A CROSSCORRELATION PREDISTORTER USING MEMORY POLYNOMIALS

André B.J. Kokkeler

University of Twente
P.O. Box 217, 7500 AE Enschede, The Netherlands.
Phone: +31 53 4894291, Fax: +31 53 4894571,
kokkeler@cs.utwente.nl

ABSTRACT

Amplification of signals with fluctuating envelopes inevitably leads to distortion because of non-linear behavior of the Power Amplifier (PA). Digital Predistortion can counteract these non-linear effects. In this paper, a novel Digital Predistortion architecture is presented which is based on the calculation of crosscorrelation functions using coarsely quantized signals. The crosscorrelation functions are transformed to the frequency domain and the spectra are used to calculate the coefficients of the predistorter memory polynomial. This method has reduced complexity and slightly improved average performance in comparison with existing schemes.

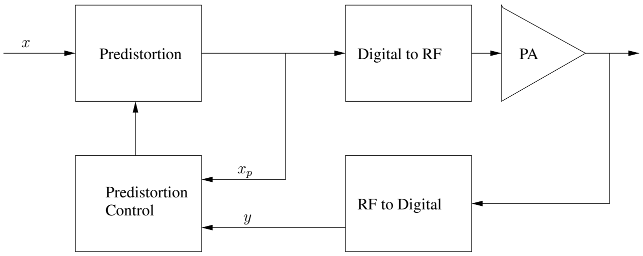

Fig. 1. Digital Predistortion

1. INTRODUCTION

One of the characteristics of modern, spectrally efficient transmission formats (CDMA, OFDM) is their relatively high peak-to-average ratio. Non-linear Power Amplifiers (PA) fed by signals with high peak-to-average ratio introduce severe distortion. Distortion within the band of transmission degrades signal detection. Distortion outside the band of transmission (spectral regrowth) interferes with the signals transmitted by other users at adjacent channels and is therefore called Adjacent Channel Interference (ACI).

Power Amplifiers are costly and their operation is power-inefficient because of the measures to limit distortion. To counteract the non-linearity of a PA, several techniques have been investigated: feedforward, feedback and predistortion. In this document we will concentrate on predistortion. The general principle is to apply the inverse input-output relation of the PA to the signal at the input of the PA. Predistortion followed by the PA (and its inherent distortion) should result in linear amplification. Predistortion is applied in the analog domain (see [1]) and in the digital domain (see [2]). In digital predistortion, the baseband signal is predistorted before it is converted to the analog domain, frequency translated to RF and amplified. (see figure 1).

Because the input-output relation of the PA changes in time due to temperature changes and aging of components in the analog part, a control mechanism constantly adapts the predistortion. A small fraction of the PA output is fed back and converted from RF to baseband. An adaptation algorithm compares this signal with the output of the predistorter. Two different approaches exist: Direct- and Indirect Learning. When using Direct Learning, the input-output relation of the PA is determined. This relation is inverted and used for predistortion. For an example, see [3]. When using Indirect Learning, the input-output relation of the PA is determined indirectly, the inverse relation directly. This inverse relation is used as predistorter. Recent Indirect Learning schemes (see [4]) involve multiplications of relatively large matrices. If the number of samples involved in calculating a new setting of the Predistorter equals T, the number of multiplications is $O(T)$. PA's suffer from memory effects. The output signal of the PA at any time-instant depends on the input at that instant and on the input signal at previous time-instants. Recent Indirect Learning schemes are suitable for counteracting the memory effects. The purpose of this paper is to introduce a novel predistortion architecture with reduced complexity. In Section 2, the architecture will be presented. It is based on the Indirect Learning scheme. In Section 3, implementation issues are considered and in Section 4, simulation results are given.

