NLR-TP-2014-285

Affine LPV Modeling: An H-infinity Based Approach

S. Taamallah, X. Bombois and P.M.J. Van den Hof

Nationaal Lucht- en Ruimtevaartlaboratorium

National Aerospace Laboratory NLR

Executive summary

Affine LPV Modeling: An H-infinity Based Approach

Problem area

One of the challenges in the design of high bandwidth control systems is the development of a global model, suitable for high performance controller synthesis, while concomitant with both high model fidelity in Open-Loop, e,g. for trajectory planning, and high computational efficiency for on-line processing. It is the purpose of this paper to present such a modeling structure, anchored in the realm of affine quasi- Linear Parameter-Varying (LPV) systems, for the specific case where a plant's complex nonlinear model is already available.

Description of work

We propose a novel LPV modeling method, having the following characteristics: 1) rather than fitting the LPV model within the popular H_2 norm framework, as is invariably the case for procedures the like of Prediction Error Methods (PEM), it forms the LPV model in the H∞ norm paradigm, much better suited for subsequent robust control design; 2) it allows the user to specify an input-signal frequency range of interest, in which the LPV model fidelity should be best; 3) it is not restricted to either, simple, or to specific subclasses of first-principles-based plant models, as it does not require for the availability of such white-box representations, and rather applies to any numerically-based model; 4) it is not restricted to equilibrium points, rather it captures the non-stationary and transient dynamical behavior; and 5) it is not limited by practical implementation constraints such as state differentiation or model-order increase.

Results and conclusions

We have presented a comprehensive affine quasi-LPV modeling method: 1) suitable for high-performance controller design, over the complete system operating regime; 2) having high model fidelity in Open-Loop; while 3) retaining high computational efficiency.

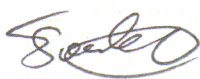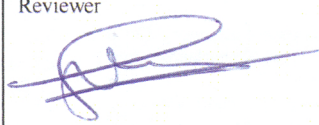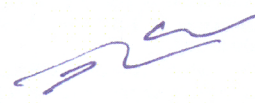### Applicability

System modeling and design of (flight) control systems.

This report is based on a presentation held at the IEEE Conference on Decision and Control (CDC), December 10 – 13, 2013, Florence, Italy.

NLR-TP-2014-285

Affine LPV Modeling: An H-infinity Based Approach

S. Taamallah, X. Bombois[1] and P.M.J. Van den Hof[2]

[1] TU-Delft

[2] TU-Eindhoven

Affine LPV Modeling: An $\mathcal{H}_\infty$ Based Approach

Skander Taamallah, Xavier Bombois and Paul M. J. Van den Hof

Abstract—**One of the challenges in the design of high-bandwidth control systems is the development of a global model, suitable for high-performance controller synthesis, while concomitant with both high model fidelity in Open-Loop (OL), e,g. for trajectory planning, and high computational efficiency for on-line processing. It is the purpose of this paper to present such a modeling structure, anchored in the realm of affine quasi- Linear Parameter-Varying (LPV) systems, for the specific case where a plant's complex Nonlinear Model (NM) is already available.**

I. INTRODUCTION

Our objective can be summed up as follows. Given an industrial plant for which a high-fidelity, yet complex, Nonlinear Model (NM) is available, obtained from either first-principles, empirical knowledge, or hybrid modeling, and given a simulated Input-Output (IO) signal sequence, collected under the desired operating conditions, find - for a user-defined input-signal frequency range of interest - a Reduced Complexity Model (RCM), that fulfills the following specifications

(1) Suitable for high-performance controller design, over the complete operating regime.
(2) Having high model fidelity in Open-Loop (OL). The RCM model shall be such that both: (i) accurate OL optimal trajectories may be obtained, and (ii) adequate feedforward controllers may be designed.
(3) Retaining high computational efficiency. The RCM shall be used as a substitute for a computationally intensive NM. Clearly, this may be the case for on-line use in a hard real-time environment (e.g. optimal trajectory generation, fault detection and isolation, estimation, adaptive control), whenever stringent timing constraints may need to be met, especially for high-bandwidth systems.

Two aspects within the aforementioned specifications - a global controller and OL model fidelity - may plead towards the use of nonlinear modeling, and its traditionally associated nonlinear control methods, which effectively respect and exploit the system's nonlinear structure. However, both the real-time computational requirement, and the sometimes endemic difficulty of these nonlinear approaches to easily handle performance criteria, have led

Skander Taamallah is with the National Aerospace Laboratory (NLR), Anthony Fokkerweg 2, 1059 CM, Amsterdam, The Netherlands, email: `staamall@nlr.nl`

Skander Taamallah, Xavier Bombois, are with the Delft Center for Systems and Control, Delft University of Technology, Mekelweg 2, 2628 CD, Delft, The Netherlands, email: `x.j.a.bombois@tudelft.nl`

Paul M. J. Van den Hof is with the Department of Electrical Engineering, Eindhoven University of Technology, Den Dolech 2, 5612 AZ, Eindhoven, The Netherlands, email: `P.M.J.vandenhof@tue.nl`

us towards the realm of Linear Parameter-Varying (LPV) systems. Indeed, LPV models are efficiently run on-line, and LPV control design problems are efficiently solved, by first expressing the problems as Linear Matrix Inequality (LMI) optimizations [1] - subsequently formulated as Semi-Definite Programs (SDP) [2] - for which there are several powerful numerical solutions [3], [4]. Within this setting, a literature review shows that twelve methods have been devised to translate a NM into a LPV one: (M1) direct conversion of simplified linear or nonlinear dynamics into LPV format, (M2) extended linearization [5] which is loosely based on pseudolinearization [6] and global linearization [7] ideas, (M3) Jacobian linearization [8], (M4) state transformation [9], (M5) Singular Value based Decompositions (SVD) [10], (M6) velocity-based formulation [11], (M7) Tensor-Product polytopic decomposition [12], [13], (M8) automated LPV model generation [14], [15], (M9) interpolation-based methods [16], [17], (M10) multivariable polynomial fitting [18], (M11) $\mathcal{H}_2$ norm minimization at sampled data points [19], and (M12) function substitution [20], [21], see also [22], [15] and references therein for a comprehensive review of LPV modeling. For all their benefits, these methods have also their shortcomings, namely the need for a white-box representation (M1,M2,M4,M8,M12), the restriction to simple plants, e.g. excluding non-algebraic forms such as look-up tables (M1,M2,M8), the lack of a general method to choose the scheduling variables (M1,M2,M3,M7), the difficulty in fulfilling the linearizability conditions (M2), the griding and selection of equilibrium points (M3,M4,M12), the model validity being only local around a set of equilibrium points (M4,M9,M10,M12), the misfit approximation, resulting from local changes of variable, as to eliminate the remainder terms and thus obtain a State-Space (SS) representation (M3,M7), the restrictions to a special class of nonlinear systems (M4,M12), the practical implementation constraints due to state differentiation and increase in model order (M6), the attempt to directly match the coefficients of each linearized SS matrices, since it is well known that the eigenvalues of some matrices may be highly sensitive to small, or even infinitesimal, perturbations of their matrix elements (M7), the practical implementation constraints, due to the rapid increase in the number of summands and leafs, together with the exponential growth of the number of tree routes (M8), the restriction to stable systems (M11), and the affine nature (in states and inputs) rather than linear representation (M5).

