Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 379249, 10 pages
doi:10.1155/2010/379249

Research Article

Efficient Lookup Table-Based Adaptive Baseband Predistortion Architecture for Memoryless Nonlinearity

Seydou N. Ba,[1] Khurram Waheed,[2] and G. Tong Zhou[1]

[1] *School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0250, USA*
[2] *RF-CMOS Radio Design Group of the Wireless Terminals Business Unit, Texas Instruments, Inc., Dallas, TX 75243, USA*

Correspondence should be addressed to Seydou N. Ba, seydou@ieee.org

Received 24 November 2009; Revised 23 March 2010; Accepted 14 May 2010

Academic Editor: Markus Rupp

Digital predistortion is an effective means to compensate for the nonlinear effects of a memoryless system. In case of a cellular transmitter, a digital baseband predistorter can mitigate the undesirable nonlinear effects along the signal chain, particularly the nonlinear impairments in the radiofrequency (RF) amplifiers. To be practically feasible, the implementation complexity of the predistorter must be minimized so that it becomes a cost-effective solution for the resource-limited wireless handset. This paper proposes optimizations that facilitate the design of a low-cost high-performance adaptive digital baseband predistorter for memoryless systems. A comparative performance analysis of the amplitude and power lookup table (LUT) indexing schemes is presented. An optimized low-complexity amplitude approximation and its hardware synthesis results are also studied. An efficient LUT predistorter training algorithm that combines the fast convergence speed of the normalized least mean squares (NLMSs) with a small hardware footprint is proposed. Results of fixed-point simulations based on the measured nonlinear characteristics of an RF amplifier are presented.

1. Introduction

High-efficiency RF amplifiers have nonlinear amplitude and phase transfer characteristics, which distort the transmitted signals, causing undesired out-of-band spectral regrowth and an increase in error vector magnitude (EVM) and bit error rate (BER). Digital baseband predistortion is an effective means to reconcile the conflicting requirements of linearity and power efficiency. For resource-limited low-cost handsets, the implementation complexity of the predistorter must be minimized. This paper proposes optimizations that facilitate the design of a cost-effective and high-performance adaptive digital baseband predistorter, while minimizing expensive factory calibration requirements. These attributes render this work highly desirable to meet the stringent linearity requirements of the modern third and fourth generation (3G/4G) wireless systems, which employ complex amplitude and phase domain modulations to achieve superior spectral efficiency [1].

While 2.5G EDGE and 3G WCDMA voice waveforms used simpler modulation schemes that exhibited less than 3.5 dB of peak-to-average power ratio (PAPR), advanced WCDMA (or HSPA) waveforms exhibit PAPRs in excess of 6 dB and modern 4G (LTE, WiMax) use more complex signal constellations resulting in PAPRs of up to 12 dB [1]. Such a high PAPR mandates higher linearity requirements from the RF physical layer, which is in sharp contrast to the stronger demand for increased power efficiency and maximization of the handset battery life. These conflicting requirements can be tamed by resorting to the use of RF front-end amplifiers in their most power-efficient regime, while using signal predistortion schemes to achieve the desired linearity.

The nonlinear gain and phase distortions of RF amplifiers are a strong function of the envelope fluctuations in an RF signal [2, 3]. Consequently, most digital baseband predistorters are implemented as a function of the amplitude of the baseband input. In the case of the complex-gain lookup table (LUT) predistorter [4, 5], the most significant bits (MSBs) of the signal magnitude can be directly used to address the physical memory containing the LUT entries. For example, the first seven MSBs can be used to address an LUT with 128 entries [6]. The precise amplitude computation requires a square-root operation, which is not directly amenable to efficient hardware implementation, especially at very high processing rates. A square-root approximation proposed in [7] has a performance close to the ideal amplitude calculation. But in addition to the squared magnitude computation, the square-root approximation requires additional LUTs and a linear interpolation calculation. Other practical digital baseband predistorters [4] have been implemented as a function of the instantaneous envelope power $I^2 + Q^2$, where I is the inphase, Q is the quadrature component of the complex baseband signal. The resulting, but often unintended effect, is a concentration of the LUT entries around the higher amplitude region [7, 8]. This power indexing scheme is suitable for class-A and mild class-AB amplifiers since their characteristics are mostly linear until close to saturation. However, this is not well suited to amplifiers with higher power efficiency, such as deep class-AB, class-B, C, and E [9], which exhibit significant nonlinear amplitude and phase distortions across the entire amplitude range. A comparative performance analysis of the amplitude- and power-indexing schemes will be presented in this paper. A suitable low-complexity amplitude approximation for digital baseband predistorters is then applied. The proposed amplitude approximation has lower complexity than the squared magnitude computation and a performance that is close to the ideal amplitude-indexed LUT predistorter.

Furthermore, the nonlinear characteristics of power amplifiers can display significant variations when the operating temperature fluctuates and as the device ages. To maintain effectiveness of the predistorter and minimize residual distortions as well as calibration requirements, an adaptive predistorter [2, 10] must be used. This problem is further exacerbated by the high PAPR of the modern 3G/4G modulation waveforms. In this paper, an efficient least mean squares (LMS)-based [11] adaptation technique for LUT predistorters is presented as well as its optimization for low complexity hardware implementation.

Section 2 presents a comparative performance analysis between amplitude and power LUT indexing schemes and studies the design and implementation of a suitable amplitude approximation for digital baseband predistorters. Section 3 presents a low-complexity training approach for LUT-based complex-gain predistorters.

2. Performance of Amplitude and Power LUT Indexing

The indexing of a predistorter LUT with the squared signal magnitude is an attractive approach because of the relative ease of computation of $I^2 + Q^2$. But it is reported in [7] that the magnitude indexing generally results in significantly better performance for a given LUT size. The performance gap is further exacerbated when the source signal is scaled for the purpose of power control. An LUT-based square-root approximation proposed in [7] has a performance that is close to the ideal amplitude calculation. In this section, we show that an accurate magnitude approximation for digital baseband predistorters, with lower hardware footprint, can be obtained directly from the inphase and quadrature components of the input signal.

Simple amplitude approximation techniques have been used for radar detection applications [12–15]. Most of the methods presented result in relatively coarse approximations, even though their precision is within the tolerance of the target applications. But since the digital baseband predistorter is located in the direct transmit path, such large amplitude approximation errors would severely limit the performance of the predistorter, resulting in both residual EVM degradation and spectral distortions.

