Complex-Valued B-Spline Neural Network and its Application to Iterative Frequency-Domain Decision Feedback Equalization for Hammerstein Communication Systems

Sheng Chen, Xia Hong, Emad Khalaf, Fuad E. Alsaadi and Chris J. Harris

Abstract— **Complex-valued (CV) B-spline neural network approach offers a highly effective means for identification and inversion of Hammerstein systems. Compared to its conventional CV polynomial based counterpart, CV B-spline neural network has superior performance in identifying and inverting CV Hammerstein systems, while imposing a similar complexity. In this paper, we review the optimality of CV B-spline neural network approach and demonstrate its excellent approximation capability for a real-world application. More specifically, we develop a CV B-spline neural network based approach for the nonlinear iterative frequency-domain decision feedback equalization (NIFDDFE) of single-carrier Hammerstein channels. Advantages of B-spline neural network approach as compared to polynomial based modeling approach are extensively discussed, and the effectiveness of CV neural network based NIFDDFE is demonstrated in a simulation study.**

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 grey-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 is the nonlinear transmit HPA, and the linear dynamic subsystem is the dispersive channel which can usually be modeled as a finite-duration impulse response (FIR) filter.

CV B-spline neural network has been applied as an effective means for identification and inversion of CV Hammerstein systems [1]–[3]. Compared to its conventional polynomial based counterpart, B-spline models are proven 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. This paper reviews the CV B-spline neural network model as an effective means for identifying and inverting practical Hammerstein systems. We analyze its optimal robustness property, and provide the computational complexity required to calculate the output of a B-spline model, which turns out to be similar to that of the conventional polynomial model.

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 channels. The existing CV B-spline based nonlinear frequency-domain equalization (NFDE) scheme [3] becomes naturally the first iteration of this new NIFDDFE whereby no past detected data are available yet. Therefore, our new NIFDDFE design significantly outperforms the previous NFDE design. We also 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.

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 $(\ )^{\mathrm{T}}$ and $(\ )^{\mathrm{H}}$, respectively, while $(\ )^{-1}$ stands for the inverse operation and $(\ )^*$ denotes the conjugate operation. Furthermore, the expectation operator is denoted by $\mathrm{E}\{\ \}$.

II. NIFDDFE for Hammerstein Channels

We begin by introducing our application senario, the SC block transmission system [7]–[9], where each transmit block consists of N data symbols with M-quadrature amplitude modulation (QAM) expressed as

$$\boldsymbol{x} = \begin{bmatrix} x_0 & x_1 \cdots x_{N-1} \end{bmatrix}^{\mathrm{T}}, \tag{1}$$

where x_k, $0 \le k \le N-1$, take the values from the 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 $2d$ being the minimum distance between symbol points. Adding the cyclic prefix (CP) of length N_{cp} to $\boldsymbol{x}$ yields

$$\bar{\boldsymbol{x}} = \begin{bmatrix} x_{-N_{\mathrm{cp}}} & x_{-N_{\mathrm{cp}}+1} \cdots x_{-1} \mid \boldsymbol{x}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{3}$$

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

$$\bar{\boldsymbol{w}} = \begin{bmatrix} w_{-N_{\mathrm{cp}}} & w_{-N_{\mathrm{cp}}+1} \cdots w_{-1} \mid \boldsymbol{w}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{4}$$

where $\boldsymbol{w} = \begin{bmatrix} w_0 & w_1 \cdots w_{N-1} \end{bmatrix}^{\mathrm{T}}$ and

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

S. Chen (sqc@ecs.soton.ac.uk) and C.J. Harris (cjh@ecs.soton.ac.uk) are with Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK. S. Chen is also with King Abdulaziz University, Jeddah 21589, Saudi Arabia.

X. Hong (x.hong@reading.ac.uk) is with School of Systems Engineering, University of Reading, Reading RG6 6AY, UK.

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

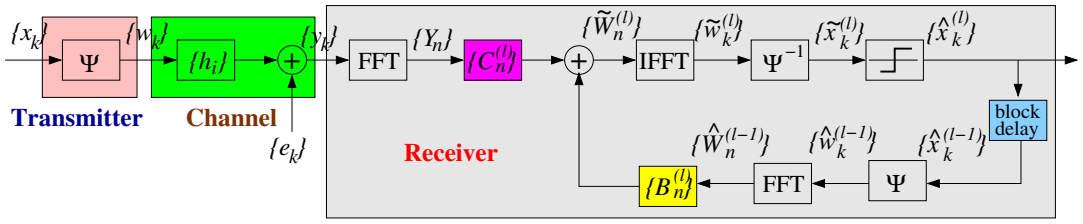


Fig. 1. System schematic of the NIFDDFE for SC Hammerstein communication systems with the nonlinear HPA Ψ at transmitter.

in which $\Psi(\ )$ represents the CV static nonlinearity of HPA, and $w_{-k} = w_{N-k}$ for $1 \le k \le N_{\rm cp}$. Typical HPA in transmitter is the solid state power amplifier [10]–[12], whose nonlinearity $\Psi(\ )$ is constituted by the HPA's amplitude response $A(r)$ and phase response $\Upsilon(r)$ given by

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

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

where r denotes the amplitude of the input to HPA, g_a is the small gain signal, β_a is the smoothness factor and $A_{\rm sat}$ 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 [11], [12]

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

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

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

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 $P_{\rm max}$ of the HPA to the average output power $P_{\rm aop}$ of the HPA output signal, given by

