Baseband Predistortion Lineariser Using Direct Spline Computation

G. Acciari, F. Giannini, E. Limiti, M. Rossi
Department of Electronic Engineering, University of Roma "Tor Vergata" - Roma - Italy

A baseband predistorter is presented. Key features of the predistorter resides in the use of cubic splines interpolation to generate predistorted input data to the power amplifier, resulting in a reduction of computational effort with respect to traditional polynomial interpolators. Simulated behaviour of the proposed scheme is presented, demonstrating the effectiveness of the approach.

INTRODUCTION

The widespread diffusion of radiocommunication systems naturally requires high spectral efficiency modulation schemes. Linear modulations can achieve the required efficiency performances, but the resulting non-constant envelope is not well-suited to cope with the stringent requirements requested to transmitter stage power amplifiers (PAs). A non constant envelope signal passing through a non linear high power amplifier results in a distorted output suffering from spectral regrowth and intermodulation effects. In order to avoid the expensive solution of operating the transmitter with a high back-off level, several linearisation techniques can be adopted, operating both at microwave and baseband frequencies. In this work the attention is focused in a digital realization of baseband predistorter. The task of such a digital predistorter is to modify the signal, before its upconversion, to compensate for the subsequent power amplifier distortion. The resulting output signal emerges therefore simply delayed and amplified.

DESIGN APPROACH

Assuming the PA as a memoryless non-linear device and a band-limited input signal, it is possible to analyse its non linear behaviour considering two particular transfer characteristics, namely the AM/AM compression and the AM/PM conversion curves. Under this assumption, an equivalent model of the rf-PA is derived, considering the latter as operating in the domain of the complex envelope: every point in the input complex plane is non linearly mapped into a complex plane by the PA. In this picture, the predistorter modifies the complex envelope of the original input signal so that the output is simply obtained by the product between a real constant quantity (gain) and the input complex envelope. In this work a digital baseband lineariser is proposed, operating the necessary predistortion by means of a cubic spline numerical approximation.
Simple equations for the predistortion procedure are derived, and a new way to execute the predistortion, with a direct calculation in order to simplify the computational complexity, is proposed. The cubic splines as tool for obtaining the predistorsion coefficients without the need of Cartesian to polar coordinates transformations [1] are introduced. At the same time, the need of higher order polynomial fitting [2] is avoided by a third order polynomial computation and predistortion is achieved with minor computational effort. With a multistage implementation it's also possible to filter the noisy data [3].
A typical AM/AM compression curve, drawn using the normalized Saleh [4] model, is reported in figure 1.

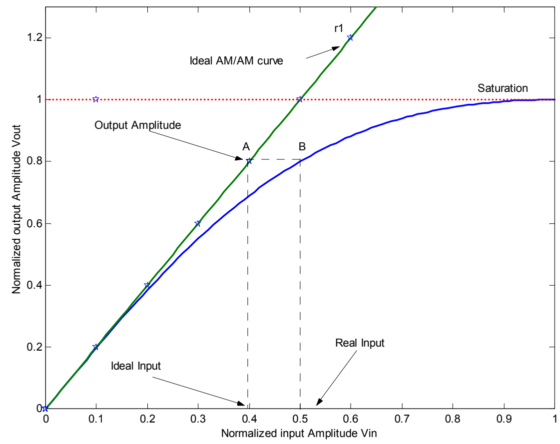

Fig. 1. AM/AM compression and ideal curve

Operating in the non linear region in order to obtain an output signal corresponding to an input signal passing through a linear amplifier (point A on line r_1), the PA needs to be overdriven until point B. Let us assume the gain of the amplifier to be complex in order to take into account for phase in-out relationship. If an input vector $r_0 \cdot e^{j\vartheta_0}$ is assumed, the actual output vector can be expressed as: $A_{(r_0)} \cdot e^{j[\vartheta_0 + \phi(r_0)]}$, while the desired ideal output is expressed by : $G_1 r_0 \cdot e^{j[\vartheta_0 + \phi_1]}$ where $G_1 e^{j\phi_1}$ is the PA linear gain.
Such in-out relationship can be rewritten in the form: $r \cdot e^{j\vartheta} \rightarrow A_{(r)} \cdot e^{j[\vartheta + \phi(r)]}$ and the PA complex gain becomes therefore $G_{A(r)} = \frac{A_{(r)}}{r} \cdot e^{j\phi(r)}$. Given a generic input $r_0 \cdot e^{j\vartheta_0}$ it is possible to compute the predistorted input signal $r_p \cdot e^{j\vartheta_p}$ such that the output is as in the linear case, i.e. :