The general approach to linear amplitude approximation is explained in [13]. It consists of rotating the complex input signal $X = I + jQ$ such that its phase lies in $[0, \pi/4]$, then computing a linear combination of the real and imaginary parts of the rotated signal $Y = I_r + jQ_r$. The rotated signal Y is given by

$$I_r = \max(|I|, |Q|), \qquad Q_r = \min(|I|, |Q|). \tag{1}$$

It can be easily observed that the magnitude of the rotated vector Y is equal to the magnitude of the initial vector X:

$$\begin{aligned} R = |Y| &= \sqrt{[\max(|I|, |Q|)]^2 + [\min(|I|, |Q|)]^2} \\ &= \sqrt{|I|^2 + |Q|^2} = |X|. \end{aligned} \tag{2}$$

The approximated amplitude is then obtained by evaluating a linear combination of the real and imaginary parts of Y:

$$\widehat{R} = aI_r + bQ_r \quad \text{with } a, b \geq 0. \tag{3}$$

In [13], the approximation accuracy is improved by further dividing the angular interval $[0, \pi/4]$ into two intervals, and using two different sets of coefficients (a_k, b_k), $k \in \{1, 2\}$ that are optimized for their corresponding angular intervals.

This approach can be further extended to arbitrarily improve the approximation accuracy by increasing the number of angular intervals N. If the complex input falls in the kth angular interval, the amplitude approximation is given by

$$\widehat{R}_k = a_k I_r + b_k Q_r, \quad \text{for } \theta_{k-1} \leq \theta < \theta_k, \tag{4}$$

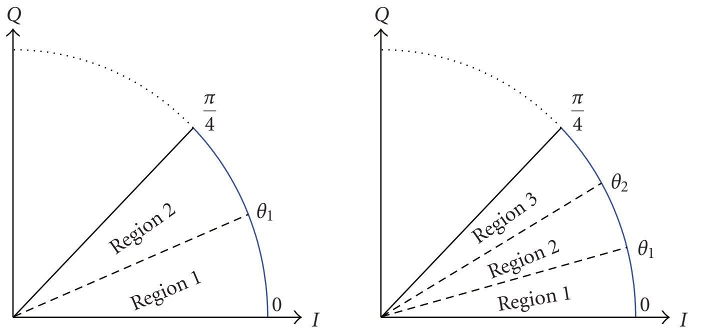
where $\theta = \arctan(Q_r/I_r)$, θ_k and θ_{k-1} are the threshold angles delimiting the angular intervals, with $\theta_0 = 0$ and $\theta_N = \pi/4$. Figures 1(a) and 1(b) illustrate the use of two and three equal angular intervals, respectively.

The amplitude error in the kth angular interval can be computed as

$$\begin{aligned} R - \widehat{R}_k &= R - (a_k I_r + b_k Q_r) \\ &= R(1 - a_k \cos\theta - b_k \sin\theta). \end{aligned} \tag{5}$$

The relative amplitude error in the kth interval ε_k is given by

$$\varepsilon_k = \frac{R - \widehat{R}_k}{R} = 1 - a_k \cos\theta - b_k \sin\theta. \tag{6}$$



(a)

(b)

Figure 1: Linear amplitude approximations. (a) Two angular intervals. (b) Three angular intervals.

The amplitude error (6) is a function of the input angle. The coefficients (a_k, b_k) must be chosen to minimize a given error metric for each angular interval delimited by the angles θ_{k-1} and θ_k. Assuming that the input angle θ is uniformly distributed, we can obtain a closed-form solution for the coefficients (a_k, b_k) that minimizes the mean square of the relative amplitude error ε_k. The mean squared error $\mathcal{J}_k$ can be evaluated as follows:

$$\mathcal{J}_k = E\left[\varepsilon_k^2\right] = p_0 \int_{\theta_{k-1}}^{\theta_k} \varepsilon_k^2 d\theta, \tag{7}$$

where $p_0 = 1/(\theta_k - \theta_{k-1})$. The optimal coefficients are obtained by setting the partial derivatives of $\mathcal{J}_k$ with respect to the coefficients a_k and b_k to zero. Taking the partial derivative of the mean squared error $\mathcal{J}_k$ with respect to the coefficient a_k gives

$$\begin{aligned}
\frac{\partial \mathcal{J}_k}{\partial a_k} &= p_0 \int_{\theta_{k-1}}^{\theta_k} \frac{\partial \varepsilon_k^2}{\partial a_k} d\theta \\
&= p_0 \int_{\theta_{k-1}}^{\theta_k} 2\varepsilon_k \frac{\partial \varepsilon_k}{\partial a_k} d\theta \\
&= 2p_0 \int_{\theta_{k-1}}^{\theta_k} a_k \cos^2\theta + b_k \cos\theta \sin\theta - \cos\theta \, d\theta \\
&= p_0 \int_{\theta_{k-1}}^{\theta_k} a_k(1 + \cos 2\theta) + b_k \sin 2\theta - 2\cos\theta \, d\theta \\
&= \frac{p_0}{2}\left[a_k(2\Delta\theta_k + \alpha_k) + b_k\beta_k - 4c_k\right],
\end{aligned} \tag{8}$$

where

$$\begin{aligned}
\alpha_k &= \sin 2\theta_k - \sin 2\theta_{k-1}, & c_k &= \sin\theta_k - \sin\theta_{k-1}, \\
\beta_k &= \cos 2\theta_{k-1} - \cos 2\theta_k, & \Delta\theta_k &= \theta_k - \theta_{k-1}.
\end{aligned} \tag{9}$$

Similarly, taking the partial derivative with respect to b_k gives

$$\frac{\partial \mathcal{J}_k}{\partial b_k} = \frac{p_0}{2}\left[b_k(2\Delta\theta_k - \alpha_k) + a_k\beta_k - 4d_k\right], \tag{10}$$

with

$$d_k = \cos\theta_{k-1} - \cos\theta_k. \tag{11}$$

Setting the partial derivatives to zero yields

$$\begin{bmatrix} 2\Delta\theta_k + \alpha_k & \beta_k \\ \beta_k & 2\Delta\theta_k - \alpha_k \end{bmatrix} \begin{bmatrix} a_k \\ b_k \end{bmatrix} = 4 \begin{bmatrix} c_k \\ d_k \end{bmatrix}. \tag{12}$$

It should be noted that since $0 \le \theta_k \le \pi/4$ and $\theta_k > \theta_{k-1}$, the coefficients α_k, β_k, c_k and d_k are all strictly positive. The optimal coefficients for the kth angular interval are obtained by solving the above system of linear equations (12),

$$\begin{bmatrix} a_k \\ b_k \end{bmatrix} = \frac{2}{h_k} \begin{bmatrix} (2\Delta\theta_k + \alpha_k)c_k - \beta_k \, d_k \\ (2\Delta\theta_k - \alpha_k)d_k - \beta_k \, c_k \end{bmatrix}, \tag{13}$$

with $h_k = 2\Delta\theta_k^2 + \cos(2\Delta\theta_k) - 1$. For any angular interval delimited by the angles θ_{k-1} and θ_k, the relatively simple closed-form solution (13) can be evaluated to find the optimal coefficients (a_k, b_k) in the mean squared error sense. Figure 2 shows the mean squared and peak errors of ε as the number of angular intervals is increased from $N = 1$ to $N = 8$.

