Identification of multi-model LPV model with two scheduling variables using transition test

Jiangyin Huang*
Department of Electrical Engineering and Automation, Xiamen University of Technology, Xiamen 361024, Fujian, China
Email: jiangyinhuang@xmut.edu.cn
*Corresponding author

Jing Zhao
Department of Electrical Engineering and Automation, Xiamen University of Technology, Xiamen 361024, Fujian, China
Email: ztulipwork@139.com

Abstract: This paper presents the research findings of identification method for LPV models with two scheduling variables using transition tests. The LPV model is parameterised as blended linear models, which is also called a multi-model structure. The identification method proposed in this paper can be used in batch process identification. The usefulness of the method is verified by modelling a high purity distillation column. The outputs of the LPV models are compared and analysed with three kinds of weighting functions namely: linear, polynomial and Gaussian functions. The case study shows that the multi-model LPV models can yield a better model accuracy with respect to simulation outputs and step response fittings than linear models.

Keywords: LPV model; multi-model; transition test; high purity distillation column.


Biographical notes: Jiangyin Huang received her Doctor’s degree in Control Theory and Control Engineering from Xiamen University, Fujian, China in 2013. She is working as a Lecturer in the Department of Electrical Engineering and Automation at the Xiamen University of Technology, Fujian, China. Her current research focuses on model identification and model-based control for complex nonlinear systems. She has published over ten Scopus-indexed papers in international journal and conferences.

Jing Zhao received her BS and MS degrees from the Department of Electrical Engineering, Fuzhou University, Fuzhou, Fujian, China, in 1997 and 2000, respectively. She is currently pursuing her PhD degree at the Department of Electrical Engineering, Yuan Ze University, Taoyuan, Taiwan. Since 2009, she is with the School of Electrical Engineering & Automation, Xiamen University of Technology, Xiamen, China, as an Associate Professor. Her current research interests include intelligent control, adaptive control, fuzzy logic systems, and neural network applications.

This paper is a revised and expanded version of a paper entitled ‘Identification of LPV models with two scheduling variables using transition test’, presented at 2013 International Symposium on Vehicle, Mechanical and Electrical Engineering, HsinChu, Taiwan, 20–22 December 2013.

1 Introduction

It is well understood that nonlinear AR(MA)X models and neural-network models are often used in nonlinear system identification. It is to our knowledge that these models are complex in structure and often difficult to compute numerically. While block-oriented nonlinear models such as Hammerstein models and Wiener models (Meenakshipriya and Manikandan, 2015) are known to be simpler, they can only model nonlinear processes with static gains that are often too limited for process control applications. There are also feedback nonlinearities-based models like LuGre. They can express dynamic aspects with a static block, which make them preferable to Wiener and Hammerstein models. However, one downside is that feedback-based nonlinearities are not easy to identify. Recently, the linear parameter varying (LPV) model
identification has attracted great attention in both academia and industry (Laurain et al., 2010). The LPV model is capable of creating a nonlinear model for simulating and controlling parameter-varying process. The terminology of LPV was first introduced by Shamma and Athans (1991) in their gain-scheduling-control research. Articles on the applications of LPV (or gain scheduling) control, as well as the control of electro-mechanical systems have also been published by Rugh and Shamma (2000). Since then, several LPV model identification and control algorithms have been developed and published. Among other research are multiple-model approaches by Murray-Smith and Johansen (1997); subspace approach by Lopes et al. (2007) and Felici et al. (2007) and orthonormal basis functions by Tóth et al. (2009). Most references available in the literature about LPV models are based on parameter interpolation, including (Previdi and Lovera, 2003; Butcher and Karimi, 2008; Chuk et al., 2016). We have found that this kind of LPV models is often complicated. They have heavy dependencies on exogenous parameters. These dependencies cause them to have difficulty guaranteeing stability during the identification procedure. To overcome this problem, Zhu and Xu (2008) proposed a LPV model by interpolating local linear models. It was followed another form of LPV model proposed by Zhu and Ji (2009). They proposed to add weightings to the input sides. By doing so, it is easy to guarantee the stability of this kind of LPV models if the local linear models are all stable. Furthermore, these LPV models can be easily identified by least squares methods (Schimmack and Mercorelli, 2016). Some further researches, such as multi-model LPV model with two scheduling variables has been developed (see Huang et al., 2012).

There are two ways to perform identification tests for model-interpolation-based LPV models. One is an operating point test plus a transition test. The other is only a transition test. All of the LPV models mentioned in the publications (Zhu and Xu, 2008; Zhu and Ji, 2009; Huang et al., 2012) are identified using the first test method. Using such LPV models in process control has several advantages:

1. they can be identified easily
2. they can model both a static and a dynamic nonlinear process
3. they can incorporate their knowledge into the system by selecting the scheduling variables
4. they can be used in control conveniently.

The identification test which consists of both operating point test and transition test can only be used to identify the nonlinear processes with local steady states. It cannot be applied to identify the complex nonlinear systems with continuous change of operating characteristics, such as batch systems. As a supplement to the previous research, in this work, we study the identification algorithm of LPV model with two scheduling variables using the dataset collected only from a transition test. The identification methods of the LPV models with three kinds of weighting functions are proposed in this paper. For the LPV model with linear weights, because the parameters only exist in the local linear models, Gauss-Newton method and least squares method are combined to estimate all the parameters. For the LPV model with polynomial weights and Gaussian weights, the parameters exist in both the weighting functions and local linear models. Narendra-Gallman method is used to estimate the parameters. In the method, all the parameters are divided into linear and nonlinear parts according to the relationship between them and the optimisation objectives. Then these two parts are estimated using alternating iterative method.

