Complex-Valued B-Spline Neural Networks for Modeling and Inverting Hammerstein Systems

Sheng Chen, *Fellow, IEEE*, Xia Hong, *Senior Member, IEEE*, Junbin Gao, and Chris J. Harris

***Abstract*— Many communication signal processing applications involve modeling and inverting complex-valued (CV) Hammerstein systems. We develop a new CV B-spline neural network approach for efficient identification of the CV Hammerstein system and effective inversion of the estimated CV Hammerstein model. In particular, the CV nonlinear static function in the Hammerstein system is represented using the tensor product from two univariate B-spline neural networks. An efficient alternating least squares estimation method is adopted for identifying the CV linear dynamic model's coefficients and the CV B-spline neural network's weights, which yields the closed-form solutions for both the linear dynamic model's coefficients and the B-spline neural network's weights, and this estimation process is guaranteed to converge very fast to a unique minimum solution. Furthermore, an accurate inversion of the CV Hammerstein system can readily be obtained using the estimated model. In particular, the inversion of the CV nonlinear static function in the Hammerstein system can be calculated effectively using a Gaussian–Newton algorithm, which naturally incorporates the efficient De Boor algorithm with both the B-spline curve and first-order derivative recursions. The effectiveness of our approach is demonstrated using the application to equalization of Hammerstein channels.**

***Index Terms*— B-spline neural networks, complex-valued (CV) neural networks, De Boor algorithm, equalization, Hammerstein model, Wiener model.**

I. Introduction

COMPLEX-VALUED (CV) artificial neural networks have attracted considerable attention from both theoretical research and practical application communities [1]–[12]. In particular, the communication signal processing community has long been interested in neural network representations for the CV nonlinear systems as well as in inverting the CV nonlinear systems. It is well-known that most artificial neural networks cannot be automatically extended from the real-valued (RV) domain to the CV domain because the resulting model would in general violate Cauchy–Riemann conditions, and this means that the training algorithms become unusable. A number of analytic functions were introduced for the fully CV multilayer perceptrons [4]. A fully CV radial basis function network was introduced in [8] for regression and classification applications. Alternatively, the problem can be avoided using two RV artificial neural networks, one processing the real part and the other processing the imaginary part of the CV signal/system. A more challenging problem is the inversion of a CV nonlinear system, which is typically found in communication signal processing applications. This is a much under-researched area, and a few existing methods, such as the algorithm proposed in [10], are not very effective in tackling practical CV signal processing problems.

The RV signal processing field offers motivations and inspirations for the development of efficient techniques for modeling and inversion of the CV nonlinear systems. A popular approach to nonlinear systems modeling in the RV domain is to use block-oriented nonlinear models, which comprise the linear dynamic models and static or memoryless nonlinear functions [13]–[19]. In particular, the two types of RV block-oriented nonlinear models that have found wide range of applications are the Wiener model [20]–[26], which comprises a linear dynamical model followed by a nonlinear static transformation, and the Hammerstein model [27]–[34], which consists of a nonlinear static transformation followed by a linear dynamical model. An efficient B-spline neural network approach for modeling CV Wiener systems was derived in [35]. With its best conditioning property, the RV B-spline curve has been used in computer graphics and computer-aided geometric design [36]. The B-spline curves consist of many polynomial pieces, offering versatility. In particular, the De Boor algorithm [37], which uses numerically stable recurrence relations, offers a highly efficient means of constructing B-spline curve. The B-spline basis functions have been widely applied for the RV nonlinear system modeling [38]–[41]. The CV B-spline neural network algorithm derived in [35] is highly efficient for modeling the CV Wiener systems. Recently, we have extended [35] and further developed this highly effective B-spline neural network method for both modeling and inverting the CV Wiener systems, with an application to digital predistorter design [42].

Many communication signal processing applications involve propagating the CV signals through the CV nonlinear dynamic systems that can be represent by the Hammerstein model. For example, in order to improve the power efficiency of modern digital communication systems, the high power amplifier (HPA) at the transmitter is often driven to near the nonlinear saturation operating region [43]–[46]. This nonlinear distortion at the transmitter combined with the dispersive transmission medium forms a Hammerstein channel that seriously distorts the transmitted signal. At the receiver, equalization of

Manuscript received November 28, 2012; revised October 9, 2013 and January 1, 2014; accepted January 2, 2014. Date of publication January 21, 2014; date of current version August 15, 2014.

S. Chen is with the Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K., and also with the Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: sqc@ecs.soton.ac.uk).

X. Hong is with the School of Systems Engineering, University of Reading, Reading RG6 6AY, U.K. (e-mail: x.hong@reading.ac.uk).

J. Gao is with the School of Computing and Mathematics, Charles Sturt University, Bathurst 2795, Australia (e-mail: jbgao@csu.edu.au).

C. J. Harris is with the Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: cjh@ecs.soton.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2014.2298535

this CV Hammerstein channel is necessary for recovering the transmitted information sequence, which can be formulated as an identification and inversion of the CV Hammerstein system. We develop a novel CV B-spline neural network approach for efficient identification of the CV Hammerstein system as well as accurate inversion of the CV Hammerstein system based on the estimated CV Hammerstein model. Our original contribution is twofold. First, we extend [35] for the CV B-spline modeling to accommodate CV Hammerstein systems, and we develop a highly efficient alternating least squares (LS) method to identify the CV model coefficients of the linear dynamic subsystem as well as the CV weights of the B-spline neural network that models the static nonlinear subsystem, in the closed-form solutions. Since the cost function for this identification task is convex with respect to the coefficients of the linear model or the weights of the B-spline model, separately, and moreover, the solution is unique, our alternating LS estimator is guaranteed to converge very fast to a unique minimum, unlike the standard coordinate gradient-descent algorithm of [47] and [48]. Second, we develop an accurate and efficient inversion of the CV Hammerstein system based on the estimated B-spline model. In particular, the inversion of the CV nonlinear static function in the Hammerstein system can be effectively achieved using the Gauss–Newton algorithm that utilizes naturally the B-spline curve and first-order derivative recursions.

The effectiveness of this general approach for identification and inversion of the CV Hammerstein systems is demonstrated using the application to nonlinear equalization of Hammerstein channels, which is a nontrivial and challenging task. First, input signals for identifying the Hammerstein channel are not persistently exciting. In particular, under normal operating conditions of the HPA, there are very few or no transmitted digital signal near or in the saturation region of the nonlinear HPA. But, the CV B-spline neural network is capable of extrapolating well the nonlinear characteristics of the HPA into the saturation region where there are no training data. Second, the received signal is corrupted by the noise and, therefore, the input to the nonlinear equalizer or the inversion of the Hammerstein channel is highly noisy. The inversion of the Hammerstein channel under a noisy input causes the well-known noise enhancement problem. The most challenging difficulty, however, is due to the fact that the nonlinear static mapping of the HPA is one to one only up to the input saturation point. When the HPA is operating into the highly saturation region, transmitted digital symbols at the edges of the symbol constellation are distorted into the same saturated output amplitude value. Consequently, the CV B-spline model identified may not be strictly invertible over the output saturation region of the HPA. Despite of these difficulties, our proposed approach is effective and works well for this challenging application.

II. Identification and Inversion of CV Hammerstein Systems

A CV number $x \in \mathbb{C}$ can be represented either by the rectangular form $x = x_R + \mathrm{j}x_I$, where $\mathrm{j} = \sqrt{-1}$, whereas $x_R = \Re[x]$ and $x_I = \Im[x]$ denote the real and imaginary parts of x, or alternatively by the polar form $x = |x| \cdot \exp(\mathrm{j}\angle^x)$ with $|x|$ denoting the amplitude of x and $\angle^x$ its phase.

A. CV Hammerstein System

The generic CV Hammerstein system considered in this paper consists of a cascade of two subsystems: 1) a CV nonlinear static function $\Psi(\cdot) : \mathbb{C} \to \mathbb{C}$ and 2) a CV linear dynamic system presented by a finite-duration impulse response (FIR) filter of order L. Furthermore, the output of the system is corrupted by a CV additive white Gaussian noise (AWGN). Therefore, the system is represented by

$$w(k) = \Psi\left(x(k)\right) \tag{1}$$

$$y(k) = \widetilde{y}(k) + n(k) = \sum_{i=0}^{L} h_i w(k-i) + n(k), \ h_0 = 1 \tag{2}$$

where $x(k) \in \mathbb{C}$ and $y(k) \in \mathbb{C}$ are the system's input and output, respectively, and $n(k)$ is the system's AWGN with $E[|n_R(k)|^2] = E[|n_I(k)|^2] = \sigma_n^2$, whereas $\widetilde{y}(k)$ represents the noise-free system's output. The z transfer function of the FIR filter is defined by

$$H(z) = \sum_{i=0}^{L} h_i z^{-i}, \ h_0 = 1 \tag{3}$$

with the CV coefficient vector given by $\boldsymbol{h} = [h_1 \ h_2 \cdots h_L]^{\mathrm{T}} \in \mathbb{C}^L$. Note that $h_0 = 1$ is assumed. This is because, if this is not the case, h_0 can always be absorbed into the CV static nonlinearity $\Psi(\cdot)$, and the linear filter's coefficients are rescaled as h_i/h_0 for $0 \le i \le L$. Without loss of generality, the following assumptions are made regarding the CV nonlinear static function (1) of the Hammerstein system.

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

Assumption 2: $x_R(k)$ and $x_I(k)$ are upper and lower bounded by some finite and known real values.

For practical applications, these two assumptions typically hold. Our aim is to identify the above Hammerstein system, i.e., given the input–output data set $D_K = \{x(k), y(k)\}_{k=1}^{K}$, to identify the underlying nonlinear function $\Psi(\cdot)$ and to estimate the FIR filter parameter vector $\boldsymbol{h}$, as well as to provide an accurate inversion of the above Hammerstein system based on the identified model. Note that the signal $w(k)$ between the two subsystems is unavailable and the output data $\{y(k)\}_{k=1}^{K}$ are noisy. We will develop a CV B-spline neural network approach for an efficient identification of this Hammerstein system and then derive an effective algorithm for an accurate inversion of this Hammerstein system based on the estimated Hammerstein model $\hat{\Psi}(\cdot)$ and $\hat{\boldsymbol{h}}$.

B. CV B-Spline Neural Network

The CV B-spline neural network [35] is adopted to represent the mapping $\hat{w} = \hat{\Psi}(x_R + \mathrm{j}x_I) : \mathbb{C} \to \mathbb{C}$ that is the estimate of the underlying CV nonlinear function $\Psi(\cdot)$. Assume that $U_{\min} < x_R < U_{\max}$ and $V_{\min} < x_I < V_{\max}$, where $U_{\min}$, $U_{\max}$, $V_{\min}$, and $V_{\max}$ are known finite real values.

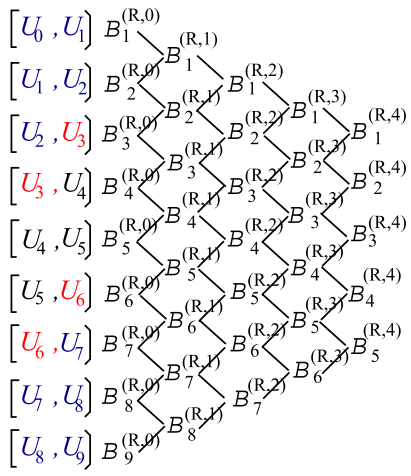


Fig. 1. Visualisation of the De Boor recursion for $P_o = 4$ and $N_b = 5$, where $U_{\min} = U_3$ and $U_{\max} = U_6$.

Within this range, the B-spline basis functions as model basis have the best approximation capability according to the Stone–Weierstrass approximation theorem, i.e., the basis function is complete. Although any polynomial function can also be used to approximate a continuous function, the B-spline functions are proven to be optimally stable bases [49].

