LPV System Local Model Interpolation Based on Combined Model Reduction

Qinghua Zhang

LPV System Local Model Interpolation Based on Combined Model Reduction

Qinghua Zhang [*]

* Inria, IFSTTAR, Université de Rennes, Campus de Beaulieu, 35042 Rennes Cedex, France. Email: qinghua.zhang@inria.fr

Abstract: The local approach to linear parameter varying (LPV) system identification consists in interpolating a collection of linear time invariant (LTI) models, which have been estimated from data acquired at different working points of a nonlinear system. Interpolation is essential in this approach. When the local LTI models are in state-space form, as each local model can be estimated with an arbitrary state basis, it is widely acknowledged that the local models should be made coherent before their interpolation. In order to avoid the delicate task of making local state-space models coherent, a new interpolation method of local state-space models is proposed in this paper, which does not require coherent local models. This method is based on the reduction of the large state-space model built by combining the local models. Numerical examples are presented to illustrate the effectiveness of this method.

Keywords: LPV system identification, local model interpolation, coherence of local state-space models.

1. INTRODUCTION

Linear parameter varying (LPV) models provide an effective approach to handling nonlinear control systems (Tóth, 2010; Mohammadpour and Scherer, 2012; Lopes dos Santos et al., 2012; Sename et al., 2013). In order to build LPV models, various system identification methods have been proposed (Van Wingerden and Verhaegen, 2009; Tóth, 2010; Mercere et al., 2011; Lopes dos Santos et al., 2011; Tóth et al., 2012; Zhao et al., 2012; Piga et al., 2015). *The local approach* to LPV system identification consists in interpolating a collection of linear time invariant (LTI) models, which have been estimated from data acquired at different working points of the underlying nonlinear system. Being valid around the corresponding working point only, such an LTI model is also referred to as a *local linear model*, or *local model* for short. Compared to the global approach, which builds LPV models by processing in one shot all the available data, the local appraoch is less demanding on computational ressources when building LPV models, and offers the possibility of refining an existing LPV model by adding more LTI models estimated at new working points.

Interpolation is essential in the local approach to LPV system identification. Depending on the form of local linear models, interpolation is a more or less tricky task. When input-output (I-O) local models are used, the linear coefficients of LTI models can be interpolated (Nemani et al., 1995; Bamieh and Giarré, 2002; Bolea et al., 2007; Martínez-González et al., 2009). Frequency domain local models can be interpolated in a similar way (Petersson and Löfberg, 2009). When state-space local models are used, however, as each local model can be estimated with an arbitrary state basis, it is necessary to make the local models "coherent" before their interpolation (De Caigny et al., 2011, 2014).

It is shown in (Zhang and Ljung, 2017) that, if local linear state-space models are estimated without any global structural assumption, the local models themselves do not contain the information to make them coherent for the purpose of their interpolation. This problem can be solved by introducing structural assumptions, but such a practice requires relevant physical knowledges about the underlying system. As pointed out in (Zhang and Ljung, 2017), it is also possible to build a coherent set of local state-space models from I-O data sequences involving sufficiently working point transitions, but then it is no longer a truly local approach.

Another possibility is to interpolate the outputs of the local models, instead of the local models themselves (Zhu and Xu, 2008; Zhu and Ji, 2009). With this method, the actual form of local models does not matter, and it is even possible to mix up local models of different forms. In particular, when local state-space models are used, there is no need to make them "coherent". However, the result of such an interpolation is *the output* of an LPV model, instead of the LPV model itself, which is never built with this method. It is thus necessary to repeat the interpolation when the system input changes. Moreover, it is not possible to analyze the underlying LPV model.

In this paper, a new method is proposed to interpolate local state-space models, *without requiring "coherent" local models*. It is thus of particular interest to applications requiring state-space LPV models. This method has been inspired by the interpolation of local model outputs, but it does lead to an LPV model in state-space form.

This paper is organized as follows. In Section 2 the considered problem is formulated. In Section 3 local model output interpolation is shortly introduced. In Section 4 the new interpolation method is presented. Numerical examples are presented in Section 5. Conclusions are drawn in Section 6.

