



ELSEVIER

Contents lists available at ScienceDirect

Applied Mathematical Modelling

journal homepage: www.elsevier.com/locate/apm

Identification of multiple-input single-output Hammerstein models using Bezier curves and Bernstein polynomials

Mohamadreza Ahmadi, Hamed Mojallali*

Electrical Engineering Department, Faculty of Engineering, University of Guilan, P.O. Box 3756, Rasht, Iran

ARTICLE INFO

Article history:

Received 29 July 2010

Received in revised form 26 September 2010

Accepted 12 November 2010

Available online 18 November 2010

Keywords:

System identification

Hammerstein model

Bezier–Bernstein polynomial functions

Levenberg–Marquart algorithm

ABSTRACT

This paper considers the implementation of Bezier–Bernstein polynomials and the Levenberg–Marquart algorithm for identifying multiple-input single-output (MISO) Hammerstein models consisting of nonlinear static functions followed by a linear dynamical subsystem. The nonlinear static functions are approximated by the means of Bezier curves and Bernstein basis functions. The identification method is based on a hybrid scheme including the inverse de Casteljau algorithm, the least squares method, and the Levenberg–Marquart (LM) algorithm. Furthermore, results based on the proposed scheme are given which demonstrate substantial identification performance.

© 2010 Elsevier Inc. All rights reserved.

1. Introduction

The Hammerstein model is composed of a nonlinear static memoryless subsystem which is in series with a linear dynamic block. Thus far, the Hammerstein model has received major attention in modeling a myriad of nonlinear systems including chemical processes, DC/DC convertors, electrically stimulated muscles, actuators, RF transmitters, stretch reflexes and etc. [1–9]. As a consequence of this wide variety of applications, different identification algorithms for Hammerstein models are vastly addressed in literature. These identification methods differ mostly in the way the nonlinear static functions are represented. The identification schemes can be roughly divided into two categories: the non-parametric and the parametric. The non-parametric methods usually involve probabilistic calculations for recovering the system's unknown nonlinear properties [10,11]. In the parametric approaches, on the other hand, the nonlinear functions are defined by polynomials, neural networks, or expansions of basis functions with a finite number of parameters to be determined [2,12,13]. Nonlinear optimization algorithms can be readily applied to approximate the parameters representing these Hammerstein models. Ding et al. investigated several invaluable parametric approaches on the identification problems associated with Hammerstein models which are addressed in [12–16]. In [12], Ding and Chen proposed a mathematical foundation based on iterative and recursive least squares for estimating the parameters related to Hammerstein structures described by AR-MAX/CARMA models. Ref. [13] suggests the use of the gradient search and the Newton–Raphson method to the identification problem of Hammerstein models. Auxiliary model-based approaches to identification of Hammerstein output-error systems are also delineated in [14–16].

Despite the wealth of papers, existing formulations do not sufficiently capture the system nonlinearities. This defect mostly results from considering the nonlinear functions with a pre-defined structure such as polynomials with a known order, multi-segment piecewise linear forms, and etc. Obviously, many practical plants do not fit into these categories. Hong

* Corresponding author. Tel.: +98 131 6690276–8; fax: +98 131 6690271.

E-mail addresses: mrezaahmadi@ieee.org (M. Ahmadi), mojallali@guilan.ac.ir (H. Mojallali).

and Mitchell [17] proposed a parametric identification algorithm for single-input single-output (SISO) Hammerstein systems based on Bezier–Bernstein approximation. Albeit their approach was successful in identifying numerical examples, it suffered from several drawbacks which made it almost impossible to be used for identifying real world nonlinear processes. Firstly, the algorithm only considered the delayed versioned of the inputs to the nonlinear subsystem, whereas in most practical cases the inputs to the nonlinear blocks are not delayed. A more subtle disadvantage is that the Gauss–Newton algorithm used to estimate the nonlinear coefficients is, up to a point, slow, and inaccurate.

In the proposed scheme in this paper, the Hammerstein model is considered in a rather comprehensive mode, that is, both the delayed and un-delayed versions of the inputs appear in the system model and the SISO Hammerstein model is expanded to a multiple-input single-output (MISO) structure. Additionally, the Levenberg–Marquart (LM) algorithm is used for estimating a mixture of the nonlinear and linear parameters. The LM algorithm has been previously implemented in many nonlinear least squares problems ranging from applications in nuclear to biomedical engineering [6,18,19].

The main objective of this paper is to introduce a new scheme for identification of MISO Hammerstein models. The chief virtue of the proposed scheme is that given an input/output data set from a plant, the model can be readily identified. The strength of the proposed scheme is that the form of the nonlinear functions is considered to be unknown; thus, the identification algorithm can furnish a better understanding of the nonlinear blocks. The nonlinear gain functions in Hammerstein model are parameterized by Bezier curves, which is a linear combination of a set of Bernstein basis functions. These basis functions are fabricated over the input data by exploiting the inverse de Casteljau algorithm [20]. The remaining parameters in the model are approximated using the least squares algorithm and the Levenberg–Marquart (LM) algorithm subject to constraints.

The balance of this paper proceeds as follows. The MISO Hammerstein model is described in Section 2. Section 3 points out the proposed identification algorithm. The results established upon the proposed scheme are given in Section 4. The paper ends with conclusions in Section 5.

2. The MISO Hammerstein model

A MISO Hammerstein model is composed of a cascade of two subsystems including nonlinear gain functions Ψ^* as the nonlinear subsystem, and a dynamic linear part as the linear subsystem. Generally, the system can be modeled by

$$y(t) = - \sum_{i=1}^{n_a} a_i y(t-i) + \sum_{k=0}^{n_{b_1}} b_{k,1} \psi_1(u_1(t-k)) + \dots + \sum_{k=0}^{n_{b_m}} b_{k,m} \psi_m(u_m(t-k)) + \eta(t) \tag{1}$$

where $y(t)$ is the system output, and u_1, \dots, u_m denote the inputs. $\eta(t)$ is a Gaussian random noise with zero mean and variance of σ^2 . $\psi_i(t)$ $i = 1, \dots, m$ are the outputs of the nonlinear subsystem (or the nonlinear gain functions) and the input to the linear block. n_a and n_{b_i} , $i = 1, \dots, m$ are the input and output lags for the linear subsystem. The gain of the linear subsystem is given by

$$G = \frac{\sum_{i=1}^m \sum_{k=1}^{n_{b_i}} b_{k,i}}{1 + \sum_{j=1}^{n_a} a_j} \tag{2}$$

A simple block diagram of the MISO Hammerstein model is depicted in Fig. 1.

3. The Identification algorithm

3.1. Modeling of nonlinear gain functions Ψ^* using Bezier–Bernstein polynomial functions

The Bezier curve is a parametric curve characterized by Bernstein basis functions. With a set of pre-set two dimensional control points, the Bezier curve can be readily constructed through the de Casteljau algorithm. Previously, the inverse de Casteljau algorithm has been proposed to map the input data to $(0, 1)$, the inverse procedure of the functional mapping in the de Casteljau algorithm, through iterative error feedback [20].

The univariate Bernstein polynomial basis function $B_j^d(x)$ are the expansion of $[x + (1-x)]^d$ [21–27], defined by

$$B_j^d(x) = \binom{d}{j} x^j (1-x)^{d-j} \tag{3}$$

where j and d are nonnegative integers $j \leq d$ over the region $x \in [0, 1]$. The total number of the univariate d th order Bernstein polynomials is $d + 1$. It has been shown that Bernstein polynomials can be computed following this recursion [23,24]

$$B_j^d(x) = (1-x)B_j^{d-1}(x) + xB_{j-1}^{d-1}(x) \tag{4}$$

Ψ^* can be modeled as

$$\psi(u(t)) = \sum_{j=0}^d B_j^d(x(u(t))) \delta_j \tag{5}$$

where δ_j 's are the weights to be determined.

