Improved Embedding of Nonlinear Systems in Linear Parameter-Varying Models with Polynomial Dependence

Arash Sadeghzadeh and Roland Tóth

***Abstract*— In this paper, the problem of automated generation of *linear parameter-varying* (LPV) state-space models to embed the dynamic behavior of nonlinear systems is addressed. The LPV model depends polynomially on an introduced set of scheduling variables. This set comprises linear combinations of residuals, the differences between nonlinear functions in the original model description with their polynomial approximations, in addition to some of the states. The salient feature of the proposed method is that LPV model complexity, LPV model accuracy, and LPV model conservativeness are jointly considered through the embedding procedure. To quantitatively evaluate the LPV model accuracy and conservativeness, two cost functions are introduced based on which the model complexity can be adjusted. Numerical studies reveal that the presented method is capable to deliver more accurate and less conservative LPV models in comparison with the available approaches. The effectiveness of the proposed method is shown in an empirical case study where the dynamic behavior of a 3DOF gyroscope is embedded into an LPV model. Exploiting the obtained LPV model, a gain-scheduled state feedback controller is designed and validated on the gyroscope, clearly demonstrating the applicability of the presented method.**

***Index Terms*— Linear parameter-varying (LPV) system, nonlinear system, LPV embedding, multivariate polynomial regression, principle component analysis.**

I. Introduction

The *linear parameter-varying* (LPV) framework has already received significant attention to tackle the control synthesis problem of nonlinear (NL) and time-varying (TV) systems via convex methods, which has resulted in widespread practical applications ranging from aerospace to process control [1]. By introducing so-called scheduling variables, one can express NL and/or TV systems in an LPV forms. This facilitates the extension of many powerful methods for LTI control design and analysis to NL and/or TV systems. However, to this date, an exhaustive method to address LPV embedding of NL system on a systematic basis has not been presented yet; most of the available methods are ad-hoc approaches with inherent lack of generality and/or have shortcomings in addressing predominant issues regarding the constitution of the embedding.

Although LPV system identification methods have substantially matured, see e.g. [2]–[8] and references therein, conversion methods to obtain an LPV representation directly from an existing NL and/or TV model is still in a state of development. Since for many practical applications, NL and/or TV high-fidelity models based on first-principles are initially available, conversion methods to attain LPV description for such systems are of great importance. In this context, the so called local methods refer to those techniques in which the NL description is initially linearized at several operating points, and then the obtained linear models are interpolated into an LPV description, see [7]–[11] for an overview. The underlying drawback of the local methods is that closed-loop stability and performance cannot be guaranteed for the whole operating range by the controllers designed based on the resulting LPV models, especially for those systems with relatively fast dynamic variation. This is the consequence of that the time propagation of the scheduling variables is not taken into account in the LPV modeling procedure. On the other hand, there are global methods which generate LPV models that preserve the dynamic behavior of the NL system [12]–[19]. In [12], an LPV model is derived via a state transformation of the original nonlinear system. A velocity-based framework is employed in [13] based on which the plant dynamics at non-equilibrium operating points are also incorporated into the LPV model, which enables the transient behavior of the nonlinear system to be captured. A function substitution approach is proposed in [14] to obtain an LPV model. An algorithm for automated generation of affine LPV representations is presented in [15]. The method exploits heuristic quality measures to evaluate different possible LPV representations. Inspired by the feedback linearization theory, a procedure to convert a NL state-space representation into an LPV representation having state minimal observable canonical form is proposed in [16]. Affine LPV embedding problem of nonlinear systems represented by *linear fractional representation* (LFR) with a nonlinear feedback block is considered in [17]. The proposed LPV

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement nr. 714663) and from the Ministry of Innovation and Technology NRDI Office within the framework of the Autonomous Systems National Laboratory Program.

A. Sadeghzadeh is with the Faculty of Electrical Engineering, Shahid Beheshti University, Tehran, Iran (e-mail: a_sadeghzadeh@sbu.ac.ir)

R. Tóth is with the Control Systems Group, Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands (e-mail: r.toth@tue.nl). He is also associated with the Systems and Control Laboratory, Institute for Computer Science and Control, Budapest, Hungary.

embedding method in [18] is constructed based on minimizing the projection of the nonlinearities onto directions which are deleterious for performance. Taking advantage of the *principle component analysis* (PCA), automated generation of affine LPV state-space models to embed the dynamical behavior of the NL systems is investigated in [19], inspired by the method in [20].

Three main factors that characterize the quality of embedding procedures are (i) LPV model complexity which can be assessed by the number of scheduling variables and functional complexity on the scheduling variables, e.g. affine dependency, polynomial dependency, etc., (ii) LPV model accuracy which is an index representing the discrepancy between the dynamical behavior of the original NL model and the embedded LPV one, (iii) LPV model conservativeness which is a measure of expressing the difference of admissible operating regions of the original NL model and the embedded LPV model. Ideally, an embedding procedure should be flexible enough so that the LPV model complexity can be determined based on the tolerable conservativeness and the desired accuracy. In [19] and [20], accuracy indices are introduced to tackle the trade-off problem between the number of scheduling variables and the accuracy. Nevertheless, the impact of the number of scheduling variables on the conservativeness has been ignored, and the scheduling variables are chosen considering just the accuracy. By comparing the volumes of operating regions of the NL description and the LPV model, a conservativeness measure is proposed in [15]; however, it is just an intuitive measure which cannot exactly represent the conservativeness, as the authors have also mentioned. To the best of our knowledge, none of the existing methods in the literature consider jointly the complexity, the accuracy, and the conservativeness in the embedding procedure.

Motivated by the above given observations, in this article, we aim to develop a systematic method for the embedding of the NL models into LPV ones while the trade-off between complexity, accuracy, and conservativeness of the desired LPV model are simultaneously taken into account through the embedding procedure. The underlying idea of the proposed method is to approximate the nonlinear functions in the NL description using multivariate polynomial regression to obtain an LPV model which depends polynomially on some of the states, considered later as the scheduling variables. To cope with the approximation errors, and enhance LPV model accuracy, the residuals of the approximation are also put into the scheduling variable set. However, some of the residuals may be correlated, thus, the residuals can be projected onto a lower-dimensional space using PCA, which directly leads to less number of scheduling variables while accuracy of the LPV model is preserved. One of the main features of the obtained LPV model is that it depends polynomially on the scheduling variables, which yields a more accurate model with a lower number of scheduling variables, compared to those models that depend affinely on the scheduling variables. As it becomes clear later, there exist connections between the degrees of polynomials used for the approximation and the number of scheduling variables (LPV model complexity) with the LPV model accuracy and conservativeness. This implies that by choosing appropriate LPV model complexity, one can balance the LPV model accuracy and conservativeness. Two indices are introduced to evaluate the accuracy and the conservativeness, which enable us to properly choose the LPV model complexity. Our contribution with respect to existing approaches, is to propose an LPV embedding method in which the complexity, the accuracy, and the conservativeness are all under control through the embedding procedure. Contrary to the available methods, LPV model complexity, LPV model accuracy, and LPV model conservativeness are *jointly* considered through the embedding procedure. This way, one would be able to regulate the trade-off between the complexity, accuracy, and conservativeness. This is the main contribution of the paper. A numerical comparison with some available methods shows that the presented method results in more accurate and less conservative LPV models for a prescribed number of scheduling variables. Furthermore, to reveal the capabilities of the presented method, the proposed method in this paper is applied on an empirical case study where the nonlinear dynamic equations of a 3DOF control moment gyroscope are embedded into an LPV model. Exploiting the obtained LPV model, a gain-scheduled state feedback controller is designed and empirically validated.

Notation: $L_{i,j} \in \mathbb{R}$ denotes the elements of matrix $L \in \mathbb{R}^{m\times n}$, i.e. $L \coloneqq [L_{i,j}]_{m\times n}$. $\overrightarrow{\Gamma}$ refers to row-wise vectorization of L:

$$\overrightarrow{L} \coloneqq \begin{bmatrix} L_{1,1}\cdots L_{1,n} & L_{2,1}\cdots L_{2,n} & \cdots & L_{m,1}\cdots L_{m,n} \end{bmatrix}.$$

Let us define $\mathbb{I}_{s_1}^{s_2} \coloneqq \{i \in \mathbb{Z} \mid s_1 \le i \le s_2\}$ and

$$\{L_{i,j}\}_{i\in\mathbb{I}_1^n, j\in\mathbb{I}_1^m} \coloneqq \{L_{1,1},\cdots,L_{1,m},\cdots,L_{n,1},\cdots,L_{n,m}\}.$$

The notation $\|L\|_F \coloneqq \sqrt{\sum_{i=1}^m \sum_{j=1}^n |L_{i,j}|^2}$ corresponds to the Frobenius norm. A vector-valued function $\alpha : \mathbb{R} \to \mathbb{R}^k$ is called class $\mathcal{C}_1^k$, if α is continuously differentiable on $\mathbb{R}$. If $P(s)$ is a real rational transfer function of s, $P^\star(s) = P^\top(-s)$. The determinant of a matrix X is denoted by $\det(X)$. Let $\mathfrak{P}_p[\mathbb{R}^{n_\rho}]$ denote the set of all real n_ρ dimensional monomials of degree at most p given by

$$\gamma(\rho(t)) \coloneqq \rho_1^{c_1}(t)\rho_2^{c_2}(t)\cdots\rho_{n_\rho}^{c_{n_\rho}}(t)$$

where $c_1,\cdots,c_{n_\rho} \in \mathbb{N}\cup 0$ and $c_1 + c_2 + \cdots + c_{n_\rho} \le p$.

II. Problem Description

Consider the following continuous-time nonlinear system

$$\dot{x}(t) = F_1(x(t)) + G_1(x(t))u(t), \tag{1a}$$
$$y(t) = F_2(x(t)) + G_2(x(t))u(t), \tag{1b}$$

where $x : \mathbb{R} \to \mathbb{X} \subset \mathbb{R}^{n_x}$ is the state vector, $u : \mathbb{R} \to \mathbb{U} \subset \mathbb{R}^{n_u}$ is the input, and $y : \mathbb{R} \to \mathbb{Y} \subset \mathbb{R}^{n_y}$ is the output of the system. $\mathbb{X}$ is assumed to be a compact polyhedron with known vertices

[1] An initial version of the developed method in this paper has been presented in [21]. The current paper seriously extends [21] in terms of sparse polynomial approximations, accuracy and conservativeness indices, introduced to address the trade-off between complexity, accuracy, and conservativeness in the embedding procedure. Furthermore, an empirical case study is provided in this paper to demonstrate the capability of the proposed method in practice.

