Comparative Performance of Complex-Valued B-Spline and Polynomial Models Applied to Iterative Frequency-Domain Decision Feedback Equalization of Hammerstein Channels

Sheng Chen, *Fellow, IEEE*, Xia Hong, *Senior Member, IEEE*, Emad F. Khalaf, Fuad E. Alsaadi, and Chris J. Harris

***Abstract*—Complex-valued (CV) B-spline neural network approach offers a highly effective means for identifying and inverting practical Hammerstein systems. Compared with its conventional CV polynomial-based counterpart, a CV B-spline neural network has superior performance in identifying and inverting CV Hammerstein systems, while imposing a similar complexity. This paper reviews the optimality of the CV B-spline neural network approach. Advantages of B-spline neural network approach as compared with the polynomial based modeling approach are extensively discussed, and the effectiveness of the CV neural network-based approach is demonstrated in a real-world application. More specifically, we evaluate the comparative performance of the CV B-spline and polynomial-based approaches for the nonlinear iterative frequency-domain decision feedback equalization (NIFDDFE) of single-carrier Hammerstein channels. Our results confirm the superior performance of the CV B-spline-based NIFDDFE over its CV polynomial-based counterpart.**

***Index Terms*—Complex-valued (CV) polynomial model, CV B-spline neural network, identification and inversion of Hammerstein channels, nonlinear iterative frequency-domain decision feedback equalization (NIFDDFE).**

I. Introduction

In many real-world applications, the underlying system that generates complex-valued (CV) signals can be modeled by the CV Hammerstein model. The system is gray-box, as its structure is known to be consisting of an unknown static nonlinearity followed by an unknown linear dynamic model. A well-known example of CV Hammerstein systems is the single-carrier (SC) block transmission communication channel with nonlinear high-power amplifier (HPA) at transmitter, whereby the CV static nonlinearity of the Hammerstein system is constituted by the nonlinear transmit HPA, and its linear dynamic subsystem is the dispersive channel, which can usually be modeled as a finite-duration impulse response (FIR) filter. Effective identification and inversion of CV Hammerstein systems is, therefore, crucial in these practical applications.

The CV B-spline neural network has widely been used as an effective means for identification and inversion of CV Hammerstein systems [1]–[3]. Compared with its conventional polynomial-based counterpart, B-spline models are proved to have the optimal stability or numerical robustness [4]–[6], and achieve superior performance in challenging practical applications [1]–[3], while maintaining a similar computational complexity. In this paper, we review the CV B-spline neural network model as an effective means for identifying and inverting practical Hammerstein systems. In particular, we analyze its optimal robustness property and provide the computational complexity required for calculating the output of a B-spline model, which turns out to be slightly higher than that of the conventional polynomial model, and the both the models have the same order of complexity.

Our main contribution is, however, the derivation of a new CV B-spline neural network-based design for the nonlinear iterative frequency-domain decision feedback equalization (NIFDDFE) of SC Hammerstein communication systems. Effective identification and inverting algorithms are provided for the SC Hammerstein channel based on the CV B-spline neural network approach. We use this challenging real-world application to evaluate the comparative performance of the CV B-spline neural network-based NIFDDFE and its CV polynomial-based NIFDDFE counterpart. The results obtained clearly demonstrate that our B-spline-based NIFDDFE has a superior performance over the polynomial-based NIFDDFE. Our novel application, therefore, reinforces the CV B-spline neural network as a versatile and effective means for solving real-world applications where the underlying systems can be represented by CV Hammerstein models.

Manuscript received January 4, 2016; revised July 15, 2016; accepted September 12, 2016. Date of publication September 23, 2016; date of current version November 15, 2017. This work was supported by the Deanship of Scientific Research, King Abdulaziz University.

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

X. Hong is with the Department of Computer Science, School of Mathematical, Physical, and Computational Sciences, University of Reading, Reading, RG6 6AY, U.K. (e-mail: x.hong@reading.ac.uk).

E. Khalaf and F. E. Alsaadi are with the Electrical and Computer Engineering Department, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia (e-mail: ekhalaf@kau.edu.sa; fuad_alsaadi@yahoo.com).

C. J. Harris is with the Department of Electronics and Computer Science, University of Southampton, Southampton, SO17 1BJ, U.K.

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.2016.2609001

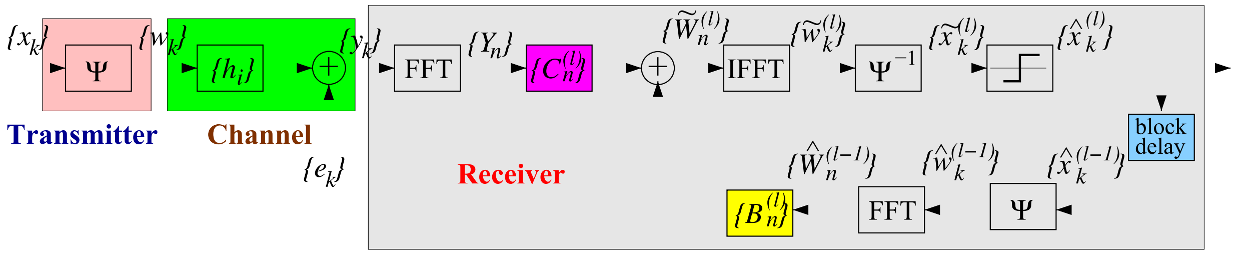


Fig. 1. System schematic of the NIFDDFE for SC Hammerstein communication systems with the nonlinear HPA Ψ at transmitter.

Throughout this contribution, a CV number $x \in \mathbb{C}$ is represented either by $x = x_R + \mathrm{j}x_I$ or by $x = |x| \exp(\mathrm{j}\angle^{x})$. The transpose and conjugate transpose operators are denoted by $(\ )^T$ and $(\ )^H$, respectively, while $(\ )^{-1}$ stands for the inverse operation and $(\ )^*$ denotes the conjugate operation. Furthermore, the expectation operator is denoted by $E\{\ \}$.

II. NIFDDFE for Hammerstein Channels

To illustrate the necessity for identifying and inverting CV Hammerstein systems, we begin by introducing our challenging application scenario, the SC block transmission communication system [7]–[10], where each transmit block consists of N data symbols with M-quadrature amplitude modulation (QAM) expressed as

$$\boldsymbol{x} = [x_0\ x_1 \cdots x_{N-1}]^T \tag{1}$$

where x_k, $0 \le k \le N-1$, takes the values from the M-QAM symbol set

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

with $2d$ denoting the minimum distance between symbol points. Adding the cyclic prefix (CP) of length N_{cp} to $\boldsymbol{x}$ yields

$$\bar{\boldsymbol{x}} = [x_{-N_{\text{cp}}}\ x_{-N_{\text{cp}}+1} \cdots x_{-1} | \boldsymbol{x}^T]^T \tag{3}$$

with $x_{-k} = x_{N-k}$ for $1 \le k \le N_{\text{cp}}$. The signal block $\bar{\boldsymbol{x}}$ is amplified by the HPA to yield the transmitted signal block

$$\bar{\boldsymbol{w}} = [w_{-N_{\text{cp}}}\ w_{-N_{\text{cp}}+1} \cdots w_{-1} | \boldsymbol{w}^T]^T \tag{4}$$

where $\boldsymbol{w} = [w_0\ w_1 \cdots w_{N-1}]^T$ and

$$w_k = \Psi(x_k),\ -N_{\text{cp}} \le k \le N-1 \tag{5}$$

in which $\Psi(\ )$ represents the CV static nonlinearity of HPA and $w_{-k} = w_{N-k}$ for $1 \le k \le N_{\text{cp}}$. Typical HPA in the transmitter is the solid-state power amplifier [11]–[13], whose nonlinearity $\Psi(\ )$ is constituted by the HPA's amplitude response $A(r)$ and phase response $\Upsilon(r)$ given by

$$A(r) = \frac{g_a r}{\left(1 + \left(\frac{g_a r}{A_{\text{sat}}}\right)^{2\beta_a}\right)^{\frac{1}{2\beta_a}}} \tag{6}$$

$$\Upsilon(r) = \frac{\alpha_\phi r^{q_1}}{1 + \left(\frac{r}{\beta_\phi}\right)^{q_2}} \tag{7}$$

where r denotes the amplitude of the input to HPA, g_a is the small gain signal, β_a is the smoothness factor, and A_{sat} is the saturation level, while the phase response parameters α_ϕ, β_ϕ, q_1, and q_2 are adjusted to match the specific amplifier's characteristics. We adopt the following parameter set defined in the standardization [12], [13]:

$$\begin{aligned} g_a &= 19,\ \beta_a = 0.81,\ A_{\text{sat}} = 1.4 \\ \alpha_\phi &= -48000,\ \beta_\phi = 0.123,\ q_1 = 3.8,\ q_2 = 3.7. \end{aligned} \tag{8}$$

Given the input $x_k = |x_k| e^{\mathrm{j}\angle^{x_k}}$, the output of the HPA is

$$w_k = A(|x_k|) e^{\mathrm{j}(\angle^{x_k} + \Upsilon(|x_k|))}. \tag{9}$$

The operating status of the HPA is specified by the output back-off (OBO), which is defined as the ratio of the maximum output power $P_{\max}$ of the HPA to the average output power P_{aop} of the HPA output signal, given by

$$\text{OBO} = 10 \cdot \log_{10} \frac{P_{\max}}{P_{\text{aop}}}. \tag{10}$$

The smaller the OBO is, the more the HPA is operating into the nonlinear saturation region.

The amplified signal block $\bar{\boldsymbol{w}}$ is transmitted through the channel whose channel impulse response (CIR) coefficient vector is

$$\boldsymbol{h} = [h_0\ h_1 \ldots h_{L_{\text{cir}}}]^T \tag{11}$$

where L_{cir} denotes the CIR length. Note that the CP must be chosen to be $N_{\text{cp}} \ge L_{\text{cir}}$. We can always assume that $h_0 = 1$, because if this is not the case, h_0 can be absorbed into the CV nonlinearity $\Psi(\ )$, and the CIR coefficients are rescaled as h_i/h_0 for $0 \le i \le L_{\text{cir}}$. The combined transmission channel and transmitter, as shown in Fig. 1, is a Hammerstein system containing the nonlinearity $\Psi(\ )$ defined by (6) and (7) followed by the FIR filter with the CIR (11).

