Combined state and parameter estimation for Hammerstein systems with time-delay using the Kalman filtering

Junxia Ma¹, Feng Ding¹∗†, Weili Xiong¹ and Erfu Yang²

¹Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), School of Internet of Things Engineering, Jiangnan University, Wuxi 214122, PR China
²Department of Design, Manufacture and Engineering Management, Space Mechatronic Systems Technology Laboratory, Strathclyde Space Institute, University of Strathclyde, Glasgow G1 1XJ, Scotland, United Kingdom

SUMMARY

This paper discusses the state and parameter estimation problem for a class of Hammerstein state space systems with time-delay. Both the process noise and the measurement noise are considered in the system. Based on the observable canonical state space form and the key term separation, a pseudo-linear regressive identification model is obtained. For the unknown states in the information vector, the Kalman filter is used to search for the optimal state estimates. A Kalman-filter based least squares iterative and a recursive least squares algorithms are proposed. Extending the information vector to include the latest information terms which are missed for the time-delay, the Kalman-filter based recursive extended least squares algorithm is derived to obtain the estimates of the unknown time-delay, parameters and states. The numerical simulation results are given to illustrate the effectiveness of the proposed algorithms. Copyright © 2016 John Wiley & Sons, Ltd.

KEY WORDS: Parameter identification; Kalman filter; State estimation; Least squares; Hammerstein state space model

1. INTRODUCTION

The Hammerstein model consists of an input static nonlinear block in series with a dynamic linear subsystem and can describe a wide variety of practical nonlinear systems, e.g., wind turbines [1], valve actuators [2] and power amplifiers [3]. Despite its simplicity, as a block-oriented nonlinear system, the Hammerstein model can include many different kinds of components in the nonlinear and linear blocks. The memoryless nonlinearities include polynomial, piecewise linear descriptions, saturation, preload, deadzone, backlash and so on [4, 5]. The linear subsystems may be some parametric forms, such as state space representations, transfer functions, FIR, IIR and so on [6, 7].

Much research work has been performed on the identification of Hammerstein models. For example, Wang and Zhang studied an improved least squares identification algorithm for multivariable Hammerstein output error moving average systems by using the Taylor expansion on a quadratic criterion function and defining the information vector as the derivative of the noise variable to the unified parameter vector [8]; Chen et al derived a hierarchical gradient parameter estimation algorithm for Hammerstein nonlinear systems using the key term separation principle [9];
Liu and Bai presented a normalized iterative identification algorithm for Hammerstein systems which nonlinearities are with nonsmooth piecewise-linear structures [10].

In practical applications, disturbances widely exist [11, 12]. Especially in the area of signal processing and parameter estimation [13], the measured output always contain the disturbances from process environments [14]. The disturbance can be white noise or colored noise. Both the process noise and the measurement noise will bring some influence to the result of the systems to be identified. For the parameter estimation, the measurement noise has been widely discussed in the literatures [15, 16, 17], but the process noise is few considered in the model structure. For example, Hu et al. studied a parameter estimation problem for a Wiener system which is disturbed by the moving average measurement noise and derived two recursive extended least squares parameter estimation algorithms based on the over-parameterization models [18]; Li derived a maximum likelihood parameter estimation algorithm for Hammerstein systems which are disturbed by autoregressive moving average noise [19].

The state space models can describe dynamic linear and nonlinear systems [20, 21, 22] and play an important part in system identification [23] and signal filtering [24, 25]. The identification of the state space models has received much attention [26]. Chen et al. discussed the parameter and state estimation problem of a single-input single-output dual-rate system with time-delay based on the gradient search and the least squares principle [27]; Xie and Yang derived a gradient-based iterative identification algorithm for nonuniform sampling state space models [28]. In the field of nonlinear state space models, Schön et al. derived an expectation maximization (EM) algorithm under the framework of a maximum likelihood for the parameter estimation of a class of nonlinear state space dynamic systems [29]; Deng and Huang studied the identification problem of nonlinear parameter varying state space models with missing output data by using the particle filter to compute the expectation functions under the scheme of the EM algorithm [30].

For the input nonlinear state space systems, Wang and Ding derived an over-parameterization model based stochastic gradient algorithm to obtain the parameter estimates, but they did not consider the process noise and the time-delay in the model structure [31]. On the basis of the work in [31], this paper investigates the state and parameter estimation problem for the input nonlinear Hammerstein systems with time-delay. The difficulties are that the system not only contains the unknown parameters but also the unknown system states and the time-delay. The Kalman-filter based least squares iterative (LSI) algorithm and recursive least squares (RLS) algorithm are derived for the combined estimation of the state and parameter. For the unknown time-delay, a Kalman-filter based recursive extended least squares (KF-RELS) algorithm is proposed by extending the information vector and the parameter vector. The proposed algorithms are different from the least squares algorithm in [32], which decomposes the bilinear cost function into three linear functions by using the hierarchical identification principle and uses the state observer to get the estimates of the unknown states. Also, the proposed algorithms are different from the over-parameterization based recursive least squares algorithm in [33], which ignores the process noise in the model structure and the influence of the measurement noise in the process of updating the estimates of the system states.

The main contributions of this paper are as follows.

- A more common model structure is considered which contains both process noise and measurement noise. By using the key term separation technique, the output of the system is expressed as a linear combination of all the unknown parameters. Then a pseudo-linear regressive identification model is obtained;
- For the known and unknown time-delay, a recursive least squares algorithm and a recursive extended least squares algorithm are derived to identify the unknown parameters;
- By using the Kalman filter, a joint state and parameter estimation algorithm is presented to obtain the estimates of the unknown parameters both in the linear subsystem and in the memoryless nonlinear block as well as the unmeasured system states.

The rest of this paper is organized as follows. Section 2 gives the identification model of Hammerstein systems with dynamic state space subsystem. Section 3 derives the Kalman-filter based LSI and RELS identification algorithms for a class of Hammerstein nonlinear systems. The
KF-RELS algorithm is proposed in Section 4. Section 5 provides two numerical examples to show the effectiveness of the proposed algorithms. Finally, Section 6 offers some concluding remarks.

2. SYSTEM DESCRIPTION AND IDENTIFICATION MODEL

Let us define some notation. “$A := X$” or “$X := A$” stands for “$A$ is defined as $X$”; $\hat{x}_k$ denotes the estimate of $x$ at time $k$; $\hat{x}_k^s$ denotes the estimate of $x_k$ at iteration $s$; the symbol $I (I_n)$ stands for an identity matrix of appropriate sizes ($n \times n$); the superscript $T$ denotes the matrix/vector transpose; $1_n$ represents an $n$-dimensional column vector whose elements are 1.

Consider the following Hammerstein nonlinear system as shown in Figure 1, which is composed of a static nonlinear block followed by a linear dynamic subsystem. The linear subsystem is described by a state space model with moving average measurement noise. The time-delay is considered in the output block.

