

Adaptive B-spline neural network based nonlinear equalization for high-order QAM systems with nonlinear transmit high power amplifier [☆]



Sheng Chen ^{a,c,*}, Xia Hong ^b, Emad Khalaf ^c, Ali Morfeq ^c, Naif D. Alotaibi ^c

^a Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK

^b School of Systems Engineering, University of Reading, Reading RG6 6AY, UK

^c Electrical and Computer Engineering Department, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:

Available online 20 February 2015

Keywords:

Quadrature amplitude modulation

Peak-to-average power ratio

Nonlinear high power amplifier

Hammerstein channel

B-spline neural network

Equalization

ABSTRACT

High bandwidth-efficiency quadrature amplitude modulation (QAM) signaling widely adopted in high-rate communication systems suffers from a drawback of high peak-to-average power ratio, which may cause the nonlinear saturation of the high power amplifier (HPA) at transmitter. Thus, practical high-throughput QAM communication systems exhibit nonlinear and dispersive channel characteristics that must be modeled as a Hammerstein channel. Standard linear equalization becomes inadequate for such Hammerstein communication systems. In this paper, we advocate an adaptive B-Spline neural network based nonlinear equalizer. Specifically, during the training phase, an efficient alternating least squares (LS) scheme is employed to estimate the parameters of the Hammerstein channel, including both the channel impulse response (CIR) coefficients and the parameters of the B-spline neural network that models the HPA's nonlinearity. In addition, another B-spline neural network is used to model the inversion of the nonlinear HPA, and the parameters of this inverting B-spline model can easily be estimated using the standard LS algorithm based on the pseudo training data obtained as a natural byproduct of the Hammerstein channel identification. Nonlinear equalisation of the Hammerstein channel is then accomplished by the linear equalization based on the estimated CIR as well as the inverse B-spline neural network model. Furthermore, during the data communication phase, the decision-directed LS channel estimation is adopted to track the time-varying CIR. Extensive simulation results demonstrate the effectiveness of our proposed B-Spline neural network based nonlinear equalization scheme.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

High-order quadrature amplitude modulation (QAM) signaling [1] has found its way into many recent high-rate wireless communications system standards, owing to its desired property of high achievable bandwidth efficiency. The higher the order of QAM signaling, the better the bandwidth efficiency but also the higher the peak-to-average power ratio (PAPR) of the resulting transmit signal. As practical high power amplifiers (HPAs) exhibit nonlinear saturation characteristics [2–6], the high PAPR signal may drive the

HPA at transmitter into the nonlinear saturation region, which will significantly degrade the system's achievable bit error rate (BER) performance. An effective means of compensating the nonlinear distortions of HPA is to implement a digital predistorter at the transmitter, and various predistorter techniques have been developed [7–13], which are capable of achieving excellent performance. However, implementing the predistorter is attractive for the downlink, where the base station (BS) transmitter has the sufficient hardware and software capacities to accommodate the hardware and computational requirements for implementing digital predistorter. By contrast, in the uplink, implementing predistorter at transmitter is much more difficult, because it is extremely challenging for a pocket-size handset to absorb the additional hardware and computational complexity required. Alternatively, in the uplink, the nonlinear distortions of the transmitter HPA can be dealt with at the BS receiver, which has sufficient hardware and software resources.

[☆] This paper was funded by the Deanship of Scientific Research (DSR), King Abdulaziz University, under grant No. (5-135-35-HiCi). The authors, therefore, acknowledge with thanks the DSR technical and financial support.

* Corresponding author.

E-mail addresses: sqc@ecs.soton.ac.uk (S. Chen), x.hong@reading.ac.uk (X. Hong), ekhalaf@kau.edu.sa (E. Khalaf), morfeq@kau.edu.sa (A. Morfeq), ndalotaibi@kau.edu.sa (N.D. Alotaibi).

With the nonlinear HPA at transmitter, the high-order QAM uplink is a complex-valued (CV) nonlinear Hammerstein system and, moreover, the received signal is further impaired by the channel additive white Gaussian noise (AWGN) while the coefficients of the channel impulse response (CIR) are time-varying. Therefore, nonlinear equalization of such a CV time-varying Hammerstein channel is a challenging task. A recent work [14] developed a generic method for identification and inversion of CV stationary Hammerstein systems based on a novel CV B-spline neural network approach. In this paper, we advocate the extension of this CV B-spline neural network based approach to nonlinear equalization of the time-varying QAM uplink with the nonlinear HPA at transmitter. Our original contribution is twofold.

Firstly, during the training phase, we propose a similar approach to the one adopted by [14] to construct a nonlinear equalizer. More specifically, as in [14], a CV B-spline neural network is utilized to model the HPA's nonlinearity, and an efficient alternating least squares (ALS) identification algorithm is employed to estimate the CIR coefficients as well as the parameters of the CV B-spline neural network that models the static nonlinearity of the Hammerstein channel. Nonlinear equalization can then be naturally accomplished by standard linear equalization based on the CIR as well as the inversion of the HPA's nonlinearity. The inversion of the HPA's nonlinearity is implemented in [14] as a root finding problem based on the estimated B-spline neural network, which requires to carry out the iterative root finding procedure for detecting every data symbol and is time-consuming. We adopt a more efficient approach of directly inverting the HPA's nonlinearity. Specifically, we use another CV B-spline neural network to model the inversion of the HPA nonlinearity. Although the HPA's output at the transmitter is unobservable at the receiver for identifying this inverse model, the pseudo training data obtained as a natural byproduct of the Hammerstein channel identification can be used to estimate the parameters of the inverting B-spline model using the standard least squares (LS) algorithm. Secondly, in order to cope with the fading CIR, during the data communication phase, the usual decision-directed (DD) LS channel estimator is employed to track the time-varying CIR coefficients and, therefore, to adapt the equalizer. Extensive simulation results are presented to demonstrate the effectiveness of our proposed B-spline neural network based nonlinear equalizer for combating the detrimental effects of the Hammerstein channel.

The remainder of this paper is organized as follows. Section 2 presents the high-order QAM uplink, where the channel is modelled as a CV Hammerstein system with the nonlinear HPA at the transmitter and the fading CIR, while a nonlinear equalizer is required at the receiver to combat the adverse effects of the Hammerstein channel. Our adaptive B-spline neural network based nonlinear equalizer is detailed in Section 3, and the simulation results are presented in Section 4. Our concluding remarks are offered in Section 5.

Throughout our discussions, a CV number $x \in \mathbb{C}$ is represented either by the rectangular form $x = x_R + jx_I$, where $j = \sqrt{-1}$, while $x_R = \Re[x]$ and $x_I = \Im[x]$ denote the real and imaginary parts of x , or alternatively by the polar form $x = |x| \cdot e^{j\angle x}$ with $|x|$ denoting the amplitude of x and $\angle x$ its phase. $E[\cdot]$ denotes the expectation operator, while $(\cdot)^{-1}$ denotes the inversion. The conjugate operation is denoted by $(\cdot)^*$, while $(\cdot)^T$ and $(\cdot)^H$ represent the transpose and conjugate transpose operators, respectively. The $L \times L$ identity matrix is denoted by I_L .

2. High-order QAM nonlinear uplink

We consider the M -QAM signaling and, therefore, the transmitted symbols $x(k) \in \mathbb{C}$, where k denotes the symbol index, take the values from the M -QAM symbol set

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

with $2d$ being the minimum distance between symbol points.

2.1. The channel model

With the transmitter HPA exhibiting nonlinear saturation characteristics, the high-order QAM uplink can be represented by the generic CV Hammerstein system consisting of a cascade of two subsystems: a CV nonlinear static function $\Psi(\bullet) : \mathbb{C} \rightarrow \mathbb{C}$ that represents the HPA at the transmitter, followed by a CV linear dynamic system with a finite-duration impulse response (FIR) filter of order L_h which represents the dispersive channel. Furthermore, the received signal $y(k) \in \mathbb{C}$ is corrupted by the channel AWGN $n(k) \in \mathbb{C}$. Therefore, $y(k)$ is represented by

$$w(k) = \Psi(x(k)), \quad (2)$$

$$y(k) = \sum_{i=0}^{L_h} h_i w(k-i) + n(k), \quad (3)$$

where the channel AWGN has the power of $E[|n(k)|^2] = 2\sigma_n^2$, and $\mathbf{h} = [h_0 \ h_1 \ \dots \ h_{L_h}]^T$ is the CIR coefficient vector, while the HPA output $w(k)$ is unobserved and, therefore, is unavailable at the receiver.

The most widely used HPA is the solid state power amplifier [5,6], whose nonlinearity $\Psi(\cdot)$ is constituted by the HPA's amplitude response $A(r)$ and phase response $\Upsilon(r)$ given by

$$A(r) = \frac{g_a r}{\left(1 + \left(\frac{g_a r}{A_{\text{sat}}}\right)^{2\beta_a}\right)^{\frac{1}{2\beta_a}}}, \quad (4)$$

$$\Upsilon(r) = \frac{\alpha_\phi r^{q_1}}{1 + \left(\frac{r}{\beta_\phi}\right)^{q_2}} [\text{degree}], \quad (5)$$

where r denotes the amplitude of the input to the HPA, g_a is the small gain signal, β_a is the smoothness factor and A_{sat} is the saturation level, while the parameters of the phase response, α_ϕ , β_ϕ , q_1 and q_2 , are adjusted to match the specific amplifier's characteristics. The NEC GaAs power amplifier used in the recent wireless standards [5,6] for example has the parameter set

$$g_a = 19, \beta_a = 0.81, A_{\text{sat}} = 1.4; \\ \alpha_\phi = -48000, \beta_\phi = 0.123, q_1 = 3.8, q_2 = 3.7. \quad (6)$$

Therefore, given the input $x(k) = |x(k)| \cdot e^{j\angle x(k)}$ to the HPA, the output of the HPA can be expressed as

$$w(k) = A(|x(k)|) \cdot e^{j(\angle x(k) + \Upsilon(|x(k)|))}. \quad (7)$$

The operating status of the HPA may be 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

$$\text{OBO} = 10 \cdot \log_{10} \frac{P_{\text{max}}}{P_{\text{aop}}}. \quad (8)$$

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

To model the time-varying nature of the CIR, it is usually assumed that the CIR taps $\{h_0, h_1, \dots, h_{L_h}\}$ follow the Rayleigh fading distribution with a given fading rate [15]. In this paper, we consider the slow fading scenario. More specifically, as transmission is organized in frames and each frame contains N_F symbols, we assume that within a frame, the CIR taps remain constant, while between frames the CIR taps are faded with the normalized Doppler frequency of f_d .

