

Linear Parameter Varying Representation of a class of MIMO Nonlinear Systems

Maarten Schoukens* Roland Tóth*

* *Control Systems, Eindhoven University of Technology, Eindhoven,
The Netherlands (e-mail: m.schoukens@tue.nl, r.toth@tue.nl).*

Abstract: Linear parameter-varying (LPV) models form a powerful model class to analyze and control a (nonlinear) system of interest. Identifying an LPV model of a nonlinear system can be challenging due to the difficulty of selecting the scheduling variable(s) a priori, especially if a first principles based understanding of the system is unavailable. Converting a nonlinear model to an LPV form is also non-trivial and requires systematic methods to automate the process. Inspired by these challenges, a systematic LPV embedding approach starting from multiple-input multiple-output (MIMO) linear fractional representations with a nonlinear feedback block (NLFR) is proposed. This NLFR model class is embedded into the LPV model class by an automated factorization of the (possibly MIMO) static nonlinear block present in the model. As a result of the factorization, an LPV-LFR or an LPV state-space model with affine dependency on the scheduling is obtained. This approach facilitates the selection of the scheduling variable and the connected mapping of system variables. Such a conversion method enables to use nonlinear identification tools to estimate LPV models. The potential of the proposed approach is illustrated on a 2-DOF nonlinear mass-spring-damper example.

Keywords: Nonlinear Systems, Linear-Parameter Varying Systems, System Identification, LPV Embedding, Linear Fractional Representation, MIMO

1. INTRODUCTION

The Linear-Parameter Varying (LPV) framework offers a powerful tool set to model and control nonlinear systems (Mohammadpour and Scherer, 2012). Many control applications depend on the availability of high-quality LPV models, fueling the need for LPV identification algorithms. LPV identification has been studied in detail (Tóth, 2010; dos Santos et al., 2011) in the past. However, in the majority of these works it is assumed that the scheduling signals are known a priori, relying on the user's expertise to design them for the considered system. Furthermore, often the noise in the measured version of the scheduling signals are left unattended in LPV identification, leading to a possible bias of the estimates (for approaches to handle this see (Piga et al., 2015)).

Embedding nonlinear models into the LPV model class offers an alternative approach to obtain LPV models of nonlinear systems without having to identify an LPV model directly. This avoids the selection of appropriate scheduling signals during LPV identification: the scheduling signal(s) are obtained as a result of the nonlinear embedding. Although embedding of nonlinear systems into the LPV framework is a popular approach in nonlinear systems control, only a few systematic embedding methods are discussed in the literature (Chisci et al., 2003; Tóth,

* Maarten Schoukens is supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie Fellowship (grant agreement nr. 798627). Furthermore, R. Tóth is supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement nr. 714663).

2010; Young, 2011; Mohammadpour and Scherer, 2012; Abbas et al., 2014, 2017).

LPV embedding of NLFR structures, sometimes called Lur'e systems, has been studied in prior works (Seron and De Doná, 2015; Hanafi et al., 2018; Schoukens and Tóth, 2018). However, these works are limited to single-input-single-output static nonlinearities. This paper presents a systematic embedding approach for MIMO nonlinear systems represented by NLFRs with a MIMO static nonlinear block (see Figure 1) (Schoukens and Tiels, 2017). It is shown how one can embed this model in an automated and systematic way into a MIMO LPV representation such that an LPV-LFR, or alternatively an affine state-space LPV model results, without introducing new singularity points in the representation.

The considered MIMO NLFR system class is discussed in Section 2. Section 3 discusses the embedding of these nonlinear systems into an LPV representation using a systematic embedding algorithm. Finally, a 2-DOF nonlinear mass-spring-damper system is analyzed to illustrate the developed embedding process.

