

Modelling and control of Hammerstein system using B-spline approximation and the inverse of De Boor algorithm

X. Hong^{a*}, R.J. Mitchell^a and S. Chen^b

^aSchool of Systems Engineering, University of Reading, Reading RG6 6AY, UK; ^bSchool of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

(Received 2 June 2010; final version received 23 January 2011)

In this article a simple and effective controller design is introduced for the Hammerstein systems that are identified based on observational input/output data. The nonlinear static function in the Hammerstein system is modelled using a B-spline neural network. The controller is composed by computing the inverse of the B-spline approximated nonlinear static function, and a linear pole assignment controller. The contribution of this article is the inverse of De Boor algorithm that computes the inverse efficiently. Mathematical analysis is provided to prove the convergence of the proposed algorithm. Numerical examples are utilised to demonstrate the efficacy of the proposed approach.

Keywords: B-spline; De Boor algorithm; Hammerstein model; pole assignment controller; system identification

1. Introduction

The Hammerstein model, comprising a nonlinear static functional transformation followed by a linear dynamical model, has been applied to nonlinear plant/process modelling in a wide range of biological/engineering problems (Hunter and Korenberg 1986; Balestrino, Landi, Ould-Zmirli and Sani 2001; Bloemen, van den Boom, and Verbruggen 2001; Turunen, Tanttu, and Loula 2003). For example, it is a suitable model for signal processing applications involving any nonlinear distortion followed by a linear filter, the modelling of the human heart in order to regulate the heart rate during treadmill exercises (Su 2007), and the modelling of hydraulic actuator friction dynamics (Kwak, Yagle, and Levitt 1998). The Hammerstein model has been widely researched (Billings and Fakhouri 1979; Stoica and Söderström 1982; Greblicki and Pawlak 1986; Greblicki 1989, 2002; Verhaegen and Westwick 1996; Lang 1997; Bai and Fu 2002; Chen 2004; Chaoui, Giri, Rochdi, Haloua, and Naitali 2005). The model characterisation/representation of the unknown nonlinear static function is fundamental to the identification of Hammerstein model. Various approaches have been developed in order to capture the *a priori* unknown nonlinearity by use of both parametric (Verhaegen and Westwick 1996; Chaoui et al. 2005) and nonparametric methods (Lang 1997; Greblicki 2002; Chen 2004). It has been shown that the Bernstein basis is the best conditioned and the most stable among all other

polynomial basis (Farouki and Goodman 1996). The inverse of de Casteljau's algorithm was introduced to identify the Bezier–Bernstein neural network using the Bernstein approximation and from observational data (Hong and Harris 2000). Recently a new identification algorithm (Hong and Mitchell 2007) for the Hammerstein model has been introduced based on the Bezier–Bernstein approximation and the inverse of de Casteljau's algorithm. Alternatively, the special structure of Hammerstein models can be exploited to develop hybrid parameter estimation algorithms (Bai 1998; Bai and Fu 2002; Chaoui et al. 2005). Similar to Bezier curve, the B-spline curve has also been widely used in computer graphics and computer aided geometric design (CAGD; Farin 1994). B-spline curves consist of many polynomial pieces, offering much more versatility than do Bezier curves while maintaining the same advantage of the best conditioning property. The early work on the construction of B-spline curve is mathematically involved and numerically unstable (De Boor 1978). De Boor algorithm uses recurrence relations and is numerically stable (De Boor 1978). The B-spline basis functions for nonlinear systems modelling have been widely applied (Kavli 1993; Brown and Harris 1994; Harris, Hong and Gan 2002).

Model based control for the Hammerstein system has been well studied (Anbumani, Patnaik, and Sarma 1981; Bloemen, van den Boom, and Verbruggen 2000;

*Corresponding author. Email: x.hong@reading.ac.uk

Bloemen et al. 2001). A popular treatment of handling the Hammerstein model is to remove the nonlinearity via an inversion (Fruzzetti, Palazoglu, and McDonald 1997; Patwardhan, Lakshminarayanan, and Shah 1998; Bloemen et al. 2000). This enables the celebrated self tuning control methods to be readily applicable (Astrom and Wittenmark 1989). The implementation of model based control for *a priori* unknown Hammerstein model requires system identification including the modelling and identification of the nonlinear static function. Different nonlinear model representations result in variations in controller design algorithms. For example, in Anbumani et al. (1981) and Zhu, Warwick, and Douce (1991) the nonlinear static function is based on an explicit polynomial function of the input. The optimal control law satisfies a polynomial equation of the input, which is then found via root solving. In Hong and Mitchell (2006), the closed-loop system is linearised by inserting the inverse of the identified static nonlinearity, and the nonlinear subsystems' inverse is calculated using the inverse of de Casteljau's algorithm.

Computationally efficient and numerically stable algorithms are in general desirable in nonlinear system identification and control. In this article the control for Hammerstein system as characterised by the B-spline polynomial based model is investigated through the removal of the nonlinearity via the inversion of the B-spline neural network, and the application of a pole assignment controller. Note that for the system identification of the resultant model representation, Bai's overparameterisation approach is directly applicable (Bai and Fu 2002). In this article we used the Gauss–Newton algorithm subject to constraints as proposed in Hong and Mitchell (2007). The linearisation of the closed-loop system is achieved by inserting the inverse of the identified static nonlinearity via the inverse of De Boor algorithm. In spite of being an iterative numerical one, the proposed algorithm is shown to have a very low computational cost. In addition, the inverse of De Boor algorithm is numerically stable as it builds upon De Boor algorithm (De Boor 1978). Convergence analysis based on the Lyapunov method is included.

2. Modelling of the Hammerstein system based on B-spline functions

2.1. The Hammerstein system

The Hammerstein system, as shown in Figure 1, consists of a cascade of two subsystems, a nonlinear memoryless function $\Psi(\bullet)$ as the first subsystem, followed by a linear dynamic part as the second subsystem.