0-7803-8251-X/04/$17.00 ©2004 IEEE

2. THE CROSSCORRELATION PREDISTORTER

The predistorted signal x_p and the PA output (converted back to baseband) y are used to determine the output-to-input relation of the PA. This relation can be described by means of a memory polynomial (see [3]). The polynomial used in our paper is similar to the polynomial used in [4]:

$$x_p(t) = \sum_{k=1}^{K} \sum_{\tau=0}^{\tau_{max}-1} a_{k\tau} y(t-\tau) \mid y(t-\tau) \mid^{k-1} \tag{1}$$

The memory length of the predistorter equals τ_{max} and the polynomial order equals K. To find the values $a_{k\tau}$, powers of y and of delayed versions of y are generated:

$$y_{k\tau} = y(t-\tau) \mid y(t-\tau) \mid^{k-1} \tag{2}$$

for $t \in \mathbb{Z}$. Via linear combination of these signals, an estimate of signal x_p can be constructed using the least squares criterion:

$$\begin{aligned} \mathbf{x}_p &= [x_p(1), ..., x_p(T)]^T & (3)\\ \mathbf{y}_{k\tau} &= [y_{k\tau}(1), ..., y_{k\tau}(T)] & (4)\\ \mathbf{Y} &= [\mathbf{y}_{10}, ..., \mathbf{y}_{K0}, ..., \mathbf{y}_{1\tau_{max}-1}, ..., \mathbf{y}_{K\tau_{max}-1}] & (5)\\ \hat{\mathbf{a}} &= (\mathbf{Y^H Y})^{-1} \mathbf{Y^H x_p} & (6) \end{aligned}$$

where ^{H} indicates the complex conjugate transpose. T is the number of consecutive samples, available to the predistorter. In the remainder of this paper the predistorter based on this solution will be called: LS-predistorter.

In the crosscorrelation predistorter, the signals $y_{k\tau}$ are not combined directly. First $y_{k\tau}$ and x_p are crosscorrelated with a reference signal. We chose to crosscorrelate with a single-bit representation x_Q of x_p, because the quantized signal x_Q has significant power in the adjacent channels when the predistorter is operational. Single-bit quantization is defined as:

$$x_Q(x_p) = \text{sign}(\text{Re}(x_p)) + j\text{sign}(\text{Im}(x_p)) \tag{7}$$

where sign() is defined as:

$$\text{sign}(x) = \begin{cases} -1 & x < 0 \\ 1 & x \geq 0 \end{cases} \tag{8}$$

Re(x) indicates the real part of a complex value x and Im(x) the imaginary part. Using a single-bit quantized value as one of the operands of a crosscorrelation drastically reduces the complexity. The multiplications involved become very simple to implement: when the first operand equals 1 or -1, the second operand of the multiplication is either left unchanged or its sign is reversed respectively. The number of points (or lags) of the crosscorrelation is even and equals N. Using vector notations, the single-bit quantized signal equals:

$$\mathbf{x_Q} = \left[x_Q(x_p(\frac{1}{2}N+1)), ..., x_Q(x_p(T-\frac{1}{2}N+1)) \right]^T \tag{9}$$

We define the matrices $\mathbf{Y_{k\tau}}$ for $k = 1, ..., K$ and $\tau = 0, ..., \tau_{max}-1$, and $\mathbf{X_p}$ as:

$$\begin{aligned} \mathbf{Y_{k\tau}} = \quad & \Big[[y_{k\tau}(1), ..., y_{k\tau}(T-N+1)]^T, ..., \\ & \qquad [y_{k\tau}(N), ..., y_{k\tau}(T)]^T \Big]^T \\ \mathbf{X_p} = \quad & \Big[[x_p(1), ..., x_p(T-N+1)]^T, ..., \\ & \qquad [x_p(N), ..., x_p(T)]^T \Big]^T \end{aligned} \tag{10}$$

The crosscorrelations are defined as:

$$\begin{aligned} \mathbf{r_{k\tau}} &= \mathbf{Y_{k\tau} x_Q^*} & (11)\\ \mathbf{r_p} &= \mathbf{X_p x_Q^*} & (12) \end{aligned}$$