2.2. The receiver model

To combat the adverse efforts of the dispersive CIR, the standard linear equalizer takes the form of an FIR filter with the order L_g , and the output of this linear equalizer is

$$\widehat{w}(k - \iota) = \sum_{i=0}^{L_g} g_i^* y(k - i), \quad (9)$$

where ι is known as the decision delay. In particular, if the CIR is minimum phase, $\iota = 0$. If the HPA at the transmitter were linear, $\widehat{w}(k - \iota)$ would be a sufficient statistic for estimating the transmitted data symbol $x(k - \iota)$. To guarantee an accurate equalization, the length of the linear equalizer L_g should be chosen to be three to four times of the length of \mathbf{h} , but not too long in order not to amplify the noise in the input signal too much. The well-known minimum mean square error (MMSE) solution [16] can readily be applied to obtain the linear equalizer's weight vector $\mathbf{g} = [g_0 \ g_1 \ \dots \ g_{L_g}]^T$. Define the CIR coefficient matrix $\mathbf{H} \in \mathbb{C}^{(L_g+1) \times (L_H+1)}$

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \dots & h_{L_H} & 0 & \dots & 0 \\ 0 & h_0 & h_1 & \dots & h_{L_H} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & h_0 & h_1 & \dots & h_{L_H} \end{bmatrix} \\ = [\mathbf{h}_0 \ \mathbf{h}_1 \ \dots \ \mathbf{h}_\iota \ \dots \ \mathbf{h}_{L_H}], \quad (10)$$

where $L_H = L_h + L_g$. Then the MMSE solution of \mathbf{g} is expressed as

$$\mathbf{g}_{\text{MMSE}} = \left(\mathbf{H}\mathbf{H}^H + \frac{2\sigma_n^2}{\sigma_w^2} \mathbf{I}_{L_H+1} \right)^{-1} \mathbf{h}_\iota, \quad (11)$$

in which $\sigma_w^2 = E[|\Psi(x(k))|^2]$ is the power of $w(k)$ which needs to be estimated. The optimal value for ι can be chosen to minimise the MMSE of the combined linear system of \mathbf{h} and \mathbf{g}

$$J_{\text{cmmse}}(\iota) = \sigma_w^2 \left(1 - \mathbf{h}_\iota^H \left(\mathbf{H}\mathbf{H}^H + \frac{2\sigma_n^2}{\sigma_w^2} \mathbf{I}_{L_H+1} \right)^{-1} \mathbf{h}_\iota \right). \quad (12)$$

Since the transmitter HPA is nonlinear, the linear equalizer (9) alone is insufficient for estimating the transmitted data symbol $x(k - \iota)$. If the inversion of the HPA's nonlinearity, $\Psi^{-1}(\cdot)$, is known, then the transmitter HPA's nonlinear distortion can be removed, yielding the estimate of $x(k - \iota)$

$$\widehat{x}(k - \iota) = \Psi^{-1}(\widehat{w}(k - \iota)). \quad (13)$$

3. The proposed adaptive nonlinear equalizer

As discussed in the previous section, in order to accomplish the equalization objective, it is necessary to identify the Hammerstein channel, including both the CIR \mathbf{h} and the HPA's nonlinearity $\Psi(\cdot)$, as well as to invert the nonlinearity $\Psi(\cdot)$.

3.1. Complex-valued B-spline neural network

Note that the HPA's nonlinearity, (4) and (5), is unknown to the receiver and $w(k)$ is unobserved. We adopt the CV B-spline neural network [14,17,18] to represent the mapping $\widehat{w} = \widehat{\Psi}(x_R + j \cdot x_I) : \mathbb{C} \rightarrow \mathbb{C}$ that is the estimate of $\Psi(\cdot)$. Before introducing the B-spline modeling of $\Psi(\cdot)$, we point out that the HPA $\Psi(\cdot)$ satisfies the following conditions.

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

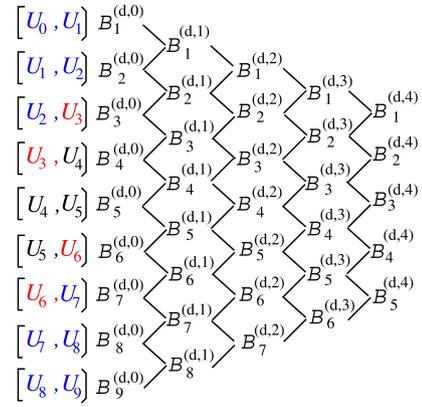


Fig. 1. Visualisation of the De Boor recursion for $P_0 = 4$ and $N_d = 5$, where $U_{\min} = U_3$ and $U_{\max} = U_6$.

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

According to the property 2), we assume that $U_{\min} < x_d < U_{\max}$, where U_{\min} and U_{\max} are known finite real values, while x_d represents either x_R or x_I , namely, the subscript d is either R or I .

A set of univariate B-spline basis functions for $x_d \in \mathbb{R}$ is parametrised by the order P_0 of a piecewise polynomial and a knot sequence which is a set of values defined on the real line that break it up into a number of intervals. To have N_d basis functions, the knot sequence is specified by $(N_d + P_0 + 1)$ knot values, $\{U_0, U_1, \dots, U_{N_d+P_0}\}$, with

$$U_0 < U_1 < \dots < U_{P_0-2} < U_{P_0-1} = U_{\min} < U_{P_0} < \dots < U_{N_d} < U_{N_d+1} = U_{\max} < U_{N_d+2} < \dots < U_{N_d+P_0}. \quad (14)$$

At each end, there are $P_0 - 1$ "external" knots that are outside the input region and one boundary knot. As a result, the number of "internal" knots is $N_d + 1 - P_0$. Given the set of predetermined knots (14), the set of N_d B-spline basis functions can be formed by using the De Boor recursion [19], yielding for $1 \leq l \leq N_d + P_0$,

$$B_l^{(d,0)}(x_d) = \begin{cases} 1, & \text{if } U_{l-1} \leq x_d < U_l, \\ 0, & \text{otherwise,} \end{cases} \quad (15)$$

as well as for $l = 1, \dots, N_d + P_0 - p$ and $p = 1, \dots, P_0$,

$$B_l^{(d,p)}(x_d) = \frac{x_d - U_{l-1}}{U_{p+l-1} - U_{l-1}} B_l^{(d,p-1)}(x_d) \\ + \frac{U_{p+l} - x_d}{U_{p+l} - U_l} B_{l+1}^{(d,p-1)}(x_d). \quad (16)$$

Here we have the subscript/superscript $d = R$ or I .

The De Boor recursion is illustrated in Fig. 1. $P_0 = 3$ to 4 is sufficient for most practical applications. The number of B-spline basis functions should be chosen to be sufficiently large to provide accurate approximation capability but not too large as to cause overfitting and to impose unnecessary modeling complexity. More specifically, $N_d = 6$ to 10 is usually sufficient for accurate modeling in the finite and known interval $[U_{\min}, U_{\max}]$. The two boundary knots are obviously related to the known values U_{\min} and U_{\max} , respectively. The $N_d + 1 - P_0$ internal knots may be uniformly spaced in the interval $[U_{\min}, U_{\max}]$. The extrapolation capability of the B-spline model is influenced by the choice of the external knots. Note that there exist no data for $x_d < U_{\min}$ and $x_d > U_{\max}$ in identification but it is desired that the B-spline model has certain

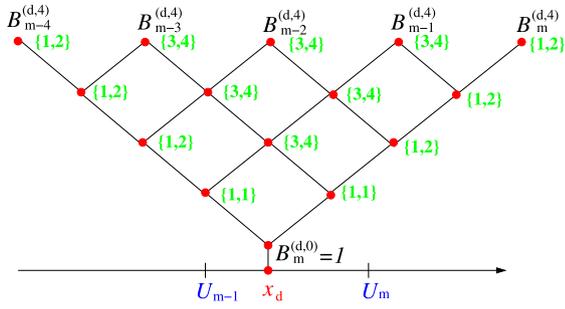


Fig. 2. Complexity of the B-spline model with $P_0 = 4$ using the De Boor recursion, where $\{a, b\}$ beside a node indicates that it requires a additions and b multiplications to compute the basis function value at this node.

extrapolating capability outside the interval $[U_{\min}, U_{\max}]$. The external knots can be set empirically to meet the required extrapolation capability. In fact, since no data appears outside $[U_{\min}, U_{\max}]$, the precise choice of these external knots does not really matter, in terms of modeling accuracy. Also note that for QAM signals, the distribution of x_d is naturally symmetric and, therefore, the knot sequence should be chosen to be symmetric too.

Using the tensor product between the two sets of univariate B-spline basis functions [20], $B_l^{(R, P_0)}(x_R)$ for $1 \leq l \leq N_R$ and $B_m^{(I, P_0)}(x_I)$ for $1 \leq m \leq N_I$, a set of the new B-spline basis functions $B_{l,m}^{(P_0)}(x)$ can be formed and used in the CV B-spline neural network, giving rise to

$$\begin{aligned} \widehat{w} &= \widehat{\Psi}(x) = \sum_{l=1}^{N_R} \sum_{m=1}^{N_I} B_{l,m}^{(P_0)}(x) \omega_{l,m} \\ &= \sum_{l=1}^{N_R} \sum_{m=1}^{N_I} B_l^{(R, P_0)}(x_R) B_m^{(I, P_0)}(x_I) \omega_{l,m}, \end{aligned} \quad (17)$$

where $\omega_{l,m} = \omega_{l,m_R} + j \cdot \omega_{l,m_I} \in \mathbb{C}$, $1 \leq l \leq N_R$ and $1 \leq m \leq N_I$, are the CV weights. Denote the weight vector of the B-spline model (17) as

$$\omega = [\omega_{1,1} \ \omega_{1,2} \ \cdots \ \omega_{l,m} \ \cdots \ \omega_{N_R, N_I}]^T \in \mathbb{C}^{N_B}, \quad (18)$$

where $N_B = N_R N_I$. The task of identifying the nonlinearity $\Psi(\cdot)$ is turned into one of estimating the parameter vector.

Remark 1. Because of the piecewise nature of B-spline functions, given a value $x_d \in [U_{\min}, U_{\max}]$, there are only $P_0 + 1$ basis functions with nonzero values at most. This is advantageous as P_0 can be set to a quite low value, e.g. $P_0 = 4$ is often sufficient. The complexity of the De Boor recursion is, therefore, on the order of P_0^2 [17,19]. Fig. 2 shows the complexity of generating the B-spline basis function set for $P_0 = 4$ using the De Boor recursion. Note that the complexity does not depend on the number of basis functions N_d employed. For the B-spline model with the polynomial degree $P_0 = 4$, the total computational requirements are 26 additions and 38 multiplications at most.