2. NLFR SYSTEM CLASS

In this work, we considered the class of nonlinear systems that can be represented by a MIMO continuous-time NLFR. The NLFR system class is a general system class comprising a, possibly MIMO, localized static nonlinearity (see Figure 1). Identification of such NLFR structures is considered in (Vandersteen and Schoukens, 1999; Hsu et al., 2008; Novara et al., 2011; Vanbeylen, 2013). General-

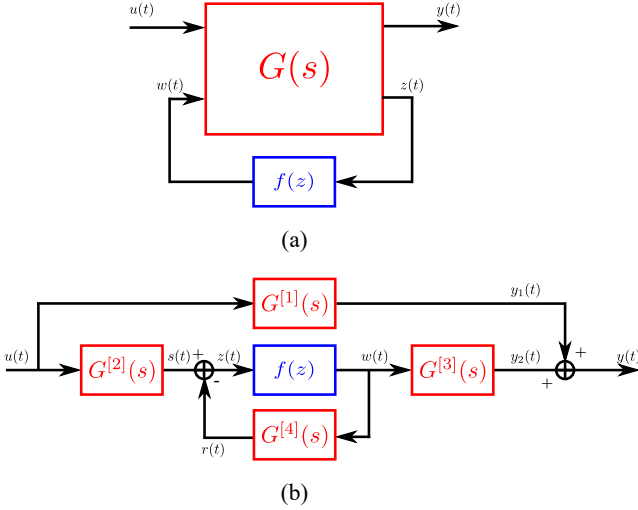


Fig. 1. The NLFR structure represented by (a) a MIMO LTI block G , and (b) an equivalent block-oriented structure with 4 MIMO LTI blocks $G^{[i]}$ and the MIMO static nonlinear function $f(z)$.

ity of this representation is based upon the fact that widely used block-oriented Hammerstein, Wiener and Wiener-Hammerstein structures are special cases of the NLFR structure (see (Giri and Bai, 2010; Schoukens and Tiels, 2017) for more information on the identification of block-oriented structures and some of their applications).

The input-output relation of a continuous-time MIMO NLFR can be expressed via a (minimal) state-space form as:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + [B_w \ B_u] \begin{bmatrix} w(t) \\ u(t) \end{bmatrix}, \\ \begin{bmatrix} z(t) \\ y(t) \end{bmatrix} &= \begin{bmatrix} C_z \\ C_y \end{bmatrix} x(t) + \begin{bmatrix} D_{zw} & D_{zu} \\ D_{yw} & D_{yu} \end{bmatrix} \begin{bmatrix} w(t) \\ u(t) \end{bmatrix}, \\ w(t) &= f(z(t)), \end{aligned} \quad (1)$$

where $u(t) \in \mathbb{R}^{n_u \times 1}$ is the input of the NLFR, $w(t) \in \mathbb{R}^{n_w \times 1}$ is the output of the static nonlinearity, $x(t) \in \mathbb{R}^{n_x \times 1}$ are the system states, $y(t) \in \mathbb{R}^{n_y \times 1}$ is the output of the NLFR, $z(t) \in \mathbb{R}^{n_z \times 1}$ is the input of the static nonlinearity, $f(z(t))$, $\mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_w}$ is a multi-dimensional, static and bounded nonlinear function, and A, \dots, D_{yu} are real constant matrices of appropriate dimensions.

The signal transfers corresponding to eq. (1) can be also equivalently represented by MIMO LTI transfer matrices $G^{[i]}$:

$$\begin{aligned} G^{[1]}(s) &= C_y(sI - A)^{-1}B_u + D_{yu}, \\ G^{[2]}(s) &= C_z(sI - A)^{-1}B_u + D_{zu}, \\ G^{[3]}(s) &= C_y(sI - A)^{-1}B_w + D_{yw}, \\ G^{[4]}(s) &= C_z(sI - A)^{-1}B_w + D_{zw}, \end{aligned} \quad (2)$$

where $s \in \mathbb{C}$ denotes the complex frequency (Laplace variable).

Substituting $z(t)$ into (1) results in:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) \\ &\quad + B_w f(C_z x(t) + D_{zu} u(t) + D_{zw} w(t)), \\ y(t) &= C_y x(t) + D_{yu} u(t) \\ &\quad + D_{yw} f(C_z x(t) + D_{zu} u(t) + D_{zw} w(t)), \\ w(t) &= f(C_z x(t) + D_{zu} u(t) + D_{zw} w(t)). \end{aligned} \quad (3)$$

To simplify our problem setting the following assumption is taken:

Assumption 1. $G^{[4]}$ is strictly proper (in other words, $D_{zw} = 0$).