Hence, for complex, high-order, highly nonlinear, real-world industrial applications, the existing methods

of rewriting/approximating the nonlinear plant into LPV format are at best conservative (over-bounding), often inaccurate (poor transient performance), at times impractical (computational cost), and in some cases simply inadequate (inability to preserve stability). With respect to these LPV modeling pathologies, we propose a novel LPV modeling method, virtually eliminating most of these issues, as it is endowed by the following assets (i) not restricted to either, simple, or to specific subclasses of first-principles-based plant models, as it does not require for the availability of such white-box representations, and rather applies to any numerically-based model, (ii) not restricted to equilibrium points, rather it captures the non-stationary and transient system behavior, and (iii) not limited by practical implementation constraints such as state differentiation or model-order increase. In addition our method comes with the following benefits: (iv) allows the user to specify an input-signal frequency range of interest, in which the LPV model fidelity should be best, and (v) rather than fitting the LPV model within the popular $\mathcal{H}_2$ norm framework, as is invariably the case for procedures the like of Prediction Error Methods (PEM), it forms the LPV model in the $\mathcal{H}_\infty$ norm paradigm. Indeed, when spectral mask constraints are assigned, and compared to the $\mathcal{H}_2$ norm, we believe that the $\mathcal{H}_\infty$ norm framework provides the best compromise between modeling for control and modeling for OL simulation [23].

The nomenclature is fairly standard. Vectors are printed in boldface. M^T, M^*, $M^\dagger$ denote the transpose, the complex-conjugate transpose, and the Moore-Penrose inverse of a real or complex matrix M, whereas $\mathrm{He}(M)$ (resp. $\mathrm{Sym}(M)$) is shorthand for $M + M^*$ (resp. $M + M^T$). We use $\star$ as an ellipsis for terms that are induced by symmetry. Matrix inequalities are considered in the sense of *Löwner*. Further $\lambda(M)$ denotes the zeros of the characteristic polynomial $\det(sI - M) = 0$. $\mathcal{L}_\infty$ is the *Lebesgue* normed space s.t. $\|G\|_\infty := ess \sup_{\omega \in \mathbb{R}} \bar{\sigma}(G(j\omega)) < \infty$, with $\bar{\sigma}(G)$ the largest singular value of matrix $G(\cdot)$. Similarly, $\mathcal{H}_\infty \subset \mathcal{L}_\infty$ is the *Hardy* normed space s.t. $\|G\|_\infty := \sup_{\mathrm{Re}(s)>0} \bar{\sigma}(G(s))$. For $\omega_1 < \omega_2$, $\Delta_\omega = [\omega_1, \omega_2]$, we use $\|G\|_{\Delta_\omega} := \sup_{\omega \in \Delta_\omega} \bar{\sigma}(G(j\omega))$. $\mathcal{RL}_\infty$ (resp. $\mathcal{RH}_\infty$) represent the subspace of real rational Transfer Functions (TFs) in $\mathcal{L}_\infty$ (resp. $\mathcal{H}_\infty$).

II. PROBLEM STATEMENT

We suppose now that the noise-free NM is given[1] by

$$\forall t \geq 0 \qquad \dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \qquad \mathbf{y}(t) = \mathbf{x}(t) \qquad (1)$$

with $f(\cdot)$ a deterministic, Continuous-Time (CT), nonlinear function, which is at least C^1, and *locally* Lipschitz continuous. This *local* nature allows us to consider unstable systems as well. Further, we have $\mathbf{x}(t) \in \mathcal{P}_x \subset \mathbb{R}^{n_x}$ the plant state, $\mathbf{y}(t) \in \mathcal{P}_y \subset \mathbb{R}^{n_y}$ the plant output, $\mathbf{u}(t) \in \mathcal{P}_u \subset \mathbb{R}^{n_u}$ the control input, t the time variable,

---

[1] Since the nonlinear model (NM) is given, we assume that we have access to the full state vector.

and $(\mathcal{P}_x, \mathcal{P}_y, \mathcal{P}_u)$ some compact sets.

Remark 1: As stated earlier, we assume that a simulated IO signal sequence, collected under the desired operating conditions, is made available, and that this sequence is informative enough for the identification of the LPV model. Now since the notion of informativity has yet to be formalized within the LPV context, it has been excluded from our current framework.

Remark 2: In this paper we will encompass our discussion within the CT framework, since stability and performance requirements, for controller synthesis, are generally much more conveniently expressed in this framework. In case an equivalent LPV Discrete-Time (DT) realization is needed, this may be easily achieved by, either, discretizing the obtained CT LPV model through one of the LPV discretization methods presented in [24], or alternatively, by using the equivalent DT formulations of the machinery outlined in this paper.

Now we denote our RCM as the LPV model $\mathsf{S}(\theta(t))$

$$\mathsf{S}(\theta(t)) := \begin{cases} \forall t \geq 0 \\ \dot{\mathbf{x}}(t) = A_0 \mathbf{x}(t) + B_0 \mathbf{u}(t) + ... \\ \sum_{r=1}^{R} \theta_r(t) \big(A_r \mathbf{x}(t) + B_r \mathbf{u}(t) \big) \end{cases} \qquad (2)$$

with $\theta(t) := [\theta_r(t)]_{R \times 1} \in \mathbb{R}^R$, the non-stationary scheduling parameters, not known a priori, but on-line measurable and defined on the compact set $\mathcal{P}_\theta$, known as the scheduling space, and matrices (A_0, B_0, A_r, B_r), of appropriate sizes, representing the basis functions. For the case of endogenous parameter dependence, i.e. $\theta(\mathbf{x}(t), \mathbf{u}(t))$, the *quasi*-LPV prefix is added. Note that in this case, and in the occurrence of an unstable plant, such *quasi* parametrization is only licit if one assumes an upper bound on the sizes of $\mathcal{P}_x$ and $\mathcal{P}_y$, *a priori*. Further, we also chose to enclose our analysis within the affine LPV setting, with static scheduling-parameter dependence.

Remark 3: It is now well known that equivalent LPV IO vs. SS representations may generally necessitate dynamic dependence of the SS realization [25], since neglecting this fact may result in significant performance losses. However, as dynamic dependence may lead to difficulties in terms of controller design and implementation, we chose to limit our current discussion to static dependence only.

Remark 4: The affine LPV structure does not introduce any loss of generality, since it is well known that affine LPV formulations may (i) easily be cast into minimal Linear Factional Transformations (LFT) forms, and (ii) readily be transformed into (potentially non-unique) polytopic LPV forms, through barycentric computation [26].

Next we consider the practical situation where one needs to build a CT LPV model from sampled measurements

of the CT signals $(\mathbf{y}(t), \mathbf{u}(t), \boldsymbol{\theta}(t))$. These DT signals, subsumed with the sampling period $T_s > 0$, are denoted $\mathbf{u}(t_i) = \mathbf{u}(i.T_s), i \in \mathbb{Z}$, as illustrated here for the input signal $\mathbf{u}(\cdot)$. We assume that a simulated IO signal sequence $\mathcal{Z}^N := \{(\mathbf{u}(t_i), \mathbf{y}(t_i))\}_{i=1}^N$, collected under the *desired operating conditions*, is available. Building a CT LPV model from samples of measured CT signals has only been addressed recently in [27]. Our problem is however simpler, since we are dealing with a noise-free NM, avoiding thus the difficult question of CT random process modeling from a sampled CT noise source. Further, for LPV systems with static dependence, and concomitant to classical discretization theory [28], if the sampled and free-CT (i.e. inputs and exogenous parameters) signals can be assumed to be piecewise constant on a sampling period, then the CT output trajectory may be completely reconstructed from its sampled observations [15]. To fulfill this condition we obtain the following expression for the sampling period $T_s = 1/[K_s.max(f_{B_\mathbf{u}}, f_{B_\theta})]$ (resp. $T_s = 1/[K_s.f_{B_\mathbf{u}}]$) in case of LPV system (resp. quasi-LPV), with $f_{B_\mathbf{u}}$ and f_{B_θ} the bandwidth of the input and scheduling-parameter signals respectively, and $K_s \in [10 - 20]$ for all practical purposes.

