\ x_2]^\top = [0\ 0]^\top$, used for $\mathcal{H}_\infty$ controller design, is given by

$$A_{\mathrm{nom}} = \begin{bmatrix} 0 & 1 \\ -3.2667 & -2 \end{bmatrix}, \quad B_{\mathrm{nom}} = \begin{bmatrix} 0 \\ 4 \end{bmatrix}.$$

The resulting LPV model in terms of (3) is given by

$$A_0 = \begin{bmatrix} 0 & 1 \\ -2.7915 & -2 \end{bmatrix}, \quad B_0 = \begin{bmatrix} 0 \\ 3.0631 \end{bmatrix},$$
$$A_1 = \begin{bmatrix} 0 & 0 \\ -0.0170 & 0 \end{bmatrix}, \quad B_1 = \begin{bmatrix} 0 \\ -1 \end{bmatrix},$$
$$A_2 = \begin{bmatrix} 0 & 0 \\ 0.2205 & -0.3446 \end{bmatrix}, \quad B_2 = \begin{bmatrix} 0 \\ -0.9125 \end{bmatrix}.$$

The matrices for the normalized and centered model (22) are

$$\tilde{A}_0 = \begin{bmatrix} 0 & 1 \\ -2.8896 & -1.8459 \end{bmatrix}, \quad \tilde{B}_0 = \begin{bmatrix} 0 \\ 3.4962 \end{bmatrix},$$

DOI: 10.1002/rnc

$$\tilde{A}_1 = \begin{bmatrix} 0 & 0 \\ -0.0159 & 0 \end{bmatrix}, \quad \tilde{B}_1 = \begin{bmatrix} 0 \\ -0.9342 \end{bmatrix},$$

$$\tilde{A}_2 = \begin{bmatrix} 0 & 0 \\ 0.1556 & -0.2433 \end{bmatrix}, \quad \tilde{B}_2 = \begin{bmatrix} 0 \\ -0.6441 \end{bmatrix}.$$

The maximum rates of scheduling variations for the LPV-LFT synthesis are

$$\bar{\dot{\alpha}}_1 = 11.59, \ \underline{\dot{\alpha}}_1 = -12.10, \quad \bar{\dot{\alpha}}_2 = 11.13, \ \underline{\dot{\alpha}}_2 = -11.72.$$

The LTI performance weights in Fig. 4 are chosen as

$$W_{\mathrm{u}}(s) = \frac{s}{s + 2\pi}, \quad W_{\mathrm{n}}(s) = 0.005.$$

For the $\mathcal{H}_\infty$, μ, and LPV-LFT controllers, the performance wight has been tuned to

