DIGITAL PREDISTORTION EXPERIMENTS WITH MATLAB-CONTROLLED SOFTWARE DEFINED RADIO

Martin Pospíšil
Doctoral Degree Programme (4), FEEC BUT
E-mail: xpospi29@stud.feec.vutbr.cz

Supervised by: Roman Maršálek, Tomáš Gotthans
E-mail: marsaler@feec.vutbr.cz

Abstract: This paper presents the experiments with digital predistortion for power amplifier linearization. It describes the architecture of used measurement setup based on the software defined radio USRP N200 with the focus on our custom MATLAB-based control of the USRP through UDP packets. Measurement results obtained for the 800 MHz power amplifier linearized by memory polynomial predistorter are presented, but the prepared framework to control of USRP in MATLAB can of course be used for experiments not only with such type of predistorter but for wide range of experiments where the USRP serves as the receiving/transmitting unit.

1. INTRODUCTION

Signals with non-constant envelope are often used in current wireless communication systems in order to fulfill demands for high data rates. In order to gain better Power Amplifier (PA) efficiency, the amplifiers are often used close to the saturation point. This leads to significant nonlinear distortion of the transmitted signals with undesirable effects on the increased bit error rate and spurious transmission into adjacent channels.

One of the solutions to combat the power amplifier effects is a digital predistortion (DPD) of PA, usually performed in the complex baseband domain. The first experiments with digital predistortion date back to 1980's [1]. DPD principle is based on the artificial distortion of the transmitted signal with a nonlinear function modeling the inverse of PA. The inverse function can be implemented in several ways differing in the complexity as well as in the performance. The simplest way is based on the Look Up Table [2] or polynomial function approximation. These two approaches cannot compensate for the memory effects in PA, significant especially for the wide-band input signals. In order to count for the memory effects, approaches like memory polynomials, Volterra series or generalized memory polynomials have been proposed. DPD is often adaptive - the inverse function adapts to PA changes caused by changing carrier frequency, power or temperature variations. DPD performance can be affected by RF impairments, e.g. IQ imbalances that can be corrected for, [3].

Currently chips for DPD [4] or specialized IP cores [5] are also available. Unfortunately, these solutions are not easy to use in the commercial software defined radios with small FPGA devices. To facilitate the experiments with DPD, it is desirable to be able to perform the algorithmic part in the simulation software (e.g. MATLAB) used in connection to available RF equipment.

2. PRINCIPLE OF ADAPTIVE DPD

The basic principle of adaptive DPD s is shown in left part of Figure 1. We consider the baseband DPD, where a transmitted baseband signal is upconverted to desired carrier frequency. A part of the transmitted signal is fed back to the baseband through a directional coupler to have a reference signal for DPD adaptation. Such situation perfectly matches with a device available at a workplace - software defined radio USRP N200 populated with WBX front-end.

2.1. USED PREDISTORTER AND ADAPTATION METHOD

Several ways to adapt DPD block exists, the main two being a direct and an indirect method. In our example experiment presented below, the adaptation of DPD has been done using indirect-learning Recursive Least Squares (RLS) method, according to block schematic on right part of Fig. 1. The blocks of N data samples from the PA input and output are transferred from USRP to MATLAB, delay compensated [6] and DPD coefficients are computed from these blocks. Then the original data signal is processed by DPD and predistorted signal is uploaded to USRP for transmission.

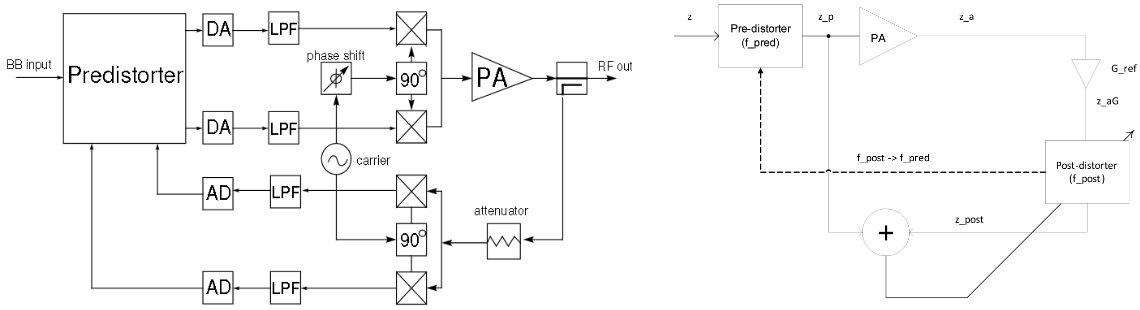


Figure 1: Digital predistortion principle (left) and indirect learning architecture (right)