A set of univariate B-spline basis functions based on x_R is parameterized by the order (P_o-1) of a piecewise polynomial and a knot sequence, which is a set of values defined on the real line that break it up into a number of intervals. To have N_R basis functions, the knot sequence is specified by (N_R+P_o+1) knot values, $\{U_0, U_1, \ldots, U_{N_R+P_o}\}$, with

$$\begin{aligned} U_0 &< U_1 < \cdots < U_{P_o-2} < U_{P_o-1} \\ &= U_{\min} < U_{P_o} < \cdots < U_{N_R} < U_{N_R+1} \\ &= U_{\max} < U_{N_R+2} < \cdots < U_{N_R+P_o}. \end{aligned} \tag{4}$$

At each end, there are P_o-1 external knots that are outside the input region and one boundary knot. As a result, the number of internal knots is N_R+1-P_o. Given the set of predetermined knots (4), the set of N_R B-spline basis functions can be formed by using the De Boor recursion [37], yielding

$$B_l^{(\Re,0)}(x_R) = \begin{cases} 1, & \text{if } U_{l-1} \le x_R < U_l \\ 0, & \text{otherwise} \end{cases} \quad 1 \le l \le N_R+P_o \tag{5}$$

$$\begin{aligned} B_l^{(\Re,p)}(x_R) = &\frac{x_R - U_{l-1}}{U_{p+l-1} - U_{l-1}} B_l^{(\Re,p-1)}(x_R) \\ &+ \frac{U_{p+l} - x_R}{U_{p+l} - U_l} B_{l+1}^{(\Re,p-1)}(x_R) \\ &\text{for } l = 1, \ldots, N_R + P_o - p \text{ and } p = 1, \ldots, P_o. \end{aligned} \tag{6}$$

The derivatives of the basis functions $B_l^{(\Re,P_o)}(x_R)$ for $1 \le l \le N_R$ can also be computed recursively according to

$$\begin{aligned} \frac{dB_l^{(\Re,P_o)}(x_R)}{dx_R} = &\frac{P_o}{U_{P_o+l-1} - U_{l-1}} B_l^{(\Re,P_o-1)}(x_R) \\ &- \frac{P_o}{U_{P_o+l} - U_l} B_{l+1}^{(\Re,P_o-1)}(x_R). \end{aligned} \tag{7}$$

The De Boor recursion is illustrated in Fig. 1. $P_o = 3$ or 4 is sufficient for the most practical applications. The number of B-spline basis functions should be chosen to be sufficiently large to provide accurate approximation capability but not too large as to cause overfitting and to impose unnecessary complexity. The internal knots may be uniformly spaced in the interval $[U_{\min},\ U_{\max}]$. The extrapolation capability of the B-spline model is influenced by the choice of the external knots. Note that there exist no data for $x_R < U_{\min}$ and $x_R > U_{\max}$ in identification but it is desired that the B-spline model has certain extrapolating capability outside the interval $[U_{\min},\ U_{\max}]$. The external knots can be set empirically to meet the required extrapolation capability.

Similarly, a set of univariate B-spline basis functions based on x_I can be established. Suppose that the order of the piecewise polynomial is again (P_o-1) and there are N_I basis functions. Then, the knot vector is defined on the imaginary line in a similar manner, which is specified by the (N_I+P_o+1) knot values, $\{V_0, V_1, \ldots, V_{N_I+P_o}\}$. In particular

$$\begin{aligned} V_0 &< V_1 < \cdots < V_{P_o-2} < V_{P_o-1} \\ &= V_{\min} < V_{P_o} < \cdots < V_{N_I} < V_{N_I+1} \\ &= V_{\max} < V_{N_I+2} < \cdots < V_{N_I+P_o}. \end{aligned} \tag{8}$$

Again, at each end, there are P_o-1 external knots that are outside the input region and one boundary knot, leaving N_I+1-P_o internal knots. Similarly, the set of N_I B-spline basis functions are constructed by the De Boor recursion [37] as

$$B_m^{(\Im,0)}(x_I) = \begin{cases} 1, & \text{if } V_{m-1} \le x_I < V_m \\ 0, & \text{otherwise} \end{cases} \quad 1 \le m \le N_I+P_o \tag{9}$$

$$\begin{aligned} B_m^{(\Im,p)}(x_I) = &\frac{x_I - V_{m-1}}{V_{p+m-1} - V_{m-1}} B_m^{(\Im,p-1)}(x_I) \\ &+ \frac{V_{p+m} - x_I}{V_{p+m} - V_m} B_{m+1}^{(\Im,p-1)}(x_I) \\ &\text{for } m = 1, \ldots, N_I + P_o - p \text{ and } p = 1, \ldots, P_o \end{aligned} \tag{10}$$

while the derivatives of the B-spline basis functions $B_m^{(\Im,P_o)}(x_I)$ for $1 \le m \le N_I$ are given recursively by

$$\begin{aligned} \frac{dB_m^{(\Im,P_o)}(x_I)}{dx_I} = &\frac{P_o}{V_{P_o+m-1} - V_{m-1}} B_m^{(\Im,P_o-1)}(x_I) \\ &- \frac{P_o}{V_{P_o+m} - V_m} B_{m+1}^{(\Im,P_o-1)}(x_I). \end{aligned} \tag{11}$$

Using the tensor product between the two sets of univariate B-spline basis functions [40], $B_l^{(\Re,P_o)}(x_R)$ for $1 \le l \le N_R$ and $B_m^{(\Im,P_o)}(x_I)$ for $1 \le m \le N_I$, a set of new B-spline basis functions $B_{l,m}^{(P_o)}(x)$ can be formed and used in the CV B-spline neural network, giving rise to

$$\begin{aligned} \hat{w} = \hat{\Psi}(x) &= \sum_{l=1}^{N_R} \sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(x) \omega_{l,m} \\ &= \sum_{l=1}^{N_R} \sum_{m=1}^{N_I} B_l^{(\Re,P_o)}(x_R) B_m^{(\Im,P_o)}(x_I) \omega_{l,m} \end{aligned} \tag{12}$$

where $\omega_{l,m} = \omega_{R_{l,m}} + \mathrm{j}\omega_{I_{l,m}} \in \mathbb{C}$, $1 \le l \le N_R$ and $1 \le m \le N_I$, are the CV weights. The CV B-spline neural network

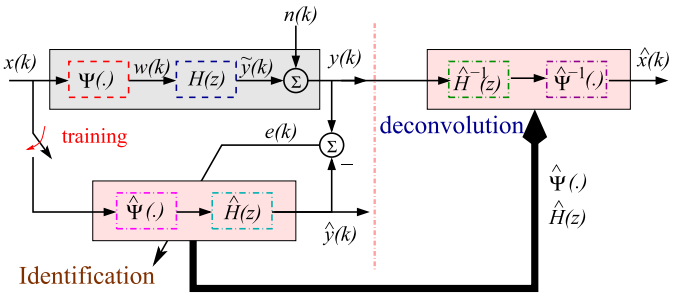


Fig. 2. Schematics of Hammerstein system identification and deconvolution.

(12) can obviously be decomposed as the following two RV B-spline neural networks:

$$\hat{w}_R = \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_l^{(\Re,P_o)}(x_R)B_m^{(\Im,P_o)}(x_I)\omega_{R_{l,m}} \tag{13}$$

$$\hat{w}_I = \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_l^{(\Re,P_o)}(x_R)B_m^{(\Im,P_o)}(x_I)\omega_{I_{l,m}}. \tag{14}$$

Because of the piecewise nature of B-spline functions, for any point evaluation, there are only P_o basis functions with nonzero values for each of the real and imaginary parts, leading to P_o^2 nonzero terms in both (13) and (14). This is advantageous as P_o can be set to a quite low value. The complexity of the De Boor recursion is in the order of P_o^2, $\mathcal{O}(P_o^2)$. Thus, the computational cost of calculating both (13) and (14) scales up to about three times of the De Boor recursion, including evaluation of both real and imaginary parts as well as the tensor product calculation. Notably, additional cost for derivative evaluation is minimal, as (7) and (11) are a byproduct of the De Boor recursion. In addition, there are only P_o nonzero first-order derivative terms in each of (7) and (11). Compared with other CV neural networks based on different spline functions [3], [50], [51], our approach is clearly different in terms of model representation and identification algorithm. The advantages of our CV B-spline neural network, in comparison with these other spline functions [3], [50], [51], are discussed in [35].

C. Hammerstein System Identification

The schematic view of the CV Hammerstein system identification is depicted in the left part of Fig. 2. For the chosen two sets of knots, (4) and (8), and the polynomial degree P_o, denote the weight vector of the CV B-spline neural network (12) as

$$\boldsymbol{\omega} = \left[\omega_{1,1}\ \omega_{1,2}\cdots\omega_{l,m}\cdots\omega_{N_R,N_I}\right]^{\mathrm{T}} \in \mathbb{C}^N \tag{15}$$

where $N = N_R N_I$. Given a block of training input–output data $\{\boldsymbol{x}(k), y(k)\}_{k=1}^K$, where $\boldsymbol{x}(k) = [x(k)\ x(k-1)\cdots x(k-L)]^{\mathrm{T}}$, the task is to estimate $\boldsymbol{\omega}$ and $\boldsymbol{h}$. The identification task can be formulated as the one that minimises the cost function

$$J_{\mathrm{icf}}(\boldsymbol{h},\boldsymbol{\omega}) = \frac{1}{K}\sum_{k=1}^{K}|e(k)|^2 = \frac{1}{K}\sum_{k=1}^{K}|y(k)-\hat{y}(k)|^2 \tag{16}$$

with

$$\hat{y}(k) = \sum_{i=0}^{L} h_i\hat{w}(k-i) = \sum_{i=0}^{L} h_i \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(x(k-i))\omega_{l,m} \tag{17}$$

in which $h_0 = 1$. Note that the cost function (16) is convex with respect to $\boldsymbol{h}$ when fixing $\boldsymbol{\omega}$ and is convex with respect to $\boldsymbol{\omega}$ given a fixed $\boldsymbol{h}$. We adopt an iterative procedure of alternating the LS estimation of $\boldsymbol{h}$ and the LS estimation of $\boldsymbol{\omega}$. Unlike a generic coordinate gradient descent algorithm [47], [48], we have the close-form solutions for both $\boldsymbol{h}$ and $\boldsymbol{\omega}$ and our estimate is unique, owing to the unique parameterization of $\boldsymbol{h}$ and $\boldsymbol{\omega}$. Thus, our estimation algorithm converges fast to a unique and unbiased estimate of $\boldsymbol{h}$ and $\boldsymbol{\omega}$ jointly.

Initialization. Define the amalgamated parameter vector as

$$\boldsymbol{\theta} = \left[\boldsymbol{\omega}^{\mathrm{T}}\ h_1\boldsymbol{\omega}^{\mathrm{T}}\ h_2\boldsymbol{\omega}^{\mathrm{T}}\cdots h_L\boldsymbol{\omega}^{\mathrm{T}}\right]^{\mathrm{T}} \in \mathbb{C}^{(L+1)N} \tag{18}$$

and the B-spline basis function vector $\boldsymbol{\phi}(k) \in \mathbb{R}^N$ for the input $x(k)$ as

$$\boldsymbol{\phi}(k) = \left[\phi_{1,1}(k)\ \phi_{1,2}(k)\cdots\phi_{l,m}(k)\cdots\phi_{N_R,N_I}(k)\right]^{\mathrm{T}} \tag{19}$$

with

$$\phi_{l,m}(k) = B_{l,m}^{(P_o)}(x(k)),\ 1\le l\le N_R, 1\le m\le N_I. \tag{20}$$

Furthermore, define the desired output vector as

$$\boldsymbol{y} = \left[y(1)\ y(2)\cdots y(K)\right]^{\mathrm{T}} \in \mathbb{C}^K \tag{21}$$

and the regression matrix $\boldsymbol{P} \in \mathbb{R}^{K\times(L+1)N}$ as

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{\phi}^{\mathrm{T}}(1) & \boldsymbol{\phi}^{\mathrm{T}}(0) & \cdots & \boldsymbol{\phi}^{\mathrm{T}}(1-L) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\phi}^{\mathrm{T}}(k) & \boldsymbol{\phi}^{\mathrm{T}}(k-1) & \cdots & \boldsymbol{\phi}^{\mathrm{T}}(k-L) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\phi}^{\mathrm{T}}(K) & \boldsymbol{\phi}^{\mathrm{T}}(K-1) & \cdots & \boldsymbol{\phi}^{\mathrm{T}}(K-L) \end{bmatrix}. \tag{22}$$

Then, the LS estimate of $\boldsymbol{\theta}$ is readily given by

$$\hat{\boldsymbol{\theta}} = \left(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{P}\right)^{-1}\boldsymbol{P}^{\mathrm{T}}\boldsymbol{y}. \tag{23}$$

The first N elements of $\hat{\boldsymbol{\theta}}$ provide a unique and unbiased LS estimate for the weight vector of the CV B-spline neural network $\boldsymbol{\omega}$ defined in (15), which will be denoted as $\hat{\boldsymbol{\omega}}^{(0)}$.