In Section 2, identification algorithms of LPV models with two scheduling variables using a transition test are developed as well as their three kinds of weighting functions, namely: linear, polynomial and Gaussian. In Section 3, LPV models of a rigorous model of a high purity distillation column are proposed. The validation of effectiveness of the LPV models is also given. Section 4 is the conclusions.

## 2 Identification of two scheduling variables LPV model using transition test

### 2.1 Two scheduling variables LPV model parameterisations

For industrial processes with complicated nonlinearity, LPV models with one scheduling variable may not provide enough accuracy. In the following, an identification method for LPV models with two scheduling variables is described for a multi-input single-output (MISO) process. For a multi-input multi-output (MIMO) process, the procedure can be repeated for each output.

LPV-IO model and LPV-SS model are two common structures in LPV identification. Both of them can be used in industrial application. In this paper, we only study the identification method for LPV-IO model. Given a MISO system with $m$ inputs; $y(t_k)$ and $u_i(t_k), (i = 1, 2, ..., m)$ are the process output and inputs at discrete time $t_k$. One type of LPV-IO model can be written as:

$$y(t_k) = \frac{1}{A(q, w_1, w_2)} \left[ B_1(q, w_1, w_2) u_1(t_k) + \cdots + B_m(q, w_1, w_2) u_m(t_k) \right] + \nu(t_k) \quad (1)$$

where

$$B_i(q, w_1, w_2) = \left[ h^{i}_{q^{-1}} w_1, w_2 q^{d_i} \right] q^{-d_i}$$

$$A(q, w_1, w_2) = 1 + a_1(q, w_1, w_2) q^{-1} + \cdots + a_m(q, w_1, w_2) q^{-m} \quad (2)$$

are polynomials of $q^{-1}, d_i$ is the delay from the $i^{th}$ input to the output. $q^{-1}$ denotes the unit delay operator and $\nu(t_k)$ is a
stationary stochastic process with zero mean and bounded variance. \( n \) is the order of the model. The variables \( w_1(t_k) \) and \( w_2(t_k) \) are the scheduling variables, which determines the operation points of the process operation. They are measured variables from the process or can be calculated from measurable process variables. Assume \( w_1(t_k) \) and \( w_2(t_k) \) are in the ranges of

\[
w_1(t_k) \in [w_{1l}, w_{1h}], \quad w_2(t_k) \in [w_{2l}, w_{2h}]
\]

where \( w_{1l}, w_{1h} \) are the low and high limits of \( w_1(t_k) \), \( w_{2l}, w_{2h} \) are the low and high limits of \( w_2(t_k) \).

This is the most commonly used LPV-IO model in literature (see e.g., Laurain et al., 2010). It should be noted that in order to simplify the model, we use a common denominator \( A(q, w_1, w_2) \) for all the sub models.

As it is commonly used in the literature, the parameters \( a_i(w_1, w_2) \) and \( b_j(w_1, w_2) \) can be estimated using polynomial functions of \( w_1(k) \) and \( w_2(k) \) as follows:

\[
b_j(w_1, w_2) = \beta_{i,j}^1 + \beta_{i,j}^2 w_1(t_k) + \beta_{i,j}^3 w_2(t_k)
\]

\[
+ \beta_{i,j}^4 w_1^2(t_k) + \beta_{i,j}^5 w_1 w_2(t_k)
\]

\[
+ \beta_{i,j}^6 w_2(t_k) + \beta_{i,j}^7 w_2^2(t_k)
\]

\[
+ \beta_{i,j}^8 w_1(t_k)^3 + \beta_{i,j}^9 w_1^2 w_2(t_k)
\]

\[
+ \beta_{i,j}^{10} w_1 w_2^2(t_k)
\]

\[
+ \beta_{i,j}^{11} w_2(t_k)^3
\]

(3)

\[
a_j(w_1, w_2) = \alpha_{i,j}^0 + \alpha_{i,j}^1 w_1(t_k) + \alpha_{i,j}^2 w_2(t_k)
\]

\[
+ \alpha_{i,j}^3 w_1(t_k)^2 + \alpha_{i,j}^4 w_1^2 w_2(t_k)
\]

\[
+ \alpha_{i,j}^5 w_2(t_k)^2 + \alpha_{i,j}^6 w_2^2(t_k)
\]

\[
+ \alpha_{i,j}^7 w_1(t_k) w_2(t_k) + \alpha_{i,j}^8 w_1(t_k)^2 w_2(t_k)
\]

\[
+ \alpha_{i,j}^9 w_1^2 w_2^2(t_k)
\]

(4)

Of course, higher order polynomials can also be used.