These results show that the use of three angular intervals is sufficient to decrease the mean square of the relative amplitude error below −50 dB. This ensures that there is negligible transmit EVM and ACLR contribution due to the predistorter implementation. As shown by these results, an arbitrary amplitude approximation accuracy can be achieved by selecting a large enough number of angular intervals. But a larger number of angular intervals will result in a more complex decision process and the approximation is useful only if it is amenable to efficient implementation. It should be noted that the optimal coefficients obtained here are based on the assumption that the phase of the input signal is uniformly distributed. This assumption applies very well to most signal modulations. In the special case of a skewed phase probability density, the true optimal coefficients can be better approached using unequal angular intervals.

For practical implementation, the approximation based on three angular intervals is chosen. The angular intervals are equally spaced. The threshold angles are $\theta_1 = \pi/12$, and $\theta_2 = \pi/6$. For each input sample $(I_r + jQ_r)$, the corresponding

Table 1: Amplitude approximation over three angular intervals: amplitude approximation coefficients and resulting relative amplitude errors for floating-point and fixed-point implementations.

Quantities	Floating-point	Fixed-point
[a$_1$ a$_2$ a$_3$]	[0.994 0.927 0.796]	[1 60/64 51/64]
[b$_1$ b$_2$ b$_3$]	[0.131 0.384 0.610]	[6/64 23/64 39/64]
$\tan(\theta_1)$	0.268	1/4
$\tan(\theta_2)$	0.577	9/16
$\varepsilon_{\text{peak}}$ (%)	0.572	0.712
$\varepsilon_{\text{mean}}$ (%)	0.001	0.082
ε_{rms} (%)	0.256	0.306

$\varepsilon_{\text{peak}} = \max|\varepsilon_m|$, $\varepsilon_{\text{mean}} = (1/M)\sum\varepsilon_m$, and $\varepsilon_{\text{rms}} = (1/M)\sqrt{\sum\varepsilon_m^2}$.
ε_m is the relative amplitude error for the mth input.
$M = 1000$ is the total number of test samples.

Table 2: EVM and ACLR performances of an LUT predistorter with amplitude versus power indexing; the input is WCDMA.

DPD Status	Indexing Scheme	EVM (dB)	ACLR1 (dBc/Hz)	ACLR2 (dBc/Hz)
DPD OFF	—	−21.71	−32.18	−49.81
DPD ON	Power	−44.71	−53.60	−53.86
DPD ON	Amplitude approx	−61.57	−67.86	−69.95
DPD ON	Amplitude ideal	−63.60	−68.99	−71.68

angular interval is determined by comparing Q_r to $I_r \tan(\theta_k)$ since $\tan(\cdot)$ is a monotonic function in the interval $[0, \pi/4]$

$$\theta < \theta_k \Longrightarrow \tan(\theta) < \tan(\theta_k) \Longrightarrow Q_r < I_r \tan(\theta_k). \tag{14}$$

For efficient hardware implementation, we select $\tan(\theta_1) = 1/4$ and $\tan(\theta_2) = 9/16$. The coefficients obtained from (13) are quantized to six bits of resolution. For best results, the quantized coefficients a_k are used to generate new suboptimal coefficients b_k, which are in turn quantized. This two-step process results in a slightly better performance than the direct quantization of the coefficients a_k and b_k. The coefficients and error characteristics of the floating point and quantized amplitude approximations are summarized in Table 1. We observe that the fixed-point approximation has the advantage of being more practical with a smaller hardware footprint, while achieving a performance that is very close to that of the floating-point approximation. Note that the difference in ε_{rms} for the floating- versus fixed-point implementation is caused by the round-off errors implemented in the fixed-point hardware.

The performance of the fixed-point amplitude approximation was simulated within a SIMULINK model of a complete transmitter including predistortion. The amplifier model is based on the extracted AM-AM and AM-PM characteristics of a class-E amplifier [16, 17]. The real and imaginary parts of the class-E amplitude-dependent complex-gain $g(\cdot)$ are shown in Figure 3 or three different temperature settings. The nominal curve at 25°C is used for the purpose of the present experiment.

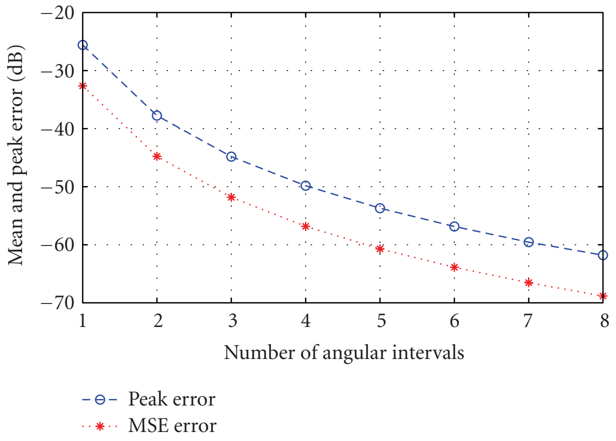


Figure 2: Mean squared and peak error (ε_k) as a function of the number of angular intervals N.

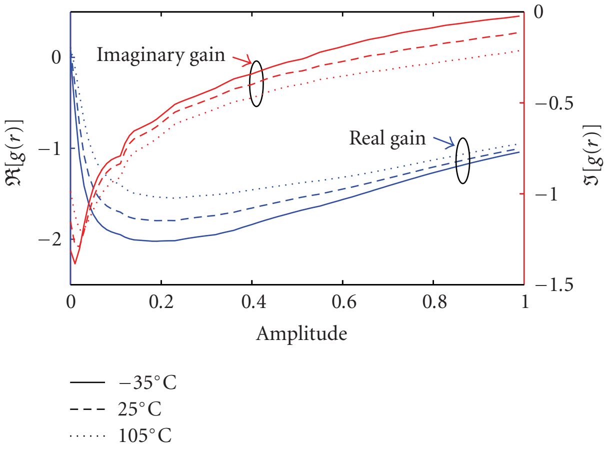


Figure 3: Real and imaginary parts of a class-E amplifier nonlinearity expressed as a complex-gain, over different temperature settings.