Alternating LS estimation. For $1 \le \tau \le \tau_{\max}$, where $\tau_{\max}$ is the maximum number of iterations, the following is performed. **1)** Given the fixed $\hat{\boldsymbol{\omega}}^{(\tau-1)}$, calculate the LS estimate $\hat{\boldsymbol{h}}^{(\tau)}$. Especially, define the desired output vector as

$$\boldsymbol{y}_h = \left[y_h(1)\ y_h(2)\cdots y_h(K)\right]^{\mathrm{T}} \in \mathbb{C}^K \tag{24}$$

with

$$y_h(k) = y(k) - \boldsymbol{\phi}^{\mathrm{T}}(k)\hat{\boldsymbol{\omega}}^{(\tau-1)} = y(k) - \hat{w}_\omega(k) \tag{25}$$

and the regression matrix $\boldsymbol{Q} \in \mathbb{C}^{K\times L}$ as

$$\boldsymbol{Q} = \begin{bmatrix} \hat{w}_\omega(0) & \hat{w}_\omega(-1) & \cdots & \hat{w}_\omega(1-L) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{w}_\omega(k-1) & \hat{w}_\omega(k-2) & \cdots & \hat{w}_\omega(k-L) \\ \vdots & \vdots & \vdots & \vdots \\ \hat{w}_\omega(K-1) & \hat{w}_\omega(K-2) & \cdots & \hat{w}_\omega(K-L) \end{bmatrix}. \tag{26}$$

Then, the LS estimate $\hat{\boldsymbol{h}}^{(\tau)}$ is readily given by

$$\hat{\boldsymbol{h}}^{(\tau)} = \left(\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^{\mathrm{H}}\boldsymbol{y}_h. \tag{27}$$

2) Given the fixed $\hat{\boldsymbol{h}}^{(\tau)}$, calculate the LS estimate $\hat{\boldsymbol{\omega}}^{(\tau)}$. Especially, set $\hat{h}_0^{(\tau)} = 1$, and introduce

$$\varphi_{l,m}(k) = \sum_{i=0}^{L} \hat{h}_i^{(\tau)} B_{l,m}^{(P_o)}(x(k-i)) \in \mathbb{C}. \tag{28}$$

Now, introduce the regressor vector $\boldsymbol{\varphi}(k) \in \mathbb{C}^N$ given by

$$\boldsymbol{\varphi}(k) = \left[\varphi_{1,1}(k)\ \varphi_{1,2}(k) \cdots \varphi_{l,m}(k) \cdots \varphi_{N_R,N_I}(k)\right]^{\mathrm{T}} \tag{29}$$

and define the regression matrix

$$\boldsymbol{S} = \left[\boldsymbol{\varphi}(1)\ \boldsymbol{\varphi}(2) \cdots \boldsymbol{\varphi}(K)\right]^{\mathrm{T}} \in \mathbb{C}^{K\times N}. \tag{30}$$

Then, the LS estimate $\hat{\boldsymbol{\omega}}^{(\tau)}$ is readily given by

$$\hat{\boldsymbol{\omega}}^{(\tau)} = \left(\boldsymbol{S}^{\mathrm{H}}\boldsymbol{S}\right)^{-1}\boldsymbol{S}^{\mathrm{H}}\boldsymbol{y}. \tag{31}$$

A few iterations are sufficient for this estimation procedure to converge to a joint unbiased estimate of $\boldsymbol{h}$ and $\boldsymbol{\omega}$ that is the unique minimum solution of the cost function (16).

D. Hammerstein System Inversion

For the Hammerstein system (1) and (2), there exists two types of inversion: 1) the preinversion and 2) the postinversion. In either case, the exact inversion of the Hammerstein system is a Wiener system consisting of a linear filter followed by a nonlinear static function. However, for the preinversion, the input to the Wiener inverse model is typically a clean, i.e., noise-free, signal, whereas for the postinversion, the input signal to the Wiener inverse model is noisy. In this paper, we consider the more challenging postinversion or deconvolution of the CV Hammerstein system as depicted in the right part of Fig. 2.

1) Inversion of Hammerstein System's Linear Filter: The identification algorithm of Section II-C produces the estimate of the Hammerstein system's linear filter

$$\hat{H}(z) = 1 + \sum_{i=1}^{L} \hat{h}_i z^{-i}.$$

Let the transfer function of the Wiener inverse model's linear filter be

$$G(z) = z^{-\iota} \cdot \sum_{i=0}^{L_g} g_i z^{-i} \tag{32}$$

where the delay $\iota = 0$ if $H(z)$ is minimum phase. To guarantee an accurate inversion, the length of the inverse linear filter L_g should be chosen to be three to four times of the length of $\boldsymbol{h}$, but not too long in order not to amplify the noise in the input signal too much. The solution of the Wiener inverse model's linear filter $\boldsymbol{g} = [g_0\ g_1 \cdots g_{L_g}]^{\mathrm{T}}$ may be obtained directly by solving the set of linear equations specified by

$$G(z) \cdot \hat{H}(z) = z^{-\iota} \tag{33}$$

which is also known as the zero-forcing (ZF) solution. Note that $g_0 = 1$ as $h_0 = 1$. However, as the input $y(k)$ to the filter $G(z)$ is noisy, the ZF solution may suffer from the drawback of amplifying the noise $n(k)$ in the input signal $y(k)$ too much. The well-known minimum mean square error (MMSE) solution [52] offers a better tradeoff between the accuracy of inversion and the noise amplification.

Define the coefficient matrix $\hat{\boldsymbol{H}} \in \mathbb{C}^{(L_g+1)\times(L_H+1)}$, where $L_H = L + L_g$, corresponding to the estimated Hammerstein system's linear filter as

$$\begin{aligned}\hat{\boldsymbol{H}} &= \begin{bmatrix} \hat{h}_0 & \hat{h}_1 & \cdots & \hat{h}_L & 0 & \cdots & 0 \\ 0 & \hat{h}_0 & \hat{h}_1 & \cdots & \hat{h}_L & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \cdots & \ddots & 0 \\ 0 & \cdots & 0 & \hat{h}_0 & \hat{h}_1 & \cdots & \hat{h}_L \end{bmatrix} \\ &= \begin{bmatrix} \hat{\mathbf{h}}_0 & \hat{\mathbf{h}}_1 & \cdots & \hat{\mathbf{h}}_\iota & \cdots & \hat{\mathbf{h}}_{L_H} \end{bmatrix}\end{aligned} \tag{34}$$

where $\hat{h}_0 = 1$. Then, the MMSE solution of $\boldsymbol{g}$ is expressed as

$$\boldsymbol{g}_{\mathrm{MMSE}} = \left(\hat{\boldsymbol{H}}\hat{\boldsymbol{H}}^{\mathrm{H}} + \frac{2\hat{\sigma}_n^2}{\sigma_w^2}\boldsymbol{I}_{L_H+1}\right)^{-1}\hat{\mathbf{h}}_\iota \tag{35}$$

in which $\boldsymbol{I}_L$ denotes the $L \times L$ identity matrix

$$\sigma_w^2 = \mathrm{E}\left\{\left|\Psi(x(k))\right|^2\right\} \approx 1/K \sum_{k=1}^{K} \left|\hat{\Psi}(x(k))\right|^2$$

and $2\hat{\sigma}_n^2$ is the estimate of the noise power given by $2\hat{\sigma}_n^2 \approx J_{\mathrm{icf}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{\omega}})$. The optimal value for ι can in fact be chosen to minimize the MMSE of the combined linear system of $\hat{\boldsymbol{h}}$ and $\boldsymbol{g}$

$$J_{\mathrm{cmmse}}(\iota) = \sigma_w^2\left(1 - \hat{\mathbf{h}}_\iota^{\mathrm{H}}\left(\hat{\boldsymbol{H}}\hat{\boldsymbol{H}}^{\mathrm{H}} + \frac{2\hat{\sigma}_n^2}{\sigma_w^2}\boldsymbol{I}_{L_H+1}\right)^{-1}\hat{\mathbf{h}}_\iota\right). \tag{36}$$

2) Inversion of Hammerstein System's Static Nonlinear Function: Given the CV Hammerstein system's static nonlinearity $\Psi(\cdot)$, we wish to compute its inversion defined by $x(k) = \Psi^{-1}(w(k))$. This task is identical to find the CV root of $w(k) = \Psi(x(k))$, given $w(k)$. In Section II-C, the estimate $\hat{\Psi}(\cdot)$ for $\Psi(\cdot)$ has been obtained based on the CV B-spline neural network. We now show that $\hat{\Psi}^{-1}(\cdot)$ can be effectively obtained. Given $\hat{\Psi}(\cdot)$, we have

$$\hat{w}_R(k) = \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_l^{(\Re,P_o)}(x_R(k)) B_m^{(\Im,P_o)}(x_I(k))\hat{\omega}_{R_{l,m}} \tag{37}$$

$$\hat{w}_I(t) = \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_l^{(\Re,P_o)}(x_R(k)) B_m^{(\Im,P_o)}(x_I(k))\hat{\omega}_{I_{l,m}} \tag{38}$$

where $\hat{\omega}_{l,m} = \hat{\omega}_{R_{l,m}} + \mathrm{j}\hat{\omega}_{I_{l,m}}$. Define $\zeta(k) = w(k) - \hat{w}(k)$ and the squared error (SE)

$$S(k) = \zeta_R^2(k) + \zeta_I^2(k). \tag{39}$$

If $S(k) = 0$, then $x(k)$ is the CV root of $w(k) = \hat{\Psi}(x(k))$. Thus, the task is equivalent to the one that minimizes the SE (39). We propose to use the following Gauss–Newton algorithm to solve this optimization problem with the aid of the efficient De Boor algorithm.

