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 lead 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 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 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 leaves, 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 differentialization 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 like the one 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. $M^T$, $M^*$, $M'$ denote the transpose, the complex-conjugate transpose, and the Moore-Penrose inverse of a real or complex matrix $M$, whereas $\text{He}(M)$ (resp. $\text{Sym}(M)$) is shorthand for $M + M^T$ (resp. $M + M^\dagger$). We use $\ast$ as an ellipsis for terms that are induced by symmetry. Matrix inequalities are considered in the sense of L"owner. Further $\lambda(M)$ denotes the zeros of the characteristic polynomial det$(sI - M) = 0$. $L_\infty$ is the Lebesgue normed space s.t. $\|G\|_\infty := \text{ess sup}_s \sigma(G(j\omega)) < \infty$, with $\sigma(G)$ the largest singular value of matrix $G$). Similarly, $\mathcal{H}_\infty \subset L_\infty$ is the Hardy normed space s.t. $\|G\|_{\mathcal{H}_\infty} := \sup_{s \in \mathbb{R} \setminus \{0\}} \sigma(G(s))$. For $\omega_1 < \omega_2$, $\Delta_\omega = [\omega_1, \omega_2]$, we use $\|G\|_{\mathcal{H}_\omega} := \sup_{\omega \in \Delta_\omega} \sigma(G(j\omega))$. $\mathcal{R}_{L_\infty}$ (resp. $\mathcal{RH}_{\infty}$) represent the subspace of real rational Transfer Functions (TFs) in $L_\infty$ (resp. $\mathcal{H}_\infty$).

II. PROBLEM STATEMENT

We suppose now that the noise-free NM is given\footnote{Since the nonlinear model (NM) is given, we assume that we have access to the full state vector.} by

$$\forall t \geq 0 \quad x(t) = f(x(t), u(t)) \quad y(t) = x(t)$$

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 $x(t) \in P_x \subset \mathbb{R}^n_x$ the plant state, $y(t) \in P_y \subset \mathbb{R}^n_y$ the plant output, $u(t) \in P_u \subset \mathbb{R}^n_u$ the control input, $t$ the time variable, and $(P_x, P_y, P_u)$ 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 $S(\theta(t))$

$$S(\theta(t)) := \begin{cases} \forall t \geq 0 & \bar{x}(t) = A_0x(t) + B_0u(t) + \ldots \\ \sum_{r=1} \theta_r(t)(A_r x(t) + B_r u(t)) \end{cases}$$

with $\theta(t) := [\theta_r(t)]_{r=1}^{\infty} \in \mathbb{R}^R$, the non-stationary scheduling parameters, not known a priori, but on-line measurable and defined on the compact set $P_{\theta}$, 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. $\theta(x(t), u(t))$, the quasi-LPV prefix is added. Note that in this case, and in the occurrence of an unstable plant, such quasi parameterization is only licit if one assumes an upper bound on the sizes of $P_x$ and $P_y$, 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 Fractional 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 \((y(t), u(t), \theta(t))\). These DT signals, subsumed with the sampling period \(T_s > 0\), are denoted \(u(t) = u(iT_s), i \in \mathbb{Z}\), as illustrated here for the input signal \(u(t)\). We assume that a simulated IO signal sequence \(Z^N := \{u(iT_s), y(iT_s)\}_{i=1}^N\), 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 \(T_s = 1/\max\{f_b,x, f_b,u\}\) (resp. \(T_s = 1/\max\{f_b,x\}\)) in case of LPV system (resp. quasi-LPV), with \(f_b,u\) and \(f_b,x\) the bandwidth of the input and scheduling-signal parameters 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 \(\theta := \theta(t)\), 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 \(Z^N\), we create a set of triplet elements \(Z^N_{Lin} := \{(\bar{A}_i, \bar{B}_i, \bar{d}_i)\}_{i=1}^N\)

\[
\bar{A}_i = \frac{\partial f(x, u)}{\partial x} \bigg|_{(x,u)} \quad \bar{B}_i = \frac{\partial f(x, u)}{\partial u} \bigg|_{(x,u)} \\
\bar{d}_i = f(x, u) - \bar{A}_i x - \bar{B}_i u_i
\]

