Proceedings of International Joint Conference on Neural Networks, San Jose, California, USA, July 31 – August 5, 2011

B-spline neural network based digital baseband predistorter solution using the inverse of De Boor algorithm

Xia Hong, Yu Gong and Sheng Chen

***Abstract*— In this paper a new nonlinear digital baseband predistorter design is introduced based on direct learning, together with a new Wiener system modeling approach for the high power amplifiers (HPA) based on the B-spline neural network. The contribution is twofold. Firstly, by assuming that the nonlinearity in the HPA is mainly dependent on the input signal amplitude the complex valued nonlinear static function is represented by two real valued B-spline neural networks, one for the amplitude distortion and another for the phase shift. The Gauss-Newton algorithm is applied for the parameter estimation, in which the De Boor recursion is employed to calculate both the B-spline curve and the first order derivatives. Secondly, we derive the predistorter algorithm calculating the inverse of the complex valued nonlinear static function according to B-spline neural network based Wiener models. The inverse of the amplitude and phase shift distortion are then computed and compensated using the identified phase shift model. Numerical examples have been employed to demonstrate the efficacy of the proposed approaches.**

I. Introduction

The operation of high power amplifiers (HPA) in modern wireless communication systems introduces serious nonlinearities, potentially leading to the deterioration in system performance. The digital baseband predistorter is a functional device precedes the HPA. It acts as the inverse function of the HPA and predistorts the HPA input signal. The modeling of high power amplifiers (HPA) is crucial in any linearisation techniques of broadband communication systems employing power-efficient nonlinear HPA transmitter [1], [2]. Various HPA models and the associated predistorter (PD) design have been researched [3], [4], [5], [6]. The available digital baseband PD can be classified as indirect learning or direct learning. The indirect learning approaches are realised by initially identifying a post-inverse filter for the HPA followed by copying it into the PD [7], [8], [9]. The direct learning PD is obtained by initially identifying the input-output relation of the HPA, followed by designing the PD based on the identified HPA model [10], [11], [2], [12]. The indirect learning PD is simpler to implement but it may suffer from modelling bias due to error propagation from noise, thus the deterioration in performance. Thus it is highly desirable to develop highly efficient PD approaches based on direct learning.

The Wiener model has been used as a model for some industrial/biological systems [13], [14], [15], [16], and notably the HPA in broadband communication transmitters [17]. Various approaches have been developed [18], [19], [20], [14], [21]. The model characterization/representation of the unknown nonlinear static function in the Wiener model is fundamental to its identification, control and/or other signal processing applications. Based on the approximation theory, the polynomial functions are appropriate in approximating the unknown nonlinear static functions. The B-spline curves consist of many polynomial pieces, offering versatility. The early work on the construction of B-spline curve is mathematically involved and numerically unstable [22]. The De Boor algorithm uses recurrence relations and is numerically stable [22]. The B-spline basis functions for nonlinear systems modelling have been widely applied [23].

In this work a new digital baseband predistorter solution is introduced based on direct learning. We initially consider HPA as complex valued Wiener system. Without losing generality it is assumed that the nonlinearity in the Wiener model is mainly dependent on the input signal amplitude. Consequently the complex valued nonlinear static function is represented by two real valued B-spline neural networks, one for the amplitude distortion and another for the phase shift, respectively. By minimizing the mean square error (MSE) between the model output and the system output, the Gauss-Newton algorithm is readily applicable for the parameter estimation in the proposed model. The Gauss-Newton algorithm is applied, which incorporates with the De Boor algorithm, including both the B-spline curve and the first order derivatives recursion. The proposed model based on B-spline functions is advantageous as this offers modeling versatility as well as enables stable and efficient evaluations of functional and derivative values, as required in nonlinear optimization algorithm. The Wiener model for memory HPAs is described in Section II, followed by the proposed new B-spline Wiener system modeling approach that is introduced in Section III.

Because the predistorter acts as the inverse function of the HPA and predistorts the HPA input signal, clearly the resultant PD is a Hammerstein model that is the inverse of the complex valued nonlinear static function, followed by the inverse of the linearity filter in the Wiener system. In Section IV a new PD design is introduced, including the efficient algorithm to calculate the inverse of the nonlinear static function based on the proposed Wiener models. The inverse of the amplitude distortion is computed using the Newton-Raphson formula combining with De Boor algorithms, referred as the inverse of De Boor algorithm, followed by the calculation of the phase distortion using the identified phase shift model. Numerical examples have been employed to demonstrate the

Xia Hong and Yu Gong are with School of Systems Engineering, University of Reading,UK (email: x.hong@reading.ac.uk), Sheng Chen is with School of Electronics and Computer Science, University of Southampton,UK

978-1-4244-9636-5/11/$26.00 ©2011 IEEE

efficacy of the proposed approaches.

II. The Wiener model for memory HPAs

The general complex valued Wiener system consists of a cascade of two subsystems, a linear filter of order n representing the memory effect on the input signal as the first subsystem, followed by a nonlinear memoryless function $\Psi(\bullet) : \mathcal{C} \to \mathcal{C}$ as the second subsystem. The system can be represented by

$$\begin{aligned} w(t) &= H(z)y(t) \\ &= y(t) + h_1 y(t-1) ... + h_n y(t-n) \qquad (1) \\ d(t) &= \Psi\left(w(t)\right) + \xi(t) \qquad (2) \end{aligned}$$

with z transfer function $H(z)$ defined by

$$H(z) = \sum_{i=0}^{n} h_i z^{-i}, \quad h_0 = 1 \qquad (3)$$

where $d(t) = d_R(t) + j \cdot d_I(t) \in \mathcal{C}$ is the system output and $y(t) = y_R(t) + j \cdot y_I(t) \in \mathcal{C}$ is the system input. $j = \sqrt{-1}$. $\xi(t) = \xi_R(t) + j\xi_I(t) \in \mathcal{C}$ is assumed to be a white complex valued noise sequence independent of $y(t)$. Both $\xi_R(t)$ and $\xi_I(t)$ are zero mean and have a variance of σ^2. $w(t) = w_R(t) + j \cdot w_I(t) \in \mathcal{C}$ is the output of linear filter subsystem and the input to the nonlinear subsystem. $h_i = h_{i,R} + j \cdot h_{i,I}$, $(i = 1, \cdots, n)$ are complex valued coefficients of the linear filter. n is assumed known. Denote $\mathbf{h} = [h_1, ..., h_n]^T \in \mathcal{C}^n$.