In order to extend the validity of GS or LPV controllers to operating regions far from equilibrium points, as to better capture the transient behavior of the NM, we base our method upon the idea of non-stationary linearizations of the NM, along a given trajectory $\boldsymbol{\theta} := \boldsymbol{\theta}(t)$, as was suggested for the GS modeling framework in [29], [30], [31], and for the LPV modeling framework in [11], [32], [10]. These linearizations may be computed via first-order Taylor-series expansions, or via classical numerical perturbation methods. From (1) and set $\mathcal{Z}^N$, we create a set of triplet elements $\mathcal{Z}_{Lin}^N := \{(\bar{A}_i, \bar{B}_i, \mathbf{d}_i)\}_{i=1}^N$

$$\bar{A}_i = \frac{\delta f(\mathbf{x}, \mathbf{u})}{\delta \mathbf{x}^T}\bigg|_{(\mathbf{x}_i, \mathbf{u}_i)} \qquad \bar{B}_i = \frac{\delta f(\mathbf{x}, \mathbf{u})}{\delta \mathbf{u}^T}\bigg|_{(\mathbf{x}_i, \mathbf{u}_i)}$$
$$\mathbf{d}_i = f(\mathbf{x}_i, \mathbf{u}_i) - \bar{A}_i \mathbf{x}_i - \bar{B}_i \mathbf{u}_i \qquad (3)$$

with matrices $(\bar{A}_i, \bar{B}_i)$ bounded maps, since based on the regularity assumptions of $f(\cdot)$, and where we have also used the shorthand $\mathbf{x}_i := \mathbf{x}(t_i), \mathbf{u}_i := \mathbf{u}(t_i)$ to streamline notations. For each operating point $(\mathbf{x}_i, \mathbf{u}_i)$, we can approximate the NM model (1), in a local neighborhood, as

$$\dot{\mathbf{x}}(t_i) = f(\mathbf{x}(t_i), \mathbf{u}(t_i)) \approx \bar{A}_i \mathbf{x}(t_i) + \bar{B}_i \mathbf{u}(t_i) + \mathbf{d}_i \quad i = 1, ..., N \quad (4)$$

Next, from set $\mathcal{Z}_{Lin}^N$ we can generate a sequence of CT LTI TFs $\bar{G}_i(s) := \left[\begin{array}{c|c} \bar{A}_i & \bar{B}_i \\ \hline \bar{C} & \bar{D} \end{array}\right]$, with matrices $\bar{C} = I$, $\bar{D} = 0$ of appropriate size. Now, returning to our LPV definition of (2), we can expand it as

$$\mathsf{S}\big(\boldsymbol{\eta}(t), \boldsymbol{\zeta}(t)\big) := \begin{cases} \forall t \geq 0 \\ \dot{\mathbf{x}}(t) = A_0 \mathbf{x}(t) + B_0 \mathbf{u}(t) + ... \\ \sum_{s=1}^S \eta_s(t)\big(L_s \mathbf{x}(t) + R_s \mathbf{u}(t)\big) + ... \\ \sum_{w=1}^W \zeta_w(t)\big(T_w \mathbf{x}(t) + Z_w \mathbf{u}(t)\big) \end{cases} \quad (5)$$

for some scheduling parameters $\boldsymbol{\eta}(t) := [\eta_s(t)]_{S \times 1} \in \mathbb{R}^S$, $\boldsymbol{\zeta}(t) := [\zeta_w(t)]_{W \times 1} \in \mathbb{R}^W$, and matrices (L_s, R_s, T_w, Z_w), of appropriate sizes, s.t. $\sum_{r=1}^R \theta_r(t)A_r = \sum_{s=1}^S \eta_s(t)L_s + \sum_{w=1}^W \zeta_w(t)T_w$ and $\sum_{r=1}^R \theta_r(t)B_r = \sum_{s=1}^S \eta_s(t)R_s + \sum_{w=1}^W \zeta_w(t)Z_w$.

For each frozen-time trajectory $\eta_{s_i} := \eta_s(t_i)$, $\zeta_{w_i} := \zeta_w(t_i)$, $s = 1, ..., S$, $w = 1, ..., W$, associated with the time indexes $i = 1, ..., N$ of set $\mathcal{Z}^N$, we create both (i) a sequence of CT LTI TFs $G_i(s) := \left[\begin{array}{c|c} A_0 + \sum_{s=1}^S \eta_{s_i}L_s & B_0 + \sum_{s=1}^S \eta_{s_i}R_s \\ \hline C & D \end{array}\right]$, with matrices $C = I$, $D = 0$ of appropriate size, and (ii) a sequence of vectors $\sum_{w=1}^W \zeta_{w_i}\big(T_w \mathbf{x}_i + Z_w \mathbf{u}_i\big)$. From here on, we also assume $C = \bar{C} = I$ and $D = \bar{D} = 0$.

Remark 5: Note that we restrict our discussion to full-order LPV modeling, hence matrices (A_0, L_s, T_w) and $\bar{A}_i$ have same size (resp. (B_0, R_s, Z_w) and $\bar{B}_i$).

A. The Optimization Problems

The purpose of our work is now twofold. The first objective consists in using the frozen-time information available in set $\mathcal{Z}_{Lin}^N$ to identify the unknown scheduling parameters $\boldsymbol{\eta}(t)$, $\boldsymbol{\zeta}(t)$, and basis functions $(A_0, B_0, L_s, R_s, T_w, Z_w)$.

Remark 6: Note that we are well aware that LPV properties cannot in general be inferred from underlying LTI properties, i.e. frozen-time deductions do not generally ensure that LPV modeling characteristics will be preserved with rapid parameter variations [33]. Hence, no formal proofs of convergence between the NM and our RCM LPV model may be given via this engineering practice. Nonetheless, and bestowed by its simplicity and its previous track record [11], [10], this methodology will be pursued in the sequel of this paper.

Formalizing the first objective we obtain the following aggregated optimization problems

Problem 1 For a given frequency range $\Delta_\omega = [\omega_1, \omega_2]$, find

$$(\hat{A}_0, \hat{B}_0, \hat{L}, \hat{R}, \hat{\Pi}) = \cdots$$

$$\arg\inf_{\left(A_0 \in \mathbb{R}^{n_x \times n_x}, B_0 \in \mathbb{R}^{n_x \times n_u}, L_s \in \mathbb{R}^{n_x \times n_x}, R_s \in \mathbb{R}^{n_x \times n_u}, \eta_{s_i} \in \mathbb{R}\right)} \sum_{i=1}^N \left\|\bar{G}_i(s) - G_i(s)\right\|_{\Delta_\omega}$$

with $\hat{L} := [\hat{L}_s]_{S n_x \times n_x}$, $\hat{R} := [\hat{R}_s]_{S n_x \times n_u}$, and $\hat{\Pi} := [\hat{\eta}_{s_i}]_{S \times N}$

Problem 2 Find

$$(\hat{T}, \hat{Z}, \hat{\Xi}) = ...$$

$$\arg\inf_{\left(T_w \in \mathbb{R}^{n_x \times n_x}, Z_w \in \mathbb{R}^{n_x \times n_u}, \zeta_{w_i} \in \mathbb{R}\right)} \sum_{i=1}^N \left\|\mathbf{d}_i - \sum_{w=1}^W \zeta_{w_i}\big(T_w \mathbf{x}_i + Z_w \mathbf{u}_i\big)\right\|_2$$

with $\hat{T} := [\hat{T}_w]_{Wn_x \times n_x}$, $\hat{Z} := [\hat{Z}_w]_{Wn_x \times n_u}$, and $\hat{\Xi} := [\hat{\zeta}_{w_i}]_{W \times N}$

Our second objective is set within the quasi-LPV framework. We aim to find a relationship between the here-above computed scheduling parameters, $\hat{\Pi} := [\hat{\eta}_{s_i}]_{S \times N}$ and $\hat{\Xi} := [\hat{\zeta}_{w_i}]_{W \times N}$, and the on-line measurable states and control inputs. In other words, we need to find smooth and CT nonlinear mappings, $g(\cdot)$ and $h(\cdot)$, s.t. $\boldsymbol{\eta}(t) = g(\mathbf{x}(t), \mathbf{u}(t))$, $\zeta(t) = h(\mathbf{x}(t), \mathbf{u}(t))$.