At the receiver, after CP removal, the channel-impaired received signals y_k are given by

$$y_k = \sum_{i=0}^{L_{\text{cir}}} h_i w_{k-i} + e_k, \quad 0 \le k \le N-1 \tag{12}$$

in which $w_{k-i} = w_{N+k-i}$ for $k < i$, where e_k is the additive white Gaussian noise (AWGN) with $E\{|e_k|^2\} = 2\sigma_e^2$. Our NIFDDFE receiver is shown in Fig. 1. First, passing $\boldsymbol{y} = [y_0\ y_1 \cdots y_{N-1}]^T$ through the N-point fast Fourier transform (FFT) processor yields the frequency-domain (FD) received signal block $\boldsymbol{Y} = [Y_0\ Y_1 \cdots Y_{N-1}]^T$ with elements

$$Y_n = \sum_{k=0}^{N-1} y_k e^{-\mathrm{j}\frac{2\pi kn}{N}}, \quad 0 \le n \le N-1. \tag{13}$$

Due to the well-known circular property of CP [7]–[10]

$$Y_n = H_n W_n + \Xi_n, \quad 0 \le n \le N-1 \tag{14}$$

in which Ξ_n is the FD representation of the AWGN with $E\{|\Xi_n|^2\} = 2\sigma_e^2$, and $\boldsymbol{W} = [W_0 \ W_1 \cdots W_{N-1}]^T$ is the N-point FFT of $\boldsymbol{w}$, that is

$$W_n = \sum_{k=0}^{N-1} w_k e^{-\mathrm{j}\frac{2\pi kn}{N}}, \quad 0 \le n \le N-1 \tag{15}$$

with $E\{|W_n|^2\} = NE\{|w_k|^2\} = N\sigma_w^2$, while the FD channel transfer function coefficients H_n, $0 \le n \le N-1$, are the N-point FFT of $\boldsymbol{h}$ given by

$$H_n = \sum_{i=0}^{L_{\mathrm{cir}}} h_i e^{-\mathrm{j}\frac{2\pi in}{N}}, \quad 0 \le n \le N-1. \tag{16}$$

Our new NIFDDFE involves an iterative detection procedure with the iteration index $l \ge 1$. Typically, three to four iterations are sufficient. In particular, let the FD feedforward and feedback equalizers coefficients at the lth iteration by $\{C_n^{(l)}\}_{n=0}^{N-1}$ and $\{B_n^{(l)}\}_{n=0}^{N-1}$, respectively. Furthermore, denote the estimate of $\{W_n\}_{n=0}^{N-1}$ at the previous iteration be $\{\widehat{W}_n^{(l-1)}\}_{n=0}^{N-1}$. Then the "soft" estimate of W_n is given by

$$\widetilde{W}_n^{(l)} = C_n^{(l)} Y_n + B_n^{(l)} \widehat{W}_n^{(l-1)}, \quad 0 \le n \le N-1. \tag{17}$$

Passing $\widetilde{W}_n^{(l)}$ for $0 \le n \le N-1$ through the N-point inverse FFT processor yields the soft estimate of the time-domain (TD) transmitted signals $\{w_k\}_{k=0}^{N-1}$ as

$$\widetilde{w}_k^{(l)} = \frac{1}{N}\sum_{n=0}^{N-1} \widetilde{W}_n^{(l)} e^{\mathrm{j}\frac{2\pi nk}{N}}, \quad 0 \le k \le N-1. \tag{18}$$

For the convenience of discussion, assume that the nonlinearity $\Psi(\ )$ of the transmitter HPA and its inversion $\Psi^{-1}(\ )$ are both known at the receiver. The soft estimate $\{\widetilde{x}_k^{(l)}\}_{k=0}^{N-1}$ of the transmitted data symbols can be calculated according to

$$\widetilde{x}_k^{(l)} = \Psi^{-1}(\widetilde{w}_k^{(l)}), \quad 0 \le k \le N-1. \tag{19}$$

By quantizing $\widetilde{x}_k^{(l)}$, we obtain the hard-decision estimate $\{\widehat{x}_k^{(l)}\}_{k=0}^{N-1}$ of the transmitted data block. Further distorting $\{\widehat{x}_k^{(l)}\}_{k=0}^{N-1}$ by $\Psi(\ )$ yields the TD estimate $\{\widehat{w}_k^{(l)}\}_{k=0}^{N-1}$, which is transformed by the N-point FFT to produce the FD estimate $\{\widehat{W}_n^{(l)}\}_{k=0}^{N-1}$ to be used in the next iteration.

If the HPA is linear, and hence $w_k = x_k$, we have the existing linear iterative FD decision feedback equalization (LIFDDFE), for which $\{C_n^{(l)}\}_{n=0}^{N-1}$ and $\{B_n^{(l)}\}_{n=0}^{N-1}$ can be obtained by minimizing the mean square error but the computation is quite involved [8]. Extending this LIFDDFE design to our new NIFDDFE also yields poor performance. However, we find that the extension of the low-complexity simplified LIFDDFE design of [10] to our NIFDDFE works well with some modifications. We now present how to calculate $\{C_n^{(l)}\}_{n=0}^{N-1}$ and $\{B_n^{(l)}\}_{n=0}^{N-1}$ for our new NIFDDFE.

At the first iteration $l = 1$, $\widehat{W}_n^{(0)} = 0$ and $B_n^{(1)} = 0$ for $0 \le n \le N-1$, and we have

$$C_n^{(1)} = \frac{H_n^*}{|H_n|^2 + \frac{2\sigma_e^2}{\sigma_w^2}}, \quad 0 \le n \le N-1 \tag{20}$$

which is identical to the nonlinear FD equalization (NFDE) solution of [3]. For the iterations $l \ge 2$, we have

$$C_n^{(l)} = C_n = \frac{(1-\gamma)H_n^*}{\mathrm{SNR}_{\mathrm{pre}}^{-1} + \beta P_{,\mathrm{pre}}|H_n|^2}, \quad 0 \le n \le N-1 \tag{21}$$

$$B_n^{(l)} = B_n = -(C_n H_n - 1), \quad 0 \le n \le N-1 \tag{22}$$

with

$$\varpi = \frac{1}{N}\sum_{n=0}^{N-1} \frac{|H_n|^2}{\mathrm{SNR}_{\mathrm{pre}}^{-1} + \beta P_{e,\mathrm{pre}}|H_n|^2} \tag{23}$$

$$\gamma = \frac{\varpi}{1+\varpi}. \tag{24}$$

For the LIFDDFE, the work [10] finds that the performance is insensitive to the predefined signal-to-noise ratio (SNR) value $\mathrm{SNR}_{\mathrm{pre}}$ and the predefined symbol error probability $P_{e,\mathrm{pre}}$. In particular, $\mathrm{SNR}_{\mathrm{pre}}^{-1} = 0.1$ and $P_{e,\mathrm{pre}} = 0.1$ yield excellent results. In our NIFDDFE, we also find that $\mathrm{SNR}_{\mathrm{pre}}^{-1} = 0.1$ and $P_{e,\mathrm{pre}} = 0.1$ are appropriate. In the LIFDDFE case, i.e., $w_k = x_k$, β is a parameter depending on the modulation scheme for x_k. In particular, $\beta = 2$, $2/5$, and $2/21$ for 4-QAM, 16-QAM, and 64-QAM, respectively. In our NIFDDFE, w_k is a nonlinearly distorted x_k and the severity of this nonlinear distortion depends on the OBO of the transmitter HPA. Intuitively, β should be smaller than the linear case and how small β is also depends on the value of OBO. For 64-QAM with OBO = 3 dB, we find that $\beta = 0.01$ is appropriate, i.e., ten times smaller than the linear case. With OBO = 5 dB, an appropriate value is $\beta = 0.05$, i.e., only two times smaller than the linear case. This makes sense, as with OBO = 5 dB, the HPA is operating closer to the linear region than the case of OBO = 3 dB. Another modification made is in the feedback coefficients B_n of (22). In the LIFDDFE design [10], $B_n = -(C_n H_n - \gamma)$. But we find that with B_n of (22), the performance is better for the NIFDDFE.

III. CV B-Spline and Polynomial Implementations of NIFDDFE

It can be seen that implementing the NIFDDFE requires to identifying and inverting the Hammerstein channel that consists of the unknown static nonlinearity $\Psi(\ )$ followed by the FIR filter with the unknown CIR vector $\boldsymbol{h}$.

A. CV B-Spline and Polynomial Models for $\Psi(\ )$

1) CV B-Spline Neural Network: The CV B-spline neural network approach [1]–[3] offers an effective means for identifying and inverting this Hammerstein channel. We first point out that $\Psi(\ )$ meets the following conditions.

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

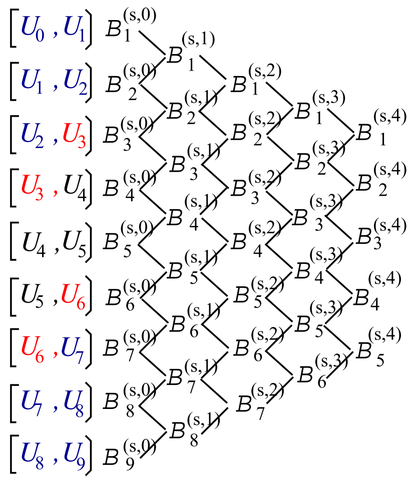


Fig. 2. De Boor recursion: $P_o = 4$, $N_s = 5$, $U_{\min} = U_3$, and $U_{\max} = U_6$.

2) x_R and x_I are upper and lower bounded by some known finite real values, where $x = x_R + \mathrm{j}x_I$ denotes the CV input to $\Psi(\,)$, and the distributions of x_R and x_I are identical.

According to property 2), we have $U_{\min} < x_s < U_{\max}$, where $U_{\min}$ and $U_{\max}$ are the known finite real values, while x_s denotes either x_R or x_I, i.e., the subscript s is either R or I. To use a B-spline neural network for modeling $\Psi(\,)$, a set of N_s univariate B-spline basis functions on x_s is parametrized by the piecewise polynomial degree P_o and a knot sequence of $(N_s + P_o + 1)$ knot values $\{U_0, U_1, \ldots, U_{N_s+P_o}\}$ with

$$U_0 < U_1 < \cdots < U_{P_o-2} < U_{P_o-1} = U_{\min} < U_{P_o} < \cdots < U_{N_s} < U_{N_s+1} = U_{\max} < U_{N_s+2} < \cdots < U_{N_s+P_o}. \tag{25}$$

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 $N_s + 1 - P_o$. Given the set of predetermined knots (25), the set of N_s B-spline basis functions can be formed by using the De Boor recursion [14], yielding for $1 \le l \le N_s + P_o$

$$B_l^{(s,0)}(x_s) = \begin{cases} 1, & \text{if } U_{l-1} \le x_s < U_l \\ 0, & \text{otherwise} \end{cases} \tag{26}$$

as well as for $l = 1, \ldots, N_s + P_o - p$ and $p = 1, \ldots, P_o$

$$B_l^{(s,p)}(x_s) = \frac{x_s - U_{l-1}}{U_{p+l-1} - U_{l-1}} B_l^{(s,p-1)}(x_s) + \frac{U_{p+l} - x_s}{U_{p+l} - U_l} B_{l+1}^{(s,p-1)}(x_s). \tag{27}$$

De Boor recursion is shown in Fig. 2.