For the baseband HPA model, $\Psi\left(w(t)\right)$ can be specified by a nonlinearity of the traveling wave tube (TWT) [17]. The input to the TWT nonlinearity can be expressed as

$$w(t) = |w(t)| \exp(j \angle w(t)) = r(t) \exp(j \cdot \phi(t)) \qquad (4)$$

where $r(t) = \sqrt{w_R^2(t) + w_I^2(t)}$ and $\phi(t) = \arctan(w_I(t)/w_R(t))$, denoting the amplitude and phase of $w(t)$ respectively. The output of TWT, $\Psi\left(w(t)\right)$, is distorted in both the amplitude and the phase, with the distortion dependent mainly on the input signal amplitude, i.e. $r(t)$. So $\Psi\left(w(t)\right)$ can be expressed by [17]

$$\begin{aligned} \Psi\left(w(t)\right) &= \left|\Psi\left(w(t)\right)\right| \exp(j \cdot \angle \Psi\left(w(t)\right)) \\ &= r_\Psi(t) \exp(j \cdot [\phi_\Psi(t) + \phi(t)]) \qquad (5) \end{aligned}$$

where $r_\Psi(t)$ and $\phi_\Psi(t)$ denote the amplitude and phase shift by $\Psi\left(w(t)\right)$ respectively, and these are given by

$$r_\Psi(t) = \left\{ \begin{array}{cc} \frac{\alpha_1 r(t)}{1+\alpha_2 r^2(t)}, & 0 \le r(t) \le r_{Sat} \\ \Psi_{max}, & r(t) > r_{Sat} \end{array} \right. \qquad (6)$$

$$\phi_\Psi(t) = \frac{\beta_1 r^2(t)}{1 + \beta_2 r^2(t)}. \qquad (7)$$

where $\alpha_1, \alpha_2, \beta_1, \beta_2$ are unknown parameters. $r_{Sat} = \sqrt{\frac{1}{\alpha_2}}$ and $\Psi_{max} = \frac{\alpha_1}{2\sqrt{\alpha_1}}$. Our aim is the system identification for the above HPA model, i.e. given an observational input/output data set $D_N = \{y(t), d(t)\}_{t=1}^{K}$, to identify the underlying nonlinear function $\Psi(\bullet)$ and to estimate the parameters h_i's of the linear filter simultaneously. Note that the signal $w(t)$ between the two subsystems are unavailable. In this work we propose to model $\Psi(\bullet)$ by using B-spline curves. Specifically, $r_\Psi(t)$ and $\phi_\Psi(t)$ are modeled by two univariate B-spline networks (B-spline curves) respectively as described in the following.

III. The system identification algorithm

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

Univariate B-spline basis functions are parameterized by the order of a piecewise polynomial of order k, and also by a knot vector which is a set of values defined on the real line that break it up into a number of intervals. Supposing that there are d basis functions, the knot vector is specified by $(M+k)$ knot values, $\{R_1, R_2, \cdots, R_{M+k}\}$. At each end there are k knots satisfying the condition of being external to the input region, and as a result the number of internal knots is $(M-k)$. Specifically

$$\begin{aligned} & R_1 < R_2 < R_k = R_{min} < R_{k+1} < R_{k+2} < \\ & \cdots < R_M < R_{max} = R_{M+1} < \cdots < R_{M+k}. \qquad (8) \end{aligned}$$

Given these predetermined knots, a set of M B-spline basis functions can be formed by using the De Boor recursion [22], given by

$$\mathcal{B}_l^{(0)}(r) = \left\{ \begin{array}{ll} 1 & \text{if } R_l \le r < R_{l+1} \\ 0 & \text{otherwise} \end{array} \right. \qquad (9)$$

$$l = 1, \cdots, (M+k)$$

$$\left. \begin{array}{c} \mathcal{B}_l^{(i)}(r) = \frac{r-R_l}{R_{i+l}-R_l} \mathcal{B}_l^{(i-1)}(r) \\ + \frac{R_{i+l+1}-r}{R_{i+l+1}-R_{l+1}} \mathcal{B}_{l+1}^{(i-1)}(r), \\ l = 1, \cdots, (M+k-i) \end{array} \right\}$$

$$i = 1, \cdots, k \qquad (10)$$

The derivative of B-spline basis function $\mathcal{B}_l^{(k)}(r)$ can be readily computed as

$$\begin{aligned} \frac{d}{dr}[\mathcal{B}_l^{(k)}(r)] &= \frac{k}{R_{k+l} - R_l} \mathcal{B}_l^{(k-1)}(r) \\ &\quad - \frac{k}{R_{k+l+1} - R_{l+1}} \mathcal{B}_{l+1}^{(k-1)}(r) \\ &\quad l = 1, \cdots, M \qquad (11) \end{aligned}$$