The model in equations (1) to (4) is called as parameter-interpolation LPV-IO model by Zhao et al. (2012). We often found the result was overly complicated and its intensive nonlinear dependencies have made it difficult to guarantee stability during identification. For example, based on a quadratic stability criterion (Gilbert et al., 2010), to overcome this problem, Zhu and Xu (2008) proposed a simpler LPV model structure based on multi-model interpolation. Assume there are linear models at \( n_1 \times n_2 \) operation points:

\[
w_1(t_k); \quad w_1^1 < w_1^2 < ... < w_1^{n_1}; \quad w_2(t_k); \quad w_2^1 w_2^2 ... < w_2^{n_2}
\]

Denote \( \hat{y}^i (t_k) \) as the output of the linear model at \( w_1(t_k) = w_1^i, w_2(t_k) = w_2^j \); \( G_i^j(q) \), \( (i = 1, 2, ..., m) \) as the transfer function from \( v_i(t_k) \) to \( \hat{y}^i (t_k) \). To simplify the operation, we assume that all the local models have the same order \( r \). It should be noted that if the orders are different, the parameters can be estimated by an optimisation algorithm described later. The linear models at each of the operation points can be written as:

\[
\hat{y}^i (t_k) = G_i^j(q)u_i(t_k) + ... + G_m^j(q)u_m(t_k)
\]

(5)

\[
= \frac{1}{A^j(q)} [B_i^j(q)u_i(t_k) + ... + B_m^j(q)u_m(t_k)]
\]

(5)

\[
(i = 1, 2, ..., n_i; \quad j = 1, 2, ..., n_2)
\]

where

\[
A^j(q) = 1 + a_{1,j}^1 q^{-1} + ... + a_{r,j}^1 q^{-r};
\]

\[
B_i^j(q) = b_{1,j}^1 q^{-1} + ... + b_{r,j}^1 q^{-r}; \quad (l = 1, ..., m)
\]

The model parameter vector to be estimated is given as:

\[
\theta_{Model} = [a_{1,1}^1, ..., a_{n_1,1}^{n_2}, b_{1,1}^1, ..., b_{n_1,1}^{n_2, m}]
\]

(7)

Then the multi-model LPV model can be obtained by interpolating the linear models in equation (5):

\[
\hat{y}(t) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \alpha_i^j (w_1, w_2) \hat{y}^i (t_k)
\]

(8)

where the model weighting function \( \alpha_i^j(w_1, w_2) \) is a scalar function of \( w_1(t_k) \) and \( w_2(t_k) \). The structures of the weighting functions will be specified later. It is easy to prove that the LPV model (8) is stable if all the linear models are stable (Huang et al., 2012).

There are two ways to perform an identification test for multi-model LPV models. One is operating point tests plus a transition test, the other is only a transition test (see Figure 1). The identification method for multi-model LPV model depends on the test method used. When the first test method is used, first the linear models at each operating points are identified using the data from each operating point tests. Next the weighting functions are estimated using the total data including operating point tests and transient tests. This method was used in the publications (Zhu and Xu, 2008; Huang et al., 2012). When the second test method is used, some nonlinear optimisation algorithms are needed to estimate the parameters, which are often numerically complicated. For our research, the algorithms used to construct such LPV model using a transition test are studied.
2.2 Identification of parameter-interpolation LPV models

The model in equations (1) to (4) can be written as a linear regression equation:

\[ y(t_k) = \phi(t_k) \theta + e(t_k) \]  

where

\[ e(t_k) = A(q, w_1, w_2)v(t_k) \]
\[ \phi(t_k) = \left[ \phi_2(t_k) \quad \phi_3(t_k) \right] \]
\[ \phi_2(t_k) = \left[ -y(t_k - 1) \quad -w_2(t_k) \quad y(t_k - 1) \right] \]
\[ \phi_3(t_k) = \left[ u_1(t_k - 1) \quad w_1(t_k - 1) \quad u_1(t_k - 1) \right] \]
\[ \theta = \left[ \alpha_1 \alpha_2 \beta_1 \beta_2 \beta_3 \right]^T \]

Collect the input-output data for \( k = 1, \ldots, N \) and fill the data in equation (9). Then, the parameters in \( \theta \) can be estimated by minimising the equation error criterion:

\[ V_{EE}^N = \frac{1}{N-n} \sum_{k=n+1}^{N} \varepsilon_{EE}(t_k)^2 \]  

where subscripts \( EE \) denotes equation error:

\[ \varepsilon_{EE}(t_k) = y(t_k) - \phi(t_k) \theta \]

In order to solve the optimisation problem, the least-squares method is used and the solution is

\[ \hat{\theta} = (\Phi^T \Phi)^{-1} \Phi^T Y \]  

where

\[ Y = \begin{bmatrix} y(t_k + 1) \\ y(t_k + 2) \\ \vdots \\ y(N) \end{bmatrix}, \Phi = \begin{bmatrix} \phi(t_k + 1) \\ \phi(t_k + 2) \\ \vdots \\ \phi(N) \end{bmatrix} \]

The advantage of the least-squares method is its numerical simplicity. It has an analytical solution and does not have local minimum problem. However, it is well-known that the least-squares method is often biased when there are unmeasured disturbances at the output.

In nonlinear model identification, it is more important to assure good approximation property, that is, to minimise the bias error, than to assure noise reduction property. Zhu (2001) has argued that if the bias is the main concern, an output error criterion is a good criterion for model identification. In the output error method, the parameters of the model in equations (1) to (4) are obtained by minimising the output error criterion as:

\[ V_{OE}^N = \frac{1}{N-n} \sum_{k=n+1}^{N} \varepsilon_{OE}(t_k)^2 \]  

where the subscripts \( OE \) denotes output error:

\[ \varepsilon_{OE}(t_k) = y(t_k) - \frac{1}{A(q, w_1, w_2)} \left[ B_1(q, w_1, w_2)u_1(t_k) + \ldots + B_m(q, w_1, w_2)u_m(t_k) \right] \]

Because the errors are nonlinear in the model parameters, the output error method does not have an analytical solution. Therefore, a numerical optimisation method is needed to search for a solution. However, numerical problems such as local minima and divergence may occur in numerical optimisation. Such problems indeed occurred in the case study in Section 3. We will not give details of these methods as we are mainly studying multi-model LPV models.