2. PROBLEM FORMULATION

Consider a nonlinear system whose behavior can be appropriately approximated by linearized models at fixed working points. Assume that the working points are characterized by a variable $p \in \mathbb{R}$ that is accessible, directly or indirectly, from sensor signals. In the LPV system literature, p is usually called a *scheduling variable* or a *scheduling parameter*. There is no theoretical obstacle to generalize the result of this paper to vector valued p, however, in practice it is difficult to obtain a sufficient number of local models to be interpolated in a vector space. Typically these local models are obtained with classical system identification methods (Ljung, 1999), by processing data collected at different working points corresponding to different values of p. The higher is the dimension of p, the more local models are required so that their interpolation leads to a relevant LPV model.

Let

$$\mathbb{P} = \{p_1, p_2, \ldots, p_m\} \subset \mathbb{R} \qquad (1)$$

be a finite set of scheduling values corresponding to working points where local linear models in state-space form are available. For every scheduling value $p_i \in \mathbb{P}$, let the local state-space model be

$$\sigma_i : \begin{cases} x_i(t+1) = A_i x_i(t) + B_i u(t) + w_i(t) \\ \qquad y(t) = C_i x_i(t) + D_i u(t) + v_i(t), \end{cases} \qquad (2)$$

where $t = 0, 1, 2 \ldots$ is the discrete time index, $x_i(t) \in \mathbb{R}^n$ the state, $u(t) \in \mathbb{R}^q$ the input, $y(t) \in \mathbb{R}^s$ the output, $w_i(t) \in \mathbb{R}^n$ the state noise, $v_i(t) \in \mathbb{R}^s$ the output noise, and A_i, B_i, C_i, D_i are matrices of appropriate sizes. The set of available local models corresponding to the entries of $\mathbb{P}$ is denoted by

$$\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}. \qquad (3)$$

Let $\mathbb{S} \subset \mathbb{R}$ be an interval in which the local models are to be interpolated. The purpose of local model interpolation is to build an LTI model $\sigma(p)$ for every given value of $p \in \mathbb{S}$, based on the set Σ of available local models. The resulting model $\sigma(p)$ should be similar, in some sense, to those local models $\sigma_i \in \Sigma$ corresponding to the scheduling values p_i close to p.

A naive method for local model interpolation would be to choose a set of weighting functions $\rho_i : \mathbb{S} \to [0, 1]$, typically bell-shaped functions centered at $p = p_i \in \mathbb{P}$, such that

$$\sum_{i=1}^{m} \rho_i(p) = 1 \ \text{ for all } \ p \in \mathbb{S}, \qquad (4)$$

and then to interpolate the matrices A_i, B_i, C_i, D_i as weighted sums, *i.e.*,

$$A(p) = \sum_{i=1}^{m} \rho_i(p) A_i, \qquad (5)$$

and similarly for $B(p), C(p), D(p)$.

Such a naive method has a serious drawback: typically each of the local models σ_i, which have been estimated from local I-O data, corresponds to an arbitrary state basis, but the naive interpolation does not take into account this fact. In other words, each matrix A_i has been estimated up to an arbitrary similarity transformation $T_i A_i T_i^{-1}$ with an *unknown and arbitrary* invertible matrix $T_i \in \mathbb{R}^{n \times n}$, and similarly for B_i, C_i. As the matrix T_i differs for different

σ_i, the result of the naive interpolation strongly depends on the arbitrary matrices T_i.

It is widely acknowledged in the LPV system identification community that local state-space models should be made "coherent" before being interpolated. A simple idea is to put the available local models into some canonical form before their interpolation. For example, the control canonical form is used in (Steinbuch et al., 2003), the balanced form in (Lovera and Mercere, 2007), and a zero-pole decomposition-based form in (De Caigny et al., 2009, 2011). This practice assumes that the local models in the same canonical form are coherent. An important question then arises naturally: are these different forms of "coherent" local models compatible with each other?

It is recently reported in (Zhang and Ljung, 2017) that, if the local state-space models σ_i have been built without any global structural assumption, then these local models do not contain sufficient information to make themselves coherent. Global structural assumptions can help to solve this problem, but they should be based on physical knowledges, *i.e.*, the local models σ_i should be physically parametrized.