for some scheduling parameters \(\eta(t) = [\eta_i(t)]_{i=1}^N \in \mathbb{R}^\xi\), \(\zeta(t) = [\zeta_w(t)]_{w=1}^W \in \mathbb{R}^\omega\), and matrices \((L_x, R_x, T_w, Z_w)\), of appropriate sizes, s.t. \(\sum_{r=1}^R \theta_i(t) A_r = \sum_{s=1}^S \eta_i(t) L_s + \sum_{w=1}^W \zeta_w(t) T_w\), and \(\sum_{r=1}^R \theta_i(t) B_r = \sum_{s=1}^S \eta_i(t) R_s + \sum_{w=1}^W \zeta_w(t) Z_w\).

For each frozen-time trajectory \(\eta_i := \eta_i(t), \zeta_w := \zeta_w(t), s = 1, ..., S, w = 1, ..., W\), associated with the time indexes \(i = 1, ..., N\) of set \(Z^N\), we create both (i) a sequence of CT LTI TFs \(G_i(s) := \begin{bmatrix} A_0 + \sum_{s=1}^S \eta_i(t) L_s & B_0 + \sum_{s=1}^S \eta_i(t) R_s \\
C & D \end{bmatrix}\)

with matrices \(C = I, D = 0\) of appropriate size, and (ii) a sequence of vectors \(\sum_{w=1}^W \zeta_w(t) T_w x_i + Z_w u_i\). 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 \((A_0, B_0, L, R, T, Z)\) and \(\hat{A}\) have same size (resp. \((B_0, R, X, \hat{A})\) and \(\hat{B}\)).

**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 \(Z^N_{Lin}\) to identify the unknown scheduling parameters \(\eta(t)\), \(\zeta(t)\), and basis functions \((A_0, B_0, L, R, T, Z)\).

**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

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

\[
\arg \inf_{(A_0, B_0, L, R, T, Z) \in \mathbb{R}^{n x n}, \mathbb{R}^{n x m}, \mathbb{R}^{n x n}, \mathbb{R}^{m x m}, \mathbb{R}^{m x n}} \sum_{i=1}^N \left\| \bar{G}_i(s) - G_i(s) \right\|_{\Delta_\omega}
\]

with \(\hat{L} := \left[ \bar{L}_1, \mathbb{R}^{n x n}, \hat{R} := [\bar{R}_1, \mathbb{R}^{n x n}, \end{bmatrix} \right) \), and \(\hat{\Pi} := [\hat{\eta}_i]_{i=1}^N\).

**Problem 2** Find

\[
(\hat{\Pi}, \hat{Z}, \hat{\Xi}) = \cdots
\]

\[
\arg \inf_{(T, \mathbb{R}^{n x n}, \mathbb{R}^{n x m}, \mathbb{R}^{m x n})} \sum_{i=1}^N \left\| \bar{d}_i - \sum_{w=1}^W \zeta_w(t) (T_w x_i + Z_w u_i) \right\|_2
\]
with \( \hat{T} := [\hat{T}_w]_{w,n \times n}, \hat{Z} := [\hat{Z}_w]_{w,n \times n}, \) and \( \hat{\xi} := [\hat{\xi}_w]_{w \times N} \).  

Our second objective is set within the quasi-LPV framework. We aim to find a relationship between the here-above computed scheduling parameters, \( \hat{T} := [\hat{T}_w]_{w,n \times n} \) and \( \hat{\xi} := [\hat{\xi}_w]_{w \times N} \), 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. \( \eta(t) = g(x(t), u(t)) \), \( \zeta(t) = h(x(t), u(t)) \).

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 been seen limited usage in the LPV field, except for the derivation of quasi-LPV SS models from NN 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 \( \hat{\xi} \). 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 \( \omega_1 \leq \omega_2, \omega_c = (\omega_1 + \omega_2)/2 \), and a TF \( G(s) := \frac{A(s)}{B(s)} \) be given, then the following statements are equivalent.