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

The smaller 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

$$\boldsymbol{h} = \begin{bmatrix} h_0 & h_1 \cdots h_{L_{\rm cir}} \end{bmatrix}^{\rm T}. \tag{11}$$

The CIR length satisfies $L_{\rm cir} \le N_{\rm cp}$. It is assumed that $h_0 = 1$ because if this is not the case, h_0 can always be absorbed into the CV nonlinearity $\Psi(\ )$, and the CIR coefficients are re-scaled as h_i/h_0 for $0 \le i \le L_{\rm cir}$. The combined transmission channel and transmitter, as seen 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 receiver, after CP removal, the channel-impaired received signals y_k are given by

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

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 depicted in Fig. 1. First, passing $\boldsymbol{y} = \begin{bmatrix} y_0 & y_1 \cdots y_{N-1} \end{bmatrix}^{\rm T}$ through the N-point fast Fourier transform (FFT) processor yields the frequency-domain (FD) received signal block $\boldsymbol{Y} = \begin{bmatrix} Y_0 & Y_1 \cdots Y_{N-1} \end{bmatrix}^{\rm T}$ with elements

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

Due to the well-known circular property of CP [7]–[9],

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

in which Ξ_n is the FD AWGN with $\mathrm{E}\{|\Xi_n|^2\} = 2\sigma_e^2$, and $\boldsymbol{W} = \begin{bmatrix} W_0 & W_1 \cdots W_{N-1} \end{bmatrix}^{\rm T}$ is the N-point FFT of $\boldsymbol{w}$, i.e.,

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

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

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

The proposed NIFDDFE involves an iterative detection procedure with the iteration index $l \ge 1$. Typically 2 to 4 iterations are sufficient. Specifically, let the FD feedforward and feedback equalizers coefficients at the lth iteration be $\{C_n^{(l)}\}_{n=0}^{N-1}$ and $\{B_n^{(l)}\}_{n=0}^{N-1}$, respectively. Further 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

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

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

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

For the time being 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

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

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.

For the linear iterative FD decision feedback equalisation (LIFDDFE), i.e., the HPA is linear and $w_k = x_k$, $\{C_n^{(l)}\}_{n=0}^{N-1}$ 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 NIFDDFE also seems to yield poor performance. However, we find the extension of the low-complexity simplified LIFDDFE design [9] works well with some modification. 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

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

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

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

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

with

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

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

For the LIFDDFE, the work [9] finds that the performance is insensitive to the predefined SNR value $\mathrm{SNR}_{\mathrm{pre}}$ and the predefined symbol error probability $P_{\mathrm{e,pre}}$. In particular, $\mathrm{SNR}_{\mathrm{pre}}^{-1} = 0.1$ and $P_{\mathrm{e,pre}} = 0.1$ yields excellent results. In our NIFDDFE, we also find that $\mathrm{SNR}_{\mathrm{pre}}^{-1} = 0.1$ and $P_{\mathrm{e,pre}} = 0.1$ are appropriate. In the LIFDDFE case, i.e., $w_k = x_k$, β is a parameter depending on the modulation scheme for x_k. Specifically, $\beta = 2$, $2/5$ 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 $\beta = 0.01$ 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 [9], $B_n = -\big(C_n H_n - \gamma\big)$. But we find with B_n of (22), the performance is better for the NIFDDFE.

III. CV B-Spline Approach for 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. Complex-valued B-spline neural network approach

The CV B-spline neural network approach [1]–[3] offers an effective means for identifying and inverting this Hammerstein channel. Before introducing the B-spline model of $\Psi(\ )$, we point out that $\Psi(\ )$ meets the following conditions.
1) $\Psi(\ )$ is a one-to-one mapping, i.e., a continuous and invertible function.
2) x_R and x_I are upper and lower bounded by some known finite real values, where $x = x_R + \mathrm{j}x_I$ is the input to $\Psi(\ )$, and the distributions of x_R and x_I are identical.

According to the property 2), we have $U_{\min} < x_{\mathrm{s}} < U_{\max}$, where $U_{\min}$ and $U_{\max}$ are known finite real values, while x_{s} represents either x_R or x_I, namely, the subscript s is either R or I. 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 $(N_{\mathrm{s}} + P_o + 1)$ knot values, $\{U_0, U_1, \cdots, U_{N_{\mathrm{s}}+P_o}\}$, with

$$\begin{aligned} U_0 < U_1 < \cdots < U_{P_o-2} < U_{P_o-1} = U_{\min} < U_{P_o} < \cdots < \\ U_{N_{\mathrm{s}}} < U_{N_{\mathrm{s}}+1} = U_{\max} < U_{N_{\mathrm{s}}+2} < \cdots < U_{N_{\mathrm{s}}+P_o}. \end{aligned} \tag{25}$$

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 $N_{\mathrm{s}} + 1 - P_o$. Given the set of predetermined knots (25), the set of N_{s} B-spline basis functions can be formed by using the famous De Boor recursion [13], yielding for $1 \le l \le N_{\mathrm{s}} + P_o$,

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

as well as for $l = 1, \cdots, N_{\mathrm{s}} + P_o - p$ and $p = 1, \cdots, P_o$,

$$\begin{aligned} B_l^{(\mathrm{s},p)}(x_{\mathrm{s}}) = &\frac{x_{\mathrm{s}} - U_{l-1}}{U_{p+l-1} - U_{l-1}} B_l^{(\mathrm{s},p-1)}(x_{\mathrm{s}}) \\ &+ \frac{U_{p+l} - x_{\mathrm{s}}}{U_{p+l} - U_l} B_{l+1}^{(\mathrm{s},p-1)}(x_{\mathrm{s}}). \end{aligned} \tag{27}$$