As it is not always possible to physically parametrize local models, other interpolation methods should be developed for building LPV models following the local approach.

3. OUTPUT INTERPOLATION

A simple method is to interpolate the outputs of the local models, instead of the local models themselves (Zhu and Xu, 2008; Zhu and Ji, 2009). Let $y_i(t)$ denote the output delivered by the local model σ_i. At every instant t, the outputs of the local models are first computed, then the output $\bar{y}(t)$ of the underlying LPV system at the same time instant can be computed as a weighted average:

$$\bar{y}(t) = \sum_{i=1}^{m} \rho_i(p(t)) y_i(t) \qquad (6)$$

where the weighting functions $\rho_i(\cdot)$ are as those introduced in the previous section satisfying the constraint (4), and $p(t) \in \mathbb{S}$ is the scheduling value at instant t.

This method has the advantage of being applicable to local models of any form (state-space, transfer function, etc.). However, the interpolation must be made in real time, at every time instant t. Let ρ^* be the scheduling value at instant t, *i.e*, $\rho(t) = \rho^*$. Even if an interpolation has been made for the same scheduling value ρ^* at an earlier time instant $t' < t$, it has to be computed again at instant t, because the local model outputs $y_i(t)$ have changed.

Moreover, as this interpolation does not yield a local model, it is not possible to analyze the behavior of the local model, nor the underlying LPV model, which is never built with this method.

4. NEW LOCAL MODEL INTERPOLATION BASED ON COMBINED MODEL REDUCTION

The method proposed in this section has been inspired by the output interpolation introduced in the previous section, but it does result in a state-space model $\sigma(p)$ for any given scheduling value $p \in \mathbb{S}$, and remarkably, *there is no need* to make the local models $\sigma_i \in \Sigma$ coherent before their interpolation.

Let $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_m\}$ be a set of local models to be interpolated, with each σ_i expressed in the state-space form (2), typically locally estimated from I-O data. Choose a set of weighting functions $\rho_i : \mathbb{S} \to [0,1]$, typically bell-shaped functions, satisfying (4).

The basic idea of the new interpolation method is based on the following large state-space model built by combining all the local models $\sigma_i \in \Sigma$:

$$x_1(t+1) = A_1 x_1(t) + B_1 u(t) + w_i(t) \tag{7a}$$
$$x_2(t+1) = A_2 x_2(t) + B_2 u(t) + w_i(t) \tag{7b}$$
$$\vdots$$
$$x_m(t+1) = A_m x_m(t) + B_m u(t) + w_i(t) \tag{7c}$$

$$y(t) = \sum_{i=1}^{m} \rho_i(p) \left[C_i x_i(t) + D_i u(t) + v_i(t) \right]. \tag{7d}$$

The state vector of this large system is a concatenation of the states of the local models σ_i, and its output is a weighted average of the outputs of the local models.

By viewing t as a dummy variable (it does not mean the current time instant, but serves to recursively define the states $x_i(t)$), this large set of equations defines an LTI state-space model for any specified value of p. This (large) model does not depend on the time t, other than via the dependence of p on t. Therefore, the combined state-space model (7), as (an intermediate step of) an interpolation of the local models $\sigma_i \in \Sigma$, is indeed an LTI model for any given $p \in \mathbb{S}$, unlike the result of the output interpolation introduced in the previous section.

At time instant t, the actual value of p is $p(t)$, then the output equation (7d) becomes

$$y(t) = \sum_{i=1}^{m} \rho_i(p(t)) \left[C_i x_i(t) + D_i u(t) + v_i(t) \right] \tag{8}$$
$$= \sum_{i=1}^{m} \rho_i(p(t)) y_i(t) \tag{9}$$

where $y_i(t)$ denotes the output of the local model σ_i. Then the resulting $y(t)$ computed from equation (7d) is equivalent to the result of output interpolation $\bar{y}(t)$, as expressed in (6).

Due to this equivalence, the combined (large) state-space model (7) is an interpolation of the local models σ_i.

However, this result has the drawback of yielding a large state-space model, of order mn. *For a given scheduling value p*, because the resulting model (7) is expected to be similar to the local models $\sigma_i \in \Sigma$ corresponding to scheduling values $p_i \in \mathbb{P}$ close to p, and each of these local models σ_i is of order n, the large model of order mn must have many insignificant states. Therefore, the interpolation procedure should be completed by reducing the order of the large state-space model (7).