containing the origin. $\mathbb{U}$ and $\mathbb{Y}$ are considered to be open sets containing the origin. $F_1 : \mathbb{X} \to \mathbb{R}^{n_x}$, $F_2 : \mathbb{X} \to \mathbb{R}^{n_y}$, $G_1 : \mathbb{X} \to \mathbb{R}^{n_x \times n_u}$, $G_2 : \mathbb{X} \to \mathbb{R}^{n_y \times n_u}$ are bounded and smooth static real-valued nonlinear functions of $x \in \mathbb{X}$ such that their first order partial derivatives exist on $\mathbb{X}$. The solution set of (1) is denoted by

$$\mathbb{B}_{\mathrm{NL}} = \{(y,x,u) \in (\mathbb{Y} \times \mathbb{X} \times \mathbb{U})^{\mathbb{R}_0^+} \mid (y,x,u) \text{ s.t. (1) holds } \forall t \in \mathbb{R}_0^+ \text{ with } x \in \mathcal{C}_1^{n_x}\},$$

which represents the so-called behavior of the nonlinear system (1). It is assumed that all solutions in the solution set $\mathbb{B}_{\mathrm{NL}}$ are forward-complete. Let us define

$$\begin{aligned} F(x(t)) &\coloneqq \begin{bmatrix} F_1^\top(x(t)) & F_2^\top(x(t)) \end{bmatrix}^\top \\ &= \begin{bmatrix} f_1(x(t)) & f_2(x(t)) \cdots & f_{n_x+n_y}(x(t)) \end{bmatrix}^\top, \end{aligned}$$

$$\begin{aligned} G(x(t)) &\coloneqq \begin{bmatrix} G_1^\top(x(t)) & G_2^\top(x(t)) \end{bmatrix}^\top \\ &= \begin{bmatrix} g_{1,1}(x(t)) & \cdots & g_{1,n_u}(x(t)) \\ \vdots & \vdots & \vdots \\ g_{(n_x+n_y),1}(x(t)) & \cdots & g_{(n_x+n_y),n_u}(x(t)) \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} x(t) &\coloneqq \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_{n_x}(t) \end{bmatrix}^\top, \\ u(t) &\coloneqq \begin{bmatrix} u_1(t) & u_2(t) & \cdots & u_{n_u}(t) \end{bmatrix}^\top. \end{aligned}$$

Assumption 1: All of the continuously-differentiable functions $f_i(x(t))$, $i = 1, \cdots, n_x + n_y$ satisfy that $f_i(0) = 0$. Note that $f_i(0) = 0$ can be easily achieved by state and input transformation in (1a) and by output transformation in (1b).

Systems in the form of (1) are commonly referred to as control-affine nonlinear systems, often encountered in many practical applications in process control and mechatronics [22], [23]. The ultimate goal considered in this paper is to embed the nonlinear system (1) in a linear parameter-varying representation as follows:

$$\dot{x}(t) = A(\lambda(t))x(t) + B(\lambda(t))u(t), \tag{2a}$$
$$y(t) = C(\lambda(t))x(t) + D(\lambda(t))u(t), \tag{2b}$$

such that $\lambda(t) \coloneqq \mu(x(t))$ where the scheduling map $\mu : \mathbb{X} \to \Lambda \subseteq \mathbb{R}^{n_\lambda}$ is automatically constructed, and the resulting scheduling variable $\lambda(t) = \begin{bmatrix} \lambda_1(t) & \lambda_2(t) & \cdots & \lambda_{n_\lambda}(t) \end{bmatrix}^\top$ belongs to a polyhedron Υ ($\Lambda \subseteq \Upsilon$) defined by

$$\underline{\lambda}_i \leq \lambda_i(t) \leq \overline{\lambda}_i, \quad i = 1, \cdots, n_\lambda \tag{3}$$

with $\underline{\lambda}_i, \overline{\lambda}_i \in \mathbb{R}$ which are determined in the procedure. For reasons that will be clear later, the scheduling variable $\lambda(t)$ is assumed to be partitioned as $\lambda(t) \coloneqq [\rho^\top(t) \ \theta^\top(t)]^\top \in \Upsilon \subseteq \mathbb{R}^{n_\lambda}$ where

$$\begin{aligned} \rho(t) &\coloneqq [\rho_1(t) \cdots \rho_{n_\rho}(t)]^\top \in \Psi \subseteq \mathbb{R}^{n_\rho}, \\ \theta(t) &\coloneqq [\theta_1(t) \cdots \theta_{n_\theta}(t)]^\top \in \Theta \subseteq \mathbb{R}^{n_\theta}. \end{aligned}$$

Thus, we have $\Upsilon = \Psi \times \Theta$. It is assumed that $\Psi \coloneqq S\mathbb{X}$, where the full row rank matrix S is a selector matrix in which one of the elements of each row is one and the other elements are zero, enabling us to determine which of the state-space variables to be included directly in the scheduling. Furthermore, it is supposed that the state-space matrices have the following dependency on the scheduling variables (affine dependency on $\theta(t)$ and polynomial dependency on $\rho(t)$):

$$M(\rho(t), \theta(t)) \coloneqq \sum_{i=1}^{n_\theta} \theta_i(t) M_i + \sum_{j=1}^{n_\gamma} \gamma_j(\rho(t)) M_{n_\theta + j}. \tag{4}$$

Here, $\gamma_j(\rho(t))$ are n_ρ dimensional monomials belonging to a selected set of n_γ basis functions, namely $\gamma_j(\rho(t)) \in \Gamma_M$, where

$$\Gamma_M \coloneqq \{\gamma_j\}_{j=1}^{n_\gamma} \subset \mathfrak{P}_p[\mathbb{R}^{n_\rho}]. \tag{5}$$

Note that $M(\lambda)$ represents any of the matrix functions $A(\lambda(t))$, $B(\lambda(t))$, $C(\lambda(t))$, and $D(\lambda(t))$ where the basis set Γ_M can be different for each of the state-space matrices. M_i, $i \in \{1, \cdots, n_\theta + n_\gamma\}$ are constant real matrices with compatible dimensions.

To obtain an "efficient" representation of (1) by (2), one should take into account complexity, accuracy and conservativeness as there exists a trade-off among these aspects (as it will become clear later). First, let us provide conceptual definitions for these notations. Then, in the subsequent sections, we will provide explicit quantitative measures to evaluate the conservativeness and accuracy.

The solution set of (2) can be denoted by

$$\mathbb{B}_{\mathrm{LPV}} = \{(y,x,u) \in (\mathbb{Y} \times \mathbb{X} \times \mathbb{U})^{\mathbb{R}_0^+} \mid \exists \lambda \in \Upsilon^{\mathbb{R}_0^+} \text{ that } (y,x,u,\lambda) \text{ satisfies (2) } \forall t \in \mathbb{R}_0^+ \text{ and } x \in \mathcal{C}_1^{n_x}\}. \tag{6}$$

Due to the fact that $\lambda(t)$ can get any value in Υ irrespective of the current value of $x(t)$, the sets $\mathbb{B}_{\mathrm{NL}}$ and $\mathbb{B}_{\mathrm{LPV}}$ are generally different. Thus, any appropriate measure on $\mathbb{B}_{\mathrm{LPV}} \backslash \mathbb{B}_{\mathrm{NL}}$ can represent the conservativeness of the embedding. Additionally, the accuracy index can be characterized by introducing a measure on the discrepancy between $F_1(x(t))$, $F_2(x(t))$, $G_1(x(t))$, $G_2(x(t))$ and $A(\mu(x(t)))x(t)$, $C(\mu(x(t)))x(t)$, $B(\mu(x(t)))$, $D(\mu(x(t)))$, respectively, on a representative data set $\mathcal{D}$ defined as

$$\mathcal{D} \coloneqq \{x(kT)\}_{k=0}^{N}, \tag{7}$$

which represents typical operation of the system and $T > 0$ is the sampling time.

Remark 1: For many engineering systems, we often know the typically occurring responses for common disturbances and reference signals. This can be seen as the typical operational behaviour of the system. On the other hand, for many systems, we initially know how much the responses are allowed to deviate form the preferred ones forming a tube where the system response must be contained for acceptable performance. This tube can be considered as the set of typical operating trajectories of the system, and the set $\mathcal{D}$ given in (7) can be constructed based on those trajectories. Nevertheless, if such a tube of trajectories is not known, one may resort to a fine grid of $\mathbb{X}$ in lieu of desired trajectories to obtain the set $\mathcal{D}$. As it becomes clear later, the more representative the data set $\mathcal{D}$ is, a less conservative and more accurate LPV model is obtained.

The state-space matrices in (2) are assumed to be polynomially parameter-dependent with respect to $\rho(t)$ and affine

in $\theta(t)$. The complexity of the LPV model is characterized by the number of scheduling variables and the polynomial degrees associated with the dependence of the state-space matrices of (2). It is worth mentioning that the LPV models depending polynomially on the scheduling variables encompass the traditional affine parameter-dependent ones that have often been considered in the literature [24]–[26]. Nowadays, many approaches have been developed for both performance analysis and controller synthesis for polynomially parameter-dependent LPV models [27]–[31]. Alternatively, LPV models polynomially dependent on scheduling variables can be written as *linear fractional representations* (LFR) [32]. There exist a wide variety of methods to tackle the control problem of LPV systems represented by LFRs, see e.g. [33], [34].

III. LPV Conversion

A. Basic Idea

We start this section with a simple example to reveal the underlying idea of the proposed method in this paper.

Example 1: Consider the following nonlinear system

$$\dot{x} = x^2 + g(x)u,$$
$$y = x,$$

which is desired to be embedded into an LPV model. It is supposed that $\underline{x} \leq x \leq \overline{x}$, where $\underline{x}$ and $\overline{x}$ are given in advance. The nonlinear function $g(x)$ is depicted in Fig. 1.b. To obtain an LPV model, one can choose to consider the direct mapping of the nonlinearities as the scheduling variables:

$$\lambda_1 \coloneqq x, \quad \lambda_2 \coloneqq g(x),$$

which leads to the following LPV model

$$\dot{x} = \lambda_1 x + \lambda_2 u, \tag{8}$$
$$y = x,$$

with the admissible region for the scheduling variables shown in Fig. 1.a that is considerably a large set causing too much conservatism. This region is obtained based on the upper and lower bounds on λ_1 and λ_2. Alternatively, one can approximate $g(x)$ in the form of a linear function of x as follows:

$$g(x) \approx ax + b,$$

where a and b are determined in the approximation process (see Fig. 1.b). While this together with $\lambda_1 = x$ introduces no conservatism, it certainly has a large approximation error if we just replace $g(x)$ by $ax + b$. To cope with the approximation error, let us define the residual as

$$r \coloneqq g(x) - ax - b.$$

Subsequently, one can obtain the following alternative LPV model

$$\dot{x} = \lambda_1 x + (a\lambda_1 + b + \lambda_2)u, \tag{9}$$
$$y = x,$$

where $\lambda_1 \coloneqq x$ and $\lambda_2 \coloneqq r = g(x) - ax - b$. Note that (9) is an exact alternative representation for (8) with a different admissible region for scheduling given in Fig. 1.c.

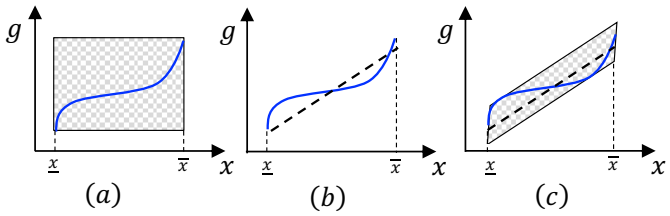

Fig. 1. The nonlinear function $g(x)$ (——), the linear approximation of the nonlinear function (----), and the admissible scheduling region (shaded area) using direct mapping of the nonlinearities as scheduling variables (a) and using a residual based embedding (c) for Example 1.