$$r_p e^{j\phi_p} \cdot G_{A(r_p)} = G_1 \cdot r_0 e^{j(\vartheta_0 + \phi_1)} \qquad (1)$$

If $R_{A(r_p)} + jI_{A(r_p)} = 1/G_{A(r_p)}$ is a complex attenuation, the previous expression can be put in the form:

$$r_p e^{j\phi_p} = \left(R_{A(r_p)} + jI_{A(r_p)}\right) \cdot G_1 \cdot r_0 e^{j(\vartheta_0 + \phi_1)} \qquad (2)$$

The complex attenuation of the PA can be separated into its real and imaginary parts and then interpolated with cubic splines, once the breakpoints have been properly selected. We choose to calculate the predistorted sample directly, without the need to index a LUT[4]. In order to reduce the computational effort an interpolation based on the modulus squared of the sample has been used. The PA model used in the simulation is obtained from a real K-Band TWTA, characterized by the AM/AM compression curve as in fig.2

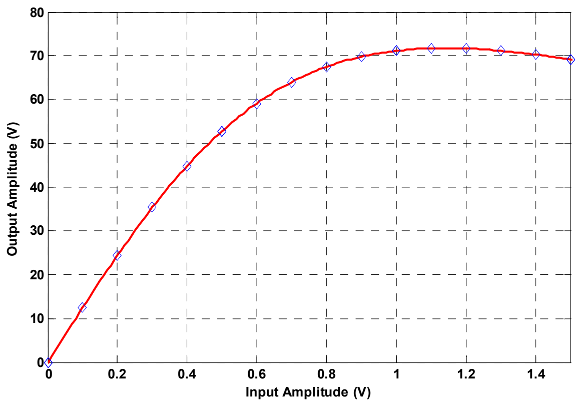

Fig. 2. TWTA AM/AM compression

And an AM/PM distorsion curve as in fig. 3

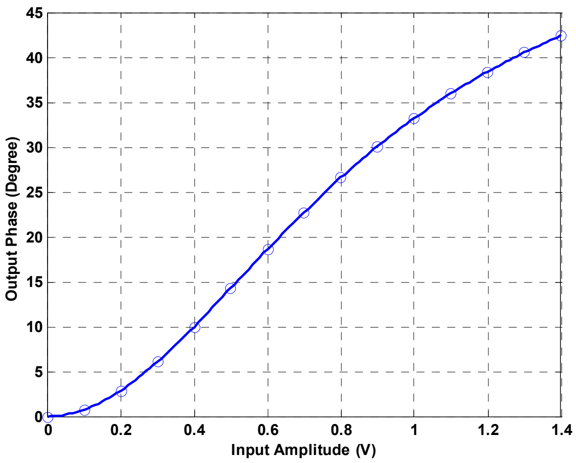

Fig. 3.TWTA AM/PM distortion curve

In fig.4 a block diagram of the predistorter is shown.

