Digital Pre-Distortion Implemented Using FPGA

Declan Byrne, Ronan Farrell, Sidath Madhuwantha, John Dooley
Maynooth University
Maynooth, Ireland
(declan.byrne, ronan.farrell, sidath.madhuwantha, john.dooley)@mu.ie

Miriam Leeser
Northeastern University
Boston, USA
mel@ece.neu.edu

***Abstract*—Massive-MIMO and beamforming techniques have long been proposed as a means of increasing cellular network capacity and improving signal to interference ratio performance. The implementation of such systems requires a large number of signal transmission paths. To realize this, a distributed array of power amplifiers (PAs) is likely to be needed. These PAs will possess similar, but unique, characteristics which will alter over time independently due to temperature drift and component ageing. In order to operate all PAs in both a linear and efficient fashion a linearisation technique, such as Digital Pre-Distortion (DPD), must be used. DPD algorithms benefit from reconfigurability, low latency and power efficiency, all traits associated with Field Programmable Gate Arrays (FPGAs). This demonstration shows how an FPGA, specifically a ZYNQ System on a Chip (SoC), can be used in tandem with a transceiver board, the FMCOMMS2, to implement a DPD system.**

***Keywords*-digital pre-distortion, polynomial evaluation, 5G, FPGA, DSP.**

I. Introduction

The Power Amplifier, used to amplify Radio Frequency (RF) signals in wireless transmitters, is often one of the main consumers of power in the transmitter. Typically, the PA must be operated in a nonlinear region of operation in order to run it efficiently. This in turn causes the PA output to become distorted. Digital Pre-Distortion is a PA linearisation technique that manipulates the PA input in an opposite but equal manner to the nonlinear behaviour of the PA. The resulting PA output is linearised. Digital Pre-Distortion involves two main stages: The first is deciding on the function used to pre-distort and training coefficients for that function (post-distorter). The second stage is where the coefficients are applied to the function to pre-distort the PA input signal (pre-distorter).

The function employed to perform DPD can vary widely in terms of the digital structure, type of calculations, the number of coefficients and methods used for training the coefficients. All of these depend on the type of PA being linearised and/or baseband hardware constraints. Massive MIMO introduces the concept of large arrays of PAs used in signal transmission, whose characteristics will be intially unique and also change over time independently. For these reasons DPD is not ideally implemented on fixed baseband hardware. To maintain high performance in the event of the PA behaviour varying, reconfigurable hardware solutions, such as FPGAs, are an attractive solution [1].

This demonstration showcases how a DPD system can be implemented using a ZedBoard (ZYNQ SoC) and MATLAB [2]. The ZedBoard is connected to an FMCOMMS2 transceiver board [3] which allows the ZYNQ to transmit a signal to a PA and receive its output, which is necessary to perform DPD.

II. Background

As mentioned previously, DPD is performed over two distinct stages, the post-distorter and the pre-distorter. These blocks can be connected to each other in various ways, such as Direct or Indirect Learning. An example of an Indirect Learning DPD Architecture can be seen in Figure (1).

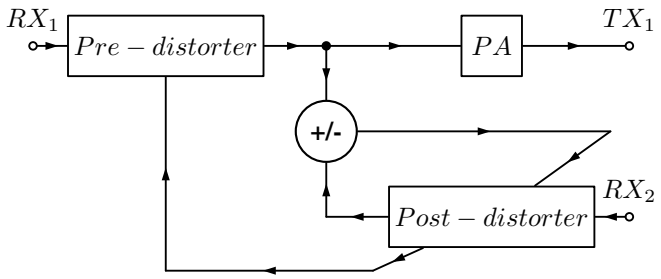

Fig. 1: Indirect Learning Architecture

The first step in performing DPD is training a set of DPD coefficients in the post-distorter block. Typically optimisation algorithms, such as the Least Squares (LS), are employed to obtain and then refine a set of DPD coefficients. The LS process requires a large number of measured PA input and output samples. These are acquired and stored in vectors x and y respectively, each of length n, the amount of samples taken.

An n by m matrix A can now be obtained using the x vector data in association with a digital pre-distortion structure. The matrix length n is equal to the amount of samples taken and the matrix width m is equal to the amount of DPD coefficients to be calculated. This figure is determined by which DPD structure is employed. Each structure caters for different or a combination of different nonlinear PA characteristics and are each of varying performance. One of the most commonly used structures, due to its compact size and relative high performance, is the Memory Polynomial

(MP) [4]. The MP structure, shown in (1), can linearise PAs exhibiting memory effects.

$$y_n = \sum_{m=0}^{M} \sum_{k=0}^{K} a_{mk} \cdot x(n-m) \cdot |x(n-m)|^{k-1}. \quad (1)$$

Parameters M and K are the order of nonlinearity and amount of memory taps used respectively. These parameters are altered for each PA to account for their unique nonlinear characteristics and determine the amount of DPD coefficients required for linearisation. Over time these parameters will have to be adjusted as PA characteristics drift, which can change the amount of DPD coefficients used. The DPD coefficients, contained in the m element long vector b, can be obtained using the matrix A and the vector y as shown in (2).

$$Ab = y \quad (2)$$

As the number of samples, n, is usually larger than the number of coefficients, m, the matrix A is said to be overdetermined. Therefore to solve (2) a Moore-Penrose pseudo inversion technique such as QR decomposition, followed by back substitution, must be used.