The admissible set for the nonlinear function $g(x)$ is shown in Fig. 1.c constructed according to the upper and lower bounds on λ_1 and λ_2. As one can easily see, the conservativeness is significantly reduced as common terms all dependent directly on x are used in the embedding. It is worth mentioning that leveraging to a more accurate approximation for $g(x)$, e.g. a higher order polynomial approximation resulting in less approximation error, may lead to even less conservativeness in expense of having a more complex LPV model. As one can see, there exists a trade off between model complexity and conservativeness in the embedding procedure.

Based on the presented idea, to embed the nonlinear model in an LPV one, one can first extract a polynomially parameter-dependent approximation of the nonlinear functions with a fixed order (i.e. complexity) for common terms based embedding (with no induced conservatism), and then take into account the residuals of the approximation as the scheduling variables to reduce the approximation error at the expense of conservatism; however, the number of the introduced scheduling variables should be retained as low as possible from the practical application perspective due to both complexity and conservativeness. Since some of the residuals may be correlated, PCA can be exploited to project the correlated residuals to a minimal dimensional scheduling variable set. In what follows, the steps of the above LPV model construction process are presented in details. The main feature of the proposed approach is that the designer gains control on the LPV model accuracy and conservativeness with respect to the complexity of the model.

B. Polynomial approximation

At the first step, all functions $f_i(x(t))$ and $g_{i,j}(x(t))$ in (1) are approximated with polynomials of user-chosen degree p in order to obtain $\tilde{f}_1(x(t)), \cdots, \tilde{f}_{n_x+n_y}(x(t))$ and $\tilde{g}_{11}(x(t)), \cdots, \tilde{g}_{(n_x+n_y)n_u}(x(t))$. Suppose that $l(x(t))$ denotes generic representations for either $f_i(x(t))$ or $g_{i,j}(x(t))$, and $\tilde{l}(x(t))$ represents their related approximations. In other words, the nonlinear functions are approximated as follows:

$$\tilde{l}(x(t)) = \sum_{j \in \mathcal{I}_l} \eta_j \gamma_j(x(t)), \quad \gamma_j \in \Gamma_l \tag{10}$$

where $\eta_j \in \mathbb{R}$ are decision variables, and Γ_l is a basis set that is defined as in (5). The set $\mathcal{I}_l$ denotes which basis functions of the basis set Γ_l are contributing to construct the polynomial

approximation $\tilde{l}(x(t))$. The coefficients η_j can be obtained by minimizing the ℓ_2 norm of the approximation error over $\mathcal{D}$:

$$\min_{\eta} \sum_{k=0}^{N} \Big(l(x(kT)) - \tilde{l}(x(kT))\Big)^2. \tag{11}$$

Note that to approximate the nonlinear functions $l(x(t))$, one should determine Γ_l in advance based on those state variables appearing in $l(x(t))$ and taking into account the mathematical complexity of $l(x(t))$; nonetheless, it is a high expectation from the user to be able to choose the monomials that have to be included in Γ_l. To automatically select the most essential monomials, one may resort to the following optimization problem:

$$\min_{\eta} \frac{\alpha}{n_\eta} \|\eta\|_0 + \frac{1}{N+1} \sum_{k=0}^{N} \Big(l(x(kT)) - \tilde{l}(x(kT))\Big)^2, \tag{12}$$

where $\Gamma_l = \mathfrak{P}_p[\mathbb{R}^{n_x}]$ is chosen as a trivial selection for Γ_l at first, which implies considering all possible monomials of degree at most p. n_η denotes the number of basis functions contributing to the construction of $\tilde{l}(x(t))$. Due to the fact that $\|\eta\|_0$ is the number of non-zero elements of η, solving (12) results in the minimization of the approximation error while sparsity of the expansion weights η_j is also taken into account. Here, the positive scalar value α regulates the trade-off between accuracy and complexity. Nevertheless, the optimization problem (12) is generally an intractable non-convex optimization problem and difficult to solve. Alternatively, one can resort to the ℓ_1 norm as a proxy for the ℓ_0 sparsity count by substituting $\|\eta\|_0$ by $\|\eta\|_1$ in (12). Note that the ℓ_1 relaxation problem is convex and can be cast as a quadratic programming problem. Using ℓ_1 norm as a sparsity-promoting function dates back to several decades [35], [36]. However, larger η_i coefficients are penalized more severely than smaller ones, unlike in the original optimization problem (12). To cope with this issue, one can consider the weighted ℓ_1 norm as a sparsity-promoting function which leads to the following optimization problem:

$$\min_{\eta} \frac{\alpha}{n_\eta} \|V\eta\|_1 + \frac{1}{N+1} \sum_{k=0}^{N} \Big(l(x(kT)) - \tilde{l}(x(kT))\Big)^2, \tag{13}$$

that immediately raises the question how to choose the weighting matrix V to enhance the sparsity. If we know the optimal solution for η_i denoted by $\eta_i^\star$ in advance, then we could choose the weighting matrix V as follows:

$$V = \text{diag}(v_1, v_2, \cdots, v_{n_\gamma}),$$

where

$$v_i := \begin{cases} \frac{1}{|\eta_i^\star|}, & |\eta_i^\star| \neq 0, \\ \infty, & |\eta_i^\star| = 0, \end{cases}$$

In fact, this way we put emphasis on the lower η_i elements and force them to zero, i.e. to discard the related monomials from the approximation $\tilde{l}(x(t))$. Note that large weights (small value of $|\eta_i^\star|$) could encourage the elimination of the related monomials in the polynomial approximation. Nonetheless, the values of $\eta_i^\star$ are not initially available. To surmount this drawback, using the same idea as in [37], we employ an iterative algorithm that alternates between computing η_i and redefining the weights. To this aim, the algorithm may be initiated by a weight of $V = I_{n_\gamma}$. Upon reaching a small value for $|\eta_i|$, we can deliberately discard the related monomial from the approximation (setting the related weight as ∞) to reach to a compressive sensing based approximation. When the most relevant monomials are determined, the final polynomial approximation can be obtained by the optimization problem (11) where just those monomials are kept for the approximation. Algorithm 1 summarizes this compressive sensing based polynomial approximation method. n_κ is the considered iteration number to reach an appropriate approximation. ϵ is a chosen threshold to discard the insignificant monomials in the polynomial approximation.

Algorithm 1 Compressive Sensing Based Polynomial Approximation Algorithm

1: Given an initial basis function set $\Gamma_l = \{\gamma_j\}_{j=1}^{n_\gamma}$
2: Select initial values for $n_\kappa > 0$ (number of iterations), $\alpha > 0$ (trade-off parameter), and $\epsilon > 0$ (elimination threshold)
3: $V^{(0)} = I_{n_\gamma}$, $\kappa = 0$
4: **while** $\kappa \leq n_\kappa$ **do**
5: $\quad \kappa \leftarrow \kappa + 1$
6:
$$\eta^{(\kappa)} := \arg \min_{\eta} \frac{1}{N+1} \sum_{k=0}^{N} \left(l(x(kT)) - \sum_{j=1}^{n_\gamma} \eta_j \gamma_j(x(kT))\right)^2 + \frac{\alpha}{n_\gamma} \left\|V^{(\kappa-1)}\eta\right\|_1$$
7: $\quad V^{(\kappa)} := \text{diag}(v_1, \cdots, v_{n_\gamma})$ where
$$v_j := \begin{cases} \frac{1}{|\eta_j^{(\kappa)}|}, & |\eta_j^{(\kappa)}| \geq \epsilon \\ \infty, & |\eta_j^{(\kappa)}| < \epsilon \end{cases}$$
8: **end while**
9: $\mathcal{I} := \left\{ j \in \mathbb{I}_1^{n_\gamma} \mid |\eta_j^{(\kappa)}| \geq \epsilon \right\}$
10:
$$\eta = \arg\min_{\eta} \sum_{k=0}^{N} \left(l(x(kT)) - \sum_{j \in \mathcal{I}} \eta_j \gamma_j(x(kT))\right)^2$$
11: **return**
$$\tilde{l}(x(t)) = \sum_{j \in \mathcal{I}} \eta_j \gamma_j(x(t)),$$

Now, let us obtain the polynomial approximations for the nonlinear functions $f_i(x(t))$ or $g_{i,j}(x(t))$ using Algorithm 1. Then, the related residuals are readily derived as follows for all $i = 1, \cdots, n_x + n_y$ and $j = 1, \cdots, n_u$:

$$e_i^f(x(t)) = f_i(x(t)) - \tilde{f}_i(x(t)), \tag{14a}$$
$$e_{i,j}^g(x(t)) = g_{i,j}(x(t)) - \tilde{g}_{i,j}(x(t)). \tag{14b}$$

Remark 2: The polynomial degree p can be different for the different nonlinear functions $f_i(x(t))$ or $g_{i,j}(x(t))$. However, in the sequel, to ease the complexity of the notation we consider all to be equal without loss of generality.

C. Factorization

The nonlinear system (1) can straightforwardly written as follows:

$$\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = F(x(t)) + G(x(t))u(t),$$

with

$$F(x(t)) \coloneqq \left[\tilde{f}_i(x(t)) + e_i^f(x(t))\right]_{i\in\mathbb{I}_1^{n_x+n_y}},$$
$$G(x(t)) \coloneqq \left[\tilde{g}_{i,j}(x(t)) + e_{i,j}^g(x(t))\right]_{i\in\mathbb{I}_1^{n_x+n_y},j\in\mathbb{I}_1^{n_u}},$$

where $\tilde{f}_i(\rho(t))$ and $\tilde{g}_{i,j}(\rho(t))$ are respectively the polynomial approximations of $f_i(x(t))$ and $g_{i,j}(x(t))$ obtained in the previous section, and $e_i^f(x(t))$ and $e_{i,j}^g(x(t))$ are the related residuals introduced in (14).

As the first step in obtaining the LPV representation (2), $F(x(t))$ needs to be factorized. This means that it should be decomposed as $F(x(t)) = \bar{F}(x(t))x(t)$. One may think of individually factorizing $\tilde{f}_i(x(t))$ and $e_i^f(x(t))$ to reach to the required factorization. However, note that to be able to factorize $\tilde{f}_i(x(t))$ and $e_i^f(x(t))$, it is required that $\tilde{f}_i(0) = 0$ and $e_i^f(0) = 0$. As it was explained in Assumption 1, the primal system representation can be transformed such that $f_i(0) = 0$, then as $\tilde{f}_i(x(t))$ is a polynomial, hence $\tilde{f}_i(0) = 0$; consequently, $e_i^f(0) = 0$.