Thus, in the tensor-product B-spline model of (17), there are only $(P_0 + 1)^2$ nonzero basis functions at most for any given input. This is in fact comparable to the conventional polynomial modeling. For the polynomial model with the polynomial degree P_0 , there are also $P_0 + 1$ basis functions which are given by

$$1, x_d, x_d^2, \dots, x_d^{P_0}.$$

Thus, the tensor-product polynomial model also have $(P_0 + 1)^2$ nonzero basis functions.

Remark 2. B-splines have been widely studied in the subjects of approximation theory and numerical analysis, owing to their many desired properties. In particular, the B-spline basis functions as model basis have the best approximation capability, because the basis function is complete. Although any polynomial function can also be used to approximate a continuous function, the B-spline functions are proven to be optimally stable bases [21–23]. A critical aspect to consider in a model representation is its stability with respect to perturbation of the model parameters, because in any identification, the data are inevitably noisy, which will perturb the model parameters away from their true values. A significant advantage of using the B-spline model with De Boor algorithm for functional approximation over many other polynomial forms is its superior numerical stability [21–23]. Let us further analyze this aspect. Assume that the real-valued true system can be represented by the polynomial model of degree P_0 as

$$y = \sum_{i=0}^{P_0} a_i \cdot x^i,$$

as well as by the following B-spline model exactly

$$y = \sum_{i=1}^{N_d} b_i \cdot B_i^{(d, P_0)}(x),$$

where $y, x \in \mathbb{R}$. Because the identification data are noisy, 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, namely, $|\varepsilon_i| < \varepsilon_{\max}$. The upper bound of $|y - \widehat{y}|$ for the B-spline model can be worked out to be

$$\begin{aligned} |y - \widehat{y}| &= \left| \sum_{i=1}^{N_d} b_i \cdot B_i^{(d, P_0)}(x) - \sum_{i=1}^{N_d} \widehat{b}_i \cdot B_i^{(d, P_0)}(x) \right| \\ &< \varepsilon_{\max} \cdot \left| \sum_{i=1}^{N_d} B_i^{(d, P_0)}(x) \right| = \varepsilon_{\max}. \end{aligned}$$

Observe that the upper bound of the B-spline model output perturbation only depends on the upper bound of the perturbation noise, and it does not depend on the input value x , the number of basis functions N_d or the polynomial degree P_0 . This confirms that the B-spline model has the maximum numerical robustness, which is well-known. Optimality of the B-spline model in terms of numerical stability is due to the convexity of its model bases, i.e. they are all positive and sum to one. By contrast, the upper bound of $|y - \widehat{y}|$ for the polynomial model can be worked out to be

$$|y - \widehat{y}| = \left| \sum_{i=0}^{P_0} a_i \cdot x^i - \sum_{i=0}^{P_0} \widehat{a}_i \cdot x^i \right| < \varepsilon_{\max} \cdot \left| \sum_{i=0}^{P_0} x^i \right|$$

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 and the polynomial degree P_0 . The higher the polynomial degree P_0 , the more serious the polynomial model may be perturbed, a well-known drawback of using polynomial modeling.

The excellent numerical stability of the B-spline model is illustrated using a simple example. 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. Based on the knot sequence of $\{-5, -4, 0, 20, 24, 25\}$, this function is modeled as a quadratic B-spline model of $y = 0.14B_1^{(d,2)}(x) - 0.10B_2^{(d,2)}(x) + 0.14B_3^{(d,2)}(x)$, which is depicted in Fig. 3(b) in solid line. We now draw three noises ε_i , $1 \leq i \leq 3$, from a uniformly distributed random number (UDRN) in

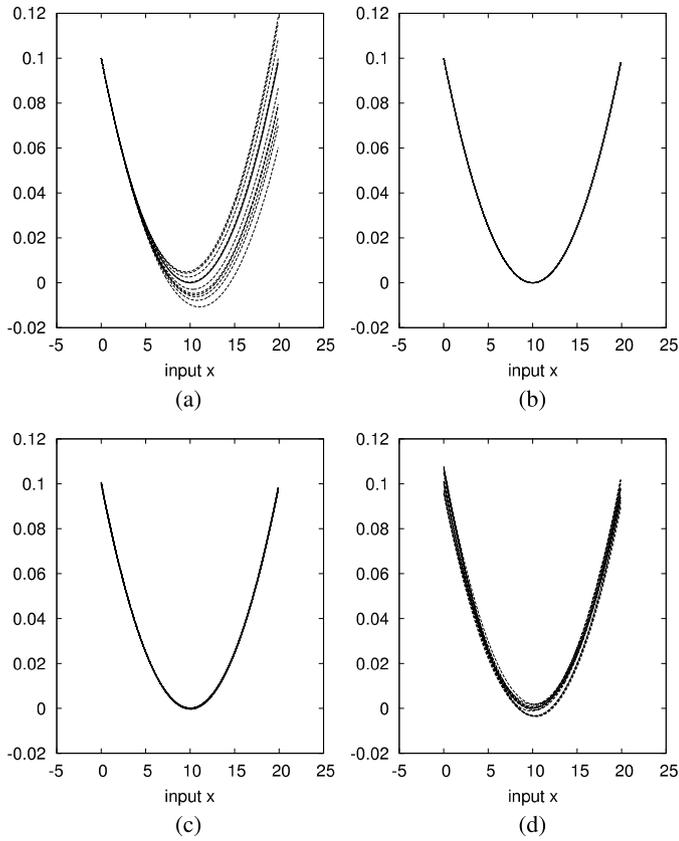


Fig. 3. (a) The polynomial model with three perturbation noises drawn from a uniformly distributed random number (UDRN) in $[-0.0001, 0.0001]$, (b) the B-spline model with three perturbation noises drawn from a UDRN in $[-0.0001, 0.0001]$, (c) the B-spline model with three perturbation noises drawn from a UDRN in $[-0.001, 0.001]$, and (d) the B-spline model with three perturbation noises drawn from a UDRN in $[-0.01, 0.01]$.

$[-0.0001, 0.0001]$, and add them to the three parameters in the two models, respectively, to simulate the effects of the noise in identification. Figs. 3(a) and (b) depict the ten sets of the perturbed functions in dashed line generated by perturbing the two models, respectively. 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 quadratic B-spline model. To further demonstrate the maximum robustness of the B-spline model, we next 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 Figs. 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.

3.2. Training

3.2.1. Hammerstein channel identification

The identification of the Hammerstein channel (2) and (3) then involves estimating the parameter vector ω of the CV B-spline neural network (17) that represents the HPA nonlinearity $\Psi(\cdot)$ as well as the CIR coefficient vector \mathbf{h} . Note that during the identification of this Hammerstein channel, $h_0 = 1$ can be assumed because if this is not the case, h_0 can always be absorbed into the CV static

nonlinearity $\Psi(\cdot)$, and the CIR coefficients are re-scaled as h_i/h_0 for $0 \leq i \leq L_h$. Consider the joint estimation of ω and \mathbf{h} based on a block of the training data with K samples $\{x(k), y(k)\}_{k=1}^K$. The identification task can be formulated as the one that minimizes the cost function

$$J_{\text{icf}} = \frac{1}{K} \sum_{k=1}^K |e(k)|^2 = \frac{1}{K} \sum_{k=1}^K |y(k) - \hat{y}(k)|^2, \quad (19)$$

subject to the constraint $h_0 = 1$, in which the model prediction $\hat{y}(k)$ is given by

$$\hat{y}(k) = \sum_{i=0}^{L_h} h_i \hat{\omega}(k-i) = \sum_{i=0}^{L_h} h_i \sum_{l=1}^{N_R} \sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(x(k-i)) \omega_{l,m}. \quad (20)$$

Note that the cost function (19) is convex with respect to \mathbf{h} when fixing ω , and it is convex with respect to ω given a fixed \mathbf{h} . This is simply because the model (20) can be viewed as two different linear regression models, namely, one is with respect to \mathbf{h} when fixing ω and the other is with respect to ω given a fixed \mathbf{h} , each problem having a closed-form solution. According to [24,25], the estimates of ω and \mathbf{h} are unbiased, irrespective the optimization algorithm used to minimize the cost function (19). We adopt the following efficient ALS procedure to estimate both ω and \mathbf{h} .

Initialisation Noting $h_0 = 1$, define the amalgamated parameter vector as

$$\theta = [\omega^T h_1 \omega^T h_2 \omega^T \dots h_{L_h} \omega^T]^T \in \mathbb{C}^{(L_h+1)N_B}, \quad (21)$$

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

$$\phi(k) = [\phi_{1,1}(k) \phi_{1,2}(k) \dots \phi_{l,m}(k) \dots \phi_{N_R, N_I}(k)]^T \quad (22)$$

with

$$\phi_{l,m}(k) = B_{l,m}^{(P_o)}(x(k)), \quad 1 \leq l \leq N_R, 1 \leq m \leq N_I. \quad (23)$$

Further define the desired output vector as

$$\mathbf{y} = [y(1) y(2) \dots y(K)]^T \in \mathbb{C}^K, \quad (24)$$

and the regression matrix $\mathbf{P} \in \mathbb{R}^{K \times (L_h+1)N_B}$ as

$$\mathbf{P} = \begin{bmatrix} \phi^T(1) & \phi^T(0) & \dots & \phi^T(1-L_h) \\ \vdots & \vdots & \vdots & \vdots \\ \phi^T(k) & \phi^T(k-1) & \dots & \phi^T(k-L_h) \\ \vdots & \vdots & \vdots & \vdots \\ \phi^T(K) & \phi^T(K-1) & \dots & \phi^T(K-L_h) \end{bmatrix}. \quad (25)$$

Then the LS estimate of θ is readily given by

$$\hat{\theta} = (\mathbf{P}^T \mathbf{P})^{-1} \mathbf{P}^T \mathbf{y}. \quad (26)$$

Obviously, $\hat{\theta}$ is a unique and unbiased estimate of θ . Therefore, the first N_B elements of $\hat{\theta}$ provide a unique and unbiased LS estimate for the weight vector of the CV B-spline neural network ω , which will be denoted as $\hat{\omega}^{(0)}$.