![Hammerstein state space model with time-delay](image)

\[ x_{k+1} = Ax_k + bu_k + w_k, \]
\[ y_k = cx_{k-\tau} + \sum_{i=1}^{n_d} d_i v_{k-i} + v_k, \]

where $y_k$ is the measured output, $\bar{u}_k$ is the output of the static nonlinear block, $w_k := [w_{1,k}, w_{2,k}, \cdots, w_{n,k}]^T \in \mathbb{R}^n$ and $v_k$ represent mutually independent process noise and measurement noise with zero mean and variance $Q$ and $R$, respectively. \( x_k := [x_{1,k}, x_{2,k}, \cdots, x_{n,k}]^T \in \mathbb{R}^n \) is the system state vector. \( A \in \mathbb{R}^{n \times n} \) is the system parameter matrix, \( b \in \mathbb{R}^n \) and \( c \in \mathbb{R}^{1 \times n} \) are the parameter vectors,

\[
A := \begin{bmatrix}
-a_1 & 1 & 0 & \cdots & 0 \\
-a_2 & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
-a_{n-1} & 0 & \cdots & 0 & 1 \\
-a_n & 0 & \cdots & \cdots & 0
\end{bmatrix} \in \mathbb{R}^{n \times n}, \quad b := \begin{bmatrix}
b_1 \\
b_2 \\
\vdots \\
b_{n-1} \\
b_n
\end{bmatrix} \in \mathbb{R}^n,
\]

\[
c := [1, 0, 0, \cdots, 0] \in \mathbb{R}^{1 \times n}.
\]

Assume that the output of the nonlinear block is a linear function of the known basis:

\[
\bar{u}_k = \gamma_1 f_1(u_k) + \gamma_2 f_2(u_k) + \cdots + \gamma_m f_m(u_k) = \sum_{j=1}^m \gamma_j f_j(u_k).
\]

where $\gamma'_s$ are the unknown coefficients and $f_i(u_k)'s$ are the base functions. Let

\[
\gamma := [\gamma_1, \gamma_2, \cdots, \gamma_m]^T \in \mathbb{R}^m, \\
f(u_k) := [f_1(u_k), f_2(u_k), \cdots, f_m(u_k)] \in \mathbb{R}^{1 \times m}.
\]
From (1), we have
\begin{align}
x_{1,k+1} &= -a_1 x_{1,k} + x_{2,k} + b_1 \bar{u}_k + w_{1,k}, \quad (4) \\
x_{2,k+1} &= -a_2 x_{1,k} + x_{3,k} + b_2 \bar{u}_k + w_{2,k}, \\
&\vdots \\
x_{i,k+1} &= -a_i x_{1,k} + x_{i+1,k} + b_i \bar{u}_k + w_{i,k}, \quad (5) \\
&\vdots \\
x_{n,k+1} &= -a_n x_{1,k} + b_n \bar{u}_k + w_{n,k}.
\end{align}

Multiplying \( z^{-j-1} \) on both sides of (5) gives
\begin{align}
x_{i,k-j} &= -a_i x_{1,k-j-1} + x_{i+1,k-j-1} + b_i \bar{u}_{k-j-1} + w_{i,k-j-1}, \quad j = 0, 1, 2, \ldots, n. \quad (6)
\end{align}

Substituting (6) into (4), we have
\begin{align}
x_{1,k+1} &= -a_1 x_{1,k} - a_2 x_{1,k-1} + x_{3,k-1} + b_2 \bar{u}_{k-1} + b_1 \bar{u}_k + w_{2,k-1} + w_{1,k} \\
&= -a_1 x_{1,k} - a_2 x_{1,k-1} - a_3 x_{1,k-2} + x_{4,k-2} + b_3 \bar{u}_{k-2} + b_2 \bar{u}_{k-1} + b_1 \bar{u}_k + w_{3,k-2} \\
&\quad + w_{2,k-1} + w_{1,k} \\
&= -a_1 x_{1,k} - a_2 x_{1,k-1} - \cdots - a_n x_{1,k-n+1} + b_n \bar{u}_{k-n+1} + \cdots + b_2 \bar{u}_{k-1} + b_1 \bar{u}_k \\
&\quad + w_{n,k-n+1} + \cdots + w_{1,k} \\
&= -\sum_{i=1}^{n} a_i x_{1,k-i+1} + \sum_{i=1}^{n} b_i \sum_{j=1}^{m} \gamma_{j,i} f_j(u_{k-i+1}) + \sum_{i=1}^{n} w_{i,k-i+1}. \quad (7)
\end{align}

Multiplying both sides of (7) by \( z^{-\tau-1} \) gives
\begin{align}
x_{1,k-\tau} &= -\sum_{i=1}^{n} a_i x_{1,k-\tau-i} + \sum_{i=1}^{n} b_i \sum_{j=1}^{m} \gamma_{j,i} f_j(u_{k-\tau-i}) + \sum_{i=1}^{n} w_{i,k-\tau-i}. \quad (8)
\end{align}

Substituting (8) into (2) gives
\begin{align}
y_k &= x_{1,k-\tau} + D(z)v_k \\
&= -\sum_{i=1}^{n} a_i x_{1,k-\tau-i} + \sum_{i=1}^{n} b_i \sum_{j=1}^{m} \gamma_{j,i} f_j(u_{k-\tau-i}) + \sum_{i=1}^{n} w_{i,k-\tau-i} + \sum_{i=1}^{n} d_{n,i} v_{k-i} + v_k. \quad (9)
\end{align}

Note that the model in (9) contains the product of the parameter vectors \( \mathbf{b} \) and \( \mathbf{\gamma} \) of the linear and nonlinear blocks, which increases the complexity of identification. Besides, any identification scheme cannot distinguish \( (\mathbf{b}, \mathbf{\gamma}) \) from \( (\alpha \mathbf{b}, \mathbf{\gamma}/\alpha) \) for some nonzero and finite constant \( \alpha \), because any pair \( (\alpha \mathbf{b}, \bar{u}_k/\alpha) \) would produce identical input and output measurements. Therefore, in order to get unique parameter estimates, one of the entries of \( \mathbf{b} \) and \( \mathbf{\gamma} \) has to be fixed. Here, adopt the key term separation technique \( [34, 35] \) and let the first element of the vector \( \mathbf{b} \) equal 1; i.e., \( b_1 = 1 \). Then Equation (9) can be rewritten as
\begin{align}
y_k &= -\sum_{i=1}^{n} a_i x_{1,k-\tau-i} + \sum_{j=1}^{m} \gamma_{j,i} f_j(u_{k-\tau-i}) + \sum_{i=2}^{n} b_i \bar{u}_{k-\tau-i} + \sum_{i=1}^{n} w_{i,k-\tau-i} + \sum_{i=1}^{n} d_{n,i} v_{k-i} + v_k. \quad (10)
\end{align}

Let \( n_1 = 2n + m + n_d - 1 \) and define parameter vectors and information vectors as
\begin{align}
\mathbf{\vartheta} &:= [a_1, a_2, \cdots, a_n, \mathbf{\gamma}^T, b_2, b_3, \cdots, b_n]^T \in \mathbb{R}^{2n+m-1}, \\
\mathbf{d} &:= [d_1, d_2, \cdots, d_{n_d}]^T \in \mathbb{R}^{n_d}, \\
\mathbf{\varphi}_k &:= [-x_{1,k-\tau-1}, -x_{1,k-\tau-2}, \cdots, -x_{1,k-\tau-n}, f(u_{k-\tau-1}), \bar{u}_{k-\tau-2}, \bar{u}_{k-\tau-3}, \cdots, \bar{u}_{k-\tau-n}, \Phi_k^T]^T \in \mathbb{R}^{n_1},
\end{align}
expresses the parameter vector as $\theta \in \mathbb{R}^{n_1}$, the number of the unknown parameters in the vector $\theta$. There are two steps in the Kalman filter, one is called the time update (or prediction), the other is called the measurement update (or modification). The Kalman filter can give the state estimates $\hat{x}_k$ of the state vector $x_k$. 