Due to the fact that $\tilde{f}_i(x(t)) = \sum_{j\in\mathcal{I}} \eta_j\gamma_j(x(t))$ are polynomials with respect to $x(t)$ without any constant terms, $\tilde{f}_i(x(t))$ can immediately be factorized as follows:

$$\tilde{f}_i(x(t)) = \begin{bmatrix} \tilde{\beta}_{i,1}(x(t)) & \tilde{\beta}_{i,2}(x(t)) & \cdots & \tilde{\beta}_{i,n_x}(x(t)) \end{bmatrix} x(t), \tag{15}$$

in which some of the $\tilde{\beta}_{i,j}(x(t))$ terms are possibly zero. Subsequently, we take advantage of the presented method in [17]. Because the nonlinear functions $f(x(t))$ are bounded and assumed to be smooth, their first order partial derivatives exist. Hence, we can also factorize $e_i^f(x(t))$ as follows:

$$e_i^f(x(t)) = \begin{bmatrix} e_{i,1}^f(x(t)) & e_{i,2}^f(x(t)) & \cdots & e_{i,n_x}^f(x(t)) \end{bmatrix} x(t), \tag{16}$$

with

$$e_{i,k}^f(x(t)) = \begin{cases} \frac{e_i^f(\breve{x}_k(t)) - e_i^f(\breve{x}_{k-1}(t))}{x_k} & \text{if } x_k(t) \neq 0, \\ \left.\frac{\partial e_i^f(\breve{x}_k(t))}{\partial x_k}\right|_{x=\breve{x}_{k-1}} & \text{if } x_k(t) = 0, \end{cases} \tag{17}$$

and

$$e_{i,1}^f(x(t)) = \begin{cases} \frac{e_i^f(\breve{x}_1(t))}{x_1} & \text{if } x_1(t) \neq 0, \\ \left.\frac{\partial e_i^f(\breve{x}_1(t))}{\partial x_1}\right|_{x=0} & \text{if } x_1(t) = 0, \end{cases}$$

for $i = 1, \cdots, n_x + n_y$ and $k = 1, \cdots, n_x$, where

$$\breve{x}_k(t) \coloneqq \begin{bmatrix} x_1(t) & x_2(t) & \cdots & x_k(t) & 0 & \cdots & 0 \end{bmatrix}^\top \in \mathbb{R}^{n_x}.$$

Note that in (17), If the function e_i^f is continuous and differentiable, then the limit of the rational expression on the top for $x_k \to 0$ is equivalent with the partial derivative at the bottom at $x_k = 0$. This follows from the fundamental theorem of calculus.

Considering the fact that $f_i(x(t)) = \tilde{f}_i(x(t)) + e_i^f(x(t))$, we finally obtain

$$f_i(x(t)) = \begin{bmatrix} a_{i,1}(x(t)) & a_{i,2}(x(t)) & \cdots & a_{i,n_x}(x(t)) \end{bmatrix} x(t),$$

where

$$a_{i,k}(x(t)) \coloneqq \tilde{\beta}_{i,k}(x(t)) + e_{i,k}^f(x(t)). \tag{18}$$

Additionally, let us define

$$b_{i,j}(x(t)) \coloneqq \tilde{g}_{i,j}(x(t)) + e_{i,j}^g(x(t)). \tag{19}$$

Note that some of the state variables may appear in none of the $\tilde{\beta}_{i,k}$, $\tilde{g}_{i,j}$ due to the following reasons: (i) factorization process is not unique in general. By changing the order in which the variables x_k are considered, different factorizations of the residuals could be obtained [17]. In other words, one may deliberately exclude some of the state variables from $\tilde{\beta}_{i,k}$, e.g. $f_1(x) = x_1^2 + x_1x_2$ can either be factorized as $f_1(x) = \begin{bmatrix} x_1 + x_2 & 0 \end{bmatrix} x$ or $f_1(x) = \begin{bmatrix} x_1 & x_1 \end{bmatrix} x$ with $x = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top$, where in the later case x_2 appears neither in $\tilde{\beta}_{1,1}$ nor $\tilde{\beta}_{1,2}$, (ii) some of the state variables may not contribute to the most relevant monomials determined and used in Algorithm 1. Those state variables contributing to $\tilde{\beta}_{i,k}$, $\tilde{g}_{i,j}$ can be denoted by $\rho(t) \in \mathbb{R}^{n_\rho}$ as follows:

$$\rho(t) = Sx(t) \tag{20}$$

where the full row rank matrix S is the selector matrix in which one of the elements of each row is one and the other elements are zero.

To proceed further, let us also define the residual vector $\mathcal{E}(x(t))$ containing all the residuals $e_{i,k}^f(x(t))$ and $e_{i,j}^g(x(t))$ as follows:

$$\mathcal{E}(x(t)) = \begin{bmatrix} \mathcal{E}^f(x(t)) & \mathcal{E}^g(x(t)) \end{bmatrix}^\top \in \mathbb{R}^{(n_x+n_y)(n_x+n_u)\times 1}, \tag{21}$$

where

$$\mathcal{E}^f(x(t)) \coloneqq \overrightarrow{\mathrm{E}^f(x(t))}, \quad \mathcal{E}^g(x(t)) \coloneqq \overrightarrow{\mathrm{E}^g(x(t))},$$

with

$$\mathrm{E}^f(x(t)) = \left[e_{i,k}^f(x(t))\right]_{i\in\mathbb{I}_1^{n_x+n_y},k\in\mathbb{I}_1^{n_x}},$$
$$\mathrm{E}^g(x(t)) = \left[e_{i,j}^g(x(t))\right]_{i\in\mathbb{I}_1^{n_x+n_y},j\in\mathbb{I}_1^{n_u}}.$$

Now, we are in the position to introduce a nonlinear state-space representation for the nonlinear system (1) as follows:

$$\begin{aligned} \dot{x}(t) &= A_N(\rho(t), \mathcal{E}(x(t)))x(t) + B_N(\rho(t), \mathcal{E}(x(t)))u(t), \\ y(t) &= C_N(\rho(t), \mathcal{E}(x(t)))x(t) + D_N(\rho(t), \mathcal{E}(x(t)))u(t), \end{aligned} \tag{22}$$

with the following state-space matrices

$$\begin{aligned} A_N(\rho(t), \mathcal{E}(x(t))) &\coloneqq [a_{i,k}(\rho(t), \mathcal{E}(x(t)))]_{i\in\mathbb{I}_1^{n_x},k\in\mathbb{I}_1^{n_x}}, \\ B_N(\rho(t), \mathcal{E}(x(t))) &\coloneqq [b_{i,j}(\rho(t), \mathcal{E}(x(t)))]_{i\in\mathbb{I}_1^{n_x},j\in\mathbb{I}_1^{n_u}}, \\ C_N(\rho(t), \mathcal{E}(x(t))) &\coloneqq [a_{i,k}(\rho(t), \mathcal{E}(x(t)))]_{i\in\mathbb{I}_{n_x+1}^{n_x+n_y},k\in\mathbb{I}_1^{n_x}}, \\ D_N(\rho(t), \mathcal{E}(x(t))) &\coloneqq [b_{i,j}(\rho(t), \mathcal{E}(x(t)))]_{i\in\mathbb{I}_{n_x+1}^{n_x+n_y},j\in\mathbb{I}_1^{n_u}}. \end{aligned} \tag{23}$$

Bear in mind that $a_{i,k}$ and $b_{i,j}$ are polynomial functions with respect to $\rho(t)$ and affine functions in the residuals $\mathcal{E}(x(t))$.

Remark 3: Since the employed factorization (16) is not unique, the way that the factorization is carried out may affect the model conservativeness, which implies that we require a kind of an index to evaluate the conservatism. The latter allows to chose an appropriate LPV model among the possiblities, which is addressed in Section IV. Moreover, it should be further investigated in future research work which factorization results in better controllability and observability indices [38].

D. PCA-based scheduling variable selection

As it was mentioned earlier, (22) is a state-space representation for the nonlinear system (1). It is possible to envisage (22) as an LPV model by considering both those states appearing in $\rho(t)$ and all the residuals ($\mathcal{E}(x(t))$) as the scheduling variables. However, this leads to an LPV model that is probably highly conservative because the residuals are often not independent of $\rho(t)$. Additionally, the resulting model can be highly complex due to introducing high number of scheduling variables. Note that some of the residual variables in $\mathcal{E}(x(t))$ may be the same or can be written as the linear combinations of some of the others. That enables us to obtain an accurate LPV model having less scheduling variables, which in turn leads to less conservative and less complex LPV model. Furthermore, there exist two scenarios in which one may try to find an approximate LPV model for (1); firstly, because some of the residuals are highly correlated, one may consider reducing the number of scheduling variables while the accuracy is preserved; secondly, one may require an LPV model of the nonlinear system (1) having a predefined number of scheduling variables. To uniquely address the embedding problem with a reduced number of scheduling variables for all these cases (both the accurate and approximate LPV embeddings), we can resort to PCA to define a new set of variables in terms of linear combinations of the introduced residuals; subsequently, the residuals in (22) are replaced by appropriate linear combinations of the newly introduced variables. The new set of variables and $\rho(t)$ comprise the final set of scheduling variables. Inspired by the presented method in [20], first let us generate the following data matrix:

$$\Pi = \begin{bmatrix} \mathcal{E}(x(0)) & \mathcal{E}(x(T)) & \cdots & \mathcal{E}(x(NT)) \end{bmatrix}. \tag{24}$$

Then, the row Π_i of this matrix is normalized by an affine law $\mathcal{N}_i$ to obtain scaled (unit variance), zero mean data matrix

$$\Pi_i^n = \mathcal{N}_i(\Pi_i),$$

leading to the normalized matrix $\Pi^n = \mathcal{N}(\Pi)$ that can be employed for the PCA which is based on singular value decomposition (SVD). Singular values indicate the principal components of the data. In case the data are correlated, some singular values are small in comparison with the others. Small singular values indicate relatively insignificant components. This implies that the projection onto a low-dimensional subspace spanned by the dominant singular vectors will not lead to losing too much information. Now, consider the singular value decomposition (SVD) of Π^n as follows:

$$\Pi^n = U\Sigma V'.$$

Subsequently, matrices U, V, and $\Sigma = \text{diag}(\sigma_1, \cdots, \sigma_{n_\theta}, \sigma_{n_\theta+1}, \cdots, \sigma_{(n_x+n_y)(n_x+n_u)})$ are partitioned as

$$U \coloneqq \begin{bmatrix} U_s & U_n \end{bmatrix}, \quad \Sigma \coloneqq \begin{bmatrix} \Sigma_s & 0 & 0 \\ 0 & \Sigma_n & 0 \end{bmatrix},$$
$$V \coloneqq \begin{bmatrix} V_s & V_n \end{bmatrix},$$

where U_s, Σ_s, and V_s correspond to n_θ significant singular values. By neglecting the singular values $\sigma_{n_\theta+1}, \cdots, \sigma_{(n_x+n_y)(n_x+n_u)}$ and consequently their related principal components of the data, the following approximation for Π^n is obtained:

$$\widetilde{\Pi}^n \coloneqq \mathcal{N}^{-1}\left(U_s U_s^\top \mathcal{N}\left(\Pi^n\right)\right) \approx \Pi^n,$$

where $\mathcal{N}^{-1}$ denotes the rescaling and translation such that $\mathcal{N}^{-1}\left(\mathcal{N}\left(\Pi^n\right)\right) = \Pi^n$. This immediately implies that introducing the scheduling variable vector

$$\theta(t) \coloneqq \begin{bmatrix} \theta_1(t) & \theta_2(t) & \cdots & \theta_{n_\theta}(t) \end{bmatrix}^\top,$$

as follows

$$\theta(t) \coloneqq \mathcal{T}(x(t)) = U_s^\top \mathcal{N}(\mathcal{E}(x(t))), \tag{25}$$

enables us to obtain an approximation $\widetilde{\mathcal{E}}(t)$ for the residuals $\mathcal{E}(t)$ as:

$$\widetilde{\mathcal{E}}(\theta(t)) \coloneqq \mathcal{N}^{-1}(U_s\theta(t)) \approx \mathcal{E}(x(t)).$$

In principle, this ensures that the number of scheduling variables introduced by the residual terms will be minimal. In case some of the residuals can be interpreted as the linear combinations of the others (special case is when they are the same), the corresponding singular values are zero. Hence, by taking Σ_n to contain only those zero singular values, we obtain an exact LPV model with a reduced number of scheduling variables. Note that the approximation $\widetilde{\mathcal{E}}(t)$ of $\mathcal{E}(t)$ is a multivariate affine function with respect to the newly introduced scheduling variable vector $\theta(t)$. A higher number of scheduling variables n_θ leads to a more accurate LPV model. As a matter of fact, there exists a trade-off between the number of scheduling variables n_θ and the desired accuracy of the model. It is worth mentioning that we employ PCA strategy to cope with the residuals contrary to the method in [20] in which the PCA is applied on the scheduling variables.