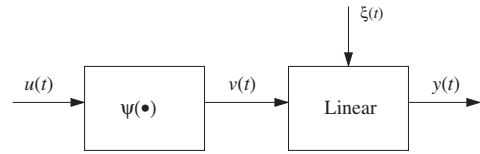


Figure 1. The Hammerstein system.

The system can be represented by

$$y(t) = \hat{y}(t) + \xi(t) \\ = -a_1y(t-1) - a_2y(t-2) - \dots - a_{n_a}y(t-n_a) \\ + b_1v(t-1) + \dots + b_{n_b}v(t-n_b) + \xi(t), \quad (1)$$

$$v(t-j) = \Psi(u(t-j)), \quad j = 1, \dots, n_b, \quad (2)$$

where $y(t)$ is the system output and $u(t)$ the system input. $\xi(t)$ is assumed to be a white noise sequence independent of $u(t)$ with zero mean and variance of σ^2 . $v(t)$ is the output of nonlinear subsystem and the input to the linear subsystem. a_j 's and b_j 's are parameters of the linear subsystem. n_a and n_b are assumed to be known system output and input lags, respectively. Denote $\mathbf{a} = [a_1, \dots, a_{n_a}]^T \in \mathfrak{R}^{n_a}$ and $\mathbf{b} = [b_1, \dots, b_{n_b}]^T \in \mathfrak{R}^{n_b}$. It is assumed that $A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a}$ and $B(q^{-1}) = b_1q^{-1} + \dots + b_{n_b}q^{-n_b}$ are coprime polynomials of q^{-1} , where q^{-1} denotes the backward shift operator. The gain of the linear subsystem is given by

$$G = \lim_{q \rightarrow 1} \frac{B(q^{-1})}{A(q^{-1})} = \frac{\sum_{j=1}^{n_b} b_j}{1 + \sum_{j=1}^{n_a} a_j}. \quad (3)$$

The two objectives of the work are that of the system identification and the subsequent controller design for the identified model. The objective of system identification for the above Hammerstein model is that, given an observational input/output data set $D_N = \{y(t), u(t)\}_{t=1}^N$, to identify $\Psi(\bullet)$ and to estimate the parameters a_j, b_j in the linear subsystems. Note that the signals between the two subsystems are unavailable.

Without losing generality the following assumptions are initially made about the problem.

Assumption 1: $\Psi(\bullet)$ is a one-to-one mapping, i.e. it is an invertible and continuous function.

Assumption 2: $u(t)$ is bounded by $U_{\min} < u(t) < U_{\max}$, where U_{\min} and U_{\max} are assumed to be known finite real values.

In this work the B-spline basis functions are adopted in order to model $\Psi(\bullet)$. Specifically, the B-spline basis functions are initially formed by using the De Boor algorithm (de Boor 1978) for the input data sets.

2.2. Modelling of $\Psi(\bullet)$ using B-spline function approximation

Univariate B-spline basis functions are parameterised by the order of a piecewise polynomial of order k , and also by a knot vector which is a set of values defined on the real line that break it up into a number of intervals. Supposing that there are d basis functions, the knot vector is specified by $(d+k)$ knot values, $\{U_1, U_2, \dots, U_{d+k}\}$. At each end there are k knots satisfying the condition of being external to the input region, and as a result the number of internal knots is $(d-k)$. Specifically

$$\begin{aligned} U_1 &< U_2 < U_3 \\ &= U_{\min} < U_{k+1} < U_{k+2} < \dots < U_d < U_{\max} \\ &= U_{d+1} < \dots < U_{d+k}. \end{aligned} \quad (4)$$

Given these predetermined knots, a set of d B-spline basis functions can be formed by using the De Boor recursion (de Boor 1978), given by

$$\mathcal{B}_j^{(0)}(u) = \begin{cases} 1 & \text{if } U_j \leq u < U_{j+1}, \\ 0 & \text{otherwise,} \end{cases} \quad (5)$$

$$j = 1, \dots, (d+k)$$

$$\left. \begin{aligned} \mathcal{B}_j^{(i)}(u) &= \frac{u - U_j}{U_{i+j} - U_j} \mathcal{B}_i^{(i-1)}(u) + \frac{U_{i+j+1} - u}{U_{i+j+1} - U_{j+1}} \mathcal{B}_{j+1}^{(i-1)}(u), \\ j &= 1, \dots, (d+k-i) \end{aligned} \right\} \quad (6)$$

$$i = 1, \dots, k.$$

We model $\Psi(\bullet)$ as a B-spline neural network (Harris et al. 2002), in the form

$$\Psi(u) = \sum_{j=1}^d \mathcal{B}_j^{(k)}(u) \omega_j, \quad (7)$$

where ω_j 's are weights to be determined. Denote $\boldsymbol{\omega} = [\omega_1, \dots, \omega_d]^T \in \mathfrak{R}^d$. Note that due to the piecewise nature of B-spline functions, there are only $(k+1)$ basis functions with non-zero values for any point u . Hence, the computational cost for the evaluation of $\Psi(u)$ based on the De Boor algorithm is determined by the polynomial order k , rather than the number of knots, and this is in the order of $O(k^2)$.

2.3. The system identification algorithm

With the B-spline approximation, the model predicted output $\hat{y}(t)$ in (1) can be written as

$$\begin{aligned} \hat{y}(t) &= -a_1 y(t-1) - a_2 y(t-2) - \dots - a_{n_a} y(t-n_a) \\ &+ b_1 \sum_{j=1}^d \omega_j \mathcal{B}_j^{(k)}(t-1) + \dots + b_{n_b} \sum_{j=1}^d \omega_j \mathcal{B}_j^{(k)}(t-n_b). \end{aligned} \quad (8)$$