By denoting the iteration step with the superscript $^{(t)}$ and giving a random initialization of $x^{(0)}(k)$ that satisfies $U_{\min} <$

$x_R^{(0)}(k) < U_{\max}$ and $V_{\min} < x_I^{(0)}(k) < V_{\max}$, the iterative procedure is given by

$$\begin{bmatrix} x_R^{(t)}(k) \\ x_I^{(t)}(k) \end{bmatrix} = \begin{bmatrix} x_R^{(t-1)}(k) \\ x_I^{(t-1)}(k) \end{bmatrix} - \eta\Big(\big(\mathbf{J}_x^{(t)}\big)^{\mathrm{T}}\mathbf{J}_x^{(t)}\Big)^{-1}\big(\mathbf{J}_x^{(t)}\big)^{\mathrm{T}} \times \begin{bmatrix} \zeta_R^{(t-1)}(k) \\ \zeta_I^{(t-1)}(k) \end{bmatrix} \tag{40}$$

where $\eta > 0$ is the step size, $\zeta^{(t)}(k) = w(k) - \hat{w}^{(t)}(k)$ with $\hat{w}^{(t)}(k) = \hat{\Psi}\big(x^{(t)}(k)\big)$, and $\mathbf{J}_x^{(t)}$ is the 2×2 Jacobian matrix given by

$$\mathbf{J}_x^{(t)} = \begin{bmatrix} \dfrac{\partial \zeta_R(k)}{\partial x_R(k)} & \dfrac{\partial \zeta_R(k)}{\partial x_I(k)} \\ \dfrac{\partial \zeta_I(k)}{\partial x_R(k)} & \dfrac{\partial \zeta_I(k)}{\partial x_I(k)} \end{bmatrix}_{|x(k)=x^{(t)}(k).} \tag{41}$$

The entries in (41) are given by

$$\begin{cases} \dfrac{\partial \zeta_R(k)}{\partial x_R(k)} = -\displaystyle\sum_{l=1}^{N_R}\sum_{m=1}^{N_I} \frac{dB_l^{(\Re,P_o)}(x_R(k))}{dx_R(k)} B_m^{(\Im,P_o)}(x_I(k))\hat{\omega}_{R_{l,m}} \\ \dfrac{\partial \zeta_R(k)}{\partial x_I(k)} = -\displaystyle\sum_{l=1}^{M_R}\sum_{m=1}^{N_I} B_l^{(\Re,P_o)}(x_R(k)) \frac{dB_m^{(\Im,P_o)}(x_I(k))}{dx_I(k)}\hat{\omega}_{R_{l,m}} \\ \dfrac{\partial \zeta_I(k)}{\partial x_R(k)} = -\displaystyle\sum_{l=1}^{N_R}\sum_{m=1}^{N_I} \frac{dB_l^{(\Re,P_o)}(x_R(k))}{dx_R(k)} B_m^{(\Im,P_o)}(x_I(k))\hat{\omega}_{I_{l,m}} \\ \dfrac{\partial \zeta_I(k)}{\partial x_I(k)} = -\displaystyle\sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_l^{(\Re,P_o)}(x_R(k)) \frac{dB_m^{(\Im,P_o)}(x_I(t))}{dx_I(k)}\hat{\omega}_{I_{l,m}} \end{cases} \tag{42}$$

for which the De Boor algorithm, (5)–(7) and (9)–(11), can be used for their calculation efficiently. The algorithm is terminated when $S(k) < \rho$, where ρ is a preset required precision, e.g., $\rho = 10^{-8}$, or when t reaches a predetermined maximum value. The step size η of the Gauss–Newton algorithm can be set to a relatively large value. As this is a 2-D problem, 10 iterations are often sufficient for the Gauss–Newton algorithm to converge. It can readily be seen that our B-spline function approach has clear advantages in calculating $\Psi^{-1}(\cdot)$, in comparison with other alternative spline function approaches [3], [50], [51].

III. Equalization of Hammerstein Channels

Modern digital communication systems, such as the high-order quadrature amplitude modulation (QAM) system [53] and the multicarrier orthogonal frequency division multiplexing system [54], are highly bandwidth efficient, but they require that the HPA at the transmitter has a very large linear dynamic range, which cannot be met by practical HPAs [43]–[46]. Consequently, the HPAs at the transmitters of these systems are often driven to near the nonlinear saturation operating region, causing nonlinear distortions to the transmitted signals. Note that, in order to achieve better power efficiency, the HPAs may be deliberately operating near their saturation regions. Transmission media or channels for these high bandwidth-efficiency communication systems are typically dispersive and can be represented by linear FIR filters [55]. Therefore, the communication channel for these systems is a Hammerstein system consisting of the HPA nonlinearity at the transmitter and the linear FIR filter of the transmission medium. At the receiver, a nonlinear equalizer must be deployed to remove the effects of this Hammerstein channel in order to recover the transmitted information sequence.

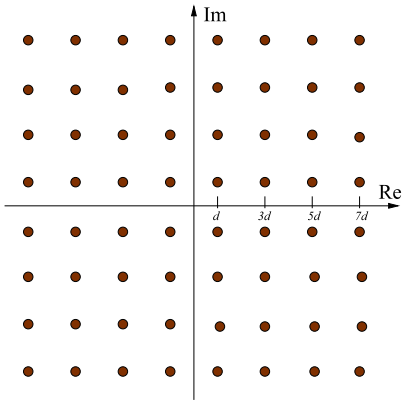


Fig. 3. 64-QAM symbol constellation.

A. Hammerstein Channel

Without loss of generality, we consider the high-order QAM signaling [53], in which the transmitted signal $x(k)$ takes the values from the CV M-QAM symbol set

$$\mathbb{S} = \{d(2l - \sqrt{M} - 1) + \mathrm{j}d(2q - \sqrt{M} - 1), 1 \le l, q \le \sqrt{M}\} \tag{43}$$

where $2d$ is the minimum distance between symbol points. The index k here denotes the discrete time or symbol index. Each symbol of the M-QAM signaling conveys $\log_2(M)$ information bits. The 64-QAM symbol constellation is illustrated in Fig. 3. The Hammerstein channel is represented by: 1) the CV static nonlinearity and 2) the CV FIR filter of order L, where $n(k)$ is the channel's AWGN. The signal-to-noise ratio (SNR) of the system is usually expressed as $\mathrm{SNR} = E_b/N_o$, where $N_o = 2\sigma_n^2$ is the AWGNs power, whereas E_b denotes the average energy per bit for the M-QAM signaling.

Two typical CV nonlinearities $\Psi(\cdot)$ of HPAs are the traveling-wave tube (TWT) nonlinearity [43] and the nonlinearity of solid state power amplifiers [44]. Nonlinear characteristics of these two types of HPAs are similar. Without loss of generality, we consider the TWT HPA in this paper. The output $w(k)$ of the HPA for the given input $x(k) = |x(k)| \cdot \exp(\mathrm{j}\angle^{x(k)})$ can be expressed as

$$w(k) = A(|x(k)|) \cdot \exp\big(\mathrm{j}\angle^{x(k)} + \mathrm{j}\Upsilon(|x(k)|)\big). \tag{44}$$

Denote the input amplitude $r_x(k) = |x(k)|$. The output amplitude $A\big(r_x(k)\big)$ and the phase $\Upsilon\big(r_x(k)\big)$ of the TWT HPA are specified respectively by [43] and [45]

$$A\big(r_x\big) = \begin{cases} \dfrac{\alpha_a r_x}{1 + \beta_a r_x^2}, & 0 \le r_x \le r_{\mathrm{sat}} \\ A_{\max}, & r_x > r_{\mathrm{sat}} \end{cases} \tag{45}$$

$$\Upsilon\big(r_x\big) = \frac{\alpha_\phi r_x^2}{1 + \beta_\phi r_x^2} \tag{46}$$

where the saturating input amplitude is defined as

$$r_{\mathrm{sat}} = \frac{1}{\sqrt{\beta_a}} \tag{47}$$

while the saturation output amplitude is given by

$$A_{\max} = \frac{\alpha_a}{2\sqrt{\beta_a}}. \tag{48}$$

The underlying physics require that $A_{\max} > r_{\text{sat}}$ and for the M-QAM signal $x(k)$, $r_x(k)$ meets the condition $r_x(k) \leq \sqrt{2}\big(\sqrt{M}-1\big)d$. The operating status of the HPA is specified by the input back-off (IBO), which is defined as

$$\text{IBO} = 10 \cdot \log_{10} \frac{P_{\text{sat}}}{P_{\text{avg}}} \tag{49}$$

where $P_{\text{sat}} = r_{\text{sat}}^2$ is the saturation input power, whereas

$$\begin{aligned} P_{\text{avg}} &= \text{E}\big\{|x(k)|^2\big\} \\ &= \frac{1}{M}\sum_{l=1}^{\sqrt{M}}\sum_{q=1}^{\sqrt{M}}\Big(d^2\big(2l-\sqrt{M}-1\big)^2+d^2\big(2q-\sqrt{M}-1\big)^2\Big) \end{aligned} \tag{50}$$

is the average power of the input signal $x(k)$ to the HPA. Note that $E_b = P_{\text{avg}}/\log_2 M$. A small IBO value indicates that the HPA operates in the nonlinear saturation region. In summary, the Hammerstein channel is specified by the CV HPAs nonlinearity (45) and (46) with the positive RV parameter vector $\mathbf{t} = [\alpha_a \ \beta_a \ \alpha_\phi \ \beta_\phi]^{\text{T}} \in \mathbb{R}_+^4$ as well as the FIR filter of order L with the CV parameter vector $\boldsymbol{h} = \big[h_1 \ h_2 \ldots h_L\big]^{\text{T}} \in \mathbb{C}^L$.

B. Nonlinear Equalization

Based on the generic technique developed in Section II for identification and inversion of the CV Hammerstein system, we design a novel nonlinear Wiener equalizer for the Hammerstein channel presented in Section III-A. During the training, the receiver has access to the transmitted M-QAM input signal $\boldsymbol{x}(k) = \big[x(k) \ x(k-1) \ldots x(k-L)\big]^{\text{T}}$ for $1 \leq k \leq K$ and the Hammerstein channel output signal $\{y(k)\}_{k=1}^K$. Note that the HPAs output $w(k)$ at the transmitter is unavailable at the receiver, and the target or desired output $y(k)$ for identification of the Hammerstein channel is corrupted by the CV AWGN $n(k)$ with the variance σ_n^2 per dimension.

Given the training data $\{\boldsymbol{x}(k), y(k)\}_{k=1}^K$, the Hammerstein channel model $\hat{\Psi}(\cdot)$ and $\hat{H}(z)$ can be identified accurately and efficiently using the identification method of Section II-C. Since the distributions of $x_R(k)$ and $x_I(k)$ are symmetric, the distributions of $w_R(k)$ and $w_I(k)$ are also symmetric. Furthermore

$$X_{\max} = \max\big\{x_R(k)\big\} = \max\big\{x_I(k)\big\} = \big(\sqrt{M}-1\big)d \tag{51}$$

is known. Therefore, the two knot sequences (4) and (8) can be chosen to be identical with $U_{\max} = V_{\max} = X_{\max}$, $U_{\min} = V_{\min} = -X_{\max}$, and $N_R = N_I = \sqrt{N}$. In practice, $P_o = 4$ is sufficient, and an appropriate value of $\sqrt{N}$ can be chosen by trail and error. In particular, the number of internal knots should be sufficient to provide a good modeling capability but should not be too large in order to avoid overfitting and unnecessary computational costs. Since $X_{\max}$ depends on the operating condition of the HPA, namely, the value of IBO, the knot sequence should be empirically tuned according to the operating condition of the HPA.

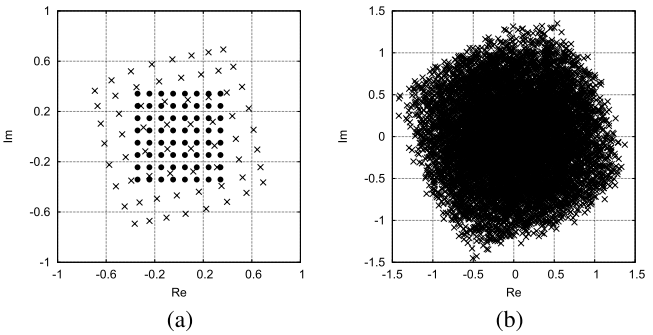


Fig. 4. Case of IBO = 10 dB with the Hammerstein channel's input $x(k)$ marked by $\cdot$. (a) HPAs output $w(k)$ marked by $\times$. (b) Noise-free Hammerstein channel's output $\tilde{y}(k)$ marked by $\times$.

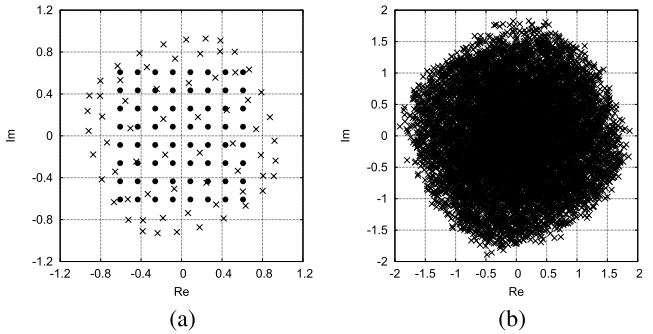


Fig. 5. Case of IBO = 5 dB with the Hammerstein channel's input $x(k)$ marked by $\cdot$. (a) HPAs output $w(k)$ marked by $\times$. (b) Noise-free Hammerstein channel's output $\tilde{y}(k)$ marked by $\times$.