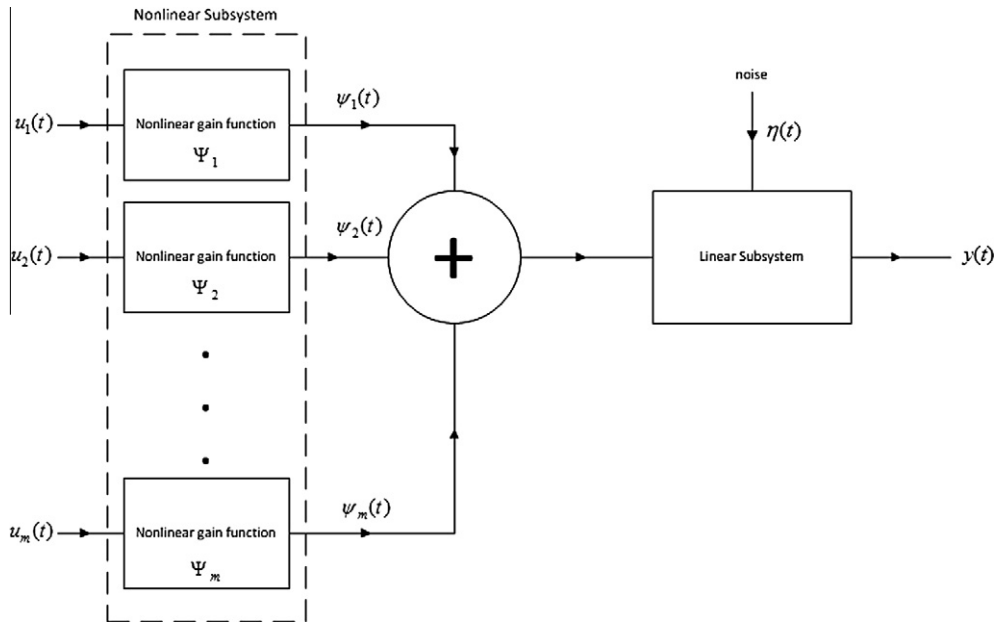


Fig. 1. The multi-input-single-output Hammerstein model.

In order to model the system, firstly the inverse de Casteljaou algorithm is applied to map each input data to $x \in [0, 1]$, so that $x(u(t))_{t=1}^N$ can be used to fabricate the Bernstein polynomial basis functions $B_j^d(x(u(t)))_{t=1}^N$. The algorithm is described as follows [14]:

Given a desired mapping point u a set of knots $U_j \in [U_0, U_d] \in [\min(u), \max(u)]$, $j = 0, \dots, d$ are preset. Denote the iteration step in the following procedure as n .

- [1] Initially set $n = 1$, and $x \in [0, 1]$ as a random number.
- [2] Calculate the corresponding first component of Bezier curve points, $U_j^{(r)}$, using the de Casteljaou's recursive formula:

$$U_j^{(r)}(x) = (1 - x)U_j^{(r-1)}(x) + xU_{j+1}^{(r-1)}(x) \tag{6}$$

until $r = d$

$$U(x^{(n)}) = (1 - x^{(n)})U_0^{(d-1)}(x^{(n)}) + x^{(n)}U_1^{(d-1)}(x^{(n)}) \tag{7}$$

- [3] The difference between the desired point u and the estimated point $\hat{u}(x^{(n)})$ is used to adjust the search direction of x . A new point is created as

$$\tilde{u}(x^{(n)}) = \hat{u}(x^{(n)}) + \gamma[u - \hat{u}(x^{(n)})] \tag{8}$$

where γ , the learning rate, is a very small positive integer ($0 < \gamma \ll 1$).

- [4] The desired solution of x at iteration step $(n + 1)$ is computed such that $\tilde{u}(x^{(n)})$ is the first order Bezier point with respect to the two end knots $U_0^{(d-1)}(x)$ and $U_1^{(d-1)}(x)$. The solution is thus given by

$$x^{(n+1)} = \frac{\tilde{u}(x^{(n)}) - U_0^{(d-1)}(x^{(n)})}{U_1^{(d-1)}(x^{(n)}) - U_0^{(d-1)}(x^{(n)})} \tag{9}$$

The procedure continues until $\|u - \tilde{u}(x^{(n)})\| \leq \varepsilon$, where ε is an arbitrary small positive number close to zero. If $\|u - \tilde{u}(x^{(n)})\| \leq \varepsilon$ set $x = x^{(n)}$. Otherwise, set $n = n + 1$ and go to step 2.

Having mapped the input data to $(0, 1)$, we can now calculate the Bernstein basis functions $B_j^d(x(u(t)))_{t=1}^N$ according to the initially chosen knots U_j , $j = 0, \dots, d$. Consequently, the unknown nonlinear function can be explained by

$$\psi_d(t) = \sum_{j=0}^d B_j^d(x(u(t)))\delta_j \tag{10}$$

This algorithm continues, until the mapping is done for all inputs to the Hammerstein model. Consequently, the Bernstein basis functions can be formed for all nonlinear gain functions. However, δ_j , $j = 0, \dots, d$ still has to be estimated. The approximated output $y(t)$ can be expressed by substituting Eq. (10) in (1).

$$\hat{F}_h = - \sum_{i=1}^{n_a} a_i y(t-i) + \sum_{k=0}^{n_{b_1}} b_{k,1} \sum_{j=0}^{d_1} \delta_{j,1} B_j^{d_1}(x(u_1(t-k))) + \dots + \sum_{k=0}^{n_{b_m}} b_{k,m} \sum_{j=0}^{d_m} \delta_{j,m} B_j^{d_m}(x(u_m(t-k))) \tag{11}$$

Denote \hat{F}_h where $\hat{F}_h(-y(t-1), \dots, -y(t-n_a), u_1(t), \dots, u_1(t-n_{b_1}), \hat{a}, \hat{b}, \hat{\delta})$ is the estimated Hammerstein model output, $\hat{a} = [a_1, \dots, a_{n_a}]^T$, $\hat{b}_{k_i} = [b_{0,i}, \dots, b_{n_{b_i},i}]^T$, and $\hat{\delta}_i = [\delta_{0,i}, \dots, \delta_{d_i,i}]^T, i = 1, \dots, m$.

A possible way to increase the approximation accuracy is to minimize the sum of squared error

$$SSE = \sum_{t=1}^N [y(t) - \hat{F}_h]^2 \tag{12}$$

It is obvious that \hat{F}_h is not linear to \hat{b} and $\hat{\delta}$ yet it can be translated as a linear regression from $y(t)$ in vector \hat{a} . The proposed parameter estimation procedure is performed in two stages: (a) based on the estimated resultant model structure using the Bezier–Bernstein polynomial functions, the least squares algorithm is applied to estimate the parameters in the autoregressive (AR) part of the linear subsystem; (b) the remaining parameters are approximated using a Levenberg–Marquart algorithm subject to the constraint of the unit gain [20].