Using the tensor product between the two sets of univariate B-spline basis functions [15], $B_l^{(R,P_o)}(x_R)$ for $1 \le l \le N_R$ and $B_m^{(I,P_o)}(x_I)$ 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

$$\widehat{w} = \widehat{\Psi}_B(x) = \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(x)\theta_{l,m}^B = \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_l^{(R,P_o)}(x_R)B_m^{(I,P_o)}(x_I)\theta_{l,m}^B \tag{28}$$

where $\theta_{l,m}^B = \theta_{l,m_R}^B + \mathrm{j}\theta_{l,m_I}^B \in \mathbb{C}$, $1 \le l \le N_R$ and $1 \le m \le N_I$, are the CV weights. Denote

$$\boldsymbol{\theta}_B = \left[\theta_{1,1}^B \ \theta_{1,2}^B \cdots \theta_{l,m}^B \cdots \theta_{N_R,N_I}^B\right]^T \in \mathbb{C}^{N_B} \tag{29}$$

where $N_B = N_R N_I$. The task of identifying the nonlinearity $\Psi(\,)$ is turned into one of estimating $\boldsymbol{\theta}_B$.

2) CV Polynomial Model: Similarly for the conventional polynomial modeling with polynomial degree P_o, let us define the set of $P_o + 1$ polynomial basis functions as

$$P_l^{(s)}(x_s) = x_s^l, \quad 0 \le l \le P_o. \tag{30}$$

Then, using the tensor product between the two sets of univariate polynomial basis functions, $P_l^{(R)}(x_R)$ for $0 \le l \le P_o$ and $P_m^{(I)}(x_I)$ for $0 \le m \le P_o$, a set of new polynomial basis functions $P_{l,m}(x) = P_l^{(R)}(x_R)P_m^{(I)}(x_I)$ for $0 \le l, m \le P_o$ can be formed, giving rise to the CV polynomial model

$$\widehat{w} = \widehat{\Psi}_P(x) = \sum_{l=0}^{P_o}\sum_{m=0}^{P_o} P_{l,m}(x)\theta_{l,m}^P = \sum_{l=0}^{P_o}\sum_{m=0}^{P_o} P_l^{(R)}(x_R)P_m^{(I)}(x_I)\theta_{l,m}^P \tag{31}$$

where $\theta_{l,m}^P = \theta_{l,m_R}^P + \mathrm{j}\theta_{l,m_I}^P \in \mathbb{C}$, $0 \le l, m \le P_o$, are the CV weights. Define

$$\boldsymbol{\theta}_P = \left[\theta_{0,0}^P \ \theta_{0,1}^P \cdots \theta_{l,m}^P \cdots \theta_{P_o,P_o}^P\right]^T \in \mathbb{C}^{N_P} \tag{32}$$

where $N_P = (1+P_o)^2$. The task of identifying the nonlinearity $\Psi(\,)$ becomes one of estimating $\boldsymbol{\theta}_P$.

B. Model Structure Parameters

1) Polynomial Model: For the conventional polynomial model, there is only one model structure parameter, and choosing the polynomial degree $P_o = 4$ is sufficient for most practical applications.

2) B-Spline Model: For the B-spline neural network, choosing $P_o = 4$ is also sufficient for most applications. In our application, the knot sequence is symmetric and $U_{\min} = -U_{\max}$. Given the required average transmitted signal power, the peak amplitude in the symbol set (2) is known, and hence $U_{\max}$ is known. $N_R = N_I = N_s = 6$ to 10 is sufficient for accurately modeling on the finite interval $[U_{\min},\ U_{\max}]$. The $N_s + 1 - P_o$ internal knots may be uniformly spaced in the interval $[U_{\min},\ U_{\max}]$. Note that there exist no data for $x_s < U_{\min}$ and $x_s > U_{\max}$ 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. However, the precise choice of these external knots does not really matter, in terms of modeling accuracy.

TABLE I

COMPLEXITY OF POLYNOMIAL MODEL (31) FOR $P_o = 4$

Computation	Multiplications	Additions
Two sets of 1-D basis functions	2×4	0
Output of (31)	3×25	2×24
Total	83	48

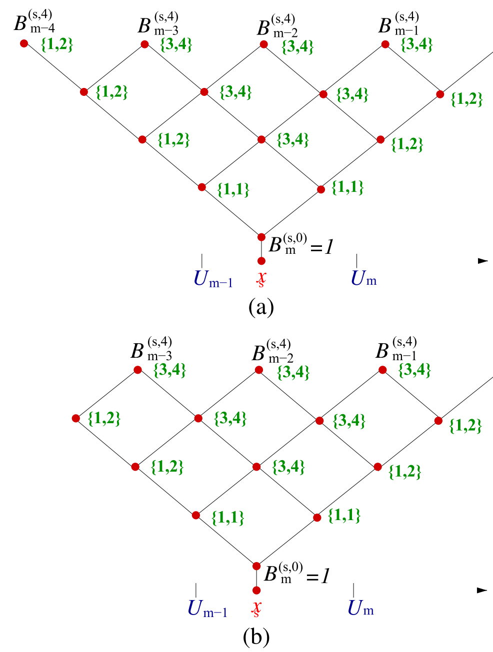


Fig. 3. Complexity of B-spline model with $P_o = 4$ using De Boor recursion, where $\{a, b\}$ beside a node indicates that it requires a additions and b multiplications to compute the basis function value at this node. The case of $m = N_s + 1$ is identical to (b).

C. Complexity Analysis

1) Complexity of Polynomial Model (31): Complexity analysis of the CV polynomial model is straightforward, and the computational complexity of computing the polynomial model (31) is obviously on the order of $(1 + P_o)^2$, denoted as $\mathcal{O}((P_o+1)^2)$. As an example, the computational requirements for $P_o = 4$ are listed in Table I.

2) Complexity of B-Spline Model (28): Comparing the B-spline modeling of (26)–(28) with the polynomial modeling of (30) and (31) and noting that N_B can be significantly larger than N_P, it would appear that the complexity of the CV B-spline model would be significantly higher than that of the CV polynomial model. This is in fact not the case, and the complexity of the CV B-spline modeling also depends only on P_o and not on the number of basis functions N_s.

Given $x_s \in [U_{\min},\ U_{\max}]$, there are only $P_o + 1$ basis functions with nonzero values at most. Fig. 3 shows the complexity of generating the B-spline basis function set for $P_o = 4$, which shows that the total requirements are 26 additions and 38 multiplications at most. Thus, in the tensor-product B-spline model (28), there are only $(P_o + 1)^2$ nonzero basis functions at most for any given input, which is comparable with the tensor-product polynomial model (31) with $(P_o+1)^2$ nonzero basis functions. The upper bound and lower bound

TABLE II

COMPLEXITY OF B-SPLINE MODEL (28) FOR $P_o = 4$

Computation	Multiplications	Additions
Upper bound:		
Two sets of 1-D basis functions	2×38	2×26
Output of (28)	3×25	2×24
Total	151	100
Lower bound:		
Two sets of 1-D basis functions	2×36	2×25
Output of (28)	3×16	2×15
Total	120	80

computational requirements for the CV B-spline modeling with $P_o = 4$ are listed in Table II, where it can be seen that the complexity of the B-spline modeling is no more than twice of the polynomial modeling. Therefore, the computational complexity of computing the B-spline model (28) is still on the order of $\mathcal{O}((P_o + 1)^2)$.

D. Optimal Robustness Property of B-Spline Model

A critical aspect to consider in a model representation is its stability with respect to perturbation of the model parameters, because in identification, the data are inevitably noisy, which will perturb the model parameters away from their true values. A significant advantage of the B-spline model over the polynomial model is its superior numerical stability. B-spline functions are optimally stable bases [4]–[6], and this optimality is due to the convexity of its model bases, i.e., they are all positive and sum up to one. In contrast, the polynomial model is far inferior in terms of numerical stability.

Let us first analyze this aspect theoretically. Assume that the real-valued true system can be represented by the polynomial model of degree P_o exactly as

$$y_s = \sum_{i=0}^{P_o} a_i x_s^i$$

as well as by the following B-spline model exactly:

$$y_s = \sum_{i=1}^{N_s} b_i B_i^{(s,P_o)}(x_s)$$

where $y_s, x_s \in \mathbb{R}$. Because of the noisy identification data, the estimated model coefficients are perturbed from their true values to $\widehat{a}_i = a_i + \varepsilon_i$ for the polynomial model and to $\widehat{b}_i = b_i + \varepsilon_i$ for the B-spline model. Assume that all the estimation noises ε_i are bounded by $|\varepsilon_i| < \varepsilon_{\max}$. The upper bound of $|y_s - \widehat{y}_s|$ for the B-spline model can be worked out to be

$$|y_s - \widehat{y}_s| = \left| \sum_{i=1}^{N_s} b_i B_i^{(s,P_o)}(x_s) - \sum_{i=1}^{N_s} \widehat{b}_i B_i^{(s,P_o)}(x_s) \right|$$
$$< \varepsilon_{\max} \left| \sum_{i=1}^{N_s} B_i^{(s,P_o)}(x_s) \right| = \varepsilon_{\max}.$$

Observe that the upper bound of the B-spline model output perturbation only depends on the upper bound of the perturbation noise, it does not depend on the input value x_s,