A linearly interpolated complex-gain LUT with 64 entries was used to predistort the class-E amplifier. A WCDMA rel. 8 HSUPA-compliant 64QAM signal, with >6.5 dB composite PAPR is used as input. The input signal (I/Q) resolution was set to 13 bits and a 3 dB backoff was selected. The EVM and adjacent channel leakage ratios (ACLRs) at 5 MHz offset (ACLR1) and 10 MHz offset (ACLR2) are shown in Table 2. The ACLR1 and ACLR2 are measured in dBc across a 5 MHz channel bandwidth. The EVM resulting from the use of the amplitude indexing is nearly 17 dB lower than that of the power indexing, and only 2 dB higher than that of the ideal amplitude indexing. The ACLR1 and ACLR2 measurements show more than 14 dB improvement when using the amplitude approximation instead of the power indexing. Figure 4 shows the WCDMA power spectral density (PSD) resulting from the above experiment. It is observed that the spectral regrowth is effectively reduced by the predistorters. The higher spectral floor resulting from the power indexing scheme indicates its relatively strong sensitivity to LUT quantization errors.

Table 3: Nand2-equivalent gate count for power index computation and amplitude approximation.

I/Q resolution (bits)	Gate count	
	Power indexing	Amplitude approx
8	1135	1248
10	1884	1629
12	2778	1970
14	3853	2324

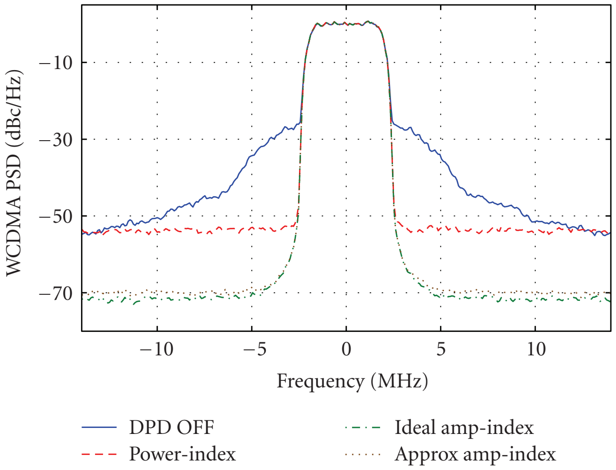


Figure 4: PSD performances of an LUT predistorter using ideal amplitude indexing, amplitude indexing with approximation, or power indexing. Input signal is WCDMA.

The fixed-point coefficients and angular thresholds are chosen to minimize the hardware implementation complexity while maintaining an approximation error close to the optimum. The diagram of Figure 5 illustrates a possible implementation.

This design requires two conditional two's complement operations to implement the abs(·) function, three comparators, and four two-to-one multiplexers. The coefficients were chosen to minimize the complexity of the scaling operations. To achieve a fair comparison, the implementation complexity of the amplitude approximation must be compared to that of the instantaneous power computation ($I^2 + Q^2$). Both options were implemented in VHDL and synthesized with the Synopsys Design Compiler. The resulting nand2-equivalent gate count is obtained for different resolutions of the inphase/quadrature components (I/Q). The synthesis results are summarized in Table 3.

It is clear from these results that the amplitude approximation design results in lower gate count for the input signal resolutions of interest (>10 bits). The gap between the amplitude-indexing and power-indexing schemes increases rapidly as the resolution is increased from 8 to 14 bits. For input resolutions lower than 8 bits, the power computation results in a slightly lower gate count. But at such low resolutions, the performance is primarily limited by the I/Q signal quantization error. In this case, the resolution of the (a_k, b$_k$) coefficients can be reduced down to 5 or 4 bits to further reduce the gate count of the amplitude approximation block. Typically, a baseband signal resolution of more than 10 bits is required to meet the close-in spectrum and waveform quality specifications over the entire power control dynamic range as per the standard's requirements. Therefore, the proposed amplitude approximation design has a clear advantage over the power indexing, both in terms of total design area and performance.

3. Adaptation of Complex-Gain LUT Predistorters

In [4], Cavers proposed the *secant* update for fast adaptation of complex-gain LUT predistorters. But its high computational complexity makes it unsuitable for hardware implementation.

The indirect learning architecture [18] is illustrated in Figure 6. A replica of the feedforward predistorter is trained in the feedback path as the postinverse of the amplifier nonlinearity. The updated LUT is periodically copied to the feedforward predistorter. This configuration has the advantage of decoupling the transmit path from the update branch. The transmitted signal is therefore isolated from any impulse noise in the feedback path at the cost of replicating the predistorter.

The LUT is an array of L complex-gain entries $F^{[n]}$ corresponding to the input amplitudes r_n. If the LUT is not interpolated, the nth LUT entry is selected for all feedback signals y_k in the interval defined by

$$\frac{r_n + r_{n-1}}{2} \leq |y_k| < \frac{r_n + r_{n+1}}{2}. \tag{15}$$

For every signal sample y_k in this interval, an error signal e_k is generated,

$$e_k = z_k - F^{[n]} y_k. \tag{16}$$

The nth entry $F^{[n]}$ can be updated using the LMS algorithm as follows:

$$F_{k+1}^{[n]} = F_k^{[n]} - \mu \frac{\partial e_k^* e_k}{\partial F^{[n]}}. \tag{17}$$

It should be noted that $|e_k|^2$ is not a holomorphic function and therefore does not have a complex derivative. For the purpose of the steepest-descent algorithm, the complex gradient with respect to the complex gain $F^{[n]}$ can be defined as the combination of the partial derivatives with respect to the real and imaginary parts of $F^{[n]}$ [19]:

$$\frac{\partial e_k^* e_k}{\partial F^{[n]}} = \frac{1}{2}\left(\frac{\partial e_k^* e_k}{\partial \Re\{F^{[n]}\}} + j\frac{\partial e_k^* e_k}{\partial \Im\{F^{[n]}\}}\right), \tag{18}$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$, respectively, designate the real and imaginary parts of the argument. Substituting (18) into (17) and carrying out the partial derivatives yields

$$F_{k+1}^{[n]} = F_k^{[n]} + \mu y_k^* e_k. \tag{19}$$

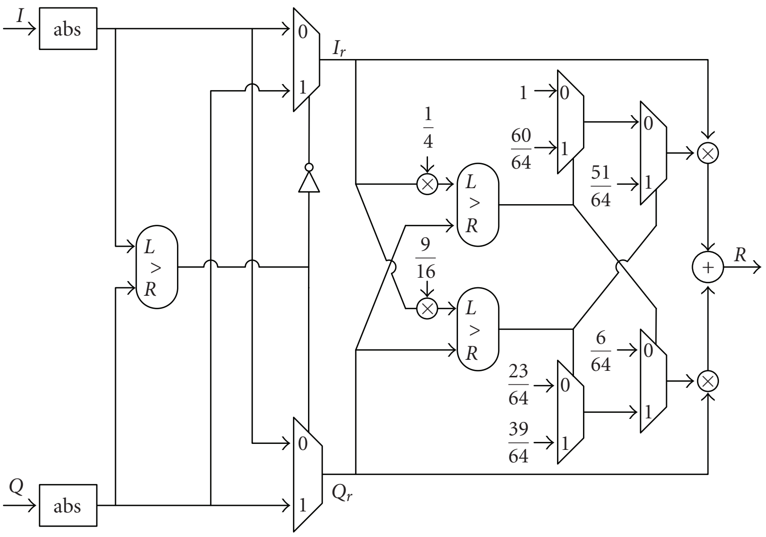


Figure 5: Implementation of the amplitude approximation with three angular intervals.