We model $\Psi(\bullet)$ as two univariate B-spline neural networks [23], one for the amplitude and another for the phase shift, in the form of

$$r_\Psi(t) = \sum_{l=1}^{M} \mathcal{B}_l^{(k)}(r(t))\omega_l \qquad (12)$$

$$\phi_\Psi(t) = \sum_{l=1}^{M} \mathcal{B}_l^{(k)}(r(t))\vartheta_l \qquad (13)$$

and their derivatives are in the form of

$$r'_\Psi(t) = \sum_{l=1}^{M} \frac{d}{dr(t)} \mathcal{B}_l^{(k)}(r(t))\omega_l \qquad (14)$$

$$\phi'_\Psi(t) = \sum_{l=1}^{M} \frac{d}{dr(t)} \mathcal{B}_l^{(k)}(r(t))\vartheta_l \qquad (15)$$

where ω_l's and ϑ_l's are weights to be determined. Denote $\boldsymbol{\omega} = [\omega_1, \cdots, \omega_M]^T \in \Re^M$ and $\boldsymbol{\vartheta} = [\vartheta_1, \cdots, \vartheta_M]^T \in \Re^M$. Note that due to the piecewise nature of B-spline functions, there are only $(k+1)$ basis functions with nonzero functional/derivative values at any point r. Hence the computational cost of the De Boor algorithm is determined by the polynomial order k, rather than the number of knots, and this is in the order of $O(k^2)$.

B. The main algorithm

The complex valued B-spline network output is denoted by $\hat{d}(t) = \hat{d}_R(t) + j \cdot \hat{d}_I(t)$ in which

$$\begin{array}{rcl} \hat{d}_R(t) & = & r_\Psi(t)\cos[\phi_\Psi(t) + \phi(t)] \\ \hat{d}_I(t) & = & r_\Psi(t)\sin[\phi_\Psi(t) + \phi(t)] \end{array}$$

Let the error between the Wiener system output $d(t)$ and the B-spline network output $\hat{d}(t)$ be denoted by $e(t) = d(t) - \hat{d}(t) = e_R(t) + j \cdot e_I(t) \in \mathcal{C}$. Our task is to estimate $\mathbf{h}$, $\boldsymbol{\omega}$ and $\boldsymbol{\vartheta}$. This could be achieved by minimizing

$$J = \sum_{t=1}^{K}[e_R(t)]^2 + \sum_{t=1}^{K}[e_I(t)]^2 \qquad (16)$$

Denote $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \cdots, \epsilon_{2K}]^T = [e_R(1), \cdots, e_R(K), e_I(1), \cdots, e_I(K)]^T \in \Re^{2K}$, $\boldsymbol{\theta} = [\theta_1, \theta_2, \cdots, \theta_{2(d+n)}]^T = [\omega_1, \omega_2, \cdots, \omega_M, \vartheta_1, \vartheta_2, \cdots, \vartheta_M, h_{1,R}, \cdots, h_{n,R}, h_{1,I}, \cdots, h_{n,I}]^T \in \Re^{2(M+n)}$. Note that from (1)

$$\left\{ \begin{array}{rcl} w_R(t) & = & y_R(t) + \sum_{i=1}^{n}(h_{i,R}y_R(t-i) - h_{i,I}y_I(t-i)) \\ w_I(t) & = & y_I(t) + \sum_{i=1}^{n}(h_{i,I}y_R(t-i) + h_{i,R}y_I(t-i)) \end{array} \right. \qquad (17)$$

and for $i = 1, \cdots, n$, we have

$$\left\{ \begin{array}{rcl} \frac{\partial w_R(t)}{\partial h_{i,R}} & = & y_R(t-i) \\ \frac{\partial w_R(t)}{\partial h_{i,I}} & = & -y_I(t-i) \\ \frac{\partial w_I(t)}{\partial h_{i,R}} & = & y_I(t-i) \\ \frac{\partial w_I(t)}{\partial h_{i,I}} & = & y_R(t-i) \end{array} \right. \qquad (18)$$

Thus it can be shown that

$$\left\{ \begin{array}{rcl} \frac{\partial r(t)}{\partial h_{i,R}} & = & \frac{1}{r(t)}[w_R(t)y_R(t-i) + w_I(t)y_I(t-i)] \\ \frac{\partial r(t)}{\partial h_{i,I}} & = & \frac{1}{r(t)}[w_I(t)y_R(t-i) - w_R(t)y_I(t-i)] \end{array} \right. \qquad (19)$$

and

$$\left\{ \begin{array}{rcl} \frac{\partial \phi(t)}{\partial h_{i,R}} & = & \frac{1}{r^2(t)}[w_R(t)y_I(t-i) - w_I(t)y_R(t-i)] \\ \frac{\partial \phi(t)}{\partial h_{i,I}} & = & \frac{1}{r^2(t)}[w_R(t)y_R(t-i) + w_I(t)y_I(t-i)] \end{array} \right. \qquad (20)$$

We denote an iteration step variable by a superscript $^{(\tau)}$, and apply the Gauss Newton algorithm as follows.

The Gauss Newton algorithm combined with the De Boor algorithm: With an initial $\boldsymbol{\theta}^{(0)}$, the iteration formula is given by

$$\boldsymbol{\theta}^{(\tau)} = \boldsymbol{\theta}^{(\tau-1)} - \alpha\{[\mathbf{J}^{(\tau)}]^T\mathbf{J}^{(\tau)}\}^{-1}[\mathbf{J}^{(\tau)}]^T\boldsymbol{\epsilon}(\boldsymbol{\theta}^{(\tau-1)}) \qquad (21)$$

where $\alpha > 0$ is a small positive step size. $\mathbf{J}$ denotes the Jacobian of $\boldsymbol{\epsilon}(\boldsymbol{\theta})$ and is given by $\mathbf{J} = \{\frac{\partial \epsilon_p}{\partial \theta_q}\}$, where for $p = 1, \cdots, K$, and $t = p$.