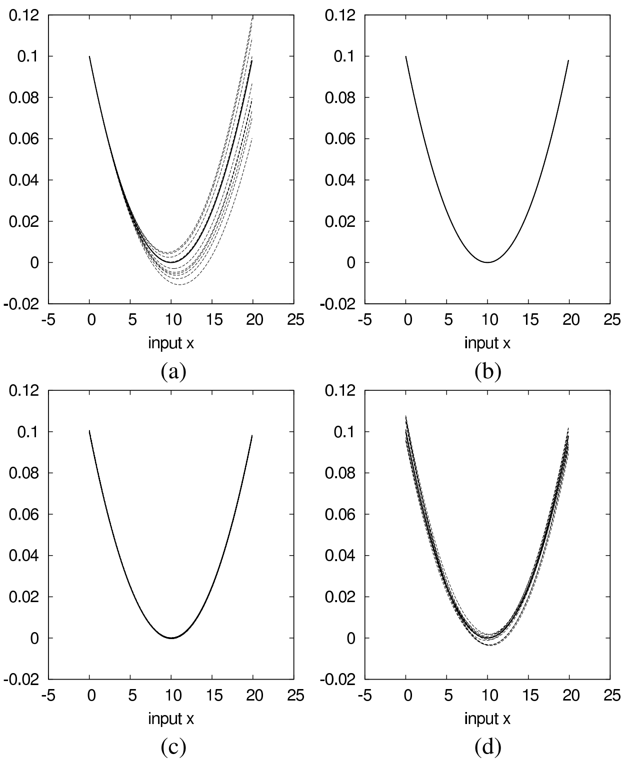


Fig. 4. (a) Polynomial model with UDRN perturbation noises drawn from $[-0.0001,\ 0.0001]$. (b) B-spline model with UDRN perturbation noises drawn from $[-0.0001,\ 0.0001]$. (c) B-spline model with UDRN perturbation noises drawn from $[-0.001,\ 0.001]$. (d) B-spline model with UDRN perturbation noises drawn from $[-0.01,\ 0.01]$. Cited from [16].

the number of basis functions N_s, or the polynomial degree P_o. Hence, the B-spline model enjoys the maximum numerical robustness, and this optimal robustness property is well known. In contrast, the upper bound of $|y_s - \widehat{y}_s|$ for the polynomial model can be worked out to be

$$|y_s - \widehat{y}_s| = \left|\sum_{i=0}^{P_o} a_i x_s^i - \sum_{i=0}^{P_o} \widehat{a}_i x_s^i\right| < \varepsilon_{\max} \left|\sum_{i=0}^{P_o} x_s^i\right|.$$

Observe that the upper bound of the polynomial model output perturbation depends not only on the upper bound of the perturbation noise but also on the input value x_s and the polynomial degree P_o. The higher the polynomial degree P_o, the more serious the polynomial model may be perturbed, a well-known drawback of using polynomial modeling.

We use the simple example of [16] to demonstrate the excellent numerical stability of the B-spline model over the polynomial model in Fig. 4. Fig. 4(a) shows a quadratic polynomial function $y_s = 0.001x_s^2 - 0.02x_s + 0.1$ defined over $x_s \in [0,\ 20]$ in solid curve. With the knot sequence $\{-5, -4, 0, 20, 24, 25\}$, this function is modeled as a quadratic B-spline model of $y_s = 0.14B_1^{(s,2)}(x_s) - 0.10B_2^{(s,2)}(x_s) + 0.14B_3^{(s,2)}(x_s)$, which is shown in Fig. 4(b) in solid curve. We draw three noises from a uniformly distributed random number (UDRN) in $[-0.0001,\ 0.0001]$, and add them to the three parameters in the two models, respectively. Fig. 4(a) and (b) shows the ten sets of the perturbed functions in dashed curve generated by perturbing the two models' parameters. It can be clearly seen from Fig. 4(a) that the polynomial model is seriously perturbed, but there is no noticeable change at all in Fig. 4(b) for the B-spline model. Next we draw three perturbation noises from a UDRN in $[-0.001,\ 0.001]$, and add them to the three parameters of the B-spline model. Again, the B-spline model is hardly affected, as shown in Fig. 4(c). We then draw three perturbation noises from a UDRN in $[-0.01,\ 0.01]$ to add to the three B-spline parameters, and the results obtained are shown in Fig. 4(d). Observe from Fig. 4(a) and (d) that, despite the fact that the strength of the perturbation noise added to the B-spline model coefficients is 100 times larger than that added to the polynomial model coefficients, the B-spline model is still much less seriously perturbed than the polynomial model.

E. Identifying Hammerstein Channel

We will present the identification of the Hammerstein channel using the CV B-spline neural network approach, since the identification algorithm is identical using the CV polynomial modeling approach. Therefore, we drop the subscript $_B$ and superscript ^{B} from the B-spline model.

Given a block of N training data, $\{x_k, y_k\}_{k=0}^{N-1}$, the identification task is to obtain the estimates of $\boldsymbol{h}$ and $\boldsymbol{\theta}$ by minimizing the cost function

$$J(\boldsymbol{h}, \boldsymbol{\theta}) = \frac{1}{N}\sum_{k=0}^{N-1} |\widehat{e}_k|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |y_k - \widehat{y}_k|^2 \tag{33}$$

subject to the constraint of $h_0 = 1$, in which the prediction of y_k is given by

$$\widehat{y}_k = \sum_{i=0}^{L_{\mathrm{cir}}} h_i \widehat{w}_{k-i} = \sum_{i=0}^{L_{\mathrm{cir}}} h_i \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(x_{k-i})\theta_{l,m} \tag{34}$$

where $x_{k-i} = x_{N+k-i}$ if $k < i$. The cost function (33) is convex with respect to $\boldsymbol{h}$ when fixing $\boldsymbol{\theta}$, and convex with respect to $\boldsymbol{\theta}$ given $\boldsymbol{h}$. According to [17] and [18], the estimates of $\boldsymbol{\theta}$ and $\boldsymbol{h}$ are unbiased, irrespective to the algorithm used to minimize the cost function (33). In [16], an alternating least squares (ALS) procedure was proposed, which guarantees to find the unique optimal solution of $\boldsymbol{\theta}$ and $\boldsymbol{h}$ in only a few iterations. We adopt this ALS procedure in our current application. This ALS procedure is summarized below.

Initialization: Define the amalgamated parameter vector

$$\boldsymbol{\omega} = [\boldsymbol{\theta}^T\ h_1\boldsymbol{\theta}^T\ h_2\boldsymbol{\theta}^T \cdots h_{L_{\mathrm{cir}}}\boldsymbol{\theta}^T]^T \in \mathbb{C}^{(L_{\mathrm{cir}}+1)N_B}. \tag{35}$$

Further define the regression matrix $\boldsymbol{P} \in \mathbb{R}^{N\times(L_{\mathrm{cir}}+1)N_B}$

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{\phi}^T(0) & \boldsymbol{\phi}^T(-1) & \cdots & \boldsymbol{\phi}^T(-L_{\mathrm{cir}}) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\phi}^T(k) & \boldsymbol{\phi}^T(k-1) & \cdots & \boldsymbol{\phi}^T(k-L_{\mathrm{cir}}) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\phi}^T(N-1) & \boldsymbol{\phi}^T(N-2) & \cdots & \boldsymbol{\phi}^T(N-1-L_{\mathrm{cir}}) \end{bmatrix} \tag{36}$$

with $\boldsymbol{\phi}(k) = [\phi_{1,1}(k)\ \phi_{1,2}(k)\cdots\phi_{l,m}(k)\cdots\phi_{N_R,N_I}(k)]^T$, in which $\phi_{l,m}(k) = B_{l,m}^{(P_o)}(x_k)$ for $1 \le l \le N_R$ and $1 \le m \le N_I$. The regularized least squares (LS) estimate of $\boldsymbol{\omega}$ is $\widehat{\boldsymbol{\omega}} = (\boldsymbol{P}^T\boldsymbol{P}+\rho\boldsymbol{I})^{-1}\boldsymbol{P}^T\boldsymbol{y}$, where $\boldsymbol{I}$ denotes the identity matrix of appropriate dimension and ρ is a small positive constant, e.g., $\rho = 10^{-5}$. The first N_B elements of $\widehat{\boldsymbol{\omega}}$ provide an initial estimate for $\boldsymbol{\theta}$, which is denoted as $\widehat{\boldsymbol{\theta}}^{(0)}$. Note that $\widehat{\boldsymbol{\theta}}^{(0)}$ is an unbiased estimate for $\boldsymbol{\theta}$ for sufficiently small ρ.

ALS Estimation Procedure: For $1 \leq \tau \leq \tau_{\max}$, e.g., $\tau_{\max} = 4$, perform the following.

1) Given $\widehat{\boldsymbol{\theta}}^{(\tau-1)}$, calculate the LS estimate $\widehat{\boldsymbol{h}}^{(\tau)}$. In particular, define the regression matrix $\boldsymbol{Q} \in \mathbb{C}^{N\times(L_{\mathrm{cir}}+1)}$

$$\boldsymbol{Q} = \begin{bmatrix} \widehat{w}_0 & \widehat{w}_{-1} & \cdots & \widehat{w}_{-L_{\mathrm{cir}}} \\ \vdots & \vdots & \vdots & \vdots \\ \widehat{w}_k & \widehat{w}_{k-1} & \cdots & \widehat{w}_{k-L_{\mathrm{cir}}} \\ \vdots & \vdots & \vdots & \vdots \\ \widehat{w}_{N-1} & \widehat{w}_{N-2} & \cdots & \widehat{w}_{N-1-L_{\mathrm{cir}}} \end{bmatrix} \quad (37)$$

in which

$$\widehat{w}_k = \widehat{\Psi}(x_k) = \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(x_k)\widehat{\theta}_{l,m}^{(\tau-1)}. \quad (38)$$

The LS estimate $\widehat{\boldsymbol{h}}^{(\tau)}$ is readily given by

$$\widehat{\underline{\boldsymbol{h}}}^{(\tau)} = (\boldsymbol{Q}^H\boldsymbol{Q})^{-1}\boldsymbol{Q}^H\boldsymbol{y} \quad (39)$$

$$\widehat{h}_i^{(\tau)} = \widehat{\underline{h}}_i^{(\tau)}/\widehat{\underline{h}}_0^{(\tau)},\ 0 \leq i \leq L_{\mathrm{cir}}. \quad (40)$$

2) Given $\widehat{\boldsymbol{h}}^{(\tau)}$, calculate the LS estimate $\widehat{\boldsymbol{\theta}}^{(\tau)}$. Specifically introduce

$$\varphi_{l,m}(k) = \sum_{i=0}^{L_{\mathrm{cir}}} \widehat{h}_i^{(\tau)} B_{l,m}^{(P_o)}(x_{k-i}) \in \mathbb{C}. \quad (41)$$

Further define the regression matrix

$$\boldsymbol{S} = \left[\boldsymbol{\varphi}(0)\ \boldsymbol{\varphi}(1)\cdots\boldsymbol{\varphi}(N-1)\right]^T \in \mathbb{C}^{N\times N_B} \quad (42)$$

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

The LS estimate $\widehat{\boldsymbol{\theta}}^{(\tau)}$ is given by $\widehat{\boldsymbol{\theta}}^{(\tau)} = (\boldsymbol{S}^H\boldsymbol{S})^{-1}\boldsymbol{S}^H\boldsymbol{y}$.