Using the tensor product between the two sets of univariate B-spline basis functions [14], $B_l^{(R,P_o)}(x_R)$ 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

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

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

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

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

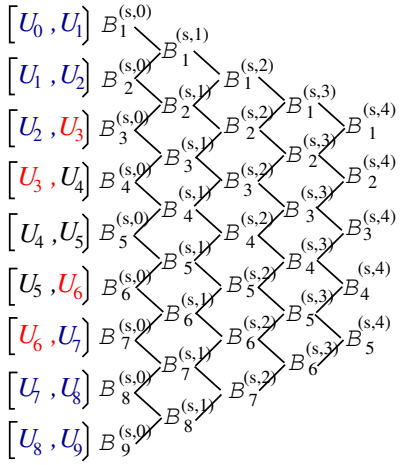


Fig. 2. De Boor recursion: $P_o = 4$, $N_\mathrm{s} = 5$, $U_{\min} = U_3$ and $U_{\max} = U_6$.

B-spline structure parameters: De Boor recursion is illustrated in Fig. 2. $P_o = 4$ is 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. $N_R = N_I = N_\mathrm{s} = 6$ to 10 is sufficient for accurately modeling on the finite interval $\big[U_{\min},\ U_{\max}\big]$. The $N_\mathrm{s} + 1 - P_o$ internal knots may be uniformly spaced in the interval $\big[U_{\min},\ U_{\max}\big]$. Note that there exist no data for $x_\mathrm{s} < U_{\min}$ and $x_\mathrm{s} > U_{\max}$ in identification but it is desired that the B-spline model has certain extrapolating capability outside the interval $\big[U_{\min},\ U_{\max}\big]$. 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.

Optimal robustness property: 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. Although the conventional polynomial modeling with polynomial degree P_o, as defined by the set of $P_o + 1$ basis functions

$$1,\ x_\mathrm{s},\ x_\mathrm{s}^2, \cdots, x_\mathrm{s}^{P_o}, \tag{30}$$

can be used to approximate a continuous function, 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 to one. We use the simple example of [15] to demonstrate the excellent numerical stability of the B-spline model over the polynomial model of (30) in Fig. 3.

Fig. 3 (a) plots a quadratic polynomial function $y = 0.001x^2 - 0.02x + 0.1$ defined over $x \in [0,\ 20]$ in solid line. With the knot sequence $\{-5, -4, 0, 20, 24, 25\}$, this function is modeled as a quadratic B-spline model of $y = 0.14B_1^{(\mathrm{s},2)}(x) - 0.10B_2^{(\mathrm{s},2)}(x) + 0.14B_3^{(\mathrm{s},2)}(x)$, which is depicted in Fig. 3 (b) in solid line. 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. 3 (a) and (b) depict the ten sets of the perturbed functions in dashed line generated by perturbing the two models' parameters. It can be clearly seen from Fig. 3 (a) that the polynomial model is seriously perturbed, but there is no noticeable change at all in Fig. 3 (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 can be seen from Fig. 3 (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. 3 (d). Observe from Fig. 3 (a) and (d) that, despite of 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 much less seriously perturbed than the polynomial model.

Complexity analysis: Complexity does not depend on the number of basis functions N_s. Given $x_\mathrm{s} \in \big[U_{\min},\ U_{\max}\big]$, there are only $P_o + 1$ basis functions with nonzero values at most. Fig. 4 illustrates 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 $(P_o + 1)^2$ non-zero basis functions at most for any given input. Complexity of computing the B-spline model (28) is therefore on the order of $\mathsf{O}\big((P_o + 1)^2\big)$. The upper bound complexity for $P_o = 4$ is listed in Table I, which includes generating the two sets of B-spline basis functions for real and imaginary parts, respectively, and the output of the tensor-product B-spline model (28). This is in fact comparable to the conventional polynomial modeling. For the polynomial model with polynomial degree P_o, there are $P_o + 1$ basis functions as given in (30). Thus, the tensor-

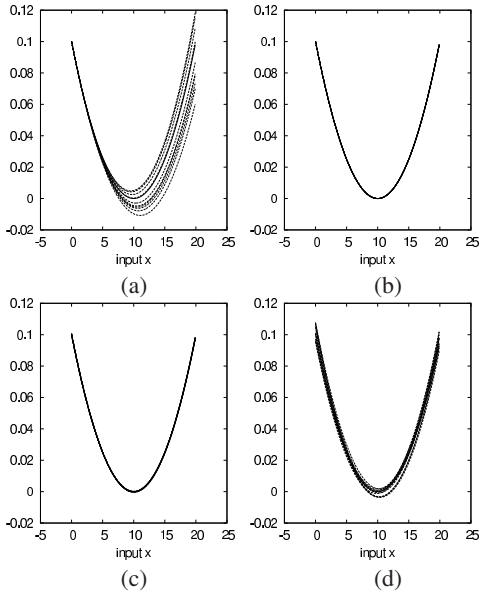
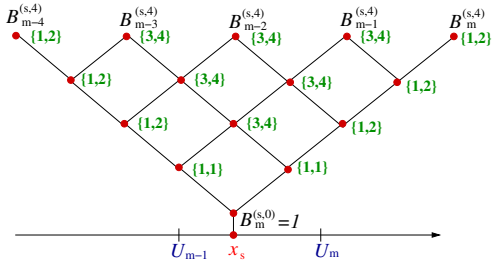
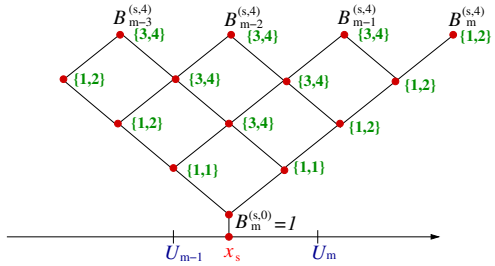