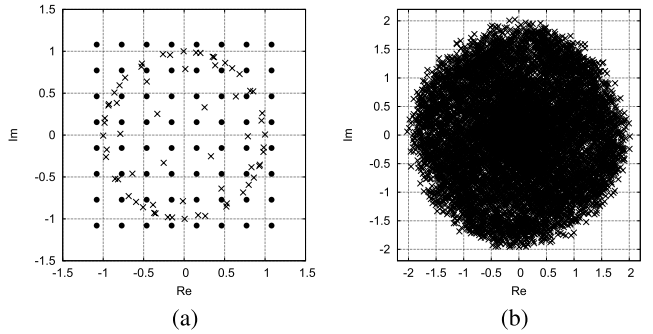


Fig. 6. Case of IBO = 0 dB with the Hammerstein channel's input $x(k)$ marked by $\cdot$. (a) HPAs output $w(k)$ marked by $\times$. (b) Noise-free Hammerstein channel's output $\tilde{y}(k)$ marked by $\times$.

TABLE I
EMPIRICALLY DETERMINED KNOT SEQUENCE

IBO	Knot sequence
10 dB	-20.0,-1.5,-0.5, **-0.2**, -0.1,-0.05,0.0,0.05,0.1, **0.2**, 0.5,1.5,20.0
5 dB	-20.0,-2.0,-0.6, **-0.4**, -0.2,-0.1,0.0,0.1,0.2, **0.4**, 0.6,2.0,20.0
0 dB	-20.0,-3.0,-1.0, **-0.7**, -0.4,-0.2,0.0,0.2,0.4, **0.7**, 1.0,3.0,20.0

Remark 1: Note that the input signal $x(k)$ takes the value from the symbol set $\mathbb{S}$, which consists of the M discrete complex values. Under normal operating conditions of the HPA, namely, the value of IBO is large, $r_{x_{\max}} < r_{\text{sat}}$, where

TABLE II
IDENTIFICATION RESULTS FOR THE LINEAR FILTER PART, $\mathbf{h}$, OF THE HAMMERSTEIN CHANNEL

True parameter vector:		$\mathbf{h}^{\mathrm{T}} = [0.75000 + \mathrm{j}0.20000\ 0.15000 + \mathrm{j}0.10000\ 0.08000 + \mathrm{j}0.00100]$
Estimate under	IBO= 10 dB and $E_{\mathrm{b}}/N_{\mathrm{o}} = 30.0$ dB:	$\widehat{\mathbf{h}}^{\mathrm{T}} = [0.75027 + \mathrm{j}0.20015\ 0.14981 + \mathrm{j}0.09995\ 0.08002 + \mathrm{j}0.00106]$
	IBO= 10 dB and $E_{\mathrm{b}}/N_{\mathrm{o}} = 20.0$ dB:	$\widehat{\mathbf{h}}^{\mathrm{T}} = [0.75085 + \mathrm{j}0.20047\ 0.14941 + \mathrm{j}0.09985\ 0.08005 + \mathrm{j}0.00119]$
	IBO= 10 dB and $E_{\mathrm{b}}/N_{\mathrm{o}} = 10.0$ dB:	$\widehat{\mathbf{h}}^{\mathrm{T}} = [0.75268 + \mathrm{j}0.20151\ 0.14812 + \mathrm{j}0.09952\ 0.08013 + \mathrm{j}0.00161]$
Estimate under	IBO= 5 dB and $E_{\mathrm{b}}/N_{\mathrm{o}} = 30.0$ dB:	$\widehat{\mathbf{h}}^{\mathrm{T}} = [0.75038 + \mathrm{j}0.20017\ 0.14977 + \mathrm{j}0.09998\ 0.08002 + \mathrm{j}0.00109]$
	IBO= 5 dB and $E_{\mathrm{b}}/N_{\mathrm{o}} = 20.0$ dB:	$\widehat{\mathbf{h}}^{\mathrm{T}} = [0.75118 + \mathrm{j}0.20054\ 0.14928 + \mathrm{j}0.09992\ 0.08006 + \mathrm{j}0.00128]$
	IBO= 5 dB and $E_{\mathrm{b}}/N_{\mathrm{o}} = 10.0$ dB:	$\widehat{\mathbf{h}}^{\mathrm{T}} = [0.75372 + \mathrm{j}0.20173\ 0.14772 + \mathrm{j}0.09975\ 0.08018 + \mathrm{j}0.00190]$
Estimate under	IBO= 0 dB and $E_{\mathrm{b}}/N_{\mathrm{o}} = 30.0$ dB:	$\widehat{\mathbf{h}}^{\mathrm{T}} = [0.75063 + \mathrm{j}0.20024\ 0.14968 + \mathrm{j}0.10000\ 0.08005 + \mathrm{j}0.00118]$
	IBO= 0 dB and $E_{\mathrm{b}}/N_{\mathrm{o}} = 20.0$ dB:	$\widehat{\mathbf{h}}^{\mathrm{T}} = [0.75198 + \mathrm{j}0.20075\ 0.14898 + \mathrm{j}0.09999\ 0.08016 + \mathrm{j}0.00156]$
	IBO= 0 dB and $E_{\mathrm{b}}/N_{\mathrm{o}} = 10.0$ dB:	$\widehat{\mathbf{h}}^{\mathrm{T}} = [0.75622 + \mathrm{j}0.20252\ 0.14678 + \mathrm{j}0.09997\ 0.08045 + \mathrm{j}0.00280]$

the maximum input amplitude $r_{x_{\max}}$ is defined by

$$r_{x_{\max}} = \sqrt{2}\left(\sqrt{M} - 1\right)d = \max_{x \in \mathbb{S}} |x|. \tag{52}$$

This means that there is no input signal in the input saturation region of $r_x \geq r_{\mathrm{sat}}$ or near r_{sat}. In other words, under normal operating conditions of the HPA, the system is poorly excited for the identification task. Since there is no training data near and in the saturation region of the HPA, we have to purely rely on the extrapolation ability of the CV B-spline neural network to capture the nonlinear characteristics $\Psi(\cdot)$ of the HPA near and in the saturation region.

Given the identified $\hat{H}(z)$, the coefficient vector $\boldsymbol{g}_{\mathrm{MMSE}}$ of the linear filter in the Wiener equalizer can be calculated and stored for the use in online nonlinear equalization operation during data communication. Based on the estimated $\hat{\Psi}(\cdot) = \hat{\Psi}_R(\cdot) + \mathrm{j}\hat{\Psi}_I(\cdot)$, an accurate inversion to $\Psi(\cdot) = \Psi_R(\cdot) + \mathrm{j}\Psi_I(\cdot)$ can readily be obtained. Note that under normal operating conditions of large IBO values and over the input range, $\Psi_R(\cdot)$ and $\Psi_I(\cdot)$ are monotonic. Since $\hat{\Psi}(\cdot)$ is an accurate estimate of $\Psi(\cdot)$, $\hat{\Psi}_R(\cdot)$ and $\hat{\Psi}_I(\cdot)$ can also be assumed to be monotonic over the input range. Therefore, provided that the AWGN $n(k)$ in the Hammerstein channel output $y(k)$ is not extremely large, the Gauss–Newton method of Section II-D based on the inversion of De Boor algorithm converges to the unique solution $\hat{\Psi}^{-1}(\cdot)$. In particular, define the input vector to the Wiener equalizer at symbol index k as

$$\boldsymbol{y}(k) = [y(k)\ y(k-1)\ y(k-L_g)]^{\mathrm{T}} \tag{53}$$

and the output of the linear filter $G(z)$ as

$$\hat{w}(k-\iota) = \boldsymbol{g}_{\mathrm{MMSE}}^{\mathrm{H}} \boldsymbol{y}(k) \tag{54}$$

where ι is also known as the decision delay. Given $\hat{w}(k-\iota)$, the algorithm of Section II-D then calculates

$$\hat{x}(k-\iota) = \hat{\Psi}^{-1}\left(\hat{w}(k-\iota)\right) \tag{55}$$

which after quantisation provides the estimate or detected symbol for $x(k-\iota)$.

Remark 2: If the HPA is operating in the highly saturated region, namely, the IBO value is small, symbol points at the edge of the constellation (43) may be distorted by the HPA into the output points $w(k)$ with the same amplitude $A_{\max}$ and only slightly differences in phase. In other words, these $w(k)$ are extremely close. As the noise in the equalizer input $\boldsymbol{y}(k)$ will alter the phase of the estimate $\hat{w}(k-\iota)$, the estimate (54) may not always be related to the true transmitted signal $w(k-\iota)$. Thus, if the HPA is operating in the saturation region and the AWGN $n(k)$ is large, which is further amplified by the inverse linear filter (54), $\hat{\Psi}_R(\cdot)$ may not always be guaranteed to be a one to one mapping over the input range, and the Gauss–Newton method may not always be guaranteed to converge to the unique solution. This will increase the error probability of the decision (55), and will degrade the equalization accuracy or the achievable system's bit error rate (BER).

C. Simulation Example

We considered the 64-QAM Hammerstein channel with the static nonlinearity described by the HPA (45) and (46) and an FIR filter of order $L = 3$. The parameters of this Hammerstein channel were given as

$$\begin{aligned} \mathbf{t}^{\mathrm{T}} &= [2.0\ 1.0\ 4.0\ 9.0], \\ \mathbf{h}^{\mathrm{T}} &= [0.75 + \mathrm{j}0.2\ 0.15 + \mathrm{j}0.1\ 0.08 + \mathrm{j}0.01]. \end{aligned} \tag{56}$$

The serious nonlinear and memory distortions caused by this Hammerstein channel are illustrated in Figs. 4–6. It is clear that even in the absence of the channel's AWGN $n(k)$, the output of this Hammerstein channel is so seriously distorted that the achievable BER is practically 50%. Note that, for the case of IBO = 10 dB, the HPA is operating in the normal operating region with $r_{x_{\max}} < r_{\mathrm{sat}}$, and for the case of IBO = 5 dB, the HPA is operating near the saturation region with $r_{x_{\max}} \approx r_{\mathrm{sat}}$, whereas for the case of IBO = 0 dB, the HPA is operating well inside the saturation region with $r_{x_{\max}} \gg r_{\mathrm{sat}}$. Since $H(z)$ in this Hammerstein channel is minimum phase, the equalizer's decision delay was set to $\iota = 0$.

1) Results of Hammerstein Channel Identification: The 64-QAM training sets each containing $K = 1000$ data samples were generated given the Hammerstein channel's parameters (56) and with various values of the HPAs IBO and the channel's SNR, respectively. The piecewise cubic polynomial ($P_o = 4$) was chosen as the B-spline basis function, and the number of B-spline basis functions was set to $\sqrt{N} = 8$. The empirically determined knot sequences for different HPA operating conditions are listed in Table I. The alternating LS estimation algorithm of Section II-C was used to identify this Hammerstein channel, in particular, to provide the estimate of the channel's linear filter coefficient vector $\hat{\boldsymbol{h}}$ and the B-spline neural network's weight vector $\hat{\boldsymbol{\omega}}$. It was observed that no more than four iterations were sufficient for the algorithm to converge. The results obtained are summarized in Table II as well as illustrated in Figs. 7–9.

