

A WIENER MODEL FOR MEMORY HIGH POWER AMPLIFIERS USING B-SPLINE FUNCTION APPROXIMATION

Xia Hong*, Yu Gong*, Sheng Chen**

*School of Systems engineering
University of Reading, UK

**School of Electronics and Computer Science
University of Southampton, UK

ABSTRACT

In this paper we introduce a new Wiener system modeling approach for memory high power amplifiers in communication systems using observational input/output data. By assuming that the nonlinearity in the Wiener model is mainly dependent on the input signal amplitude, the complex valued nonlinear static function is represented by two real valued B-spline curves, one for the amplitude distortion and another for the phase shift, respectively. The Gauss-Newton algorithm is applied for the parameter estimation, which incorporates the De Boor algorithm, including both the B-spline curve and the first order derivatives recursion. An illustrative example is utilized to demonstrate the efficacy of the proposed approach.

Index Terms— Wiener model, B-spline, Gauss-Newton algorithm, De Boor recursion

1. 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 modeling of high power amplifiers (HPA) is crucial in any linearisation techniques of broadband communication systems employing power-efficient nonlinear HPA transmitter [1, 2]. In order to tackle the problem, various HPA models and the associated predistorter design have been researched including the look up table (LUT) [3, 4, 5], the polynomials [1, 6, 7] and the Wiener model [8]. The Wiener model comprises a linear dynamical model followed by a nonlinear static functional transformation. The Wiener model is a reasonable model for any linear systems with a nonlinear measurement device and some industrial/biological systems [9, 10, 11], and the HPA in broadband communication transmitters [8], that is of particular interest in this work.

Various approaches have been developed in order to capture the *a priori* unknown nonlinearity in the Wiener system [12, 13, 14]. The B-spline curve has been widely used in computer graphics and computer aided geometric design

(CAGD) [15]. 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 [16]. The De Boor algorithm uses recurrence relations and is numerically stable [16]. In this work 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 curves, one for the amplitude distortion and another for the phase shift, respectively. It is shown that 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.

2. THE WIENER MODEL FOR MEMORY HPA

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 $\Psi(\bullet) : \mathcal{C} \rightarrow \mathcal{C}$ as the second subsystem. The system can be represented by

$$\begin{aligned} w(t) &= H(z)y(t) \\ &= y(t) + h_1y(t-1)\dots + h_ny(t-n) \end{aligned} \quad (1)$$

$$d(t) = \Psi(w(t)) + \xi(t) \quad (2)$$

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

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

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. $h_i = h_{i,R} + j \cdot h_{i,I}$, ($i = 1, \dots, n$) are complex valued coefficients of the linear filter. n is assumed known. Denote $\mathbf{h} = [h_1, \dots, h_n]^T \in \mathcal{C}^n$.

For the baseband HPA model, $\Psi(w(t))$ can be specified by a nonlinearity of the traveling wave tube (TWT) [8]. The input to the TWT nonlinearity can be expressed as

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

where $r(t) = \sqrt{w_R^2(t) + w_I^2(t)}$ and $\phi(t) = \arctan(\frac{w_I(t)}{w_R(t)})$, denoting the amplitude and phase of $w(t)$ respectively. The output of TWT, $\Psi(w(t))$, is distorted in both the amplitude and the phase, with the distortion dependent mainly on the input signal amplitude, i.e. $r(t)$. So $\Psi(w(t))$ can be expressed by [8]

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

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

$$r_\Psi(t) = \begin{cases} \frac{\alpha_1 r(t)}{1 + \alpha_2 r^2(t)}, & 0 \leq r(t) \leq r_{Sat} \\ \Psi_{max}, & r(t) > r_{Sat} \end{cases} \quad (6)$$

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

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 curves respectively as described in the following.

3. THE SYSTEM IDENTIFICATION ALGORITHM

3.1. 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, $\{R_1, R_2, \dots, R_{M+k}\}$. 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

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

Given these predetermined knots, a set of M B-spline basis functions can be formed by using the De Boor recursion [16], given by

$$\mathcal{B}_l^{(0)}(r) = \begin{cases} 1 & \text{if } R_l \leq r < R_{l+1} \\ 0 & \text{otherwise} \end{cases} \quad l = 1, \dots, (M+k) \quad (9)$$

$$\begin{aligned} \mathcal{B}_l^{(i)}(r) &= \frac{r - R_l}{R_{i+l} - R_l} \mathcal{B}_l^{(i-1)}(r) + \frac{R_{i+l+1} - r}{R_{i+l+1} - R_{l+1}} \mathcal{B}_{l+1}^{(i-1)}(r), \\ &\text{for } l = 1, \dots, (M+k-i), \quad i = 1, \dots, k \end{aligned} \quad (10)$$

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

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

We model $\Psi(\bullet)$ as two univariate B-spline curves [17], one for the amplitude and another for the phase shift, in the form of

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

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

and their derivatives are in the form of

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

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

where ω_l 's and ϑ_l 's are weights to be determined. Denote $\boldsymbol{\omega} = [\omega_1, \dots, \omega_M]^T \in \mathfrak{R}^M$ and $\boldsymbol{\vartheta} = [\vartheta_1, \dots, \vartheta_M]^T \in \mathfrak{R}^M$. 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 $O(k^2)$.