Alternating LS estimation For $1 \leq \tau \leq \tau_{\max}$, where τ_{\max} is the maximum number of iterations, perform:

a) Given the fixed $\hat{\omega}^{(\tau-1)}$, calculate the LS estimate $\hat{\mathbf{h}}^{(\tau)}$. Specifically, define the regression matrix $\mathbf{Q} \in \mathbb{C}^{K \times (L_h+1)}$ as

$$\mathbf{Q} = \begin{bmatrix} \widehat{w}(1) & \widehat{w}(0) & \cdots & \widehat{w}(1-L_h) \\ \vdots & \vdots & \vdots & \vdots \\ \widehat{w}(k) & \widehat{w}(k-1) & \cdots & \widehat{w}(k-L_h) \\ \vdots & \vdots & \vdots & \vdots \\ \widehat{w}(K) & \widehat{w}(K-1) & \cdots & \widehat{w}(K-L_h) \end{bmatrix}, \quad (27)$$

in which

$$\widehat{w}(k) = \widehat{\Psi}(x(k)) = \sum_{l=1}^{N_R} \sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(x(k)) \widehat{\omega}_{l,m}^{(\tau-1)}. \quad (28)$$

Then the LS estimate $\widehat{\mathbf{h}}^{(\tau)}$ is readily given by

$$\widehat{\mathbf{h}}^{(\tau)} = (\mathbf{Q}^H \mathbf{Q})^{-1} \mathbf{Q}^H \mathbf{y}, \quad (29)$$

$$\widehat{h}_i^{(\tau)} = \widehat{h}_i^{(\tau)} / \widehat{h}_0^{(\tau)}, \quad 0 \leq i \leq L_h. \quad (30)$$

Since $\widehat{\omega}^{(\tau-1)}$ is a unique and unbiased estimate of ω , the LS estimate $\widehat{\mathbf{h}}^{(\tau)}$ is guaranteed to be a unique and unbiased estimate of \mathbf{h} .

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

$$\varphi_{l,m}(k) = \sum_{i=0}^{L_h} \widehat{h}_i^{(\tau)} B_{l,m}^{(P_o)}(x(k-i)) \in \mathbb{C}. \quad (31)$$

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

$$\boldsymbol{\varphi}(k) = [\varphi_{1,1}(k) \varphi_{1,2}(k) \cdots \varphi_{l,m}(k) \cdots \varphi_{N_R,N_I}(k)]^T, \quad (32)$$

and define the regression matrix

$$\mathbf{S} = [\boldsymbol{\varphi}(1) \boldsymbol{\varphi}(2) \cdots \boldsymbol{\varphi}(K)]^T \in \mathbb{C}^{K \times N_B}. \quad (33)$$

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

$$\widehat{\omega}^{(\tau)} = (\mathbf{S}^H \mathbf{S})^{-1} \mathbf{S}^H \mathbf{y}. \quad (34)$$

Since $\widehat{\mathbf{h}}^{(\tau)}$ is a unique and unbiased estimate of \mathbf{h} , the LS estimate $\widehat{\omega}^{(\tau)}$ must be a unique and unbiased estimate of ω .

A few iterations are sufficient for this estimation procedure to arrive at a highly accurate and joint unbiased estimate of \mathbf{h} and ω that is the unique minimum solution of the cost function (20).

Remark 3. It is clear that this ALS procedure guarantees to converge, in fact, in no more than one iteration. A few iterations, typically 2 to 3, are adopted to improve the estimation accuracy or to reduce the estimation variances. More specifically, since $\widehat{\omega}^{(0)}$ is a unique and unbiased estimate of ω , $\widehat{\mathbf{h}}^{(1)}$ is guaranteed to be a unique and unbiased estimate of \mathbf{h} . On the other hand, the unique and unbiased estimate $\widehat{\boldsymbol{\theta}}$ of the high-dimensional amalgamated parameter vector $\boldsymbol{\theta} \in \mathbb{C}^{(L_h+1)N_B}$ may have relative high estimation variances, owing to the low ratio of the available training data over the dimension $(L_h+1)N_B$. Consequently, The first N_B elements of $\widehat{\boldsymbol{\theta}}$, i.e. $\widehat{\omega}^{(0)}$ may have a relatively poor estimation accuracy. With the fixed $\widehat{\mathbf{h}}^{(1)}$, the LS estimate $\widehat{\omega}^{(1)}$ has lower estimation variance than $\widehat{\omega}^{(0)}$. In the second iteration, the estimation accuracy of $\widehat{\mathbf{h}}^{(2)}$ and $\widehat{\omega}^{(2)}$ will be further enhanced, in comparison to the estimation accuracy of $\widehat{\mathbf{h}}^{(1)}$ and $\widehat{\omega}^{(1)}$.

3.2.2. Nonlinear equalizer construction

With the estimated HPA's nonlinearity $\widehat{\Psi}(\cdot)$ and the CIR tap vector $\widehat{\mathbf{h}}$, an estimated noise power is readily given by $2\widehat{\sigma}_n^2 = J_{\text{icf}}(\widehat{\mathbf{h}}, \widehat{\omega})$, while an estimate for the power of the unobserved $w(k)$ is given by

$$\widehat{\sigma}_w^2 = \frac{1}{K} \sum_{k=1}^K |\widehat{w}(k)|^2 = \frac{1}{K} \sum_{k=1}^K |\widehat{\Psi}(x(k))|^2. \quad (35)$$

Then the linear equalizer's weight vector $\widehat{\mathbf{g}}_{\text{MMSE}}$ is readily provided by (11) based on the estimates of $\widehat{\mathbf{h}}$, $2\widehat{\sigma}_n^2$ and $\widehat{\sigma}_w^2$.

Given $\Psi(\bullet)$, we need to compute its inversion as defined by $\widehat{x}(k-l) = \Psi^{-1}(\widehat{w}(k-l))$ of (13) in order to complete the nonlinear equalization. This task is identical to find the CV root of $\widehat{w}(k) = \Psi(\widehat{x}(k))$, given $\widehat{w}(k)$, which can be solved iteratively based on the Gauss-Newton algorithm [14,17,18]. Given the estimated $\widehat{\Psi}(\cdot)$ and during the data detection, the strategy of [14,17,18] requires to iteratively calculate the root of $\widehat{w}(k-l) = \widehat{\Psi}(\widehat{x}(k-l))$ for each linearly equalized received signal sample $\widehat{w}(k-l)$ in order to obtain the estimate $\widehat{x}(k-l)$ of the transmitted data symbol $x(k-l)$. In order to avoid the iterative root finding procedure for every sample $\widehat{w}(k)$, it is more efficient to directly construct a mapping to model $x(k) = \Phi(w(k)) = \Psi^{-1}(w(k))$. We adopt another CV B-spline neural network to represent the inverse mapping $\Phi(w(k))$. To learn the mapping $x(k) = \Phi(w(k))$, however, a training data set $\{w(k), x(k)\}$ would be needed but $w(k)$ is unobservable and, therefore, is not available. Fortunately, in the Hammerstein channel identification, we already obtain an estimate $\widehat{\Psi}(\cdot)$ for the HPA's nonlinearity $\Psi(\cdot)$. Therefore, we may construct the "pseudo" training data set $\{\widehat{w}(k), x(k)\}_{k=1}^K$ for identifying the inverse mapping $\Phi(\cdot)$, where $\widehat{w}(k)$ is computed based on the estimated $\widehat{\Psi}(\cdot)$.

More specifically, define the two knot sequences similar to (14) for w_R and w_I . Similar to (17), we construct the inverting B-spline neural network

$$\begin{aligned} \widehat{x} &= \widehat{\Phi}(w) = \sum_{l=1}^{N_R} \sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(w) \alpha_{l,m} \\ &= \sum_{l=1}^{N_R} \sum_{m=1}^{N_I} B_l^{(R,P_o)}(w_R) B_m^{(I,P_o)}(w_I) \alpha_{l,m}, \end{aligned} \quad (36)$$

where $B_l^{(R,P_o)}(w_R)$ and $B_m^{(I,P_o)}(w_I)$ are respectively calculated based on (15) and (16), while

$$\boldsymbol{\alpha} = [\alpha_{1,1} \alpha_{1,2} \cdots \alpha_{l,m} \cdots \alpha_{N_R,N_I}]^T \in \mathbb{C}^{N_B} \quad (37)$$

is the parameter vector of this inverting B-spline neural network. Here for notational simplicity, we assume that the same number of basis functions and the same polynomial degree are used for the two B-spline neural networks that model $\Psi(x)$ and $\Phi(w)$. Over the pseudo training data set $\{\widehat{w}(k), x(k)\}_{k=1}^K$, the regression matrix $\widehat{\mathbf{B}} \in \mathbb{R}^{K \times N_B}$ can be formed as

$$\widehat{\mathbf{B}} = \begin{bmatrix} \overline{B}_{1,1}^{(P_o)}(1) & \overline{B}_{1,2}^{(P_o)}(1) & \cdots & \overline{B}_{N_R,N_I}^{(P_o)}(1) \\ \overline{B}_{1,1}^{(P_o)}(2) & \overline{B}_{1,2}^{(P_o)}(2) & \cdots & \overline{B}_{N_R,N_I}^{(P_o)}(2) \\ \vdots & \vdots & \vdots & \vdots \\ \overline{B}_{1,1}^{(P_o)}(K) & \overline{B}_{1,2}^{(P_o)}(K) & \cdots & \overline{B}_{N_R,N_I}^{(P_o)}(K) \end{bmatrix}, \quad (38)$$

where $\overline{B}_{l,m}^{(P_o)}(k) = B_{l,m}^{(P_o)}(\widehat{w}(k))$, while the associated desired output vector is given by

$$\mathbf{x} = [x(1) x(2) \cdots x(K)]^T. \quad (39)$$

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

3.3. Decision-directed adaptation

During the data communication phase, since the transmitter HPA $\Psi(\cdot)$ remains the same, the estimated HPA's nonlinearity $\widehat{\Psi}(\cdot)$

and its inversion $\widehat{\Phi}(\cdot) = \widehat{\Psi}^{-1}(\cdot)$ obtained during the training remain valid. Therefore, we do not need to update these two B-spline models' parameter vectors, $\widehat{\omega}$ and $\widehat{\alpha}$. However, as the CIR changes from frame to frame, the estimated CIR tap vector $\widehat{\mathbf{h}}$ obtained during the training must be adapted. We used a decision-directed (DD) LS estimator to update the CIR coefficient vector during the data communication.