3.2. Determining the parameters \hat{a}_j using the least squares method

Eq. (1) can be rewritten as matrixes in regression form

$$y = W\Theta + H \tag{13}$$

with

$$W = \begin{bmatrix} w(F(1)) \\ w(F(2)) \\ \dots \\ w(F(N)) \end{bmatrix} \tag{14}$$

$$w(F(t)) = [-y(t-1), \dots, -y(t-n_a), B_0^{d_1}(x(u_1(t))), \dots, B_{d_1}^{d_1}(x(u_1(t))), \dots, B_0^{d_i}(x(u_i(t-n_{b_i}))), \dots, B_{d_i}^{d_i}(x(u_i(t-n_{b_i})))] \\ \in \mathfrak{R}^{n_a+(d_1+1)(n_{b_1}+1)+\dots+(d_m+1)(n_{b_m}+1)}$$

and

$$\Theta = [\hat{a}^T, (b_0 \delta_0), \dots, (b_0 \delta_d), \dots, (b_{n_b} \delta_0), \dots, (b_{n_b} \delta_d)]^T \in \mathfrak{R}^{n_a+(d_1+1)(n_{b_1}+1)+\dots+(d_m+1)(n_{b_m}+1)}$$

$H = [\eta(1), \dots, \eta(N)]^T$ is the presumed Gaussian noise associated with the system. The least squares solution of Θ is calculated as

$$\hat{\Theta} = [W^T W]^{-1} W^T y \tag{15}$$

from (15) we can simply derive \hat{a} , which is a sub vector of $\hat{\Theta}$. At this place, a sequence $r(t)$ as a collateral model output, based on the least squares solutions of \hat{a} , and the output data is generated,

$$r(t) = y(t) + \hat{a}_1 y(t-1) + \dots + \hat{a}_n y(t-n_a) \tag{16}$$

However, the intermediate model output $r(t)$, still needs to be interpreted by the unknown parameters b and δ . The approximated version of $r(t)$ based on Bezier–Bernstein basis functions is

$$\hat{r}(t) = \sum_{k=0}^{n_{b_1}} b_{k,1} \sum_{j=0}^{d_1} \delta_{j,1} B_j^{d_1}(x(u_1(t-k))) + \dots + \sum_{k=0}^{n_{b_m}} b_{k,m} \sum_{j=0}^{d_m} \delta_{j,m} B_j^{d_m}(x(u_m(t-k))) \tag{17}$$

Proposition 1. It is postulated that W is non-singular. As a consequence, the minimization of $\sum_{t=1}^N [y(t) - \hat{F}_h]^2$ is identical to that of $\sum_{t=1}^N [r(t) - \hat{r}(t)]^2$ [17].

3.3. Applying the Levenberg–Marquart algorithm to approximate b and δ

LM algorithm is a modified version of the Gauss–Newton algorithm which converges to an optimal solution through an iterative procedure. The problem with the Gauss–Newton algorithm arose as in many practical cases in which the Jacobian matrix becomes singular, thus a valid inverse matrix could not be obtained. Since the objective functions are in the quadratic form, it is convenient to use this algorithm to find the solution. The LM algorithm introduces a new multi-functional parameter μ called the damping parameter. The LM step is defined as follows [28–30]:

$$h_{lm} = \{J^T J + \mu I\}^{-1} J^T F \quad (18)$$

where J is the Jacobian matrix, I is a unit matrix with the same size as $J^T J$, and F is the function to be minimized. Adding the damping parameter leads to some positive effects [31]:

- For all positive values of the damping parameter the coefficient matrix is positive definite, and consequently h_{lm} (the LM step) is decent direction.
- For large values of μ we have

$$h_{lm} = -\frac{1}{\mu} J^T F = -\frac{1}{\mu} F' \quad (19)$$

which is a small step in the step decent direction. A desirable feature, if the current intermediate solution is far from the answer.

- For small values of LM step, $h_{lm} = h_{gn}$ where h_{gn} denotes the Gauss–Newton step. This happens in the final steps when the algorithm is converging.

One can understand from the above consequences that the damping parameter is able to influence both the direction and the size of a step. Thus, an algorithm without a specific line search can be achieved.

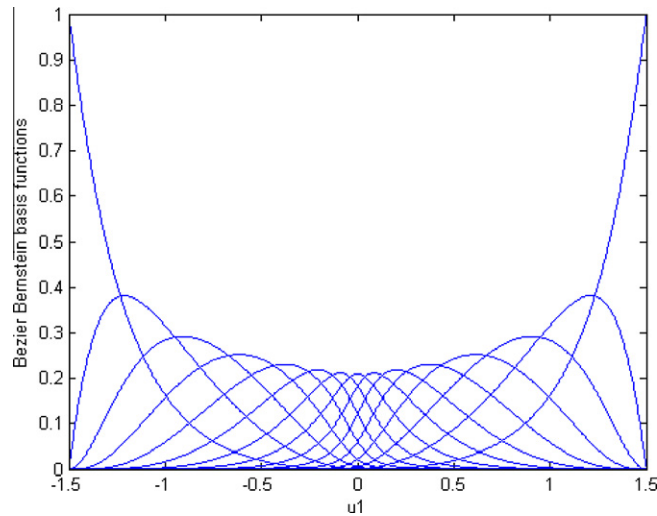


Fig. 2. Fifteen Bezier–Bernstein polynomial basis functions fabricated over the first input data u_1 .

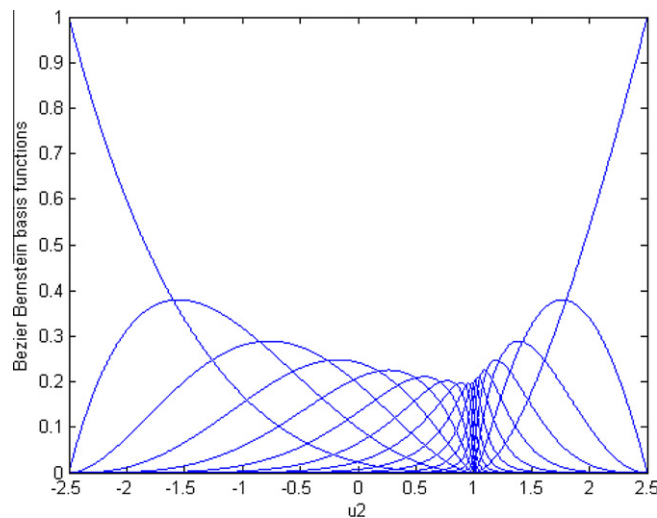


Fig. 3. Fourteen Bezier–Bernstein polynomial functions constructed from the second input data u_2 .

Another important concept in the LM algorithm is the gain ratio λ . Gain ratio is also commonly used in trust region algorithms such as Powell’s dog leg method [30,31]. The gain ratio is signified as the ratio between the actual and predicted decrease in function value, and it can be computed using the following equation:

$$\lambda = \frac{F(x) - F(x + h)}{L(0) - L(h)} \tag{20}$$

where the denominator is the gain predicted by the linear model. In the LM algorithm, the damping parameter is increased as the gain ratio becomes small, since $L(h)$ is a poor estimation of $F(x + h)$. On the other hand, a large value of the gain ratio indicates a good approximation; therefore, the damping parameter is decreased. The gain predicted by the linear model is calculated as [30]:

$$L(0) - L(h_{lm}) = -h_{lm}^T J^T F(x) - 0.5 h_{lm}^T J^T J h_{lm} = -0.5 h_{lm}^T (2J^T F(x) - (J^T J + \mu I - \mu I) h_{lm}) = 0.5 h_{lm}^T (\mu h_{lm} - J^T F(x)) \tag{21}$$

One should note that both $-h_{lm}^T J^T F(x)$ and $h_{lm}^T h_{lm}$ are positive; thus, $L(0) - L(h_{lm})$ is also positive and nonzero. A comprehensive discussion of the LM algorithm can be found in Ref. [30]. Note that the constraint of the unit gain resulting from the special structure of the Hammerstein model should be considered so an enhanced algorithm could be achieved. In other words, the algorithm would avoid being trapped in local minima [14,31]. Using Eqs. (16), (17) and proposition 1, the model residual is represented as $e(\hat{b}, \hat{\delta}, t) = r(t) - \hat{r}(t)$.