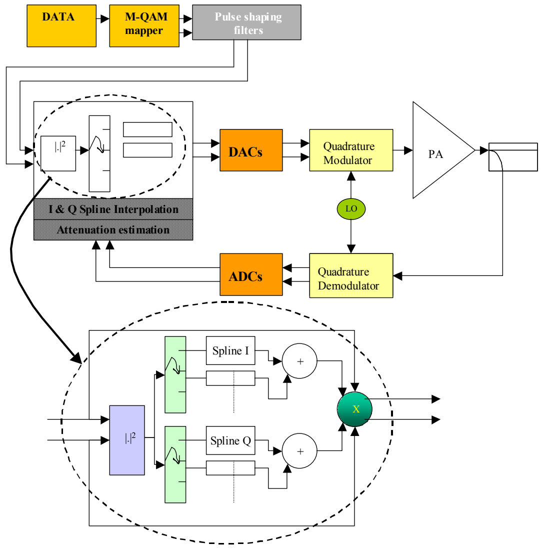

Fig. 4. Predistorster and PA block diagram

A 16-QAM constellation has been used and the resulting system has been simulated by [5]. The pulse shaping filter box applies two independents square root raised cosine FIR filter to the data with a 8X oversampling rate [6]. The signal is fed into the predistortion block and during the learning phase predistortion is obviously not applied, since PA characteristics are unknown at the moment. Complex samples are stored in a random access memory (RAM) and applied to a couple of DACs and reconstruction filters. A quadrature modulator and the PA complete the schematic.

During the learning phase the entire range of input/output values has to be swept. A sample of the amplified signal is demodulated (by a quadrature demodulator) and then fed into a pair of anti-aliasing filters and ADCs. Digital samples are stored in a RAM to be processed at the end of the learning phase. In the case here reported a number N=1000 of samples has been used for delay compensation. When the above N samples have been acquired by the system, the adaptation algorithm begins by sorting the vector of stored data. The DSP generates M breakpoints and proceeds to build the interpolators for the complex attenuation. M sets of 4 coefficients describing each a cubic spline are therefore generated. We used five equally spaced breakpoints in order to build the natural cubic spline interpolators. The synthetic reconstruction of the real and imaginary part of the complex attenuation starts with the building of the second derivatives matrix [A] considering n+1=5 breakpoints:

$$\begin{aligned} & h_i M_{i-1} + 2(h_i + h_{i+1})M_i + h_{i+1}M_{i+1} = \\ & = 6\frac{y_{i+1} - y_i}{h_{i+1}} - 6\frac{y_i - y_{i-1}}{h_i} \rightarrow i = 1,..,n-1 \end{aligned} \qquad (3)$$

Where h_i is the subinterval amplitude, M_i is the second derivative at the breakpoint *i* and y_i the ordinates. Since the breakpoints are equally spaced h_i=h, we solve for the linear system:

$$[A]=\begin{bmatrix}4&1&0\\1&4&1\\0&1&4\end{bmatrix}\Rightarrow[A]\begin{bmatrix}M_1\\M_2\\M_3\end{bmatrix}=\frac{6}{h^2}\begin{bmatrix}(y_2-y_1)-(y_1-y_0)\\(y_3-y_2)-(y_2-y_1)\\(y_4-y_3)-(y_3-y_2)\end{bmatrix} \quad (4)$$

With $M_0=M_4=0$. Next we compute the integration constants B_i and C_i as :

$$B_i=\frac{1}{h}(y_i-y_{i-1})-\frac{h}{6}(M_i-M_{i-1})$$
$$C_i=y_{i-1}-\frac{h^2}{6}M_{i-1} \quad (5)$$
$$i=0,\ldots,4$$

So the final form for each cubic spline interpolator $F_i(x)$ between each couple of breakpoints becomes :

$$F_i(x)=\frac{(x_i-x)^3M_{i-1}+(x-x_{i-1})^3}{6h}+B_i(x-x_{i-1})+C_i \quad (6)$$

The predistortion phase starts computing the squared magnitude of the complex vector entering the PD (thus requiring 1 elementary sum and 2 products). A very simple test must be executed at this time to determine the right set of polynomial coefficients and successively the calculation starts, involving about 3+6 floating point (3 sums and 3 products) operations per sample per channel (I & Q).
The operations involved are quite simple but other complex tasks must be done at the same time. One of this is to compensate for the delay between the generated and demodulated data: this is accomplished using the correlation function between of the squared modulus of the original data and the demodulated data. In order to increase resolution the demodulated data are filtered with an upsampler [7][8].
Finally, the complex product of the interpolated output data and the input vector is performed: this task implies 4 products and 2 sums per sample to generate the output.
If only one DSP is used, the computational effort is estimated to be about 21 Floating point operations/sample, but the task can be easily parallelised. For a 16-QAM modulation scheme and a source bit rate of 1 Mbit/s, the symbol rate is about 250Ksymb/s. Taking into account an oversampling rate of 8X this results in a sample rate of 2Msample/s. The single DSP solution leads to 42 MFLOPS. If the sample rate is too high for a single DSP, other faster calculus devices as FPGAs can be adopted.
Considering an oversample rate of 8X we obtain different floating point operations per sample according to the data rate and the M-QAM modulation levels, as indicated in tab.1