Specifically, let f be the data frame index, while $\widehat{\mathbf{h}}^{(f)}$ and $\widehat{\mathbf{g}}_{\text{MMSE}}^{(f)}$ denote the estimated CIR tap vector and the corresponding linear equalizer weight vector, respectively, after the detection of the f -th data frame. Also the training based estimates are denoted as $\widehat{\mathbf{h}} = \widehat{\mathbf{h}}^{(0)}$ and $\widehat{\mathbf{g}}_{\text{MMSE}} = \widehat{\mathbf{g}}_{\text{MMSE}}^{(0)}$, respectively. Given $\widehat{\mathbf{g}}_{\text{MMSE}}^{(f-1)}$ and $\widehat{\Phi}(\cdot) = \widehat{\Psi}^{-1}(\cdot)$, the detection of the f -th data frame is carried out by performing the nonlinear equalization of (9) and (13), and then quantising the resulting $\widehat{\mathbf{x}}(k)$ to obtain the hard decisions $\widetilde{\mathbf{x}}(k)$ for $1 \leq k \leq N_F$. Given the estimate of the HPA $\widehat{\Psi}(\cdot)$ and based on the hard decisions of $\{\widetilde{\mathbf{x}}(k)\}_{k=1}^{N_F}$, the DD estimate of the HPA's output can be calculated according to

$$\widetilde{\mathbf{w}}(k) = \widehat{\Psi}(\widetilde{\mathbf{x}}(k)) = \sum_{l=1}^{N_R} \sum_{m=1}^{N_I} B_{l,m}^{(P_o)}(\widetilde{\mathbf{x}}(k)) \widehat{\omega}_{l,m}. \quad (40)$$

By constructing the DD regression matrix $\widetilde{\mathbf{Q}} \in \mathbb{C}^{N_F \times (L_h+1)}$ as

$$\widetilde{\mathbf{Q}} = \begin{bmatrix} \widetilde{\mathbf{w}}(1) & \widetilde{\mathbf{w}}(0) & \cdots & \widetilde{\mathbf{w}}(1-L_h) \\ \vdots & \vdots & \vdots & \vdots \\ \widetilde{\mathbf{w}}(k) & \widetilde{\mathbf{w}}(k-1) & \cdots & \widetilde{\mathbf{w}}(k-L_h) \\ \vdots & \vdots & \vdots & \vdots \\ \widetilde{\mathbf{w}}(N_F) & \widetilde{\mathbf{w}}(N_F-1) & \cdots & \widetilde{\mathbf{w}}(N_F-L_h) \end{bmatrix} \quad (41)$$

and denoting the corresponding channel observation vector as $\widetilde{\mathbf{y}} \in \mathbb{C}^{N_F}$, the DD LS estimate is readily be obtained as

$$\widehat{\mathbf{h}}^{(f)} = (\widetilde{\mathbf{Q}}^H \widetilde{\mathbf{Q}})^{-1} \widetilde{\mathbf{Q}}^H \widetilde{\mathbf{y}}. \quad (42)$$

Given $\widehat{\mathbf{h}}^{(f)}$ as well as the training based estimates $2\widehat{\sigma}_n^2$ and $\widehat{\sigma}_w^2$, we can obtain an updated linear equalizer's weight vector $\widehat{\mathbf{g}}_{\text{MMSE}}^{(f)}$ according to (11). Algorithm 1 summarizes the detection of each data frame, which involves two iterations of the above-mentioned DD adaptation.

Algorithm 1 Detection of f -th data frame and decision-directed updating.

- 1: Give $\widehat{\mathbf{g}}_{\text{MMSE}}^{(f-1,0)} = \widehat{\mathbf{g}}_{\text{MMSE}}^{(f-1)}$, $\widehat{\Phi}(\cdot) = \widehat{\Psi}^{-1}(\cdot)$ and $\widehat{\Psi}(\cdot)$, as well as the observations $y(k)$, $1 \leq k \leq N_F$;
- 2: **for** $t = 1$ to 2 **do**
- 3: Given $\widehat{\mathbf{g}}_{\text{MMSE}}^{(f-1,t-1)}$ and $\widehat{\Phi}(\cdot)$, perform data detection according to the nonlinear equalization of (9) and (13) to obtain the hard decisions $\widetilde{\mathbf{x}}(k-t)$;
- 4: Given the detected symbols $\{\widetilde{\mathbf{x}}(k)\}_{k=1}^{N_F}$ and $\widehat{\Psi}(\cdot)$, obtain the DD LS estimate $\widehat{\mathbf{h}}^{(f-1,t)}$ according to (42) and, therefore, obtain the updated equalizer weight vector $\widehat{\mathbf{g}}_{\text{MMSE}}^{(f-1,t)}$;
- 5: **end for**
- 6: $\widehat{\mathbf{g}}_{\text{MMSE}}^{(f)} = \widehat{\mathbf{g}}_{\text{MMSE}}^{(f-1,2)}$.

Remark 4. While maintaining a high system throughput, this decision-directed updating scheme introduces a fixed delay in detecting each data frame. Since this delay is constant for every data frame, it will not cause problem even to real-time applications. Alternative would be to allocate part of each data frame, e.g. 10%, for training, at a cost of significantly reducing the achievable system throughput.

3.4. Extension to nonlinear channel with Wiener HPA

For most practical systems, the transmitter HPA can be modeled by a CV static nonlinear mapping. In some systems, HPAs may

Table 1

Empirically determined knot sequences.

x_R and x_I (OBO = 5 dB)
-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
x_R and x_I (OBO = 8 dB)
-10.0, -9.0, -0.9, - 0.5 , -0.04, -0.02, 0.0, 0.02, 0.04, 0.5 , 0.9, 9.0, 10.0
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

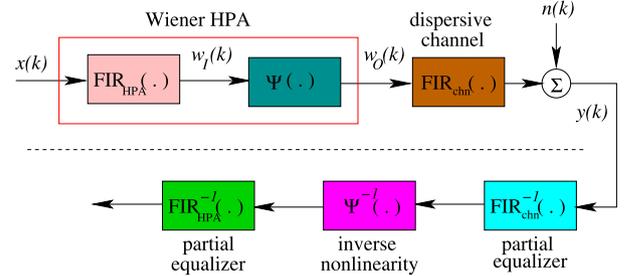


Fig. 4. Illustration of nonlinear equalization for the dispersive channel with the Wiener high power amplifier at transmitter.

exhibit memory [26]. By modeling an HPA with memory as a Hammerstein system, namely, a CV static nonlinearity $\Psi(\bullet) : \mathbb{C} \rightarrow \mathbb{C}$ followed by a FIR linear filter, then the combined HPA and dispersive channel can also be represented by our model of (2) and (3), where the linear filter (3) is the convolution of the HPA's FIR filter and the dispersive channel. Thus, the nonlinear equalization approach developed in this work can be applied directly to this class of nonlinear HPAs with memory.

Another popular model for nonlinear HPAs with memory is the Wiener model [4] which represents an HPA with memory by a FIR filter followed by a CV static nonlinearity. This class of nonlinear dispersive channels with the Wiener HPA at transmitter is depicted in the top part of Fig. 4, while the bottom part of Fig. 4 illustrates the corresponding nonlinear equalizer design. The same tensor-product B-spline model of (17) can be adopted to model the CV static nonlinearity $\Psi(\bullet)$. However, the identification of the nonlinear system illustrated in Fig. 4 is a much more difficult task. In particular, nonlinear estimation methods, such as the gradient-based algorithms [13,18,27] and the evolutionary algorithms [12], must be employed in order to estimate the parameter vectors of the linear filter $\text{FIR}_{\text{HPA}}(\bullet)$, the B-spline model of the CV static nonlinearity $\Psi(\bullet)$ and the linear filter $\text{FIR}_{\text{chn}}(\bullet)$, which are inherently high complexity and may suffer from the drawback of slow convergence.

Once the estimates of $\text{FIR}_{\text{HPA}}(\bullet)$, $\Psi(\bullet)$ and $\text{FIR}_{\text{chn}}(\bullet)$ are obtained, it is straightforward to apply the nonlinear equalizer design presented in this work. Specifically, the partial equalizers $\text{FIR}_{\text{chn}}^{-1}(\bullet)$ and $\text{FIR}_{\text{HPA}}^{-1}(\bullet)$ can readily be obtained based on the estimates of $\text{FIR}_{\text{chn}}(\bullet)$ and $\text{FIR}_{\text{HPA}}(\bullet)$, respectively. The inverse nonlinearity $\Psi^{-1}(\bullet)$ can be modeled by the same inverting B-spline model of (36), whose parameters can be estimated in a similar manner based on the pseudo training data $\{\widehat{w}_O(k), \widehat{w}_I(k)\}_{k=1}^K$ calculated using the estimates of $\text{FIR}_{\text{HPA}}(\bullet)$ and $\Psi(\bullet)$.

4. Simulation study

We considered the 64-QAM Hammerstein communication system in which the HPA employed was described by (4) and (5) with the parameter set given in (6). We assumed a quasi-frame-static Rayleigh multipath channel with an exponentially decreasing power delay profile, where the average gain for the l th path was given by

Table 2
Identification results for the CIR coefficient vector \mathbf{h} of the Hammerstein stationary channel.