Clearly, this ALS procedure guarantees to converge to the joint unbiased estimate of $\boldsymbol{h}$ and $\boldsymbol{\theta}$ that is the unique minimum solution of the cost function (33). This is simply because given the unbiased estimate $\widehat{\boldsymbol{\theta}}^{(\tau-1)}$ of $\boldsymbol{\theta}$, the LS estimate $\widehat{\boldsymbol{h}}^{(\tau)}$ is the unbiased estimate of $\boldsymbol{h}$, and given the unbiased estimate $\widehat{\boldsymbol{h}}^{(\tau)}$, the LS estimate $\widehat{\boldsymbol{\theta}}^{(\tau)}$ is the unbiased estimate of $\boldsymbol{\theta}$.

Remark 1: Because the B-spline modeling has the optimal robustness property as discussed in Section III-D, we expect that the CV B-spline-based estimate $\widehat{\Psi}_B(x)$ is a more accurate estimate of the true HPA's nonlinearity $\Psi(x)$ than the CV polynomial-based estimate $\widehat{\Psi}_P(x)$. This will be verified in our comparative performance evaluation.

F. Inverting HPA's Nonlinearity

1) CV B-Spline Inverting Model: We utilize another B-spline neural network to model the inverse mapping of the HPA's CV nonlinearity defined by

$$x_k = \Psi^{-1}(w_k) = \Phi(w_k). \quad (43)$$

Define two knot sequences similar to (25) for w_R and w_I, respectively. We can construct the inverting B-spline model

$$\begin{aligned}\widehat{x} &= \widehat{\Phi}_B(w;\boldsymbol{\alpha}_B) \\ &= \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_l^{(R,P_o)}(w_R)B_m^{(I,P_o)}(w_I)\alpha_{l,m}^B\end{aligned} \quad (44)$$

where $B_l^{(R,P_o)}(w_R)$ and $B_m^{(I,P_o)}(w_I)$ are similarly calculated based on the De Boor recursion (26) and (27), while

$$\boldsymbol{\alpha}_B = [\alpha_{1,1}^B\ \alpha_{1,2}^B\cdots\alpha_{l,m}^B\cdots\alpha_{N_R,N_I}^B]^T \in \mathbb{C}^{N_B}. \quad (45)$$

Inverting the HPA's nonlinearity becomes the problem of estimating $\boldsymbol{\alpha}_B$.

2) CV Polynomial Inverting Model: By defining the two sets of polynomial basis functions similar to (30) for w_R and w_I, respectively, we can construct the inverting polynomial model

$$\begin{aligned}\widehat{x} &= \widehat{\Phi}_P(w;\boldsymbol{\alpha}_P) \\ &= \sum_{l=0}^{P_o}\sum_{m=0}^{P_o} P_l^{(R)}(w_R)P_m^{(I)}(w_I)\alpha_{l,m}^P\end{aligned} \quad (46)$$

where

$$\boldsymbol{\alpha}_P = \left[\alpha_{0,0}^P\ \alpha_{0,1}^P\cdots\alpha_{l,m}^P\cdots\alpha_{P_o,P_o}^P\right]^T \in \mathbb{C}^{N_P}. \quad (47)$$

Inverting the HPA's nonlinearity is turned into the problem of estimating $\boldsymbol{\alpha}_P$.

3) Estimation Algorithm: To estimate $\boldsymbol{\alpha}_B$ or $\boldsymbol{\alpha}_P$ needs the input–output training data $\{w_k, x_k\}$, but w_k is unavailable. We adopt the same pseudotraining data approach of [2] and [3], by replacing w_k with its estimate $\widehat{w}_k = \widehat{\Psi}_B(x_k)$ or $\widehat{w}_k = \widehat{\Psi}_P(x_k)$ based on the identified HPA's nonlinearity $\widehat{\Psi}_B(\ )$ or $\widehat{\Psi}_P(\ )$.

Again we present the estimation algorithm for the CV B-spline inverting model (44) and drop the subscript $_B$ and superscript ^{B}, since the estimation algorithm for the CV polynomial inverting model is exactly the same. Over the pseudotraining data set $\{\widehat{w}_k, x_k\}_{k=0}^{N-1}$, the regression matrix $\boldsymbol{B} \in \mathbb{R}^{N\times N_B}$ can be formed as

$$\boldsymbol{B} = \begin{bmatrix} B_{1,1}^{(P_o)}(\widehat{w}_0) & B_{1,2}^{(P_o)}(\widehat{w}_0) & \cdots & B_{N_R,N_I}^{(P_o)}(\widehat{w}_0) \\ B_{1,1}^{(P_o)}(\widehat{w}_1) & B_{1,2}^{(P_o)}(\widehat{w}_1) & \cdots & B_{N_R,N_I}^{(P_o)}(\widehat{w}_1) \\ \vdots & \vdots & \vdots & \vdots \\ B_{1,1}^{(P_o)}(\widehat{w}_{N-1}) & B_{1,2}^{(P_o)}(\widehat{w}_{N-1}) & \cdots & B_{N_R,N_I}^{(P_o)}(\widehat{w}_{N-1}) \end{bmatrix} \quad (48)$$

and the LS solution is given by $\widehat{\boldsymbol{\alpha}} = (\boldsymbol{B}^T\boldsymbol{B})^{-1}\boldsymbol{B}^T\boldsymbol{x}$.

Remark 2: Because the pseudotraining input data $\{\widehat{w}_k\}_{k=0}^{N-1}$ are highly noisy, which will seriously affect the polynomial model but not the B-spline model as analyzed in Section III-D, the CV polynomial inverting model (46) will be a far less accurate estimate of the true HPA's inversion $\Psi^{-1}(\ )$, compared with the CV B-spline inverting model (44). This will be confirmed by our comparative performance evaluation presented in Section IV.

TABLE III

IDENTIFICATION RESULTS AVERAGED OVER 100 RUNS FOR THE CIR COEFFICIENT VECTOR $\boldsymbol{h}$ OF THE HAMMERSTEIN CHANNEL USING THE CV B-SPLINE NEURAL NETWORK APPROACH

Tap No.	True parameter	$E_x/N_o = 5$ dB		$E_x/N_o = 10$ dB	
		average estimate	standard deviation	average estimate	standard deviation
OBO = 3 dB					
h_0	1	1		1	
h_1	$-0.3732 - j0.6123$	$-0.3732 - j0.6122$	9.152e-4, 1.021e-3	$-0.3732 - j0.6123$	5.147e-4, 5.744e-4
h_2	$0.3584 + j0.3676$	$0.3586 + j0.3676$	9.702e-4, 8.555e-4	$0.3585 + j0.3676$	5.455e-4, 4.812e-4
h_3	$0.3052 + j0.2053$	$0.3052 + j0.2052$	9.278e-4, 8.596e-4	$0.3052 + j0.2052$	5.219e-4, 4.834e-4
h_4	$0.2300 + j0.1287$	$0.2300 + j0.1286$	7.806e-4, 8.650e-4	$0.2300 + j0.1286$	4.391e-4, 4.865e-4
h_5	$0.7071 + j0.7071$	$0.7070 + j0.7069$	1.161e-3, 1.178e-3	$0.7071 + j0.7070$	6.530e-4, 6.627e-4
h_6	$0.6123 - j0.3732$	$0.6122 - j0.3733$	1.051e-3, 1.115e-3	$0.6122 - j0.3732$	5.913e-4, 6.271e-4
h_7	$-0.3584 + j0.3676$	$-0.3583 + j0.3675$	9.100e-4, 1.056e-3	$-0.3584 + j0.3675$	5.119e-4, 5.939e-4
h_8	$-0.2053 - j0.3052$	$-0.2054 - j0.3051$	9.343e-4, 9.233e-4	$-0.2053 - j0.3051$	5.253e-4, 5.193e-4
h_9	$0.1287 - j0.2300$	$0.1287 - j0.2299$	8.017e-4, 8.728e-4	$0.1287 - j0.2299$	4.508e-4, 4.908e-4
OBO = 5 dB					
h_0	1	1		1	
h_1	$-0.3732 - j0.6123$	$-0.3731 - j0.6122$	7.385e-4, 8.198e-4	$-0.3732 - j0.6123$	4.154e-4, 4.611e-4
h_2	$0.3584 + j0.3676$	$0.3586 + j0.3675$	7.687e-4, 6.879e-4	$0.3585 + j0.3675$	4.322e-4, 3.869e-4
h_3	$0.3052 + j0.2053$	$0.3052 + j0.2052$	7.505e-4, 6.757e-4	$0.3052 + j0.2053$	4.221e-4, 3.799e-4
h_4	$0.2300 + j0.1287$	$0.2300 + j0.1286$	6.253e-4, 6.947e-4	$0.2300 + j0.1287$	3.517e-4, 3.907e-4
h_5	$0.7071 + j0.7071$	$0.7071 + j0.7069$	9.318e-4, 9.480e-4	$0.7071 + j0.7070$	5.239e-4, 5.332e-4
h_6	$0.6123 - j0.3732$	$0.6121 - j0.3732$	8.424e-4, 8.854e-4	$0.6122 - j0.3732$	4.739e-4, 4.978e-4
h_7	$-0.3584 + j0.3676$	$-0.3583 + j0.3675$	7.471e-4, 8.454e-4	$-0.3584 + j0.3675$	4.202e-4, 4.754e-4
h_8	$-0.2053 - j0.3052$	$-0.2053 - j0.3052$	7.568e-4, 7.381e-4	$-0.2053 - j0.3052$	4.256e-4, 4.151e-4
h_9	$0.1287 - j0.2300$	$0.1287 - j0.2299$	6.476e-4, 6.922e-4	$0.1287 - j0.2299$	3.641e-4, 3.892e-4

TABLE IV

KNOT SEQUENCES FOR B-SPLINE MODEL AND INVERSE MODEL

Knot sequence for x_R and x_I
-10.0, -9.0, -1.0, -**0.9**, -0.06, -0.04, 0.0, 0.04, 0.06, **0.9**, 1.0, 9.0, 10.0
Knot sequence for w_R and w_I
-20.0, -18.0, -3.0, -**1.4**, -0.8, -0.4, 0.0, 0.4, 0.8, **1.4**, 3.0, 18.0, 20.0

IV. COMPARATIVE PERFORMANCE EVALUATION

We evaluated the comparative performance of the CV B-spline-based NIFDDFE and the CV polynomial-based NIFDDFE for a 64-QAM Hammerstein channel, in which the HPA was described by (6) and (7) with the parameter set given in (8). The dispersive channel had 10 taps ($L_{\text{cir}} = 9$) whose CIR coefficients are given in Table III. The size of the transmitted data block was $N = 2048$. The system's SNR was defined as SNR $= E_x/N_o$, where E_x was the average power of the input signal x_k to the HPA and $N_o = 2\sigma_e^2$.