$$\frac{\partial \epsilon_p}{\partial \theta_q} = \left\{ \begin{array}{l} \frac{\partial e_R(t)}{\partial \omega_l} = -\mathcal{B}_l^{(k)}(r(t))\cos[\phi_\Psi(t) + \phi(t)], \\ \quad \text{for} \;\; q = 1, \cdots, M, \;\; (l = q) \\ \frac{\partial e_R(t)}{\partial \vartheta_l} = r_\Psi(t)\sin[\phi_\Psi(t) + \phi(t)]\mathcal{B}_l^{(k)}(r(t)), \\ \quad \text{for} \;\; q = M+1, \cdots, 2M, \;\; (l = q - M) \\ \frac{\partial e_R(t)}{\partial h_{i,R}} = -\left\{r'_\Psi(t)\cos[\phi_\Psi(t) + \phi(t)]\right. \\ \quad \left. -r_\Psi(t)\sin[\phi_\Psi(t) + \phi(t)]\phi'_\Psi(t)\right\}\frac{\partial r(t)}{\partial h_{i,R}} \\ \quad +r_\Psi(t)\sin[\phi_\Psi(t) + \phi(t)]\frac{\partial \phi(t)}{\partial h_{i,R}}, \\ \quad \text{for} \;\; q = 2M+1, \cdots, 2M+n, \\ \quad (i = q - 2M) \\ \frac{\partial e_R(t)}{\partial h_{i,I}} = -\left\{r'_\Psi(t)\cos[\phi_\Psi(t) + \phi(t)]\right. \\ \quad \left. -r_\Psi(t)\sin[\phi_\Psi(t) + \phi(t)]\phi'_\Psi(t)\right\}\frac{\partial r(t)}{\partial h_{i,I}} \\ \quad +r_\Psi(t)\sin[\phi_\Psi(t) + \phi(t)]\frac{\partial \phi(t)}{\partial h_{i,I}}, \\ \quad \text{for} \; q = 2M+n+1, \cdots, 2(M+n), \\ \quad (i = q - 2M - n) \end{array} \right. \qquad (22)$$

and for $p = K+1, \cdots, 2K$, and $t = (p - K)$

$$\frac{\partial \epsilon_p}{\partial \theta_q} = \left\{ \begin{array}{l} \frac{\partial e_I(t)}{\partial \omega_l} = -\mathcal{B}_l^{(k)}(r(t))\sin[\phi_\Psi(t) + \phi(t)], \\ \quad \text{for} \;\; q = 1, \cdots, M, \;\; (l = q) \\ \frac{\partial e_I(t)}{\partial \vartheta_l} = -r_\Psi(t)\cos[\phi_\Psi(t) + \phi(t)]\mathcal{B}_l^{(k)}(r(t)), \\ \quad \text{for} \;\; q = M+1, \cdots, 2M, \;\; (l = q - M) \\ \frac{\partial e_I(t)}{\partial h_{i,R}} = -\left\{r'_\Psi(t)\sin[\phi_\Psi(t) + \phi(t)]\right. \\ \quad \left. +r_\Psi(t)\cos[\phi_\Psi(t) + \phi(t)]\phi'_\Psi(t)\right\}\frac{\partial r(t)}{\partial h_{i,R}} \\ \quad -r_\Psi(t)\cos[\phi_\Psi(t) + \phi(t)]\frac{\partial \phi(t)}{\partial h_{i,R}}, \\ \quad \text{for} \;\; q = 2M+1, \cdots, 2M+n, \\ \quad (i = q - 2M) \\ \frac{\partial e_I(t)}{\partial h_{i,I}} = -\left\{r'_\Psi(t)\sin[\phi_\Psi(t) + \phi(t)]\right. \\ \quad \left. +r_\Psi(t)\cos[\phi_\Psi(t) + \phi(t)]\phi'_\Psi(t)\right\}\frac{\partial r(t)}{\partial h_{i,I}} \\ \quad -r_\Psi(t)\cos[\phi_\Psi(t) + \phi(t)]\frac{\partial \phi(t)}{\partial h_{i,I}} \\ \quad \text{for} \;\; q = 2M+n+1, \cdots, 2(M+n) \\ \quad (i = q - 2M - n) \end{array} \right. \qquad (23)$$

Note that we propose that the De Boor algorithm (9)-(11) are utilized for evaluating (12)-(15), which are then applied in evaluating the entries for (22)-(23). In addition (19)-(20) are also used to calculating the entries for (22)-(23). The iterative equation (21) can be terminated when $\boldsymbol{\theta}^{(\tau)}$ converges, or by predetermining a sufficiently large number of iterations. Using the final model parameters it is straightforward to produce estimated value of $\hat{r}_{Sat}$ and $\hat{\Psi}_{max}$ using numerical search that will then be used in PD design in Section IV.

As the objective function of (16) is highly nonlinear, the solution of Gauss Newton algorithm is dependent on the initial condition. It is important that $\boldsymbol{\theta}^{(0)}$ is properly initialized so that it is as closer as possible to optimal solution, and it is also desirable the parameter initialization is simple to implement.

C. A simple parameter initialization using least squares algorithm

Initially a set of $(M+k)$ knot vector is predetermined that breaks the domain of $r(t)$ up, with k knots satisfying the condition of being external to the regions for $r(t)$ at each end. In this work, the parameter $\boldsymbol{\theta}^{(0)}$ is initialized as follows.

1) Initialize $h_i^{(0)} = 0$, i.e. $h_{i,R}^{(0)} = 0$, $h_{i,I}^{(0)} = 0$, for $i = 1, ..., n$.
2) Generate a sequence $r^{(0)}(t) = \sqrt{y_R^2(t) + y_I^2(t)}$, for $t = 1, ..., K$.
3) Generate a sequence $r_d^{(0)}(t) = \sqrt{d_R^2(t) + d_I^2(t)}$, for $t = 1, ..., K$. Denote $\mathbf{r}_d = [r_d^{(0)}(1), ..., r_d^{(0)}(K)]^T$.
4) Generate a sequence $\phi_\Psi^{(0)}(t) = \arctan(\frac{d_I(t)}{d_R(t)}) - \arctan(\frac{y_I(t)}{y_R(t)})$, and then limit the values within $[\frac{\pi}{2}, \frac{\pi}{2}]$ as appropriate (by adding a multiple of $\pm 2\pi$ if out of range), for $t = 1, ..., K$. Denote $\boldsymbol{\phi}_\Psi = [\phi_\Psi^{(0)}(1), ..., \phi_\Psi^{(0)}(K)]^T$.
5) Form a regression matrix