The large state-space model (7) can be more compactly written as

$$\underline{x}(t+1) = \underline{A}\,\underline{x}(t) + \underline{B}\,u(t) + \underline{w}(t) \tag{10a}$$
$$y(t) = \underline{C}(p)\,\underline{x}(t) + \underline{D}(p)\,u(t) + \underline{v}(t), \tag{10b}$$

with

$$\underline{x}(t) \triangleq \begin{bmatrix} x_1(t) \\ \vdots \\ x_m(t) \end{bmatrix}, \quad \underline{w}(t) \triangleq \begin{bmatrix} w_1(t) \\ \vdots \\ w_m(t) \end{bmatrix}, \tag{11}$$

$$\underline{v}(t) \triangleq \sum_{i=1}^{m} \rho_i(p) v_i(t), \tag{12}$$

$$\underline{A} \triangleq \begin{bmatrix} A_1 & & \\ & \ddots & \\ & & A_m \end{bmatrix}, \quad \underline{B} \triangleq \begin{bmatrix} B_1 \\ \vdots \\ B_m \end{bmatrix}, \tag{13}$$

$$\underline{C}(p) \triangleq [\rho_1(p) C_1 \; \cdots \; \rho_m(p) C_m], \tag{14}$$

$$\underline{D}(p) \triangleq \sum_{i=1}^{m} \rho_i(p) D_i. \tag{15}$$

For every given scheduling value p, in principle, the state-space system (10) defined by the matrices $\underline{A}, \underline{B}, \underline{C}, \underline{D}$ of order mn can be reduced with any appropriate model reduction method. In what follows, the balanced reduction method will be adopted, due to its reliable numerical behavior.

Balanced reduction is a well known LTI model reduction method (Kailath, 1980; Moore, 1981). Let us shortly recall this method for completeness.

The controllability Gramian $W_c \in \mathbb{R}^{mn \times mn}$ and the observability Gramian $W_o \in \mathbb{R}^{mn \times mn}$ of the large system (10) are matrices satisfying respectively the equations

$$W_c + \underline{A} W_c \underline{A}^T = \underline{B}\underline{B}^T \tag{16}$$
$$W_o + \underline{A}^T W_o \underline{A} = \underline{C}^T(p)\underline{C}(p). \tag{17}$$

A linear transformation matrix $T \in \mathbb{R}^{mn \times mn}$ can be found so that the Gramian matrices of the transformed system, namely $\tilde{W}_c = T^{-1} W_c T^{-T}$ and $\tilde{W}_o = T^T W_o T$, are both diagonal, and equal to each other, except those entries non zero in one of the two matrices only (Glover, 1984; Varga, 1991). The square roots of the eigenvalues of $\tilde{W}_c \tilde{W}_o$ are known as *the Hankel singular values*. The balanced reduction consists in, after applying the linear transformation T to the state vector $\underline{x}(t)$, removing the states corresponding to the smallest Hankel singular values. In the present context, only the n states corresponding to the n largest Hankel singular values are kept, so that the resulting state-space model is of order n, like the local models $\sigma_i \in \Sigma$.

5. NUMERICAL EXAMPLES

As an illustration, let us consider an example of pipeline. In (Lopes dos Santos et al., 2011), an affine LPV model of a pipeline in a natural gas transportation network is established following the *global* LPV system identification approach. This LPV model is borrowed in this section for generating data by numerical simulation, in order to apply the proposed local model interpolation method.