Fig. 3. (a) Polynomial model with UDRN perturbation noises drawn from $[-0.0001,\ 0.0001]$, (b) B-spline model with UDRN perturbation noises drawn from $[-0.0001,\ 0.0001]$, (c) B-spline model with UDRN perturbation noises drawn from $[-0.001,\ 0.001]$, and (d) B-spline model with UDRN perturbation noises drawn from $[-0.01,\ 0.01]$.



(a) $P_o+1 \le m \le N_s$



(b) $m = P_o$

Fig. 4. 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.

product polynomial model have $(P_o+1)^2$ non-zero basis functions for any given input, and the complexity of the polynomial model is also on the order of $\mathsf{O}\big((P_o+1)^2\big)$.

B. Identifying Hammerstein channel

Given a block of N training data, $\{x_k, y_k\}_{k=0}^{N-1}$, the identification task is to minimize the cost function

$$J(\boldsymbol{h},\boldsymbol{\theta}) = \frac{1}{N}\sum_{k=0}^{N-1}\left|\widehat{e}_k\right|^2 = \frac{1}{N}\sum_{k=0}^{N-1}\left|y_k-\widehat{y}_k\right|^2, \tag{31}$$

subject to the constraint of $h_0 = 1$, in which

$$\widehat{y}_k = \sum_{i=0}^{L_{\rm cir}} h_i \widehat{w}_{k-i} = \sum_{i=0}^{L_{\rm cir}} h_i \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(x_{k-i})\theta_{l,m}, \tag{32}$$

where $x_{k-i} = x_{N+k-i}$ if $k < i$. The cost function (31) is convex with respect to $\boldsymbol{h}$ when fixing $\boldsymbol{\theta}$, and convex with respect to $\boldsymbol{\theta}$ given $\boldsymbol{h}$. According to [16], [17], the estimates of $\boldsymbol{\theta}$ and $\boldsymbol{h}$ are unbiased, irrespective to the algorithm used to minimize the cost function (31). In [15], 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. This ALS procedure is summarized below.

Initialisation. Define the amalgamated parameter vector

$$\boldsymbol{\omega} = \left[\boldsymbol{\theta}^{\rm T}\ h_1\boldsymbol{\theta}^{\rm T}\ h_2\boldsymbol{\theta}^{\rm T}\cdots h_{L_{\rm cir}}\boldsymbol{\theta}^{\rm T}\right]^{\rm T} \in \mathbb{C}^{(L_{\rm cir}+1)N_B}. \tag{33}$$

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

$$\boldsymbol{P} = \begin{bmatrix} \boldsymbol{\phi}^{\rm T}(0) & \boldsymbol{\phi}^{\rm T}(-1) & \cdots & \boldsymbol{\phi}^{\rm T}(-L_{\rm cir}) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\phi}^{\rm T}(k) & \boldsymbol{\phi}^{\rm T}(k-1) & \cdots & \boldsymbol{\phi}^{\rm T}(k-L_{\rm cir}) \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{\phi}^{\rm T}(N-1) & \boldsymbol{\phi}^{\rm T}(N-2) & \cdots & \boldsymbol{\phi}^{\rm T}(N-1-L_{\rm cir}) \end{bmatrix}, \tag{34}$$

TABLE I

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

Computation	Multiplications	Additions
Two sets of 1-D basis functions	2×38	2×26
Output of (28)	2×25	24
Total	126	76

with $\boldsymbol{\phi}(k) = \left[\phi_{1,1}(k)\ \phi_{1,2}(k)\cdots\phi_{l,m}(k)\cdots\phi_{N_R,N_I}(k)\right]^{\rm T}$, 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 least squares (LS) estimate of $\boldsymbol{\omega}$ is $\widehat{\boldsymbol{\omega}} = \left(\boldsymbol{P}^{\rm T}\boldsymbol{P}\right)^{-1}\boldsymbol{P}^{\rm T}\boldsymbol{y}$, and the first N_B elements of $\widehat{\boldsymbol{\omega}}$ provide an initial estimate for $\boldsymbol{\theta}$, which is denoted as $\widehat{\boldsymbol{\theta}}^{(0)}$.

ALS estimation procedure. For $1 \le \tau \le \tau_{\max}$, e.g., $\tau_{\max} = 4$, perform:

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

$$\boldsymbol{Q} = \begin{bmatrix} \widehat{w}_0 & \widehat{w}_{-1} & \cdots & \widehat{w}_{-L_{\rm cir}} \\ \vdots & \vdots & \vdots & \vdots \\ \widehat{w}_k & \widehat{w}_{k-1} & \cdots & \widehat{w}_{k-L_{\rm cir}} \\ \vdots & \vdots & \vdots & \vdots \\ \widehat{w}_{N-1} & \widehat{w}_{N-2} & \cdots & \widehat{w}_{N-1-L_{\rm cir}} \end{bmatrix}, \tag{35}$$

in which

$$\widehat{w}_k = \widehat{\Psi}(x_k) = \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(x_k)\widehat{\theta}_{l,m}^{(\tau-1)}. \tag{36}$$