The Jacobian matrix J with respect to $[b^T, \delta^T]^T$ is given by

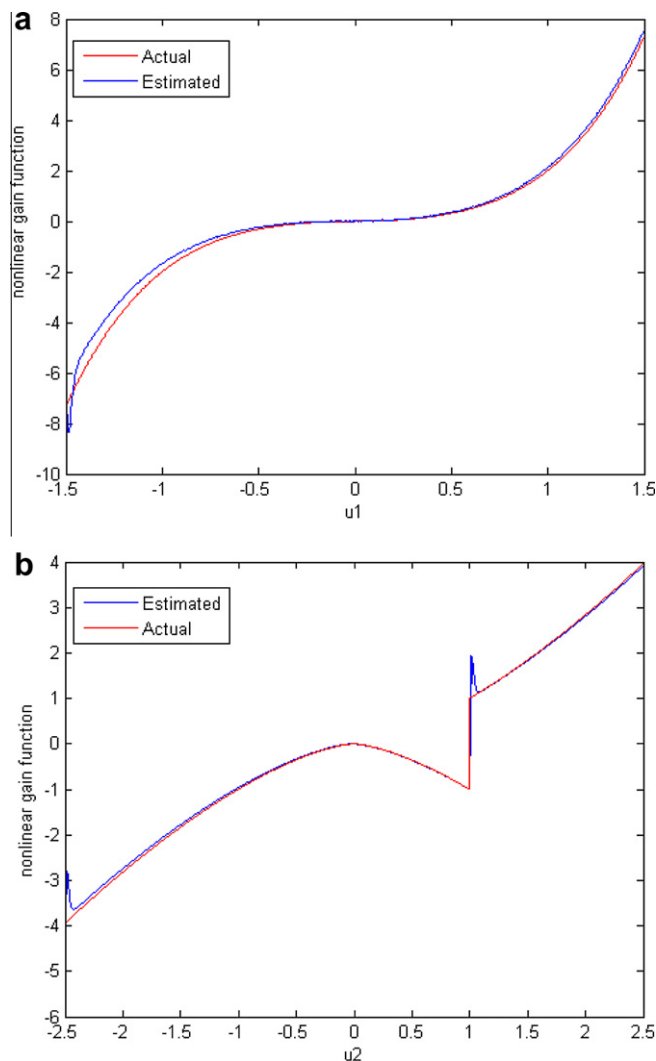


Fig. 4. The estimated (blue line) and the actual (red line) nonlinear gain functions associated with the first (a), and the second (b) input in low noise condition. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$J_i = \begin{bmatrix} \frac{\partial}{\partial b_{0i}} e(\hat{b}, \hat{\delta}, 1) & \cdots & \frac{\partial}{\partial b_{nb_i}} e(\hat{b}, \hat{\delta}, 1) & \frac{\partial}{\partial \delta_{0i}} e(\hat{b}, \hat{\delta}, 1) & \cdots & \frac{\partial}{\partial \delta_{di}} e(\hat{b}, \hat{\delta}, 1) \\ \frac{\partial}{\partial b_{0i}} e(\hat{b}, \hat{\delta}, 2) & \cdots & \frac{\partial}{\partial b_{nb_i}} e(\hat{b}, \hat{\delta}, 2) & \frac{\partial}{\partial \delta_{0i}} e(\hat{b}, \hat{\delta}, 2) & \cdots & \frac{\partial}{\partial \delta_{di}} e(\hat{b}, \hat{\delta}, 2) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial}{\partial b_{0i}} e(\hat{b}, \hat{\delta}, N) & \cdots & \frac{\partial}{\partial b_{nb_i}} e(\hat{b}, \hat{\delta}, N) & \frac{\partial}{\partial \delta_{0i}} e(\hat{b}, \hat{\delta}, N) & \cdots & \frac{\partial}{\partial \delta_{di}} e(\hat{b}, \hat{\delta}, N) \end{bmatrix}$$

$$J = [J_1 J_2 \cdots J_m] \tag{22}$$

where

$$\frac{\partial}{\partial b_i} e(\hat{b}, \hat{\delta}, t) = \sum_{j=0}^d B_j^d(x(u(t-i))) \hat{\delta}_j, \quad j = 0, 1, \dots, n_b \tag{23}$$

$$\frac{\partial}{\partial \delta_j} e(\hat{b}, \hat{\delta}, t) = \sum_{i=0}^{n_b} B_i^d(x(u(t-i))) \hat{b}_i, \quad j = 0, \dots, d$$

The algorithm is as follows:

- (I) Set $n = 0$, $v = \alpha$, $\mu = \beta$, \hat{b} , and $\hat{\delta}$ are generated as random vectors with appropriate dimensions.
- (II) Apply the Levenberg–Marquart algorithm subject to the normalization constraint in order to maintain the gain of linear subsystem as one.

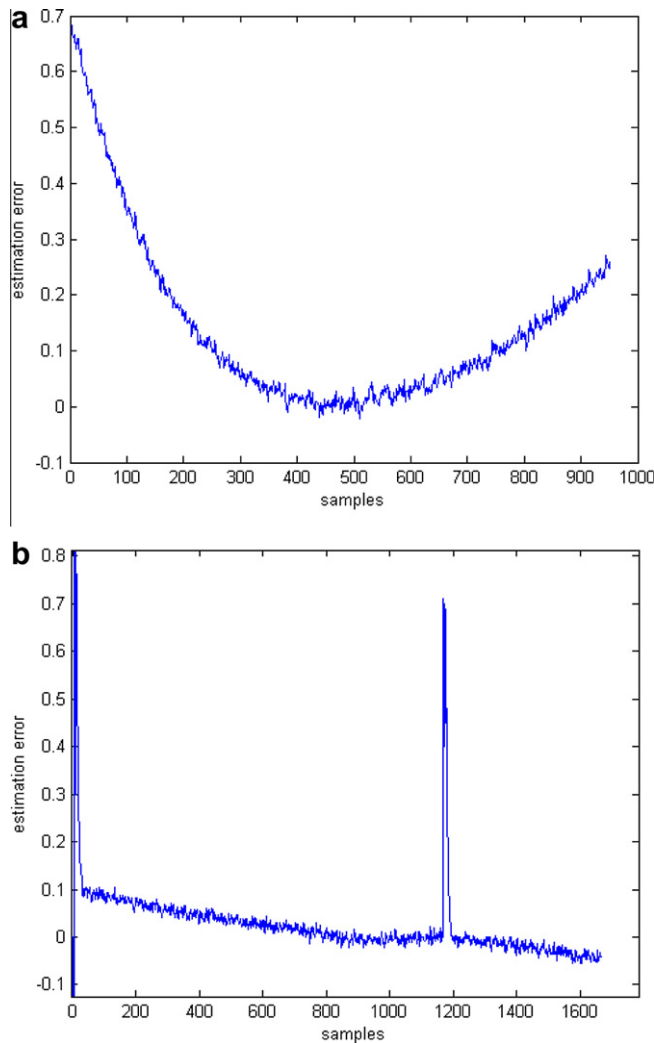


Fig. 5. Corresponding estimation errors regarding the first (a), and the second (b) nonlinear functions.