The affine LPV model of a pipeline established in (Lopes dos Santos et al., 2011) is in the form of

$$x(t+1) = A(p)x(t) + B(p)u(t) + w(t) \tag{18a}$$
$$y(t) = C(p)x(t) + D(p)u(t) + v(t) \tag{18b}$$

where $u(t) = Q_i(t) \in \mathbb{R}$ is the input mass flow, $y(t) = Q_o(t) \in \mathbb{R}$ is the output mass flow, $x(t) \in \mathbb{R}^2$ is composed of the mass flow and the pressure drop within the first

section of the modeled pipe, and $p(t) \in \mathbb{R}$ is the sum of the intake pressure and offtake pressure, with its mean value translated to zero, the matrices $A(p), B(p), C(p), D(p)$ are parametrized in the affine form

$$A(p) = A_0 + A_p p \qquad (19a)$$
$$B(p) = B_0 + B_p p \qquad (19b)$$
$$C(p) = C_0 + C_p p \qquad (19c)$$
$$D(p) = D_0 + D_p p. \qquad (19d)$$

The parameter values contained in $A_0, A_p, B_0, B_p, C_0,$ C_p, D_0, D_p, are detailed in (Lopes dos Santos et al., 2011).

Example 1 – continuously varying scheduling variable

The input-output data and the scheduling variable used in this example are plotted in Figure 1. The profiles of the input $u(t)$ and the scheduling variable $p(t)$ follow an example in (Lopes dos Santos et al., 2011), and the output $y(t)$ is simulated with the affine LPV model (18). In order to estimate local LTI models, the range of the continuously varying $p(t)$ is divided into 5 intervals of equal size, as illustrated in Figure 2 with the 5 horizontal strips partitioning the vertical axis. The time axis is then partitioned accordingly, so that each time interval corresponds to values of $p(t)$ belonging to a single partition of the vertical axis. As $p(t)$ is not monotonic, the same partition of the vertical axis may correspond to several time intervals, coded with the same color in Figure 2. The data sequences $u(t), y(t), p(t)$ are then segmented so that the subset corresponding to each of the 5 colors in Figure 2 is used for the estimation of a local LTI model σ_i.

The subspace method for LTI state-space system identification (as implemented in the Matlab System Identification Toolbox) is used to estimate the 5 second order local models σ_i corresponding to the 5 colors in Figure 2. Following the local approach to LPV system identification, the interpolation of these local models σ_i then leads to an LPV model.

The data shown in Figure 1 were simulated with the affine LPV model (18), which can be used to evaluate the interpolated LPV model by comparing the behaviors of the two models. For this purpose, the *step responses* of the two LPV models at fixed scheduling values will be compared. For the interpolated LPV model, the interpolation at any given value of p yields an LTI model. On the other hand, fixing the scheduling varible of the affine LPV model (18) to the same value p leads also to an LTI model, which is referred to as a *frozen LPV model* in the LPV literature (Tóth, 2010).

The interpolation method presented in this paper is applied to the 5 local models σ_i in this example, with the weighting functions

$$\rho_i(p) = \frac{\exp\left(-\dfrac{(p - p_i)^2}{2s_i^2}\right)}{\displaystyle\sum_{j=1}^{5} \exp\left(-\dfrac{(p - p_i)^2}{2s_i^2}\right)} \qquad (20)$$

where $p_1, p_2, \ldots, p_5$ are the middles of the 5 partitions of the vertical axis in Figure 2, and s_i the half widths of these partitions.

For $p = -4$, the step responses of the interpolated LPV model and the frozen affine LPV model are compared in
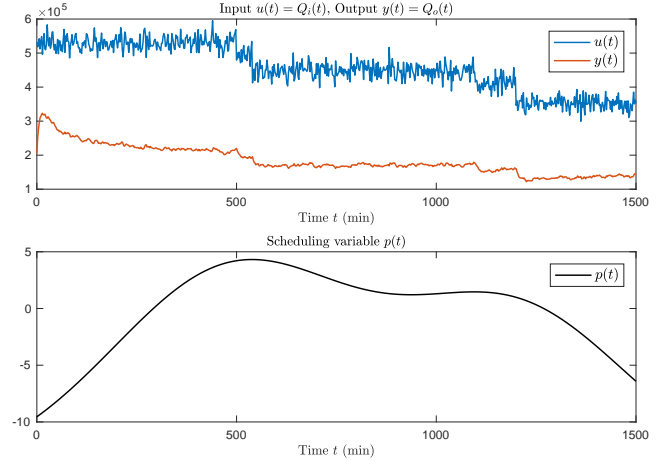