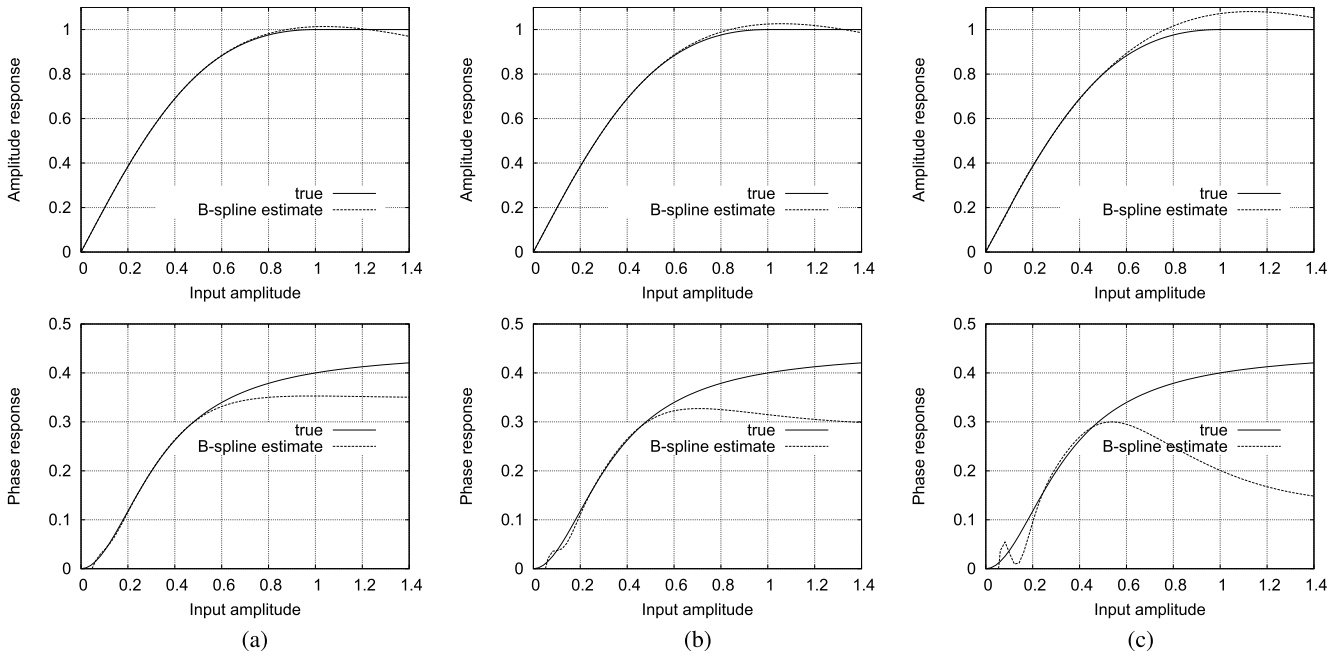


Fig. 7. Comparison of the HPAs static nonlinearity $\Psi(\cdot)$ and the estimated static nonlinearity $\hat{\Psi}(\cdot)$ under IBO $=$ 10 dB. (a) $E_b/N_o = 30$ dB. (b) $E_b/N_o = 20$ dB. (c) $E_b/N_o = 10$ dB.

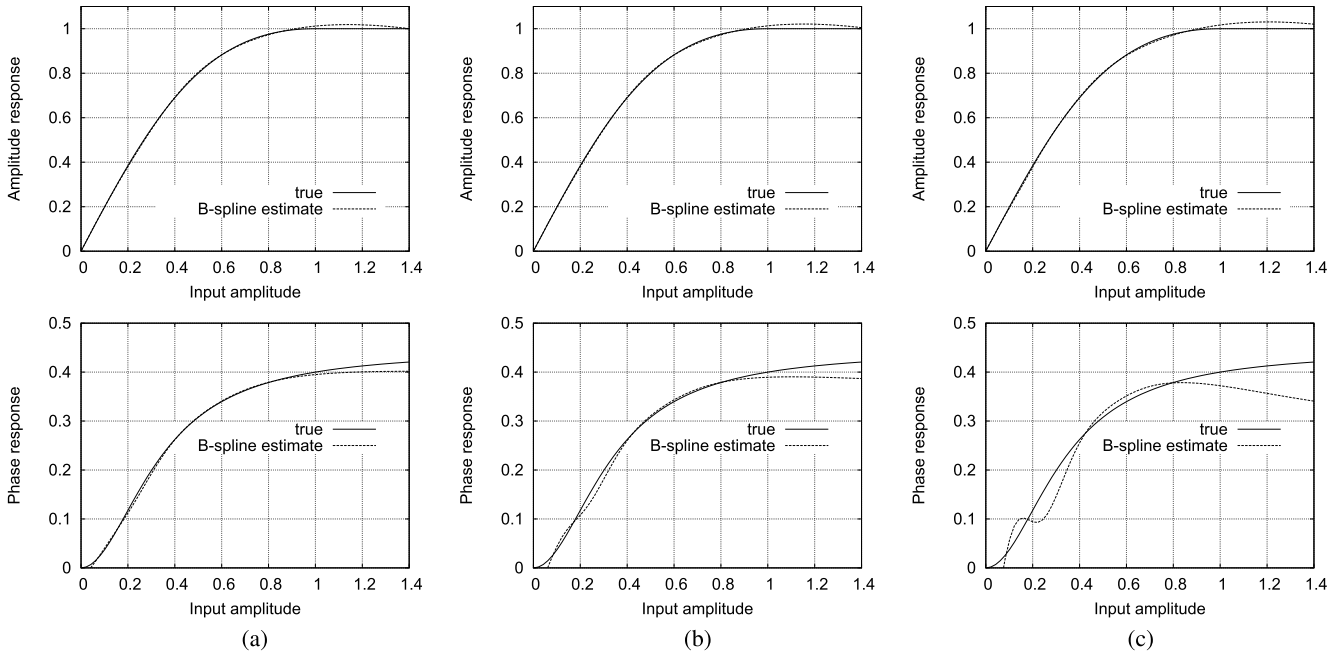


Fig. 8. Comparison of the HPAs static nonlinearity $\Psi(\cdot)$ and the estimated static nonlinearity $\hat{\Psi}(\cdot)$ under IBO $=$ 5 dB. (a) $E_b/N_o = 30$ dB. (b) $E_b/N_o = 20$ dB. (c) $E_b/N_o = 10$ dB.

Observe from Table II that the identification of the linear subsystem in the Hammerstein channel was achieved with high accuracy across wide range of the IBO values as well as under the high channel's AWGN condition. In order to achieve an accurate identification of a nonlinear system, the nonlinear system should be sufficiently excited over all the amplitudes concerned by the input signal, which is known as the persistent excitation condition. Note that, under the identification condition of IBO $=$ 10 dB, there were no data points near or in the HPAs saturation region. Consequently, the amplitude response and phase response of the estimated B-spline neural network $\hat{\Psi}(\cdot)$ exhibit noticeable deviation from the HPAs true amplitude response $A(r_x)$ and true phase response $\Upsilon(r_x)$ in the region $r_x \geq r_{\text{sat}}$, particularly under a high noise condition, as shown in Fig. 7. This of course does not matter, as this region is well beyond the given operating region of the HPA. Interestingly, as the IBO value reduced, the estimated HPA response inside the region of $r_x \geq r_{\text{sat}}$ became much more accurate, as can be noted from Figs. 8 and 9, because of the better excitation of the input signal. From Figs. 7–9, it can

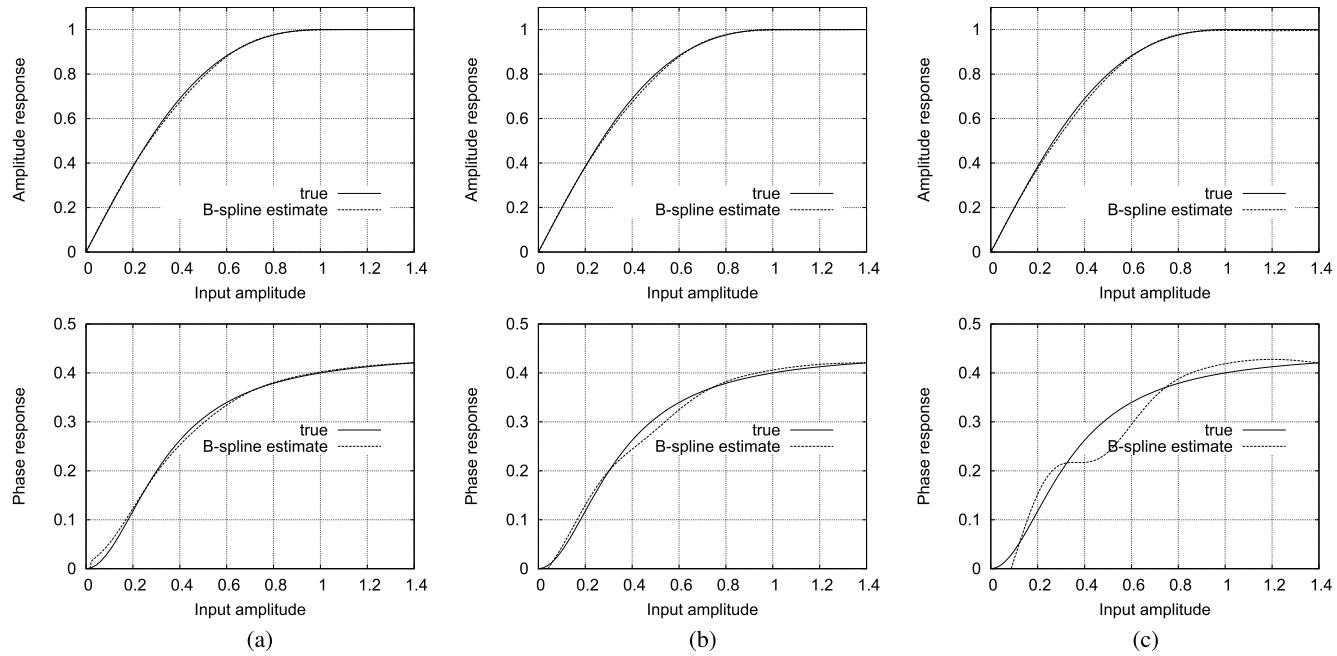


Fig. 9. Comparison of the HPAs static nonlinearity $\Psi(\cdot)$ and the estimated static nonlinearity $\hat{\Psi}(\cdot)$ under IBO = 0 dB. (a) $E_b/N_o = 30$ dB. (b) $E_b/N_o = 20$ dB. (c) $E_b/N_o = 10$ dB.

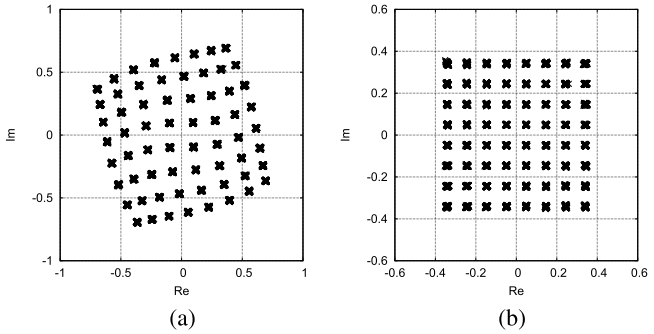


Fig. 10. Case of IBO = 10 dB and $E_b/N_o = 30$ dB. (a) HPAs output $w(k)$ is marked by $\cdot$, whereas the Wiener equalizer's linear filter output $\hat{w}(k)$ is marked by $\times$. (b) Hammerstein channel's input $x(k)$ is marked by $\cdot$, whereas the Wiener equalizer's output $\hat{x}(k)$ is marked by $\times$.

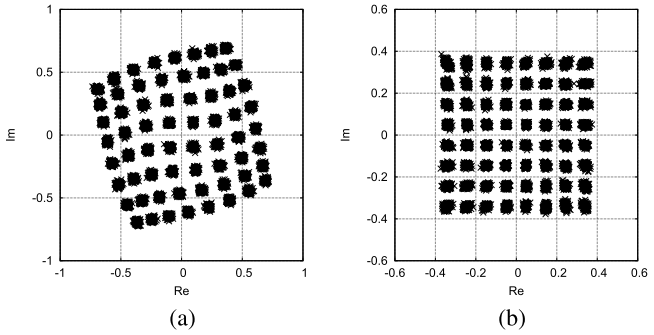


Fig. 11. Case of IBO = 10 dB and $E_b/N_o = 20$ dB. (a) HPAs output $w(k)$ is marked by $\cdot$, whereas the Wiener equalizer's linear filter output $\hat{w}(k)$ is marked by $\times$. (b) Hammerstein channel's input $x(k)$ is marked by $\cdot$, whereas the Wiener equalizer's output $\hat{x}(k)$ is marked by $\times$.

also be seen that the channel's AWGN $n(k)$ has more serious influences on the accuracy of the estimated phase response.