Remark 4: Note that to obtain an LPV model with reduced number of scheduling variables, PCA can be applied on both $\rho(t)$ and $\mathcal{E}(x(T))$ rather than just $\mathcal{E}(x(T))$. This way, the whole scheduling variables are thoroughly defined by PCA.

E. LPV model

To recapitulate, one can readily obtain an LPV model (2) for the nonlinear system (1) by replacing $\mathcal{E}(x(t))$ with $\widetilde{\mathcal{E}}(\theta(t))$ in the state-space matrices (23) to obtain

$$A(\lambda(t)) \coloneqq A_N(\rho(t), \widetilde{\mathcal{E}}(\theta(t))), \;\; B(\lambda(t)) \coloneqq B_N(\rho(t), \widetilde{\mathcal{E}}(\theta(t))),$$
$$C(\lambda(t)) \coloneqq C_N(\rho(t), \widetilde{\mathcal{E}}(\theta(t))), \;\; D(\lambda(t)) \coloneqq D_N(\rho(t), \widetilde{\mathcal{E}}(\theta(t))).$$

This way the overall scheduling variable vector $\lambda(t)$ comprises $\theta(t)$ and $\rho(t)$. The state-space matrices of the obtained LPV model depend polynomially on $\rho(t)$ and affinely on $\theta(t)$. $\lambda(t)$ belongs to a hyperrectangle Υ defined by (3). The lower and upper bounds $\underline{\lambda}_i$ and $\overline{\lambda}_i$, representing the hyperrectangle, are obtained respectively as the minimum and maximum values of $\lambda_i(t)$ over the data set $\mathcal{D}$. Considering (20) and (25), one can obtain

$$\begin{aligned}
\underline{\lambda}_j &\coloneqq \min_{x\in\mathcal{D}} \ \mathcal{U}_j^\top Sx, \quad j=1,\cdots,n_\rho\\
\overline{\lambda}_j &\coloneqq \max_{x\in\mathcal{D}} \ \mathcal{U}_j^\top Sx, \quad j=1,\cdots,n_\rho\\
\underline{\lambda}_{i+n_\rho} &\coloneqq \min_{x\in\mathcal{D}} \ \mathcal{W}_i^\top U_s^\top \mathcal{N}(\mathcal{E}(x)), \quad i=1,\cdots,n_\theta\\
\overline{\lambda}_{i+n_\rho} &\coloneqq \max_{x\in\mathcal{D}} \ \mathcal{W}_i^\top U_s^\top \mathcal{N}(\mathcal{E}(x)), \quad i=1,\cdots,n_\theta
\end{aligned}$$

where $\mathcal{U}_j \in \mathbb{R}^{n_\rho}$ and $\mathcal{W}_i \in \mathbb{R}^{n_\theta}$ are unitary vectors whose respectively jth and ith elements are one and the other elements are zero.

The accuracy of the LPV model is directly connected with the complexity of the model, i.e. the number of scheduling variables $\theta(t)$ and polynomial degree p. Deploying higher number of scheduling variables and/or higher degree polynomial approximations (increasing the model complexity) obviously delivers a more accurate LPV model. Besides that, one can expect that higher number of scheduling variables lead to a more conservative LPV model due to adding more dimensions to the admissible region Υ. On the other hand, increasing the polynomial approximation degree will reduce the conservativeness because the absolute values of residual signals are generally reduced; therefore, a smaller admissible region Υ for the scheduling variables is obtained which immediately leads to a less conservative model. Therefore, increasing just the polynomial approximation degree results in more accurate and less conservative model. Increasing just the number of scheduling variables delivers a more accurate, but more conservative model. Consequently, there exists a trade-off between model complexity, accuracy, and conservativeness.

To choose the right number of scheduling variables and polynomial approximation degree p, we need to quantify both the model accuracy and conservativeness. In the upcoming section, we develop appropriate indices to measure model accuracy and conservativeness which enable the designer to determine the model complexity in the proposed method.

IV. Accuracy and Conservativeness Indices

In order to quantitatively evaluate the accuracy and conservativeness of the LPV model (2), we introduce adequate measures in the sequel.

Note that the obtained LPV model is derived based on the approximation of the nonlinear state equations of the original system; therefore, we choose to evaluate the accuracy at the state equation level. This means that we define the accuracy index as a measure of the discrepancy between $F_1(x)$, $F_2(x)$, $G_1(x)$, and $G_2(x)$ and their LPV counterparts $A(\lambda)x$, $C(\lambda)x$, $B(\lambda)$, and $D(\lambda)$ with $\lambda = \begin{bmatrix} (Sx)^\top & \mathcal{T}(x)^\top \end{bmatrix}^\top$ over the representative data set $\mathcal{D}$, given in (7); alternatively, one may also consider $\mathcal{D}$ as any arbitrary set of samples of $\mathbb{X}$. This implies that the nonlinear equations and their LPV counterparts are compared over fixed values of the states related to a typical operation of the system or a gridded set of $\mathbb{X}$. Based on this idea, one way to define the accuracy index is as follows:

$$\Lambda_a^{\mathcal{D}} \coloneqq \frac{1}{N+1}\left\|\Pi_N - \Pi_L\right\|_F, \tag{26}$$

where

$$\begin{aligned}
\Pi_N &\coloneqq \begin{bmatrix} \Omega_N(x(0)) & \cdots & \Omega_N(x(NT)) \end{bmatrix},\\
\Pi_L &\coloneqq \begin{bmatrix} \Omega_L(\lambda(0)) & \cdots & \Omega_L(\lambda(NT)) \end{bmatrix},
\end{aligned}$$

with

$$\begin{aligned}
\Omega_N(x) &\coloneqq \begin{bmatrix} \overrightarrow{F}(x) & \overrightarrow{G}(x) \end{bmatrix}^\top,\\
\Omega_L(x) &\coloneqq \begin{bmatrix} \overrightarrow{F}_L(x) & \overrightarrow{G}_L(x) \end{bmatrix}^\top,
\end{aligned}$$

considering

$$F_L(x) \coloneqq \begin{bmatrix} A(Sx,\mathcal{T}(x)) \\ C(Sx,\mathcal{T}(x)) \end{bmatrix} x, \tag{27}$$

$$G_L(x) \coloneqq \begin{bmatrix} B(Sx,\mathcal{T}(x)) \\ D(Sx,\mathcal{T}(x)) \end{bmatrix}, \tag{28}$$

where $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, and $D(\cdot)$ are given by (4). Obviously, a lower value for the accuracy index implies a more accurate LPV model. Even though our proposed factorization in this paper is not unique, it does not affect the model accuracy since to compute the accuracy index we again multiply A and C matrices by x, which counteracts the way that $F(x(t))$ is factorized, see (27).

The gap metrics are known to provide quantitative measures to distinguish the difference between open-loop dynamical systems in terms of their closed-loop behavior. In this section, ν-gap metric is deployed to quantify the conservativeness of the LPV model in a frozen sense, i.e. for constant trajectories of $x(t)$. Note that, for such a frozen scheduling value, (2) and (22) become LTI for which the IO transfer can be well characterized by a transfer function.

The ν-gap metric between two LTI plants with transfer functions P_1 and P_2 is given by

$$\delta_\nu \coloneqq \begin{cases} \delta_\nu(P_1,P_2) & \text{if the } \textit{WNC} \text{ holds,} \\ 1 & \text{else,} \end{cases}$$

with

$$\delta_\nu \coloneqq \left\|(I+P_2P_2^*)^{-\frac{1}{2}}(P_1-P_2)(I+P_1P_1^*)^{-\frac{1}{2}}\right\|_\infty,$$

where *WNC* is the *Winding Number Condition* stated as follows [39]:

- $\det\left(I+P_2^\star(j\omega)P_1(j\omega)\right) \neq 0 \quad \forall\omega\in\mathbb{R}$,
- The counterclockwise winding number of $\det\left(I+P_2^\star(s)P_1(s)\right)$ around the origin, as s traces the standard Nyquist D-contour equals to $\Phi(P_2)-\Phi(P_1)$,

where $\Phi(P)$ denotes the number of open right-half plane poles of a system with transfer function $P(s)$. A clockwise encirclement counts as a negative encirclement and vice versa.

This metric, having a value in the range of $[0,1]$, defines a notion of distance between two LTI plants in terms of

closed-loop behaviour when both are controlled by a same controller. A close value to 0 means that the distance between two systems is small, in this case a reasonably good controller for one plant leads to similar performance with the other. On the other hand, a value closer to 1 indicates that the closed-loop dynamic behaviors of the two systems are significantly different.

Note that in the LPV model (2), the scheduling variable vectors $\theta(t)$ and $\rho(t)$ conceptually can take any values in the admissible regions Θ and Ψ irrespective of the current value of $x(t)$, which is the reason for the model conservativeness. Taking into account that the state-space model (22) can be thought as an exact state-space representation for the nonlinear system (1), the largest (over all $\theta(t) \in \Theta$ and $\rho(t) \in \Psi$) minimum metric value (over all $x(t) \in \mathcal{D}$) between the model (22) and the obtained LPV model (2) can be considered as a measure representing conservativeness. As the state-space models (22) and (2) with frozen scheduling variables convert to LTI models, the computation of the ν-gap metric can easily be carried out in terms of a convex optimization problem. However, it does not take into account conservativeness of the representation for varying scheduling variables. The conservativeness index is introduced as follows:

$$\Lambda_c^{\mathcal{D}} := \max_{\theta\in\Theta,\rho\in\Psi} \min_{x\in\mathcal{D}} \delta_\nu(\mathcal{P}_L(\rho,\theta), \mathcal{P}_N(x)), \tag{29}$$

with

$$\mathcal{P}_L(\rho,\theta) \leftrightarrow \left[\begin{array}{c|c} A(\rho,\theta) & B(\rho,\theta) \\ \hline C(\rho,\theta) & D(\rho,\theta) \end{array}\right],$$
$$\mathcal{P}_N(x) \leftrightarrow \left[\begin{array}{c|c} A_N(Sx,\mathcal{E}(x)) & B_N(Sx,\mathcal{E}(x)) \\ \hline C_N(Sx,\mathcal{E}(x)) & D_N(Sx,\mathcal{E}(x)) \end{array}\right],$$

where $A(\cdot), B(\cdot), C(\cdot), D(\cdot)$ and $A_N(\cdot), B_N(\cdot), C_N(\cdot), D_N(\cdot)$ are given by (4) and (23), respectively. Note that $\rho = Sx$. The transfer functions $\mathcal{P}_L$ and $\mathcal{P}_N$ represent LTI plants whose state-space matrices are constant and equal to the values of $A(\cdot)$, $B(\cdot)$, $C(\cdot)$, $D(\cdot)$ and $A_N(\cdot)$, $B_N(\cdot)$, $C_N(\cdot)$, $D_N(\cdot)$, respectively, evaluated at a fixed value of $x \in \mathcal{D}$. It is worth mentioning that in (29) one may apply $x \in \mathbb{X}$ rather than $x \in \mathcal{D}$. Nevertheless, since the LPV model is derived using the representative data set $\mathcal{D}$, it is sensible to also compute the conservativeness index over the set $\mathcal{D} \subset \mathbb{X}$.