Fig. 1. Example 1: input, out and continuously varying scheduling variable $p(t)$.
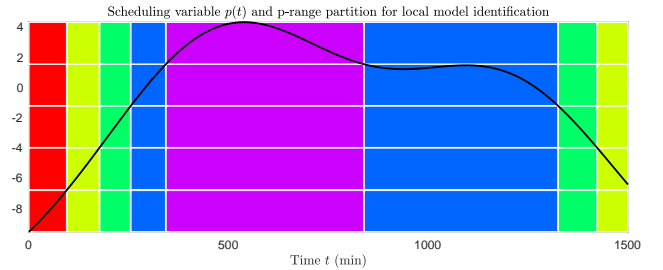

Fig. 2. Example 1: partition of the range of the scheduling variable (vertical axis) for local model identification, illustrated by the 5 horizontal strips. As $p(t)$ is not monotonic, the same partition of the vertical axis may correspond to several time intervals, coded with the same color.

Figure 3. Similarly another comparison for $p = 2$ is shown in Figure 4.

These results show that the static gains are relatively well interpolated, but the transient behaviors are significantly different between the interpolated LPV model and the frozen LPV model. Obviously, local model interpolation introduces approximation errors, but the observed differences may also have another origin: as $p(t)$ varies continuously in this example, in each of segmented time interval indicated by different colors in Figure 2, $p(t)$ is not a constant, hence each local model is estimated from data collected on a time varying system, which is then approximated by a local LTI model, introducing also approximation errors. The next example with piecewise constant $p(t)$ will confirm this fact.

Example 2 – stair-shaped scheduling variable

In the local approach to LPV system identification, typically data are collected at a working point corresponding to a fixed scheduling value, in order to estimate a local LTI model. Then the experience is repeated at another working point for another local model, and so on. Following such a procedure, the previously presented pipeline example is modified with a stair-shaped scheduling variable sequence, as shown in Figure 5 with the input-output data. Ac-
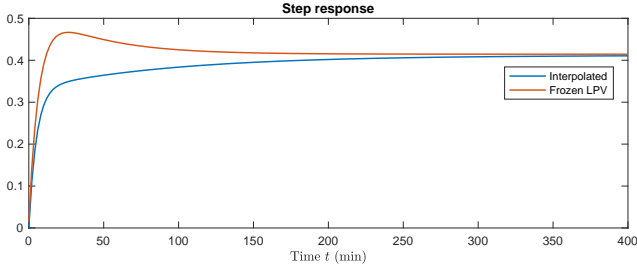
Fig. 3. Example 1: comparison between the step responses of the interpolated model and the frozen LPV model, both for $p = -4$.
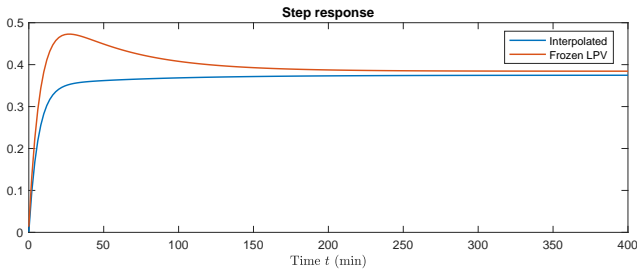

Fig. 4. Example 1: comparison between the step responses of the interpolated model and the frozen LPV model, both for $p = 2$.
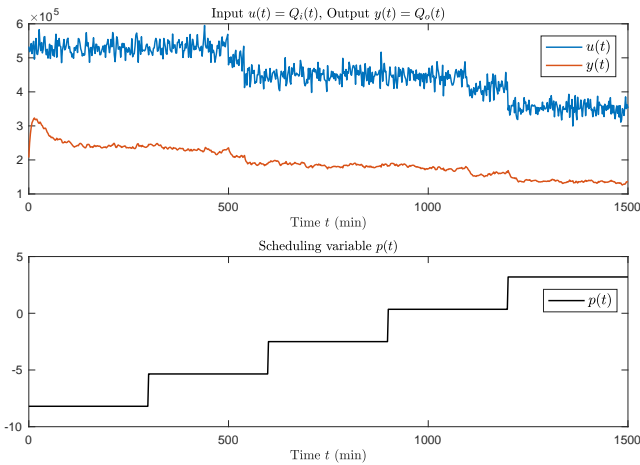