(i) \( \forall \gamma > 0, \lambda(A) \subset \mathbb{C}^-, \|G\|_{\lambda_{\infty}}^2 < \gamma^2 \) \hspace{1cm} (6)
(ii) \( \exists (P, Q), P = P^*, Q > 0, L(P, Q) + \Theta < 0 \)
\[ L(P, Q) = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} -Q & P + j\omega_c Q \\ P - j\omega_c Q & -\omega_c \end{bmatrix} \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Theta \end{bmatrix} \]
\[ \Theta = \begin{bmatrix} I & 0 \\ 0 & \Theta \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Theta \end{bmatrix} \]
\( \Theta < 0 \) \hspace{1cm} (7)
(iii) \( \exists (F, K) \)
\[ \forall i \in [1,2] \quad M_i(F, K) + \Theta < 0 \]
\[ M_i(F, K) = \text{He} \left( \begin{bmatrix} F & I \\ K & I \end{bmatrix} \begin{bmatrix} A & B \\ I & 0 \end{bmatrix} \right) \]

With \( \Theta \) given in (7)

Proof: From (6), expand \( (C(sI - A)^{-1}B + D)(C(sI - A)^{-1}B + D) - \gamma^2I < 0 \) 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 LMI’s, then comparing (ii) and (iii) shows that, while both have similar \( m \), they differ in terms of \( n \), i.e. \( n_1^2 + n_2 \) versus \( n_1 + n_2 \), respectively. Since the asymptotic computational complexity, or flop cost, of SDP solvers is in \( O(n^2m^{5/2} + m^{1.5}) \) for SeDuMi [4], and in \( O(n^3m) \) 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 \( \hat{G}_0(s) := \begin{bmatrix} \hat{A}_0 \\ \hat{B}_0 \end{bmatrix} \) s.t.

\[ (\hat{A}_0, \hat{B}_0) = \arg \inf_{(A_0 \in \mathbb{R}^n \times n, B_0 \in \mathbb{R}^n \times n)} \sum_{i=1}^{N} \|G_i(s) - \hat{G}_0(s)\|_{\lambda_{\infty}} \]

We propose a non-optimal procedure that restricts the search space to set \( \mathbb{R}_{2m}^N \). For each model \( \tilde{G}_i(s) \), we get the following mean, standard-deviation, and extrema.

\[ \forall i \in [1,\ldots,N] \quad \mu_i = (1/N) \sum_{j=1}^{N} \|G_i(s) - \tilde{G}_i(s)\|_{\lambda_{\infty}} \]
\[ s_i = \left[ 1/(2N) \sum_{j=1}^{N} \left( \|G_i(s) - \tilde{G}_i(s)\|_{\lambda_{\infty}} - \mu_i \right)^2 \right]^{1/2} \]
\[ \mu = \min_i \mu_i, \tilde{\mu} = \max_i \mu_i, \tilde{s} = \min_i s_i, \tilde{\tilde{s}} = \max_i s_i \]

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

\[ \hat{i} = \arg \min_{i \in [1,\ldots,N]} \left( \|\mu_i - \tilde{\mu}\|^2 + \left( \frac{s_i - \tilde{s}}{\tilde{s} - \tilde{s}} \right) \right) \]

with \( \rho \) a user-defined weighting parameter.

C. Determination of \( \hat{L} := [\hat{L}_i]_{n \times n}, \hat{R} := [\hat{R}_i]_{n \times n} \)

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 \( \mathcal{Y} = [1...1] \) be a row vector of length \( N \). We define the following \( \Omega \) and \( \Phi \) matrices

From set \( Z_{LM}^N \) define \( \Omega = \begin{bmatrix} \text{vec}(A_1) & \ldots & \text{vec}(A_N) \\ \text{vec}(B_1) & \ldots & \text{vec}(B_N) \end{bmatrix} \)

From (9) \( \Phi = \begin{bmatrix} \text{vec}(A_0) \\ \text{vec}(B_0) \end{bmatrix} \otimes \mathcal{Y} \) (12)

Next, obtain a SVD decomposition of the form \( \Omega - \Phi = U \Sigma \Psi^T \). Now \( U_{1,S} \), with \( S \leq n_t(n_s + n_a) \), 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 \( \hat{1} := [\hat{h}_s]_{S \times N} \)

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 \( \hat{h}_s \), from the sum minimization of the \( L_2 \)-induced gains of two static operators

\[
\forall i \in \{1,...,N\} \; \hat{h}_i = \arg \min_{(\eta_i \in \mathbb{R})} \|X_A\|_2 + \|X_B\|_2
\] (13)

with \( X_A = \bar{A}_s - (A_0 + \sum_{s=1}^S \eta_s L_s) \) and \( X_B = \bar{B}_s - (B_0 + \sum_{s=1}^S \eta_s R_s) \)