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

$$\underline{\widehat{\boldsymbol{h}}}^{(\tau)} = \left(\boldsymbol{Q}^{\rm H}\boldsymbol{Q}\right)^{-1}\boldsymbol{Q}^{\rm H}\boldsymbol{y}, \tag{37}$$

$$\widehat{h}_i^{(\tau)} = \underline{\widehat{h}}_i^{(\tau)} \big/ \underline{\widehat{h}}_0^{(\tau)},\ 0 \le i \le L_{\rm cir}. \tag{38}$$

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

$$\varphi_{l,m}(k) = \sum_{i=0}^{L_{\rm cir}} \widehat{h}_i^{(\tau)} B_{l,m}^{(P_o)}(x_{k-i}) \in \mathbb{C}. \tag{39}$$

Further define the regression matrix

$$\boldsymbol{S} = [\boldsymbol{\varphi}(0)\ \boldsymbol{\varphi}(1)\cdots\boldsymbol{\varphi}(N-1)]^{\rm T} \in \mathbb{C}^{N\times N_B}, \tag{40}$$

with $\boldsymbol{\varphi}(k) = [\varphi_{1,1}(k)\ \varphi_{1,2}(k)\cdots\varphi_{l,m}(k)\cdots\varphi_{N_R,N_I}(k)]^{\rm T}$. The LS estimate $\widehat{\boldsymbol{\theta}}^{(\tau)}$ is given by $\widehat{\boldsymbol{\theta}}^{(\tau)} = \left(\boldsymbol{S}^{\rm H}\boldsymbol{S}\right)^{-1}\boldsymbol{S}^{\rm H}\boldsymbol{y}$.

C. Inverting HPA's nonlinearity

We utilize another B-spline neural network to model the inverse mapping of the HPA's CV nonlinearity defined by

$$x_k = \Psi^{-1}(w_k) = \Phi(w_k). \tag{41}$$

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

$$\widehat{x} = \widehat{\Phi}(w;\boldsymbol{\alpha}) = \sum_{l=1}^{N_R}\sum_{m=1}^{N_I} B_l^{(R,P_o)}(w_R)B_m^{(I,P_o)}(w_I)\alpha_{l,m}, \tag{42}$$

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

$$\boldsymbol{\alpha} = \left[\alpha_{1,1}\ \alpha_{1,2}\cdots\alpha_{l,m}\cdots\alpha_{N_R,N_I}\right]^{\rm T} \in \mathbb{C}^{N_B}. \tag{43}$$

TABLE III

IDENTIFICATION RESULTS FOR THE CIR COEFFICIENT VECTOR $\boldsymbol{h}$ OF THE HAMMERSTEIN CHANNEL.

Tap No.	True parameters	Estimated parameters			
		OBO $= 3$ dB $E_{\rm x}/N_{\rm o} = 5$ dB	OBO $= 3$ dB $E_{\rm x}/N_{\rm o} = 10$ dB	OBO $= 5$ dB $E_{\rm x}/N_{\rm o} = 5$ dB	OBO $= 5$ dB $E_{\rm x}/N_{\rm o} = 10$ dB
h_0	1	1	1	1	1
h_1	$-0.3732 - {\rm j}0.6123$	$-0.3722 - {\rm j}0.6116$	$-0.3724 - {\rm j}0.6118$	$-0.3722 - {\rm j}0.6116$	$-0.3726 - {\rm j}0.6119$
h_2	$0.3584 + {\rm j}0.3676$	$0.3592 + {\rm j}0.3689$	$0.3590 + {\rm j}0.3683$	$0.3594 + {\rm j}0.3685$	$0.3589 + {\rm j}0.3681$
h_3	$0.3052 + {\rm j}0.2053$	$0.3049 + {\rm j}0.2053$	$0.3051 + {\rm j}0.2052$	$0.3049 + {\rm j}0.2051$	$0.3050 + {\rm j}0.2052$
h_4	$0.2300 + {\rm j}0.1287$	$0.2301 + {\rm j}0.1289$	$0.2302 + {\rm j}0.1288$	$0.2301 + {\rm j}0.1287$	$0.2301 + {\rm j}0.1287$
h_5	$0.7071 + {\rm j}0.7071$	$0.7073 + {\rm j}0.7083$	$0.7072 + {\rm j}0.7077$	$0.7073 + {\rm j}0.7080$	$0.7072 + {\rm j}0.7076$
h_6	$0.6123 - {\rm j}0.3732$	$0.6122 - {\rm j}0.3738$	$0.6123 - {\rm j}0.3734$	$0.6121 - {\rm j}0.3736$	$0.6122 - {\rm j}0.3734$
h_7	$-0.3584 + {\rm j}0.3676$	$-0.3607 + {\rm j}0.3685$	$-0.3595 + {\rm j}0.3681$	$-0.3599 + {\rm j}0.3684$	$-0.3593 + {\rm j}0.3681$
h_8	$-0.2053 - {\rm j}0.3052$	$-0.2073 - {\rm j}0.3057$	$-0.2063 - {\rm j}0.3054$	$-0.2068 - {\rm j}0.3054$	$-0.2062 - {\rm j}0.3053$
h_9	$0.1287 - {\rm j}0.2300$	$0.1277 - {\rm j}0.2303$	$0.1283 - {\rm j}0.2301$	$0.1281 - {\rm j}0.2301$	$0.1284 - {\rm j}0.2300$