Over the estimation data set $D_N = \{y(t), u(t)\}_{t=1}^N$, (1) can be rewritten in a linear regression form

$$y(t) = [\mathbf{p}(\mathbf{x}(t))]^T \boldsymbol{\vartheta} + \xi(t), \quad (9)$$

where $\mathbf{x}(t) = [-y(t-1), \dots, -y(t-n_a), u(t-1), \dots, u(t-n_b)]^T$ is system input vector of observables with assumed known dimension of $(n_a + n_b)$, $\boldsymbol{\vartheta} = [\mathbf{a}^T, (b_1 \omega_1), \dots, (b_1 \omega_d), \dots, (b_{n_b} \omega_1), \dots, (b_{n_b} \omega_{n_b})]^T \in \mathfrak{R}^{n_a + d \cdot n_b}$,

$$\begin{aligned} \mathbf{p}(\mathbf{x}(t)) &= [-y(t-1), \dots, -y(t-n_a), \mathcal{B}_1^{(k)}(t-1), \dots, \\ &\mathcal{B}_d^{(k)}(t-1), \dots, \mathcal{B}_1^{(k)}(t-n_b), \dots, \mathcal{B}_d^{(k)}(t-n_b)]^T \end{aligned} \quad (10)$$

(9) can be rewritten in the matrix form as

$$\mathbf{y} = \mathbf{P} \boldsymbol{\vartheta} + \boldsymbol{\Xi}, \quad (11)$$

where $\mathbf{y} = [y(1), \dots, y(N)]^T$ is the output vector. $\boldsymbol{\Xi} = [\xi(1), \dots, \xi(N)]^T$, and \mathbf{P} is the regression matrix

$$\mathbf{P} = \begin{bmatrix} p_1(\mathbf{x}(1)) & p_2(\mathbf{x}(1)) & \dots & p_{n_a+d \cdot n_b}(\mathbf{x}(1)) \\ p_1(\mathbf{x}(2)) & p_2(\mathbf{x}(2)) & \dots & p_{n_a+d \cdot n_b}(\mathbf{x}(2)) \\ \dots & \dots & \dots & \dots \\ p_1(\mathbf{x}(N)) & p_2(\mathbf{x}(N)) & \dots & p_{n_a+d \cdot n_b}(\mathbf{x}(N)) \end{bmatrix}. \quad (12)$$

The parameter vector $\boldsymbol{\vartheta}$ can be found as the least squares solution of

$$\boldsymbol{\vartheta}_{LS} = \mathbf{B}^{-1} \mathbf{P}^T \mathbf{y} \quad (13)$$

provided that $\mathbf{B} = \mathbf{P}^T \mathbf{P}$ is of full rank. Alternatively, if this condition is violated, i.e. $\text{Rank}(\mathbf{B}) = r < n_a + d \cdot n_b$, then performing the singular value decomposition (SVD) $\mathbf{B} \mathbf{Q} = \mathbf{Q} \boldsymbol{\Sigma}$, where $\boldsymbol{\Sigma} = \text{diag}[\sigma_1, \dots, \sigma_r, 0, \dots, 0]$, $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_1, \dots, \mathbf{q}_{n_a+d \cdot n_b}]$, followed by truncating the eigenvectors corresponding to zero eigenvalues, we have

$$\boldsymbol{\vartheta}_{LS}^{svd} = \sum_{i=1}^r \frac{\mathbf{y}^T \mathbf{P} \mathbf{q}_i}{\sigma_i} \mathbf{q}_i. \quad (14)$$

This procedure produces our final estimate of $\hat{\mathbf{a}}$, which is simply taken as the subvector of the resultant $\boldsymbol{\vartheta}_{LS}^{svd}$, consisting of its first n_a elements. Clearly, information on $\hat{\mathbf{b}}$ and $\hat{\boldsymbol{\omega}}$ are contained in $\boldsymbol{\vartheta}_{LS}^{svd}$. Hence, it is straightforward to recover these based on Bai's approach using singular value decomposition (SVD; Bai and Fu 2002). Alternatively, the parameter estimation for \mathbf{b} and $\boldsymbol{\omega}$ can be obtained using our previous work (Hong and Mitchell 2007). This is outlined below and in the summary in Appendix A. Consider that a sequence $z(t)$ is generated, based on the derived parameter estimates $\hat{\mathbf{a}}$, as an auxiliary model

output sequence, given by

$$z(t) = y(t) + \hat{a}_1 y(t-1) + \hat{a}_2 y(t-2) + \dots + \hat{a}_{n_a} y(t-n_a). \quad (15)$$

$z(t)$ can be interpreted as an intermediate model residual after the model output $y(t)$ has been explained by the n_a regressors of $y(t-1), \dots, y(t-n_a)$. Then consider approximating $z(t)$ using the following model

$$\hat{z}(t) = b_1 \sum_{j=1}^d \omega_j \mathcal{B}_j^{(k)}(t-1) + \dots + b_{n_b} \sum_{j=1}^d \omega_j \mathcal{B}_j^{(k)}(t-n_b), \\ = g(\mathbf{x}(t), \mathbf{b}, \boldsymbol{\omega}). \quad (16)$$

By setting the objective function as

$$J_g(\mathbf{b}, \boldsymbol{\omega}) = \frac{1}{N} \sum_{i=1}^N [z(t) - g(\mathbf{x}(t), \mathbf{b}, \boldsymbol{\omega})]^2,$$

subject to

$$\frac{\sum_{j=1}^{n_b} \hat{b}_j}{1 + \sum_{j=1}^{n_a} \hat{a}_j} = 1,$$

the Gauss–Newton algorithm subject to constraints as proposed in Hong and Mitchell (2007) is used in this work. For completeness, see Appendix A.

3. The controller design combining the inverse of De Boor algorithm and pole assignment

3.1. The proposed inverse of De Boor algorithm

Using parameter estimates $\hat{\omega}_j$ (Appendix A), the output of the nonlinear subsystem is represented by

$$v = \hat{\Psi}(u) = \sum_{j=1}^d \mathcal{B}_j^{(k)}(u) \hat{\omega}_j. \quad (17)$$

In this section we consider the problem of finding its inverse, $u = \hat{\Psi}^{-1}(v)$, given that v lies in the region between two points, $\hat{\Psi}(U_{\min})$ and $\hat{\Psi}(U_{\max})$, i.e. finding the root of the polynomial equation $v = \hat{\Psi}(u) = \sum_{j=1}^d \mathcal{B}_j^{(k)}(u) \hat{\omega}_j$. In order to solve this problem, the inverse of De Boor algorithm is introduced below.