DATA RATE (kbps)	4-QAM (MFLOPS)	16-QAM (MFLOPS)	64-QAM (MFLOPS)	256-QAM (MFLOPS)
100,0	8,40	4,20	2,80	2,10
200,0	16,80	8,40	5,60	4,20
300,0	25,20	12,60	8,40	6,30
400,0	33,60	16,80	11,20	8,40
500,0	42,00	21,00	14,00	10,50
600,0	50,40	25,20	16,80	12,60
700,0	58,80	29,40	19,60	14,70
800,0	67,20	33,60	22,40	16,80
900,0	75,60	37,80	25,20	18,90
1000,0	84,00	42,00	28,00	21,00
1100,0	92,40	46,20	30,80	23,10

Tab.1 MFLOPS vs. data rate & M-QAM levels

RESULTS

The algorithm proposed in this contribution has been tested using a model of a TWTA from Thomson Tubes. The simulations have been performed in the baseband domain, varying the power level of the signal fed into the amplifier in order to demonstrate the predistortion benefits. In fig.5 the simulated output spectra of the linearly amplified input signal, together with the PA output, with and without predistortion are reported.

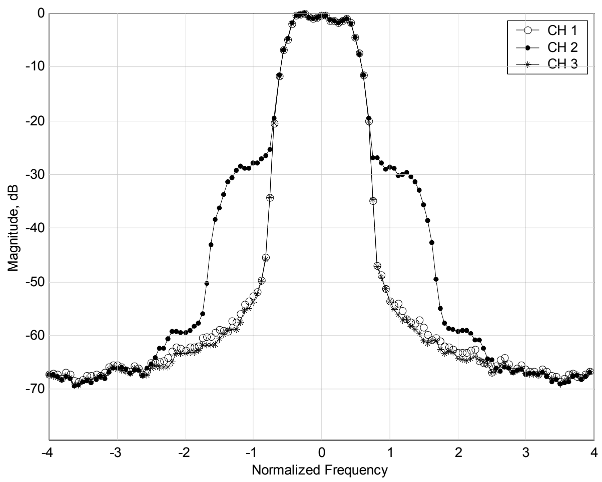

Fig. 5 Spectrum of scaled input signal (--x--), PA output with PD (--o--) and without PD (--•--)

Without predistortion a visible spectral regrowth is present while using the proposed baseband predistortion the input and output spectra appropriately scale by a constant factor. The PA has been driven near the saturation (2 dB compression) and the eye diagram and scatter plot of the received 16-QAM constellation without PD and with PD are reported in Fig.6 and Fig.7 respectively. As clearly visible from the figures the presence of the PD improve the eye opening thus reducing symbols spreading.

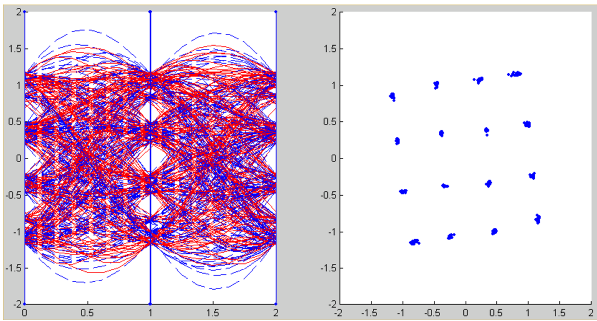

Fig. 6 eye diagram and scatter plot of a 16-QAM with the PA driven near saturation point without PD