(a) LM step

$$A^{(n)} = (J^{(n)})^T J^{(n)}, \quad h_{lm} = \{A^{(n)} + \mu^{(n)} I\}^{-1} (J^{(n)})^T e$$

$$\begin{bmatrix} \hat{\mathbf{b}}^{(n+1)} \\ \hat{\delta}^{(n+1)} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{b}}^{(n)} \\ \hat{\delta}^{(n)} \end{bmatrix} + h_{lm} \quad (24)$$

where

$$e = [e(\hat{\mathbf{b}}, \hat{\delta}, 1), \dots, e(\hat{\mathbf{b}}, \hat{\delta}, N)]^T$$

(b) Apply the constraint of the unit gain (parameter normalization). Calculate G the estimated gain of the linear subsystem

$$\hat{G}^{(n+1)} = \frac{\sum_{i=1}^m \sum_{k=1}^{n_{b_i}} \hat{b}_{k,i}^{(n+1)}}{1 + \sum_{j=1}^{n_a} a_j} \quad (25)$$

then

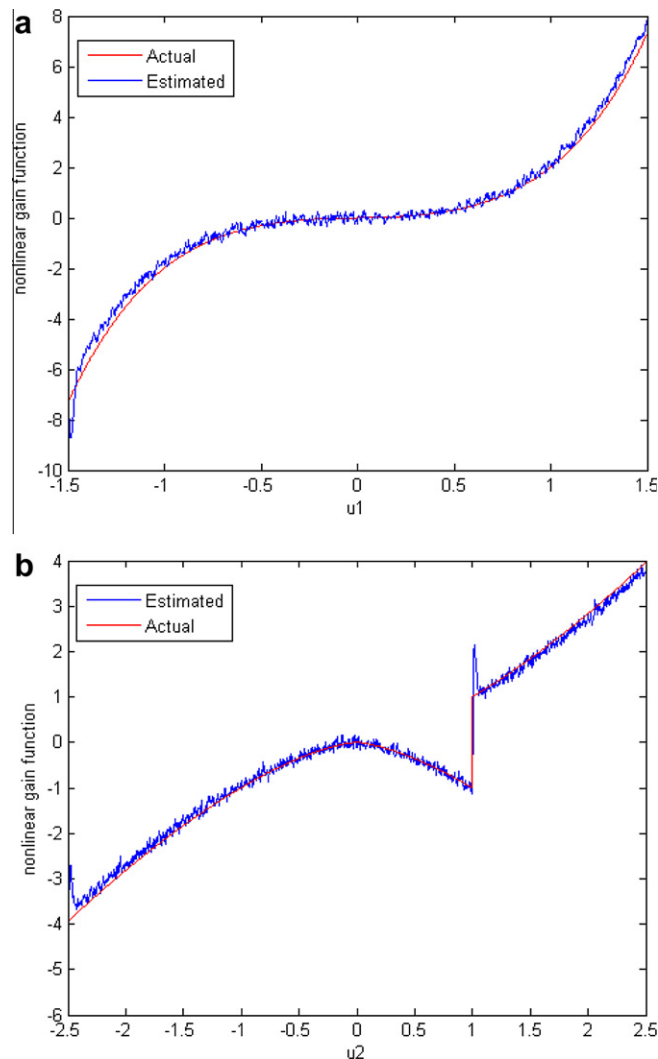


Fig. 6. The estimated (blue line) and the actual (red line) nonlinear gain functions associated with the first (a), and the second (b) inputs in high noise condition. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

$$\begin{aligned} \hat{b}^{(n+1)} &\leftarrow \hat{b}^{(n+1)} / \hat{G}^{(n+1)} \\ \hat{\delta}^{(n+1)} &\leftarrow \hat{G}^{(n+1)} \hat{\delta}^{(n+1)} \end{aligned} \tag{26}$$

Subsequently, calculate

$$\lambda = \frac{\|e^{(n)}\|^2 - \|e^{(n+1)}\|^2}{0.5h_{lm}^T(\mu^{(n)}h_{lm} - J^T e^{(n)})} \tag{27}$$

(b) If $\lambda > 0$ meaning the solution matrix has a better estimation, step is accepted.

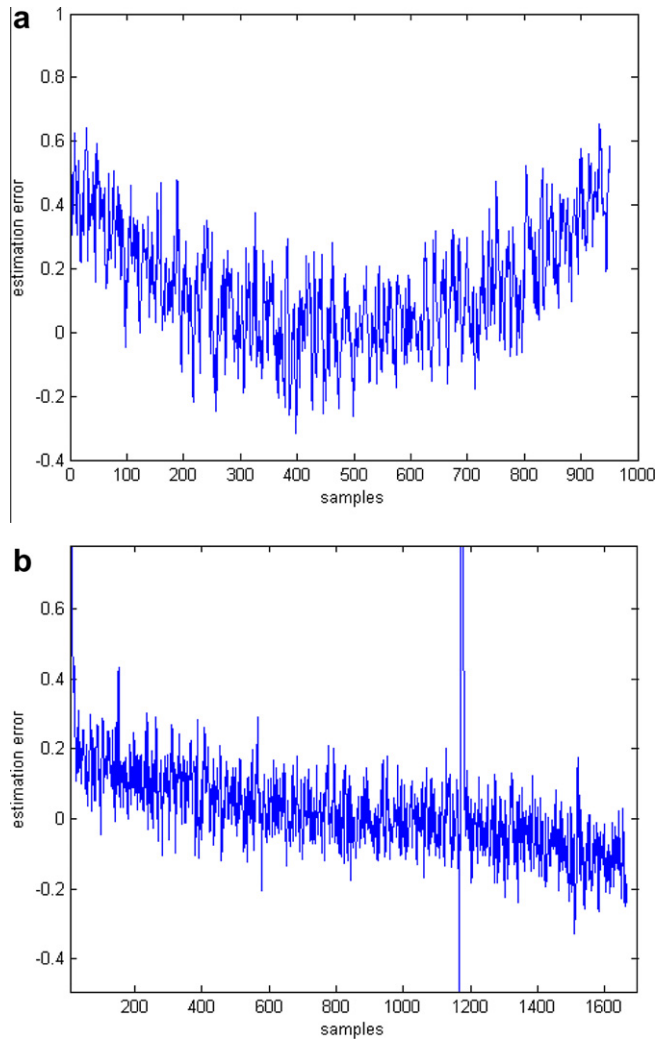


Fig. 7. Corresponding estimation errors for the first (a), and the second (b) nonlinear functions with noisy measurements.

Table 1

The approximated values of the parameters related to the Hammerstein system adopted in example 1.

Optimization method	Noise level	a_1	a_2	$b_{0,1}$	$b_{1,1}$	$b_{0,2}$	$b_{1,2}$	$b_{2,2}$
Gauss–Newton	$\sigma = 0.01$	-1.1853	0.8837	0.2888	-0.4888	0.4211	-0.5112	1.0210
	$\sigma = 0.1$	-1.1308	0.8568	0.2157	-0.4658	0.3602	-0.5523	0.9547
Levenberg–Marquart	$\sigma = 0.01$	-1.1997	0.9011	0.2994	-0.4898	0.3908	-0.4899	1.0000
	$\sigma = 0.1$	-1.1858	0.8903	0.2782	-0.44713	0.3875	-0.5203	1.1021
True values		-1.2	0.9	0.3	-0.5	0.4	-0.5	1

Then,

$$X = \begin{pmatrix} \hat{b}^{(n+1)} \\ \hat{\delta}^{(n+1)} \end{pmatrix}, A^{(n+1)} = (J^{(n+1)})^T J^{(n+1)},$$

where X denotes the solution matrix.

Afterwards, update the damping parameter

$$\mu^{(n+1)} = \mu^{(n)} \max(1/3, 1 - (2\lambda - 1)^3); v = \alpha$$

and $e^{(n+1)} = e^{(n)}$. The loop stops as $\|\sum_{t=1}^N e^2(t)\| < \xi$ where $0 < \xi \ll 1$.

If λ is not positive μ is updated (step is not accepted) $\mu = \mu * v$ and $v = \alpha * v$.
Go to step II.