3. THE KALMAN-FILTER BASED LEAST SQUARES ALGORITHMS WITH KNOWN TIME-DELAY

In process control industry, the phenomenon of time-delay often exists in the procedure of signal transmission. Based on the empirical knowledge, it is assumed that the time-delay is known. In this section, we derive the Kalman-filter based least squares algorithms to identify the Hammerstein dynamic system with known time-delay.

3.1. The state estimation algorithm

The Kalman filter can give the state estimates $\hat{x}_k$. There are two steps in the Kalman filter, one is called the time update (or prediction), the other is called the measurement update (or modification). On the prediction step, the prior estimates of the state and its covariance are predicted; on the modification step, the newest measurement and prior estimates are combined together to modify the posterior estimates.

Because the parameter matrix/vectors are unknown, we need to use the estimated parameter vector $\hat{\theta}_k$ to construct the estimates $\hat{A}_k$ and $\hat{b}_k$ of $A$ and $b$, and use the estimated parameter matrix $\hat{A}_k$ and the estimated parameter vector $\hat{b}_k$ to compute the estimate $\hat{x}_{k+1}$ of the state vector $x_{k+1}$. Here, the
Kalman filter is used to generate the optimal state estimate. Considering the time delay \( \tau \) in the output, we can use the following Kalman filter to obtain the state estimate \( \hat{x}_{k+1} \) \[23\]:

\[
\begin{align*}
\hat{x}_{k+1} &= \hat{A}_k \hat{x}_k + \hat{b}_k \hat{u}_k, \quad \hat{x}_1 = \frac{1}{p_0} \\
P_k &= \hat{A}_k P_k \hat{A}_k^T + Q, \quad P_1 = I_n \\
L_k &= P_k c^T (c \hat{P}_k c^T + R)^{-1}, \\
\hat{x}_{k+1} &= \hat{x}_{k+1} + L_k (y_{k+1+\tau} - c \hat{x}_{k+1}), \\
P_{k+1} &= \hat{P}_k - L_k c \hat{P}_k.
\end{align*}
\]

(13) (14) (15) (16) (17)

where \( \hat{x}_{k+1} \) is a prior estimate of the state \( x_{k+1} \) for given measurements up to and including time \( k \); \( \hat{x}_{k+1} \) is a posterior estimate of the state \( x_{k+1} \) for given measurements up to and including time \( k + 1 \); \( P_k \) is the variance of the prior estimation error; \( P_{k+1} \) is the variance of the posterior estimation error; \( L_k \) is the Kalman gain vector.

Another frequently used method for state estimation is to use the state observer \[32\] which can be used to get the approximate values of the system states. The drawback of the state observer is that it is generally suitable for deterministic systems. Thus this paper uses the parameter estimates based Kalman filter for generating the state estimates in the stochastic framework.

3.2. The iterative parameter estimation algorithm

Let \( s = 1, 2, 3, \cdots \) be an iterative variable and \( \hat{\theta}_k^s \) be the estimate of \( \theta \) at iteration \( s \). Minimizing the criterion function \( J_1(\theta) \) and letting the derivative of \( J_1(\theta) \) with respect to \( \theta \) be zero gives

\[
\frac{\partial J_1(\theta)}{\partial \theta} \bigg|_{\theta = \hat{\theta}_k^s} = -2 \hat{P}_k^T (\hat{Y}_{k,L} - \hat{X}_{k,L} \hat{\theta}_k^s) = 0.
\]

Then we can obtain the least squares estimate of the parameter vector \( \theta \):

\[
\hat{\theta}_k^s = [\Phi_{k,L}^T \Phi_{k,L}]^{-1} \Phi_{k,L}^T \hat{Y}_{k,L}.
\]

(20)

But we cannot obtain the estimate \( \hat{\theta}_k^s \) directly from (20), because the information vector \( \phi_k \) in \( \Phi_{k,L} \) contains the unknown state variable \( \tilde{a}_{1,k-\tau-1} \), the unknown output \( \tilde{u}_{k-\tau} \) of the nonlinear block and the unknown noise terms \( \tilde{v}_{k-j} \). The scheme here is to replace \( x_{1,k-\tau-1} \), \( \tilde{u}_{k-\tau} \) and \( \tilde{v}_{k-j} \) in \( \phi_k \) with their corresponding estimates \( \hat{x}_{1,k-\tau-1}, \hat{u}_{k-\tau}, \cdots \hat{u}_{k-\tau-n}, f(\hat{u}_{k-\tau-1}), \cdots, \hat{v}_{k-\tau-1}, \cdots, \hat{v}_{k-\tau-n} \), \( \hat{v}_{k-1} \), respectively.