Once extracted the DPD coefficients can be applied to the PA input signal, using either Look-Up Tables (LUTs) or pipelining. This function is performed in the pre-distorter block continuously after the initial coefficient training. The output of this function, the pre-distorted signal, is now used as the new PA input signal. The PA output signal is now linearised.

III. Experimental Validation

The hardware testbench, seen in Figure (2), consists of an FMCOMMS2 board, which utilises the AD9361 transceiver, connected to a ZedBoard, which uses the Zynq SoC.

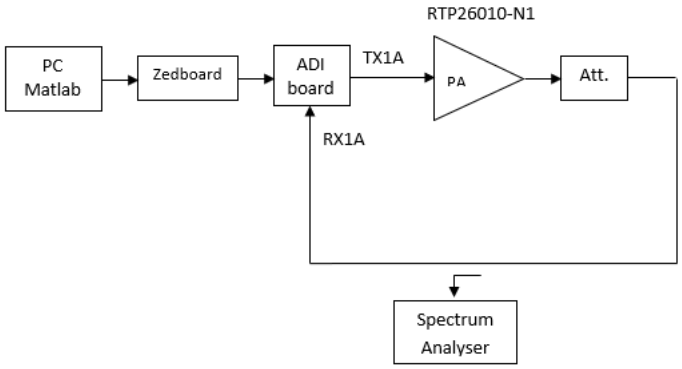

Fig. 2: DPD Testbench

The FMCOMMS2 Tx1 path is connected to the PA input and the PA output is connected to a spectrum analyser for validation. The ZedBoard is connected to a PC via Ethernet, which enables a MATLAB script to control the hardware. DPD in this demonstration is performed using an Indirect Learning architecture, the MP DPD structure and the LS optimisation algorithm using the QR decomposition Moore-Penrose matrix inversion technique. Samples of the PA input and output signals were captured using the FMCOMMS2, ZedBoard and MATLAB. Coefficient training was then conducted offline using MATLAB. The DPD response, alongside the original and distorted signals, can be seen in Figure (3).

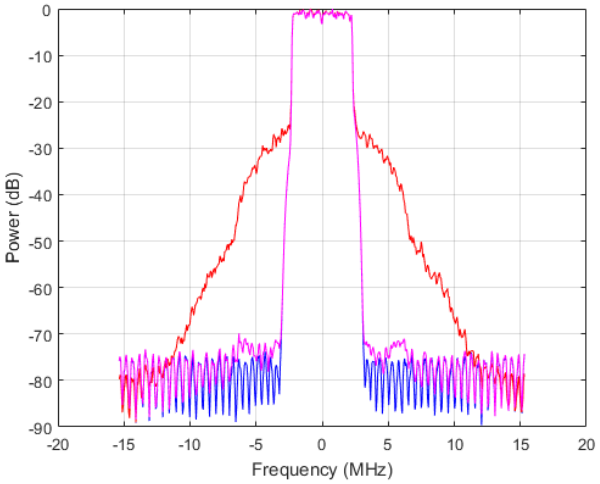

Fig. 3: DPD Response

IV. Conclusions

This demo illustrates how a compact function for digital pre-distortion can be implemented on an FPGA. As the number of signal paths increases the total number of coefficients required to implement DPD on all paths will increase proportionally. As such, a compact model, such as the memory polynomial, can provide a good compromise between computational efficiency for the full array and linearisation performance for each path in an array.

Acknowledgements

This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077.

References

[1] N. Lashkarian and C. Dick, "Fpga implementation of digital predistortion linearizers for wideband power amplifiers," in *Proceeding of the SDR 04 Technical Conference and Product Exposition*, 2004.
[2] Xilinx, "Zynq-7000 all programmable soc data sheet: Overview," Tech. Rep., 2017.
[3] A. Devices, "Ad9361 reference manual," Tech. Rep., 2016.
[4] J. Kim and K. Konstantinou, "Digital predistortion of wideband signals based on power amplifier model with memory," *Electronics Letters*, vol. 37, no. 23, pp. 1417–1418, Nov 2001.