For physically-intuitive plants, one may select the required states and inputs, based on engineering judgment, and derive these mappings through popular curve-fitting methods. For non-transparent systems, exhibiting significant dependences among variables, one may consider formal/systematic approaches such as principal component analysis, statistical analysis, fuzzy tools, or Neural Networks (NNs). Now, NNs have found a wide range of applications in control theory. Indeed, 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 IO relationship arbitrarily well, provided there are enough neurons in the hidden layer [34], [35]. However, despite their powerful features, NNs have only seen limited usage in the LPV field, except for the derivation of quasi-LPV SS models from NNs representations [36], [37], [38]. Hence, we propose here to base the $g(\cdot)$ and $h(\cdot)$ modeling on NNs.

III. A Solution to Problem 1

While several approaches, such as [39], [40], may potentially be considered, no efficient solution is currently known. Consequently, a sub-optimal procedure will be outlined. We opt for the three-stage philosophy introduced in [10], consisting in obtaining first $(\hat{A}_0, \hat{B}_0)$, then $(\hat{L}, \hat{R})$, and finally $\hat{\Pi}$. However, contrary to [10] with the use of Least-Squares (LS) methods, we proceed here with new approaches in the $\mathcal{H}_\infty$ framework.

A. Preliminaries

This section introduces the Kalman-Yakubovich-Popov (KYP) Lemma [41], with spectral mask constraints.

Lemma 1: Let real scalars $\omega_1 \le \omega_2$, $\omega_c = (\omega_1 + \omega_2)/2$, and a TF $G(s) := \left[\begin{array}{c|c} A & B \\ \hline C & D \end{array}\right]$ be given, then the following statements are equivalent.

(i) $\forall \gamma > 0$, $\lambda(A) \subset \mathbb{C}^- \cup \mathbb{C}^+$, $\|G\|^2_{\Delta_\omega} < \gamma^2$ (6)

(ii) $\exists (P, Q)$, $P = P^*$, $Q > 0$, $L(P, Q) + \Theta < 0$
$$L(P, Q) = \dots$$
$$\left[\begin{array}{cc} A & B \\ I & 0 \end{array}\right]^* \left[\begin{array}{cc} -Q & P + j\omega_c Q \\ P - j\omega_c Q & -\omega_1\omega_2 Q \end{array}\right] \left[\begin{array}{cc} A & B \\ I & 0 \end{array}\right]$$
$$\Theta = \left[\begin{array}{cc} C & D \\ 0 & I \end{array}\right]^* \left[\begin{array}{cc} I & 0 \\ 0 & -\gamma^2 I \end{array}\right] \left[\begin{array}{cc} C & D \\ 0 & I \end{array}\right] \quad (7)$$

(iii) $\exists (F, K)$
$\forall l \in \{1, 2\}$ $M_l(F, K) + \Theta < 0$
$$M_l(F, K) = \mathrm{He}\left(\left[\begin{array}{c} F \\ K \end{array}\right]\left[\begin{array}{cc} I & -j\omega_l I \end{array}\right]\left[\begin{array}{cc} A & B \\ I & 0 \end{array}\right]\right)$$
With Θ given in (7)   (8)

Proof: From (6), expand $(C(sI - A)^{-1}B + D)^*(C(sI - A)^{-1}B + D) - \gamma^2 I < 0$ as partitioned matrices, and invoke the KYP Lemmas with spectral mask constraints, from [42] and [43], to prove (ii) and (iii) respectively. Note that for the case where $\lambda(A) \subset \mathbb{C}^-$ we need to add the stability constraint $P > 0$ in (ii), and for the case where $\lambda(A) \subset \mathbb{C}^0$ it is standard practice to perturb A by $-\epsilon I$, with $0 < \epsilon \ll 1$ [44].

Both approaches (ii) and (iii) of Lemma 1 will be used in this paper. Now let n be the number of decision variables, and m the number of rows of LMIs, then comparing (ii) and (iii) 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 $O(n^2 m^{2.5} + m^{3.5})$ for SeDuMi [4], and in $O(n^3 m)$ for MATLAB LMI-lab [45], the former approach is more efficient for large problems, however, the latter has the advantage that, for fixed F and K, it is also affine in the problem's A and B matrices.

B. Determination of $(\hat{A}_0, \hat{B}_0)$

The goal is to find the optimal $\hat{G}_0(s) := \left[\begin{array}{c|c} \hat{A}_0 & \hat{B}_0 \\ \hline C & D \end{array}\right]$ s.t.

$$(\hat{A}_0, \hat{B}_0) = \arg \inf_{(A_0 \in \mathbb{R}^{n_x \times n_x}, B_0 \in \mathbb{R}^{n_x \times n_u})} \sum_{i=1}^{N} \|\bar{G}_i(s) - G_0(s)\|_{\Delta_\omega} \quad (9)$$

We propose a non-optimal procedure that restricts the search space to set $\mathcal{Z}_{Lin}^N$. For each model $\bar{G}_i(s)$, we get the following mean, standard-deviation, and extrema

$$\forall i \in \{1, ..., N\} \quad \mu_i = (1/N) \sum_{j=1}^{N} \|\bar{G}_i(s) - \bar{G}_j(s)\|_{\Delta_\omega}$$
$$s_i = \left[(1/N) \sum_{j=1}^{N} \left(\|\bar{G}_i(s) - \bar{G}_j(s)\|_{\Delta_\omega} - \mu_i\right)^2\right]^{1/2}$$
$$\underline{\mu} = \min_i \mu_i, \quad \bar{\mu} = \max_i \mu_i, \quad \underline{s} = \min_i s_i, \quad \bar{s} = \max_i s_i \quad (10)$$

where $\|\bar{G}_i(s) - \bar{G}_j(s)\|_{\Delta_\omega}$ is obtained by minimizing the bound γ defined in (6). This is computationally done by minimizing γ subject to the LMIs of (7). Subsequently, the optimal model $\hat{G}_0(s)$ is designated as $\hat{G}_0(s) := \bar{G}_{\hat{i}}(s)$, with the index $\hat{i}$ resulting from a, readily solved, mean vs. standard-deviation minimization problem

$$\hat{i} = \arg \min_{i \in \{1, ..., N\}} \left(\left([\mu_i - \underline{\mu}]/[\bar{\mu} - \underline{\mu}]\right)^2 \rho + \left([s_i - \underline{s}]/[\bar{s} - \underline{s}]\right)^2\right) \quad (11)$$

with ρ a user-defined weighting parameter.