From among a variety of predistorting functions, we have chosen Memory Polynomials with nonlinearity order K and memory depth P due to its good compromise between the linearization performance and complexity. Given DPD coefficients f_{kp}, the expression for signal $z_p(n)$ at the output of DPD driven by the original data signal $z(n)$ at time instant n is then:

$$z_p(n) = \sum_{k=0}^{K} \sum_{p=0}^{P} f_{kp} z(n-p) \left| z(n-p) \right|^k \tag{1}$$

3. USRP AS A TRANSMITTING/RECEIVING DEVICE

The use of USRP from Ettus company (now National Instruments) is very popular among researchers in both academia and industry. In many cases, the USRP is used in connection with a host PC, and the more advanced data processing is done in PC using either GNU radio, GNU radio companion graphical environment, Simulink or LabView as in [7].

The main board of USRP N200 is equipped with the Xilinx Spartan 3A-DSP 1800 device and dual 14 bits A/D converters with 100 MSa./s. sampling rate. The USRP can be used in connection with various front-end modules ranging from DC to 6 GHz frequency bands. In the USRP, data from the antenna are received by the front-end module, converted to digital domain by the A/D converters and subsequently Digitally Down-Converted (DDC) to baseband. In the receiver, the user can access data from the front-end, input to the DDC, output of the DDC or baseband data. The transmitting chain is very similar. In our experiments we have worked with the final baseband data.

For the communication of the modules in the general USRP N2x0 design, an open source Wishbone bus is used. In this particular case the bus has 32-bit data width and 32-bit address space. There is one wishbone master in USRP design - the ZPU soft processor. The bus is configured to 16 slaves, but some of the slave positions are unused.

3.1. ARCHITECTURE CHANGES

In order to make use of USRP as a MATLAB-controlled device able to continuously transmit a vector of samples that drives the PA and to receive the PA output signal, some changes of the internal FPGA design had to be done. The overall architecture is shown on Fig. 2.

Two Block-RAMs (for transmitting-Tx and receiving-Rx path) are implemented in FPGA, both having a length of 4096 IQ samples. A multiplexer chooses between normal TX stream (i.e., from host PC) or data from TX BRAM.In this case, only TX BRAM is used as a transmission source. A counter is used to address port B on both BRAMs and is clocked with baseband sample rate. The counter operates in two modes: free-running or single pass. In the free-running mode, the signal from TX RAM is continuously repeated. It is used as normal output, for the measurements on the spectrum analyzer. The counter can be switched to other mode seamlessly, with no breaks in the transmitted signal. In the other mode, single pass, the counter stops at the end address. This ensures the received signal in the RX BRAM is one continuous block of data ready to be downloaded to MATLAB in a host PC.

On both BRAMs, the port A is connected to the Wishbone interface. This interface implements wishbone slave and it is connected to the normally unused position #4. Registers are mapped to the ZPU memory space starting with a base address of 0x5800. Data from BRAMs are accessible sample by sample only. To access the data, first the address register is written, then read or write operation is performed.