Tap No.	True parameters	Estimated parameters			
		OBO = 5 dB $E_s/N_0 = 5$ dB	OBO = 5 dB $E_s/N_0 = 10$ dB	OBO = 8 dB $E_s/N_0 = 5$ dB	OBO = 8 dB $E_s/N_0 = 10$ dB
h_0	1	1	1	1	1
h_1	$-0.31606 + j0.47804$	$-0.31562 + j0.47833$	$-0.31580 + j0.47820$	$-0.31571 + j0.47825$	$-0.31586 + j0.47816$
h_2	$0.09484 + j0.15786$	$0.09373 + j0.15752$	$0.09422 + j0.15767$	$0.09388 + j0.15756$	$0.09430 + j0.15769$
h_3	$0.03030 + j0.05838$	$0.03005 + j0.05879$	$0.03016 + j0.05862$	$0.03012 + j0.05869$	$0.03020 + j0.05856$

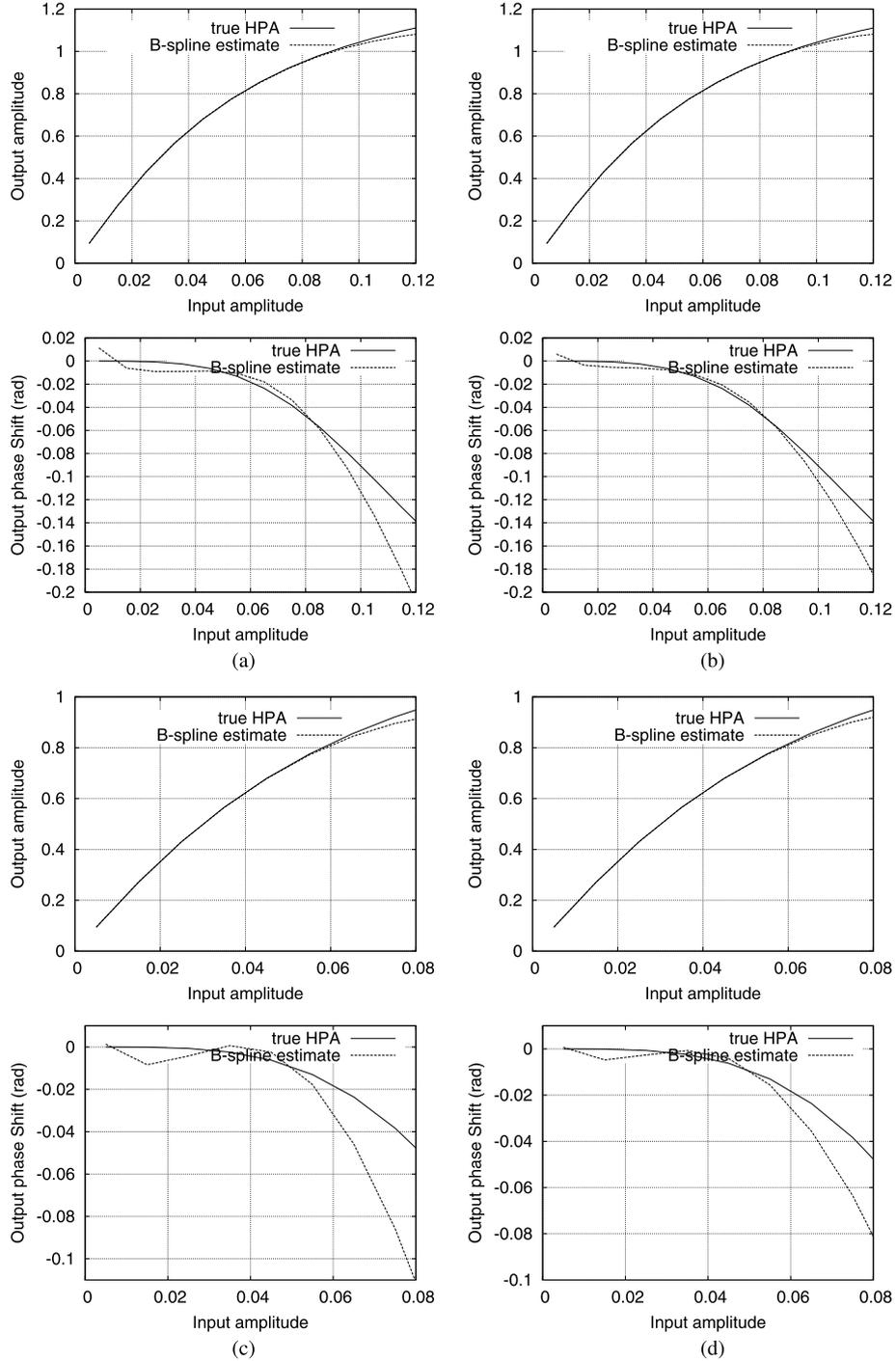


Fig. 5. Comparison of the HPA's static nonlinearity $\Psi(\bullet)$ and the B-spline estimated static nonlinearity $\hat{\Psi}(\bullet)$ under: (a) OBO = 5 dB and $E_s/N_0 = 5$ dB, (b) OBO = 5 dB and $E_s/N_0 = 10$ dB, (c) OBO = 8 dB and $E_s/N_0 = 5$ dB, and (d) OBO = 8 dB and $E_s/N_0 = 10$ dB.

$$E[|h_l|] = e^{-\frac{l}{\eta}}, 0 \leq l \leq L_h. \quad (43)$$

In the simulation, the channel degradation factor was chosen to be $\eta = 2$ while the channel length was set to $L_h = 3$. Each frame contained $N_F = 2000$ 64-QAM data symbols, while the first frame was used for training, yielding $K = N_F = 2000$. The CIR coefficients h_l for $0 \leq l \leq L_h$ changed at the beginning of each frame according to a Rayleigh fading distribution with the normalized Doppler frequency of $f_d = 0.0001$, but they remained constant within a frame. The linear equalizer's length was chosen to be $L_g = 15$ and the decision delay was set to $\iota = 0$. The piecewise quartic polynomial of $P_o = 4$ was chosen as the B-spline basis function, and the number of B-spline basis functions was set to $N_R = N_I = 8$. The empirically determined knot sequences for the two B-spline models covering different HPA operating conditions are listed in Table 1. The system's signal-to-noise ratio (SNR) was defined as $\text{SNR} = E_s/N_o$, where E_s was the average power of the input signal $x(k)$ to the HPA and $N_o = 2\sigma_n^2$ was the channel AWGN's power.

4.1. Training performance

We first considered the stationary channel, whose CIR tap vector \mathbf{h} is listed in Table 2, in order to evaluate the training performance of the proposed B-spline neural network based nonlinear equalizer. The ALS algorithm of Section 3.2 was used to identify this Hammerstein channel, specifically, to provide both the estimates of the CIR vector $\hat{\mathbf{h}}$ and the B-spline neural network's weight vector $\hat{\omega}$. It was observed that no more than four iterations were sufficient for the algorithm to obtain the highly accurate estimation results as summarized in Table 2 as well as illustrated in Fig. 5. Indeed, observe from Table 2 that the identification of the CIR tap vector in the nonlinear Hammerstein channel was achieved with high precision even under the adverse operational condition of $\text{OBO} = 5$ dB and $E_s/N_o = 5$ dB. Note that under the HPA operational condition of $\text{OBO} = 5$ dB, the peak amplitude of $|x(k)|$ was less than 0.09, while under the condition of $\text{OBO} = 8$ dB, the peak amplitude of $|x(k)|$ was less than 0.06. The results of Fig. 5 clearly demonstrate the capability of the proposed CV B-spline neural network to accurately model the HPA's nonlinearity $\Psi(\cdot)$. To be more specifically, within the HPA's operational input range, the estimated amplitude response exhibits negligible deviation from the true HPA's amplitude response, while the maximum error in the estimated phase response is no more than 0.01 radian. From Fig. 5, it can also be seen that the channel noise $n(k)$ has little effort on the accuracy of the amplitude response estimate, but the noise has noticeable influence on the accuracy of the phase response estimate.

The effectiveness of the proposed CV B-spline inversion based on the pseudo training data is next demonstrated, and Fig. 6 depicts the combined response of the HPA's true nonlinearity and its estimated inversion obtained under the operating conditions of $\text{OBO} = 5$ dB and $E_s/N_o = 5$ dB as well as $\text{OBO} = 5$ dB and $E_s/N_o = 10$ dB, respectively. The results of Fig. 6 show that the CV B-spline inversion estimated based on the pseudo training data is highly accurate. Specifically, within the HPA's operating input range, the combined amplitude response of the HPA's true nonlinearity and the estimated B-spline inversion matches the ideal response, while the combined phase response only deviates from the ideal response of zero by no more than 0.01 radian. For the given operating condition of $\text{OBO} = 8$ dB and $E_b/N_o = 10$ dB, Fig. 7 plots the HPA's input signal $x(k)$, the noisy Hammerstein channel's output signal $y(k)$, the linearly equalized signal $\hat{w}(k)$ based on the estimated CIR $\hat{\mathbf{h}}$, and the nonlinearly equalized signal $\hat{x}(k)$ based on the estimated CIR $\hat{\mathbf{h}}$ and the estimated B-spline inversion $\hat{\Phi}(\cdot)$.

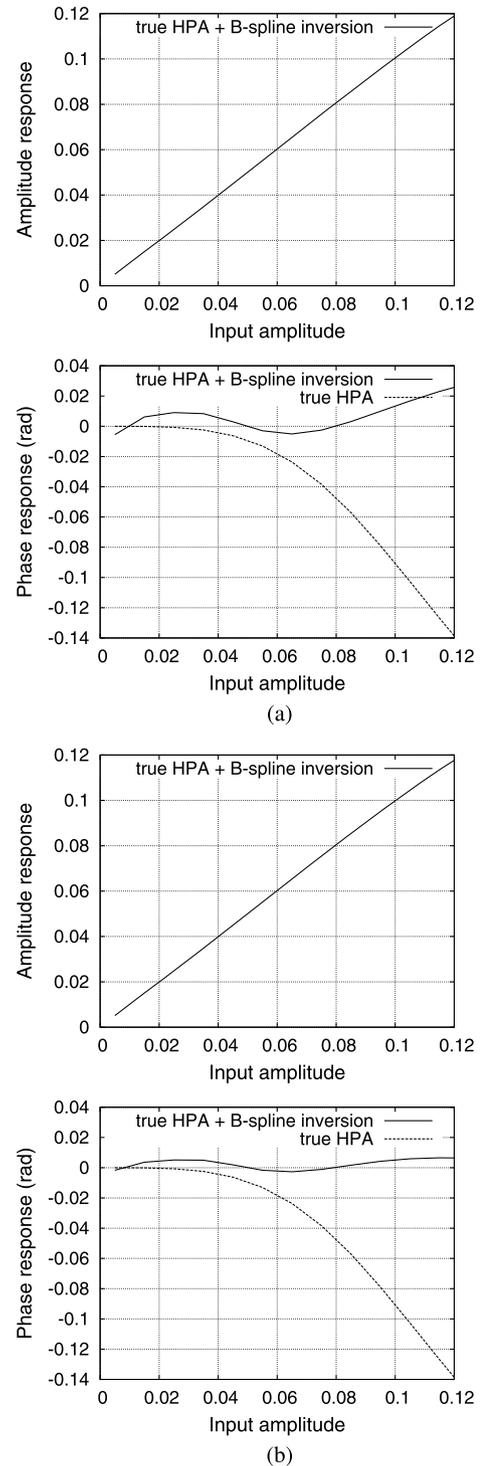


Fig. 6. Combined response of the true HPA and its estimated B-spline inversion under $\text{OBO} = 5$ dB and the SNR of: (a) $E_s/N_o = 5$ dB, and (b) $E_s/N_o = 10$ dB.

The results of Fig. 7 further illustrate the power of the proposed B-spline neural network based nonlinear equalizer for combating the dispersive channel as well as compensating the transmitter HPA's nonlinearity.