2.3 Identification of multi-model LPV models

In order to construct multi-model LPV models, the original linear models at each of the operation points should be identified first. As shown in Figure 2, the corresponding data slices

\[ w_1(t_k) \in [w_1 - \delta_1, w_1 + \delta_1], \quad w_2(t_k) \in [w_2 - \delta_1, w_2 + \delta_1] \]

where

\[ 0 < \delta_i \leq \min \{ w_{i+1} - w_i, w_i - w_{i-1} \}; \quad i = 2, \ldots, m - 1 \]
\[ 0 < \delta_i \leq \min \{ w_{i+1} - w_i, w_i - w_{i-1} \}; \quad i = 1 \]
\[ 0 < \delta_i \leq \min \{ w_{i+1} - w_i, w_i - w_{i-1} \}; \quad i = m. \]
and
\[ 0 < \delta_2 \leq \min \left( w_{j+1}^2 - w_j^2, w_j^2 - w_{j-1}^2 \right); \quad j = 2, ..., n_2 - 1 \]
\[ 0 < \delta_2 \leq \min \left( w_{j+1}^2 - w_j^2, w_j^2 - w_{j-1}^2 \right); \quad j = 1 \]
\[ 0 < \delta_2 \leq \min \left( w_{j+1}^2 - w_j^2, w_j^2 - w_{j-1}^2 \right); \quad j = n_1 \]
are collected to initialise the linear model \( \hat{y}^{i,j}(k) \) by common linear identification method.

### 2.3.1 Identification of multi-model LPV model with linear weights

In the multi-model LPV models, both of the parameters in the local linear models and in the weighting functions need to be estimated. As mentioned before, some algorithms should be used to estimate these parameters. The weighting function can be constructed in different ways. The easiest way is to use linear weights that can be pre-assigned without estimation. Other methods include cubic splines, polynomial, Gaussian functions and so on, but these methods require the estimation of some extra hyper parameters.

Assume \( \hat{y}(t_k|w_1(t_k), w_2(t_k)) \) is the output of LPV model which to be estimated where:
\[
\begin{align*}
    w_k^1 &< w_1(t_k) < w_{k+1}^1, \ (k = 1, 2, ..., n_1 - 1) \\
    w_k^2 &< w_2(t_k) < w_{k+1}^2, \ (l = 1, 2, ..., n_2 - 1)
\end{align*}
\]

The linear model weights are given in equation (13) (Huang et al., 2012):
\[
\alpha^{i,j}(w_1, w_2) = \begin{cases} 
    \frac{w_{k+1}^2 - w_k^1}{w_{k+1}^2 - w_k^1} \cdot \frac{w_{k+1}^1 - w_k^2}{w_{k+1}^2 - w_k^1}, & \text{if } i = k, j = l; \\
    \frac{w_{k+1}^2 - w_k^1}{w_{k+1}^2 - w_k^1} \cdot \frac{w_{k+1}^1 - w_k^2}{w_{k+1}^2 - w_k^1}, & \text{if } i = k, j = l+1; \\
    \frac{w_{k+1}^2 - w_k^1}{w_{k+1}^2 - w_k^1} \cdot \frac{w_{k+1}^1 - w_k^2}{w_{k+1}^2 - w_k^1}, & \text{if } i = k+1, j = l; \\
    \frac{w_{k+1}^2 - w_k^1}{w_{k+1}^2 - w_k^1} \cdot \frac{w_{k+1}^1 - w_k^2}{w_{k+1}^2 - w_k^1}, & \text{if } i = k+1, j = l+1; \\
    0, & \text{others}
\end{cases}
\]

It can be seen that there are no parameters in the linear weighting function. Therefore, the work left is to optimise the local linear models. Denote the process dataset as:
\[
Z^N = \{ u_1(t_k), ..., u_m(t_k), y(t_k), w_1(t_k), w_2(t_k) \}, \quad t_k = 1, 2, ..., N
\]

The parameters in the linear models can be estimated by minimising the output error loss function:
\[
\begin{align*}
    V_{OE}^{\text{N}} &= \frac{1}{N-n} \sum_{i=n+1}^{N} e_{OE}(t_k)^2 \\
    &= \frac{1}{N-n} \sum_{i=n+1}^{N} [y(t_k) - \hat{y}(t_k)]^2
\end{align*}
\]

where
\[
\begin{align*}
    y(t_k) - \hat{y}(t_k) &= y(t_k) - \sum_{i=1}^{n} \sum_{j=1}^{n_2} \alpha^{i,j}(w_1, w_2) * \hat{y}^{i,j}(t_k) \\
    &= y(t_k) - \sum_{i=1}^{n} \sum_{j=1}^{n_2} \alpha^{i,j}(w_1, w_2) * \frac{1}{A^{i,j}(q)} \left[ B_{1,j}^{i} (q) u_1(t_k) + ... + B_{n,j}^{i} (q) u_n(t_k) \right] \\
    &= y(t_k) - \sum_{i=1}^{n} \sum_{j=1}^{n_2} \alpha^{i,j}(w_1, w_2) \left[ \left( B_{1,j}^{i} q^{-1} + ... + B_{n,j}^{i} q^{-r} \right) u_1(t_k) \right] \\
    &= 1 + a_{1,j}^{i} q^{-1} + ... + a_{n,j}^{i} q^{-r}
\end{align*}
\]

Because of the nonlinear relationships between \( \alpha_{l,j} \) (\( l = 1, 2, ..., r \)) and \( V_{OE}^{\text{N}} \), some numerical algorithm should be used to solve the optimisation problem in equation (15). In this paper, we use the Gauss-Newton method (Björck, 1996) which is often used to minimise a sum of squared function values.