Under Assumption 1, it is possible to eliminate $w(t)$. This results in the following simplified expression:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u u(t) + B_w f(C_z x(t) + D_{zu} u(t)), \\ y(t) &= C_y x(t) + D_{yu} u(t) + D_{yw} f(C_z x(t) + D_{zu} u(t)). \end{aligned} \quad (4)$$

Analogously, similar expressions can be obtained in discrete-time form in a straightforward manner.

Assumption 1 forces the nonlinearity $f(z)$ to be explicit. The presence of a direct feedthrough term D_{zw} would allow the nonlinearity to present itself in as an implicit function.

3. LPV EMBEDDING

3.1 Embedding Concept

The static nonlinearity that is present in the nonlinear model needs to be factorized to obtain an LPV representation. Multiple factorization approaches are possible. Here, the nonlinear function $f(z(t))$ is decomposed as $\bar{f}(z(t))z(t) + c$.

Denote \mathbb{P} as the set of values that can be reached by $z(t)$ for a specified class of inputs $u(t) \in \mathbb{U}$ with $t \in [0, \infty)$ and initial conditions $x(0) \in \mathbb{X}$.

Assumption 2. The static nonlinearity $f(z(t))$ can be represented as: $\bar{f}(z(t))z(t) + c$, such that $\bar{f}(z(t))$ does not contain singular points for $z(t) \in \mathbb{P}$, and c is finite.

This assumption excludes for instance functions $f(z(t))$ that have singularities in the region of interest (e.g. $\frac{1}{z}$, if $z = 0$ lies within the range of interest). Note that decomposition of a function in the form given in Assumption 2 is non-unique. The function $\bar{f}(z(t))$, $\mathbb{R}^{n_z} \rightarrow \mathbb{R}^{n_w \times n_z}$ is called the scheduling map in the sequel.

3.2 Constant Offset

Assumption 3. All linear subsystems $G^{[i]}$ (see Eq. (2)) are bounded-input bounded-output stable.

Assumption 3 is only required for the offset propagation algorithm. If no constant offset ($f(0) = 0$) is present in the static nonlinearity, this assumption is not required. Note that this assumption does not impose stability of the total NLFR system.

Under Assumptions 2 and 3, the NLFR structure with $D_{zw} = 0$ can be represented by:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B_u \tilde{u}(t) + B_w \tilde{f}(C_z x(t) + D_{zu} \tilde{u}(t)), \\ \tilde{y}(t) &= C_y x(t) + D_{yu} \tilde{u}(t) + D_3 \tilde{f}(C_z x(t) + D_{zu} \tilde{u}(t)), \end{aligned} \quad (5)$$

where:

$$\begin{aligned}\tilde{f}(z(t)) &= \bar{f}(z(t))z(t), \\ \tilde{u}(t) &= u(t) - G_0^{[2]-1}G_0^{[4]}c, \\ \tilde{y}(t) &= y(t) - \left(G_0^{[3]} + G_0^{[1]}G_0^{[2]-1}G_0^{[4]}\right)c,\end{aligned}\quad (6)$$

where $G_0^{[i]}$ is obtained by evaluating the steady-state gain of $G^{[i]}(s)$, i.e. $G_0^{[i]} = \lim_{s \rightarrow 0} G^{[i]}(s)$. Eq. (7) to hold, all $G_0^{[2]-1}G_0^{[4]}$ and $\left(G_0^{[3]} + G_0^{[1]}G_0^{[2]-1}G_0^{[4]}\right)$ should exist and be finite. In the case $n_z \leq n_u$ and the rank of $G_0^{[2]}$ is equal to n_z , the Moore-Penrose left-inverse of the matrix $G_0^{[2]}$ can be used (Golub and Van Loan, 1996). This is however a conservative requirement. It suffices that the following assumption holds:

Assumption 4. The vector $G_0^{[4]}c$ is part of the column space of the matrix $G_0^{[2]}$.

Under Assumption 4 one can always find a vector d such that $G_0^{[2]}d = G_0^{[4]}c$, resulting in the following offset mitigation equations:

$$\begin{aligned}\tilde{u}(t) &= u(t) - d, \\ \tilde{y}(t) &= y(t) - G_0^{[3]}c - G_0^{[1]}d,\end{aligned}\quad (7)$$

Note that Assumption 4 is only required for the offset propagation algorithm.

The constant offset at both the input and the output can be dealt with during the LPV control design process as a disturbance or by using input or output trimming methods.