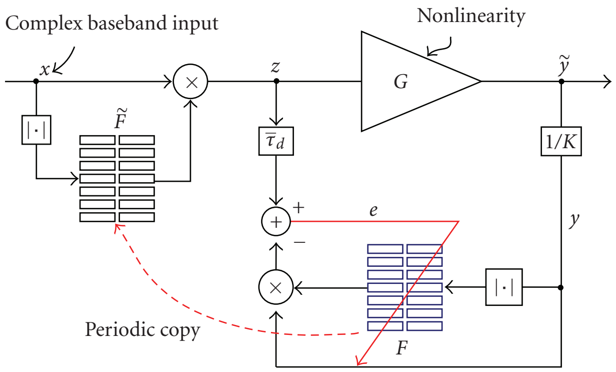


Figure 6: Adaptation of complex-gain LUT predistorters using the indirect learning architecture.

The gradient definition in (18) is equivalent to separately deriving the LMS algorithm for the real and imaginary parts of the complex-gain predistorter, respectively [20]. Considering one single interval at a time allows to simplify the problem by reducing it to finding an approximate inverse of the average amplifier complex gain within the considered interval. For each incoming feedback sample, only the corresponding entry that is addressed by its magnitude is updated. This process is similar to the partial update LMS [21, 22]. The update operation requires two complex multiplies (one to compute the error e_k and one to evaluate the gradient), two additions and the scaling by μ, which can be simplified if it is restricted to powers of two. The update system is stable provided that $0 < \mu < 2/\lambda_n^2$ [23], with λ_n^2 being equal to $E[|y_k|^2]$ for all y_k falling in the nth interval. If the LUT size is large, the samples y_k can be assumed to have a uniform distribution across the interval. In this case, the expectation can be approximated by the square of the average magnitude, which is the point located at the center of the interval: $\lambda_n^2 \approx |y_n|^2$.

If the regular LMS update equation (19) is used, the convergence speed will vary across the table entries. The upper entries will converge significantly faster than the lower entries. To avoid this issue, the normalized LMS (NLMS) algorithm [24, 25] can be used,

$$F_{k+1}^{[n]} = F_k^{[n]} + \frac{\mu}{|y_k|^2} y_k^* e_k. \qquad (20)$$

The NLMS update of (20) results in faster and uniform convergence of the entries across the LUT. But its direct implementation has two limitations.

(i) For very low values of $|y_k|$ the system becomes susceptible to noise in the feedback path, with a potential to drive the update system into instability.

(ii) The scaling by the magnitude is an expensive operation that is not directly amenable to efficient hardware implementation.

An approximation of the NLMS similar to the clipped LMS algorithm [26–28] is proposed. This approach, termed low-complexity normalized LMS (LCNLMS), is suitable for efficient hardware implementation and maintains the fast convergence of the NLMS. First, the update equation of (20) can be conveniently reformulated as follows:

$$\begin{aligned} F_{k+1}^{[n]} &= F_k^{[n]} + \frac{\mu}{|y_k|} \frac{y_k^*}{|y_k|} e_k \\ &= F_k^{[n]} + \mu_k e^{j\phi_k} e_k, \end{aligned} \qquad (21)$$

where $\phi_k = \angle y_k^*$ is the complex argument of y_k^* and $\mu_k = \mu/|y_k|$. It is clear from this incremental update that the NLMS is equivalent to using a variable update coefficient that is inversely proportional to the input amplitude $|y_k|$ and replacing the complex multiply with a rotation of the error by ϕ_k. The computational complexity of the rotation operation can be greatly simplified by quantizing the angle ϕ_k. To do

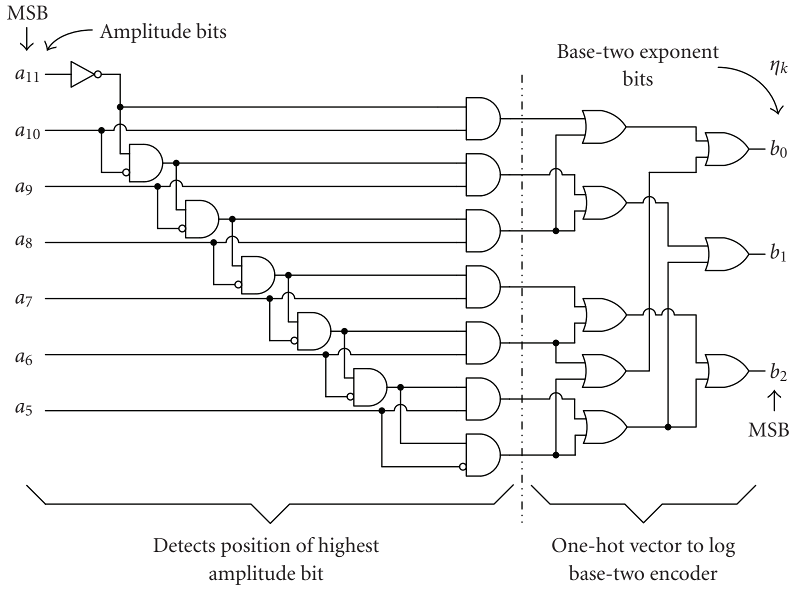


Figure 7: Circuit that generates the base-two exponent η_k.

so, let us define the sign function sgn(·) corresponding to the sign bit in the two's complement representation as

$$\operatorname{sgn}(x) = \begin{cases} +1, & \text{if } x \geq 0, \\ -1, & \text{if } x < 0. \end{cases} \tag{22}$$

Let S_I and S_Q, respectively, be the signs of the real and imaginary parts of the feedback signal y_k,

$$S_I = \operatorname{sgn}\{\Re[y_k]\}, \qquad S_Q = \operatorname{sgn}\{\Im[y_k]\}. \tag{23}$$

Quantization of the angle ϕ_k can be achieved by using the following update equation:

$$\begin{aligned} F_{k+1}^{[n]} &= F_k^{[n]} + \mu_k (S_I - jS_Q) e_k \\ &= F_k^{[n]} + \mu_k (\pm 1 \pm j) e_k \\ &= F_k^{[n]} + \sqrt{2}\mu_k e^{jm\pi/4} e_k, \end{aligned} \tag{24}$$

with

$$m = S_Q(S_I - 2). \tag{25}$$

The phase ϕ_k is therefore quantized to four possible values, that is, $\phi_k \in \{\pm\pi/4, \pm 3\pi/4\}$, thus effectively eliminating one complex multiplier (or four real multipliers).