Denote \( \hat{\theta}_{\text{Model}}^k \) as the estimated model parameter vector of the \( k \)th iteration. Assume \( \hat{\theta}_{\text{Model}}^k \) is close to the local minimum. Using Taylor’s theorem, the output error \( e_{\text{Model}}(t_k, \theta_{\text{Model}}) \) can be written as:
\[
\begin{align*}
    e_{\text{Model}}(t_k, \theta_{\text{Model}}) &\approx e(t_k, \hat{\theta}_{\text{Model}}^k) \\
    &= e(t_k, \hat{\theta}_{\text{Model}}^k) + \frac{\partial e}{\partial \theta_{\text{Model}}} (\theta_{\text{Model}} - \hat{\theta}_{\text{Model}}^k) \\
    &= e(t_k, \hat{\theta}_{\text{Model}}^k + \hat{\theta}_{\text{Model}}^k) + \phi(t_k)(\theta_{\text{Model}} - \hat{\theta}_{\text{Model}}^k) \\
    &= e(t_k, \hat{\theta}_{\text{Model}}^k) + \phi(t_k)(\theta_{\text{Model}} - \hat{\theta}_{\text{Model}}^k) + \phi(t_k)\hat{\theta}_{\text{Model}}^k
\end{align*}
\]
\[ \phi^k(t_k) = \frac{\partial \epsilon(t_k, \theta_{\text{Model}})}{\partial \theta_{\text{Model}}} \bigg|_{\theta_{\text{Model}} = \hat{\theta}_{\text{Model}}} \]

\[
\begin{bmatrix}
\alpha^{1,1}(w_1, w_2) \cdot B_1^{1,1}(q)u_l(t_k - 1) + \ldots + B_m^{1,1}(q)u_m(t_k - 1) \\
\vdots \\
\alpha^{n,q_2}(w_1, w_2) \cdot B_1^{n,q_2}(q)u_l(t_k - r) + \ldots + B_m^{n,q_2}(q)u_m(t_k - r) \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
\alpha^{1,1}(w_1, w_2) \cdot B_1^{1,1}(q)u_l(t_k - 1) \\
\vdots \\
\alpha^{n,q_2}(w_1, w_2) \cdot B_1^{n,q_2}(q)u_l(t_k - r) \\
\end{bmatrix}
\]

After approximate treatment, the relationships between the parameters and output error become linear. Therefore, the optimisation problem in equation (15) can be solved by least-square method and the solution is:

\[
\hat{\theta}_{\text{Model}}^{k+1} = \left[ \sum_{q_{i,j} = 1}^{N} \left[ \phi^k(t_k) \right]^T \phi^k(t_k) \right]^{-1} \sum_{q_{i,j} = 1}^{N} \left[ \phi^k(t_k) \right]^T \left[ \phi^k(t_k) \hat{\theta}_{\text{Model}} - \epsilon(t_k, \hat{\theta}_{\text{Model}}) \right] 
\]

\[
\begin{align*}
\theta_{\text{Weight}} &= \left[ \beta_1^{1,1}, \beta_2^{1,1}, \ldots, \beta_1^{n,q_2}, \beta_2^{n,q_2} \right]^T \\
\end{align*}
\]

The weighting parameter vector is:

\[
\epsilon_{\text{el}}(t_k) = y(t_k) - \hat{y}(t_k) \\
= y(t_k) - \sum_{i=1}^{m} \sum_{j=1}^{\infty} \alpha^{i,j}(w_1, w_2) \cdot \hat{y}^{i,j}(t_k) \\
= y(t_k) - \sum_{i=1}^{m} \sum_{j=1}^{\infty} \left[ \beta_1^{i,j}w_1^2(t_k) + \ldots + \beta_1^{i,j}w_1(t_k) \right] \\
\]

The parameters can be estimated by minimising the same output error loss function as equation (15). It can be seen from equation (22) that the parameters exist in both the weight function and linear models. Therefore, the optimisation problem is more complex than using linear weights. For our research purpose, Narendra-Gallman method (Narendra and Gallman, 1966) is used to solve this problem. In the method, the optimisation problem is divided into two parts: linear part and nonlinear part. These two parts are estimated separately in each iterative procedure. The optimisation consists of four steps:

Step 1 Local linear models at each of the operation points are initialised using the corresponding test data slices. Denote error threshold \( \epsilon_{\text{el}} > 0 \) and iteration count \( k = 1 \).