C. Determination of $\hat{L} := [\hat{L}_s]_{Sn_x \times n_x}$, $\hat{R} := [\hat{R}_s]_{Sn_x \times n_u}$

To determine these basis functions we will anchor our approach on Singular Value Decompositions (SVD). Within the realm of LPV modeling, the use of the SVD machinery has been independently pioneered by several researchers [13], [46], [10], and our approach will parallel the results

of [13], [10]. Let $\Upsilon = [1...1]$ be a row vector of length N. We define the following Ω and Φ matrices

From set $\mathcal{Z}_{Lin}^N$ define $\quad \Omega = \left[\begin{array}{ccc} \mathrm{vec}(\bar{A}_1) & ,..., & \mathrm{vec}(\bar{A}_N) \\ \mathrm{vec}(\bar{B}_1) & ,..., & \mathrm{vec}(\bar{B}_N) \end{array} \right]$

From (9) $\quad \Phi = \left[\begin{array}{c} \mathrm{vec}(\hat{A}_0) \\ \mathrm{vec}(\hat{B}_0) \end{array} \right] \otimes \Upsilon$ $\qquad$ (12)

Next, obtain a SVD decomposition of the form $\Omega - \Phi = U\Sigma V^*$. Now $U_{1..S}$, with $S \leq n_x(n_x + n_u)$, contains the first S columns of the left singular vectors matrix U. Then $\hat{L}$ and $\hat{R}$ are recovered from the matricization of each column of $U_{1..S}$. Note that, for high model fidelity in OL, one could keep a maximum number of basis functions S, whereas for controller design, one could cope with fewer ones, and perform a posteriori stability tests along the lines of [47].

D. Determination of $\hat{\Pi} := [\hat{\eta}_{s_i}]_{S \times N}$

The procedure has a two-stage modus operandi: (i) an initialization stage, followed by (ii) a nonlinear-based refinement stage. The first stage computes reasonable guess values for $\hat{\eta}_{s_i}$, from the sum minimization of the $\mathcal{L}_2$-induced gains of two static operators

$$\forall i \in \{1, ..., N\} \quad [\hat{\eta}_{s_i}]_{S \times 1} = \arg \min_{\left(\eta_{s_i} \in \mathbb{R}\right)} \left\| X_A \right\|_2 + \left\| X_B \right\|_2 \quad (13)$$

with $X_A = \bar{A}_i - (\hat{A}_0 + \sum\limits_{s=1}^{S} \eta_{s_i} \hat{L}_s)$ and $X_B = \bar{B}_i - (\hat{B}_0 + \sum\limits_{s=1}^{S} \eta_{s_i} \hat{R}_s)$

This is readily recast into a standard LMI problem

$$\forall i \in \{1, ..., N\} \quad \text{minimize} \quad \gamma_A + \gamma_B$$
$$\text{subject to} \quad \gamma_A > 0, \quad \gamma_B > 0$$
$$\left[\begin{array}{cc} \gamma_A I & \star \\ X_A & I \end{array} \right] > 0 \quad \left[\begin{array}{cc} \gamma_B I & \star \\ X_B & I \end{array} \right] > 0 \quad (14)$$

The second stage uses the previously computed $(\hat{A}_0, \hat{B}_0, \hat{L}, \hat{R})$ and the start values for $[\hat{\eta}_{s_i}]_{S \times 1}$ to solve

$$\forall i \in \{1, ..., N\} \quad [\hat{\eta}_{s_i}]_{S \times 1} = \cdots$$
$$\arg \inf_{\left(\eta_{s_i} \in \mathbb{R}\right)} \| \bar{G}_i(s) - G_i(s) \|_{\Delta_\omega} \quad (15)$$

This is a non-convex problem. To compute $\| \bar{G}_i(s) - G_i(s) \|_{\Delta_\omega}$ we call now upon (8). As stated earlier, the advantage of (8) is that it is convex in either (F, K) or (A, B) matrices. These (A, B) matrices are given by: $\bar{G}_i(s) - G_i(s) :=$

$$\left[\begin{array}{c|c} A & B \\ \hline C & D \end{array} \right] = \left[\begin{array}{ccc|c} \bar{A}_i & 0 & & \bar{B}_i \\ 0 & \hat{A}_0 + \sum\limits_{s=1}^{S} \eta_{s_i} \hat{L}_s & & \hat{B}_0 + \sum\limits_{s=1}^{S} \eta_{s_i} \hat{R}_s \\ \hline I & -I & & 0 \end{array} \right].$$ Our

proposed approach is a simple two-step iterative LMI search, in spirit reminiscent of μ $D - K$-iteration synthesis [48]. The procedure reads as follows: partition F and K as $F = \left[\begin{array}{cc} F_{11} & F_{12} \\ F_{21} & F_{22} \end{array} \right]$ and $K = \left[\begin{array}{cc} K_{11} & K_{12} \\ K_{21} & K_{22} \end{array} \right]$ and start with the initial value $\hat{\Pi}$ obtained from (14). Now from (8), (i) minimize γ with respect to (F, K), (ii) keep $(F_{12}, F_{22}, K_{12}, K_{22})$ from step (i) and minimize γ with respect to $(\hat{\Pi}, F_{11}, F_{21}, K_{11}, K_{21})$, (iii) repeat (i) and (ii) until

convergence or maximum iteration reached.

Remark 7: Aside from $D - K$-iteration, similar heuristics appear to work well in practice, such as model order reduction [49], LPV-LFR controller with parameter-dependent scalings [50], or gain scheduled controller with inexact scheduling parameters [51]. Analogously to D-K iteration convergence [52], [53] - for which convergence towards a global optimum, or even a local one, is not guaranteed - the above iterative method does not inherit any convergence certificates, however in practice convergence has been achieved within a few iterations.

IV. A Solution to Problem 2

It is precisely this feature that endows our LPV model with its global nature. Suppose we can find scheduling parameters ζ_{w_i} and basis functions (T_w, Z_w) s.t.

$$\forall i \in \{1, ..., N\} \quad \mathbf{d}_i \left[\begin{array}{c} \mathbf{x}_i \\ \mathbf{u}_i \end{array} \right]^\dagger \approx \left[\begin{array}{cc} \sum\limits_{w=1}^{W} \zeta_{w_i} T_w & \sum\limits_{w=1}^{W} \zeta_{w_i} Z_w \end{array} \right] \quad (16)$$

with $[\cdot]^\dagger$ the left inverse. Then by right-multiplying both sides with $[\mathbf{x}_i^T \ \mathbf{u}_i^T]^T$ we recover $\mathbf{d}_i \approx \sum\limits_{w=1}^{W} \zeta_{w_i}(T_w \mathbf{x}_i + Z_w \mathbf{u}_i)$. To determine the basis functions, we will again use SVDs. First, we construct the matrices Λ_i and Ψ s.t.

$$\Lambda_i = \mathbf{d}_i \left[\begin{array}{c} \mathbf{x}_i \\ \mathbf{u}_i \end{array} \right]^\dagger \quad \Psi = \left[\begin{array}{ccc} \mathrm{vec}(\Lambda_1) & ,..., & \mathrm{vec}(\Lambda_N) \end{array} \right] \quad (17)$$

Then, obtain a SVD decomposition of the form $\Psi = U\Sigma V^*$. Now $U_{1..W}$, with $W \leq n_x(n_x + n_u)$, contains the first W columns of the left singular vectors matrix U. Then $\hat{T}$ and $\hat{Z}$ are recovered from the matricization of each column of $U_{1..W}$. Next, we use LS to compute $\hat{\Xi} := [\hat{\zeta}_{w_i}]_{W \times N}$

$$\forall i \in \{1, ..., N\} \quad [\hat{\zeta}_{w_i}]_{W \times 1} = \cdots$$
$$\arg \min_{\left(\zeta_{w_i} \in \mathbb{R}\right)} \left\| \mathrm{vec}(\Lambda_i) - U_{1..W} \left[\zeta_{1_i}, ..., \zeta_{W_i} \right]^T \right\|_2^2 \quad (18)$$

The solution reduces to $\forall i \in \{1, ..., N\} \ [\hat{\zeta}_{w_i}]_{W \times 1} = U_{1..W}^T \mathrm{vec}(\Lambda_i)$ since $U_{1..W}$ is an orthogonal matrix.