The amplitude-dependent coefficient μ_k could be implemented as a lookup table with one coefficient per table entry. To minimize the required memory space and further reduce the implementation costs, μ_k can be constrained to powers of two and generated from the magnitude $|y_k|$ as follows:

$$u_k = 2^{\eta_k} \quad \text{with } \eta_k = \min\left\{-\left\lceil \log_2(|y_k|) \right\rceil, \eta_0\right\}, \tag{26}$$

where $\lceil\cdot\rceil$ stands for the ceil(·) rounding function (round to the nearest integer towards infinity) and η_0 is an arbitrary integer. In the above expression, it is assumed without loss of generality that the signal is normalized such that $|y_k| < 1$. Forcing the maximum exponent to η_0 sets a maximum value for μ_k to prevent any instability caused by the sensitivity to noise at low amplitudes. The base-two exponent η_k can be very efficiently generated with the simple combinatorial circuit illustrated in Figure 7. The amplitude is represented with 12 bits of resolution and the exponent η_k is represented with a three-bit binary word. This is equivalent to setting $\eta_0 = 7$. The first stage of the circuit outputs a one-hot binary vector (i.e., only one bit is set at a time) corresponding to the position of highest nonzero amplitude bit. The second stage encodes the position of the nonzero bit into a binary number, effectively computing a rounded base-two logarithm of the input amplitude. The scaling by μ_k can be implemented by a simple binary shifter.

The combinatorial logic implementation of the amplitude-dependent update coefficient lacks flexibility since the update speed cannot be changed. This issue can be tackled by introducing an additional coefficient μ_a that is programmable,

$$F_{k+1}^{[n]} = F_k^{[n]} + \mu_a \mu_k (S_I - jS_Q) e_k. \tag{27}$$

It should be noted that this low complexity update is even simpler to realize in hardware than the regular LMS, which requires two complex multipliers and has a much slower convergence speed.

This low-complexity update method (LCNLMS) was simulated and compared to the LMS and the NLMS. The previously described class-E amplifier is used in this experiment and a 10 MHz LTE signal with a composite PAPR of 8.5 dB is used to train the feedback LUT in the indirect learning setup. The size of the complex-gain LUTs is set to $L = 64$ entries. The complex-gain LUT entries are initially set to unity, which is functionally equivalent to bypassing the

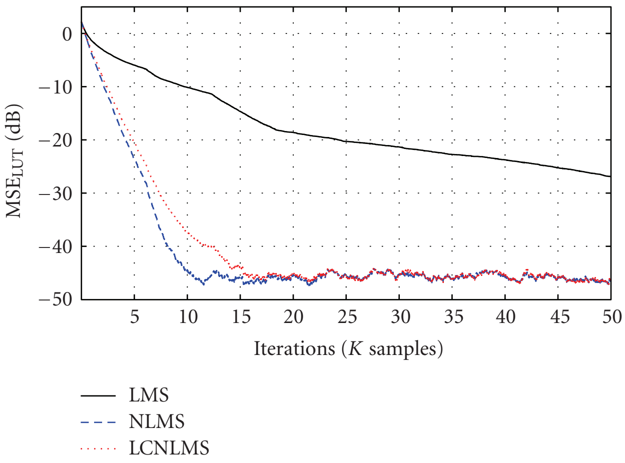


Figure 8: Convergence speed of LMS, NLMS, and LCNLMS.

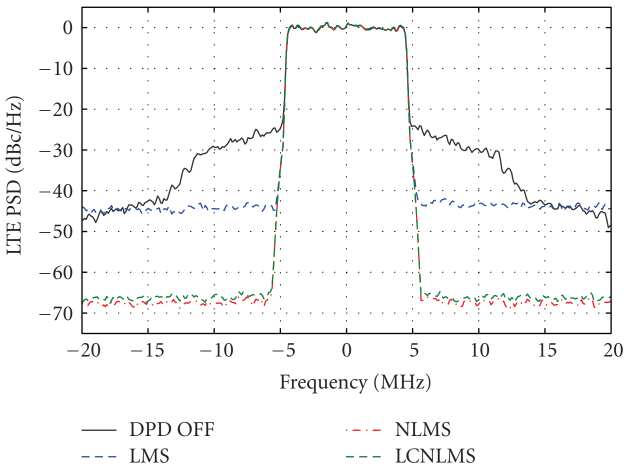


Figure 9: PSD performances of a complex-gain LUT predistorter trained using LMS, NLMS, and LCNLMS. Input signal is 10 MHz LTE signal.

predistorter. The resolution of the inphase and quadrature (I/Q) signal components is set to 13 bits. To measure the sensitivity of the adaptation to noise, the feedback signal is corrupted by additive white Gaussian noise (AWGN) and has an SNR of 33 dB. The LUT is updated at a rate of 30.76 MHz and the overall simulation was run at a sampling rate of 61.52 MHz. The update coefficient μ for LMS and NMLS is set to $\mu = 1/16$. Comparing (21) and (24) shows that the LCNLMS intrinsically increases the update rate by a factor of $\sqrt{2}$. On the other hand, the biased quantization of η_k in (26) approximately compensates for this factor. Therefore, setting $\mu_a = 1/16$ for the LCNLMS ensures a fair comparison.

Figure 8 compares the convergence of the regular LMS, the NLMS, and the proposed LCNLMS. It shows the instantaneous mean squared error MSE$_{\text{LUT}}$ between the updated LUT F and an optimal reference LUT H obtained via least-square approximations in each interval

$$\text{MSE}_{\text{LUT}} = \frac{1}{L}\sum_{n=1}^{L}\left|F^{[n]} - H^{[n]}\right|^2. \tag{28}$$

These results show that the convergence speed of the proposed LCNLMS is close to that of the NLMS. It should also be noted that the LCNLMS leads to an implementation complexity even lower than the generic LMS.

The adaptation was disabled after 5 ms and the trained LUT was used in the feedforward path. The resulting output PSDs are shown in Figure 9. The LCNLMS has the same performance as the NLMS. Despite the relatively long training time, the lower entries of the LMS-trained LUT did not converge, which explains the poor performance compared to the NLMS and LCNLMS.