Step 2 The local linear models \( \hat{y}^{i,j}(l_k) \) are fixed and the weighting parameters are estimated using least-squares method. The scheduling variable is expressed as:
\[
\varphi(t_k) = \begin{bmatrix} 1 & w_1(t_k) & w_2(t_k) & w_1(t_k)^2 \\
\end{bmatrix}
\]

(23)

The LPV model using cubic polynomial weights can be written as:
\[
\hat{y}^{(k)}(t_k) = \varphi(t_k) \hat{\theta}^{(1)}(t_k) \quad \cdots \quad \varphi(t_k) \hat{\theta}^{(2)}(t_k) \theta_{\text{Weight}}
\]

(24)

Then \( \theta_{\text{Weight}} \) can be estimated by minimising the output error loss function in equation (15). Since the linear models are fixed, the optimisation is a linear least-squares problem and has the following solution:
\[
\hat{\theta}^{(k)}_{\text{Weight}} = (\Phi^T \Phi)^{-1} \Phi^T y
\]

(25)

where
\[
Y = \begin{bmatrix} y(1) \\ y(2) \\ \vdots \\ y(N) \\
\end{bmatrix}, \quad \Phi = \begin{bmatrix} \varphi(1) \hat{\theta}^{(1)}(1) & \varphi(1) \hat{\theta}^{(2)}(1) \\ \varphi(2) \hat{\theta}^{(1)}(2) & \varphi(2) \hat{\theta}^{(2)}(2) \\ \vdots \\ \varphi(N) \hat{\theta}^{(1)}(N) & \varphi(N) \hat{\theta}^{(2)}(N) \\
\end{bmatrix}
\]

(26)

The data needs to satisfy certain conditions in order to guarantee that the matrix \( \Phi \) has a full column rank and equation (25) has a unique solution (Zhu and Xu, 2008). These conditions are:

A1 During the test, the number of distinct values of \( w_1(t_k) \) should be larger than \( n_1 \); the number of distinct values of \( w_2(t_k) \) should be larger than \( n_2 \).

A2 All the identified models are stable and dynamically different, meaning that the time constants (or poles and zeros) are different.

Step 3 The model weights \( \hat{\theta}^{(k)}_{\text{Weight}} \) are fixed and the linear model parameters are estimated by minimising the output error again. The Gauss-Newton method mentioned in 2.2.1 is used to obtain \( \hat{\theta}^{(k)}_{\text{Model}} \). During the iteration process, all unstable intermediate models should be replaced by approximate stable ones.

Step 4 The best fit (BFT) percentage (Ljung, 2006) of the intermediate LPV model is calculated. If \( \text{BFT} < \varepsilon_k \), iteration is stopped. Otherwise let \( k = k + 1 \), then go to Step 2.

\[
\text{BFT} = 100\% \times \max \left( 1 - \frac{\| y(t) - \hat{y}^{(k)}(t_k) \|^2}{\| y(t) - \bar{y} \|^2}, 0 \right)
\]

(27)

where \( \bar{y} \) is the mean of \( y(t) \).

2.3.3 Identification of multi-model LPV model with Gaussian weights

Though using polynomial function can yield smaller output error than using linear weights, at times the step response accuracy is poor. This is probably caused by the unconstrained shape (form) of the cubic splines functions. According to the LPV structure in equation (8), it is required that \( \alpha_{ij}(w_1, w_2) \in [0, 1] \) and \( \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \alpha^{ij}(w_1, w_2) = 1 \). However, the polynomial weights may exceed the expected range because of the excitation conditions or poor linear model quality. Therefore, the polynomial weights should be checked carefully before performing LPV model interpolation. Here, the Gaussian weighting function is adopted, which has a constrained shape and does not have the same problem the polynomial function has.

When using the Gaussian function as the model weights, the weightings can be written as:
\[
\alpha^{ij}(w_1, w_2) = \frac{\tilde{\alpha}^{ij}(w_1, w_2)}{\sum_{k=1}^{n_1} \sum_{j=1}^{n_2} \tilde{\alpha}^{ij}(w_1, w_2)},
\]

\[(i = 1, 2, ..., n_1; j = 1, 2, ..., n_2) \]

(28)

where
\[
\tilde{\alpha}^{ij}(w_1, w_2) = \exp \left\{ -\frac{1}{2} \left( \frac{w_1(t_k) - w_1^i}{\sigma_{i,j}^1} \right)^2 + \left( \frac{w_2(t_k) - w_2^j}{\sigma_{i,j}^2} \right)^2 \right\}
\]

(29)

\( \sigma_{i,j}^1 \) and \( \sigma_{i,j}^2 \) are the width coefficients.

It is worth noting that due to the symmetry of Gaussian functions, the operation points need to be uniformly spaced. The Narendra-Gallman method mentioned in 2.2.2 can also be used here to estimate the parameters in the weighting functions and in the local linear models.

3 Identification of a high purity distillation column

Nonlinear system identification is much more complex than linear system identification. When a nonlinear system identification method is developed, it is important to verify its ability to approximating a certain class of life systems. In this section, accuracy of the LPV model is studied by
identifying a high purity distillation column. The test data is generated by simulating a rigorous physical model of the distillation column.

Distillation is one of the most important unit operations in the chemical industry. Among various distillation operations, high purity distillation poses difficult identification and control problems (Pan et al., 2016). In this section, a high purity distillation column model is used as a test process in order to demonstrate the usefulness of the LPV model identification methods. The distillation column under investigation operated in the LV-configuration (Skogestad, 1997). The inputs or manipulated variables (MVs) are the reflux (LT) and boil-up rate (VB); the outputs or controlled variables (CVs) are the top product composition (yD) and bottom product composition (xB). Besides the two MVs, there is also a disturbance variable (DV): the feed rate (F). See Figure 3 for a schematic representation of the distillation column. When the column is operated over a relatively wide operation region, it reveals a significant nonlinear behaviour. Publications for the research of high purity distillation column modelling include NARX model (Sriniwas et al., 1995), Wiener model (Bloemen et al., 2001) and LPV model with two scheduling variables using working point tests and transition tests (Huang et al., 2012). The main difference between this paper and (Huang, et al., 2012) is the identification test method, which leads to different optimisation algorithms.