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

Over the pseudo training data set $\left\{\widehat{w}_k, x_k\right\}_{k=0}^{N-1}$, the regression matrix $\boldsymbol{B} \in \mathbb{R}^{N \times N_B}$ can be formed as

$$\boldsymbol{B} = \begin{bmatrix} B_{1,1}^{(P_o)}(\widehat{w}_0) & B_{1,2}^{(P_o)}(\widehat{w}_0) & \cdots & B_{N_R,N_I}^{(P_o)}(\widehat{w}_0) \\ B_{1,1}^{(P_o)}(\widehat{w}_1) & B_{1,2}^{(P_o)}(\widehat{w}_1) & \cdots & B_{N_R,N_I}^{(P_o)}(\widehat{w}_1) \\ \vdots & \vdots & \vdots & \vdots \\ B_{1,1}^{(P_o)}(\widehat{w}_{N-1}) & B_{1,2}^{(P_o)}(\widehat{w}_{N-1}) & \cdots & B_{N_R,N_I}^{(P_o)}(\widehat{w}_{N-1}) \end{bmatrix}, \tag{44}$$

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

IV. SIMULATION STUDY

We considered a 64-QAM Hammerstein channel in which the HPA was described by (6) and (7) with the parameter set given in (8). The size of the transmitted data block was $N = 2048$. The piecewise quartic polynomial of $P_o = 4$ was chosen, and the number of B-spline basis functions was set to $N_R = N_I = 8$. The knot sequences adopted by the two CV B-spline neural networks for identifying and inverting the HPA's nonlinearity are listed in Table II. The dispersive channel had 10 taps ($L_{\rm cir} = 9$) whose CIR coefficients are given in Table III. The system's signal-to-noise ratio (SNR) was defined as SNR $= E_{\rm x}/N_{\rm o}$, where $E_{\rm x}$ was the average power of the input signal x_k to the HPA and $N_{\rm o} = 2\sigma_e^2$.

TABLE II

KNOT SEQUENCES.

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

The effectiveness of the proposed CV B-spline neural network approach to identify this Hammerstein channel is

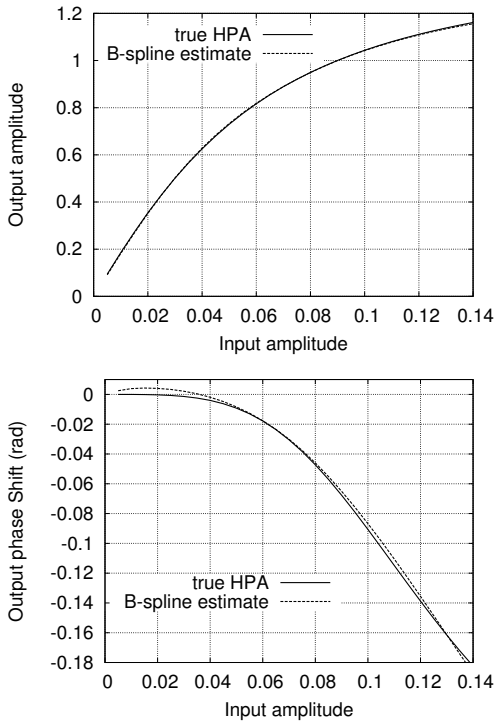


Fig. 5. Comparison of the HPA's nonlinearity $\Psi(\,)$ and its B-spline estimate $\widehat{\Psi}(\,)$ under OBO$= 3$ dB and $E_{\rm x}/N_{\rm o} = 5$ dB.

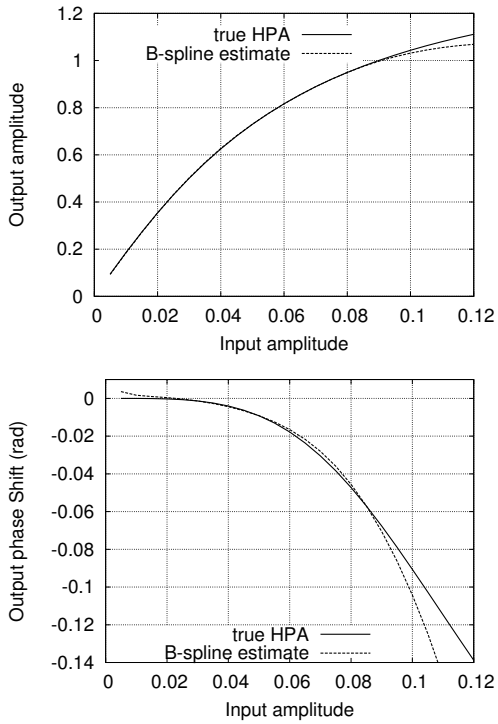


Fig. 6. Comparison of the HPA's nonlinearity $\Psi(\,)$ and its B-spline estimate $\widehat{\Psi}(\,)$ under OBO$= 5$ dB and $E_{\rm x}/N_{\rm o} = 10$ dB.

demonstrated in Table III as well as 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 $E_{\rm x}/N_{\rm o}=5\,$dB. Note that under the HPA operational condition of OBO$=5\,$dB, the peak amplitude of $|x_k|$ was less less than 0.09, while under the condition of OBO$=3\,$dB, the peak amplitude of $|x_k|$ was less less than 0.14. The results of Figs. 5 and 6 clearly demonstrate the capability of the proposed CV B-spline neural network to accurately model the HPA's nonlinearity, within the HPA's operational input range. The combined responses of the HPA's true nonlinearity and its estimated inversion obtained by the proposed B-spline inverting scheme under the two operating conditions are depicted in Figs. 7 and 8. 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 pseudo training data. More specifically, the results of Figs. 7 and 8 clearly indicate that the combined response of the true HPA's nonlinearity $\Psi(\ )$ and its estimated inversion $\widehat{\Phi}(\ )$ satisfies $\widehat{\Phi}\big(\Psi(x)\big)\approx x$. That is, the magnitude of the combined response is $\big|\widehat{\Phi}\big(\Psi(x)\big)\big|\approx|x|$ and the phase shift of the combined response is approximately zero. In other words, $\widehat{\Phi}(\ )$ is an accurate inversion of $\Psi(\ )$.