$$W_{\mathrm{p}}(s) = \frac{s/2 + 0.25\pi}{s + \frac{0.25\pi}{10^2}}.$$

For the polytopic LPV controller, we have used

$$W_{\mathrm{p}}(s) = \frac{s/2 + 0.25\pi}{s + \frac{0.25\pi}{10^6}}.$$

REFERENCES

1. J. S. Shamma and M. Athans, "Analysis of gain scheduled control for nonlinear plants," *IEEE Trans. Autom. Control*, vol. 35, no. 8, p. 898907, 1990.
2. R. Toth, *Identification and Modeling of Linear Parameter-Varying Systems. Lecture Notes in Control and Information Sciences, Vol. 403*. Heidelberg: Springer, 2010.
3. A. Packard, "Gain scheduling via linear fractional transformations," *Systems & Control Letters*, vol. 22, pp. 79–92, 1994.
4. P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled h_∞ control of linear parameter-varying systems: A design example," *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.
5. F. Wu, X. H. Yang, A. K. Packard, and G. Becker, "Induced $\mathcal{L}_2$ norm control for lpv systems with bounded parameter variation rates," *Int. J. Of Robust And Nonlinear Control*, vol. 6, no. 9/10, pp. 983–998, 1996.
6. C. W. Scherer, "Lpv control and full block multipliers," *Automatica*, vol. 37, pp. 361–375, 2001.
7. K. J. Astrom and B. Wittenmark, *Adaptive Control*. Reading, MA: Addison-Wesley, 1989.
8. G. J. Balas, I. Fialho, A. Packard, J. Renfrow, and C. Mullaney, "On the design of lpv controllers for the f-14 aircraft lateral-directional axis during powered approach," in *Am. Control Conf.*, 1997.
9. G. J. Balas, J. B. Mueller, and J. M. Barker, "Application of gain-scheduled, multivariable control techniques to the f/a-18 system research aircraft," in *AIAA Guidance Navigation and Control Conf.*, 1999.
10. A. Marcos and G. J. Balas, "Development of linear-parameter-varying models for aircraft," *AIAA J. of Guidance, Control, and Dynamics*, vol. 27, no. 2, pp. 218–228, 2004.
11. K. Z. Ostergaard, "Robust, gain-scheduled control of wind turbines," Ph.D. dissertation, Aalborg University, 2008.
12. M. G. Wassink, M. van de Wal, C. Scherer, and O. Bosgra, "Lpv control for a wafer stage: Beyond the theoretical solution," *Control Engineering Practice*, vol. 13, pp. 231–245, 2004.
13. R. Tóth, M. van de Wal, P. S. C. Heuberger, and P. M. J. Van den Hof, "LPV identification of high performance positioning devices," in *Proc. of the American Control Conf.*, San Francisco, California, USA, June 2011, pp. 151–158.
14. M. Dettori, "Lmi techniques for control," Ph.D. dissertation, Delft University of Technology, 2001.
15. R. Murray-Smith and T. Johansen (Ed), *Multiple Model Approaches to Modeling and Control*. CRC Press, 1997.
16. W. Rugh, "Analytical framework for gain scheduling," *IEEE Control Systems Magazine*, vol. 11, pp. 74–84, 1991.
17. J. Shamma and M. Athans, "Gain scheduling: Potential hazards and possible remedies," *IEEE Control Systems Magazine*, vol. 12, pp. 101–107, 1992.
18. R. Bos, X. Bombois, and P. M. J. Van den Hof, "Accelerating simulations of computationally intensive first principle models using accurate quasi-linear parameter varying models," *J. of Process Control*, vol. 19, no. 10, pp. 1601–1609, 2009.
19. D. Petersson and J. Lofberg, "Optimization based lpv-approximation of multi-model systems," in *Europ. Control Conf.*, 2009.
20. D. Vízer, G. Mercère, O. Prot, and E. Larochen, "Combining analytic and experimental information for linear parameter-varying model identification: Application to a flexible robotic manipulator," *Period. Polytech. Elec. Eng. Comp. Sci.*, vol. 58, no. 4, pp. 133–148, 2014.
21. H. Pfifer and S. Hecker, "Generation of optimal linear parametric models for lft-based robust stability analysis and control design," *IEEE Trans. on Control Systems Technology*, vol. 19, no. 1, pp. 118–131, 2011.