The achievable BER performance of the proposed nonlinear equalizer constructed based on the estimated CIR $\hat{\mathbf{h}}$ and the estimated B-spline inversion $\hat{\Phi}(\cdot)$ are depicted in Fig. 8 under the two operating conditions of the transmitter HPA, in comparison to the BER performance attained by the standard linear equalizer. It can be seen from Fig. 8 that as expected, the linear equalization alone

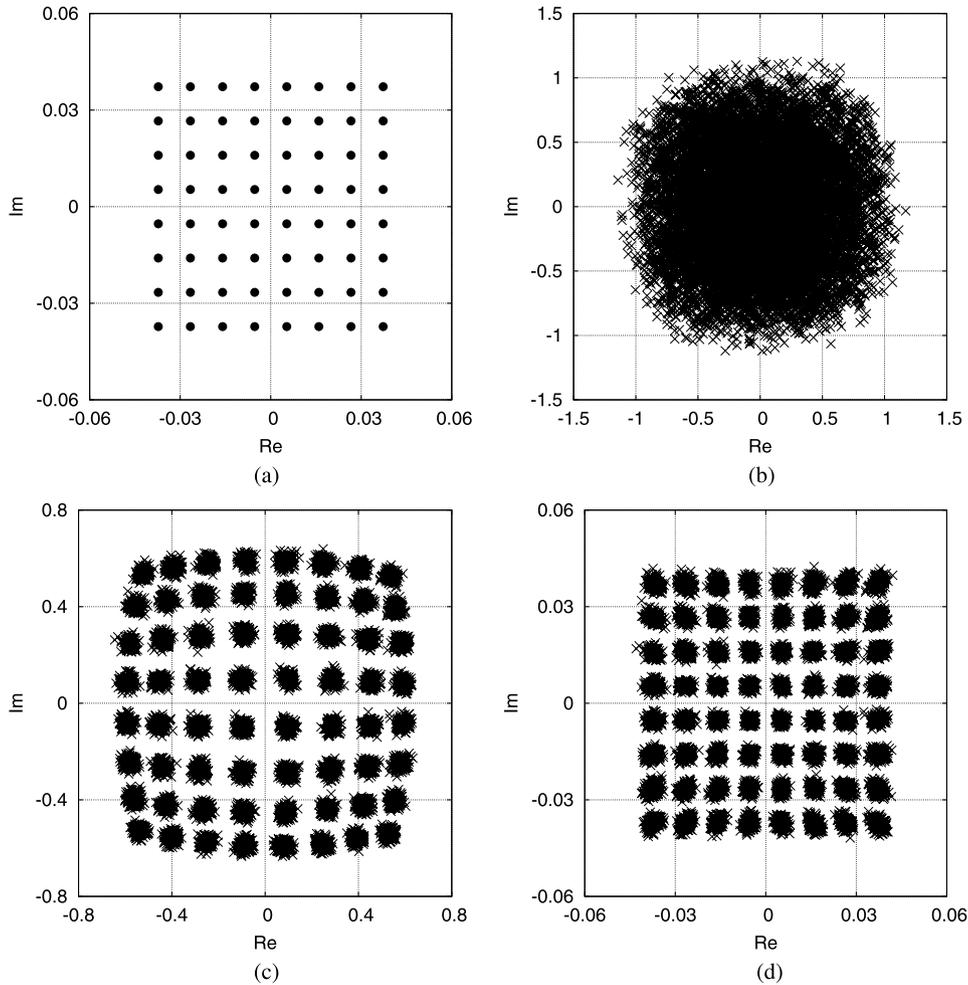


Fig. 7. The case of OBO = 8 dB and $E_s/N_o = 10$ dB: (a) the HPA's input signal $x(k)$, (b) the noisy Hammerstein channel's output signal $y(k)$, (c) the linearly equalized signal $\hat{w}(k)$ based on the estimated CIR, and (d) the nonlinearly equalized signal $\hat{x}(k)$ based on the estimated CIR and the estimated B-spline inversion of the HPA's nonlinearity.

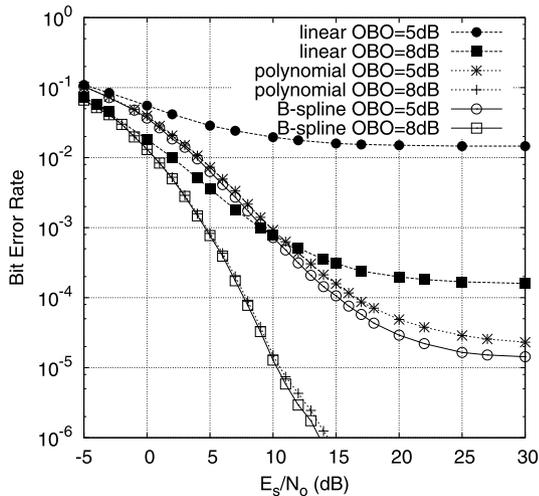


Fig. 8. The bit error rate performance comparison of the proposed CV B-spline neural network based nonlinear equalizer, the standard linear equalizer and the polynomial based nonlinear equalizer for the stationary channel case with the transmitter nonlinear HPA operating conditions given by OBO = 5 dB and OBO = 8 dB, respectively.

is incapable of compensating the transmitter HPA's nonlinearity. More specifically, for the 64-QAM signaling, the HPA operates at OBO = 5 dB exhibits the relatively severe nonlinearity and as a result, the linear equalizer performs poorly with a high error floor of

above 10^{-2} . By adopting the proposed nonlinear equalizer, a large part of the nonlinearity is removed, which enables the equalizer to lower the error floor by the three orders of magnitude, compared with the linear equalizer. Even when the transmitter HPA operates under the condition of OBO = 8 dB, which only exhibits relatively mild nonlinearity, the linear equalizer still shows an error floor of above 10^{-4} . By contrast, the proposed nonlinear equalizer significantly outperforms the linear equalizer, as confirmed in Fig. 8.

As mentioned in Section 3.1, two tensor-product polynomial models, both having a polynomial degree of P_o in each dimension, can also be utilized to estimate the CV HPA's static nonlinearity $\Psi(\bullet)$ and its inversion $\Psi^{-1}(\bullet)$, respectively, based on the same identification procedure developed in Section 3, yielding a polynomial based nonlinearity equalizer. In our simulation study, we also adopted the tensor-product polynomial model of degree $P_o = 4$, which had 25 basis functions that was comparable to the tensor-product B-spline model of at most 25 nonzero basis functions for any given input. The achievable BER performance of this polynomial based nonlinear equalizer are also shown in Fig. 8, where it can be seen that the performance of the polynomial based nonlinear equalizer is slightly inferior to that of the proposed B-spline based nonlinear equalizer, particularly when the HPA is operating in the severe nonlinear region.

4.2. Decision-directed adaptive performance

We then investigated the achievable performance of the proposed decision-directed adaptive B-spline nonlinear equalizer

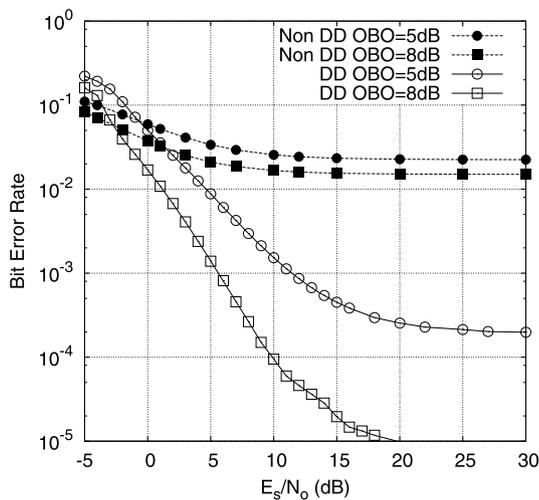


Fig. 9. The bit error rate performance comparison of the proposed decision-directed adaptive B-spline based nonlinear equalizer and the non-adaptive B-spline based nonlinear equalizer for the frame faded channel case with the transmitter nonlinear HPA operating conditions given by OBO = 5 dB and OBO = 8 dB, respectively.

based on the DD adaptive scheme of Algorithm 1, under the environment of frame-fading CIR, where the CIR tap coefficients faded at the beginning of each frame with the normalized Doppler frequency of $f_d = 0.0001$, but they remained constant within each frame. Fig. 9 compares the performance of the proposed DD adaptive B-spline nonlinear equalizer with that of the non-adaptive B-spline nonlinear equalizer which was fixed based on the training results obtained in the first frame. As expected, the non-adaptive nonlinear equalizer is unable to track the time-varying channel, and its BER performance is very poor. By contrast, with the aid of the DD adaptive scheme of Algorithm 1, the DD adaptive B-spline nonlinear equalizer is capable of tracking the time-varying channel reasonably well and, consequently, significantly improves the attainable BER performance. Observe from Fig. 9 that for the extremely low SNR conditions, the DD adaptive nonlinear equalizer actually performs worse than the non-adaptive one. Specifically, given the OBO = 5 dB and $\text{SNR} \leq -1$ dB as well as given the OBO = 8 dB and $\text{SNR} \leq -3$ dB the non-adaptive equalizer outperforms the DD adaptive equalizer. This is due to the well-known error propagation phenomenon. Under such extremely low SNR conditions, a large portion of the decisions are erroneous, which degrade the DD adaptive algorithm severely. Fig. 10 compares the performance of the DD adaptive B-spline based nonlinear equalizer with that of the DD adaptive polynomial based nonlinear equalizer. The results of Fig. 10 demonstrate that the DD adaptive B-spline based nonlinear equalizer outperforms the DD adaptive polynomial based nonlinear equalizer.