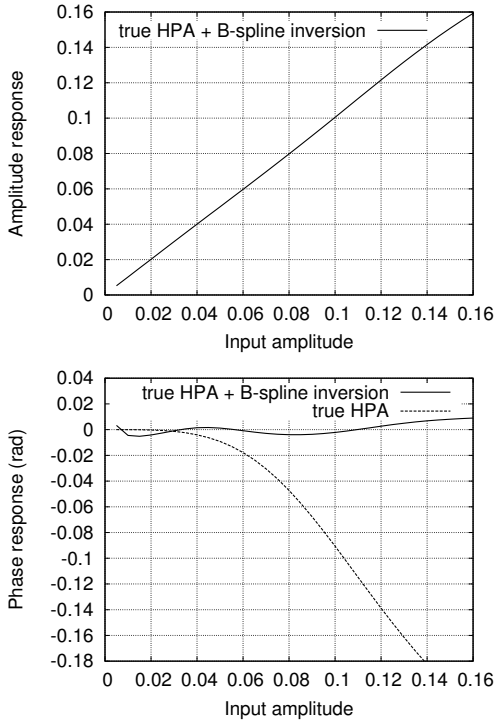


Fig. 7. Combined response of the true HPA and its estimated B-spline inversion under OBO$=3\,$dB and $E_{\rm x}/N_{\rm o}=5\,$dB.

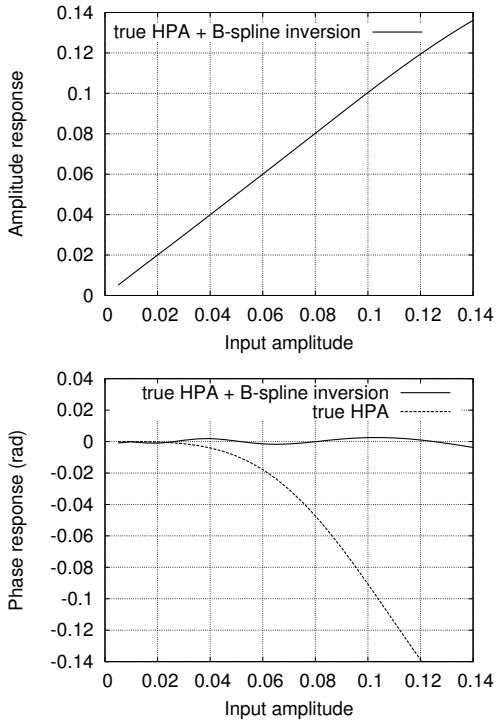


Fig. 8. Combined response of the true HPA and its estimated B-spline inversion under OBO$=5\,$dB and $E_{\rm x}/N_{\rm o}=10\,$dB.

We also used two tensor-product polynomial models, both having a polynomial degree of $P_o=4$ in each dimension, to estimate the CV HPA's nonlinearity $\Psi(\ )$ and its inversion $\Psi^{-1}(\ )$, respectively, based on the same identification procedure of Sections III-B and III-C. Each tensor-product polynomial model had 25 basis functions which was comparable to the tensor-product B-spline model of at most 25 non-zero basis functions for any given input. As expected, the CIR tap vector was identified with similarly high accuracy but the polynomial based estimate $\widehat{\Psi}(\ )$ was less accurate than the B-spline based estimate. Most strikingly, the polynomial based inversion estimate $\widehat{\Phi}(\ )$ was much less accurate than the B-spline based estimate, as demonstrated in Fig. 9. This is obviously due to the fact that the pseudo identification input $\widehat{w}_k$ is highly noisy, and the polynomial model is much less robust to noise as clearly shown in Section III-A.

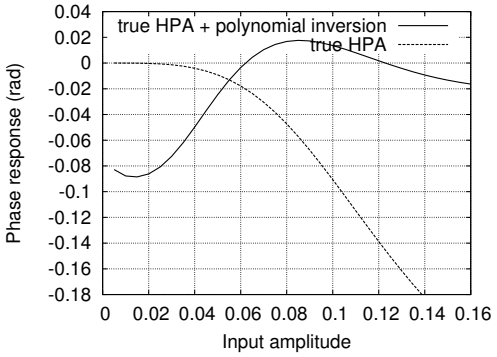


Fig. 9. Combined phase response of the true HPA and its estimated polynomial inversion under OBO$=3\,$dB and $E_{\rm x}/N_{\rm o}=5\,$dB.