Replacing the unknown coefficient $\gamma_j$ in (3) with its estimate $\hat{\gamma}_{j,k}$ gives
\[
\hat{v}_k = \gamma_{1,k} f_1(u_k) + \gamma_{2,k} f_2(u_k) + \cdots + \gamma_{m,k} f_m(u_k).
\]
From (2), we have
\[
v_k = y_k - c x_{k-\tau} - d^T \phi_k. \tag{21}
\]
Replace $x_{k-\tau}$, $\phi_k$ and $d$ in (21) with their estimates $\hat{x}_{k-\tau}^s$, $\hat{\phi}_k^s$ and $\hat{d}_k^s$, the estimate $\hat{v}_k^s$ can be computed by
\[
\hat{v}_k^s = y_k - c \hat{x}_{k-\tau}^s - (\hat{d}_k^s)^T \hat{\phi}_k^s.
\]
Define
\[
\hat{\Phi}_{k,L}^s := [\hat{\phi}_k^s, \hat{\phi}_{k+1}^s, \cdots, \hat{\phi}_{k+L-1}^s]^T \in \mathbb{R}^{L \times n_1}.
\]
Replacing the information matrix $\hat{\Phi}_{k,L}^s$ in (20) with its estimate $\hat{\Phi}_{k,L}^s$ and using the Kalman filter to obtain the estimates of the unknown states, we can get the Kalman-filter based least squares iterative (KF-LSI) algorithm for estimating the parameters and states of the Hammerstein state space models:
\[
\hat{\Theta}_k = [\hat{\Phi}_{k,L}^s]^T \hat{\Phi}_{k,L}^s \hat{\Phi}_{k,L}^s \hat{Y}_{k,L}, \tag{22}
\]
\[
\hat{Y}_{k,L} = [y_k, y_{k+1}, \cdots, y_{k+L-1}]^T, \tag{23}
\]
\[
\hat{\Phi}_{k,L}^s = [\hat{\phi}_k^s, \hat{\phi}_{k+1}^s, \cdots, \hat{\phi}_{k+L-1}^s]^T, \tag{24}
\]
\[
\hat{\phi}_k^s = [\hat{x}_{1,k-1}^s, \hat{x}_{1,k-2}^s, \cdots, \hat{x}_{1,k-n}^s, \hat{u}_{k-\tau-1}^s, \hat{u}_{k-\tau-2}^s, \cdots, \hat{u}_{k-\tau-n}^s, \hat{\gamma}_{1,k-1}^s, \hat{\gamma}_{2,k-1}^s, \cdots, \hat{\gamma}_{m,k-1}^s, \hat{\phi}_k^s, \cdots, \hat{\phi}_{k+L-1}^s]^T, \tag{25}
\]
\[
\hat{\phi}_k = y_k - c \hat{x}_{k-\tau}^s - (\hat{d}_k^s)^T \hat{\phi}_k^s, \tag{27}
\]
\[
\hat{\Theta}_k = [\hat{a}_{1,k}^s, \hat{a}_{2,k}^s, \cdots, \hat{a}_{n,k}^s, \hat{\gamma}_{1,k}^s, \hat{\gamma}_{2,k}^s, \cdots, \hat{\gamma}_{m,k}^s, \hat{b}_{2,k}^s, \cdots, \hat{b}_{n,k}^s, \hat{d}_{1,k}^s, \cdots, \hat{d}_{n,\tau,k}^s]^T, \tag{28}
\]
\[
\hat{A}_k^s = \begin{bmatrix}
-\hat{a}_{1,k}^s & 1 & 0 & \cdots & 0 \\
-\hat{a}_{2,k}^s & 0 & 1 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \vdots \\
-\hat{a}_{n-1,k}^s & 0 & \cdots & 0 & 1 \\
-\hat{a}_{n,k}^s & 0 & \cdots & 0 & 0
\end{bmatrix}, \tag{29}
\]
\[
\hat{b}_k^s = [1, \hat{b}_{2,k}^s, \hat{b}_{3,k}^s, \cdots, \hat{b}_{n,k}^s]^T, \tag{30}
\]
\[
\hat{x}_{k+1} = \hat{A}_k^s \hat{x}_k^s + \hat{b}_k^s \hat{u}_k, \quad \hat{x}_1^s = I_n/p_0, \tag{31}
\]
\[
\hat{P}_k = \hat{A}_k^s \hat{P}_k^s (\hat{A}_k^s)^T + Q, \quad \hat{P}_1 = I_n, \tag{32}
\]
\[
\hat{L}_k = \hat{P}_k^s (c \hat{P}_k^s c^T + R)^{-1}, \tag{33}
\]
\[
\hat{x}_{k+1} = \hat{x}_{k+1} + \hat{L}_k(y_{k+1} + c \hat{x}_{k+1}), \tag{34}
\]
\[
\hat{P}_{k+1} = \hat{P}_k - \hat{L}_k c \hat{P}_k. \tag{35}
\]

**Remark 1** The iterative algorithm is implemented off-line, it repeatedly uses a batch of observed data and can get good parameter estimates after only several iterations [37, 38, 39, 40]. In the LSI algorithm, the measured input-output data should be collected in advance.

The flowchart of computing the parameter estimates $\hat{\Theta}_k^s$ and the state estimate $\hat{x}_k^s$ by the KF-LSI algorithm is shown in Figure 2.
Start: set $\tau$ and $\epsilon$

Initialize: $s = 1$, give $\hat{x}^1_k, \hat{\bar{u}}_{i-1}^0$, $\vec{x}^0_{1-i}$ and $\vec{y}^0_{i-1}, i = 0, 1, 2, \ldots$

Collect input and output data

Construct $\phi^s_k, \Phi^s_{k,L}$ and $Y_{k,L}$

Update $\hat{\theta}^s_k$

Construct $\hat{A}^s_k$ and $\hat{b}^s_k$

Compute $\bar{x}^s_{k+1}$ and $\bar{P}_{k+1}$

Compute $L_k$, update $\bar{x}^s_{k+1}$

Update $P_{k+1}$

Update $\hat{u}^s_k$ and $\hat{v}^s_k$

$||\hat{\theta}^s_k - \hat{\theta}^{s-1}_k|| > \varepsilon$?

Yes

Obtain the estimate $\hat{\theta}^s_k$ and $\hat{x}^s_k$

$k := k + 1$

No

Figure 2. The flowchart of the KF-LSI algorithm for computing $\hat{\theta}^s_k, \hat{x}^s_k$

3.3. The recursive parameter estimation algorithm

Unlike the LSI algorithm, the RLS algorithm may be carried out on-line. By defining a covariance matrix, the RLS algorithm uses the input and output data to update the parameter estimates at each step. In this section, a Kalman filter based RLS algorithm is derived.

Similarly, according to (11), define a quadratic criterion function

$$ J_2(\theta) = \sum_{j=1}^{k} [y(j) - \varphi_j^T \theta]^2. $$

Minimizing $J_2(\theta)$ using the least squares principle, letting the partial derivative of $J_2(\theta)$ with respect to $\theta$ be zero, we can obtain

$$ \dot{\theta}_k = \dot{\theta}_{k-1} + L_{1,k}[y_k - \varphi^T_k \dot{\theta}_{k-1}], $$

(36)
Because the information vector \( \varphi_k \) contains the unmeasurable state variable \( x_{1,k-\tau-i} \), unknown output of nonlinear block \( \tilde{u}_{k-i} \) and the unknown noise \( v_{k-j} \), the algorithm in (36)–(38) cannot be implemented. Here we replace \( x_{1,k-\tau-i}, \tilde{u}_{k-i} \) and \( v_{k-j} \) in \( \varphi_k \) with their corresponding estimates \( \hat{x}_{1,k-\tau-i}, \hat{\tilde{u}}_{k-i} \) and \( \hat{v}_{k-j} \) at time \( k \) and define

\[
\hat{\varphi}_k := \begin{bmatrix} \hat{x}_{1,k-\tau-1}, \ldots, \hat{x}_{1,k-\tau-n}, f(u_{k-\tau-1}), \hat{\tilde{u}}_{k-\tau-2}, \ldots, \hat{\tilde{u}}_{k-\tau-n}, \\
\hat{v}_{k-1}, \ldots, \hat{v}_{k-n}\end{bmatrix}^T \in \mathbb{R}^{n_1}.
\]

Replacing the unknown coefficient \( \gamma_j \) in (3) with its estimate \( \hat{\gamma}_{j,k} \) gives

\[
\hat{u}_k = \hat{\gamma}_{1,k} f_1(u_k) + \hat{\gamma}_{2,k} f_2(u_k) + \cdots + \hat{\gamma}_{m,k} f_m(u_k).
\]

Replacing \( x_{k-\tau}, \phi_k \) and \( d \) in (21) with their estimates \( \hat{x}_{k-\tau}, \hat{\phi}_k \) and \( \hat{d}_k \), the estimate \( \hat{v}_k \) can be computed by

\[
\hat{v}_k = y_k - c \hat{x}_{k-\tau} - d^T \hat{\phi}_k.
\]