For the CV B-spline neural network-based approach, the piecewise quartic polynomial of $P_o = 4$ was chosen, and the number of B-spline basis functions was set to $N_R = N_I = 8$, while the knot sequences adopted by the two CV B-spline neural networks for identifying and inverting the HPA's nonlinearity are listed in Table IV. For the CV polynomial modeling-based approach, we set the polynomial degree to $P_o = 4$. All the estimation results were obtained by averaging over 100 random runs.

The effectiveness of the CV B-spline neural network-based approach to identify this Hammerstein channel is demonstrated in Table III as well as in Figs. 5 and 6. It can be seen from Table III that the identification of the CIR tap vector in the Hammerstein channel was achieved with high precision even under the adverse operational condition of OBO = 3 dB and $E_x/N_o = 5$ dB. Note that under the HPA operational condition of OBO = 5 dB, the peak amplitude of $|x_k|$ was less than 0.09, while under the condition of OBO = 3 dB, the peak amplitude of $|x_k|$ was less than 0.14. The results of Figs. 5 and 6 clearly demonstrate the capability of the proposed CV B-spline

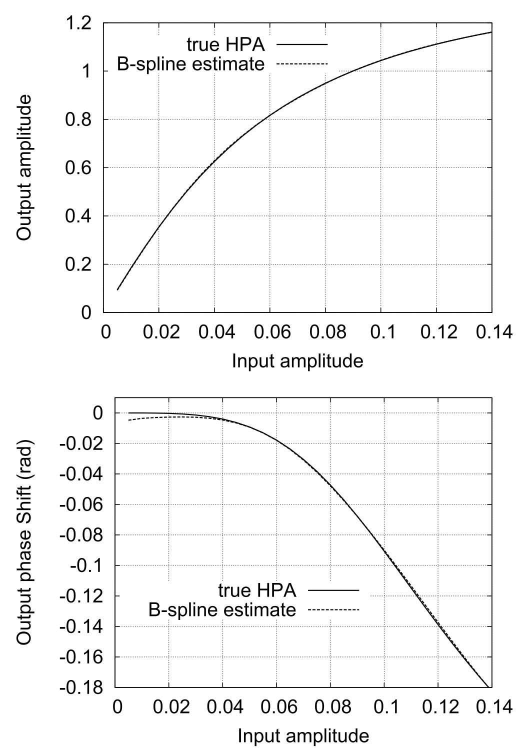


Fig. 5. Comparison of the HPA's nonlinearity $\Psi(\,)$ and its B-spline estimate $\widehat{\Psi}_B(\,)$ averaged over 100 runs, under OBO = 3 dB and $E_x/N_o = 5$ dB.

TABLE V

IDENTIFICATION RESULTS AVERAGED OVER 100 RUNS FOR THE CIR COEFFICIENT VECTOR $\boldsymbol{h}$ OF THE HAMMERSTEIN CHANNEL USING THE CV POLYNOMIAL MODELING APPROACH

Tap No.	True parameter	$E_x/N_o = 5$ dB		$E_x/N_o = 10$ dB	
		average estimate	standard deviation	average estimate	standard deviation
OBO = 3 dB					
h_0	1	1		1	
h_1	$-0.3732 - j0.6123$	$-0.3735 - j0.6120$	9.176e-4, 1.027e-3	$-0.3735 - j0.6120$	5.160e-4, 5.778e-4
h_2	$0.3584 + j0.3676$	$0.3596 + j0.3680$	9.723e-4, 8.540e-4	$0.3595 + j0.3680$	5.468e-4, 4.805e-4
h_3	$0.3052 + j0.2053$	$0.3052 + j0.2058$	9.262e-4, 8.591e-4	$0.3053 + j0.2059$	5.209e-4, 4.831e-4
h_4	$0.2300 + j0.1287$	$0.2310 + j0.1277$	7.786e-4, 8.603e-4	$0.2310 + j0.1277$	4.379e-4, 4.837e-4
h_5	$0.7071 + j0.7071$	$0.7072 + j0.7066$	1.165e-3, 1.187e-3	$0.7072 + j0.7067$	6.552e-4, 6.677e-4
h_6	$0.6123 - j0.3732$	$0.6118 - j0.3721$	1.052e-3, 1.116e-3	$0.6118 - j0.3721$	5.920e-4, 6.278e-4
h_7	$-0.3584 + j0.3676$	$-0.3582 + j0.3689$	9.077e-4, 1.055e-3	$-0.3582 + j0.3689$	5.105e-4, 5.930e-4
h_8	$-0.2053 - j0.3052$	$-0.2064 - j0.3052$	9.327e-4, 9.284e-4	$-0.2063 - j0.3052$	5.245e-4, 5.221e-4
h_9	$0.1287 - j0.2300$	$0.1284 - j0.2291$	8.057e-4, 8.615e-4	$0.1284 - j0.2292$	4.531e-4, 4.844e-4
OBO = 5 dB					
h_0	1	1		1	
h_1	$-0.3732 - j0.6123$	$-0.3740 - j0.6121$	7.360e-4, 8.281e-4	$-0.3741 - j0.6121$	4.138e-4, 4.657e-4
h_2	$0.3584 + j0.3676$	$0.3595 + j0.3681$	7.778e-4, 6.846e-4	$0.3594 + j0.3681$	4.374e-4, 3.851e-4
h_3	$0.3052 + j0.2053$	$0.3058 + j0.2058$	7.471e-4, 6.809e-4	$0.3058 + j0.2058$	4.202e-4, 3.829e-4
h_4	$0.2300 + j0.1287$	$0.2310 + j0.1271$	6.298e-4, 6.991e-4	$0.2310 + j0.1272$	3.542e-4, 3.931e-4
h_5	$0.7071 + j0.7071$	$0.7074 + j0.7074$	9.378e-4, 9.594e-4	$0.7074 + j0.7074$	5.273e-4, 5.396e-4
h_6	$0.6123 - j0.3732$	$0.6124 - j0.3729$	8.423e-4, 8.941e-4	$0.6125 - j0.3729$	4.737e-4, 5.028e-4
h_7	$-0.3584 + j0.3676$	$-0.3583 + j0.3686$	7.338e-4, 8.443e-4	$-0.3584 + j0.3686$	4.127e-4, 4.748e-4
h_8	$-0.2053 - j0.3052$	$-0.2056 - j0.3056$	7.538e-4, 7.359e-4	$-0.2056 - j0.3056$	4.239e-4, 4.138e-4
h_9	$0.1287 - j0.2300$	$0.1285 - j0.2297$	6.469e-4, 6.860e-4	$0.1285 - j0.2297$	3.638e-4, 3.858e-4

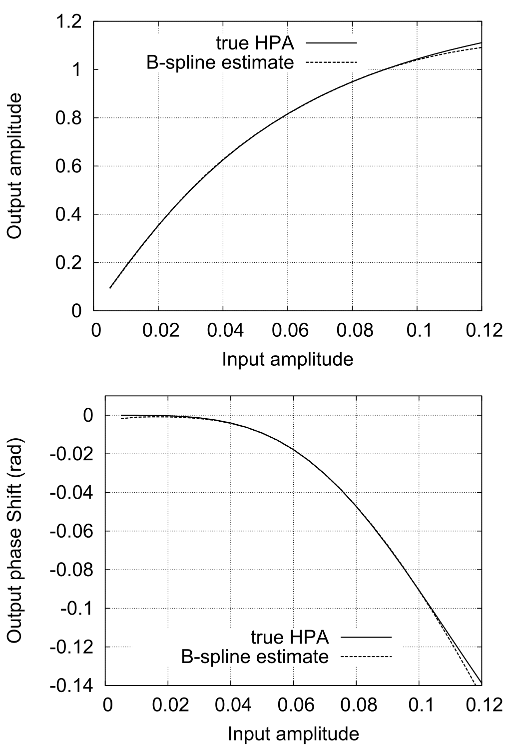


Fig. 6. Comparison of the HPA's nonlinearity $\Psi(\,)$ and its B-spline estimate $\widehat{\Psi}_B(\,)$ averaged over 100 runs, under OBO = 5 dB and $E_x/N_o = 10$ dB.