This is readily recast into a standard LMI problem

\[
\forall i \in \{1,...,N\} \; \text{minimize} \; \gamma_A + \gamma_B \\
\text{subject to} \; \gamma_A > 0, \; \gamma_B > 0 \\
\begin{bmatrix} \gamma_A I & \ast \\ X_A & I \end{bmatrix} > 0, \begin{bmatrix} \gamma_B I & \ast \\ X_B & I \end{bmatrix} > 0
\] (14)

The second stage uses the previously computed \( \hat{A}_0, \hat{B}_0, \hat{L}, \hat{R} \) and the start values for \( \hat{h}_s \) \( s \times 1 \) to solve

\[
\forall i \in \{1,...,N\} \; [\hat{h}_s]_{s \times 1} = \ldots \\
\text{arg} \min \| \hat{G}(s) - G_i(s) \|_{\Lambda_s}
\] (15)

This is a non-convex problem. To compute \( \| \hat{G}(s) - G_i(s) \|_{\Lambda_s} \) we call now up (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: \( G_i(s) - G_i(s) := \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} \hat{A}_1 & \hat{0} \\ \hat{0} & \hat{A}_0 + \sum_{s=1}^S \eta_s L_s \\ \hat{B}_1 & \hat{0} \\ \hat{0} & \hat{B}_0 + \sum_{s=1}^S \eta_s R_s \end{bmatrix} \). Our proposed approach is a simple two-stage iterative LMI search, in spirit reminiscent of \( \mu \) \( D - K \)-iteration synthesis [48]. The procedure reads as follows: partition \( F \) and \( K \) as \( F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \) and \( K = \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \) and start with the initial value \( \hat{1} \) obtained from (14). Now from (8), (i) minimize \( c \) with respect to \( (F,K) \), (ii) keep \( (F_{12}, F_{22}, K_{12}, K_{22}) \) from step (i) and minimize \( c \) with respect to \( (\hat{1}, F_{11}, F_{21}, K_{11}, K_{21}) \), (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 \( \zeta_{w_i} \) and basis functions \( (T_w, Z_w) \) s.t.

\[
\forall i \in \{1,...,N\} \; d_i \begin{bmatrix} x_i \\ u_i \end{bmatrix} = \begin{bmatrix} \sum_{w=1}^W \zeta_{w}T_w \sum_{w=1}^W \zeta_{w}Z_w \end{bmatrix}
\] (16)

with \([-1]^T\) the left inverse. Then by right-multiplying both sides with \([x_i^T \; u_i^T]^T\) we recover \( d_i \approx \sum_{w=1}^W \zeta_{w}(T_w x_i + Z_w u_i) \).

To determine the basis functions, we will again use SVDs.

First, we construct the matrices \( \Lambda_i \) and \( \Psi \) s.t.

\[
\Lambda_i = d_i \begin{bmatrix} x_i \\ u_i \end{bmatrix}^T \Psi = \begin{bmatrix} \text{vec}(A_i) \\ \ldots \\ \text{vec}(A_N) \end{bmatrix}
\] (17)

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

\[
\forall i \in \{1,...,N\} \; [\hat{\zeta}_{w}]_{W \times 1} = \ldots \\
\text{arg} \min_{(\hat{\zeta}_w \in \mathbb{R}^+)} \| \text{vec}(A_i) - U_{1,w} [\hat{\zeta}_1, ..., \hat{\zeta}_w] \|^2_2
\] (18)