DOI: 10.1002/rnc

22. J. S. Shamma and J. R. Cloutier, "Gain-scheduled missile autopilot design using linear parameter varying transformations," *AIAA J. of Guidance, Control, and Dynamics*, vol. 16, no. 2, pp. 256–263, 1993.
23. D. J. Leith and W. E. Leithhead, "Gain-scheduled controller design: An analytic framework directly incorporating non-equilibrium plant dynamics," *Int. J. of Control*, vol. 70, pp. 249–269, 1998.
24. W. Tan, "Applications of linear parameter-varying control theory," Master Thesis, University of California at Berkeley, Tech. Rep., 1997.
25. H. Abbas, R. Tóth, M. Petreczky, N. Meskin, and J. Mohammadpour, "Embedding of nonlinear systems in a linear parameter-varying representation," in *Proc. of the 19th IFAC World Congress*, Cape Town, South Africa, Aug. 2014, pp. 6907–6913.
26. A. Kwiatkowski, M. T. Boll, and H. Werner, "Automated generation and assessment of affine lpv models," in *IEEE Conf. on Decision and Control*, 2006.
27. R. Tóth, P. S. C. Heuberger, and P. M. J. Van den Hof, "Pediction error identification of LPV systems: present and beyond," in *Control of Linear Parameter Varying Systems with Applications*, J. Mohammadpour and C. W. Scherer, Eds. Heidelberg: Springer, 2012, pp. 27–60.
28. T. A. Johansen, R. Shorten, and R. Murray-Smith, "On the interpretation and identification of dynamic takagi-sugeno fuzzy models," *IEEE Trans. on Fuzzy Systems*, vol. 8, no. 3, pp. 297–313, 2000.
29. R. Shorten, R. Murray-Smith, R. Bjorgan, and H. Gollee, "On the interpretation of local models in blended multiple model structures," *Int. J. of Control*, vol. 72, no. 7/8, pp. 620–628, 1999.
30. R. Toth, P. S. C. Heuberger, and P. M. J. Van den Hof, "Discretization of linear parameter-varying state-space representations," *IET Control Theory & Applications*, vol. 4, no. 10, pp. 2082–2096, 2010.
31. L. H. Lee and K. Poolla, "Identification of linear parameter-varying systems using nonlinear programming," *J. of Dynamic Systems, Measurement and Control*, vol. 121, p. 7178, 1999.
32. V. Verdult, L. Ljung, and M. Verhaegen, "Identification of composite local linear state-space models using a projected gradient search," *Int. J. of Control*, vol. 75, no. 16-17, pp. 1125–1153, 2002.
33. E. Prempain and I. Postlethwaite, "Extended lmi projection conditions," in *IEEE Conf. on Decision and Control*, 1999.
34. A. Helmersson, "Model reduction using lmis," in *IEEE Conf. on Decision and Control*, 1994.
35. P. Apkarian and R. J. Adams, "Advanced gain scheduling techniques for uncertain systems," *IEEE Trans. Automatic Control*, vol. 6, no. 1, pp. 21–32, 1998.
36. G. Stein and J. C. Doyle, "Beyond singular values and loop shapes," *AIAA J. of Guidance, Control, and Dynamics*, vol. 14, no. 1, pp. 5–16, 1991.
37. K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.
38. J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *CACSD Conf.*, 2004.
39. J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11-12, pp. 625–653, 1999.
40. S. Taamallah, X. Bombois, and P. M. J. Van den Hof, "Affine lpv modeling: An h_∞ based approach," in *IEEE Conf. on Decision and Control*, 2013.
41. M. B. Tischler, J. G. M. Leung, and D. C. Dugan, "Neural network-based trajectory optimization for unmanned aerial vehicles," *AIAA J. of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 548–562, 2012.
42. P. Gahinet and P. Apkarian, "A linear matrix inequality approach to h_∞ control," *Int. J. Of Robust And Nonlinear Control*, vol. 4, pp. 421–448, 1994.
43. K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. New Jersey: Prentice Hall, 1996.
44. P. Apkarian, J. M. Biannic, and P. Gahinet, "Self-scheduled h_∞ control of missile via linear matrix inequalities," *AIAA J. of Guidance, Control and Dynamics*, vol. 18, no. 3, pp. 532–538, 1995.
45. C. Scherer and I. E. Kose, "Gain-scheduled control synthesis using dynamic d-scales," *IEEE Trans. Autom. Control*, vol. 57, no. 9, pp. 2219–2234, 2012.
46. T. Iwasaki, G. Meinsma, and M. Fu, "Generalized s-procedure and finite frequency kyp lemma," *Math. Prob. Eng.*, vol. 6, pp. 305–320, 2000.
47. P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox Users Guide*. The Mathworks Partner Series, 1995.