3.2. 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

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

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} , ω and ϑ . This could be achieved by minimizing

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

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

$$\begin{cases} w_R(t) = y_R(t) + \sum_{i=1}^n (h_{i,R} y_R(t-i) - h_{i,I} y_I(t-i)) \\ w_I(t) = y_I(t) + \sum_{i=1}^n (h_{i,I} y_R(t-i) + h_{i,R} y_I(t-i)) \end{cases} \quad (17)$$

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

$$\begin{cases} \frac{\partial w_R(t)}{\partial h_{i,R}} = y_R(t-i) \\ \frac{\partial w_R(t)}{\partial h_{i,I}} = -y_I(t-i) \\ \frac{\partial w_I(t)}{\partial h_{i,R}} = y_I(t-i) \\ \frac{\partial w_I(t)}{\partial h_{i,I}} = y_R(t-i) \end{cases} \quad (18)$$

Thus it can be shown that

$$\begin{cases} \frac{\partial r(t)}{\partial h_{i,R}} = \frac{1}{r(t)} [w_R(t) y_R(t-i) + w_I(t) y_I(t-i)] \\ \frac{\partial r(t)}{\partial h_{i,I}} = \frac{1}{r(t)} [w_I(t) y_R(t-i) - w_R(t) y_I(t-i)] \end{cases} \quad (19)$$

and

$$\begin{cases} \frac{\partial \phi(t)}{\partial h_{i,R}} = \frac{1}{r^2(t)} [w_R(t) y_I(t-i) - w_I(t) y_R(t-i)] \\ \frac{\partial \phi(t)}{\partial h_{i,I}} = \frac{1}{r^2(t)} [w_R(t) y_R(t-i) + w_I(t) y_I(t-i)] \end{cases} \quad (20)$$

We denote an iteration step variable by a superscript (τ) , and apply the Gauss Newton algorithm as follows.

The Gauss Newton algorithm combined with the De Boor algorithm

With an initial $\theta^{(0)}$, the iteration formula is given by

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

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

$1, \dots, K$, and $t = p$.

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

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

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

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 $\theta^{(\tau)}$ converges, or by predetermining a sufficiently large number of iterations.

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 $\theta^{(0)}$ 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.

3.3. 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 con-

dition of being external to the regions for $r(t)$ at each end. In this work, the parameter $\theta^{(0)}$ is initialized as follows.

1. Initialize $h_i^{(0)} = 0$, i.e. $h_{i,R}^{(0)} = 0$, $h_{i,I}^{(0)} = 0$, for $i = 1, \dots, n$.
2. Generate a sequence $r^{(0)}(t) = \sqrt{y_R^2(t) + y_I^2(t)}$, for $t = 1, \dots, K$.
3. Generate a sequence $r_d^{(0)}(t) = \sqrt{d_R^2(t) + d_I^2(t)}$, for $t = 1, \dots, K$. Denote $\mathbf{r}_d = [r_d^{(0)}(1), \dots, r_d^{(0)}(K)]^T$.
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 $t = 1, \dots, K$. Denote $\phi_\Psi = [\phi_\Psi^{(0)}(1), \dots, \phi_\Psi^{(0)}(K)]^T$.
5. Form a regression matrix $\mathbf{B} = \{\mathcal{B}_l^{(k)}(r^{(0)}(t))\} \in \mathfrak{R}^{K \times M}$.
6. Compute the least square estimate $\omega^{(0)} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{r}_d$, and $\vartheta^{(0)} = (\mathbf{B}^T \mathbf{B})^{-1} \mathbf{B}^T \phi_\Psi$.
7. Finally set $\theta^{(0)} = \underbrace{[0, \dots, 0]}_{2n}, [\omega^{(0)}]^T, [\vartheta^{(0)}]^T$.