Initially, a sequence in the domain of v is generated as

$$V_i = \sum_{j=1}^d \mathcal{B}_j^{(k)}(U_i) \hat{\omega}_j, \quad i = 1, 2, \dots, (d+k). \quad (18)$$

Note that $v = \hat{\Psi}(u)$ is a one-to-one mapping, and this means that the resultant sequence due to the internal knots $[V_k, \dots, V_d]$ is either increasing or decreasing.

The algorithm is as follows.

- (1) Given v , and the sequence $\{V_i\}$, initially find

$$l = \arg\{(v - V_i)(v - V_{i+1}) < 0, \\ i = k, k+1, \dots, (d-1)\}. \quad (19)$$

- (2) Initialise $u^{(0)}$ as a random number with $U_l < u^{(0)} < U_{l+1}$.
- (3) The $(m+1)$ th step is given by

$$u^{(m+1)} = u^{(m)} + \Delta u^{(m)} \\ = u^{(m)} + \eta \cdot \text{sign} \left[\frac{V_d - V_k}{U_d - U_k} \right] (v - \hat{\Psi}(u^{(m)})), \quad (20)$$

where

$$\text{sign}(s) = \begin{cases} 1 & \text{if } s \geq 0, \\ -1 & \text{if } s < 0, \end{cases} \quad (21)$$

$0 < \eta \ll 1$ is the learning rate, that is preset empirically. $\hat{\Psi}(u^{(m)})$ is calculated using De Boor algorithm ((5) and (6) and (17)).

- (4) Set $m = m+1$, repeat Step 3 and 4, until $\frac{|\Delta u^{(m)}|}{U_d - U_k} < \varepsilon$, where $\varepsilon > 0$ is a predetermined small number in order to achieve the required precision, e.g. $\varepsilon = 10^{-3}$. Or the iteration can be terminated when m reaches a predetermined maximum value.

Theorem 1: The sequence $\{u^{(0)}, \dots, u^{(m)}, \dots\}$ converges to the unique solution of $u = \hat{\Psi}^{-1}(v)$, as $m \rightarrow \infty$, if $\eta < \frac{2}{|\hat{\Psi}'(u^{(m)})|}$, where $\hat{\Psi}'(u^{(m)}) = \frac{d\hat{\Psi}}{du} |_{u=u^{(m)}}$.

Proof: (i) Proof of the existence of the inverse. This is straightforward as it is easy to verify by the intermediate value theorem that there exists a point $U_l < u < U_{l+1}$, such that $v = \hat{\Psi}(u)$.

(ii) Proof of convergency. Construct a Lyapunov function in the form of

$$\mathcal{L}_m = (v - \Psi(u^{(m)}))^2 \geq 0. \quad (22)$$

Since η can be set as arbitrarily small, it is assumed that the following first order Taylor expansion is adequate

$$\hat{\Psi}(u^{(m+1)}) \approx \hat{\Psi}(u^{(m)}) + \hat{\Psi}'(u^{(m)}) \Delta u^{(m)}. \quad (23)$$

Clearly we have $\text{sign} \left[\frac{V_d - V_k}{U_d - U_k} \right] = \text{sign}[\hat{\Psi}'(u^{(m)})]$, such that

$$\text{sign} \left[\frac{V_d - V_k}{U_d - U_k} \right] \hat{\Psi}'(u^{(m)}) = |\hat{\Psi}'(u^{(m)})|. \quad (24)$$

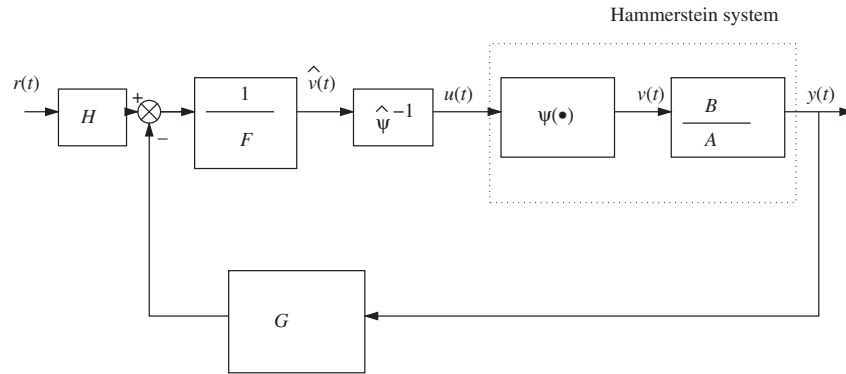


Figure 2. The control of Hammerstein system using pole assignment and the inverse of De Boor algorithm.

Applying (20), (23) and (24) to yield

$$\begin{aligned}
 \mathcal{L}_{m+1} - \mathcal{L}_m &= (2v - \hat{\Psi}(u^{(m)}) - \hat{\Psi}(u^{(m+1)}))(\hat{\Psi}(u^{(m)}) - \hat{\Psi}(u^{(m+1)})) \\
 &= -\hat{\Psi}'(u^{(m)})(2v - 2\hat{\Psi}(u^{(m)}) - \hat{\Psi}'(u^{(m)})\Delta u^{(m)})\Delta u^{(m)} \\
 &= -|\hat{\Psi}'(u^{(m)})|(2 - \eta|\hat{\Psi}'(u^{(m)})|)(v - \hat{\Psi}(u^{(m)}))^2 \\
 &= -|\hat{\Psi}'(u^{(m)})|(2 - \eta|\hat{\Psi}'(u^{(m)})|)\mathcal{L}_m < 0 \\
 \text{if } \eta &< \frac{2}{|\hat{\Psi}'(u^{(m)})|}. \tag{25}
 \end{aligned}$$

Hence, $\mathcal{L}_{m+1} - \mathcal{L}_m \rightarrow 0$, as $m \rightarrow \infty$. It follows from (25) that $\mathcal{L}_m \rightarrow 0$. Finally, because $\hat{\Psi}(u)$ is a one-to-one mapping, then u is the unique solution of $v = \hat{\Psi}(u)$. This concludes the proof.