3.3 Factorization

The time-dependent notation $z(t)$ is simplified to z to lighten the notation in this section.

This subsection introduces a systematic approach to perform the factorization $\tilde{f}(z) = \bar{f}(z)z$ for $z \in \mathbb{P}$. One can write:

$$\begin{aligned}\tilde{f}(z) &= \tilde{f}(z_1, z_2, \dots, z_{n_z}) \\ &= \sum_{i=1}^{n_z} \bar{f}_i(z_1, z_2, \dots, z_i)z_i.\end{aligned}\quad (8)$$

Assumption 5. All the first order partial derivatives:

$$\left. \frac{\partial \tilde{f}(z)}{\partial z_i} \right|_{z=(z_1, \dots, z_{i-1}, 0, \dots, 0)} \quad (9)$$

of $\tilde{f}(z)$, $z \in \mathbb{P}$ exist.

This assumption holds, for example, for all continuously differentiable functions $C_1^{n_w}(\mathbb{R}^{n_z})$.

Under assumptions 2 and 5, the functions \bar{f}_i are given by:

$$\bar{f}_i(z_1, z_2, \dots, z_i) = \begin{cases} \frac{\tilde{f}(z_1, \dots, z_i, 0, \dots, 0) - \tilde{f}(z_1, \dots, z_{i-1}, 0, \dots, 0)}{z_i} & \text{if } z_i \neq 0 \\ \left. \frac{\partial \tilde{f}(z_1, \dots, z_i, 0, \dots, 0)}{\partial z_i} \right|_{z=(z_1, \dots, z_{i-1}, 0, \dots, 0)} & \text{if } z_i = 0 \end{cases} \quad (10)$$

and

$$\bar{f}_1(z_1) = \begin{cases} \frac{\tilde{f}(z_1, 0, \dots, 0)}{z_1} & \text{if } z_1 \neq 0 \\ \left. \frac{\partial \tilde{f}(z_1, 0, \dots, 0)}{\partial z_1} \right|_{z=0} & \text{if } z_1 = 0 \end{cases} \quad (11)$$

Theorem 1. The function $\bar{f}_i(z_1, \dots, z_i)$ is continuous at all $z \in \mathbb{P}$ if $\tilde{f}(z)$ is continuous at all $z = (z_1, \dots, z_{i-1}, 0, \dots, 0)$, $z \in \mathbb{P}$ and if Assumption 5 holds.

Proof 1. The quotient of two continuous functions $\frac{h_1(z)}{h_2(z)}$ is continuous everywhere, except in the zeros of $h_2(z)$.

The function $\bar{f}_i(z_1, \dots, z_i)$ is given by Eq. (10), where both the numerator and the denominator are formed by continuous functions by assumption. Hence, $\bar{f}_i(z_1, \dots, z_i)$ is continuous everywhere, except in the point $z_i = 0$.

For $\bar{f}_i(z_1, \dots, z_i)$ to be continuous in $z_i = 0$ we need that:

$$\lim_{z_i \rightarrow 0} \bar{f}_i(z_1, \dots, z_i) = \bar{f}_i(z_1, \dots, z_{i-1}, 0) \quad (12)$$

and Eq. (13) should be finite. It is easy to observe that:

$$\begin{aligned}\lim_{z_i \rightarrow 0} \bar{f}_i(z_1, \dots, z_i) \\ = \left. \frac{\partial \tilde{f}(z_1, \dots, z_i, 0, \dots, 0)}{\partial z_i} \right|_{z=(z_1, \dots, z_{i-1}, 0, \dots, 0)}\end{aligned}\quad (13)$$

which exists and is finite (see Assumption 5).

Hence, $\bar{f}_i(z_1, \dots, z_i)$ is continuous everywhere and no singularities are introduced by the factorization. ■

It can be observed that for a polynomial (or rational) function $\tilde{f}(z)$, also the functions $\bar{f}_i(z_1, z_2, \dots, z_i)$ are polynomial (rational) (remember that $\tilde{f}(z)$ never contains a constant term: $\tilde{f}(0) = 0$).

In case the function $\tilde{f}(z)$ is not partially differentiable in the points of interest, but its left and right derivative are finite in those points, the partial derivatives can be replaced by a finite constant without introducing a singularity in $\tilde{f}(z)$. However, the continuity of $\tilde{f}(z)$ is lost.