3.1. Updating a Linearly-Interpolated LUT. Linear interpolation greatly reduces the LUT approximation errors and enables significant reduction of the required LUT size [6, 29]. If linear interpolation is used, for each feedback sample magnitude $|y_k|$ falling between addresses n and $n+1$, the interpolated complex-gain is

$$F_k = F^{[n]} + \gamma_k\left(F^{[n+1]} - F^{[n]}\right), \tag{29}$$

where γ_k is the interpolation factor. For the purpose of practical implementation, the address n and the interpolation factor γ_k are readily obtained from the amplitude bits

$$|y_k| \Longrightarrow \underbrace{\mathrm{a}_{11}\mathrm{a}_{10}\mathrm{a}_{09}\mathrm{a}_{08}\mathrm{a}_{07}\mathrm{a}_{06}}_{\text{address bits }(n)}\underbrace{\mathrm{a}_{05}\mathrm{a}_{04}\mathrm{a}_{03}\mathrm{a}_{02}\mathrm{a}_{01}\mathrm{a}_{00}}_{\text{interpolation factor }(\gamma_k)}. \tag{30}$$

It should be noted that for each input sample, two consecutive LUT entries must be fetched from memory and interpolated to compute the complex-gain. The hardware implementation and the sequencing of operations can be greatly simplified by using a dual-port memory. In general, dual-port memories are more expensive and larger in size than single-port memories of the same capacity. But in the case of the LUT interpolation, the two entries to be fetched are always located at consecutive addresses. Consequently, a dual-port memory of size L can be emulated using two single-port memory blocks of size $L/2$ and simple additional logic. One of the blocks stores the entries located at even addresses, and the other one stores the entries at odd addresses. This process allows the implementation of a pseudo dual-port memory at the same cost as a single-port memory. The only limitation is that simultaneous read/write operations require one address to be odd and the other to be even. In the case of a linearly interpolated LUT, this requirement is always satisfied because the addresses n and $n+1$ are consecutive. If linear interpolation is used in the feedback path (or updated LUT), the error signal e_k is given by

$$\begin{aligned} e_k &= z_k - F_k y_k \\ &= z_k - \left[(1-\gamma_k)F^{[n]} - \gamma_k F^{[n+1]}\right] y_k. \end{aligned} \tag{31}$$

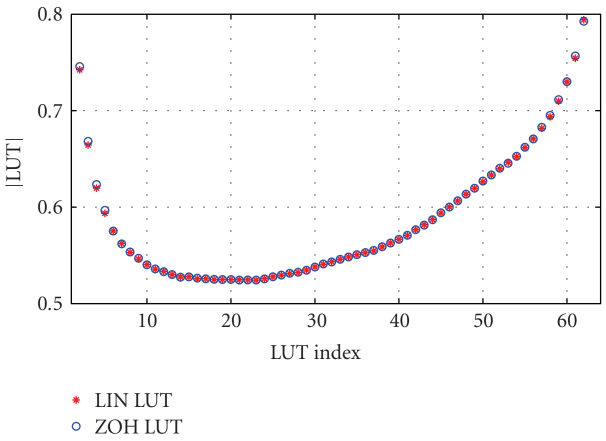


Figure 10: Converged LUT Predistorters using LCNLMS with linear (LIN) and nearest-neighbor (ZOH) interpolation in the feedback predistorter.

Since two entries are used to generate the interpolated complex-gain, both entries should be updated with each new data sample. The application of the same LMS algorithm by alternatively computing the gradients with respect to $F^{[n]}$ and $F^{[n+1]}$ results in the following update equations:

$$\begin{aligned} F_{k+1}^{[n]} &= F_k^{[n]} + (1-\gamma_k)\mu y_k^* e_k, \\ F_{k+1}^{[n+1]} &= F_k^{[n+1]} + \gamma_k \mu y_k^* e_k. \end{aligned} \quad (32)$$

Similarly to (27), the LCNLMS can also be applied to the linearly interpolated case, leading to the following update equations:

$$\begin{aligned} F_{k+1}^{[n]} &= F_k^{[n]} + (1-\gamma_k)\mu_a \mu_k (S_I - jS_Q) e_k, \\ F_{k+1}^{[n+1]} &= F_k^{[n+1]} + \gamma_k \mu_a \mu_k (S_I - jS_Q) e_k. \end{aligned} \quad (33)$$

Figure 10 shows that both the nearest neighbor and linear interpolation adaptations converge to the same solution. For the same update coefficient μ_a, the linearly interpolated adaptation has lower LUT approximation errors and therefore, results in a slightly better steady state performance. The steady state performance of the nearest neighbor adaptation can generally be improved by decreasing the update coefficient, at the cost of slower convergence.

Figure 11 uses a 4G LTE 10 MHz single-carrier (orthogonal) frequency-division multiple access (SC-FDMA) input stimulus with greater than 8.5 dB of composite PAPR to illustrate the resulting signal PSDs using the nearest neighbor and the linearly interpolated adaptation schemes. The simulation setup described in the previous section was reused, where the root mean square level of the digital signal was adjusted to account for the higher PAPR of the modulation waveform. The feedforward predistorter is linearly interpolated in both cases and the update coefficient is set to $\mu_a = 1/16$. It is evident that the close-in performances achieved using either scheme are quite comparable. The spectral regrowth is significantly reduced. The spectral floor using ZOH is 2 to 3 dB higher due to the intrinsic half-bit excess quantization noise of the ZOH as compared to the linear interpolation [29].

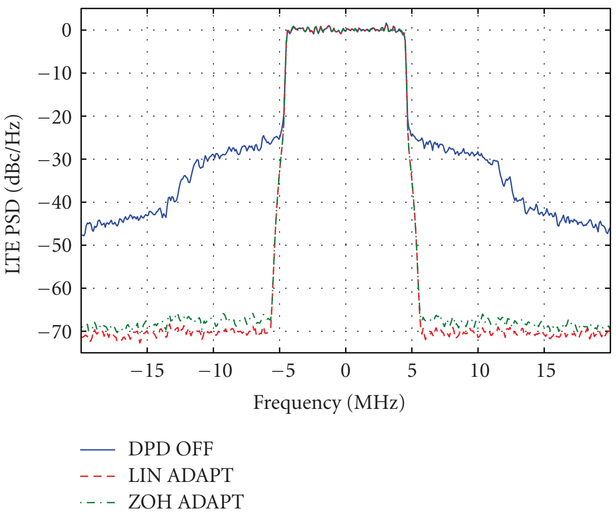


Figure 11: PSD performance of a complex-gain LUT predistorter trained using LCNLMS with linear (LIN) and nearest-neighbor (ZOH) interpolation in the feedback predistorter. Input waveform is a 10 MHz LTE OFDM signal.

Therefore, even when the feedforward predistorter is chosen to be linearly interpolated, the nearest neighbor adaptation can be used in the update branch of the indirect learning architecture, without much performance penalty. Note that ZOH requires only one memory read and write for each data sample. On the other hand, the linearly interpolated adaptation requires two memory reads and writes per data sample, placing more stringent timing requirements on the adaptation hardware.