### 3.1 Transition test with feed and top purity as scheduling variables

In a high purity distillation column, the feed rate (F) often causes huge change in production condition. Furthermore, the top product purity (yD) is the most important quality index of the process. Therefore, yD and F are selected as the two scheduling variables. Assume that yD and F change continuously along the operating trajectory as shown in Figure 4. An identification test only includes transition period is performed during the whole process. The test lasts for 50,000 minutes. The sampling time is 1 minute.

Test signals which are added to LT and VB are generalised binary noise (GBN) signals. Since a GBN signal has a continuous spectrum over \([0, \pi]\), it is a persistent excitation of any finite order. This can ensure the identifiability of the local linear models. The switching time of the test signals is 50 minutes. The amplitudes of the GBN signals are very small in order to prevent the excitation of nonlinearity.

In order to test the influence of unmeasured disturbances and measurement noises, a noisy dataset is also generated and used. The noisy dataset is generated in two steps:

1. A filtered white noise sequence is added at F, in order to represent unmeasured process disturbance. The variance of the disturbance is 50% of the GBN signal at F. The filter is
   \[
   \frac{1}{1 - 0.8q^{-1}}
   \]

2. 1% (in variance) white noises are added at the outputs xB and yD, in order to represent the measurement noises.

Figure 5 shows the identification test dataset.

The operating points are selected based on trial-and-errors. For a given operating range, the first two or three operating points are used at each scheduling variable. If the obtained LPV model is not accurate enough with respect to simulation errors and step responses, the number of operating points should be increased.

In this research, a total of \(5 \times 3 = 15\) operation points are selected at:

- F: \(w_1^f = 0.8, w_2^f = 0.9, w_3^f = 1.1, w_4^f = 1.2\)
- yD: \(w_1^D = 0.95, w_2^D = 0.97, w_3^D = 0.99\)

Then the test data is divided into data slices. The data slices are close to the corresponding operation points and are used to identify original local linear models. All linear models are identified by the asymptotic method (Zhu, 1998). The method can provide not only input/output models and
Identification of multi-model LPV model with two scheduling variables using transition test

unmeasured disturbance models that are asymptotic maximum likelihood estimates. It also can provide the upper bound matrix for the model errors. These can be used for the analysis of model validation and robustness. In this work, all the linear models are two-order.

Figure 5 Identification test dataset (see online version for colours)

3.2 Parameter-interpolation LPV models

First, the parameter-interpolation LPV model is identified using the least-squares method in equation (11). The model order (the degrees of the polynomials) for yD is 5; the model order for xB is 3. This is because among the identified linear models at all 15 operating points, 13 of them have order 3 for xB and 10 of them have order 5 for yD. The quality of the LPV model is assessed on the validation data which is different from the estimation data. The predicted outputs and the measured outputs are shown in Figure 6. The used prediction formula is based on (9) and the predicted output is a function of previous output and previous inputs:

\[
\hat{y}(t_k) = \phi(t_k) \hat{\theta} = \left[1 - \hat{A}(q, w_1, w_2)\right] y(t_k) + B_1(q, w_1, w_2) u_1(t_k) + \ldots + B_m(q, w_1, w_2) u_m(t_k)
\]

(30)

It can be seen from Figure 6 that the predicted outputs follow the measured output almost perfectly. However, it only means that the model makes a good one-step-ahead prediction for the distillation column. It does not mean that the model is suitable for control. It is well-known that a good input-output model or simulation model is important for control (Zhu, 2001). The model simulation, which is based on equation (1), is only a function of inputs:

\[
\hat{y}(t) = \frac{1}{\hat{A}(q, w_1, w_2)} \left[ B_1(q, w_1, w_2) u_1(t) + \ldots + B_m(q, w_1, w_2) u_m(t) \right]
\]

(31)

The simulated outputs and the measured outputs are shown in Figure 7. The step responses of LPV model for yD at F = 0.817, yD = 0.989 are shown in Figure 8. It is obvious that the identified LPV model at this working point is unstable. Table 1 compares the prediction BFTs and output BFTs of the least-squares LPV model. According to the bad result of the model for yD, the model can be regarded as useless for control. This result shows clearly that, for the given process and LPV model structure, the use of equation error criterion may not produce a good LPV model for control.

Figure 6 Outputs of real process and the least-squares LPV model prediction (see online version for colours)

Figure 7 Outputs of the process and the least-squares LPV model simulation (see online version for colours)

Table 1 Prediction BFTs and output BFTs of the least-squares LPV model

<table>
<thead>
<tr>
<th></th>
<th>yD</th>
<th>xB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction BFT</td>
<td>98.9%</td>
<td>99.3%</td>
</tr>
<tr>
<td>Output BFT</td>
<td>0.0%</td>
<td>93.2%</td>
</tr>
</tbody>
</table>

Next, an output error method is used to identify the parameter-interpolation LPV model in equations (1) to (4),
where the Gauss-Newton method is used in numeric optimisation. Unfortunately, the optimisation failed to converge due to the instability of the intermediate models and no useful result can be obtained.

Figure 8  Step responses of the models for \( y_D \) at \( F = 0.817 \), \( y_D = 0.989 \) (see online version for colours)

3.3 Multi-model LPV models with three kinds of weighting functions

The multi-model LPV models with three kinds of weighting functions, namely, linear, polynomial and Gaussian function are identified. In order to assess the usefulness of optimising the local linear models, multi-model LPV models without linear model optimisation are also identified. The simulated LPV model outputs are compared to the measured outputs in Figures 9–11.