4. Illustrative examples and discussion

4.1. Example 1

A simulated MISO Hammerstein model is generated with the following equation

$$y(t) = 1.2y(t - 1) - 0.9y(t - 2) + 0.3\psi_1(u_1(t)) - 0.5\psi_1(u_1(t - 1)) + 0.4\psi_2(u_2(t)) - 0.5\psi_2(u_2(t - 1)) + \psi_2(u_2(t - 2)) + \eta(t) \tag{28}$$

Where

$$\Psi_1(u) = u^2(u^2 + 1)\text{sign}(u) \tag{29}$$

$$\Psi_2(u) = |u|\sqrt{|u|}\text{sign}(u - 1) \tag{30}$$

and $\eta(t) \in N(0, \sigma^2)$ in which $\sigma^2 = 0.0001$. Based on the Eqs. (28)–(30), input/output data samples were generated. The inputs u_1 and u_2 are uniformly distributed in the ranges $[-1.5, 1.5]$ and $[-2.5, 2.5]$, respectively. A set of 15 knots for the first input u_1 were selected as

$$u_1 = [-1.5, -1.2, -1, -0.5, -0.2, -0.1, -0.02, 0, 0.02, 0.1, 0.2, 0.5, 1, 1.2, 1.5]$$

Consequently, the polynomial degree of the Bernstein basis functions was set as $d = 14$. Afterward, the input data is fed into the inverse de Casteljaeu algorithm with learning rate $\gamma = 0.01$ in order to perform the functional mapping into $x \in (0, 1)$, a data range suitable for the Bernstein basis functions to be formed using Eqs. (3), (4). Therefore, a sequence of regressors $B_j^d(x(u(t))), j = 0, 1, \dots, 14$ were generated. The resultant 15 Bezier–Bernstein polynomial functions are shown in Fig. 2. Similarly, 17 knots were preset for the second input u_2 as follows:

$$u_2 = [-2.5, -1.5, -0.5, 0, 0.5, 0.8, 0.97, 0.98, 0.999, 1, 1.001, 1.002, 1.01, 1.1, 1.2, 1.5, 2.5]$$

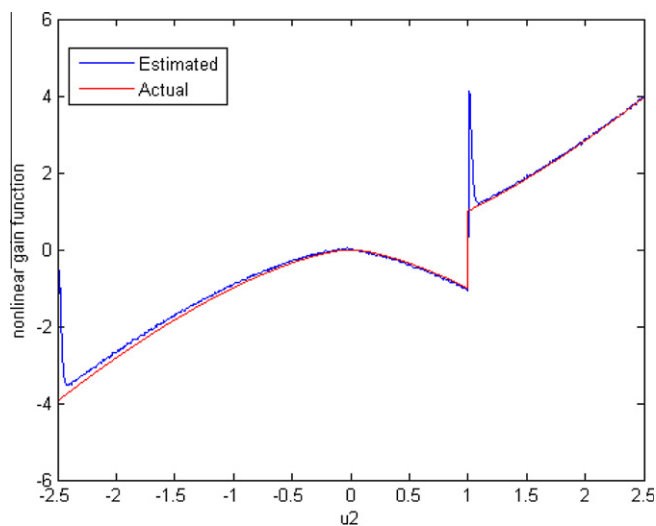


Fig. 8. Deterioration of estimation accuracy as observed from a comparison between the estimated (blue line) and the actual (red line) nonlinear gain function regarding the second input as a result of reducing knots. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The constructed basis functions are illustrated in Fig. 3. It is important to note that these functions are quite dense where most of the nonlinearity exists (here, the most nonlinear span is in the neighborhood around $u_2 = 1$). Implementing the least squares algorithm discussed in Section 3.2, the linear coefficients $a_j, j = 1, 2, \dots, n_a$ are computed. The model auxiliary output $r(t)$ is formed on the basis of the estimated values of a . Henceforth, the task of the identification algorithm is to minimize $\sum_{t=1}^N e^2(t)$ through the LM algorithm, that is, $\chi^* = \arg \left[\min \left\{ \sum_{t=1}^N e^2(t), \forall \hat{\chi} \right\} \right]$, subject to $G = 1$ where $\chi = [a^T, b^T, \delta^T]^T$.

It is important to note that α should be chosen quite carefully. In conventional nonlinear least squares problems solved using the LM method, the value of α is chosen as 2 [30]. However, this choice of α , fails to appropriately approximate the nonlinear parameters in the proposed identification algorithm in this paper due to the number of parameters to be tuned and the abundance of local minimums. A large value of α results in a fast convergence, the algorithm would get trapped in local minima though. For very small values of α , the LM algorithm loses its convergence speed, and, up to a point, its accuracy. The most suitable value of α was obtained empirically as 1.01. This would guarantee less iteration cycles, and thus a relatively fast convergence. Furthermore, this would lead to a more precise estimation of the parameters.

With the above considerations in mind, The LM algorithm as discussed in Section 3.3 is applied. After 88 iteration cycles the value of the sum of squared errors was achieved as 0.0028. The estimated nonlinear gain functions and the corresponding estimation errors are depicted in Figs. 4 and 5. Moreover, the same simulation was performed based on a rather noisy data basis, i.e. $\sigma^2 = 0.01$. The results are displayed in Figs. 6 and 7 which demonstrates the algorithm's capability to estimate the nonlinear functions with a high precision, in spite of the lack of accurate output measurements.

For the purpose of comparison, the estimated parameters associated with the Hammerstein system under both noise levels by incorporating both the Gauss–Newton and the Levenberg–Marquart method are outlined in Table 1.

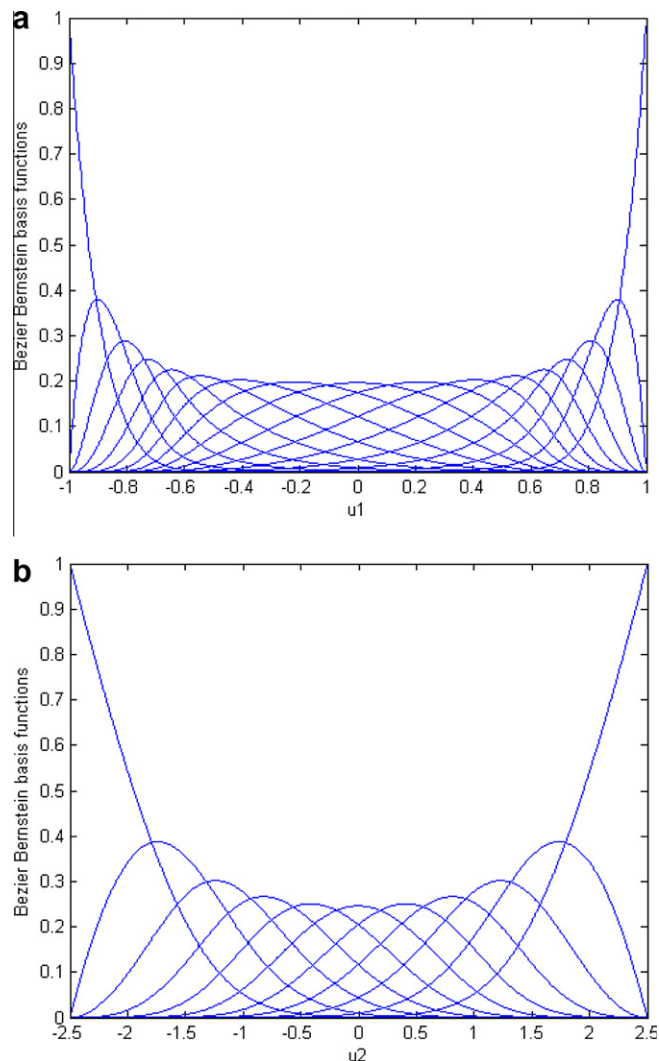


Fig. 9. Bezier–Bernstein basis functions formed based on the data from the first (a) and the second (b) input regarding example 2.