Fig. 5. Example 2: input, out and continuously varying scheduling variable $p(t)$.

cordingly, the range of the scheduling variable is then partitioned, as illustrated in Figure 6, where the time axis is also segmented, as indicated with the 5 different colors. Like in the previous example, local models are estimated from the data segments, and then interpolated. The step responses of the interpolated LPV model and of the affine LPV model (18) are then compared, for $p = -4$ in Figure 7, and for $p = 2$ in Figure 8. The two LPV models exhibit similar step responses, with smaller differences than in the previous example, notably in their transient behaviors. This improvement can be explained by the fact that now each of the local models is estimated from data collected when $p(t)$ is kept constant, hence there is no approximation of a time varying system by an LTI model.
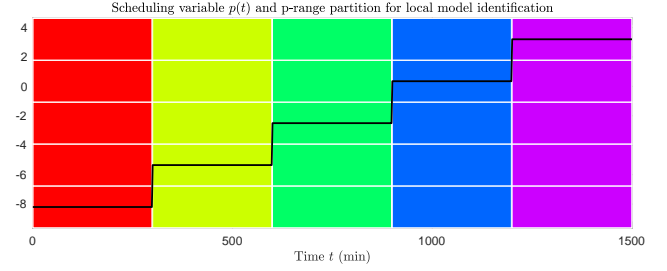

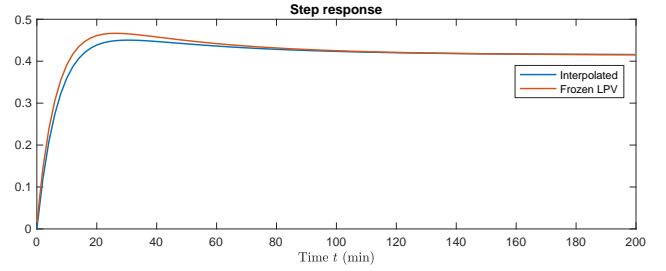Fig. 6. Example 2: the scheduling variable range partition for local model identification.

Fig. 7. Example 2: comparison between the step responses of the interpolated model and the frozen LPV model, both for $p = -4$.
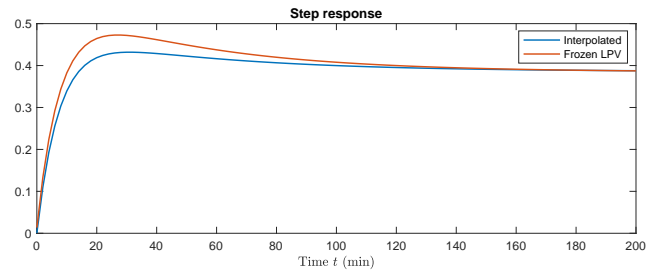

Fig. 8. Example 2: comparison between the step responses of the interpolated model and the frozen LPV model, both for $p = 2$.

6. CONCLUSION

Interpolation is essential in the local approach to LPV system identification. When local state-space models are used, as each local model can be estimated with an arbitrary state basis, existing interpolation methods require coherent local models. However, it is known that, if local state-space models are estimated without any global structural assumption, the local models themselves do not contain the information to make them coherent for the purpose of their interpolation. In order to avoid this delicate problem, a new method has been proposed in this paper for the interpolation of local state-space models, without requiring coherent local models. Such interpolated models can be used at fixed working points or during slow transitions between working points. It should be noted that, if an LPV model obtained by interpolating local models is to be used in situations with fast transitions between different working points, the interpolation of coherent state-space models remains necessary. This difficult issue is out of the scope of this paper.

REFERENCES

Bamieh, B. and Giarré, L. (2002). Identification of linear parameter varying models. *International Journal of Robust and Nonlinear Control*, 12, 841–853.

Bolea, Y., Blesa, J., and Puig, V. (2007). LPV modeling and identification of an open canal for control. In *European Control Conference*, 3584–3591. Kos, Greece.

De Caigny, J., Pintelon, R., Camino, J., and Swevers, J. (2014). Interpolated modeling of LPV systems. *IEEE Trans. on Control System Technology*, 22(6), 2232 – 2246.