3.2. Pole assignment controller

The closed-loop control system for Hammerstein model is as shown in Figure 2, in which

$$F(q^{-1}) = 1 + f_1q^{-1} + \dots + f_{n_f}q^{-n_f}, \tag{26}$$

$$G(q^{-1}) = g_0 + g_1q^{-1} + \dots + g_{n_g}q^{-n_g}, \tag{27}$$

$$H(q^{-1}) = h_0 + h_1q^{-1} + \dots + h_{n_h}q^{-n_h}, \tag{28}$$

where n_f , n_g and n_h are lags in the controller to be determined. Here the problem under study is the control of the Hammerstein system, of which the nonlinear subsystem is modelled as a B-spline curve and identified from input/output data. The proposed controller is the pole assignment design scheme for F , G , H (Wellstead, Edmunds, Pragerand, and Zanker 1979; Wellstead and Zarrop 1991), followed by $\hat{\Psi}^{-1}$, which is calculated using the inverse of the De Boor

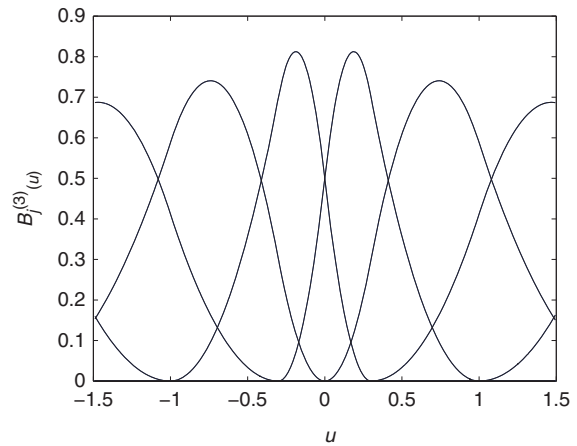


Figure 3. Eight B-spline basis functions used in the two Hammerstein systems.

algorithm as introduced in Section 3.1. Note that Figure 2 simply depicts the gain schedule linearisation using an approximation inverse (Astrom and Wittenmark 1989). In the following, it is shown that the B-spline based Hammerstein model is amenable to linear controller design, e.g. pole assignment.

We assume that the modelling of $\hat{\Psi}^{-1}$ using the inverse of De Boor algorithm, as proposed in this work, can cancel the actual nonlinearity in Hammerstein system. Hence, the closed-loop description of the system is

$$\underbrace{[AF + BG]}_{\text{closed loop denominator}} y(t) = BHr(t), \tag{29}$$

where $r(t)$ is a reference signal for the system output $y(t)$ to follow. The dynamics of the closed-loop are specified by a stable polynomial

$$AF + BG = T(q^{-1}) = 1 + t_1q^{-1} + \dots + t_nq^{-n}. \tag{30}$$

Table 1. Results of linear subsystem parameter estimation for two systems.

	a_1	a_2	b_1	b_2
True parameter	-1.2	0.9	1.7	-1
Estimate parameters (System 1, $\sigma^2=0.0001$)	-1.1986	0.8993	1.6886	-0.9880
Estimate parameters (System 1, $\sigma^2=0.01$)	-1.2003	0.9006	1.7115	-1.0112
Estimate parameters (System 2, $\sigma^2=0.0001$)	-1.1989	0.8994	1.6887	-0.9882
Estimate parameters (System 2, $\sigma^2=0.01$)	-1.1992	0.8999	1.6781	-0.9774

The coefficients of polynomials F , G can be solved by setting $n_f = n_b + 1$, $n_g = n_a - 1$, $n \leq n_a + n_b + 1$. H can be predetermined as desired subject to

$$\lim_{q \rightarrow 1} \frac{B(q^{-1})H(q^{-1})}{T(q^{-1})} = 1. \quad (31)$$

From Figure 2, it is clear that the actual control input $u(t)$ applied to the Hammerstein system is given by

$$u(t) = \hat{\Psi}^{-1}(\hat{v}(t)) = \hat{\Psi}^{-1}\left(\frac{Hr(t) - Gy(t)}{F}\right) \quad (32)$$

Rewriting (32) in a recursive form yields the following control law

- $\hat{v}(t) = \sum_{j=0}^{n_h} h_j r(t-j) - \sum_{j=0}^{n_g} g_j y(t-j) - \sum_{j=1}^{n_f} f_j \hat{v}(t-j)$,
- Find $u(t) = \hat{\Psi}^{-1}(\hat{v}(t))$ using the inverse of De Boor algorithm. (33)

Note that in practice if $\hat{v}(t)$ is out of the region between $\hat{\Psi}(U_{\min})$ and $\hat{\Psi}(U_{\max})$ (which is undesirable), $\hat{v}(t)$ is reset as 0 to avoid this to happen at the next time step. The computational cost of the inverse of De Boor algorithm is that of De Boor algorithm at $O(k^2)$, scaled by the number of iterations. The proposed controller is very efficient. First, because of the piecewise nature of B-spline, the polynomial order k can be kept quite low without losing its approximation performance. Second the number of iterations needed for convergence should be low as the proposed algorithm provides an initial solution that is close to the real inverse.