The crosscorrelation vectors $\mathbf{r_{k\tau}}$ and $\mathbf{r_p}$ are tapered with a Hanning taper:

$$h(j) = \frac{1}{2}(1 - \cos(2\pi(j-1)/(N-1))) \tag{13}$$

$$\mathbf{H} = \mathrm{d}iag(h(1), ..., h(N)) \tag{14}$$

where 'diag()' indicates a diagonal $N \times N$ matrix. A Discrete Fourier Transform is applied to the tapered crosscorrelation vectors:

$$\begin{aligned} \mathbf{f_{k\tau}} &= \mathbf{WHr_{k\tau}} & (15)\\ \mathbf{f_p} &= \mathbf{WHr_p} & (16) \end{aligned}$$

where $\mathbf{W}$ equals the DFT kernel. The elements w_{pq} of the kernel are defined as:

$$w_{pq} = \exp^{-i2\pi \frac{p-1}{N}(q-1)} \tag{17}$$

The vectors (spectra) $\mathbf{f_{k\tau}}$ are concatenated into a matrix $\mathbf{F}$:

$$\mathbf{F} = [\mathbf{f_{10}}, ..., \mathbf{f_{K0}}, ..., \mathbf{f_{1\tau_{max}-1}}, ..., \mathbf{f_{K\tau_{max}-1}}] \tag{18}$$

The vector $\mathbf{a}$ is defined as:

$$\mathbf{a} = [a_{10}, ..., a_{K0}, ..., a_{1\tau_{max}-1}, ..., a_{K\tau_{max}-1}] \tag{19}$$

The memory polynomial predistorter is then described in the frequency domain as:

$$\mathbf{f_p} = \mathbf{Fa} \tag{20}$$

The least squares solution minimizes the *absolute* error over the frequency domain. Because the signal has a relatively low power spectral density in the adjacent channels, the *relative* errors in the adjacent channels can be (too) large. For that reason, we minimize the relative error. This is realized by normalizing the spectra with $\mathbf{f_p}$:

$$g_{k\tau}(n) = \frac{f_{k\tau}(n)}{f_p(n)} \tag{21}$$

$$\mathbf{g_{k\tau}} = [g_{k\tau}(1), ..., g_{k\tau}(N)]^T \tag{22}$$

$$\mathbf{G} = [\mathbf{g_{10}}, ..., \mathbf{g_{K0}}, ..., \mathbf{g_{1\tau_{max}-1}}, ..., \mathbf{g_{K\tau_{max}-1}}] \tag{23}$$

$$\mathbf{g_p}(n) = 1, \;\; n = 1, ..., N \tag{24}$$

The normalized version of equation (20) becomes:

$$\mathbf{g_p} = \mathbf{Ga} \tag{25}$$

The least-squares solution $\hat{\mathbf{a}}$ equals:

$$\hat{\mathbf{a}} = (\mathbf{G^H G})^{-1}\mathbf{G^H g_p} \tag{26}$$

To reduce the effects of noise on the correlation functions $\mathbf{g_{k\tau}}$, we select only those vector elements which represent the power in the main channel and the first adjacent channel leading to vectors $\mathbf{g_p}$ and $\mathbf{g_{k\tau}}$ with limited length.

3. IMPLEMENTATION ASPECTS

The complexity of current polynomial predistorters is at least $O(T)$; the number of complex multiplications scales linearly with the number of samples used to update the predistorter polynomial coefficients $\hat{\mathbf{a}}$. In the crosscorrelation predistorter, the size of the vectors $\mathbf{f_p}$ and $\mathbf{f_{k\tau}}$ is reduced to N (instead of T). N equals the number of points (or lags) of the crosscorrelation functions and in general N is much smaller than T. If an FFT is used to transform the vectors from the time domain to the frequency domain, the complexity is $O(N \log_2 N)$. The reduction of the length of the vectors is due to the crosscorrelation. The crosscorrelation is easy to implement. Because of the single-bit quantization of x_p, no full-precision complex multiplications are required, so the overall complexity of the crosscorrelation predistorter is determined by the FFT.