V. Quasi-LPV Framework

The aim is now to find suitable representations for $g(\cdot)$ and $h(\cdot)$, s.t. $\eta(t) = g(\mathbf{x}(t), \mathbf{u}(t))$, $\zeta(t) = h(\mathbf{x}(t), \mathbf{u}(t))$, by illustrating the applicability of a two-layer feedforward NNs, the first being sigmoid and the second linear, with l neurons (l large enough)

$$\eta(t) = g(\mathbf{x}(t), \mathbf{u}(t)) = C_\eta . \mathbf{s}_\eta(t) \quad \zeta(t) = h(\mathbf{x}(t), \mathbf{u}(t)) = C_\zeta . \mathbf{s}_\zeta(t)$$
$$\mathbf{s}_\eta(t) = W_{o_\eta} . \kappa(W_{x_\eta} \boldsymbol{x}(t) + W_{u_\eta} \boldsymbol{u}(t) + W_{b_\eta})$$
$$\mathbf{s}_\zeta(t) = W_{o_\zeta} . \kappa(W_{x_\zeta} \boldsymbol{x}(t) + W_{u_\zeta} \boldsymbol{u}(t) + W_{b_\zeta}) \quad (19)$$

where $W_{o_\eta} \in \mathbb{R}^{S \times l}$, $W_{o_\zeta} \in \mathbb{R}^{W \times l}$ and $W_{x_\eta} \in \mathbb{R}^{l \times n_x}$, $W_{u_\eta} \in \mathbb{R}^{l \times n_u}$, $W_{x_\zeta} \in \mathbb{R}^{l \times n_x}$, $W_{u_\zeta} \in \mathbb{R}^{l \times n_u}$ contain the output and hidden layer weights respectively. Further, $W_{b_\eta} \in \mathbb{R}^l$, $W_{b_\zeta} \in \mathbb{R}^l$ contain the sets of biases in the hidden layer, $C_\eta \in \mathbb{R}^{S \times S}$, $C_\zeta \in \mathbb{R}^{W \times W}$ contain the output linear maps, and $\kappa(\cdot)$ is the activation function, taken as a continuous, diagonal, differentiable, and bounded static sigmoid nonlinearity.

VI. Numerical Experiment

We compare here the effectiveness of our proposed LPV modeling strategy with the original nonlinear model, and with the combined SVD least-squares based LPV model [10], using the most familiar nonlinear physical system that exhibits harmonic motion, i.e. the simple, driven and damped, pointmass pendulum, for which the rotational motion is given by

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

$$\text{with} \quad \alpha(u(t)) := u(t)^2 \sin u(t) \quad (20)$$

with $[x_1 \ x_2]^T = [\theta \ \dot{\theta}]^T$ the states, θ the rotation angle, u the input torque, $\beta = \sqrt{g/L}$ the angular frequency, g the acceleration due to gravity, L the pendulum length, b a measure of the dissipative force, with values: ($\beta = \sqrt{9.8/9}, b = 2$), and $\alpha(\cdot)$ a fictional nonlinearity with the intent of increasing the Nonlinear Model (NM) generality. To derive the LPV model, we excite the pendulum, from its rest position, with a 10 s long sine-sweep $u(t) = A \sin(2\pi.f.t)$, $A = 2$, $f \in [0.05 - 1]$ Hz, sampled with a period $T_s = 0.05$ s, resulting in 201 data points. Since we also want to compare our method with that of [10], we use a wide-band Δ_ω with $[\omega_1, \omega_2] = [0.005 - 20]$ Hz, as [10] does not handle spectral mask constraints.

For frequency-domain comparisons, we define the following cost $J_\infty := \frac{1}{N} \sum_{i=1}^{N} \left\| \bar{G}_i(s) - G_i(s) \right\|_\infty$, with N the data length. Further, the LMI problems are solved using YALMIP with the SeDuMi solver. The optimal model $\hat{G}_0(s)$, obtained according to (11) with $\rho = 1$, was found to be $G_{59}(s)$. Solving *Problem 1*, by keeping all bases ($S = 2$), with S defined in Section III-C, we get $J_\infty = 0$ with (14). For $S = 1$ we get $J_\infty = 10.17$ with (14), and $J_\infty = 2.05$ after refinement (15).

Next, we use fresh data sets, namely sine-inputs of various amplitudes and frequencies, and compare time-domain outputs in $l_2[0, \infty)$, with the following two metrics: Best-FiT (BFT) $:= 100\%.\frac{1}{n_x} \sum_{k=1}^{n_x} \max\left(1 - \frac{\left\| \mathbf{x}_k - \tilde{\mathbf{x}}_k \right\|_2}{\left\| \mathbf{x}_k - \text{mean}(\mathbf{x}_k) \right\|_2}, 0\right)$, and Variance-Accounted-For (VAF) $:= 100\%.\frac{1}{n_x} \sum_{k=1}^{n_x} \max\left(1 - \frac{\text{var}(\mathbf{x}_k - \tilde{\mathbf{x}}_k)}{\text{var}(\mathbf{x}_k)}, 0\right)$ with $\mathbf{x}_k \in \mathbb{R}^N$ the kth NM output, $\tilde{\mathbf{x}}_k \in \mathbb{R}^N$ its LPV equivalent. Now since [10] does not provide a solution for *Problem 2*, nor for mappings $g(\cdot)$ and $h(\cdot)$ of Section V, we extend it with our proposed approaches. For both LPV models we use $W = 3$ in *Problem 2*, keeping all bases in Section IV, and further a 5-neurons feedforward network, with the hyperbolic tangent activation TF in the hidden layer, and backpropagation training for the weights and biases. The outcomes are presented in Table I and II. For both models, the accuracy diminishes as the input amplitude is shifted away from the value used for estimation. For this numerical experiment, we see that, except for BFT($A = 1.5, f = 1Hz$),

TABLE I

Time Resp. For Our Model. Left value is BFT (%), Right value is VAF (%)

Input Amplitude	Input Frequency (Hz)			
	0.25	0.5	0.75	1
0.5	65 95	48 93	37 91	37 87
1	74 96	71 95	53 95	35 92
1.5	86 98	82 98	68 97	39 94
2	86 99	81 99	92 100	85 99

TABLE II

Time Resp. For Model [10]. Left value is BFT (%), Right value is VAF (%)

Input Amplitude	Input Frequency (Hz)			
	0.25	0.5	0.75	1
0.5	5 68	9 61	5 44	2 39
1	32 84	27 84	26 82	26 80
1.5	41 85	75 94	68 93	62 91
2	40 85	71 96	80 96	76 94

our model consistently outperforms the approach of [10]. For illustration, case ($A = 2, f = 0.25Hz$) is shown in Fig. 1.

VII. Conclusion

We have presented a novel and comprehensive affine quasi-LPV modeling method. For high model fidelity in Open-Loop, one could keep a maximum number of basis functions, whereas for controller design, one could cope with fewer ones. Our approach does not incorporate any information on parameter time-derivatives, hence significant enhancements could potentially be obtained in this area. Our preliminary encouraging results invite further applications of the here-presented approach.