$$\mathbf{B} = \begin{bmatrix} \mathcal{B}_1^{(k)}(r^{(0)}(1)) & \cdots & \mathcal{B}_M^{(k)}(r^{(0)}(1)) \\ \vdots & \ddots & \vdots \\ \vdots & \mathcal{B}_l^{(k)}(r^{(0)}(t)) & \vdots \\ \vdots & \ddots & \vdots \\ \mathcal{B}_1^{(k)}(r^{(0)}(K)) & \cdots & \mathcal{B}_M^{(k)}(r^{(0)}(K)) \end{bmatrix} \in \Re^{K \times M} \tag{24}$$

6) Compute the least square estimate $\boldsymbol{\omega}^{(0)} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{r}_d$, and $\boldsymbol{\vartheta}^{(0)} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\boldsymbol{\phi}_\Psi$.
7) Finally set $\boldsymbol{\theta}^{(0)} = [\underbrace{0, ..., 0}_{2n}, \ [\boldsymbol{\omega}^{(0)}]^T, \ [\boldsymbol{\vartheta}^{(0)}]^T]^T$.

IV. New predistorter solution using the inverse of De Boor algorithm

In Section III, the models of the amplitude distortion and the phase shift of the HPA have been constructed as two univariate B-spline curves respectively. The proposed PD to the Wiener HPA, as shown in Figure 1, is a Hammerstein model with a nonlinear static function followed by a linear filter, which is obtained as the inverse of the identified Wiener model. Specifically the Hammerstein model consists of $\Psi^{-1}(\bullet)$, the inverse of the nonlinearity in Wiener system $\Psi(\bullet)$, followed by the linear filter $G(z)$ that is the inverse of the linearity filter in Wiener system $H(z)$.

Without losing generality we make the following assumption.

Assumption 1: For $0 \leq r(t) \leq r_{Sat}$, both $r_\Psi(t)$ are $\phi_\Psi(t)$ are one to one mappings. That is, these can be regarded as invertible and continuous functions for our PD design problem.

Initially denote the input signal to the PD as $x(t) = x_R(t) + j \cdot x_I(t) \in \mathcal{C}$ and in polar form,

$$x(t) = |x(t)| \exp(j\angle x(t)) = r_x(t) \exp(j \cdot \phi_x(t)) \tag{25}$$

where $r_x(t) = \sqrt{x_R^2(t) + x_I^2(t)}$ and $\phi_x(t) = \arctan(x_I(t)/x_R(t))$, denoting the amplitude and phase of $x(t)$ respectively. Let the output $\Psi^{-1}(\bullet)$ be $v(t) = v_R(t) + j \cdot v_I(t) \in \mathcal{C}$, and in polar form,

$$v(t) = |v(t)| \exp(j\angle v(t)) = r_v(t) \exp(j \cdot \phi_v(t)) \tag{26}$$

where $r_v(t) = \sqrt{v_R^2(t) + v_I^2(t)}$ and $\phi_v(t) = \arctan(v_I(t)/v_R(t))$, denoting the amplitude and phase of $v(t)$ respectively.

Assumption 2: For $r_x(t) > \hat{\Psi}_{max}$, $r_\Psi^{-1}(t)$ in Figure 1 is defined as $\hat{r}_{Sat}$.

The procedure of calculating $\Psi^{-1}(\bullet)$*:*

1) Calculate the inverse of the amplitude distortion $r_v(t) = r_\Psi^{-1}(r_x(t))$ for $r_x(t)$ using the inverse of De Boor algorithm (see Section IV-A) if $0 \leq r(t) \leq r_{Sat}$. Otherwise return $r_v(t) = \hat{r}_{Sat}$.
2) Calculate the phase distortion as $\phi_\Psi(r_v(t))$ using (13).
3) Calculate the inverse of the phase distortion by $\phi_v(t) = \phi_x(t) - \phi_\Psi(r_v(t))$.

The linear filter $G(z) = \frac{1}{H(z)}$ is given by

$$G(z) = 1 + g_1 z^{-1} + ... + g_{n_g} z^{-n_g}, \tag{27}$$

in which n_g is predetermined as a sufficiently large integer, and g_i's are obtained using long division. Clearly the signal $v(t)$ acts as the input to $G(z)$, and the output of $G(z)$ acts as the input of HPA system, $y(t)$.

A. The inverse of De Boor algorithm

In Section III, the parameters ω_j have been found and can be used in (12) to calculate the amplitude distortion $r_\Psi(t)$. In this section we consider the problem of finding its inverse, $r_v = r_\Psi^{-1}(r_x)$ (i.e. Step 1 of Procedure of calculating $\Psi^{-1}(\bullet)$). Or given that r_x lies in the region between two points, $r_\Psi(R_{min})$ and $r_\Psi(R_{max})$, we aim to find the root of the polynomial equation of $r_x = \sum_{l=1}^{M} \mathcal{B}_l^{(k)}(r_v)\omega_l$. We propose to solve the problem using the inverse of De Boor algorithm described below. This algorithm effectively builds the De Boor algorithms including the B-spline curve and the first order derivatives recursion, into the Newton-Raphson formula that is modified to handle the constraint that r_v must be positive. Note that from Assumption 1, $r_\Psi(t)$ is monotonic, and this means the inverse of De Boor algorithm converges to the unique solution.