The proposed factorization is not unique. By changing the order in which the variables z_i are considered, different factorizations of the multivariate nonlinearity could be obtained.

3.4 Embedded Representation

Under Assumption 2 we can represent the NLFR structure as an LPV affine state-space representation with an additional constant offset at the input and/or output:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B_u \tilde{u}(t) + B_w p(t) (C_z x(t) + D_{zu} \tilde{u}(t)), \\ \tilde{y}(t) &= C_y x(t) + D_{yu} \tilde{u}(t) + D_{yw} p(t) (C_z x(t) + D_{zu} \tilde{u}(t)),\end{aligned}\quad (14)$$

where $p(t)$ is given by the scheduling map:

$$p(t) = \bar{f}(z(t)) = \bar{f}(C_z x(t) + D_{zu} \tilde{u}(t)). \quad (15)$$

This results in the following affine state-space LPV structure:

$$\begin{aligned}\dot{x}(t) &= (A + A_p(p(t)))x(t) + (B_u + B_p(p(t)))\tilde{u}(t), \\ \tilde{y}(t) &= (C_y + C_p(p(t)))x(t) + (D_{yu} + D_p(p(t)))\tilde{u}(t),\end{aligned}\quad (16)$$

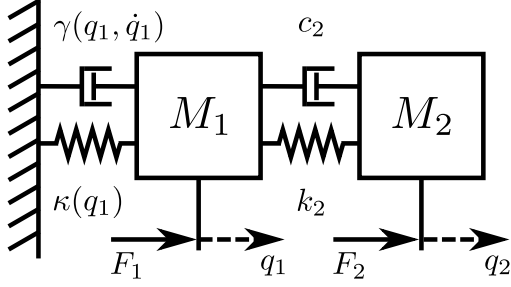


Fig. 2. The considered 2 DOF mass-spring-damper system where y_i denotes the position, and F_i the force applied on mass M_i .

with:

$$\begin{aligned} A_p(p(t)) &= B_w p(t) C_z, & C_p(p(t)) &= D_{yw} p(t) C_z, \\ B_p(p(t)) &= B_w p(t) D_{zu}, & D_p(p(t)) &= D_{yw} p(t) D_{zu}. \end{aligned} \quad (17)$$

Note that the expressions in Eq. (17) are linear in $p(t)$.

3.5 Remarks

The resulting affine state-space LPV representation is not unique, a state transformation can be introduced. Also the latent variables $z(t)$ and $w(t)$ are non-uniquely defined, similar to the non-uniqueness that is present in block-oriented systems (Schoukens et al., 2015; Schoukens, 2015). This non-uniqueness can be further explored for the scheduling variables selection. From a control design point of view it is desirable for the scheduling signals $p(t)$ or the input of the static nonlinearity $z(t)$ to be measurable or observable. This will be explored in future research.

Note that the proposed algorithm, including the constant offset removal and the nonlinear function factorization can be completely automated starting from a NLFR representation. However, in some cases it can be worthwhile to search for an 'optimal' factorization. As discussed above, multiple factorizations of a multivariate nonlinear function are possible. Hence, the search for an 'optimal' factorization is more difficult to automate and requires more study.

4. SIMULATION EXAMPLE: 2-DOF NONLINEAR MASS-SPRING-DAMPER SYSTEM

4.1 Nonlinear System Equations

A 2-DOF mass-spring-damper system with unit masses ($M_1 = M_2 = 1$ in Figure 2) a nonlinear spring $\kappa(y_1)$ and nonlinear position-dependent damping $\gamma(y_1, \dot{y}_1)$ is considered:

$$\begin{aligned} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \end{bmatrix} + \begin{bmatrix} k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \\ + \begin{bmatrix} \gamma(q_1, \dot{q}_1) + \kappa(q_1) \\ 0 \end{bmatrix} &= \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \end{aligned} \quad (18)$$

where the nonlinear damping and spring are given by:

$$\begin{aligned} \gamma(q_1, \dot{q}_1) &= c_1 \dot{q}_1 + c_3 \sin(c_4 q_1) \dot{q}_1 + c_5 \dot{q}_1^2 \\ \kappa(q_1) &= k_1 q_1 + k_3 q_1^3, \end{aligned} \quad (19)$$

with $k_1 = \pi^2$, $k_2 = (1.2\pi)^2$, $k_3 = 10$, $c_1 = 0.1$, $c_2 = 0.01$, $c_3 = 0.1$, $c_4 = 10$, $c_5 = 0.2$.