Appendix

Instead of the KYP-based formalism, and by reverting to a standard weighted $\mathcal{H}_\infty$ norm minimization, with the obvious increase in model order and complexity, we can provide an alternative to (15). Now, for the specific case of having the control-matrix independent of the time-varying scheduling parameter, and if we consider retaining all bases S, then the problem of finding $\hat{\Pi} := [\hat{\eta}_{s_i}]_{S \times N}$ becomes convex, and an optimal solution is then computable by this two-step procedure

$$\forall i \in \{1, ..., N\} \quad [\hat{\eta}_{s_i}]_{S \times 1} = \cdots$$
$$\arg \min_{\left(\eta_{s_i} \in \mathbb{R}\right)} \|W_f(\bar{G}_i(s) - G_i(s))\|_\infty \quad (21)$$

with W_f a given, strictly-proper, bandpass filter, centered at Δ_ω. We have now the following result

Lemma 2: Let $W_f(s) := \left[\begin{array}{c|c} A_f & B_f \\ \hline C_f & 0 \end{array}\right]$, $\bar{G}_i(s) :=$ $\left[\begin{array}{c|c} \bar{A}_i & \bar{B}_i \\ \hline I & 0 \end{array}\right]$, $G_i(s) := \left[\begin{array}{c|c} \hat{A}_0 + \sum_{s=1}^{S} \eta_{s_i} \hat{L}_s & \hat{B}_0 \\ \hline I & 0 \end{array}\right]$, be given, with matrices of appropriate size. Let $W_f(\bar{G}_i(s) - G_i(s)) :=$

$$\left[\begin{array}{ccc|c} A_f & B_f & -B_f & 0 \\ 0 & \bar{A}_i & 0 & \bar{B}_i \\ 0 & 0 & \hat{A}_0 + \sum\limits_{s=1}^{S} \eta_{s_i}\hat{L}_s & \hat{B}_0 \\ \hline C_f & 0 & 0 & 0 \end{array}\right] = \left[\begin{array}{cc|c} A_{11} & A_{12} & B_{11} \\ 0 & A_{22} & \hat{B}_0 \\ \hline C_{11} & 0 & 0 \end{array}\right],$$

with $A_{11} = \begin{bmatrix} A_f & B_f \\ 0 & \bar{A}_i \end{bmatrix}$, $A_{12} = \begin{bmatrix} -B_f \\ 0 \end{bmatrix}$, $B_{11} = \begin{bmatrix} 0 \\ \bar{B}_i \end{bmatrix}$,

$C_{11} = \begin{bmatrix} C_f & 0 \end{bmatrix}$, and $A_{22} = \hat{A}_0 + \sum\limits_{s=1}^{S} \eta_{s_i}\hat{L}_s$, then the following statements are equivalent.

(i) $\forall \gamma > 0$, $W_f \in \mathcal{RH}_\infty$, $(\bar{G}_i(s) - G_i(s)) \in \mathcal{RL}_\infty$, $\|W_f(\bar{G}_i(s) - G_i(s))\|_\infty^2 < \gamma^2$ (22)

(ii) $\exists (P, Q)$, $P = P^T$, $Q = Q^T = P^{-1}$

$P = \begin{bmatrix} P_{11} & P_{12} \\ P_{12}^T & P_{22} \end{bmatrix}$, $Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{12}^T & Q_{22} \end{bmatrix}$

$\Gamma(X_\eta, P_{11}, P_{12}, Q_{11}, Q_{12}) :=$

$$\left[\begin{array}{cc} \mathrm{Sym}(A_{11}Q_{11} + A_{12}Q_{12}^T) & \star \\ A_{11}^T + X_\eta & \mathrm{Sym}(P_{11}A_{11}) \\ B_{11}^T & B_{11}^T P_{11} + \hat{B}_0^T P_{12}^T \\ C_{11}Q_{11} & C_{11} \end{array}\right.$$

$$\left.\begin{array}{cc} \star & \star \\ \star & \star \\ -\gamma^2 I & \star \\ 0 & -\gamma^2 I \end{array}\right] < 0 \qquad (23)$$

with $X_\eta = P_{11}A_{11}Q_{11} + P_{11}A_{12}Q_{12}^T + P_{12}A_{22}Q_{12}^T$

Proof: The proof is a straightforward application of the Bounded Real Lemma (BRL) [54] in LMI form [55], with further (i) a congruence transformation [56] with $\mathrm{diag}(J, I, I)$, $J = \begin{bmatrix} Q_{11} & I \\ Q_{12}^T & O \end{bmatrix}$, and (ii) a change of variable given by X_η. For stable systems, i.e. $(\bar{G}_i(s) - G_i(s)) \in \mathcal{RH}_\infty$, one has to add the condition $J^T P J = \begin{bmatrix} Q_{11} & I \\ I & P_{11} \end{bmatrix} > 0$

Now, (21) reduces to a two-step approach. First, solve

$\forall i \in \{1, ..., N\}$ minimize γ

subject to $\gamma > 0$, and the LMIs of *Lemma* 2   (24)

Then compute $A_{22} = P_{12}^\dagger (X_\eta - P_{11}A_{11}Q_{11} - P_{11}A_{12}Q_{12}^T) Q_{12}^{T\dagger}$. Note that P_{12} and Q_{12}^T are skinny and fat matrices, hence, by virtue of the respective left and right inverse, A_{22} is well-defined. Next, we have the minimization of the $\mathcal{L}_2$-induced gain of the static operator $X_{A_{22}} = A_{22} - (\hat{A}_0 + \sum\limits_{s=1}^{S} \eta_{s_i}\hat{L}_s)$

$\forall i \in \{1, ..., N\}$ $[\hat{\eta}_{s_i}]_{S \times 1} = \arg \min\limits_{(\eta_{s_i} \in \mathbb{R})} \|X_{A_{22}}\|_2$ (25)

which is solved as in (14).

Fig. 1. Model outputs for $u(t) = A\sin(2\pi.f.t)$ with $A = 2$, $f = 0.25$ Hz

References

[1] S. Boyd, L. ElGhaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*. Philadelphia: Soc. for Industrial and Applied Mathematics (SIAM), 1994.

[2] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge: Cambridge University Press, 2004.

[3] Y. Nesterov and A. Nemirovski, *Interior Point Polynomial Methods in Convex Programming*. Soc. for Industrial & Applied Math, 1994.

[4] 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.
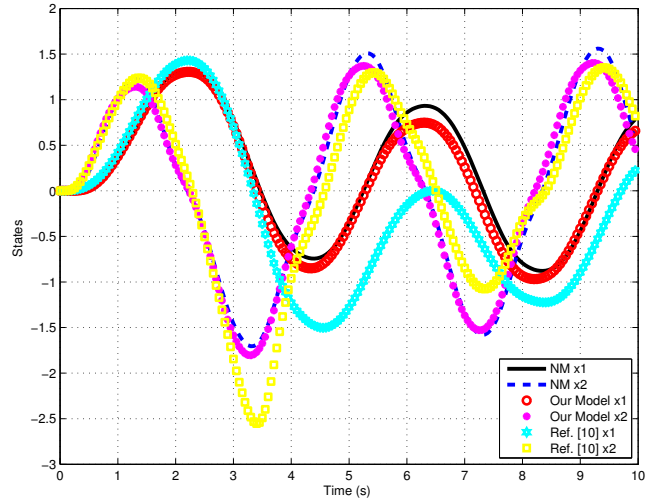
[5] W. Rugh, "Analytical framework for gain scheduling," *IEEE Control Systems Magazine*, vol. 11, pp. 74–84, 1991.