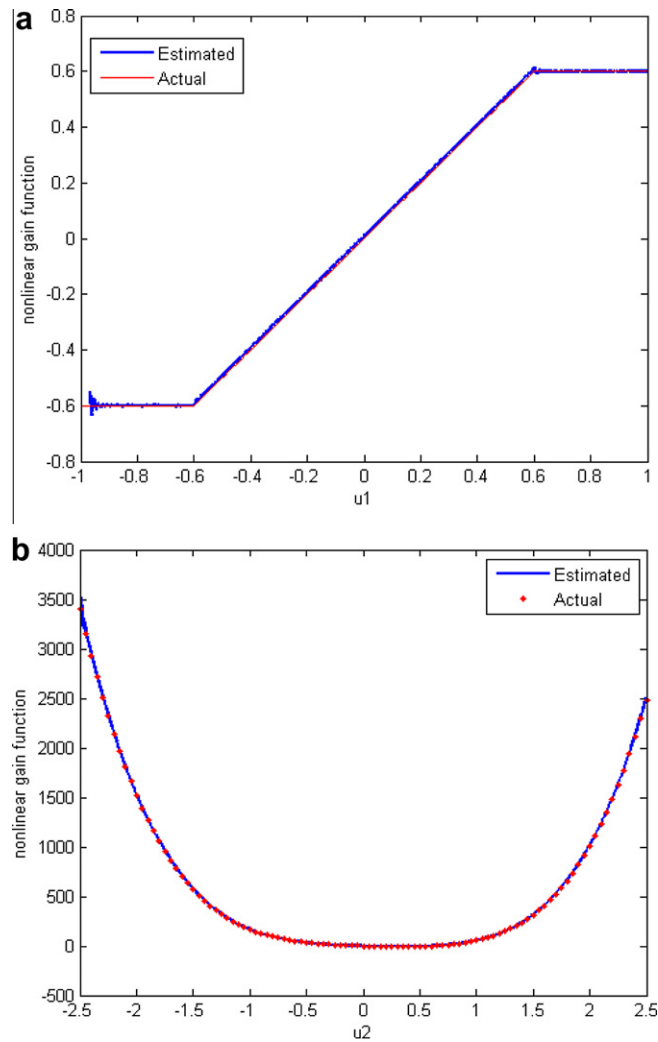


Fig. 10. The estimated and the actual nonlinear gain functions associated with the first (a), and the second (b) inputs regarding example 2.

One has to consider the effect of the initiating knots on the overall algorithm’s convergence. Fewer knots would lead to a poor estimation of the nonlinear functions. In order to illustrate this property, a set of 10 knots as given below are replaced with the previous 17 knots used to approximate Ψ_2

$$u_2 = [-2.5, -1.5, 0, 0.5, 0.8, 0.97, 1, 1.01, 1.1, 2.5]$$

The approximated Ψ_2 function is shown in Fig. 8. On the contrary, more knots would contribute to a more proper estimation, while deteriorating the computational simplicity of the overall algorithm, and elongating the convergence time. Thus, a trade-off should be made between the convergence speed and the estimation preciseness. Furthermore as mentioned earlier in this section, it is worth noting that the initiating knots should be selected closer to each other in the most nonlinear regions.

4.2. Example 2

Suppose

$$y(t) = 1.608y(t - 1) - 0.6385y(t - 2) + 0.3\psi_1(u_1(t)) + 0.207\psi_2(u_2(t - 1)) - 0.1764\psi_2(u_2(t - 2)) + \eta(t) \tag{31}$$

$$\Psi_1(u) = \text{sat}(u, -0.6, 0.6) \tag{32}$$

$$\Psi_2(u) = -31.549u + 41.732u^2 - 24.201u^3 + 68.634u^4 \tag{33}$$

where the function $\text{sat}(x, x_-, x_+)$ is the saturation function with respective left and right breaking points in x_- and x_+ . $\Psi_2(u)$ is the static nonlinear function regarding to a steam-water heat exchanger system described by Eq. (31) as studied in [32]. Note that the term consisting ψ_1 is added to the system model in order to obtain a MISO structure. The Gaussian noise $\eta(t) \in N(0, 0.1)$ is added to the Hammerstein model to account for output measurement errors, as well. The inputs u_1 and u_2 are limited to the ranges $[-1, 1]$ and $[-2.5, 2.5]$, respectively. Five hundred samples of input/output data were generated using Eqs. (31)–(33). In order to approximate system nonlinearities, a set of 16 and 10 control points (knots) were preset as given below

$$u_1 = [-1, -0.9, -0.8, -0.69, -0.65, -0.6, -0.55, -0.5, 0, 0.5, 0.55, 0.6, 0.65, 0.69, 0.8, 0.9, 1]$$

$$u_2 = [-2.5, -1.5, -1.2, -0.8, -0.5, 0, 0.5, 0.8, 1.2, 1.5, 2.5]$$

The proposed identification algorithm based on Bezier–Bernstein approximation as discussed earlier was applied. The Bezier–Bernstein basis functions formed by exploiting the data from the first and the second input are illustrated in Fig. 9. The approximated static nonlinearities are depicted in Fig. 10 and the corresponding estimation errors are shown in Fig. 11. Additionally, the approximated parameters are presented in Table 2.

Having reviewed the simulation results provided in this section, one can realize that the implementation of Bezier–Bernstein approximation and the LM algorithm brings about higher modeling and identification capabilities.

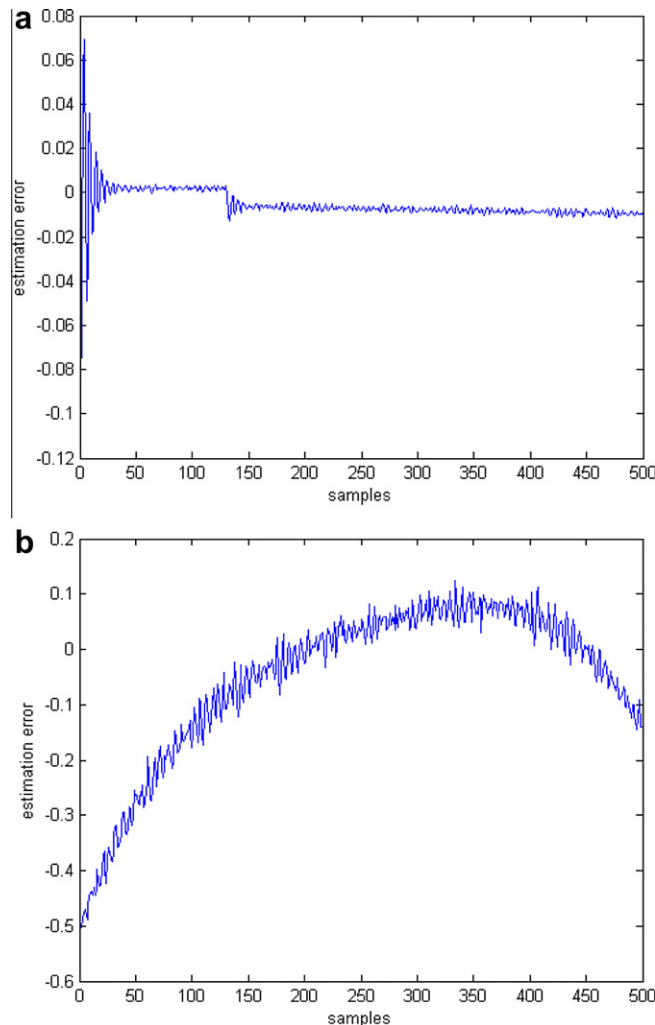


Fig. 11. Corresponding estimation errors regarding the first (a), and the second (b) nonlinear functions in example 2.