4.2 NLFR Representation

A NLFR representation can be obtained using the notation as in Eq. (1):

$$\begin{aligned} \dot{x} &= Ax + [B_w \ B_u] \begin{bmatrix} w \\ u \end{bmatrix}, \\ \begin{bmatrix} z \\ y \end{bmatrix} &= \begin{bmatrix} C_z \\ C_y \end{bmatrix} x, \end{aligned} \quad (20)$$

where

$$\begin{aligned} u &= \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}, & y &= \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}, \\ x &= [q_1 \ q_2 \ \dot{q}_1 \ \dot{q}_2]^\top, \\ z &= \begin{bmatrix} x_1 \\ x_3 \end{bmatrix}, \\ w &= f(z) \\ &= c_3 \sin(c_4 z_1) z_2 + c_5 z_2^2 + k_3 z_1^3, \end{aligned} \quad (21)$$

and

$$\begin{aligned} A &= \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -k_1 - k_2 & k_2 & -c_1 - c_2 & c_2 \\ k_2 & -k_2 & c_2 & -c_2 \end{bmatrix}, \\ B_w &= \begin{bmatrix} 0 \\ 0 \\ -1 \\ 0 \end{bmatrix}, & B_u &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \\ C_z &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, & C_y &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \end{aligned} \quad (22)$$

where z_i denotes the i -th entry from the vector z and the matrices D_{zw} , D_{zu} , D_{yw} , D_{yu} are all zero matrices.

4.3 LPV Embedding

It can be observed that no offset handling is required since it holds for the nonlinearity in the NLFR that $f(0) = 0$.

Multiple factorizations are possible using the factorization scheme presented in this paper. By first factorizing z_1 , and z_2 next, we obtain:

$$\begin{aligned} f(z) &= \bar{f}(z)z = [\bar{f}_1(z_1) \ \bar{f}_2(z_1, z_2)] \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \\ &= \begin{bmatrix} k_3 z_1^2 \\ c_3 \sin(c_4 z_1) + c_5 z_2 \end{bmatrix}^\top \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}. \end{aligned} \quad (23)$$

Note that the functions $k_3 z_1^2$ and $c_3 \sin(c_4 z_1) + c_5 z_2$ are the resulting scheduling maps of the LPV embedding.

Another possibility (first factorize z_2 , and z_1 next) is:

$$f(z) = \begin{bmatrix} k_3 z_1^2 + c_3 \text{sinc}(c_4 z_1) z_2 \\ c_5 z_2 \end{bmatrix}^\top \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}. \quad (24)$$

Many more factorizations are possible, beyond the ones obtained with the proposed factorization scheme. For instance:

$$f(z) = \begin{bmatrix} k_3 z_1^2 + (1-\alpha) c_3 \text{sinc}(c_4 z_1) z_2 \\ \alpha c_3 \text{sinc}(c_4 z_1) + c_5 z_2 \end{bmatrix}^\top \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} \quad (25)$$

for any finite α results in an equivalent representation.

Using the choice (23), the equivalent LPV representation of the considered system is given by:

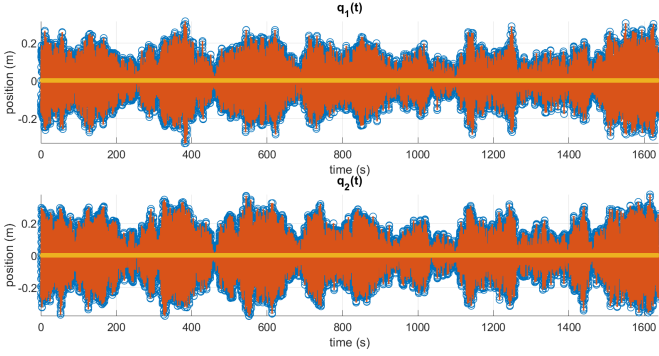


Fig. 3. Time domain output: the NLFR output, $y_1 = q_1$ and $y_2 = q_2$, (blue circles) coincides perfectly with the embedded NLFR output (red line), as is illustrated by the residual error (orange) that is equal to zero or below Matlab precision.