5. Conclusions

We have proposed a novel CV B-spline neural network based nonlinear equalizer for the nonlinear Hammerstein communication system that employs high-order QAM signaling with nonlinear transmitter high power amplifier and communicates over the dispersive channel. Specifically, we have extended a recent development of the CV B-spline neural network based approach to construct nonlinear equalizer for Hammerstein channels, which employs a CV B-spline neural network to model the HPA's nonlinearity as well as uses another CV B-spline neural network to model the inversion of the HPA's nonlinearity. During training, the Hammerstein channel model parameters that include the CIR coefficients and the B-spline neural network weights can readily be estimated using a highly efficient ALS algorithm, while the weights

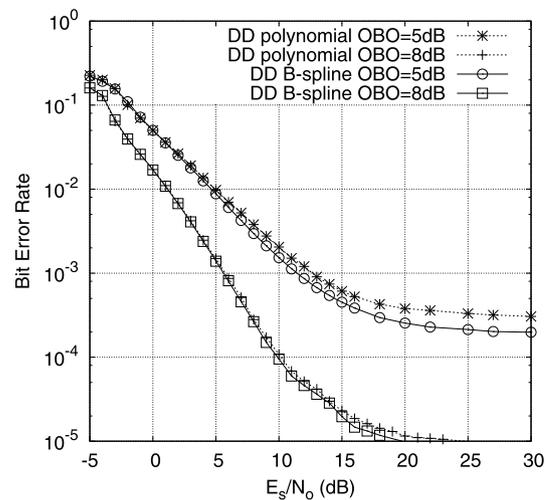


Fig. 10. The bit error rate performance comparison of the adaptive B-spline based nonlinear equalizer and the adaptive polynomial based nonlinear equalizer for the frame faded channel case with the transmitter nonlinear HPA operating conditions given by OBO = 5 dB and OBO = 8 dB, respectively.

of the B-spline inversion model can be identified using a standard LS algorithm based on the pseudo training data as a natural by-product of the Hammerstein channel model identification. Moreover, a decision-directed adaptive algorithm has been adopted to track the time-varying channel during data communication. The effectiveness of our proposed nonlinear equalization approach has been demonstrated in a simulation study, and the results obtained confirm that our B-spline neural network based nonlinear equalizer is capable of efficiently combating the dispersive transmission medium and compensating the detrimental transmitter HPA's nonlinearity as well as effectively tracking the time-varying channel.

References

- [1] L. Hanzo, S.X. Ng, T. Keller, W. Webb, *Quadrature Amplitude Modulation: From Basics to Adaptive Trellis-Coded, Turbo-Equalised and Space-Time Coded OFDM, CDMA and MC-CDMA Systems*, John Wiley, Chichester, UK, 2004.
- [2] A.A.M. Saleh, Frequency-independent and frequency-dependent nonlinear models of TWT amplifiers, *IEEE Trans. Commun.* COM-29 (11) (Nov. 1981) 1715–1720.
- [3] M. Honkanen, 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.
- [4] C.J. Clark, G. Chrisikos, M.S. Muha, A.A. Moulthrop, C.P. Silva, Time-domain envelope measurement technique with application to wideband power amplifier modeling, *IEEE Trans. Microw. Theory Tech.* 46 (12) (Dec. 1998) 2531–2540.
- [5] C.-S. Choi, et al., RF impairment models 60 GHz band SYS/PHY simulation, Document IEEE 802.15-06-0477-01-003c, <https://mentor.ieee.org/802.15/dcn/06/15-06-0477-01-003c-rf-impairment-models-60ghz-band-sysphy-simulation.pdf>, Nov. 2006.
- [6] V. Erceg, et al., 60 GHz impairments modeling, Document IEEE 802.11-09/1213r1, Nov. 2009.
- [7] L. Ding, G.T. Zhou, D.R. Morgan, Z. Ma, J.S. Kenney, J. Kim, C.R. Giardina, A robust digital baseband predistorter constructed using memory polynomials, *IEEE Trans. Commun.* 52 (1) (Jan. 2004) 159–165.
- [8] D. Zhou, V.E. DeBrunner, Novel adaptive nonlinear predistorters based on the direct learning algorithm, *IEEE Trans. Signal Process.* 55 (1) (Jan. 2007) 120–133.
- [9] M.-C. Chiu, C.-H. Zeng, M.-C. Liu, Predistorter based on frequency domain estimation for compensation of nonlinear distortion in OFDM systems, *IEEE Trans. Veh. Technol.* 57 (2) (March 2008) 882–892.
- [10] S. Choi, E.-R. Jeong, Y.H. Lee, Adaptive predistortion with direct learning based on piecewise linear approximation of amplifier nonlinearity, *IEEE J. Sel. Top. Signal Process.* 3 (3) (June 2009) 397–404.
- [11] V.P.G. Jiménez, Y. Jabrane, A.G. Armada, B. Ait Es Said, High power amplifier pre-distorter based on neural-fuzzy systems for OFDM signals, *IEEE Trans. Broadcast.* 57 (1) (March 2011) 149–158.
- [12] S. Chen, An efficient predistorter design for compensating nonlinear memory high power amplifier, *IEEE Trans. Broadcast.* 57 (4) (Dec. 2011) 856–865.

- [13] S. Chen, X. Hong, Y. Gong, C.J. Harris, Digital predistorter design using B-spline neural network and inverse of De Boor algorithm, *IEEE Trans. Circuits Syst. I* 60 (6) (June 2013) 1584–1594.
- [14] S. Chen, X. Hong, J.B. Gao, C.J. Harris, Complex-valued B-spline neural networks for modeling and inverting Hammerstein systems, *IEEE Trans. Neural Netw. Learn. Syst.* 25 (9) (Sept. 2014) 1673–1685.
- [15] J.G. Proakis, *Digital Communications*, 4th edition, McGraw-Hill, 2000.
- [16] S. Haykin, *Adaptive Filter Theory*, 2nd edition, Prentice Hall, Englewood, NJ, 1991.
- [17] X. Hong, S. Chen, Modeling of complex-valued Wiener systems using B-spline neural network, *IEEE Trans. Neural Netw.* 22 (5) (May 2011) 818–825.
- [18] X. Hong, S. Chen, C.J. Harris, Complex-valued B-spline neural networks for modeling and inverse of Wiener systems, Chapter 9 in: A. Hirose (Ed.), *Complex-Valued Neural Networks: Advances and Applications*, IEEE and Wiley, Hoboken, NJ, 2013, pp. 209–233.
- [19] C. De Boor, *A Practical Guide to Splines*, Springer-Verlag, New York, 1978.
- [20] C.J. Harris, X. Hong, Q. Gan, *Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach*, Springer-Verlag, Berlin, 2002.
- [21] J.M. Pena, B-spline and optimal stability, *Math. Comput.* 66 (220) (Oct. 1997) 1555–1560.
- [22] T. Lyche, J.M. Pena, Optimally stable multivariate bases, *Adv. Comput. Math.* 20 (1–3) (Jan. 2004) 149–159.
- [23] E. Mainar, J.M. Pena, Optimal stability of bivariate tensor product B-bases, *J. Numer. Anal. Ind. Appl. Math.* 6 (3–4) (2011) 95–104.
- [24] A.V. Ivanov, An asymptotic expansion for the distribution of the least squares estimator of the non-linear regression parameter, *Theory Probab. Appl.* 21 (3) (1977) 557–570.
- [25] C.-F. Wu, Asymptotic theory of nonlinear least squares estimation, *Ann. Stat.* 9 (3) (1981) 501–513.
- [26] J.H.K. Vuolevi, T. Rahkonen, J.P.A. Manninen, Measurement technique for characterizing memory effects in RF power amplifiers, *IEEE Trans. Microw. Theory Tech.* 49 (8) (Aug. 2001) 1383–1389.
- [27] X. Hong, R.J. Mitchell, S. Chen, System identification of Wiener systems with B-spline functions using De Boor recursion, *Int. J. Syst. Sci.* 44 (9) (Sept. 2013) 1666–1674.

Sheng Chen received his BEng degree from the East China Petroleum Institute, Dongying, China, in January 1982, and his PhD degree from the City University, London, in September 1986, both in control engineering. In 2005, he was awarded the higher doctoral degree, Doctor of Sciences (DSc), from the University of Southampton, Southampton, UK.

From 1986 to 1999, he held research and academic appointments at the Universities of Sheffield, Edinburgh and Portsmouth, all in UK. Since 1999, he has been with Electronics and Computer Science, the University of Southampton, UK, where he currently holds the post of Professor in Intelligent Systems and Signal Processing. Dr. Chen's research interests include adaptive signal processing, wireless communications, modelling and identification of nonlinear systems, neural network and machine learning, intelligent control system design, evolutionary computation methods and optimisation. He has published over 500 research papers.

Dr. Chen is a Fellow of IEEE and a Fellow of IET. He is a Distinguished Adjunct Professor at King Abdulaziz University, Jeddah, Saudi Arabia. Dr. Chen is an ISI highly cited researcher in engineering (March 2004). In 2014, he was elected as a Fellow of the United Kingdom Royal Academy of Engineering.

Xia Hong received her university education at National University of Defense Technology, P. R. China (BSc, 1984, MSc, 1987), and University of Sheffield, UK (PhD, 1998), all in automatic control.

She worked as a research assistant in Beijing Institute of Systems Engineering, Beijing, China from 1987–1993. She worked as a research fellow in the Department of Electronics and Computer Science at University of Southampton from 1997–2001. Since 2002, she has been with School of Systems Engineering, University of Reading, where she is currently a Professor. She is actively engaged in research into nonlinear systems identification, data modelling, estimation and intelligent control, neural networks, pattern recognition, learning theory and their applications. She has published over 100 research papers, and coauthored a research book.

Professor Hong was awarded a Donald Julius Groen Prize by IMechE in 1999.

Emad F. Khalaf received his BEng and MEng degrees in IT from Wroclaw University of Technology in Poland, in 1992, as one certificate, and the PhD degree in Computer networks from Wroclaw University of Technology, in Poland, in 2002.

From 2003 to 2011, he worked as an assistant professor at Computer Engineering Department, Faculty of Engineering, Philadelphia University, in Jordan. Since 2012 he is an assistant professor at Electrical and Computer Engineering Department, Faculty of Engineering, King Abdulaziz University, Jeddah, Saudi Arabia. Dr. Khalaf's research interests are in network security and cryptography, speech classification and recognition.

Ali Morfeq received the BS degree in computer engineering from the King Abdulaziz University, Saudi Arabia, in 1982, the MS degree in computer engineering from the Oregon state University, Corvallis, USA in 1985, and the PhD degree in computer science from the University of Colorado, Boulder, USA in 1990.

Since 1990, he has been with the King Abdulaziz University, Saudi Arabia, where currently he is an assistant professor and the chair of the Electrical & Computer Engineering Department, Faculty of Engineering. His research interests are in software engineering, and software systems for hospital applications.

Naif D. Alotaibi completed his BSc in Computer Engineering from the King Abdulaziz University in Jeddah, Saudi Arabia, in 2000, and received his Master and PhD degrees in Information Technology from Queensland University of Technology in Brisbane, Australia, in 2006 and 2012, respectively.

In 2013, he joined the Electrical and Computer Engineering Department at the Faculty of Engineering, King Abdulaziz University, Jeddah, Saudi Arabia as an assistant professor. His research interests include IT infrastructure, computer network quality and network QoS, and quality measurement.