The algorithm:

1) Initialize $r_v^{(0)}$ as a random number with $R_{min} < r_v^{(0)} < R_{max}$.
2) The $(\tau+1)$th step is given by

$$\begin{aligned} r_v^{(\tau+1)} &= r_v^{(\tau)} + \Delta r_v^{(\tau)} \\ &= r_v^{(\tau)} + \eta \cdot \frac{\left(r_x - r_\Psi(r_v^{(\tau)})\right)}{r'_\Psi(r_v^{(\tau)})} \end{aligned} \tag{28}$$

$$r_v^{(\tau+1)} = \max\{r_v^{(\tau+1)}, 0\} \tag{29}$$

where $0 < \eta \ll 1$ is the learning rate, that is preset empirically. $r_\Psi(r_v^{(\tau)})$ and $r'_\Psi(r_v^{(\tau)})$ are calculated using

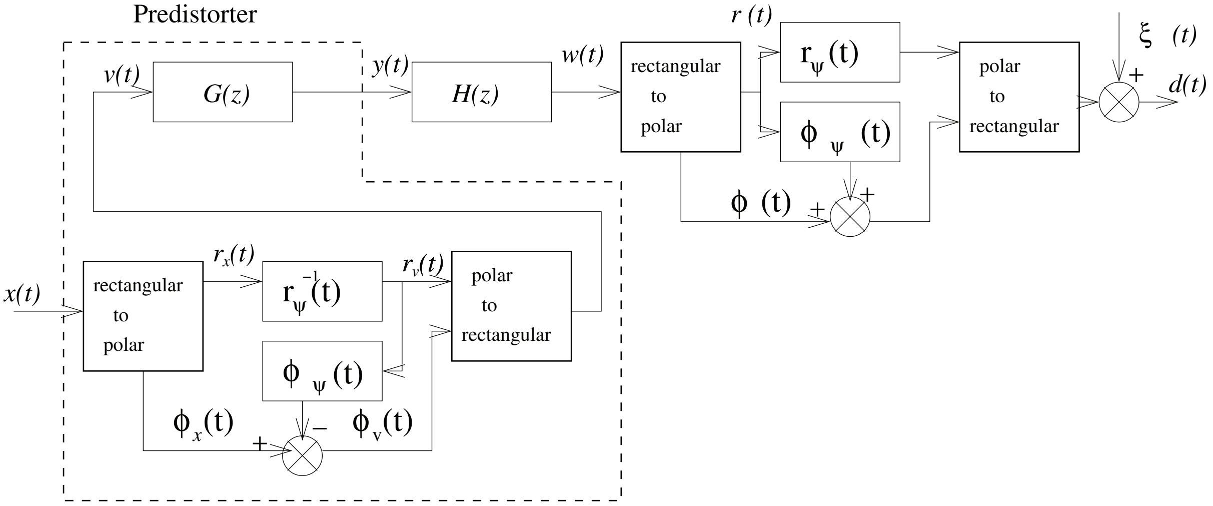


Fig. 1. The predistorter design using the Hammerstein model.

(12) and (14), in which the De Boor recursions (9)-(11) are utilized.

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

The computational cost of procedure of calculating $\Psi^{-1}(\bullet)$ is due to the cost of the inverse of De-Boor algorithm. This is very low is at $O(k^2)$, scaled by the number of iterations.

V. Experimental results

In this section, numerical examples are given to verify the proposed PD approach. We first show the performance of the HPA model identification with the B-spline approach, and then show some of the numerical results for the PD design with the inverse De Boor algorithm. The HPA is assumed to be Wiener model in the simulation.

A. HPA model identification

The approach described in Section III is used to identify the HPA model. 2000 training data samples and 500 validations data samples $d(t)$ were generated by using (1) and (2) (via (6) & (7)), where $H(z) = 1 + 0.7692z^{-1} + 0.1538z^{-2} + 0.0769z^{-3}$, in which the TWT nonlinearity is used to generate the training data set and specified by $\alpha_1 = 2.1587$, $\alpha_2 = 1.15$, $\beta_1 = 4$ and $\beta_2 = 2.1$ respectively. $y(t)$ was uniformly distributed complex random variable with $y_R(t) \in [-0.85, 0.85]$ and $y_I(t) \in [-0.85, 0.85]$. The variances of the additive noise to the system output are set $\sigma^2 = 0$. The polynomial degree of the B-spline basis function was set as three (i.e. $k = 4$, piecewise cubic). The following predetermined knot sequence

$$[-0.00002, -0.00001, -1e-6, 0.01,$$
$$0.2, 0.6, 0.8, 1.5, 1.9, 3, 5]$$

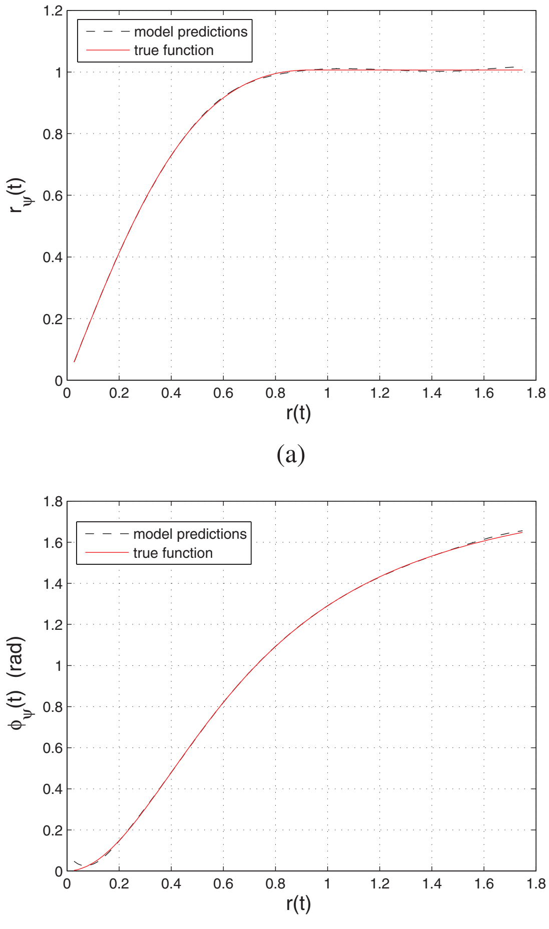
is initially set for $r(t)$ in order to generate basis functions. The system identification algorithm as described in Section III-B was carried out followed by the parameter initialization as described in Section III-C. The modelling results are shown in Table I for the linear subsystem. It is shown that the proposed system identification method is very effective in capturing the true model parameters. In order to demonstrate the the nonlinear approximation capability. The model predictions of the two B-spline models $r_\Psi(t)$ and $\phi_\Psi(t)$ were reconstructed over the validation data set, and this is compared with the true model used to generate the data set in Figure 2 (a) and (b). It is shown that the proposed approaches have excellent approximation results for modeling the complex valued nonlinear static function.