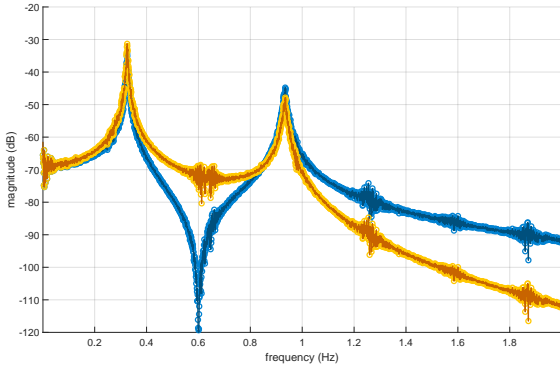


Fig. 4. Frequency domain output ($y_1 = q_1$ in blue, and $y_2 = q_2$ in orange): the NLFR output (circles, lighter color) coincides perfectly with the embedded NLFR output (line, darker color)

$$\begin{aligned} \dot{x} &= Ax + A_p(p(t))x + B_u u, \\ y &= C_y x, \end{aligned} \quad (26)$$

where

$$A_p(p(t)) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -p_1(t) & 0 & -p_2(t) & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad (27)$$

with $p_1(t) = \bar{f}_1(q_1(t))$, $p_2(t) = \bar{f}_2(q_1(t), \dot{q}_1(t))$ provided in Eq. (23).

4.4 Results

The output responses of the NLFR representation and by the embedding obtained LPV representation are depicted in the time- and frequency domain in Figures 3 and 4 respectively. A perfect match between both can be observed.

The scheduling signals p are obtained by applying the scheduling map on the trajectories x and u obtained by the nonlinear simulation.

The system nonlinearity that is present in the NLFR representation (see Figure 5) is factorized in 2 nonlinear functions $\bar{f}_1(z_1)$, $\bar{f}_2(z_1, z_2)$ shown in Figures 6 and 7 respectively.

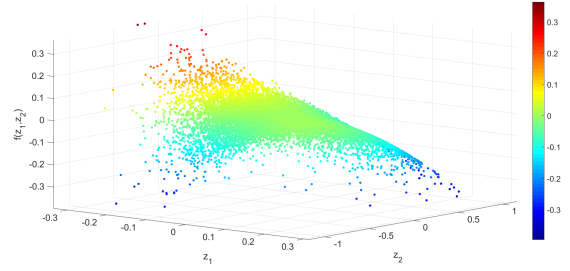


Fig. 5. Surface plot of the original nonlinearity: $c_3 \sin(c_4 q_1) \dot{q}_1 + c_5 \dot{q}_1^2 + k_3 q_1^3$.

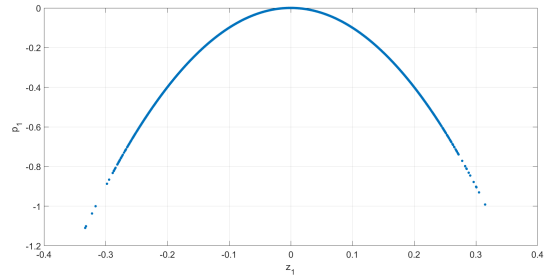


Fig. 6. The scheduling map (factorized nonlinearity $\bar{f}_1(z_1)$) of p_1 in the LPV representation.

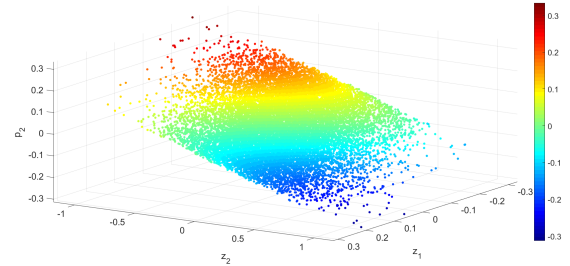


Fig. 7. The scheduling map (factorized nonlinearity $\bar{f}_2(z_1, z_2)$) of p_2 in the LPV representation.

5. CONCLUSION

This paper demonstrates how the class of MIMO NLFR systems can be exactly represented by an affine state-space LPV model under 3 mild assumptions. A systematic and automated embedding procedure is proposed for the underlying conversion problem. By this procedure an NLFR system is embedded in an LPV representation without introducing any new singularities. The effectiveness of the proposed approach is illustrated on a 2-DOF nonlinear mass-spring-damper example.