2) Results of Nonlinear Equalization: We employed the estimated Hammerstein channel model to design the Wiener

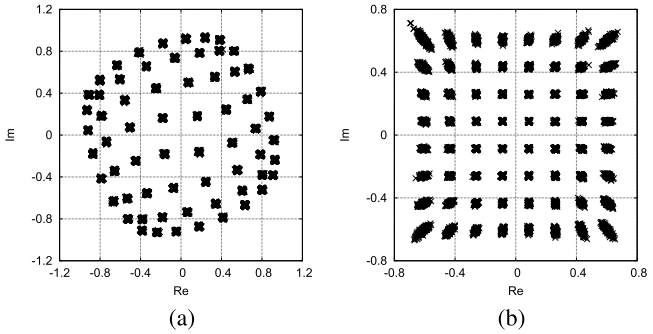


Fig. 12. Case of IBO = 5 dB and $E_b/N_o = 30$ dB. (a) HPAs output $w(k)$ is marked by $\cdot$, whereas the Wiener equalizer's linear filter output $\hat{w}(k)$ is marked by $\times$. (b) Hammerstein channel's input $x(k)$ is marked by $\cdot$, whereas the Wiener equalizer's output $\hat{x}(k)$ is marked by $\times$.

equalizer. The length of the nonlinear equalizer's inverse filter was set to $L_g = 12$, and the inverse linear filter's coefficient vector was set to the MMSE solution (35). The nonlinear static mapping $\hat{\Psi}^{-1}(\cdot)$ of the Wiener equalizer was calculated using the Gauss–Newton algorithm of Section II-D. For the case of IBO = 10 dB and $E_b/N_o = 30$ dB, the output $\hat{w}(k)$ of the Wiener equalizer's inverse linear filter is compared with the HPAs output $w(k)$ in Fig. 10(a), whereas the Wiener equalizer's output $\hat{x}(k)$ is compared with the Hammerstein channel's input $x(k)$ in Fig. 10(b). Compared with the Hammerstein channel's output $\tilde{y}(K)$ depicted in Fig. 4(b), it can be seen that the nonlinear equalizer designed based on the estimated Hammerstein channel model successfully removes the serious distortions caused by the Hammerstein channel. The equalization results for the case of IBO = 10 dB and $E_b/N_o = 20$ dB are illustrated in Fig. 11, where it can be seen again that the nonlinear equalizer successfully removes the serious nonlinear and memory distortions caused by the Hammerstein channel. In addition, from Fig. 11, the noise

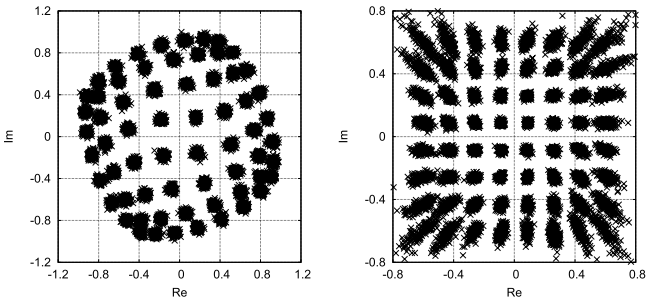


Fig. 13. Case of IBO = 5 dB and $E_b/N_o = 20$ dB. (a) HPAs output $w(k)$ is marked by ·, whereas the Wiener equalizer's linear filter output $\hat{w}(k)$ is marked by ×. (b) Hammerstein channel's input $x(k)$ is marked by ·, whereas the Wiener equalizer's output $\hat{x}(k)$ is marked by ×.

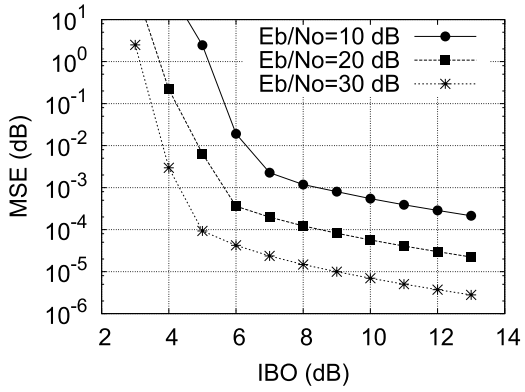


Fig. 14. MSE versus IBO performance.

enhancement phenomenon of the postinversion becomes evident. To investigate the achievable equalization performance when the HPA operates near the saturation region, we next set the IBO value to 5 dB. Figs. 12 and 13 depict the equalization results for $E_b/N_o = 30$ and 20 dB, respectively. We note that the residual nonlinear distortion coupled with the noise enhancement degrades the achievable equalization performance, in comparison with the case that the HPA operates in the normal operating region of $r_{x_{\max}} < r_{\text{sat}}$. Nevertheless, comparing Fig. 13(b) with Fig. 5(b), it can be seen that most of the nonlinear and memory distortions of the Hammerstein channel have been removed by the nonlinear equalizer.

The achievable performance of the nonlinear equalizer designed based on the estimated Hammerstein channel model was further assessed using the MSE metric defined by

$$\text{MSE} = \frac{1}{K_{\text{test}}} \sum_{k=1}^{K_{\text{test}}} |x(k) - \hat{x}(k)|^2 \tag{57}$$

and the system's BER, where K_{test} was the number of test data. With $K_{\text{test}} = 10^5$, the resulting MSE as the function of IBO is plotted in Fig. 14 for the three values of the channel SNR. The achievable BER performance of the nonlinear equalizer are plotted in Fig. 15 for the range of IBO values varying from 10 to 4 dB, in comparison with the benchmark BER curve of the ideal AWGN channel, which is defined by

$$y(k) = c_{\text{gain}} x(k) + n(k) \tag{58}$$

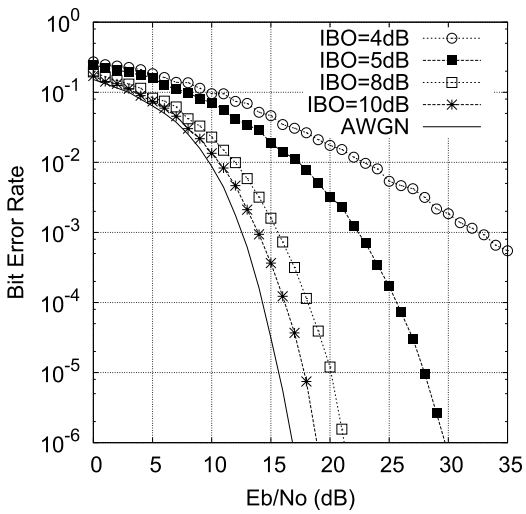


Fig. 15. BER versus channel SNR performance.

with the channel gain $c_{\text{gain}} = \sqrt{1 + \|\mathbf{h}\|^2}$. Note that the channel gain of this ideal AWGN channel was set to equal to the channel gain of the Hammerstein channel. For the Hammerstein channel with its HPA operating at IBO = 10 dB, the nonlinear equalizer based on the estimated Hammerstein channel model is capable of removing all the nonlinear and memory distortions of the Hammerstein channel to achieve a BER performance that is very close to the ideal AWGN channel case. The small degradation from this ideal AWGN channel lower bound is owing to the inevitable noise enhancement of the postinverse. From Fig. 15, it can also be seen that, as the HPA operates closer to the saturation region, the residual nonlinear distortions coupled with the noise enhancement will cause notable degradations from the ideal AWGN lower bound.

IV. Conclusion

Identification and inversion of the CV Hammerstein systems have been proposed based on the CV B-spline neural network approach. Our contribution is twofold. First, the CV nonlinear static function in the Hammerstein system is modeled based on the tensor product from two univariate B-spline neural networks that are constructed using the real and imaginary parts of the system input. A highly efficient alternating least squares estimation method has been proposed to estimate the model parameters that include the CV linear dynamic model coefficients and B-spline neural network weights, in the unique closed-form solution. Second, an accurate postinverse technique has been developed for the CV Hammerstein model. In particular, the inversion of the CV nonlinear static function in the Hammerstein system is calculated efficiently using the Gaussian–Newton algorithm based on the estimated B-spline neural network model with the aid of De Boor algorithm that utilizes naturally both the B-spline curve and first-order derivative recursions. The effectiveness of our approach for modeling and inverting the CV Hammerstein systems is demonstrated with the application to identification and equalization of the Hammerstein channel that consists of the nonlinear HPA at the transmitter and the dispersive transmission medium. This application is particularly challenging, because the identifica-

tion task suffers inherently from poor excitation and the postinversion or equalization causes inevitable noise enhancement. Despite of these difficulties, our proposed approach works well in this challenging application.

REFERENCES

[1] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued radial basis function network, Part I: Network architecture and learning algorithms," *Signal Process.*, vol. 35, no. 1, pp. 19–31, Jan. 1994.

[2] S. Chen, S. McLaughlin, and B. Mulgrew, "Complex-valued radial basis function network, Part II: Application to digital communications channel equalisation," *Signal Process.*, vol. 36, no. 2, pp. 175–188, Mar. 1994.

[3] A. Uncini, L. Vecci, P. Campolucci, and F. Piazza, "Complex valued neural networks with adaptive spline activation function for digital radio links nonlinear equalization," *IEEE Trans. Signal Process.*, vol. 47, no. 2, pp. 505–514, Feb. 1999.

[4] T. Kim and T. Adali, "Approximation by fully complex multilayer perceptrons," *Neural Comput.*, vol. 15, no. 7, pp. 1641–1666, Jul. 2003.

[5] C.-C. Yang and N. K. Bose, "Landmine detection and classification with complex-valued hybrid neural network using scattering parameters dataset," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 743–753, May 2005.

[6] M. B. Li, G. B. Guang, P. Saratchandran, and N. Sundararajan, "Fully complex extreme learning machine," *Neurocomputing*, vol. 68, pp. 306–314, Oct. 2005.

[7] A. Hirose, *Complex Valued Neural Networks*. New York, NY, USA: Springer-Verlag, 2006.

[8] S. Chen, X. Hong, C. J. Harris, and L. Hanzo, "Fully complex-valued radial basis function networks: Orthogonal least squares regression and classification," *Neurocomputing*, vol. 71, nos. 16–18, pp. 3421–3433, Oct. 2008.

[9] T. Nitta, *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*. Hershey, PA, USA: Inf. Sci. Reference, 2009.

[10] A. S. Gangal, P. K. Kalra, and D. S. Chauhan, "Inversion of complex valued neural networks using complex back-propagation algorithm," *Int. J. Math. Comput. Simul.*, vol. 3, no. 1, pp. 1–8, 2009.

[11] M. Kobayashi, "Exceptional reducibility of complex-valued neural networks," *IEEE Trans. Neural Netw.*, vol. 21, no. 7, pp. 1060–1072, Jul. 2010.

[12] A. Hirose, *Complex-Valued Neural Networks: Advances and Applications*. New York, NY, USA: Wiley, 2012.

[13] S. A. Billings, "Identification of nonlinear systems—A survey," *IEE Proc. D, Control Theory Appl.*, vol. 127, no. 6, pp. 272–285, Nov. 1980.

[14] I. W. Hunter and M. J. Korenberg, "The identification of nonlinear biological systems: Wiener and Hammerstein cascade models," *Biol. Cybern.*, vol. 55, nos. 2–3, pp. 135–144, 1986.

[15] E. W. Bai, "An optimal two-stage identification algorithm for Hammerstein-Wiener nonlinear systems," *Automatica*, vol. 34, no. 3, pp. 333–338, Mar. 1998.