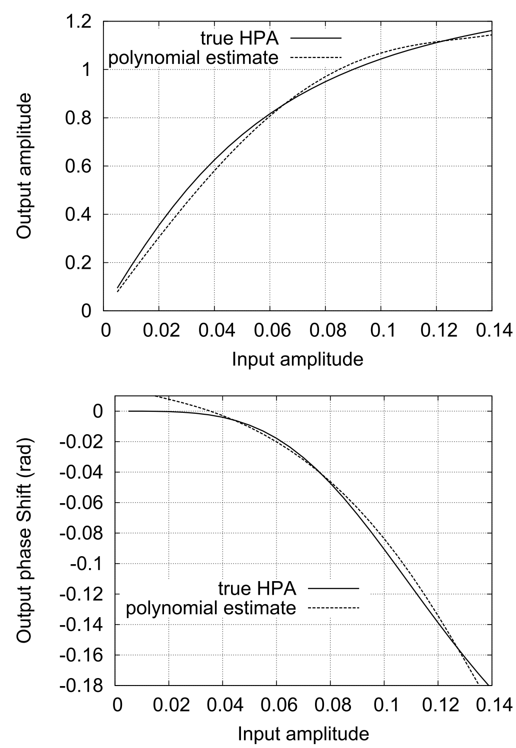


Fig. 7. Comparison of the HPA's nonlinearity $\Psi(\,)$ and its polynomial estimate $\widehat{\Psi}_P(\,)$ averaged over 100 runs, under OBO = 3 dB and $E_x/N_o = 5$ dB.

neural network to accurately model the HPA's nonlinearity, within the HPA's operational input range. As a comparison, the results obtained by applying the CV polynomial-based modeling approach to identify this Hammerstein channel are shown in Table V as well as in Figs. 7 and 8. Table V indicates that the linear subsystem of this Hammerstein channel is also identified with high precision by the CV polynomial-based approach, which is expected. By comparing Figs. 7 and 8 with

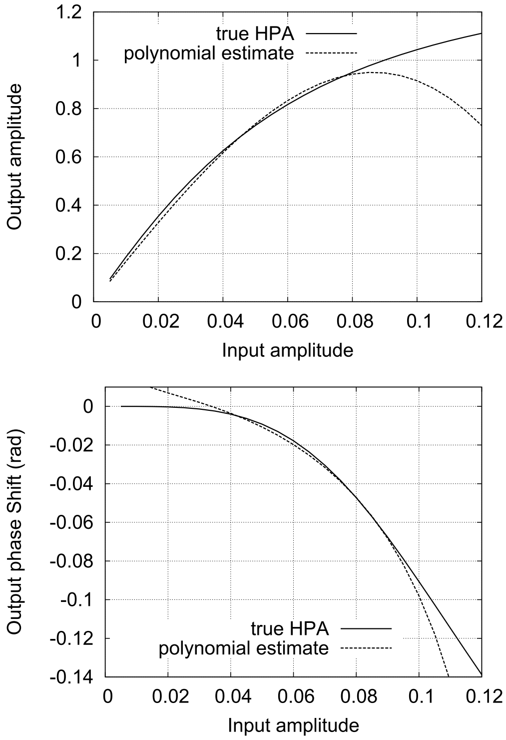


Fig. 8. Comparison of the HPA's nonlinearity $\Psi(\ )$ and its polynomial estimate $\widehat{\Psi}_P(\ )$ averaged over 100 runs, under $OBO = 5$ dB and $E_x/N_o = 10$ dB.

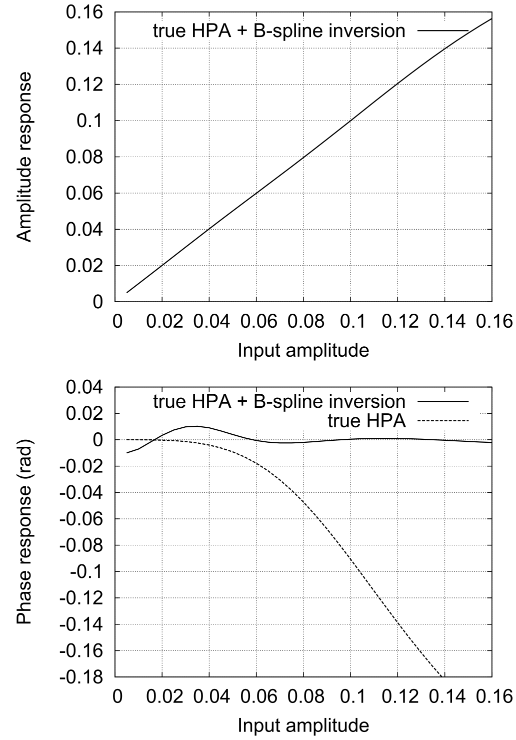


Fig. 9. Combined response of the true HPA and its estimated B-spline inversion averaged over 100 runs, under OBO = 3 dB and $E_x/N_o = 5$ dB.

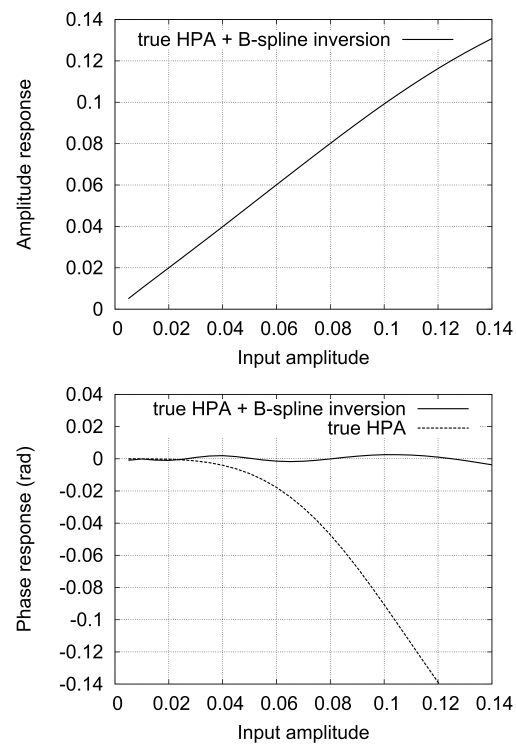


Fig. 10. Combined response of the true HPA and its estimated B-spline inversion averaged over 100 runs, under OBO = 5 dB and $E_x/N_o = 10$ dB.

Figs. 5 and 6, it can be seen that the CV HPA's nonlinearity identified by the polynomial-based approach is less accurate than the B-spline based approach within the HPA's operational input range, which confirms the analysis in Section III-D.

The combined responses of the HPA's true nonlinearity and its estimated inversion obtained by the CV B-spline inverting scheme under the two operating conditions are shown in Figs. 9 and 10. The results clearly show the capability of the CV B-spline neural network to accurately model the inversion of the HPA's nonlinearity based only on the pseudotraining data. More specifically, the results of Figs. 9 and 10 clearly indicate that the combined response of the true HPA's nonlinearity $\Psi(\ )$ and its estimated inversion $\widehat{\Phi}_B(\ )$ satisfies $\widehat{\Phi}_B(\Psi(x)) \approx x$ that is, the magnitude of the combined response is $|\widehat{\Phi}_B(\Psi(x))| \approx |x|$ and the phase shift of the combined response is approximately zero. In other words, $\widehat{\Phi}_B(\ )$ is an accurate inversion of $\Psi(\ )$. This clearly demonstrates the optimal robustness property of the B-spline modeling presented in Section III-D. In contrast, the combined responses of the HPA's true nonlinearity and its estimated polynomial inversion depicted in Figs. 11 and 12 under the two HPA operating conditions unmistakably show that the polynomial-based inversion estimate $\widehat{\Phi}_P(\ )$ is much less accurate than the B-spline based estimate. Evidently, the polynomial modeling is much more sensitive to the noise contained in the pseudo-training input $\{\widehat{w}_k\}$.

The bit error rate (BER) performance of the B-spline-based NIFDDFE constructed using the estimated CIR, HPA, and HPA's inversion is shown in Fig. 13 under the two HPA operating conditions. From Fig. 13, it can be seen that four iterations are sufficient for the NIFDDFE. Since the first iteration of the NIFDDFE is identical to the NFDE solution

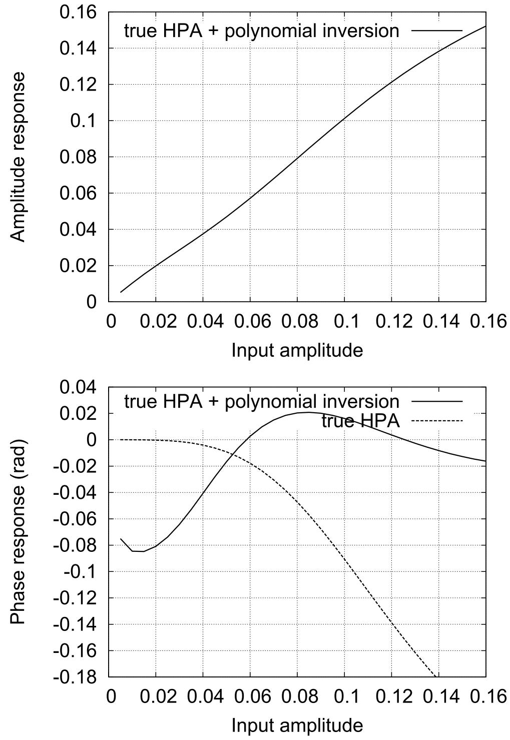


Fig. 11. Combined response of the true HPA and its estimated polynomial inversion averaged over 100 runs, under OBO = 3 dB and $E_x/N_o = 5$ dB.

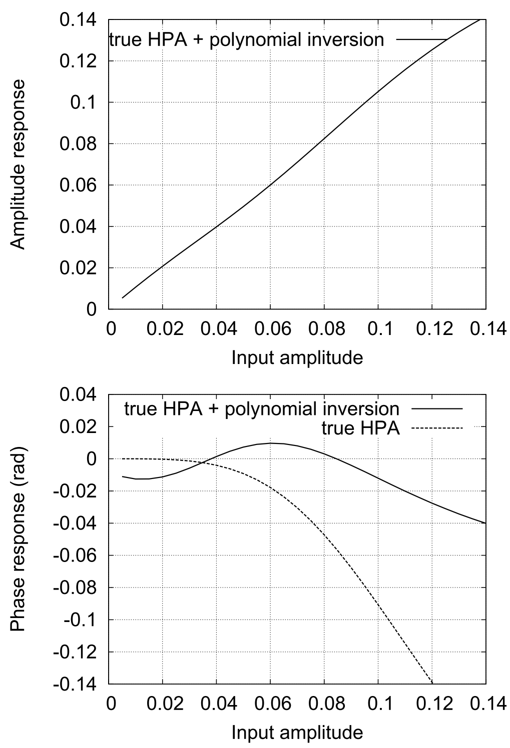


Fig. 12. Combined response of the true HPA and its estimated polynomial inversion averaged over 100 runs, under OBO = 5 dB and $E_x/N_o = 10$ dB.