4. Numerical examples

Two Hammerstein systems are simulated, in which the linear subsystems are the same for both systems, as $A(q^{-1}) = 1 - 1.2q^{-1} + 0.9q^{-2}$, $B(q^{-1}) = 1.7q^{-1} - q^{-2}$. For the nonlinear subsystem, $\Psi(\bullet)$ is given by

$$\text{System 1 : } \Psi(u) = 2 \text{ sign}(u)\sqrt{|u|}, \quad (34)$$

$$\text{System 2 : } \Psi(u) = -2 \text{ sign}(u)u^2, \quad (35)$$

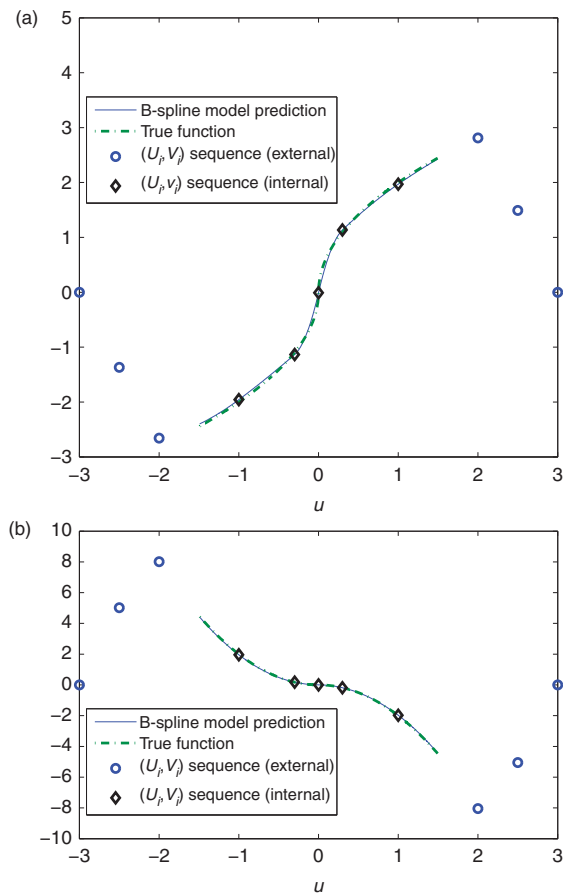


Figure 4. The modelling results for the nonlinear function $\Psi(u)$: (a) System 1 and (b) System 2.

respectively. Note that these nonlinear functions may be associated with physical devices such as nonlinear valves, e.g. the exit flow rate/fluid level in some parts of a pH neutralisation process (Henson and Seborg 1994).

The variances of the additive noise to the system output are set as $\sigma^2 = 0.0001$ and 0.01 , respectively, for both systems. For each system, 1000 training data samples $y(t)$ were generated by using (1) and (2), where $u(t)$ was uniformly distributed random variable

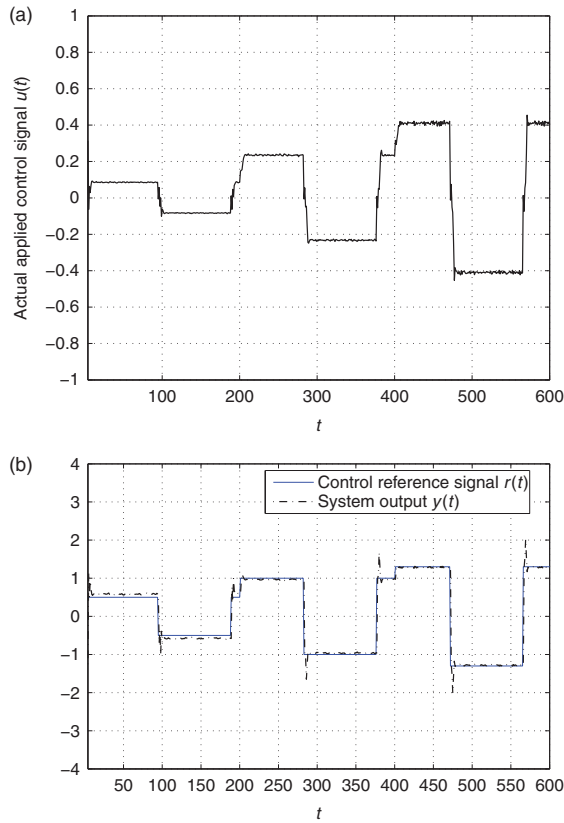


Figure 5. The results of the pole assignment controller for System 1.

$u(t) \in [-1.5, 1.5]$. The polynomial degree of B-spline basis functions was set as $k=2$ (piecewise quadratic). The knots sequence U_j is set as

$$[-3, -2.5, -2, -1, -0.3, 0, 0.3, 1, 2, 2.5, 3].$$

The resultant 8 basis functions are plotted in Figure 3. Initially, the system identification was carried out using the modelling algorithm outlined in Appendix A for each system. The modelling results are shown in Table 1, for the linear subsystem, and in Figure 4(a) and (b) for the nonlinear subsystems.

For both systems, the simulations of the pole assignment controllers were experimented based on a given polynomial $T(q^{-1}) = 1 - 0.6q^{-1} + 0.1q^{-2}$. Under the assumption that the proposed inverse of De Boor algorithm can cancel the nonlinearity in the system which is modelled as shown Figure 4, and by using parameter estimates given in Table 1, the required controller polynomials are estimated as

$$F(q^{-1}) = 1 - 0.4862q^{-1}$$

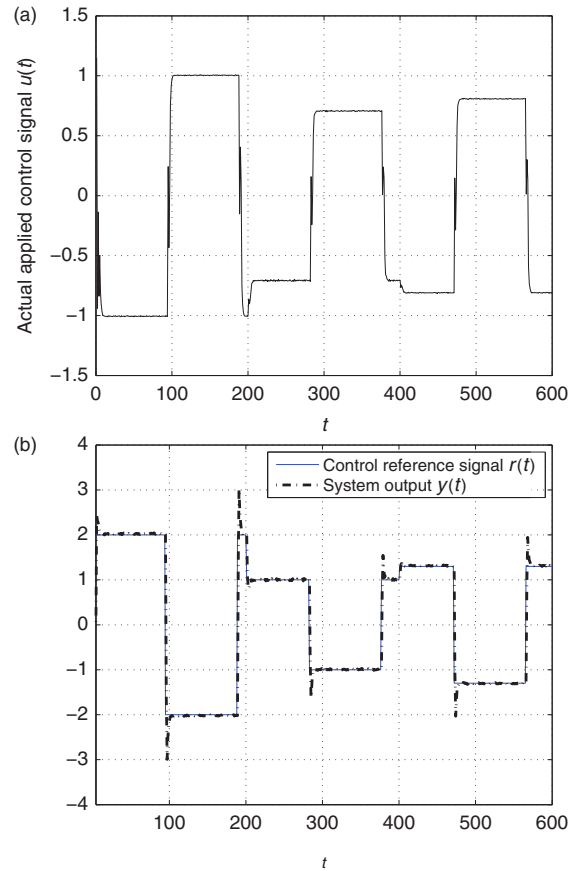


Figure 6. The results of the pole assignment controller for System 2.