The solution reduces to \( \forall i \in \{1,...,N\} \; [\hat{\zeta}_w]_{W \times 1} = U_{1,w}^T \text{vec}(A_i) \) 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. \( \eta(t) = g(x(t), u(t)) \), \( \zeta(t) = h(x(t), u(t)) \), by illustrating the applicability of a two-layer feedforward NNs, the first being sigmoid and the second linear, with \( l \) neurons (large enough)

\[
\eta(t) = g(x(t), u(t)) = C_{\eta} \cdot s_{\eta}(t) \\
\zeta(t) = h(x(t), u(t)) = C_{\zeta} \cdot s_{\zeta}(t)
\]

\[
s_{\eta}(t) = W_{\eta,\ast} \cdot C_{\eta} \cdot x(t) + W_{\eta} \cdot u(t) + W_{b_{\eta}}
\]

\[
s_{\zeta}(t) = W_{\zeta,\ast} \cdot C_{\zeta} \cdot x(t) + W_{\zeta} \cdot u(t) + W_{b_{\zeta}}
\] (19)

where \( W_{\eta,\ast} \in \mathbb{R}^{S \times l} \), \( W_{\zeta,\ast} \in \mathbb{R}^{W \times l} \) and \( W_{\eta} \in \mathbb{R}^{l \times n} \), \( W_{\zeta} \in \mathbb{R}^{l \times n} \), \( W_{b_{\eta}} \in \mathbb{R}^{l \times 1} \), \( W_{b_{\zeta}} \in \mathbb{R}^{l \times 1} \) contain the output and hidden layer weights respectively. Further, \( W_{\eta} \in \mathbb{R}^{l \times l} \), \( W_{b_{\eta}} \in \mathbb{R}^{l \times l} \) 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

\[
d\begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} x_2(t) \\ -b x_2(t) - \beta^2 \sin x_1(t) \end{bmatrix} + \frac{0}{\alpha(u(t))} u^2 \sin u(t) \tag{20}
\]

with \([x_1, x_2]^T = [\theta, \dot{\theta}]^T\) the states, \(\theta\) 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: \(\beta = \sqrt{g/L}, b = 2\), 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 sine-sweep \(u(t) = A \sin(2\pi f t), A = 2, f \in [0.05 - 1] \text{Hz}, \) sampled with a period \(T_s = 0.05 \text{s},\) resulting in 201 data points. Since we also want to compare our method with that of [10], we use a wide-band \(\Delta_w\) with \([\omega_1, \omega_2] = [0.005 - 20] \text{Hz},\) as [10] does not handle spectral mask constraints.

For frequency-domain comparisons, we define the following cost \(J_{\infty} := \frac{1}{N} \sum_{n=1}^{N} \| G_n(s) - G(s) \|_{\infty} \), 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 \(G_{00}(s)\). 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 \(L_1[0, \infty)\), with the following two metrics: Best-Fit (BFT) (\(\text{BFT} := 100\% \times \frac{1}{N} \sum_{k=1}^{N} \max \left(1 - \frac{\|x_k - \hat{x}_k\|_2}{\|x_k - \text{mean}(x_k)\|_2}, 0 \right)\), and Variance-Accounted-For (VAF) (\(\text{VAF} := 100\% \times \frac{1}{N} \sum_{k=1}^{N} \max \left(1 - \frac{\|x_k - \text{var}(x_k)\|_2}{\|x_k - \text{mean}(x_k)\|_2}, 0 \right)\)) with \(x_k \in \mathbb{R}^N\) the \(k\)th NM output, \(\hat{x}_k \in \mathbb{R}^N\) 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\((A = 1.5, f = 1 \text{Hz}),\) our model consistently outperforms the approach of [10]. For illustration, case \((A = 2, f = 0.25 \text{Hz})\) 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 \(\Pi := [\hat{h}_{ts}]_{S \times N}\) becomes convex, and an optimal solution is then computable by this two-step procedure

\[
\forall i \in \{1, \ldots, N\}, \arg \min_{(n, \hat{n})} \| W_f(G_i(s) - G_i(s)) \|_{\infty} = \cdots
\tag{21}
\]

with \(W_f\) a given, strictly-proper, bandpass filter, centered at \(\Delta_w\). We have now the following result

\[
\text{Lemma 2: Let } W_f(s) := \begin{bmatrix} A_f & B_f \\ C_f & D_f \end{bmatrix}, \\
G_i(s) := \begin{bmatrix} \tilde{A}_i & \tilde{B}_i \\ \tilde{I} & 0 \end{bmatrix} \mathbb{1}, \\
\text{be given, with matrices of appropriate size. Let } W_f(G_i(s) - G_i(s)) :=
\]