4. SIMULATION RESULTS

A simulator, based on the structure presented in the previous sections has been implemented. As a signal source we used two independent data generators: one for the real part of the signal and one for the imaginary part. A data generator is based on the signal source used in [5] and generates 416 sinewaves with different consecutive frequencies, equal amplitudes and randomly selected phase. The oversampling factor is 10. The PA model used is a Wiener-Hammerstein model as presented in [4]. We chose this model because it includes PA memory effects. The model consists of an IIR filter $H(z)$ followed by a memoryless polynomial which on its turn is followed by a second IIR filter $G(z)$. The specifications are:

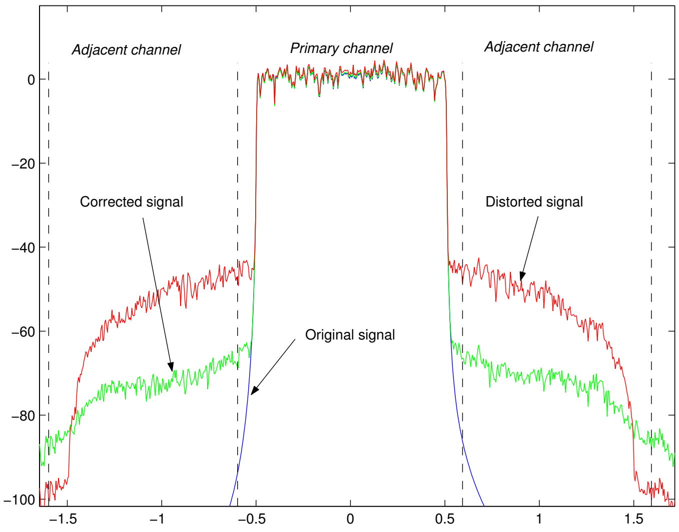

Fig. 2. Spectra

$$H(z) = \frac{1 + 0.5z^{-2}}{1 - 0.2z^{-1}}, \quad G(z) = \frac{1 - 0.1z^{-2}}{1 - 0.4z^{-1}}$$

The memoryless polynomial equals:

$$w(n) = \sum_{k=1}^{K} b_k \, v(n) \mid v(n) \mid^{k-1}$$

where v indicates the output of H and w the input of G. The coefficients are: $b_1 = 1.0108 + 0.0858j, b_3 = 0.0879 - 0.1583j, b_5 = -1.0992 - 0.8891j$. The standard deviation of the complex input signal is $4.24\,10^{-2}$. In our simulations, we used blocks of 8K samples ($T = 8192$) and 64-point crosscorrelation functions ($N = 64$). For the predistorter, we used only odd polynomials up to the fifth order ($k = 1, 3, 5$) and the memory length is 2 ($\tau = 0, 1$).

In figure 2, a typical double sided power spectral density (PSD) of the original signal is given, together with the baseband equivalent output of the PA with and without predistortion. The PSD's are normalized with the power of the original signal.

We see that the Adjacent Channel Interference (ACI) levels, have been reduced significantly. Outside the Adjacent Channels, the power has increased slightly but remains below -80 dB. The reduction depends on the input