Further research will explore the optimal selection of the scheduling map based upon the various possibilities provided by the factorization, by taking into account the control objectives in the modeling process, in terms of achievable performance and measures of conservativeness.

REFERENCES

- Abbas, H., Hanema, J., Tóth, R., Mohammadpour, J., and Meskin, N. (2017). An improved robust model predictive control for linear parameter-varying input-

- output models. *International Journal of Robust and Nonlinear Control*, 28(3), 859–880.
- Abbas, H., Tóth, R., Petreczky, M., Meskin, N., and Mohammadpour, J. (2014). Embedding of nonlinear systems in a linear parameter-varying representation. In *19th World Congress of the International Federation of Automatic Control*, 6907–6913. Cape Town, South Africa.
- Chisci, L., Falugi, P., and Zappa, G. (2003). Gain-scheduling MPC of nonlinear systems. *International Journal of Robust and Nonlinear Control*, 13(3-4), 295–308.
- dos Santos, P., Azevedo Perdicoulis, T., Novara, V., Ramos, J., and Rivera, D. (eds.) (2011). *Linear Parameter-Varying System Identification: New Developments and Trends*. World Scientific, Singapore.
- Giri, F. and Bai, E. (eds.) (2010). *Block-oriented Nonlinear System Identification*, volume 404 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, London.
- Golub, G. and Van Loan, C. (1996). *Matrix Computations - Third Edition*. Johns Hopkins University Press, Baltimore and London.
- Hanafi, H., Seron, M., and De Doná, J. (2018). Fault estimation and controller compensation in lure systems by LPV-embedding. *International Journal of Control*, 0(0), 1–14.
- Hsu, K., Poolla, K., and Vincent, T.L. (2008). Identification of structured nonlinear systems. *IEEE Transactions on Automatic Control*, 53(11), 2497–2513.
- Mohammadpour, J. and Scherer, C. (eds.) (2012). *Control of Linear Parameter Varying Systems with Applications*. Springer-Verlag, New York.
- Novara, C., Vincent, T.L., Hsu, K., Milanese, M., and Poolla, K. (2011). Parametric identification of structured nonlinear systems. *Automatica*, 47, 711–721.
- Piga, D., Cox, P., Tóth, R., and Laurain, V. (2015). LPV system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53, 329 – 338.
- Schoukens, M. (2015). *Identification of Parallel Block-Oriented Models starting from the Best Linear Approximation*. Ph.D. thesis, Vrije Universiteit Brussel, Belgium.
- Schoukens, M., Marconato, A., Pintelon, R., Vandersteen, G., and Rolain, Y. (2015). Parametric identification of parallel Wiener-Hammerstein systems. *Automatica*, 51(1), 111–122.
- Schoukens, M. and Tiels, K. (2017). Identification of block-oriented nonlinear systems starting from linear approximations: A survey. *Automatica*, 85, 272–292.
- Schoukens, M. and Tóth, R. (2018). From nonlinear identification to linear parameter varying models: Benchmark examples. In *18th IFAC Symposium on system identification (SYSID)*. Stockholm, Sweden.
- Seron, M. and De Doná, J. (2015). On robust stability and set invariance of switched linear parameter varying systems. *International Journal of Control*, 88(12), 2588–2597.
- Tóth, R. (2010). *Modeling and Identification of Linear Parameter-Varying Systems*, volume 403 of *Lecture Notes in Control and Information Sciences*. Springer-Verlag, Berlin Heidelberg.
- Vanbeylen, L. (2013). Nonlinear LFR Block-Oriented Model: Potential Benefits and Improved, User-Friendly Identification Method. *IEEE Transactions on Instrumentation and Measurement*, 62(12), 3374–3383.
- Vandersteen, G. and Schoukens, J. (1999). Measurement and identification of nonlinear systems consisting of linear dynamic blocks and one static nonlinearity. *IEEE Transactions on Automatic Control*, 44(6), 1266–1271.
- Young, P. (2011). *Recursive Estimation and Time-Series Analysis: An Introduction for the Student and Practitioner*. Springer-Verlag, Berlin Heidelberg.