The bit error rate (BER) performance of the B-spline based NIFDDFE constructed based on the estimated CIR, HPA and HPA's inversion are plotted in Fig. 10 under the two HPA operating conditions. From Fig. 10, it can be seen that four iterations are sufficient for the NIFDDFE. Since the first iteration of the NIFDDFE is identical to the NFDE, the results of From Fig. 10 confirm that the NIFDDFE significantly outperforms the NFDE. Fig. 11 demonstrates that the B-spline based NIFDDFE significantly outperforms

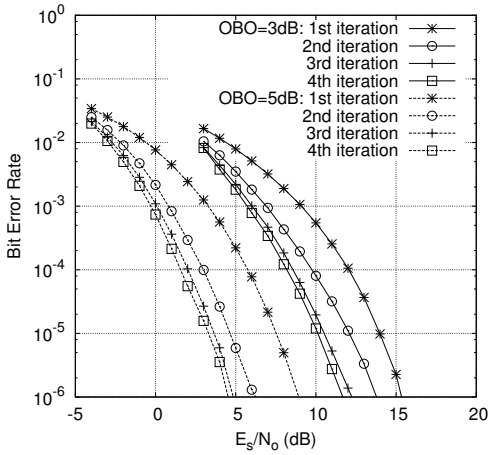
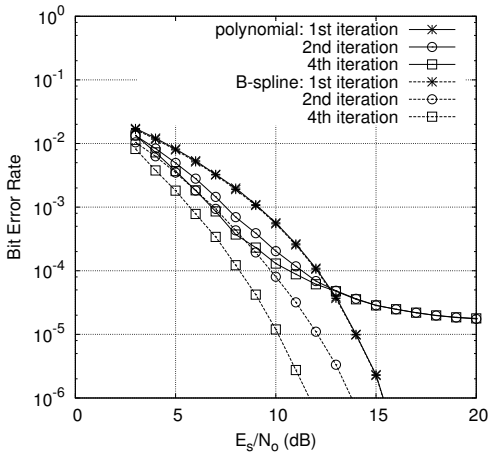
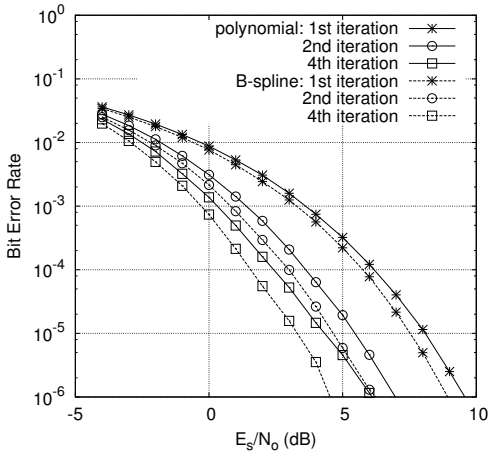


Fig. 10. BER performance of the B-spline based NIFDDFE under the two HPA operating conditions of OBO= 3 dB and OBO= 5 dB.

the polynomial based NIFDDFE, particularly when the HPA is operating in the severe nonlinear region.



(a) OBO= 3 dB



(a) OBO= 5 dB

Fig. 11. BER performance comparison of the B-spline based NIFDDFE and the polynomial based NIFDDFE.

V. Conclusions

This paper has demonstrated that the CV B-spline neural network approach offers a highly effective and accurate means for identifying and inverting Hammerstein channels with nonlinear HPA at transmitter. Optimal robust property of the B-spline modeling has bee reviewed, and it has been shown that the CV B-spline modeling approach has a comparable computational complexity to the conventional CV polynomial modeling approach. The proposed CV B-spline modeling approach has been applied to state-of-the-art iterative frequency-domain decision feedback equalization of Hammerstein communication systems. Simulation results obtained have demonstrated that the CV B-spline based NIFDDFE significantly outperforms the CV polynomial based NIFDDFE design of comparable complexity.

References

[1] S. Chen, X. Hong, J. B. Gao, and C. J. Harris, "Complex-valued B-spline neural networks for modeling and inverting Hammerstein systems," *IEEE Trans. Neural Networks and Learning Systems*, 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. 6-11, 2014, pp. 1834–1841.

[3] X. Hong, S. Chen, C. J. Harris, and E. 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, 2014.

[4] J. M. Pena, "B-spline and optimal stability," *Mathematics of Computation*, vol. 66, no. 220, pp. 1555–1560, Oct. 1997.

[5] T. Lyche and J. M. Pena, "Optimally stable multivariate bases," *Advances in Computational Mathematics*, vol. 20, nos. 1-3, pp. 149–159, Jan. 2004.

[6] E. Mainar and J. M. Pena, "Optimal stability of bivariate tensor product B-bases," *J. Numerical Analysis, Industrial and Applied Mathematics*, vol. 6 nos. 3-4, pp. 95–104, 2011.

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

[10] M. Honkanen and S.-G. Häggman, "New aspects on nonlinear power amplifier modeling in radio communication system simulations," in *Proc. PIMRC'97* (Helsinki, Finland), Sept. 1-4, 1997, pp. 844–848.

[11] C.-S. Choi, *et al.*,"RF impairment models 60 GHz band SYS/PHY simulation," Document IEEE 802.15-06-0477-01-003c, Nov. 2006. `https://mentor.ieee.org/802.15/dcn/06/15-06-0477-01-003c-rf-impairment-models-60ghz-band-sysphy-simulation.pdf`

[12] V. Erceg, *et al.*, "60 GHz impairments modeling," Document IEEE 802.11-09/1213r1, Nov. 2009.

[13] C. De Boor, *A Practical Guide to Splines*. New York: Spring Verlag, 1978.

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

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

[16] A. V. Ivanov, "An asymptotic expansion for the distribution of the least squares estimator of the non-linear regression parameter," *SIAM Theory of Probability & Its Applications*, vol. 21, no. 3, pp. 557–570, 1977.

[17] C.-F. Wu, "Asymptotic theory of nonlinear least squares estimation," *Annals of Statistics*, vol. 9, no. 3, pp. 501–513, 1981.