Then we can summarize the following RLS algorithm for estimating the parameter vector \( \theta \) as

\[
\begin{align*}
\theta_k &= \theta_{k-1} + L_{1,k} [y_k - \varphi_k^T \theta_{k-1}], \\
L_{1,k} &= P_{1,k-1} \varphi_k [1 + \varphi_k^T P_{1,k-1} \varphi_k]^{-1}, \\
P_{1,k} &= [I_n - L_{1,k} \varphi_k^T] P_{1,k-1} - \gamma_{1,k} f_1(u_k) + \cdots + \gamma_{m,k} f_m(u_k). \\
\hat{\theta}_k &= \hat{\theta}_{k-1} + \hat{L}_{1,k} [y_k - \hat{\varphi}_k^T \hat{\theta}_{k-1}], \\
\hat{L}_{1,k} &= \hat{P}_{1,k-1} \hat{\varphi}_k [1 + \hat{\varphi}_k^T \hat{P}_{1,k-1} \hat{\varphi}_k]^{-1}, \\
\hat{P}_{1,k} &= [I_n - \hat{L}_{1,k} \hat{\varphi}_k^T] \hat{P}_{1,k-1} - \hat{\gamma}_{1,k} f_1(u_k) + \cdots + \hat{\gamma}_{m,k} f_m(u_k), \\
\hat{\varphi}_k &= \begin{bmatrix} \hat{a}_{1,k}, \hat{a}_{2,k}, \ldots, \hat{a}_{n,d,k} \end{bmatrix}^T,
\end{align*}
\]

For the unknown state \( x_{1,k} \), we also use the Kalmam filter algorithm to generate its estimate \( \hat{x}_{1,k} \). Combining Equations (41)–(47) and (13)–(19), we can get the Kalman-filter based recursive least squares (KF-RLS) algorithm for identifying the Hammerstein state space systems.

To initialize the KF-RLS algorithm, the initial value \( \theta_0 \) and \( \hat{x}_1 \) is generally taken to be a real vector, e.g., \( \theta_0 = 1_{n_1}/p_0 \) and \( \hat{x}_1 = 1_n/p_0 \) with \( p_0 \) being normally a large positive number (e.g., \( p_0 = 10^6 \)). Let \( \hat{x}_{1,i} = 1/p_0 \), \( \hat{\tilde{u}}_{j,i} = 1/p_0 \) and \( \hat{v}_{j,j} = 1/p_0 \) for \( i = 0, 1, 2, \ldots, j = 0, 1, 2, \ldots, n_d-1 \). The initial values of the covariance matrices are set as \( P_{1,0} = p_0 I_{n_1}, P_1 = I_n \).

4. THE REDUNDANT RELS ALGORITHM FOR HAMMERSTEIN SYSTEMS WITH UNKNOWN TIME-DELAY

Although we can predict the time-delay based on the empirical data in some special cases, there are many uncertain varying elements which bring the unknown time-delay in the actual systems. Besides identifying the system parameters, we also need to get the estimate of the time-delay based on the observed data.

Reconsidering (11), if we extend the information vector \( \varphi_k \) and supplement the lost terms which are omitted because of the time-delay. The information vector \( \varphi_k \) is redefined as

\[
\varphi_k := \begin{bmatrix} -x_{1,k-1}, -x_{1,k-2}, \ldots, -x_{1,k-\tau-2}, -x_{1,k-\tau-1}, u_{k-\tau-2}, \ldots, u_{k-\tau-n}, v_{k-\tau-1}, v_{k-\tau-2}, \ldots, v_{k-\tau-n}, u_{k-n_d} \end{bmatrix}^T \in \mathbb{R}^{n_1+\tau}.
\]

At the same time, the parameter vector \( \theta \) is extended as the following form:

\[
\theta := [a_{01}, a_{02}, \ldots, a_{0\tau}, a_1, a_2, \ldots, a_n, \gamma_1, \gamma_2, \ldots, \gamma_m, b_2, b_3, \ldots, b_n, d_1, d_2, \ldots, d_{n_d}]^T \in \mathbb{R}^{n_1+\tau}.
\]
The dimension of $\theta$ is larger than the dimension of the original parameter vector $\theta$ because of the redundant parameters $\{a_{01}, a_{02}, \cdots, a_{0r}\}$. In fact, the true values of the redundant parameters should be zeros to guarantee the model structure unchanged. So the number of the redundant parameters is the value of the time-delay $\tau$. Once the estimates of the redundant parameters lie in a given confidence interval, we can assume them as zero. So the number of the continuous approximate zero elements in the front part of the estimate of $\theta$ is the estimate of the time-delay $\tau$.

Then our objective is how to obtain the estimate $\hat{\theta}_k$ of $\theta_k$.

Rewrite the model in (11) as

$$y_k = \varphi_k^T \theta + e_k.$$ \hfill (48)

Define a quadratic criterion function

$$J_3(\theta) := \sum_{j=1}^k [y(j) - \varphi_j^T \theta]^2.$$ \hfill (49)

Minimizing $J_3(\theta)$ and using the least squares principle, we can get the following relations:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + L_{2,k} [y_k - \varphi_k^T \hat{\theta}_{k-1}],$$ \hfill (50)

$$\hat{\theta}_{k} = \hat{\theta}_{k-1} + L_{2,k} [y_k - \varphi_k^T \hat{\theta}_{k-1}],$$ \hfill (51)

Similarly, the information vector $\varphi_k$ contains the unknown $x_{1,k-i}, \bar{u}_{k-i}$ and $v_{k-j}$, the algorithm in (49)–(51) cannot be implemented. Replace $x_{1,k-i}, \bar{u}_{k-i}$ and $v_{k-j}$ in $\varphi_k$ with their corresponding estimates $\hat{x}_{1,k-i}, \hat{u}_{k-i}$ and $\hat{v}_{k-j}$ at time $k$ and define

$$\hat{\varphi}_k := [-\hat{x}_{1,k-1}, -\hat{x}_{1,k-2}, \cdots, -\hat{x}_{1,k-\tau-1}, -\hat{x}_{1,k-\tau-2}, \cdots, -\hat{x}_{1,k-\tau-n}, f(u_k-\tau-1),$$

$$\hat{u}_{k-\tau-2}, \cdots, \hat{u}_{k-\tau-n}, \hat{v}_{k-1}, \cdots, \hat{v}_{k-n_d}]^T \in \mathbb{R}^{n_1+\tau}.$$ \hfill (55)