4. A MODELING EXAMPLE

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 $\alpha_1 = 3$, $\alpha_2 = 1.2$, $\beta_1 = \pi/12$, $\beta_2 = 0.25$. $y(t)$ was uniformly distributed complex random variable with $y_R(t) \in [-1.5, 1.5]$ and $y_I(t) \in [-1.5, 1.5]$. The variances of the additive noise to the system output are set $\sigma^2 = 0.02^2$. The polynomial degree of the B-spline basis function was set as three (i.e. $k = 4$, piecewise cubic). The following predetermined knot sequences. The 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 modelling results shown in Table 1 and Figure 1 (a) and (b). It is shown that the proposed approach has excellent approximation results.

5. CONCLUSIONS

A new Wiener system modeling approach has been introduced in order to model the memory high power amplifiers in communication systems. We model the complex valued nonlinear static function using two real valued B-spline curves,

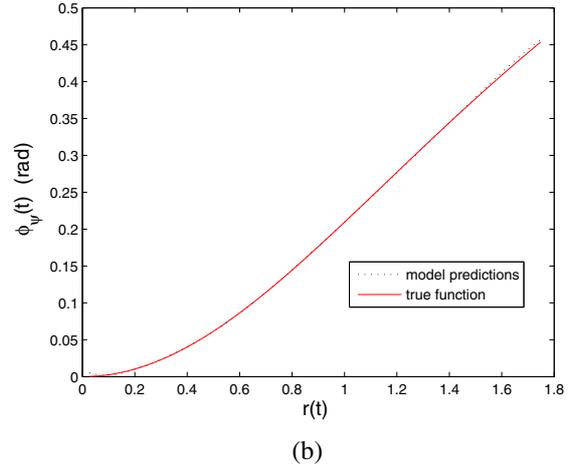
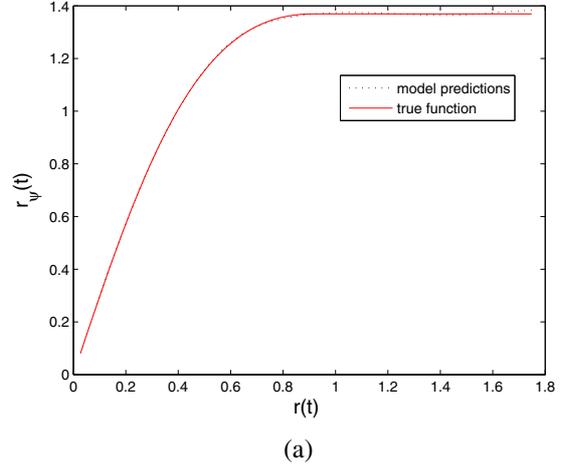


Fig. 1. The TWT nonlinearity modeling results; (a) Amplitude distortion with respect to the amplitude of the input; and (b) Phase shift with respect to the amplitude of the input.

Table 1. Results of linear subsystem parameter estimation.

	True parameters	Initial estimates	Final estimates
h_1	0.7692	0	0.7691 $-j6 \times 10^{-7}$
h_2	0.1538	0	0.1537 $+j8 \times 10^{-6}$
h_3	0.0769	0	0.0769 $-j2 \times 10^{-5}$

one for the amplitude distortion and another for the phase shift, respectively. The parameter is estimated using the Gauss-Newton algorithm. The De Boor algorithm, including both the B-spline curve and the first order derivatives recursion, is utilized in the algorithm. Future work will be focused on predistorter design based on the proposed model.

6. 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, "An analysis on lut based predistorting method for HPA with memory," *IEEE Trans. Broadcasting*, vol. 53, pp. 127–131, 2007.
- [5] P. Jardin and G. Baudoin, "Filter lookup table method for power amplifier linearization," *IEEE Trans. Vehicular Technology*, vol. 56, pp. 1076–1087, 2007.
- [6] 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.
- [7] 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.
- [8] 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.
- [9] A. Hagenblad, L. Ljung, and A. Wills, "Maximum likelihood identification of Wiener models," *Automatica*, vol. 44, pp. 2697–2705, 2008.
- [10] 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.
- [11] 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.
- [12] W. Greblicki, "Nonparametric identification of Wiener systems," *IEEE Transactions on Information Theory*, vol. 38, no. 5, pp. 1487–1493, 1992.
- [13] E. W. Bai, "An optimal two-stage identification algorithm for Hammerstein-wiener nonlinear systems," *Automatica*, vol. 34, pp. 333–338, 1998.
- [14] A. D. Kalafatis, N. Arifin and 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.
- [15] G. Farin, *Curves and Surfaces for Computer-aided Geometric Design: a Practical Guide*, Academic Press, Boston, 1994.
- [16] de Boor, *A Practical Guide to Splines*, New York: Springer Verlag, 1978.
- [17] C. J. Harris, X. Hong, and Q. Gan, *Adaptive Modelling, Estimation and Fusion from Data: A Neurofuzzy Approach*, Springer-Verlag, 2002.