<p>| TABLE I |
| Time Resp. For Our Model, Left value is BFT (%), Right value is VAF (%) |</p>
<table>
<thead>
<tr>
<th>Input Amplitude</th>
<th>Input Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>65 95</td>
</tr>
<tr>
<td>1</td>
<td>74 96</td>
</tr>
<tr>
<td>1.5</td>
<td>86 98</td>
</tr>
<tr>
<td>2</td>
<td>86 99</td>
</tr>
</tbody>
</table>

<p>| TABLE II |
| Time Resp. For Model [10], Left value is BFT (%), Right value is VAF (%) |</p>
<table>
<thead>
<tr>
<th>Input Amplitude</th>
<th>Input Frequency (Hz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>5 68</td>
</tr>
<tr>
<td>1</td>
<td>32 84</td>
</tr>
<tr>
<td>1.5</td>
<td>41 85</td>
</tr>
<tr>
<td>2</td>
<td>40 85</td>
</tr>
</tbody>
</table>
Note that $P$ statements are equivalent.

$$ \begin{bmatrix} A_f & B_f & -B_f & 0 \\ 0 & \hat{A}_f & 0 & \hat{B}_i \\ 0 & 0 & \hat{A}_0 + \sum_{i=1}^s \eta_i \hat{L}_i & \hat{B}_0 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} & B_{11} & 0 \\ 0 & A_{22} & B_{12} & 0 \\ C_{11} & 0 & 0 & 0 \end{bmatrix}, $$

with $A_{11} = \begin{bmatrix} A_f & B_f & 0 & 0 \end{bmatrix}$, $A_{12} = \begin{bmatrix} -B_f & 0 & 0 \end{bmatrix}$, $B_{11} = \begin{bmatrix} 0 & \hat{B}_i \end{bmatrix}$, $C_{11} = [C_f 0]$, and $A_{22} = \hat{A}_0 + \sum_{i=1}^s \eta_i \hat{L}_i$, then the following statements are equivalent.

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

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

$$ P = \begin{bmatrix} P_{11} & P_{12} \\ P_{T12} & P_{T12} \end{bmatrix}, \quad Q = \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{T12} & Q_{T12} \end{bmatrix} $$

$$ \Gamma(X, \eta, P_{11}, P_{12}, Q_{11}, Q_{12}) := \begin{bmatrix} \text{Sym}(A_{11}Q_{11} + A_{12}Q_{12}) & * \\ \text{Sym}(A_{11}Q_{11}) & \text{Sym}(P_{11}A_{11}) \\ \text{Sym}(P_{11}A_{11}) & \text{Sym}(A_{21}Q_{11}) \\ \text{Sym}(A_{21}Q_{11}) & \text{Sym}(Q_{11}) \\ \text{Sym}(Q_{11}) & \text{Sym}(Q_{11}) \end{bmatrix} $$

$$ \eta \begin{bmatrix} 0 \\ \gamma^2 I \end{bmatrix} 0 = 0, \quad \gamma^2 I < 0 \quad (23) $$

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

**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 diag($I, I, I$), $J = \begin{bmatrix} Q_{11} & I \\ Q_{T12} & 0 \end{bmatrix}$, and (ii) a change of variable by $X_\eta$.

For stable systems, i.e. $(G_i(s) - G_i(s)) \in \mathcal{RH}_\infty, \eta$ one has to add the condition $J^T P J \geq 0$

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

$$ \forall i \in \{1, \ldots, N\} \quad \min_{\gamma > 0} \gamma $$

subject to $\Gamma(X, \eta) \in \mathcal{RH}_\infty$, and the LMIs of Lemma 2 (24)

Then compute $A_{22} = P_{12}(X_\eta - P_{11}A_{11}Q_{11} - P_{11}A_{12}Q_{T12} + P_{12}A_{22}Q_{T12})$

Note that $P_{12}$ and $Q_{T12}$ 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 $L_2$-induced gain of the static operator $X_{A_{22}} = A_{22} - (\hat{A}_0 + \sum_{i=1}^s \eta_i \hat{L}_i)$

$$ \forall i \in \{1, \ldots, N\} \quad [\bar{\eta}_i]_{s \times 1} = \arg \min_{\eta \in \mathbb{R}} \left\| X_{A_{22}} \right\|_2 $$

which is solved as in (14).

**References**