and

$$G(q^{-1}) = 0.6425 - 0.4426q^{-1}.$$

and we predetermine

$$H(q^{-1}) = 0.5300 + 0.1836q^{-1}.$$

The learning rate was preset as $\eta = 0.02$. The maximum value of iteration number m was predetermined as 200. The performance of the controllers for the two systems were tested. For each system, the reference signals $r(t)$ are generated as a series of square waves with its magnitude at three different levels every 200 time steps. Figures 5(a) and 6(a) plot the computed control signal applied to each system, respectively. Figures 5(b) and 6(b) plot the system output $y(t)$ together with the corresponding reference signal $r(t)$ for both systems, respectively. From these figures, the proposed is shown method to have excellent results in terms of system identification as well as the subsequent control for the identified systems.

5. Conclusions

This brief article has introduced a simple and effective method for the modelling and control of the Hammerstein system based on observational input/output data. The modelling of the nonlinear static function in the Hammerstein system is based on the B-spline function approximation. The main contribution of this article is the calculation of the inverse of the estimated nonlinear function using a new algorithm, the inverse of De Boor algorithm. This step is used to remove the nonlinearity, leading to the linear controller methods. A pole assignment controller is used for completeness. Mathematical analysis is provided to prove the convergence of the proposed inverse of De Boor algorithm. The efficacy of the proposed approach have been demonstrated via two numerical examples.

Acknowledgements

The authors gratefully acknowledge that part of this work was supported by EPSRC in the UK. They also thank the reviewers for their valuable comments.

Notes on contributors



Xia Hong received her university education at National University of Defense Technology, P.R. China (BSc, 1984; MSc, 1987), and University of Sheffield, UK (PhD, 1998), all in automatic control. She worked as a research assistant in Beijing Institute of Systems Engineering, Beijing, China from 1987–1993. She worked as a research fellow in the Department of Electronics and Computer Science at University of Southampton from 1997–2001. She is currently a Reader of computational intelligence at School of Systems Engineering, University of Reading. She is actively engaged in research into nonlinear systems identification, data modelling, estimation and intelligent control, neural networks, pattern recognition, learning theory and their applications. She has published over 100 research papers, and coauthored a research book. She was awarded a Donald Julius Groen Prize by IMechE in 1999.



Richard Mitchell received his B.Sc. (Hons) in cybernetics and control engineering and his PhD in cybernetics from the Department of Cybernetics, University of Reading, Reading, UK, in 1980 and 1987, respectively. He was appointed Lecturer in Cybernetics in 1983 and is now Senior Lecturer in Cybernetics and also Senior Tutor in the School of Systems Engineering, University of Reading. He has published four textbooks, edited a custom book on cybernetics and has over 100 research papers in control engineering, robotics and learning systems.



Sheng Chen received his BEng in control engineering from the East China Petroleum Institute in 1982 and his PhD in control engineering from the City University at London in 1986. In 2005, he was awarded the Doctor of Sciences (DSc) by the University of Southampton. He joined the School of Electronics and Computer Science, the University of Southampton in September 1999. He previously held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth. Professor Chen is a Chartered Engineer (CEng), a Fellow of IET (FIET) and a Fellow of IEEE (FIEEE). His research interests are in adaptive signal processing for communications, wireless communications, modelling and identification of nonlinear systems, learning theory and neural networks, finite-precision digital controller design and networked control systems, evolutionary computation methods and optimisation. In the database of the world's most highly cited researchers in various disciplines, compiled by Institute for Scientific Information (ISI) of the USA, Professor Chen is on the list of highly cited researchers in the engineering category.

References

- Anbumani, K., Patnaik, L.M., and Sarma, I.G. (1981), 'Self-tuning Minimum Variance Control of Nonlinear Systems of the Hammerstein Model', *IEEE Transactions on Automatic Control*, AC-26, 959–961.
- Astrom, K.J., and Wittenmark, B. (1989), *Adaptive Control*, Reading, MA: Addison Wesley.
- Bai, E.W. (1998), 'An Optimal Two-stage Identification Algorithm for Hammerstein-Wiener Nonlinear Systems', *Automatica*, 34, 333–338.
- Bai, E.W., and Fu, M.Y. (2002), 'A Blind Approach to Hammerstein Model Identification', *IEEE Transactions on Signal Processing*, 50, 1610–1619.
- Balestrino, A., Landi, A., Ould-Zmirli, M., and Sani, L. (2001), 'Automatic Nonlinear Auto-tuning Method for Hammerstein Modelling of Electrical Drives', *IEEE Transactions on Industrial Electronics*, 48, 645–655.
- Billings, S.A., and Fakhouri, S.Y. (1979), 'Nonlinear System Identification Using the Hammerstein Model', *International Journal of Systems Science*, 10, 567–578.
- Bloemen, H.H., van den Boom, T.J., and Verbruggen, H.B. (2000), 'Model Based Predictive Control for Hammerstein Systems', in *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, pp. 4963–4968.
- Bloemen, H.H.J., van den Boom, T.J., and Verbruggen, H.B. (2001), 'Model-based Predictive Control for Hammerstein–Wiener Systems', *International Journal of Control*, 74, 482–295.
- Brown, M., and Harris, C.J. (1994), *Neurofuzzy Adaptive Modelling and Control*, Prentice Hall: Hemel Hempstead.
- Chaoui, F.Z., Giri, F., Rochdi, Y., Haloua, M., and Naitali, A. (2005), 'System Identification Based Hammerstein Model', *International Journal of Control*, 78, 430–442.