[16] Y. Zhu, "Estimation of an N-L-N Hammerstein-Wiener model," *Automatica*, vol. 38, no. 9, pp. 1607–1614, Sep. 2002.

[17] J. Schoukens, J. G. Nemeth, P. Crama, Y. Rolain, and R. Pintelon, "Fast approximate identification of nonlinear systems," *Automatica*, vol. 39, no. 7, pp. 1267–1274, Jul. 2003.

[18] K. Hsu, T. Vincent, and K. Poolla, "A kernel based approach to structured nonlinear system identification—Part I: Algorithms," in *Proc. 14th IFAC Symp. Syst. Identificat.*, Mar. 2006, pp. 1–6.

[19] K. Hsu, T. Vincent, and K. Poolla, "A kernel based approach to structured nonlinear system identification—Part II: Convergence and consistency," in *Proc. 14th IFAC Symp. Syst. Identificat.*, Mar. 2006, pp. 1–6.

[20] W. Greblicki, "Nonparametric identification of Wiener systems," *IEEE Trans. Inf. Theory*, vol. 38, no. 5, pp. 1487–1493, Sep. 1992.

[21] A. Kalafatis, N. Arifin, L. Wang, and W. R. Cluett, "A new approach to the identification of pH processes based on the Wiener model," *Chem. Eng. Sci.*, vol. 50, no. 23, pp. 3693–3701, Dec. 1995.

[22] A. D. Kalafatis, L. Wang, and W. R. Cluett, "Identification of Wiener-type nonlinear systems in a noisy environment," *Int. J. Control*, vol. 66, no. 7, pp. 923–941, 1997.

[23] Y. Zhu, "Distillation column identification for control using Wiener model," in *Proc. Amer. Control Conf.*, Jun. 1999, pp. 3462–3466.

[24] J. C. Gomez, A. Jutan, and E. Baeyens, "Wiener model identification and predictive control of a pH neutralisation process," *IEE Proc. Control Theory Appl.*, vol. 151, no. 3, pp. 329–338, May 2004.

[25] I. Skrjanc, S. Blazic, and O. E. Agamennoni, "Interval fuzzy modeling applied to Wiener models with uncertainties," *IEEE Trans. Syst., Man Cybern., B*, vol. 35, no. 5, pp. 1092–1095, Oct. 2005.

[26] A. Hagenblad, L. Ljung, and A. Wills, "Maximum likelihood identification of Wiener models," *Automatica*, vol. 44, no. 11, pp. 2697–2705, Nov. 2008.

[27] S. A. Billings and S. Y. Fakhouri, "Non-linear system identification using the Hammerstein model," *Int. J. Syst. Sci.*, vol. 10, no. 5, pp. 567–578, May 1979.

[28] P. Stoica and T. Söderström, "Instrumental-variable methods for identification of Hammerstein systems," *Int. J. Control*, vol. 35, no. 3, pp. 459–476, 1982.

[29] A. Balestrino, A. Landi, M. Ould-Zmirli, and L. Sani, "Automatic nonlinear auto-tuning method for Hammerstein modelling of electrical drives," *IEEE Trans. Ind. Electron.*, vol. 48, no. 3, pp. 645–655, Jun. 2001.

[30] J. Turunen, J. T. Tanttu, and P. Loula, "Hammerstein model for speech coding," *EURASIP J. Appl. Signal Process.*, vol. 2003, pp. 1238–1249, Jan. 2003.

[31] J. Jeraj and V. J. Matthews, "A stable adaptive Hammerstein filter employing partial orthogonalization of the input signals," *IEEE Trans. Signal Process.*, vol. 54, no. 4, pp. 1412–1420, Apr. 2006.

[32] S. W. Su, L. Wang, B. G. Celler, A. V. Savkin, and Y. Guo, "Identification and control for heart rate regulation during treadmill exercise," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 7, pp. 1238–1246, Jul. 2007.

[33] X. Hong and R. J. Mitchell, "Hammerstein model identification algorithm using Bezier-Bernstein approximation," *IET Control Theory Appl.*, vol. 1, no. 1, pp. 1149–1159, Apr. 2007.

[34] X. Hong, R. J. Mitchell, and S. Chen, "Modeling and control of Hammerstein system using B-spline approximation and the inverse of De Boor algorithm," *Int. J. Syst. Sci.*, vol. 43, no. 10, pp. 1976–1984, Oct. 2012.

[35] X. Hong and S. Chen, "Modeling of complex-valued Wiener systems using B-spline neural network," *IEEE Trans. Neural Netw.*, vol. 22, no. 5, pp. 818–825, May 2011.

[36] G. Farin, *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*, 4th ed. San Diego, CA, USA: Academic Press, 1996.

[37] C. De Boor, *A Practical Guide to Splines*. New York, NY, USA: Springer-Verlag, 1978.

[38] T. Kavli, "ASMOD—An algorithm for adaptive spline modelling of observation data," *Int. J. Control*, vol. 58, no. 4, pp. 947–967, 1993.

[39] M. Brown and C. J. Harris, *Neurofuzzy Adaptive Modelling and Control*. Upper Saddle River, NJ, USA: Prentice-Hall, 1994.

[40] C. J. Harris, X. Hong, and Q. Gan, *Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach*. New York, NY, USA: Springer-Verlag, 2002.

[41] Y. Yang, L. Guo, and H. Wang, "Adaptive statistic tracking control based on two-step neural networks with time delays," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 420–429, Mar. 2009.

[42] X. Hong, S. Chen, and C. J. Harris, "Complex-valued B-spline neural networks for modelling and inverse of Wiener systems," in *Complex-Valued Neural Networks: Advances and Applications*, A. Hirose, Ed. Hoboken, NJ, USA: Wiley, 2013, pp. 209–233.

[43] A. A. M. Saleh, "Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers," *IEEE Trans. Commun.*, vol. 29, no. 11, pp. 1715–1720, Nov. 1981.

[44] M. Honkanen and S.-G. Häggman, "New aspects on nonlinear power amplifier modeling in radio communication system simulations," in *Proc. 8th IEEE Int. Symp. Pers., Indoor Mobile Radio Commun.*, Sep. 1997, pp. 844–848.

[45] C. J. Clark, G. Chrisikos, M. S. Muha, A. A. Moulthrop, and C. P. Silva, "Time-domain envelope measurement technique with application to wideband power amplifier modeling," *IEEE Trans. Microw. Theory Tech.*, vol. 46, no. 12, pp. 2531–2540, Dec. 1998.

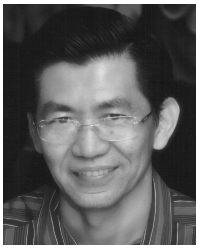
[46] J. H. K. Vuolevi, T. Rahkonen, and J. P. A. Manninen, "Measurement technique for characterizing memory effects in RF power amplifiers," *IEEE Trans. Microw. Theory Tech.*, vol. 49, no. 8, pp. 1383–1389, Aug. 2001.

[47] R. J. Hathaway and J. C. Bezdek, "Grouped coordinate minimization using Newton's method for inexact minimization in one vector coordinate," *J. Optim. Theory Appl.*, vol. 71, no. 3, pp. 503–516, Dec. 1991.

[48] Z. Q. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," *J. Optim. Theory Appl.*, vol. 72, no. 1, pp. 7–35, Jan. 1991.

[49] T. Lyche and J. M. Pena, "Optimally stable multivariate bases," *Adv. Comput. Math.*, vol. 20, nos. 1–3, pp. 149–159, Jan. 2004.

[50] B. Igelnik, "Kolmogorov's spline complex network and adaptive dynamic modeling of data," in *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*, T. Nitta, Ed. New York, NY, USA: Inf. Sci. Ref., 2009, pp. 56–78.
[51] M. Scarpiniti, D. Vigliano, R. Parisi, and A. Uncini, "Flexible blind signal separation in the complex domain," in *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*, T. Nitta, Ed. New York, NY, USA: Inf. Sci. Reference, 2009, pp. 284–323.
[52] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood, NJ, USA: Prentice-Hall, 1991.
[53] L. Hanzo, S. X. Ng, T. Keller, and W. Webb, *Quadrature Amplitude Modulation: From Basics to Adaptive Trellis-Coded, Turbo-Equalised and Space-Time Coded OFDM, CDMA and MC-CDMA Systems*. New York, NY, USA: Wiley, 2004.
[54] L. Hanzo, M. Münster, B. J. Choi, and T. Keller, *OFDM and MC-CDMA for Broadband Multi-User communications, WLANs and Broadcasting*. New York, NY, USA: Wiley, 2003.
[55] J. G. Proakis, *Digital Communications*, 4th ed. New York, NY, USA: McGraw-Hill, 2000.



Sheng Chen (M'90–SM'97–F'08) received the B.Eng. degree from the East China Petroleum Institute, Dongying, China, in 1982, the Ph.D. degree from City University, London, U.K., in 1986, both in control engineering, and the D.Sc. degree from the University of Southampton, Southampton, U.K., in 2005.

He has held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, U.K., from 1986 to 1999. Since 1999, he has been with the Electronics and Computer Science Department, University of Southampton, where he is currently a Professor of intelligent systems and signal processing. He is a Distinguished Adjunct Professor with King Abdulaziz University, Jeddah, Saudi Arabia. He is an ISI Highly Cited Researcher of engineering in 2004. He has published over 500 research papers. His current research interests include adaptive signal processing, wireless communications, modeling and identification of nonlinear systems, neural network and machine learning, intelligent control system design, evolutionary computation methods, and optimization.

Dr. Chen is a fellow of the Institution of Engineering and Technology.

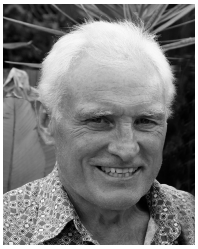
Xia Hong (SM'02) received the B.Sc. and M.Sc. degrees from the National University of Defense Technology, Hunan, China, in 1984 and 1987, respectively, and the Ph.D. degree from the University of Sheffield, Sheffield, U.K., in 1998, all in automatic control.

She was a Research Assistant with the Beijing Institute of Systems Engineering, Beijing, China, from 1987 to 1993. She was a Research Fellow with the Department of Electronics and Computer Science, University of Southampton, Southampton, U.K., from 1997 to 2001. She is currently a Professor with the School of Systems Engineering, University of Reading, Reading, U.K. She has published over 100 research papers, and co-authored a research book. Her current research interests include nonlinear systems identification, data modeling, estimation and intelligent control, neural networks, pattern recognition, learning theory, and their applications.

Prof. Hong received the Donald Julius Groen Prize from IMechE in 1999.

Junbin Gao received the B.Sc. degree in computational mathematics from the Huazhong University of Science and Technology (HUST), China, in 1982, and the Ph.D. degree from the Dalian University of Technology, Dalian, China, in 1991.

He is a Professor of computing science with the School of Computing and Mathematics, Charles Sturt University, New South Wales, Australia. He was a Senior Lecturer and Lecturer of computer science with the University of New England, Madgwick, Australia, from 2001 to 2005. From 1982 to 2001, he was an Associate Lecturer, Lecturer, Associate Professor, and Professor with the Department of Mathematics, HUST. Between 1999 and 2001, he was a Senior Research Fellow with the School of Electronics and Computer Science, University of Southampton, Southampton, U.K. His current research interests include machine learning, data mining, Bayesian learning and inference, and image analysis.



Chris J. Harris received the B.Sc. and M.A. degrees from the University of Leicester, Leicester, U.K., and the University of Oxford, Oxford, U.K., and the Ph.D. and D.Sc. degrees from the University of Southampton, Southampton, U.K., in 1972 and 2001, respectively.

He is an Emeritus Research Professor with the University of Southampton. He has held senior academic appointments at Imperial College, and Oxford and Manchester Universities, and served as the Deputy Chief Scientist for the Government of U.K. He is the co-author of more than 450 scientific research papers.

Prof. Harris received the IEE Senior Achievement Medal for Data Fusion Research and the IEE Faraday Medal for distinguished international research in machine learning. He was elected to the U.K. Royal Academy of Engineering in 1996.