TABLE I
Results of linear subsystem parameter estimation.

	h_1	h_2	h_3
True parameters	0.7692	0.1538	0.0769
Initial estimates	0	0	0
Final estimates	$0.7692-$ $j1.7 \times 10^{-5}$	$0.1537+$ $j1.5 \times 10^{-5}$	$0.0769+$ $j8.2 \times 10^{-6}$

B. PD design

With the HPA model identified as above, experiments are performed to verify the proposed PD design as described in Section IV . The symbol modulation is 16-QAM and the pulse shaping is achieved with the square root raise cosine with roll-off factor of 0.02. For comparison, the numerical results without predistortion, along with that of the indirect learning approach [1] are also shown, where the memory order Q and nonlinearity order K denoted in [1] are set as



(a)

(b)

Fig. 2. The TWT nonlinearity modeling results; (a) Amplitude distortion with respect to the amplitude of the input; and (b) Phase shift with respect to the amplitude of the input.

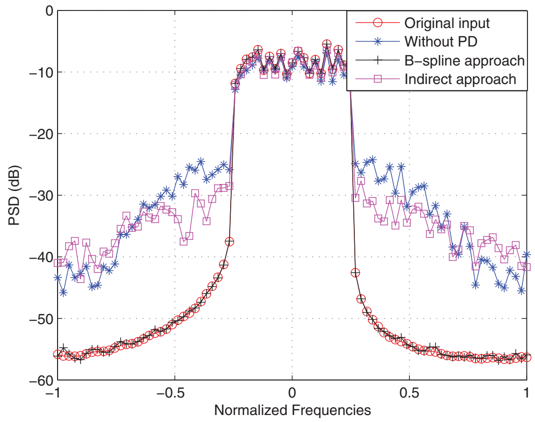


Fig. 3. Power spectrum density of the HPA output.

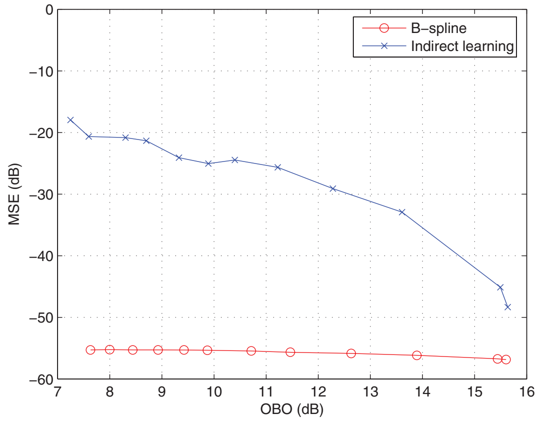


Fig. 4. MSE vs OBO performance.

10 and 6 respectively (we found that further increasing Q and K doesn't significantly improve the performance).

Figure 3 compares the power spectrum density (PSD) of the original input signal, the HPA output without any predistorter (PD), the HPA output with the indirect PD and the HPA output with the B-spline PD respectively. It is clearly shown that the PSD for the proposed approach is almost identical to that of the original input. On the other hand, the PSD for the indirect learning approach is mixed with that without a PD, indicating that the indirect learning PD approach is not effective in suppressing the spectrum regrowth of the HPA with the Wiener model.

Figure 4 compares the mean square error (MSE) performance against the output back-off (OBO) power. The MSE is defined as

$$\mathrm{MSE(dB)} = 10\log_{10}\mathrm{E}|d(t)-x(t)|^2, \tag{30}$$

where $x(t)$ and $d(t)$ are the original input and HPA output signals respectively. The OBO is obtained as

$$\mathrm{OBO(dB)} = 10\log_{10}\frac{P_{\mathrm{max}}}{P_{\mathrm{av}}}, \tag{31}$$

where P_{max} is the HPA saturation output power and P_{av} is the HPA average output power which is obtained as $P_{\mathrm{av}} = \mathrm{E}|d(t)|^2$ in the simulation. It is clearly shown in Figure 4 that, while the indirect learning approach can effectively achieve an MSE gain about 30dB, the proposed B-spline approach significantly improves the MSE performance by another $25 \sim 30$dB. This is a significant improvement, indicating that the B-spline approach can almost correct all of the non-linearity effect of the HPA with the Wiener model.

Figure 5 shows the bit error rate (BER) performance for the additive white Gaussian noise (AWGN) channel. For comparison, we also show the best possible BER performance for the ideal case that the HPA introduces no distortion at all (denoted as "Perfect linear HPA" in the figure). It is clearly shown in Figure 5 that the BER performance of the proposed B-spline approach is almost identical to that for the ideal case, and is significantly better than that of the indirect

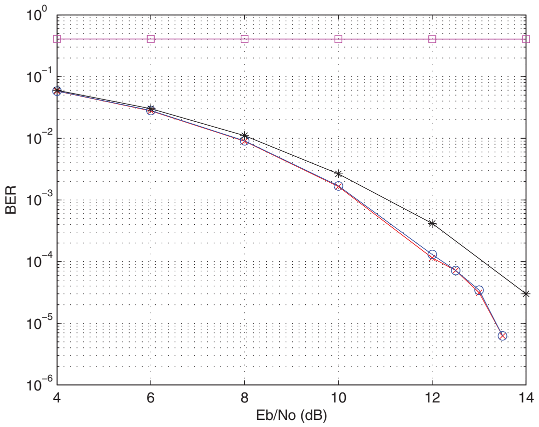


Fig. 5. BER performance for the AWGN channel.

approach. Note that only a single user AWGN channel is considered here as the details of more specific channels are beyond the scope of this paper.