Replacing $\varphi_k$ in (49)–(51) with its estimate $\hat{\varphi}_k$, we can summarize the following recursive extended least squares (RELS) algorithm for estimating $\hat{\theta}$:

$$\hat{\theta}_k = \hat{\theta}_{k-1} + L_{2,k} [y_k - \varphi_k^T \hat{\theta}_{k-1}],$$ \hfill (52)

$$\hat{\theta}_{k} = \hat{\theta}_{k-1} + L_{2,k} [y_k - \varphi_k^T \hat{\theta}_{k-1}],$$ \hfill (53)

$$\hat{\varphi}_k := [-\hat{x}_{1,k-1}, -\hat{x}_{1,k-2}, \cdots, -\hat{x}_{1,k-\tau}, -\hat{x}_{1,k-\tau-1}, -\hat{x}_{1,k-\tau-2}, \cdots, -\hat{x}_{1,k-\tau-n}, f(u_k-\tau-1),$$

$$\hat{u}_{k-\tau-2}, \cdots, \hat{u}_{k-\tau-n}, \hat{v}_{k-1}, \cdots, \hat{v}_{k-n_d}]^T \in \mathbb{R}^{n_1+\tau}.$$ \hfill (55)

$$\hat{\theta}_k = [\hat{a}_{01,k}, \hat{a}_{02,k}, \cdots, \hat{a}_{0,r,k}, \hat{a}_{1,k}, \hat{a}_{2,k}, \cdots, \hat{a}_{n,k}, \hat{\gamma}_{1,k}, \cdots, \hat{\gamma}_{m,k}, \hat{b}_{2,k}, \cdots, \hat{b}_{n,k}, \hat{d}_{1,k}, \cdots, \hat{d}_{n_d,k}]^T,$$ \hfill (56)

$$\hat{u}_k = \hat{\gamma}_{1,k} f_1(u_k) + \hat{\gamma}_{2,k} f_2(u_k) + \cdots + \hat{\gamma}_{m,k} f_m(u_k),$$ \hfill (57)

$$\hat{v}_k = y_k - c \hat{x}_{k-\tau} - \hat{d}_k \hat{\theta}_k.$$ \hfill (58)

Combining (52)–(58) and (13)–(19), the Kalman-filter based recursive extended least squares (KF-RELS) algorithm for estimating $\theta$ is obtained. Then the value of the time-delay $\tau$ can be evaluated by counting how many elements which are close to zero in the front part of $\hat{\theta}_k$.

The steps for computing the state and parameter estimates $\hat{x}_k$ and $\hat{\theta}_k$ under the KF-RELS algorithm with the increasing of $k$ are as follows.

1. To initialize. Let $k = 1$, the initial value $\hat{\theta}_0$ and $\hat{x}_1$ is generally taken to be a real vector, e.g., $\theta_0 = 1_{n_1+\tau}/p_0$ and $\bar{x}_1 = 1_n/p_0$. Let $\hat{x}_{1-i} = 1/p_0$, $\bar{u}_{-i} = 1/p_0$, $\bar{v}_{-i} = 1/p_0$ and $\hat{v}_{-i} = 1/p_0$ ($i = 0, 1, 2, \cdots, j = 0, 1, 2, \cdots, n_d - 1$). The initial values of the covariance matrices are set as: $P_0 = p_0 I_{n_1+\tau}$, $P_1 = I_n$. Give a small positive $\varepsilon$. 

2. Collect the input-output data, construct the information vector \( \hat{\varphi}_k \) by (55).
3. Compute the gain vector \( L_{2,k} \) and the covariance matrix \( P_{2,k} \) by (53) and (54).
4. Update the parameter estimates \( \hat{\theta}_k \) by (52).
5. Construct the matrix \( A_k \) by (18), and the vector \( \hat{b}_k \) by (19).
6. Compute the prior estimate \( \hat{x}_{k+1} \) and the covariance matrix \( P_k \) by (13) and (14).
7. Compute the Kalman gain \( L_k \) by (15), update the posterior state estimate \( \hat{x}_{k+1} \) by (16) and the posterior covariance matrix \( P_{k+1} \) by (17).
8. Update the estimates \( \hat{u}_k \) and \( \hat{v}_k \) by (57) and (58).
9. Evaluate the relative error of parameter estimates, if

\[
\delta = \| \hat{\theta}_k - \hat{\theta}_{k-1} \| \leq \varepsilon
\]

for some pre-set small \( \varepsilon \), then terminate this procedure and obtain the parameter estimate \( \hat{\theta}_k \) and the state estimate \( \hat{x}_{k+1} \); otherwise, increase \( k \) by 1 and go to Step 2.

5. EXAMPLE

**Example 1:** Consider the following Hammerstein system with state space model:

\[
x_{k+1} = Ax_k + b \bar{u}_k + w_k,
\]

\[
y_k = c x_{k-\tau} - 0.30 v_{k-1} + v_k,
\]

\[
\bar{u}_k = \gamma_1 u_k + \gamma_2 u_k^2 + \gamma_3 u_k^3,
\]

\[
A = \begin{bmatrix} -0.50 & 1 \\ -0.26 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1.50 \end{bmatrix},
\]

\[
c = [1, 0], \quad \gamma = [0.25, 0.60, 0.76].
\]

In simulation, the input \( \{u_k\} \) is taken as a persistent excitation sequence with zero mean and unit variance, and \( \{w_k\}, \{v_k\} \) as uncorrelated noise sequences with zero mean and variances

\[
Q := \begin{bmatrix} \sigma_w^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \quad \text{and} \quad R := \sigma_v^2,
\]

respectively. The output time-delay \( \tau = 2 \). Using the KF-LSI algorithm and KF-RLS algorithm to estimate the parameters of this model, the parameter estimates and their estimation errors \( \delta := \| \hat{\theta}_k - \theta \| / \| \theta \| \) with different noise variances are shown in Tables I–II. In the KF-LSI algorithm, the data length \( L = 1000 \).

To show the advantage of Kalman filter in obtaining the estimates of unknown states, the state observer (SO) is also used to get the estimates of the unknown system states. Combining the SO with the LSI and RLS algorithm to get the SO-LSI estimates and the SO-RLS estimates. The estimation errors versus \( s \) or \( k \) are plotted in Figure 3 and Figure 4.

From Tables I–III and Figures 3–4, we can draw the following conclusions.

1. It is clear see that the estimation errors \( \delta \) are becoming smaller (in general) as \( s \) or \( k \) increasing. This indicates that the proposed KF-LSI and KF-RLS algorithms are effective for Hammerstein state space systems.
2. A smaller noise variance leads to smaller parameter estimation errors under the same data length in the KF-LSI and KF-RLS algorithm.
3. The proposed KF-LSI and KF-RLS algorithm have better performances on estimation accuracy than the SO-LSI and SO-RLS estimation algorithm.

For comparison, we use the state observer based hierarchical stochastic gradient (SO-HSG) algorithm and the hierarchical multi-innovation stochastic gradient (SO-HMISG) algorithm in [31] and the state observer based hierarchical least squares (SO-HLS) algorithm in [32] to identify this model. In order to acquire the unique estimate, they assume that the norm of the coefficient vector \( \gamma \) is unity and the first coefficient is positive, i.e., \( \| \gamma \|^2 = 1 \) and \( \gamma_1 > 0 \). The root mean square error
is employed to reflect the errors between the predicted values and true values. Using the Monte Carlo simulations with 100 set of noise realizations, the RMSE and its bias of the five different algorithms are summarized in Table III. The results of self-validation and cross-validation of the discussed different algorithms are shown in Figures 5 and 6.

From Figures 5–6 and Table III, we can see that 1) the KF-LSI algorithm has the smallest RMSE in the self-validation and cross validation, 2) the KF-RLS algorithm has better performances in prediction than other algorithms, 3) the RMSE of the SO-HSG algorithm is obvious larger than those of other four algorithms, by extending the length of the innovation, the SO-HMISG algorithm can enhance the accuracy of parameter estimation.