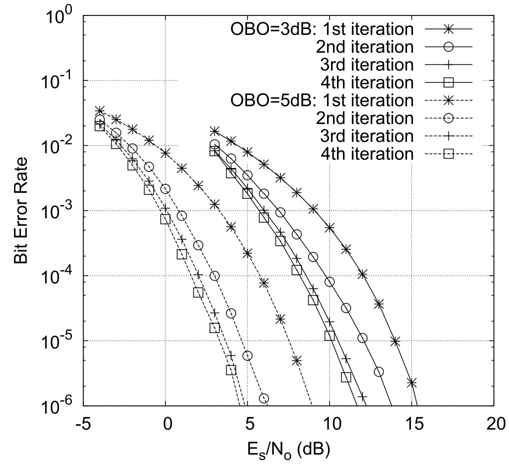
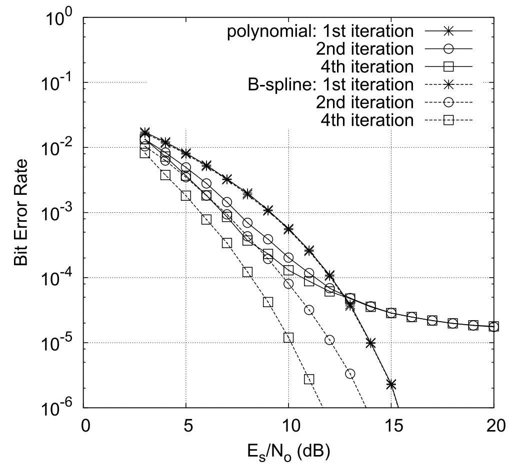
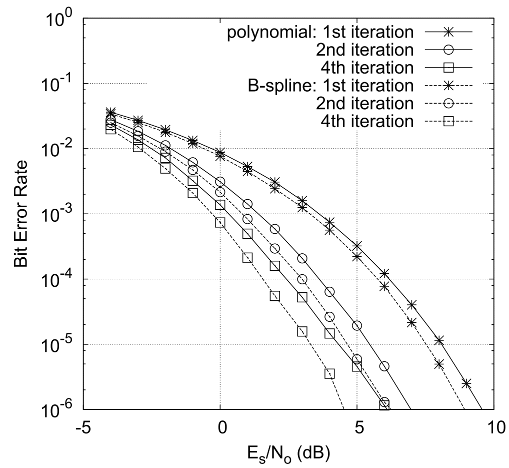


Fig. 13. BER performance of the B-spline-based NIFDDFE under the two HPA operating conditions of OBO = 3 dB and OBO = 5 dB.



(a) OBO= 3 dB



(a) OBO= 5 dB

Fig. 14. BER performance comparison of the B-spline-based NIFDDFE and the polynomial-based NIFDDFE.

without using decision feedback [3], the results of Fig. 10 confirm that the NIFDDFE significantly outperforms the NFDE. The BER performance of the polynomial-based NIFDDFE again constructed using the estimated CIR, HPA, and HPA's inversion are shown in Fig. 14, in comparison with the results of the B-spline-based NIFDDFE. The results of Fig. 14 clearly demonstrates that the B-spline-based NIFDDFE significantly outperforms the polynomial-based NIFDDFE. In particular, when the HPA is operating in the severe nonlinear region, the polynomial-based NIFDDFE exhibits a high error floor, but this is not the case for the B-spline-based NIFDDFE.

V. Conclusion

This paper has evaluated comparative performance of the CV B-spline neural network and polynomial modeling approaches applied to the state-of-the-art iterative FD decision feedback equalization of Hammerstein communication channels with the nonlinear HPA at the transmitter. The optimal robustness of the B-spline modeling has been reviewed and it has been shown that the CV B-spline modeling approach has a comparable computational complexity with the conventional CV polynomial modeling approach. Simulation results obtained have verified that the CV B-spline-based NIFDDFE significantly outperforms the CV polynomial-based NIFDDFE design of comparable complexity. Our conclusions have thus demonstrated that the CV B-spline neural network approach offers a highly effective and accurate means for identifying and inverting Hammerstein systems.

References

[1] S. Chen, X. Hong, J. Gao, and C. J. Harris, "Complex-valued B-spline neural networks for modeling and inverting Hammerstein systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 9, pp. 1673–1685, Sep. 2014.

[2] X. Hong, S. Chen, and C. J. Harris, "B-spline neural network based single-carrier frequency domain equalisation for Hammerstein channels," in *Proc. 2014 (IJCNN)*, Beijing, China, Jul. 2014, pp. 1834–1841.

[3] X. Hong, S. Chen, C. J. Harris, and E. F. Khalaf, "Single-carrier frequency domain equalization for Hammerstein communication systems using complex-valued neural networks," *IEEE Trans. Signal Proces.*, vol. 62, no. 17, pp. 4467–4478, Sep. 2014.

[4] J. M. Peña, "B-spline and optimal stability," *Math. Comput.*, vol. 66, no. 220, pp. 1555–1560, Oct. 1997.

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

[6] E. Mainar and J. M. Peña, "Optimal stability of bivariate tensor product B-bases," *J. Numer. Anal. Ind. Appl. Math.*, vol. 6, nos. 3–4, pp. 95–104, 2012.

[7] D. Falconer, S. L. Ariyavisitakul, A. Benyamin-Seeyar, and B. Eidson, "Frequency domain equalization for single-carrier broadband wireless systems," *IEEE Commun. Mag.*, vol. 40, no. 4, pp. 58–66, Apr. 2002.

[8] N. Benvenuto and S. Tomasin, "Iterative design and detection of a DFE in the frequency domain," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1867–1875, Nov. 2005.

[9] N. Benvenuto, R. Dinis, F. Falconer, and S. Tomasin, "Single carrier modulation with nonlinear frequency domain equalization: An idea whose time has come—again," *Proc. IEEE*, vol. 98, no. 1, pp. 69–96, Jan. 2010.

[10] C. Zhang, Z. Wang, C. Pan, S. Chen, and L. Hanzo, "Low-complexity iterative frequency domain decision feedback equalization," *IEEE Trans. Veh. Technol.*, vol. 60, no. 3, pp. 1295–1301, Mar. 2011.

[11] M. Honkanen and S.-G. Haggman, "New aspects on nonlinear power amplifier modeling in radio communication system simulations," in *Proc. PIMRC*, vol. 3. Helsinki, Finland, Sep. 1997, pp. 844–848.

[12] C.-S. Choi *et al. RF Impairment Models 60 GHz Band SYS/PHY Simulation*, IEEE Standard 802.15-06-0477-01-003c, Nov. 2006. [Online]. Available: https://mentor.ieee.org/802.15/dcn/06/15-06-0477-01-003c-rf-impairment-models-60ghz-band-sysphy-simulation.pdf

[13] V. Erceg *et al.*, *60 GHz Impairments Modeling*, IEEE Standard 802.11-09/1213r1, Nov. 2009.

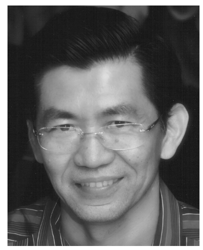
[14] C. de Boor, *A Practical Guide to Splines*. New York, NY, USA: Springer-Verlag, 1978.

[15] C. J. Harris, X. Hong, and Q. Gan, *Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach*. Berlin, Germany: Springer-Verlag, 2002.

[16] S. Chen, X. Hong, E. Khalaf, A. Morfeq, and N. D. Alotaibi, "Adaptive B-spline neural network based nonlinear equalization for high-order QAM systems with nonlinear transmit high power amplifier," *Digit. Signal Process.*, vol. 40, pp. 238–249, May 2015.

[17] A. V. Ivanov, "An asymptotic expansion for the distribution of the least squares estimator of the non-linear regression parameter," *Theory Probab. Appl.*, vol. 21, no. 3, pp. 557–570, Jul. 1977.

[18] C.-F. Wu, "Asymptotic theory of nonlinear least squares estimation," *Ann. Statist.*, vol. 9, no. 3, pp. 501–513, May 1981.



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

From 1986 to 1999, he held research and academic appointments at the University of Sheffield, Sheffield, U.K., University of Edinburgh, Edinburgh, U.K., and University of Portsmouth, Portsmouth, U.K.. Since 1999, he has been with the Electronics and Computer Science Department, University of Southampton, Southampton, U.K., where he is currently a Professor in Intelligent Systems and Signal Processing. He has authored over 550 research papers. His 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 United Kingdom Royal Academy of Engineering, a Fellow of Institution of Engineering and Technology, and a Distinguished Adjunct Professor at King Abdulaziz University, Jeddah, Saudi Arabia, and an ISI Highly Cited Researcher in Engineering in 2004.

Xia Hong (SM'02) received the B.Sc. and M.Sc. degrees from the National University of Defense Technology, Changsha, 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, and 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 Department of Computer Science, School of Mathematical, Physical and Computational Sciences, University of Reading, Reading, U.K. She has authored over 100 research papers and coauthored a research book. Her research interests include nonlinear systems identification, data modeling, estimation and intelligent control, neural networks, pattern recognition, learning theory, and their applications.

Prof. Hong was a recipient of a Donald Julius Groen Prize by IMechE in 1999.

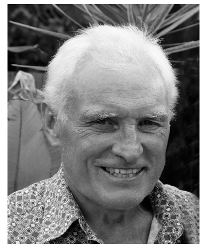
Emad F. Khalaf received the B.Eng. and M.Eng. degrees in information technology from the Wrocław University of Science and Technology, Wrocław, Poland, in 1992, as one certificate, and the Ph.D. degree in computer networks from the Wrocław University of Science and Technology in 2002.

From 2003 to 2011, he was an Assistant Professor with the Computer Engineering Department, Faculty of Engineering, Philadelphia University, Jordan. Since 2012, he has been an Assistant Professor with the Electrical and Computer Engineering Department, Faculty of Engineering, King Abdulaziz University. His current research interests include network security and cryptography, speech classification, and recognition.

Fuad E. Alsaadi received the B.S. and M.Sc. degrees in electronic and communication from King Abdulaziz University, Jeddah, Saudi Arabia, in 1996 and 2002, respectively, and the Ph.D. degree in optical wireless communication systems from the University of Leeds, Leeds, U.K., in 2011.

From 1996 to 2005, he was a Communication Instructor with the College of Electronics and Communication, Jeddah. He was a Lecturer with the Faculty of Engineering, King Abdulaziz University, Jeddah, Saudi Arabia, in 2005. He is currently an Associate Professor of the Electrical and Computer Engineering Department, Faculty of Engineering, King Abdulaziz University.

Dr. Alsaadi published widely in the top IEEE communications conferences and journals and received the Carter Award from the University of Leeds for the best Ph.D. His research interests include optical systems and networks, signal processing, network optimization, synchronization, and systems design and control.



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

He is an Emeritus Research Professor with the University of Southampton, having previously held senior academic appointments at Imperial College London, London, U.K., University of Oxford, Oxford, U.K., and University of Manchester Institute of Science and Technology, Manchester, U.K., as well as the Deputy Chief Scientist for the U.K. Government.

Prof. Harris was a recipient of 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. He is the coauthor of over 450 scientific research papers during a 45-year research career.