- Chen, H.F. (2004), 'Pathwise Convergence of Recursive Identification Algorithms for Hammerstein Systems', *IEEE Transactions on Automatic Control*, 49, 1873–1896.
- De Boor (1978), *A Practical Guide to Splines*, New York: Springer Verlag.
- Farin, G. (1994), *Curves and Surfaces for Computer-aided Geometric Design: A Practical Guide*, Boston: Academic Press.
- Farouki, R.T., and Goodman, T.N.T. (1996), 'On the Optimal Stability of the Bernstein Basis', *Mathematics of Computation*, 65, 1553–1566.
- Fruzzetti, E., Palazoglu, A., and McDonald, K.A. (1997), 'Nonlinear Model Predictive Control Using Hammerstein Models', *Journal of Process Control*, 7, 31–41.
- Greblicki, W. (1989), 'Nonparametric Orthogonal Series Identification of Hammerstein Systems', *International Journal of Systems Science*, 20, 2355–2367.
- Greblicki, W. (2002), 'Stochastic Approximation in Nonparametric Identification of Hammerstein Systems', *IEEE Transactions on Automatic Control*, 47, 1800–1810.
- Greblicki, W., and Pawlak, M. (1986), 'Identification of Discrete Hammerstein Systems Using Kernel Regression Estimate', *IEEE Transactions on Automatic Control*, AC-31, 74–77.
- Harris, C.J., Hong, X., and Gan, Q. (2002), *Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach*, Berlin, Heidelberg: Springer-Verlag.
- Henson, M.A., and Seborg, D.E. (1994), 'Adaptive Nonlinear Control of a pH Neutralization Process', *IEEE Transactions on Control Systems Technologies*, 2, 169–182.
- Hong, X., and Harris, C.J. (2000), 'Generalised Neurofuzzy Network Modelling Algorithms Using Bezier–Bernstein Polynomial Functions and Additive Decomposition', *IEEE Transactions on Neural Networks*, 11, 889–902.
- Hong, X., and Mitchell, R.J. (2006), 'A Pole Assignment Controller for Bezier–Bernstein Polynomial Based Hammerstein Model', in *Proceedings of International Control Conference (ICC) 2006*, Glasgow, UK.
- Hong, X., and Mitchell, R.J. (2007), 'A Hammerstein Model Identification Algorithm Using Bezier–Bernstein Approximation', *IET Proceedings Control Theory and Applications*, 1, 1149–1159.
- Hunter, I.W., and Korenberg, M.J. (1986), 'The Identification of Nonlinear Biological Systems: Wiener and Hammerstein Cascade Models', *Biological Cybernetics*, 55, 135–144.
- Kavli, T. (1993), 'ASMOD – An Algorithm for Adaptive Spline Modelling of Observation Data', *International Journal of Control*, 58, 947–967.
- Kwak, B., Yagle, A.E., and Levitt, J.A. (1998), 'Nonlinear System Identification of Hydraulic Actuator Friction Dynamics Using a Hammerstein Model', in *Proceedings of the IEEE ASSP'98*, Seattle, WA, pp. 1933–1936.
- Lang, Z.Q. (1997), 'A Nonparametric Polynomial Identification Algorithm for the Hammerstein System', *IEEE Transactions on Automatic Control*, 42, 1435–1441.
- Patwardhan, R.S., Lakshminarayanan, S., and Shah, S.L. (1998), 'Constrained Nonlinear MC Using Hammerstein and Wiener Model: PLS Framework', *AIChE Journal*, 44, 1611–1622.
- Stoica, P., and Söderström, T. (1982), 'Instrumental Variable Methods for Identification of Hammerstein Systems', *International Journal of Control*, 35, 459–476.
- Su, S.W. (2007), 'Identification and Control for Heart Rate Regulation During Treadmill Exercise', *IEEE Transactions on Biomedical Engineering*, 54, 1238–1246.
- Turunen, J., Tanttu, J.T., and Loula, P. (2003), 'Hammerstein Model for Speech Coding', *EURASIP Journal of Applied Signal Processing*, 12, 1238–1249.
- Verhaegen, M., and Westwick, D. (1996), 'Identifying MIMO Hammerstein Systems in the Context of Subspace Model Identification', *International Journal of Control*, 63, 331–349.
- Wellstead, P.E., Edmunds, J.M., Pragerand, D., and Zanker, P. (1979), 'Self Tuning Pole/Zero Assignment Regulator', *International Journal of Control*, 30, 1–26.
- Wellstead, P.E., and Zarrop, M.B. (1991), *Self-tuning Systems – Control and Signal Processing*, Chichester, UK: John Wiley & Sons.
- Zhu, Q.M., Warwick, K., and Douce, J.L. (1991), 'Adaptive General Predictive Controller for Nonlinear Systems', *IEE Proceedings Control Theory and Applications*, 138, 33–40.

Appendix A: A summary of the modelling algorithm

- (1) Based on the training data set and any prior knowledge of the system, predetermine the number of basis functions d , the polynomial order k and the input range $[U_{\min}, U_{\max}]$. Predetermine a set of $(d+k)$ knots according to (4).
- (2) Over the training data set $D_N = \{y(t), u(t)\}_{t=1}^N$, construct d basis functions $\mathcal{B}_i^{(d)}(u(t))$ based on (5) and (6). Subsequently, apply the method described in Section 2.3 to find the parameter vector $\hat{\mathbf{a}}$ as the subvector of \mathfrak{P}_{LS}^{svd} .
- (3) Construct the auxiliary model output sequence $z(t)$ using (15).
- (4) Apply the Gauss–Newton algorithm subject to constraint (Hong and Mitchell 2007) to find $\hat{\mathbf{b}}$ and $\hat{\omega}$.
- (5) Based on $\hat{\omega}$, the underlying function $\Psi(\bullet)$ for any point within the range (U_{\min}, U_{\max}) can be recovered by applying the De Boor algorithm (using (5)–(7)).