To compute an assured lower bound for the conservativeness index (29), the admissible set Θ can be gridded and $\Psi = S\mathcal{D}$ is considered.

Remark 5: It is worth mentioning that other kinds of approximations of nonlinear functions can also be used in the proposed method instead of polynomial approximation. One may also consider different sets of basis functions for the approximations. Once the residuals are computed (the difference between the approximation and the original nonlinear functions), one can follow the proposed method to define the scheduling variables taking advantage of the aforementioned proposed accuracy and conservativeness indices.

V. Numerical illustration

In this section, a numerical example is provided to demonstrate and analyze the main features of the proposed approach.

TABLE I

Accuracy index $\mathbf{\Lambda_a^{\mathcal{P}}}$, conservativeness index $\mathbf{\Lambda_c^{\mathcal{P}}}$, and scheduling variables $\boldsymbol{\lambda}$ for models obtained using affine and 3rd order polynomial approximations with different number of scheduling variables (No. Sch.).

	No. Sch.	1	2	3	4
$\Lambda_a^{\mathcal{P}}$	Affine	0.205	0.099	0.063	0.020
	2nd poly.	—	—	0.067	0.047
	3rd poly.	—	—	0.022	0.018
$\Lambda_c^{\mathcal{P}}$	Affine	0.244	0.964	1	1
	2nd poly.	—	—	0.202	0.216
	3rd poly.	—	—	0.013	0.021
λ	Affine	θ_1	θ_1, θ_2	$\theta_1, \theta_2, \theta_3$	$\theta_1, \theta_2, \theta_3, \theta_4$
	2nd poly.	—	—	x_1, x_2, θ_1	$x_1, x_2, \theta_1, \theta_2$
	3rd poly.	—	—	x_1, x_2, θ_1	$x_1, x_2, \theta_1, \theta_2$

Example

Consider the nonlinear system (1) with

$$\begin{bmatrix} f_1(x) \\ f_2(x) \\ f_3(x) \end{bmatrix} = \begin{bmatrix} 5x_2 + 10x_1x_2 - 2x_1^3 + 3x_1x_2\sin(\frac{\pi}{2}x_2) \\ 7x_1^4 + 4x_2^2 \\ x_1 \end{bmatrix},$$
$$\begin{bmatrix} g_{1,1}(x) & g_{1,2}(x) \\ g_{2,1}(x) & g_{2,2}(x) \\ g_{3,1}(x) & g_{3,2}(x) \end{bmatrix} = \begin{bmatrix} 10x_1^3\cos(\frac{\pi}{2}x_2) & 1 \\ 10x_1^2\sin(\frac{\pi}{2}x_2) & 0 \\ 1 & 1 \end{bmatrix},$$

with the following typical operation of the system:

$$\begin{aligned} x_1(t) &= 0.5\sin(8\pi t) + 0.5, \\ x_2(t) &= \sin(15\pi t). \end{aligned} \tag{30}$$

We treat three scenarios to embed this nonlinear system:

- First scenario (1st poly.): 1st order polynomial approximations for f_i functions and zero-order approximations for $g_{i,j}$ functions.
- Second scenario (2nd poly.): 2nd order polynomial approximations for all the functions.
- Third scenario (3rd poly.): 3rd order polynomial approximations for all the functions.

In these cases, the LPV models are synthesized with different number of scheduling variables. The quality of the embeddings which can be evaluated by the value of the accuracy index $\Lambda_a^{\mathcal{P}}$ and the conservativeness index $\Lambda_c^{\mathcal{P}}$ are reported in Table I. Furthermore, the related scheduling variables are given for all the cases. A representative data set $\mathcal{D}$ is constructed based on (30) with $N = 500$ and sampling time $T = 0.001$ sec. To compute the conservativeness index, Ψ is gridded as $\Psi = S\mathcal{D}$ based on (20), then, the ν-gap metric is computed using the `gapmetric` function in Matlab. Furthermore, each θ_i is linearly gridded in the interval $[\underline{\lambda}_i,\ \overline{\lambda}_i]$ with 5 points to obtain a gridded set of Θ. For the polynomial approximations which are carried out using Algorithm 1, we consider $n_\kappa = 5$ (arbitrarily chosen number of iterations to promote the polynomial approximation sparsity), $\alpha = 0.0002$ (user-chosen value to balance the trade-off between the accuracy and the sparsity in polynomial approximations), and $\epsilon = 0.01$ (the monomials whose coefficients are less than 0.01 are eliminated in the polynomial approximations).

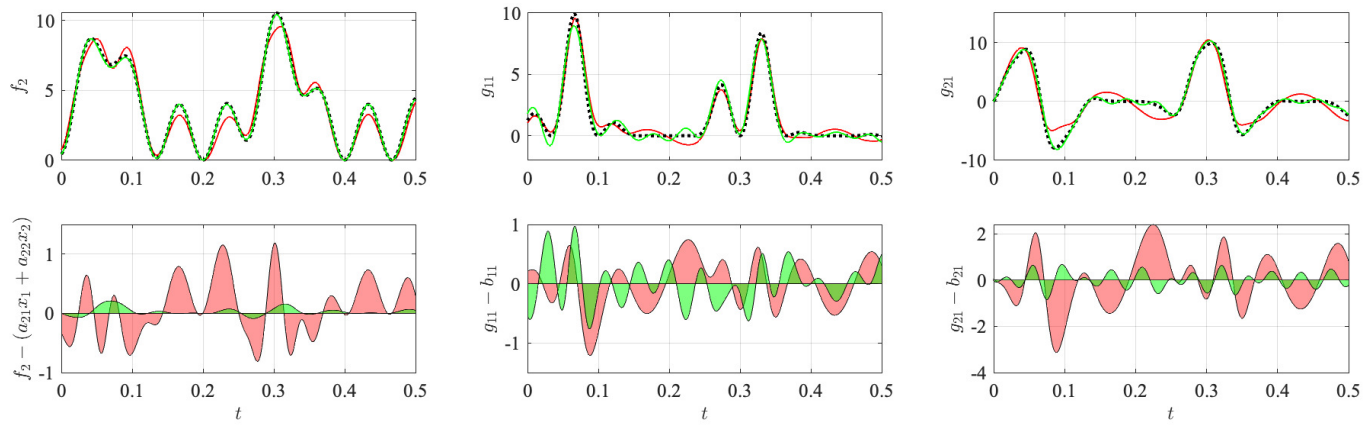

Fig. 2. Results of the 3rd poly. model in Example 1; Results of the 1st poly. model in Example 1. First row: the actual variation of $\boldsymbol{f_2(x)}$, $\boldsymbol{g_{1,1}(x)}$, and $\boldsymbol{g_{2,1}(x)}$ (·····) along a solution trajectory together with their LPV representation (——) for both the 1st poly. and the 3rd poly. cases. Second row: the errors between the true and the approximated functions.

TABLE II

THE COEFFICIENTS OF THE MONOMIALS FOR 3RD ORDER POLYNOMIAL APPROXIMATIONS OF $\boldsymbol{f_1(x)}$.

Monomial	1	x_1	x_2	x_1^2	x_1x_2	x_2^2	x_1^3	$x_1^2x_2$	$x_1x_2^2$	x_2^3
Without sparsity	−0.0002	0.1913	4.9926	0.0537	10.0506	0.0096	−2.0347	−0.0431	2.9690	−0.0018
With sparsity	0.0000	0.2105	4.9999	0.0000	10.0024	0.0000	−2.0005	0.0000	2.9795	0.0000

As it is obvious in Table I, for all the scenarios, increasing the number of the scheduling variables leads to more accurate model; this, however, results in more conservative LPV models. Moreover, the embeddings obtained using 3rd order polynomial approximation are more accurate and less conservative in comparison with the results obtained by 1st order approximation considering the same number of scheduling variables. The results in Table I reveal that one can choose the number of the scheduling variables and the polynomial degree for the approximation of the nonlinear functions based on the required accuracy and conservativeness.

The nonlinear functions $f_2(x)$, $g_{1,1}(x)$, $g_{2,1}(x)$ and their counterparts in the LPV representation, namely, $a_{21}(\lambda)x_1 + a_{22}(\lambda)x_2$, $b_{11}(\lambda)$, and $b_{21}(\lambda)$ are depicted in Fig. 2 for the first and third scenarios using 3 scheduling variables. Moreover, in the second row of the same figure, the difference between the actual functions and their counterparts in the LPV representation are depicted. For the 1st poly. case, the scheduling variables are $\lambda(t) = [\theta_1(t)\ \theta_2(t)\ \theta_3(t)]^\top$ and for the 3rd poly. case $\lambda(t) = [x_1(t)\ x_2(t)\ \theta_1(t)]^\top$ are the actual scheduling variables. According to Fig. 2, it is obvious that the 3rd poly. case leads to a more accurate model, which is expected based on the reported accuracy indices in Table I.

Finally, to evaluate the applicability of Algorithm 1 which is used to enhance sparsity of the polynomial approximations, the 3rd order polynomial approximations of $f_1(x)$ are given in Table II for the following two cases:

- Without sparsity: $\alpha = 0$, i.e. the sparsity of the polynomial approximations is not taken into account at all.
- With sparsity: $V^{(k)}$ is a varying weight which is computed in Algorithm 1 with $\alpha = 0.00002$.

TABLE III

MSE VALUES BETWEEN THE NONLINEAR FUNCTIONS AND THE LPV REPRESENTATIONS FOR THE PROPOSED METHOD (PM) AND THE METHOD OF [20] (KW) FOR DIFFERENT NUMBER OF SCHEDULING VARIABLES (NO. SCH.) WITH $\boldsymbol{c_i^f = f_i(x) - (a_{i1}(\lambda)x_1 + a_{i2}(\lambda)x_2)}$ AND $\boldsymbol{c_j^g = g_{i1}(x) - b_{i1}(\lambda)}$ FOR $\boldsymbol{i = 1, 2}$.

		c_1^f	c_2^f	c_1^g	c_2^g
No. Sch.= 2	PM	0.5799	0.6825	1.8477	1.8709
	KW	2.9565	1.4661	0.9150	2.6844
No. Sch.= 3	PM	0.1244	0.2476	0.1876	1.4796
	KW	1.7297	0.4937	0.6356	2.2176
No. Sch.= 4	PM	0.1231	0.0321	0.0011	0.0471
	KW	0.1338	0.0763	0.3514	0.3957

One can see from Table II that the polynomial approximation with sparsity is more sparse than the other as it is expected. Additionally, upon comparing the true nonlinear function $f_1(x)$ with the obtained result with the sparsity, the considered monomials for the polynomial approximation seem sensible considering the fact that the terms $1, x_1^2, x_2^2, x_1^2x_2$, and x_2^3 will also not appear in an approximation of f_1 readily obtained by replacing $\sin(\pi/2x_2)$ by its Maclaurin series.