VI. Conclusions

Based on a new B-spline based Wiener system modeling approach for HPA, a novel nonlinear digital baseband predistorter design has been introduced using direct learning. The complex valued nonlinear static function in the Wiener HPA model has been identified as the two real valued B-spline neural networks, for the amplitude distortion and the phase shift, respectively. The Gauss-Newton algorithm is applied for the parameter estimation, in which the De Boor recursion has been applied to calculate both the B-spline curve and the first order derivatives. The predistorter, acting as the inverse of Wiener system, is derived by calculating the inverse of the complex valued nonlinear static function based the identified Wiener models. The excellent performance and efficiency of the proposed approaches have been demonstrated using numerical simulations.

References

[1] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J.S. Kenney, J. Kim, and C. R. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," *IEEE Trans. Communications*, vol. 52, pp. 159–165, 2004.

[2] D. Zhou and V. E. DeBrunner, "Novel adaptive nonlinear predistorters based on the direct learning algorithm," *IEEE Trans. Signal Processing*, vol. 55, pp. 120–133, 2007.

[3] C. H. Lin, H. H. Chen, Y. Y. Yang, and J. J. Chen, "Dynamically optimum lookup table spacing for power amplifier predistortion linearisation," *IEEE Trans. Microwave Theory and Techniques*, vol. 54, pp. 2118–2127, 2006.

[4] B. Ai, Z. Y. Yang, C. P. Pan, S. G. Tang, and T. T. Zhang, "Ananysis on lut based predistorting method for HPA with memory," *IEEE Trans. Broadcasting*, vol. 53, pp. 127–131, 2007.

[5] R. Raich, H. Qian, and G. T. Zhou, "Orthogonal polynomials for power amplifier modeling and predistorter design," *IEEE Trans. Vehicular Technology*, vol. 53, pp. 1468–1479, 2004.

[6] D. R. Morgan, Z. X. Ma, J. Kim, M. G. Zierdt, and J. Pastalan, "A generalized memory polynomial model for digital predistortion of RF power amplifiers," *IEEE Trans. Signal Processing*, vol. 54, pp. 3852–3860, 2006.

[7] L. J. Xu, X. G. Wu, M. Zhang, G. Gang, and P. Zhang, "A stable recursive algorithm for memory polynomial predistorter," in *Proc. MILCOM*, Orlando, USA, 2007, vol. 5 pages.

[8] E. Abd-Elrady, L. Gan, and G. Kubin, "Distortion compensation of nonlinear systems based on indirect learning architecture," in *Proc 3rd Int. Symp. Communications, Control and Signal Processing*, ST. Julians, Malta, 2008, pp. 184–187.

[9] M. C. Chiu abd C. H. Zeng and M. C. Liu, "Predistorter based on frequency domain estimation for compensation of nonlinear distortion in OFDM systems," *IEEE Trans. on Vehicular Technology*, vol. 57, pp. 882–892, 2008.

[10] Y. H. Lim, Y. S. Cho, I. W. Cha, and D. H. Youn, "An adaptive nonlinear prefilter for compensation of distortion in nonlinear systems," *IEEE Trans. Signal Processing*, vol. 46, pp. 1726–1730, 2006.

[11] Y. Qian and T. Yao, "Structure for adaptive predistortion suitable for efficient adaptive algorithm application," *Electronic Letters*, vol. 38, pp. 1282–1283, 2002.

[12] S. Choi, E. R. Jeong, and Y. H. Lee, "Adaptive predistortion with direct learning based on piecewise linear approximation of amplifier nonlinearity," *IEEE J. Selected Topics in Signal Processing*, vol. 55, pp. 397–404, 2009.

[13] A. Hagenblad, L. Ljung, and A. Wills, "Maximum likelihood identification of Wiener models," *Automatica*, vol. 44, pp. 2697–2705, 2008.

[14] A. D. Kalafatis, L. Wang, and W. R. Cluett, "Identification of Wiener-type nonlinear systems in a noisy environment," *International Journal of Control*, vol. 66, pp. 923–941, 1997.

[15] I. W. Hunter and M. J. Korenberg, "The identification of nonlinear biological systems: Wiener and Hammerstein cascade models," *Biological Cybernetics*, vol. 55, pp. 135–144, 1986.

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

[17] C. J. Clark, G. Chrisikos, M.S. Muha, A. A. Moulthrop, and C. P. Silva, "Time-domain envelop measurement technique with application to wideband power amplifier modeling," *IEEE Trans. Microwave Theory and Techniques*, vol. 46, pp. 2531–2540, 1998.

[18] W. Greblicki, "Nonparametric identification of Wiener systems," *IEEE Transactions on Information Theory*, vol. 38, no. 5, pp. 1487–1493, 1992.

[19] D. Westwick, "Identifying MIMO Wiener systems using subspace model identification model identification methods," *Signal Processing*, vol. 52, pp. 235–258, 1996.

[20] A. D. Kalafatis, N. Arifinand L. Wang, and W. R. Cluett, "A new approach to the identification of pH processes on the Wiener model," *Chemical Engineering Science*, vol. 50, no. 23, pp. 3693–3701, 1995.

[21] Y. Zhu, "Distillution column identification for control using wiener model," in *Proc. the American Control Conference*, San Diego, CA, USA, 1999, pp. 3462–3466.

[22] de Boor, *A Practical Guide to Splines*, New York: Spring Verlag, 1978.

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