[6] C. Reboulet and C. Champetier, "A new method for linearizing non-linear systems: the pseudolinearization," *Int. J. of Control*, vol. 40, no. 4, pp. 631–638, 1984.

[7] L. R. Hunt, R. Su, and G. Meyer, "Global transformations of nonlinear systems," *IEEE Trans. on Automatic Control*, vol. 28, no. 1, pp. 24–31, 1983.

[8] J. Shamma and M. Athans, "Gain scheduling: Potential hazards and possible remedies," *IEEE Control Systems Magazine*, vol. 12, pp. 101–107, 1992.

[9] 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.

[10] 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.

[11] 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.

[12] P. Baranyi, "Tensor-product model transformation as a way to lmi based controller design," *IEEE Trans. on Industrial Electronics*, vol. 51, no. 2, pp. 387–400, 2004.

[13] Z. Petres, "Polytopic decomposition of linear parameter-varying models by tensor-product model transformation," Ph.D. dissertation, Budapest University of Technology and Economics, 2006.

[14] A. Kwiatkowski, M. T. Boll, and H. Werner, "Automated generation and assessment of affine lpv models," in *IEEE Conf. on Decision and Control*, 2006.

[15] R. Toth, *Identification and Modeling of Linear Parameter-Varying Systems. Lecture Notes in Control and Information Sciences, Vol. 403*. Heidelberg: Springer, 2010.

[16] B. Paijmans, "Interpolating gain-scheduling control for mechatronic systems with parameter-dependent dynamics," Ph.D. dissertation, Katholieke Universiteit Leuven, 2007.

[17] J. DeCaigny, J. F. Camino, and J. Swevers, "Interpolation-based modeling of mimo lpv systems," *IEEE Trans. on Control Systems Technology*, vol. 19, no. 1, pp. 46–63, 2011.

[18] 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.

[19] D. Petersson and J. Lofberg, "Optimization based lpv-approximation of multi-model systems," in *Europ. Control Conf.*, 2009.

[20] W. Tan, "Applications of linear parameter-varying control theory," Master Thesis, University of California at Berkeley, Tech. Rep., 1997.

[21] J. Y. Shin, "Quasi-linear parameter varying representation of general

aircraft dynamics over non-trim region," NASA Langley Research Center, Tech. Rep. CR-2007-213926, 2007.

[22] 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.

[23] P. M. J. Van den Hof and R. J. P. Schrama, "Identification and control - closed-loop issues," *Automatica*, vol. 31, no. 12, pp. 1751–1770, 1995.

[24] 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.

[25] R. Toth, H. Abbas, and H. Werner, "On the state-space realization of lpv input-output models: Practical approaches," *IEEE Trans. on Control Systems Technology*, vol. 20, no. 1, pp. 139–153, 2012.

[26] J. M. Biannic, "Commande robuste des systemes a parametres variables," Ph.D. dissertation, ENSAE, 1996.

[27] V. Laurain, R. Toth, M. Gilson, , and H. Garnier, "Direct identification of continuous-time lpv input/output models," *IET Control Theory & Applications*, vol. 5, no. 7, pp. 878–888, 2011.

[28] G. Goodwin, S. Graebe, and M. Salgado, *Control System Design*. Prentice Hall, 2000.

[29] K. J. Hunt and T. A. Johansen, "Design and analysis of gain-scheduled local controller networks," *Int. J. of Control*, vol. 66, no. 5, pp. 619–651, 1997.

[30] T. A. Johansen, K. J. Hunt, P. J. Gawthrop, and H. Fritz, "Off-equilibrium linearization and design of gain scheduled control with application to vehicle speed control," *Control Engineering Practice*, vol. 6, pp. 167–180, 1998.

[31] P. Korba and H. Werner, "Counter-example to ad-hoc off-equilibrium linearisation methods in gain-scheduling," in *IEEE Conf. on Decision and Control*, 2001.

[32] C. S. Mehendale and K. M. Grigoriadis, "A new approach to lpv gain-scheduling design and implementation," in *IEEE Conf. on Decision and Control*, 2004.

[33] J. S. Shamma and M. Athans, "Guaranteed properties of gain scheduled control for linear parameter-varying plants," *Automatica*, vol. 27, no. 3, pp. 559–564, 1991.

[34] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359–366, 1989.

[35] M. Norgaard, O. Ravn, N. Poulsen, and L. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*. London: Springer-Verlag, 2000.

[36] J. D. Bendtsen and K. Trangbaek, "Robust quasi-lpv control based on neural statespace models," *IEEE Trans. on Neural Networks*, vol. 13, no. 2, pp. 355–368, 2002.

[37] H. Abbas and H. Werner, "Polytopic quasi-lpv models based on neural state-space models and application to air charge control of a si engine," in *IFAC 17th World Congress*, 2008.

[38] N. Lachhab, H. Abbas, and H. Werner, "A neural-network based technique for modelling and lpv control of an arm-driven inverted pendulum," in *IEEE Conf. on Decision and Control*, 2008.

[39] T. Zhou, H. G. M. Dotsch, and P. M. J. Van den Hof, "Model set determination and its application to the control of compact disc players," *Europ. J. of Control*, vol. 4, p. 99115, 1998.

[40] X. Bombois, H. Hjalmarsson, and G. Scorletti, "Identification for robust h_2 deconvolution filtering," *Automatica*, vol. 46, no. 3, pp. 577–584, 2010.

[41] A. Rantzer, "On the kalman-yakubovich-popov lemma," *Systems & Control Letters*, vol. 28, pp. 7–10, 1996.

[42] T. Iwasaki, G. Meinsma, and M. Fu, "Generalized s-procedure and finite frequency kyp lemma," *Math. Prob. Eng.*, vol. 6, pp. 305–320, 1997.

[43] M. R. Graham, M. C. de Oliveira, and R. A. de Callafon, "An alternative kalman-yakubovich-popov lemma and some extensions," *Automatica*, vol. 45, no. 6, pp. 1489–1496, 2009.

[44] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design, 2nd Ed.* Great Britain: Wiley-Interscience, 2005.

[45] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *LMI Control Toolbox Users Guide*. The Mathworks Partner Series, 1995.

[46] A. Kwiatkowski and H. Werner, "Pca-based parameter set mappings for lpv models with fewer parameters and less overbounding," *IEEE Trans. on Control Systems Technology*, vol. 16, no. 4, pp. 781–788, 2008.

[47] S. M. Hashemi, H. S. Abbas, and H. Werner, "Low-complexity linear parameter-varying modeling and control of a robotic manipulator," *Control Engineering Practice*, vol. 20, no. 3, pp. 248–257, 2012.

[48] J. C. Doyle, "Structured uncertainty in control system design," in *IEEE Conf. on Decision and Control*, 1985.

[49] A. Helmersson, "Model reduction using lmis," in *IEEE Conf. on Decision and Control*, 1994.

[50] 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.

[51] M. Sato and D. Peaucelle, *Gain-Scheduled Output-Feedback Controllers with Good Implementability and Robustness. In: J. Mohammadpour and C. W. Scherer (Eds.), Control of Linear Parameter Varying Systems with Applications.* Heidelberg: Springer, 2012.

[52] 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.

[53] M. A. Rotea and T. Iwasaki, "An alternative to the d-k iteration ?" in *Am. Control Conf.*, 1994.

[54] C. Scherer, "The riccati inequality and state-space h_∞-optimal control," Ph.D. dissertation, Universitat Wurzburg, 1990.

[55] 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.

[56] C. W. Scherer, P. Gahinet, and M. Chilali, "Multiobjective output-feedback control via lmi optimization," *IEEE Trans. on Automatic Control*, vol. 42, no. 7, pp. 896–911, 1997.

9