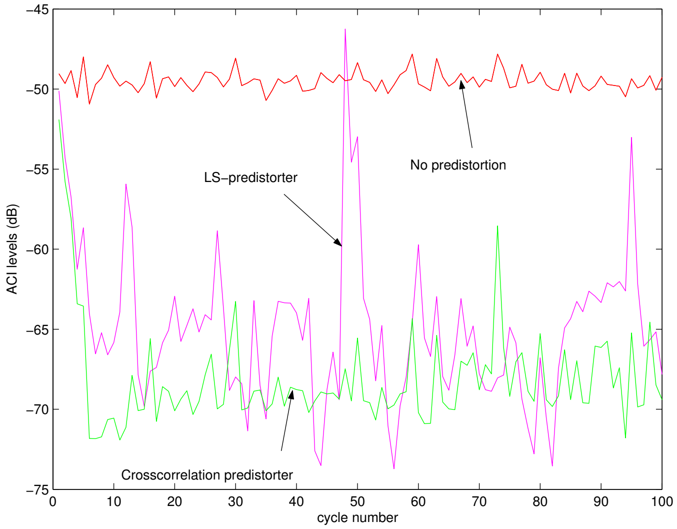

Fig. 3. ACI levels with and without predistortion for PA model 1

signal. New settings of the predistorter are based on data that already has been transmitted. New data might contain signal-excursions which have not been accounted for. This can lead to limited suppression of ACI levels and in severe cases to settings from which the predistorter cannot recover. For that reason, we simulated 100 cycles where every cycle consisted of the following stages: generation of an 8K samples dataset for cycle i, predistortion of this dataset using the predistorter settings found in cycle $i-1$, distortion, determination of the memory polynomial and determination of the new predistorter settings by averaging the memory polynomial found (weight = 0.25) with the old predistorter settings (weight = 1). The crest-factors (ratio of the peak amplitude and standard deviation) of the generated data range from 2.5 to 3.4 dB (8 to 11 dB). The ACI levels with and without predistortion for both the LS-predistorter and crosscorrelation predistorter are given in figure 3. The bandwidth of the two adjacent channels equals the bandwidth of the primary channel. The ACI-levels reflect the maximum of the total power of the two adjacent channels related to the power in the primary channel.

We see that after 10 cycles, the predistorter system stabilizes. For the crosscorrelation predistorter, the average reduction of ACI-levels is around 20 dB. For the LS-predistorter, the average reduction is slightly less and varies significantly from cycle to cycle. At cycle 48, the LS-predistorter even increases the ACI-levels.

5. CONCLUSIONS

In this paper a novel digital predistortion architecture, the crosscorrelation predistorter, is presented. It is based on the Indirect Learning scheme.

Simulations using this crosscorrelation predistorter have shown significant reductions of ACI levels. Compared with the LS-predistorter, the crosscorrelation predistorter has reduced complexity and shows a slightly increased reduction of ACI-levels. The complexity is reduced because the number of complex multiplications depends on the number of points (lags) of the crosscorrelation functions (N) and equals $O(N \log_2 N)$. This complexity is independent of the size of the data set (T), used to update the predistorter. In existing schemes, the number of complex multiplications is $O(T)$. In our simulations, $N = 64$ and $T = 8192$. The crosscorrelation predistorter is based on the calculation of crosscorrelation functions. Because one of the inputs of the crosscorrelation is a single-bit signal, no complex multiplications are involved in that part of the predistorter.

6. REFERENCES

[1] C.G. Rey, “Predistorter linearizes CDMA power amplifiers,” *Microwaves RF*, Oct. 1998, pp. 114-123.

[2] F. Zavosh, M. Thomas, C. Thron, T. Hall, D. Artusi, D. Anderson, D. Ngo, and D. Runton, “Digital predistortion techniques for RF power amplifiers with CDMA applications,” *Microwave Journal*, Oct. 1999. linearization using predistortion-Experimental results”, *IEEE Trans. Veh. Technol.*, vol. 43, no. 2, May 1994, pp. 323-332. performance of an adaptive digital linearized power amplifier”, *IEEE Trans. Veh. Technol.*, vol. 41, no.4, Dec. 1992, pp. 395-400.

[3] J. Kim and K. Konstantinou, “Digital predistortion of wideband signals based on power amplifier model with memory,” *Electronics Letters*, vol. 37, no. 23, Nov. 2001, pp. 1417-1418. memory-less nonlinearity proceded by a dynamic linear system”, *Proc. GLOBECOM 1995*, pp. 152-156.

[4] L. Ding, G.T.Zhou, Z. Ma, D.R. Morgan, J.S. Kenney, J. Kim, and C.R. Giardina, “A robust digital baseband predistorter constructed using memory polynomials,” *IEEE Trans. on Communications*, March 2002 (submitted).

[5] J. K. Cavers, “Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements”, *IEEE Trans. Veh. Technol.*, vol. 39, no. 4, Nov. 1990, pp. 374-383. predistortion linearization design based on the indirect learning architecture,” *Proc. ICASSP 2002*, May 13-17, 2002.