For the comparison purposes, the presented method in [20] which utilizes PCA-based technique of parameter set mapping to obtain a model of complexity low enough is also applied on the system. Then, the obtained results are compared with those of the 1st poly. case. For the quantitative comparison, *mean square error* (MSE) between the actual values of the nonlinear functions and their counterparts in the LPV modeling are reported in Table III. The results show the superiority of the proposed method for the LPV embedding when the same numbers of the scheduling variables are employed.

Fig. 3. The 3DOF gyroscope experimental setup by Quanser.

VI. Experimental Results

In this section, the application of the proposed method on a 3DOF gyroscope experimental setup shown in Fig. 3 is studied. The system consists of a golden flywheel (#1) mounted inside a blue gimbal (#2), which itself is mounted inside a red gimbal (#3). The entire structure is mounted inside a silver frame (#4) which is assumed to be fixed in this paper. The flywheel, the blue gimbal, and the red gimbal can freely rotate around their rotational axes by using torques provided by their own separate DC motors (#5, #6, and #7). The angular position of the flywheel, the red gimbal, and the blue gimbal are measured by optical encoders (#5, #6, and #7). The goal here is to make the blue and the red gimbals track the desired reference trajectories by supplying appropriate torques through their related motors where the speed variation of the flywheel acts as disturbance on the system.

The equations of motion can be derived using Euler-Lagrangian formulation as follows:

$$\dot{\omega}_b = f_b(q_b, q_r, \omega_g, \omega_b, \omega_r) + g_{bb}(q_b, q_r)\tau_b + g_{br}(q_b, q_r)\tau_r,$$
$$\dot{\omega}_r = f_r(q_b, q_r, \omega_g, \omega_b, \omega_r) + g_{rb}(q_b, q_r)\tau_b + g_{rr}(q_b, q_r)\tau_r,$$

where q_b and q_r are the angular position of blue and red gimbals, respectively. ω_g, ω_b, and ω_r are the angular speed of the flywheel, blue and red gimbals, respectively. τ_b and τ_r are the applied torques by the related motors to the blue and red gimbals. Due to complexity, mathematical expressions for f_b, f_r, g_{bb}, g_{br}, g_{rb}, and g_{rr} are not given here. The interested reader can refer to [40] to find them.

To obtain an LPV state-space representation for the system, let us define the state, input, and output vectors as follows:

$$x(t) \coloneqq \begin{bmatrix} q_b & q_r & \omega_b & \omega_r \end{bmatrix}^\top,$$
$$u(t) \coloneqq \begin{bmatrix} \tau_b & \tau_r \end{bmatrix}^\top, \quad y(t) \coloneqq \begin{bmatrix} q_b & q_r \end{bmatrix}^\top.$$

Using the proposed method and considering first-order polynomial approximations for f_b and f_r and zero-order polynomial approximation for g_{bb}, g_{br}, g_{rb}, and g_{rr}, an LPV model is obtained for the gyroscope. To have a simple model from the control synthesis point of view, we accomplish the embedding with two scheduling variables. The associated accuracy and conservativeness indices for the obtained LPV model are as follows:

$$\Lambda_a^{\mathcal{D}} = 0.064, \quad \Lambda_c^{\mathcal{D}} = 0.250.$$

Additionally, to evaluate the quality of the obtained LPV model, the functions f_b, f_r, g_{br}, and g_{rr} and their related

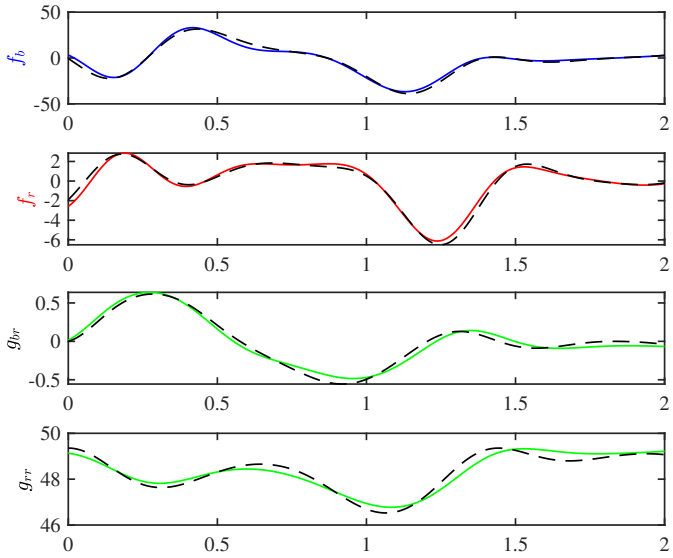

Fig. 4. True nonlinear functions (— —) and their related approximation in the LPV model (——) for the gyroscope setup.

LPV counterparts are depicted in Fig. 4. for the following trajectories for the angular position of the blue and red gimbals:

$$q_b = 0.5\sin(2t) + 0.2\sin(7t),$$
$$q_r = 0.3\sin(4t + \frac{\pi}{6}) + 0.3\sin(6t),$$

and flywheel angular speed of

$$\omega_g = 20 + 10\sin(6t).$$

Obviously, the LPV model and the true nonlinear model are well coincide which reveals that how well the LPV model captures the dynamics of the nonlinear system. Furthermore, magnitude plots of open-loop frozen frequency responses of the LPV model at the vertices of the admissible set for the scheduling variables are depicted in Fig. 5. Significant gain and pole variations are observed over the scheduling range, which underpin the requirement for a parameter-varying model to capture the dynamics of the nonlinear model.

Now, consider the closed-loop setup shown in Fig. 6. We consider the design of a gain-scheduled H_∞-type (precisely an $\mathcal{L}_2$-gain optimal) state feedback controller to achieve good tracking performance while the actuator constraints are also taken into account during shaping. To this end, the weighting functions below are employed

$$W_s = 500 \begin{bmatrix} \frac{0.0016s+1}{s} & 0 \\ 0 & \frac{0.0016s+1}{s} \end{bmatrix}, \tag{31}$$

$$W_k = 45 \begin{bmatrix} \frac{4s+1}{0.001s+1} & 0 \\ 0 & \frac{4s+1}{0.001s+1} \end{bmatrix}. \tag{32}$$

The weighting function W_s has been designed based on the standard mixed-sensitivity shaping approach for the frozen sensitivity frequency responses of the closed-loop system using a first order filter parameterization with 20dB/decade low frequency roll-off, the required chosen target closed-loop bandwidth of 100 Hz and 20% of allowed overshoot encoded as a 6dB sensitivity peak. Similarly, the control sensitivity

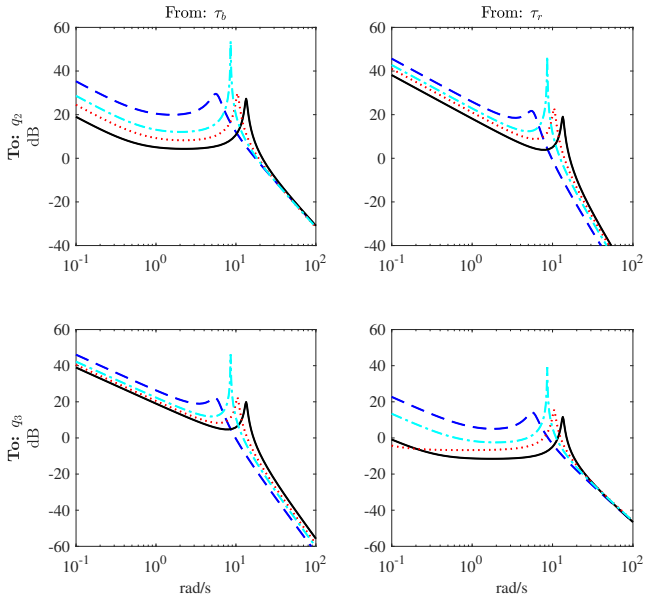

Fig. 5. Magnitude plots of the open-loop frequency responses of the LPV model at four frozen vertices of the admissible region.

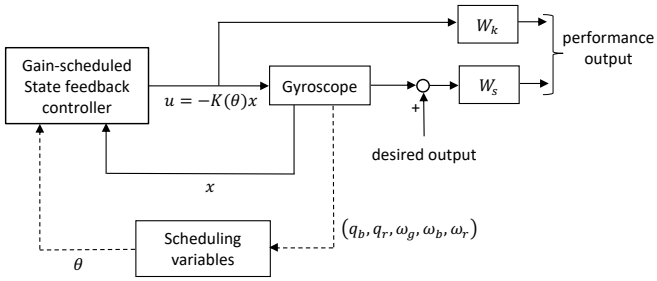

Fig. 6. The applied LPV control structure for the qyroscope.

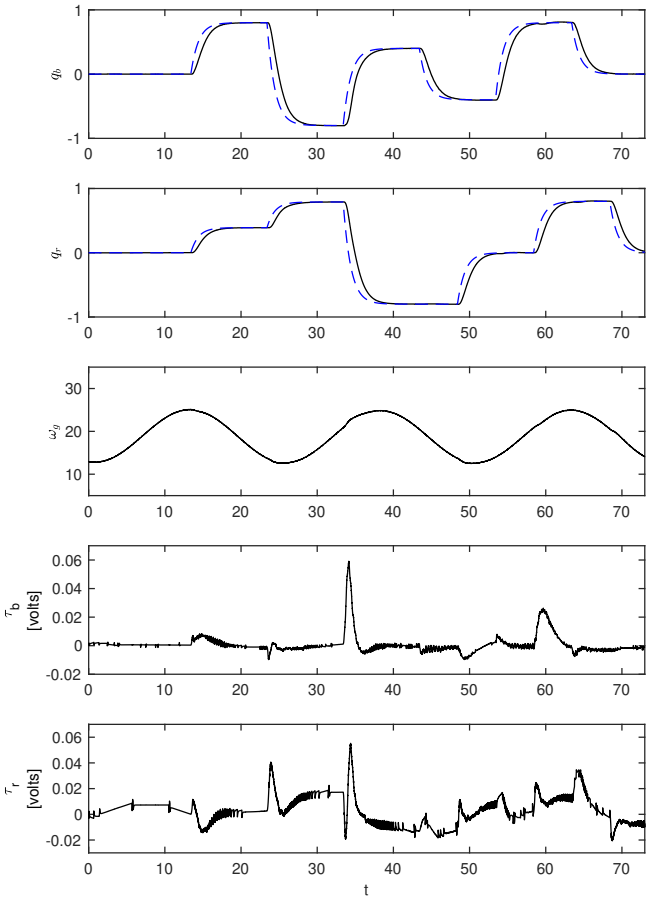

Fig. 7. Experimental results for the laboratory gyroscope setup. The desired reference (— —) and the obtained results from the real setup (——). Figures from top to bottom: angular position of the blue gimbal, angular position of the red gimbal, velocity of the flywheel, control input signal to the blue gimbal, control input signal to the red gimbal.

weight W_k has been chosen considering the amplitude and frequency constraints on the voltage to be applied to the DC motors of the experimental setup. Gain-scheduled state feedback controller is designed by the proposed method in [41] where the parameter-dependent LMIs are solved through finite-dimensional LMI relaxation based on homogeneous polynomial matrices [27]. To solve the problem, YALMIP [42] and ROLMIP [43] interfaces for the LMI solver MOSEK [44] are utilized.