4. Conclusions

In this paper, an efficient LUT-based adaptive memoryless predistorter configuration, with minimized chip area, has been presented. An amplitude approximation scheme suitable for digital baseband predistorters is proposed. A closed-form solution is derived to determine the optimal parameters for the amplitude approximation using any arbitrary angular interval size. A quantized amplitude approximation with three angular intervals is implemented in VHDL and synthesized with the SYNOPSYS DESIGN COMPILER. The predistorter performance using the proposed area-efficient scheme is shown to be within 2 dB of the ideal amplitude performance, while it outperforms the power-indexing in both design area and rejection of residual distortions by a wide margin.

An adaptation algorithm for complex-gain LUT predistorters based on the indirect learning architecture is also presented. The proposed adaptation algorithm has been optimized for efficient hardware implementation. It has a convergence speed that is comparable to the normalized

LMS and lends itself to very efficient hardware implementation. The proposed optimized adaptive predistorter can be extended to mitigate memory effects by adding a linear time-invariant filter in cascade with the memoryless complex-gain predistorter [5, 30].

References

[1] 3rd Generation Partnership Project, (3GPP), March 2010, http://www.3gpp.org/.

[2] R. J. P. de Figueiredo, L. Fang, and B. M. Lee, "Design of an adaptivepredistorter for solid state power amplifier in wireless OFDM systems," *Research Letters in Signal Processing*, vol. 2009, Article ID 515797, 5 pages, 2009.

[3] K. Waheed and S. N. Ba, "Adaptive digital linearization of a DRP based EDGE transmitter for cellular handsets," in *Proceedings of the 50th IEEE International Midwest Symposium on Circuits and Systems (MWCSAS '07)*, pp. 706–709, August 2007.

[4] J. K. Cavers, "Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements," *IEEE Transactions on Vehicular Technology*, vol. 39, no. 4, pp. 374–382, 1990.

[5] P. Jardin and G. Baudoin, "Filter lookup table method for power amplifier linearization," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 3, pp. 1076–1087, 2007.

[6] S. N. Ba, K. Waheed, and G. T. Zhou, "Efficient spacing scheme for a linearly interpolated lookup table predistorter," in *Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS '08)*, pp. 1512–1515, May 2008.

[7] L. Sundström, M. Faulkner, and M. Johansson, "Quantization analysis and design of a digital predistortion linearizer for RF power amplifiers," *IEEE Transactions on Vehicular Technology*, vol. 45, no. 4, pp. 707–719, 1996.

[8] J. K. Cavers, "Optimum table spacing in predistorting amplifier linearizers," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1699–1705, 1999.

[9] P. B. Kenington, *High Linearity RF Amplifier Design*, Artech House Publishers, Norwood, Mass, USA, 2000.

[10] K. C. Lee and P. Gardner, "Comparison of different adaptation algorithms for adaptive digital predistortion based on EDGE standard," in *Proceedings of IEEE MTT-S International Microwave Symposium Digest*, vol. 2, pp. 1353–1356, May 2001.

[11] B. Widrow and S. Stearns, *Adaptive Signal Processing*, Prentice Hall, Englewood Cliffs, NJ, USA, 1985.

[12] M. Onoe, "Fast amplitude approximation yielding either exact meanor minimum deviation for quadrature pairs," *Proceedings of the IEEE*, vol. 60, no. 7, pp. 921–922, 1972.

[13] A. E. Filip, "A baker's dozen magnitude approximations and their detection statistics," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 12, no. 1, pp. 86–89, 1976.

[14] F. Braun and H. Blaser, "Digital hardware for approximating the amplitude of quadrature pairs," *Electronics Letters*, vol. 10, no. 13, pp. 255–256, 1974.

[15] A. E. Filip, "Linear approximations to $\sqrt{x^2 + y^2}$ having equiripple error characteristics," *IEEE Trans Audio Electroacoust*, vol. AU-21, no. 6, pp. 554–556, 1973.

[16] W. A. Tsou, W. S. Wuen, T. Y. Yang, and K. A. Wen, "Analysis and compensation of the AM-AM and AM-PM distortion for CMOS cascode class-E power amplifier," *International Journal of Microwave Science and Technology*, vol. 2009, Article ID 597592, 9 pages, 2009.

[17] P. Cruise, C.-M. Hung, R. B. Staszewski et al., "A digital-to-RF-amplitude converter for GSM/GPRS/EDGE in 90-nm digital CMOS," in *Proceedings of IEEE Radio Frequency Integrated Circuits Symposium (RFIC '05)*, vol. RMO1A-4, pp. 21–24, June 2005.

[18] C. Eun and E. J. Powers, "A new volterra predistorter based on the indirect learning architecture," *IEEE Transactions on Signal Processing*, vol. 45, no. 1, pp. 223–227, 1997.

[19] D. H. Brandwood, "A complex gradient operator and its application in adaptive array theory," *IEE Proceedings F*, vol. 130, no. 1, pp. 11–16, 1983.

[20] B. Widrow, J. McCool, and M. Ball, "The complex LMS algorithm," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 719–720, 1975.

[21] S. C. Douglas, "Adaptive filters employing partial updates," *IEEE Transactions on Circuits and Systems II*, vol. 44, no. 3, pp. 209–216, 1997.

[22] P. Ramos, R. Torrubia, A. López, A. Salinas, and E. Masgrau, "Step size bound of the sequential partial update LMS algorithm with periodic input signals," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2007, Article ID 10231, 15 pages, 2007.

[23] B. Widrow and E. Walach, "On the statistical efficiency of the LMS algorithm with nonstationary inputs," *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 211–221, 1984.

[24] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*, Prentice Hall, Englewood Cliffs, NJ, USA, 1984.

[25] T. Aboulnasr and K. Mayyas, "Complexity reduction of the NLMS algorithm via selective coefficient update," *IEEE Transactions on Signal Processing*, vol. 47, no. 5, pp. 1421–1424, 1999.

[26] J. L. Moschner, *Adaptive filtering with clipped input data*, Ph.D. dissertation, Stanford University, Stanford, Calif, USA, June 1970.

[27] M. Lotfizad and H. S. Yazdi, "Modified clipped LMS algorithm," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 8, pp. 1229–1234, 2005.

[28] L. Crum and S. Wu, "Convergence of the quantizing learning method forsystem identification," *IEEE Transactions on Automatic Control*, vol. 13, no. 3, pp. 297–298, 1968.

[29] S. N. Ba, K. Waheed, and G. T. Zhou, "Optimal spacing of a linearlyinterpolated complex-gain LUT predistorter," *IEEE Transactions onVehicular Technology*, vol. 59, no. 2, pp. 673–681, 2010.

[30] L. Ding, R. Raich, and G. T. Zhou, "A hammerstein predistortion linearization design based on the indirect learning architecture," in *Proceedings of IEEE International Conference on Acoustic, Speech, and Signal Processing (ICASSP '02)*, vol. 3, May 2002.