Figure 9  Outputs of the process and of the LPV models with linear weights (see online version for colours)

It can be seen from the Figures 9–11 and Table 2 that the linear model cannot represent the process dynamics well. The LPV models yield much better results with respect to the simulation result. The LPV models with linear model optimisation yield especially better fitting result than those without linear model optimisation.

Table 2  BFTs of linear model and LPV models with different weighting functions

<table>
<thead>
<tr>
<th>Models</th>
<th>( y_D )</th>
<th>( x_B )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear model</td>
<td>0.0%</td>
<td>34.5%</td>
</tr>
<tr>
<td>LPV models without linear model</td>
<td>65.5%</td>
<td>76.8%</td>
</tr>
<tr>
<td>Linear weight</td>
<td>93.9%</td>
<td>98.3%</td>
</tr>
<tr>
<td>Polynomial weight</td>
<td>70.7%</td>
<td>86.3%</td>
</tr>
<tr>
<td>Gaussian weight</td>
<td>74.2%</td>
<td>83.4%</td>
</tr>
<tr>
<td>Gaussian weight</td>
<td>92.8%</td>
<td>98.1%</td>
</tr>
<tr>
<td>LPV models with linear model</td>
<td>77.8%</td>
<td>91.9%</td>
</tr>
</tbody>
</table>

Huang et al. (2012) have pointed out that good fit of validation data may not be sufficient to assure good model quality. A good way to verify model quality at an operating point is to check its step responses. The step responses of the real process and the LPV models are compared at...
eight points, namely, \([F, y_D] = [0.85, 0.96], [0.85, 0.98], [0.95, 0.96], [0.95, 0.98], [1.05, 0.96], [1.05, 0.98], [1.15, 0.96]\) and \([1.15, 0.98]\). Step responses of the LPV models and the real process at four operating points are shown in Figures 12–15.

**Figure 12** Step responses at \(F = 0.85, y_D = 0.96\) (see online version for colours)

<table>
<thead>
<tr>
<th>LT</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Step response at F = 0.85, yD = 0.96" /></td>
<td><img src="image2" alt="Step response at F = 0.85, yD = 0.96" /></td>
</tr>
</tbody>
</table>

**Figure 13** Step responses at \(F = 0.95, y_D = 0.98\) (see online version for colours)

<table>
<thead>
<tr>
<th>LT</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image3" alt="Step response at F = 0.95, yD = 0.98" /></td>
<td><img src="image4" alt="Step response at F = 0.95, yD = 0.98" /></td>
</tr>
</tbody>
</table>

**Figure 14** Step responses at \(F = 1.05, y_D = 0.96\) (see online version for colours)

<table>
<thead>
<tr>
<th>LT</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image5" alt="Step response at F = 1.05, yD = 0.96" /></td>
<td><img src="image6" alt="Step response at F = 1.05, yD = 0.96" /></td>
</tr>
</tbody>
</table>

**Figure 15** Step responses at \(F = 1.15, y_D = 0.98\) (see online version for colours)

<table>
<thead>
<tr>
<th>LT</th>
<th>VB</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image7" alt="Step response at F = 1.15, yD = 0.98" /></td>
<td><img src="image8" alt="Step response at F = 1.15, yD = 0.98" /></td>
</tr>
</tbody>
</table>

It is obvious that the LPV model with polynomial weights has very poor step response fittings and even has wrong gain signs at many operating points. If such a model is used for control at these operating points, the results would be disastrous. This result demonstrated that though using polynomial function can yield smaller output error than linear and Gaussian weights, sometimes it may suffer from the problem of poor step response fittings. Therefore the polynomial weights should be checked carefully before LPV model interpolation. This result reveals an unexpected phenomenon: when identifying a nonlinear process, a model that fits the test data almost perfectly may give very wrong model in the test range and be useless for control purposes. This should set off an alarm bell for researchers in nonlinear system identification. Checking step responses is a good way to find out the flaws of models and this is often used by industrial control engineers.

**4 Conclusions**

Many industrial processes do not have any steady state during their operating procedures. For the purpose of finding a useful identification method for such processes,
this research studies the identification method for multi-model LPV models with two scheduling variables using transition test. The LPV model accuracy is analysed by modelling a rigorous model of a high purity distillation column. Three kinds of weighing functions: linear, polynomial and Gaussian function are used and compared. The case study has illustrated that the LPV models can yield good result with respect to simulation errors and accurate step responses. Among the three weighting functions, the Gaussian weighting function performs the best. When the LPV model with polynomial weight is used, although the model outputs can be well fitted to the real system, model gain sign is often wrong. This is because the polynomial weighting function is unconstrained and the local model weightings may be much less than zero. Based on this result, in nonlinear process identification, the simulation error alone may be not enough for model validation. Other measures related to process knowledge, such as step responses, should be considered. In order to solve the problem of model gain error of the polynomial-weight-based LPV model, constraint of the weights of local linear models will be considered in the future studies. Furthermore, the LPV models obtained in this paper will be used in some model-based controllers, such as model predictive controller or internal model controller.

Acknowledgements

This project was funded by the Science and Technology Project of Fujian (Nos. JK2015034 and 2015J01275), National Natural Science Project of Xiamen University of Technology (No. XYK201402) and Foreign Cooperation Projects of Xiamen University of Technology (Nos. E201400900 and E201401400)

References