Table I. The KF-LSI parameter estimates and errors versus $s$

<table>
<thead>
<tr>
<th>$\sigma^2$</th>
<th>$s$</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$\gamma_1$</th>
<th>$\gamma_2$</th>
<th>$b_2$</th>
<th>$d_1$</th>
<th>$\delta$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma^2 = 0.02^2$</td>
<td>1</td>
<td>0.50024</td>
<td>0.26313</td>
<td>0.24944</td>
<td>0.59815</td>
<td>0.75849</td>
<td>1.49801</td>
<td>-0.00523</td>
</tr>
<tr>
<td>$\sigma^2 = 0.2^2$</td>
<td>2</td>
<td>0.51093</td>
<td>0.26052</td>
<td>0.24857</td>
<td>0.59917</td>
<td>0.75988</td>
<td>1.50482</td>
<td>-0.25617</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.50822</td>
<td>0.26122</td>
<td>0.24882</td>
<td>0.59880</td>
<td>0.75964</td>
<td>1.50185</td>
<td>-0.29186</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.50822</td>
<td>0.26121</td>
<td>0.24882</td>
<td>0.59875</td>
<td>0.75964</td>
<td>1.50184</td>
<td>-0.29185</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.50822</td>
<td>0.26121</td>
<td>0.24882</td>
<td>0.59876</td>
<td>0.75964</td>
<td>1.50184</td>
<td>-0.29185</td>
</tr>
<tr>
<td>$\sigma^2 = 0.05^2$</td>
<td>1</td>
<td>0.50088</td>
<td>0.27553</td>
<td>0.24811</td>
<td>0.59074</td>
<td>0.75278</td>
<td>1.49015</td>
<td>-0.02357</td>
</tr>
<tr>
<td>$\sigma^2 = 1.00^2$</td>
<td>2</td>
<td>0.50629</td>
<td>0.28097</td>
<td>0.24377</td>
<td>0.58957</td>
<td>0.75990</td>
<td>1.50340</td>
<td>-0.27305</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>0.50402</td>
<td>0.27903</td>
<td>0.24510</td>
<td>0.59069</td>
<td>0.75864</td>
<td>1.49624</td>
<td>-0.29440</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>0.50402</td>
<td>0.27897</td>
<td>0.24509</td>
<td>0.59063</td>
<td>0.75863</td>
<td>1.49619</td>
<td>-0.29440</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>0.50402</td>
<td>0.27897</td>
<td>0.24509</td>
<td>0.59064</td>
<td>0.75863</td>
<td>1.49619</td>
<td>-0.29440</td>
</tr>
<tr>
<td>True values</td>
<td>0.50000</td>
<td>0.26000</td>
<td>0.25000</td>
<td>0.60000</td>
<td>0.76000</td>
<td>1.50000</td>
<td>-0.30000</td>
<td></td>
</tr>
</tbody>
</table>

Table II. The KF-RLS parameter estimates and errors versus $k$

<table>
<thead>
<tr>
<th>$\sigma^2$</th>
<th>$k$</th>
<th>$\alpha_1$</th>
<th>$\alpha_2$</th>
<th>$\gamma_1$</th>
<th>$\gamma_2$</th>
<th>$b_2$</th>
<th>$d_1$</th>
<th>$\delta$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\sigma^2 = 0.02^2$</td>
<td>100</td>
<td>0.51279</td>
<td>0.24805</td>
<td>0.25026</td>
<td>0.53782</td>
<td>0.74856</td>
<td>1.52813</td>
<td>-0.25841</td>
</tr>
<tr>
<td>$\sigma^2 = 0.2^2$</td>
<td>200</td>
<td>0.50853</td>
<td>0.26082</td>
<td>0.24446</td>
<td>0.59100</td>
<td>0.76103</td>
<td>1.51115</td>
<td>-0.24333</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.51223</td>
<td>0.25741</td>
<td>0.24829</td>
<td>0.60433</td>
<td>0.76192</td>
<td>1.49941</td>
<td>-0.24092</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0.50970</td>
<td>0.25894</td>
<td>0.25265</td>
<td>0.60038</td>
<td>0.75789</td>
<td>1.49877</td>
<td>-0.24867</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>0.50970</td>
<td>0.25909</td>
<td>0.25178</td>
<td>0.59864</td>
<td>0.75763</td>
<td>1.50310</td>
<td>-0.26989</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>0.50774</td>
<td>0.25906</td>
<td>0.24794</td>
<td>0.60148</td>
<td>0.76008</td>
<td>1.50219</td>
<td>-0.27050</td>
</tr>
<tr>
<td>$\sigma^2 = 0.05^2$</td>
<td>100</td>
<td>0.49437</td>
<td>0.15227</td>
<td>-0.33553</td>
<td>0.65580</td>
<td>1.00088</td>
<td>1.36885</td>
<td>-0.12603</td>
</tr>
<tr>
<td>$\sigma^2 = 1.00^2$</td>
<td>200</td>
<td>0.53590</td>
<td>0.20030</td>
<td>0.13232</td>
<td>0.61041</td>
<td>1.46399</td>
<td>1.32872</td>
<td>-0.21827</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.50483</td>
<td>0.24569</td>
<td>0.22004</td>
<td>0.60475</td>
<td>0.74398</td>
<td>1.42952</td>
<td>-0.25032</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>0.49799</td>
<td>0.24748</td>
<td>0.24848</td>
<td>0.61428</td>
<td>0.75648</td>
<td>1.43737</td>
<td>-0.23780</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>0.49988</td>
<td>0.26124</td>
<td>0.23658</td>
<td>0.61713</td>
<td>0.76864</td>
<td>1.46757</td>
<td>-0.25904</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>0.49521</td>
<td>0.25721</td>
<td>0.25604</td>
<td>0.60148</td>
<td>0.76008</td>
<td>1.47596</td>
<td>-0.27036</td>
</tr>
<tr>
<td>True values</td>
<td>0.50000</td>
<td>0.26000</td>
<td>0.25000</td>
<td>0.60000</td>
<td>0.76000</td>
<td>1.50000</td>
<td>-0.30000</td>
<td></td>
</tr>
</tbody>
</table>

**Example 2:** Consider a following Hammerstein system with unknown time-delay:

\[ x_{k+1} = Ax_k + bu_k + w_k, \]
\[ y_k = cx_k - 0.18v_{k-1} + v_k, \]
\[ u_k = \gamma_1\sin(u_k) + \gamma_2\cos(u_k), \]
\[ A = \begin{bmatrix} -0.45 & 1 \\ -0.30 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1.50 \end{bmatrix}, \quad c = [1, 0], \]
\[ \gamma = [1.20, -0.32]. \]
Table III. The RMSE and its bias of different algorithms based on 100 Monte Carlo runs ($\sigma^2_w = 0.02^2$, $\sigma^2_v = 0.20^2$)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Self-validation</th>
<th>Cross-validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>KF-LSI</td>
<td>0.22374±0.00706</td>
<td>0.22602±0.02726</td>
</tr>
<tr>
<td>KF-RLS</td>
<td>0.23852±0.00708</td>
<td>0.26681±0.02815</td>
</tr>
<tr>
<td>SO-HSG</td>
<td>1.28185±0.24508</td>
<td>1.12947±0.13251</td>
</tr>
<tr>
<td>SO-HMISG(p=6)</td>
<td>0.30017±0.29869</td>
<td>0.35248±0.10866</td>
</tr>
<tr>
<td>SO-HLS</td>
<td>0.27849±0.10688</td>
<td>0.36280±0.05659</td>
</tr>
</tbody>
</table>