De Caigny, J., Camino, Juan, F., and Swevers, J. (2009). Interpolating model identification for SISO linear parameter-varying systems. *Mechanical Systems and Signal Processing*, 23(8), 2395–2417.

De Caigny, J., Camino, J.F., and Swevers, J. (2011). Interpolation-based modeling of MIMO LPV systems. *IEEE Trans. on Control System Technology*, 19(1), 46–63.

Glover, K. (1984). All optimal hankel-norm approximations of linear multivariable systems and their l-infinity error bounds. *International Journal of Control*, 39(6), 1115–1193.

Kailath, T. (1980). *Linear Systems*. Prentice-Hall.

Ljung, L. (1999). *System Identification: Theory for the User*. PTR Prentice Hall, Upper Saddle River, NJ, USA, 2nd ed. edition.

Lopes dos Santos, P., Azevedo-Perdicoúlis, T.P., Ramos, J., Martins de Carvalho, J., Jank, G., and Milhinhos, J. (2011). An LPV modeling and identification approach to leakage detection in high pressure natural gas transportation networks. *IEEE Trans. on Control System Technology*, 19(1), 77–92.

Lopes dos Santos, P., Perdicoulis, T.P.A., Novara, C., Ramos, J.A., and Rivera, D.E. (eds.) (2012). *Linear Parameter-Varying System Identification*. World Scientific Publishing Company, New Jersey.

Lovera, M. and Mercere, G. (2007). Identification for gain-scheduling: a balanced subspace approach. In *American Control Conference*, 858–863. New York City.

Martínez-González, R., Bolea, Y., Grau, A., and Martínez-García, H. (2009). An LPV fractional model for canal control. *Mathematical Problems in Engineering*, 2009(1-18).

Mercere, G., Palsson, H., and Poinot, T. (2011). Continuous-time linear parameter-varying identification of a cross: A local approach. *IEEE Trans. on Control System Technology*, 19(1), 64–76.

Mohammadpour, J. and Scherer, C.W. (eds.) (2012). *Control of Linear Parameter Varying Systems with Applications*. Springer, New York.

Moore, B.C. (1981). Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. on Automatic Control*, 26(1), 17–32.

Nemani, M., Ravikanth, R., and Bamieh, B. (1995). Identification of linear parametrically varying systems. In *34th IEEE Control and Decision Conference*, 2990–2995. New Orleans.

Petersson, D. and Löfberg, J. (2009). Optimization based LPV-approximation of multi-model systems. In *European Control Conference*. Budapest, Hungary.

Piga, D., Cox, P.B., Toth, R., and Laurain, V. (2015). LPV system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53, 329–338.

Sename, O., Gaspar, P., and Bokor, J. (eds.) (2013). *Robust Control and Linear Parameter Varying Approaches*. Springer, Berlin.

Steinbuch, M., van de Molengraft, R., and van der Voort, A. (2003). Experimental modelling and LPV control of a motion system. In *American Control Conference*, volume 2, 1374–1379. Denver.

Tóth, R., Laurain, V., Gilson, M., and Garnier, H. (2012). Instrumental variable scheme for closed-loop LPV model identification. *Automatica*, 48(9), 2314–2320.

Tóth, R. (2010). *Modeling and Identification of Linear Parameter-Varying Systems*. Springer.

Van Wingerden, J.W. and Verhaegen, M. (2009). Subspace identification of bilinear and LPV systems for open- and closed-loop data. *Automatica*, 45(2), 372–381.

Varga, A. (1991). Balancing free square-root algorithm for computing singular perturbation approximations. In *30th IEEE Conference on Decision and Control*. Brighton, UK.

Zhang, Q. and Ljung, L. (2017). From structurally independent local LTI models to LPV model. *Automatica*, 84, 232–235.

Zhao, Y., Huang, B., Su, H., and Chu, J. (2012). Prediction error method for identification of LPV models. *Journal of Process Control*, 22(1), 180–193.

Zhu, Y. and Ji, G. (2009). LPV model identification using blended linear models with given weightings. In *The 15th IFAC Symposium on System Identification*. Saint Malo, France.

Zhu, Y. and Xu, X. (2008). A method of LPV model identification for control. In *The 17th IFAC World Congress*. Seoul, Korea.