The designed controller is experimentally validated on the laboratory setup using a dSPACE board that implements a real time interface so that control implementation in MATLAB/Simulink becomes possible. Using a simple proportional controller, the angular velocity of the flywheel is made to track a sinusoidal reference between 10 to 30 rad/sec which plays the role of the disturbance to be rejected to achieve accurate angular position control of the blue and red gimbals. Note that due to the high-resolution encoders there is not much measurement error. The desired trajectories and actual angular position of the blue and red gimbals, the angular velocity of the flywheel, and the control input signals are depicted in Fig. 7. As one can see, we have a good tracking performance in a wide operating range (± 0.8 rad) for both red and blue gimbals while the control input signals remain within acceptable levels. This confirms again the accuracy of the LPV model and applicability of the proposed method on a highly nonlinear system.

VII. Conclusion

A novel method to systematically embed the dynamical behavior of nonlinear systems into LPV models depending polynomially on automatically constructed scheduling variables is presented in this paper. Considering the trade-off between model complexity, model accuracy, and model conservativeness, a tailored LPV model for the nonlinear system is obtained by the proposed procedure. The main contribution of the proposed method is to jointly consider the model accuracy, complexity, and conservativeness into embedding procedure. As a possible future research work, the possibility of incorporating the controllability and observability of the resulting LPV models in the embedding procedure can also be considered. In this paper, in addition to an academic example, the efficiency of the new method is investigated in an empirical study where the first-principle motion model of

a 3DOF control moment gyroscope is converted to an LPV model with low scheduling complexity using the proposed method. Based on this model, a gain-scheduled state feedback controller is designed and applied on the gyroscope, achieving fast and reliable reference tracking, which clearly demonstrates the practical usefulness of the presented method.

References

[1] C. Hoffmann and H. Werner, "A survey of linear parameter-varying control applications validated by experiments or high-fidelity simulations," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 2, pp. 416–433, 2015.

[2] R. Tóth, V. Laurain, M. Gilson, and H. Garnier, "Instrumental variable scheme for closed-loop LPV model identification," *Automatica*, vol. 48, no. 9, pp. 2314 – 2320, 2012.

[3] J. Goos and R. Pintelon, "Continuous-time identification of periodically parameter-varying state space models," *Automatica*, vol. 71, pp. 254 – 263, 2016.

[4] V. Laurain, R. Tóth, W.-X. Zheng, and M. Gilson, "Nonparametric identification of LPV models under general noise conditions: An LS-SVM based approach," in *Proc. of the 16th IFAC Symposium on System Identification*, Brussels, Belgium, 2012, pp. 1761–1766.

[5] R. Tóth, P. S. C. Heuberger, and P. M. J. Van den Hof, *Prediction-Error Identification of LPV Systems: Present and Beyond*. Boston, MA: Springer US, 2012, pp. 27–58.

[6] Y. Zhao, B. Huang, H. Su, and J. Chu, "Prediction error method for identification of LPV models," *Journal of Process Control*, vol. 22, no. 1, pp. 180 – 193, 2012.

[7] A. A. Bachnas, R. Tóth, J. H. A. Ludlage, and A. Mesbah, "A review on data-driven linear parameter-varying modeling approaches: A high-purity distillation column case study," *Journal of Process Control*, vol. 24, no. 4, pp. 272–285, 2014.

[8] R. Tóth, *Modeling and identification of linear parameter-varying systems*, ser. Lecture notes in control and information sciences. Germany: Springer, 2010.

[9] J. D. Caigny, J. F. Camino, and J. Swevers, "Interpolation-Based Modeling of MIMO LPV Systems," *IEEE Transactions on Control Systems Technology*, vol. 19, no. 1, pp. 46–63, 2011.

[10] H. Pfifer and S. Hecker, "Generation of optimal linear parametric models for LFT-based robust stability analysis and control design," in *Proc. of the 47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008.

[11] D. Petersson and J. Löfberg, "Optimization based LPV-approximation of multi-model systems," in *Proc. of the European Control Conference*, Budapest, Hungary, Aug 2009, pp. 3172–3177.

[12] J. S. Shamma and J. R. Cloutier, "Gain-scheduled missile autopilot design using linear parameter varying transformations," *Journal of Guidance, Control, and Dynamics*, vol. 16, no. 2, pp. 256–263, 1993.

[13] D. J. Leith and W. E. Leithead, "Gain-scheduled and nonlinear systems: Dynamic analysis by velocity-based linearization families," *International Journal of Control*, vol. 70, no. 2, pp. 289–317, 1998.

[14] Weehong Tan, A. K. Packard, and G. J. Balas, "Quasi-LPV modeling and LPV control of a generic missile," in *Proc. of the American Control Conference*, Chicago, IL, USA, Jun 2000, pp. 3692–3696.

[15] A. Kwiatkowski, M. Boll, and H. Werner, "Automated generation and assessment of affine LPV models," in *Proc. of the 45th IEEE Conference on Decision and Control*, San Diego, CA, USA, Dec 2006, pp. 6690–6695.

[16] H. Abbas, R. Tóth, M. Petreczky, N. Meskin, and J. Mohammadpour, "Embedding of nonlinear systems in a linear parameter-varying representation," in *Proc. of the 19th IFAC World Congress*, Cape Town, South Africa, Aug 2014, pp. 6907–6913.

[17] M. Schoukens and R. Tóth, "Linear parameter varying representation of a class of MIMO nonlinear systems," in *Proc. of the 2nd IFAC Workshop on Linear Parameter Varying Systems*, Florianopolis, Brazil, Sep 2018, pp. 94 – 99.

[18] R. Robles, A. Sala, and M. Bernal, "Performance-oriented quasi-LPV modeling of nonlinear systems," *International Journal of Robust and Nonlinear Control*, vol. 29, no. 5, pp. 1230–1248, 2019.

[19] A. Sadeghzadeh, B. Sharif, and R. Tóth, "Affine linear parameter-varying embedding of nonlinear models with improved accuracy and minimal overbounding," *IET Control Theory and Applications*, vol. 14, no. 20, pp. 3363–3373, 2020.

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

[21] A. Sadeghzadeh and R. Tóth, "Linear parameter-varying embedding of nonlinear models with reduced conservativeness," in *Proc. of the 21st IFAC World Congress*, Berlin, Germany, 2020.

[22] H. Nijmeijer and A. van der Schaft, *Nonlinear dynamical control systems*. New York, Berlin, Heidelberg: Springer Verlag, 1990.

[23] M. Henson and D. Seborg, *Nonlinear process control*. Prentice Hall, Englewood Cliffs, NJ, USA: Prentice Hall, 1998.

[24] M. Sato and D. Peaucelle, "Gain-scheduled output-feedback controllers using inexact scheduling parameters for continuous-time LPV systems," *Automatica*, vol. 49, no. 4, pp. 1019–1025, 2013.

[25] J. Daafouz, J. Bernussou, and J. C. Geromel, "On Inexact LPV Control Design of Continuous-Time Polytopic Systems," *IEEE Transactions on Automatic Control*, vol. 53, no. 7, pp. 1674–1678, aug 2008.

[26] P. Apkarian, P. Gahinet, and G. Becker, "Self-scheduled H_∞ control of linear parameter-varying systems: a design example," *Automatica*, vol. 31, no. 9, pp. 1251–1261, 1995.

[27] R. C. L. F. Oliveira and P. L. D. Peres, "Parameter-Dependent LMIs in Robust Analysis: Characterization of Homogeneous Polynomially Parameter-Dependent Solutions Via LMI Relaxations," *IEEE Transactions on Automatic Control*, vol. 52, no. 7, pp. 1334–1340, 2007.

[28] A. Sadeghzadeh, "On exploiting inexact scheduling parameters for gain-scheduled control of linear parameter-varying discrete-time systems," *Systems & Control Letters*, vol. 117, pp. 1 – 10, 2018.

[29] J. De Caigny, J. F. Camino, R. C. L. F. Oliveira, P. L. D. Peres, and J. Swevers, "Gain-scheduled dynamic output feedback control for discrete-time LPV systems," *International Journal of Robust and Nonlinear Control*, vol. 22, no. 5, pp. 535–558, 2012.

[30] A. Sadeghzadeh, "Gain-scheduled continuous-time control using polytope-bounded inexact scheduling parameters," *International Journal of Robust and Nonlinear Control*, vol. 28, no. 17, pp. 5557–5574, 2018.

[31] A. Sadeghzadeh, "LMI relaxations for robust gain-scheduled control of uncertain linear parameter varying systems," *IET Control Theory and Applications*, vol. 13, no. 4, pp. 486–495, 2019.

[32] J. C. Cockburn and B. G. Morton, "Linear fractional representations of uncertain systems," *Automatica*, vol. 33, no. 7, pp. 1263 – 1271, 1997.

[33] A. Packard, "Gain scheduling via linear fractional transformations," *Systems & Control Letters*, vol. 22, no. 2, pp. 79 – 92, 1994.

[34] A. Hjartarson, P. Seiler, and A. Packard, "LPVTools: A toolbox for modeling, analysis, and synthesis of parameter varying control systems," in *Proc. of the 1st workshop on Linear Parameter Varying*, Grenoble, France, Oct 2015, pp. 139–145.

[35] D. L. Donoho and P. B. Stark, "Uncertainty principles and signal recovery," *SIAM Journal on Applied Mathematics*, vol. 49, no. 3, pp. 906–931, 1989.

[36] D. L. Donoho and B. F. Logan, "Signal recovery and the large sieve," *SIAM Journal on Applied Mathematics*, vol. 52, no. 2, pp. 577–591, 1992.

[37] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing Sparsity by Reweighted l_1 Minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, dec 2008.

[38] C. Hoffmann, "Linear parameter-varying control of systems of high complexity," Ph.D. Thesis, Hamburg University of Technology, Hamburg, Germany, 2015.

[39] G. Vinnicombe, "Frequency domain uncertainty and the graph topology," *IEEE Transactions on Automatic Control*, vol. 38, no. 9, pp. 1371–1383, 1993.

[40] T. Bloemers and R. Tóth, "Equations of motion of a control moment gyroscope," Eindhoven University of Technologys, Tech. Rep., 2019.

[41] F. Wu, "Control of linear parameter varying systems," Ph.D. dissertation, University of California at Berkeley, 1995.

[42] J. Löfberg, "YALMIP: A Toolbox for Modeling and Optimization in MATLAB," in *Proc. of the IEEE International Conference on Robotics and Automation*, New Orleans, LA, USA, 2004, pp. 284–289.

[43] C. M. Agulhari, R. C. L. F. de Oliveira, and P. L. D. Peres, "Robust LMI Parser: a computational package to construct LMI conditions for uncertain systems," in *Proc. of the XIX Brazilian Conference on Automation*, Campina Grande, PB, Brazil, 2012, pp. 2298–2305.

[44] MOSEK ApS, *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015. [Online]. Available: http://docs.mosek.com/7.1/toolbox/index.html