Figure 3. The SO-LSI and KF-LSI estimation errors $\delta$ versus $s$ ($\sigma^2_w = 0.02^2$, $\sigma^2_v = 0.20^2$)

Figure 4. The SO-RLS and KF-RLS estimation errors $\delta$ versus $k$ ($\sigma^2_w = 0.02^2$, $\sigma^2_v = 0.20^2$)

Take the same simulation conditions as those in Example 1 and let $\sigma^2_w = 0.02^2$ and $\sigma^2_v = 0.20^2$. Using the KF-RELS algorithm to estimate the parameters, states and time-delay of this system, the parameter estimates and their estimation errors are shown in Table IV. Give a confidence interval $[-0.005, 0.005]$ for the redundant parameters. Checking the last second line in Table IV, we can find there are two elements which can be assumed as zero. So the time-delay $\tau$ equals 2.

As a comparison, use the state observer to estimate the unknown system states again. Combining it with the RELS algorithm to get the SO-RELS estimates. The parameter estimates and their
estimation errors are also shown in Table IV. From Table IV, we can see that the KF-RELS algorithm generates better estimates than the SO-RELS algorithm.

For model validation, a different dataset ($L_e = 200$ samples from $t = 3001$ to $3200$) and the estimated model obtained by the KF-RELS algorithm are used. The predicted output and the true output under the KF-RELS algorithm are plotted in Figure 7. The true states and the estimated states are compared in Figure 8. From Figures 7–8, we can draw the following conclusions.

1. The model outputs can fit the true outputs well, the KF-RELS algorithm can capture the dynamic system well.
2. The estimated states are very close to the true states, the parameter estimates based Kalman filter can generate highly accurate state estimates.

6. CONCLUSIONS

A Kalman filter based least squares iterative algorithm and recursive least squares algorithm are developed for Hammerstein systems with dynamic state space model. The process noise is assumed to be white noise, the measurement noise is fitted by the moving average noise. For the unknown
Table IV. The SO-RELS and KF-RELS parameter estimates and errors versus $k$

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$k$</th>
<th>$a_{01}$</th>
<th>$a_{02}$</th>
<th>$a_1$</th>
<th>$a_2$</th>
<th>$\gamma_1$</th>
<th>$\gamma_2$</th>
<th>$b_2$</th>
<th>$d_1$</th>
<th>$\delta$ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SO-RELS</td>
<td>100</td>
<td>0.02795</td>
<td>0.18176</td>
<td>0.48168</td>
<td>-0.20607</td>
<td>1.45273</td>
<td>-0.47187</td>
<td>1.51377</td>
<td>-0.27369</td>
<td>16.61618</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.01253</td>
<td>0.15293</td>
<td>0.47055</td>
<td>-0.23135</td>
<td>1.43085</td>
<td>-0.41217</td>
<td>1.45089</td>
<td>-0.27925</td>
<td>13.90675</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.00365</td>
<td>0.12384</td>
<td>0.47636</td>
<td>-0.25194</td>
<td>1.39700</td>
<td>-0.38491</td>
<td>1.40655</td>
<td>-0.28999</td>
<td>12.30822</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>-0.00232</td>
<td>0.10862</td>
<td>0.48033</td>
<td>-0.26634</td>
<td>1.36120</td>
<td>-0.36924</td>
<td>1.41128</td>
<td>-0.29300</td>
<td>10.67716</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>0.00346</td>
<td>0.08652</td>
<td>0.47777</td>
<td>-0.27692</td>
<td>1.33084</td>
<td>-0.35337</td>
<td>1.39472</td>
<td>-0.26988</td>
<td>8.79811</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>0.00633</td>
<td>0.07844</td>
<td>0.48158</td>
<td>-0.28087</td>
<td>1.32018</td>
<td>-0.35446</td>
<td>1.40312</td>
<td>-0.25708</td>
<td>7.95681</td>
</tr>
<tr>
<td>KF-RELS</td>
<td>100</td>
<td>0.03284</td>
<td>0.03973</td>
<td>0.49893</td>
<td>-0.20725</td>
<td>1.21083</td>
<td>-0.45434</td>
<td>1.30588</td>
<td>-0.32613</td>
<td>13.40018</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.02832</td>
<td>0.04168</td>
<td>0.49727</td>
<td>-0.27377</td>
<td>1.26652</td>
<td>-0.37328</td>
<td>1.39482</td>
<td>-0.23716</td>
<td>6.44682</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>0.00907</td>
<td>0.04078</td>
<td>0.49278</td>
<td>-0.29927</td>
<td>1.27048</td>
<td>-0.34336</td>
<td>1.42444</td>
<td>-0.20737</td>
<td>4.69370</td>
</tr>
<tr>
<td></td>
<td>1000</td>
<td>-0.00230</td>
<td>0.02711</td>
<td>0.49244</td>
<td>-0.30931</td>
<td>1.23701</td>
<td>-0.32881</td>
<td>1.43377</td>
<td>-0.19160</td>
<td>3.06655</td>
</tr>
<tr>
<td></td>
<td>2000</td>
<td>0.00198</td>
<td>0.00730</td>
<td>0.48081</td>
<td>-0.31664</td>
<td>1.21497</td>
<td>-0.31732</td>
<td>1.41960</td>
<td>-0.17097</td>
<td>2.49090</td>
</tr>
<tr>
<td></td>
<td>3000</td>
<td>0.00442</td>
<td>-0.00050</td>
<td>0.48446</td>
<td>-0.31938</td>
<td>1.20665</td>
<td>-0.31978</td>
<td>1.42942</td>
<td>-0.16938</td>
<td>2.32384</td>
</tr>
<tr>
<td>True values</td>
<td></td>
<td>0.00000</td>
<td>0.00000</td>
<td>0.45000</td>
<td>-0.30000</td>
<td>1.20000</td>
<td>-0.32000</td>
<td>1.50000</td>
<td>-0.18000</td>
<td></td>
</tr>
</tbody>
</table>

Solid line: the estimation error; dashed: the true $x_{1,k}$; dots: the estimated $\hat{x}_{1,k}$

Figure 7. The KF-RELS predicted output and true output versus $k$

Figure 8. The state $x_{1,k}$ and estimated estimate $\hat{x}_{1,k}$ versus $k$
obtained. The proposed algorithms can be extended to other system identification models, e.g., the linear state space models, and Wiener models [41,42] and Hammerstein-Wiener systems [43,44] or applied to nonlinear systems [45,46] and other fields [47,48,49].

7. ACKNOWLEDGEMENTS

This work was supported by the National Natural Science Foundation of China (No. 61273194), the Graduate Education Innovation Program of Jiangsu Province (No. KYLX16_0780) and the 111 Project (B12018).

REFERENCES