Table 2

The approximated values of the parameters related to the Hammerstein system adopted in Example 2.

Optimization method	Noise level	a_1	a_2	$b_{0,1}$	$b_{0,2}$	$b_{1,2}$	$b_{2,2}$
Gauss–Newton	$\sigma = 0.1$	–1.2095	0.8031	0.2459	$1.21e-4$	0.4817	–0.8005
Levenberg–Marquart	$\sigma = 0.1$	–1.5867	0.6214	0.2789	0.0000	0.2153	–0.1609
True values		–1.608	0.6385	0.3	0	0.207	–0.1764

5. Conclusions

A new straight forward identification algorithm for the MISO Hammerstein model is outlined in this paper. Bezier–Bernstein basis functions were implemented to approximate the system nonlinearities. The parameters in the Hammerstein model are estimated using the least squares and the LM algorithm subject to constraints. Lastly, the results based on the proposed scheme are given which demonstrate superior estimation performance.

References

- [1] R.K. Pearson, M. Pottman, Gray-box identification of block-oriented nonlinear models, *J. Process Control* 10 (4) (2000) 301–315.
- [2] H.T. Su, T.J. McAvoy, Integration of multilayer perception neural networks and linear dynamic models: a Hammerstein modeling approach, *Ind. Eng. Chem. Res.* 26 (1993) 1927–1936.
- [3] A. Balestrino, A. Landi, M. Ould-Zmirli, L. Sani, Automatic nonlinear auto-tuning method for Hammerstein modeling of electrical drives, *IEEE Trans. Ind. Electron.* 48 (3) (2001) 645–655.
- [4] D.K. Rollins, N. Bhandari, A.M. Bassily, G.M. Colver, S.T. Chin, A continuous-time nonlinear dynamic predictive modeling method for Hammerstein processes, *Ind. Eng. Chem. Res.* 42 (4) (2003) 860–872.
- [5] J.G. Smith, S. Kamat, K.P. Madhavan, Modelling of PH process using wavenet based Hammerstein model, *J. Process Control* 17 (2007) 551–561.
- [6] E.G. Dempsey, D.T. Westwick, Identification of Hammerstein models with cubic spline nonlinearities, *IEEE Trans. Biomed. Eng.* 51 (2) (2004) 237–245.
- [7] D.T. Westwick, R.E. Kearney, Identification of a Hammerstein model of the Stretch reflex EMG using separable Least Squares, *Proc. 22nd EMBS Int Conf.* (2000) 1901–1904.
- [8] K.J. Hunt, M. Muni, N.N. Donaldson, F.M.D. Barr, Investigation of Hammerstein hypothesis in the modelling of electrically stimulated muscle, *IEEE Trans. Biomed. Eng.* 45 (8) (1998) 998–1009.
- [9] F. Alonge, F. D'Ippolito, F.M. Raimondi, S. Tumminaro, Nonlinear modelling of DC/DC converters using the Hammerstein's approach, *IEEE Trans. Power Electronics* 22 (4) (2007) 1210–1221.
- [10] Z.Q. Lang, A nonparametric polynomial identification algorithm for the Hammerstein system, *IEEE Trans. Autom. Control* 42 (1997) 1435–1441.
- [11] W. Greblicki, Stochastic approximation in nonparametric identification of Hammerstein systems, *IEEE Trans. Autom. Control* 47 (11) (2002) 1800–1810.
- [12] F. Ding, T. Chen, Identification of Hammerstein nonlinear ARMAX systems, *Automatica* 41 (2005) 1479–1489.
- [13] F. Ding, X.P. Liu, G. Liu, Identification methods for Hammerstein nonlinear systems, *Digit. Signal Process.* (2010). 10.1016/j.dsp.2010.06.006.
- [14] F. Ding, Y. Shi, T. Chen, Auxiliary model-based least-squares identification methods for Hammerstein output-error systems, *Syst. Control Lett.* 56 (2007) 373–380.
- [15] D. Wang, Y. Chu, F. Ding, Auxiliary model-based RELS and MI-ELS algorithms for Hammerstein OEMA systems, *Comput. Math. Appl.* 59 (9) (2010) 3092–3098.
- [16] D. Wang, Y. Chu, G. Yang, F. Ding, Auxiliary model-based recursive generalized least squares parameter estimation for Hammerstein OEAR systems, *Math. Comput. Model.* 52 (1–2) (2010) 309–317.
- [17] X. Hong, R.J. Mitchell, Hammerstein model identification algorithm using Bezier–Bernstein approximation, *IET Control Theory Appl.* 1 (4) (2007) 1149–1159.
- [18] S. Kwon, M. Lynch, M. Prokop, Decoupling PI controller design for a normal conducting RF cavity using a recursive Levenberg–Marquart algorithm, *IEEE Trans. Nucl. Sci.* 52 (1) (2005) 440–449.
- [19] M. Wan, W. Zhang, J. Dang, Y. Yang, A novel cutting force modelling method for cylindrical end mill, *Appl. Math. Model.* 34 (2010) 823–836.
- [20] X. Hong, C.J. Harris, Generalized neurofuzzy network modeling algorithms using Bezier–Bernstein polynomial functions and additive decomposition, *IEEE Trans. Neural Networks* 11 (2000) 889–902.
- [21] Y. Shin, Modified Bernstein polynomial and their connectionist interpretation, *Proc. IEEE World Congress on Computational Intelligence* 3 (1994) 1433–1438.
- [22] A. Pilat, Analytical modeling of active magnetic bearing geometry, *Appl. Math. Model.* 34 (2010) 3805–3816.
- [23] G. Farin, *Curves and Surfaces for Computer-Aided Design: A Practical Guide*, Academic, Boston, MA, 1994.
- [24] T. Pavlidis, *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, MD, 1982.
- [25] Magnus B. Egerstedt, Clyde F. Martin, A note on the connection between Bezier curves and linear optimal control, *IEEE Trans. Autom. Control* 49 (10) (2004) 1728–1731.
- [26] K. Narooi, A.K. Taheri, A new model for prediction the strain field and extrusion pressure in ECAE process of circular cross section, *Appl. Math. Model.* 34 (2010) 1901–1917.
- [27] M. Zettler, J. Garloff, Robustness analysis of polynomial parameter dependency using Bernstein expansion, *IEEE Trans. Autom. Control* 43 (3) (1998) 425–431.
- [28] K. Levenberg, A method for the solution of certain non-linear problems in least squares, *Q. Appl. Math.* 2 (2) (1944) 164–168.
- [29] D. Marquardt, An algorithm for the least-squares estimation of nonlinear parameters, *SIAM J. Appl. Math.* 11 (2) (1963) 431–441.
- [30] K. Madsen, H. Nielsen, O. Tingleff, *Methods for non-linear least squares problems*. Technical University of Denmark, 2004. Lecture notes. Available from: <<http://www.imm.dtu.dk/courses/02611/nllsq.pdf>>.
- [31] M.J.D. Powell, Problems related to unconstrained optimization, in: W. Murray (Ed.), *Numerical Methods for Unconstrained Optimisation*, Academic Press, London & New York, 1972, pp. 29–55.
- [32] E. Eskinat, S.H. Johnson, W.L. Luyben, Use of Hammerstein models in identification of nonlinear systems, *Am. Inst. Chem. Eng. J.* 37 (1991) 255–268.