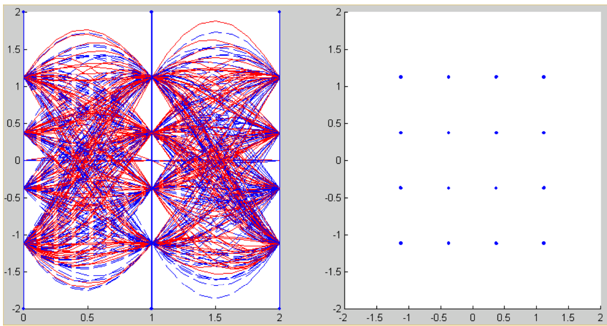

Fig. 7 eye diagram and scatter plot of a 16-QAM with the PA driven near saturation point with PD

In figure 8 is reported the power spectral density of a of a synthetic two-tone test at 19 and 23 KHz without predistorsion, it's clearly visible the intermodulation and distorsion introduced by the PA.

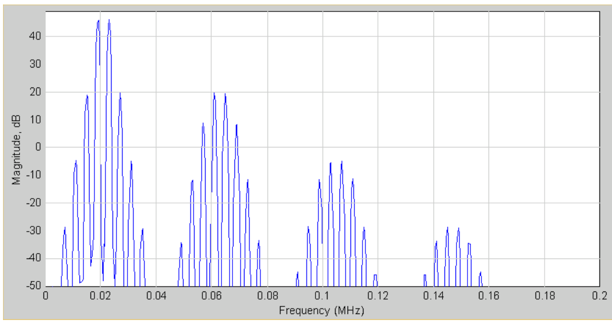

Fig. 8. Power Spectral Density of a two-tone test w/o predistorsion

When the predistorter is turned on we obtain the power spectrum reported in figure 9

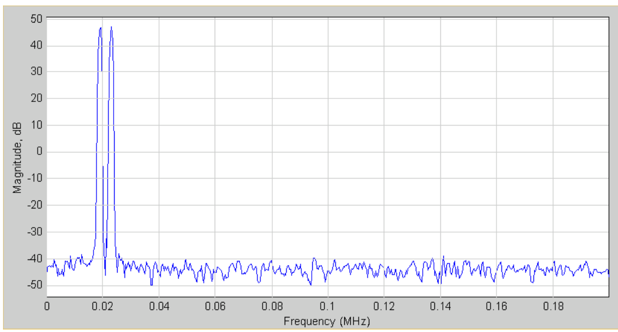

Fig. 9. Two-tone test with predistorsion

From fig.9 is clearly visible the effect of the predistorter, it's also clear a very little residual quantization noise floor.

REFERENCES

[1]. A. Lothia, P. A. Goud, C. G. Englefield, "Power Amplifier Linearization Using Cubic Spline Interpolation," Proceedings of the 43th IEEE Vehicular Technology Conference, May 1993, pp.676-679.

[2]. D. E. Johnson, "Adaptive digital predistorsion with application for LMDS systems," Master of Science in Electrical Engineering, Virginia State University, Aug.2000.

[3]. Gang Feng, "Data Smoothing by Cubic Spline Filters", IEEE Transactions on Signal Processing, Vol. 46, N° 10, October 1998.

[4]. A. Saleh, "Frequency-Independent and Frequency Dependent Non Linear Models of TWT Amplifiers," IEEE Transactions on Communications, Vol. COM-29, No.11, Nov.1981, pp.1715-1720.

[5]. James K.Cavers, "Optimum Table Spacing in Predistorting Amplifier Linearizers", IEEE Transactions on Vehicular Technology, Vol. 48, N°5, September 1999

[6]. MATLAB® Release 11 User's Guide and Simulink® 3.0 User's Guide

[7]. Y. Nagata, "Linear amplification technique for digital mobile communications," 39th IEEE Vehicular Technology Conference, 1989, vol.1, pp.159-164.

[8]. A.V.Oppenheim, R.W.Schafer, "Discrete-Time Signal Processing", Prentice Hall, 1999