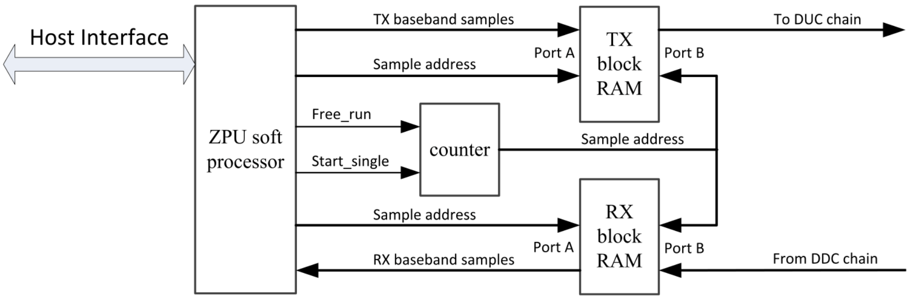


Figure 2: Block diagram of USRP N200 FPGA modifications that allow us to transmit and receive samples from MATLAB

3.2. MATLAB CONTROL OF USRP

Instead of using the standard communication with USRP using Simulink blocks, GNU radio or LabView software, we have concentrated on the USRP control from MATLAB using UDP packet operations in order to have the highest flexibility of control. For integration to MATLAB enviroment, UDP control library was written. It uses `java.net.DatagramSocket` and `java.net.DatagramPacket` classes for direct socket access.

In order to use the dynamic range of USRPs onboard A/D converters effectively and to be able to drive the power amplifier into the desired operational point, we needed to add a functionality to control the attenuators in the receiving and transmitting path. For direct control of the front-end attenuators from ZPU, a dedicated function needed to be written as this is normally only available from the standard UHD (Universal Hardware Driver) driver of USRP.

Three basic functions of our control library are thus now available, namely fill_bram, read_bram and control_set. Control_set function sets both attenuators and the control register. The data moving functions perform data type conversion, scaling and endian conversion. A new UDP listener is registered in the network stack on port 49180. The listen function implements a simple command protocol. First byte is the command number. Other bytes or packets carry data.

The stock UHD driver is still used to set-up the USRP. This is done by modified example application, written in C++. Data sent or received by streaming interface are discarded. UHD is used to set up sample rate (in our example setup set to 6,25 Msps) and carrier frequency (here 800 MHz).

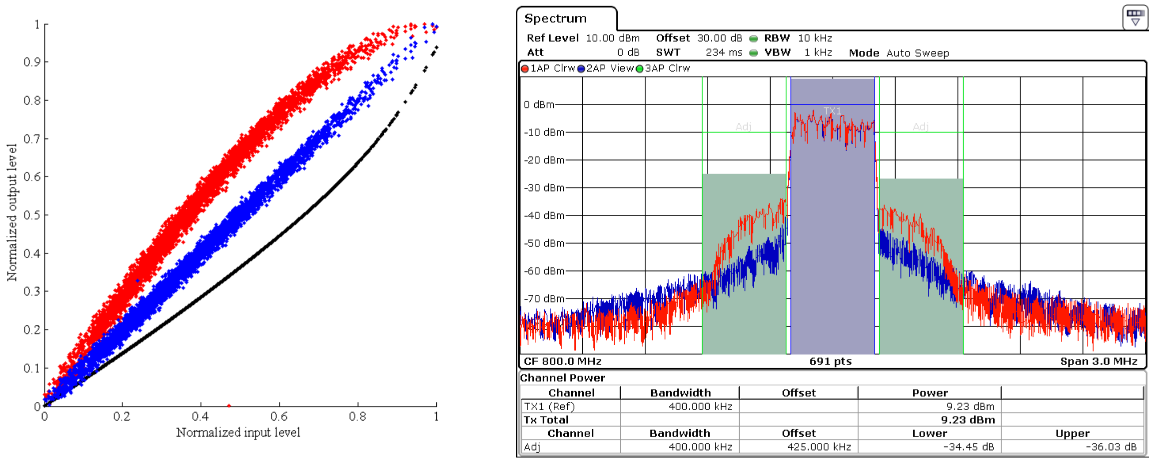
4. MEASUREMENT SETUP AND RESULTS

During the experiment, USRP N200 was populated with WBX front-end module, whose output is amplified with the PA working in nonlinear region. At the PA output a directional coupler CP0603GN with -34dB coupling from AVX company has been used to get the attenuated part of PA output fed back to the USRP RX chain. The PA output was also monitored using a Rohde&Schwarz FSVR real-time spectrum and vector signal analyzer to get the output spectrum and to evaluate the adjacent channel protection ratio (ACPR) performance. The most important used components of WBX front-end and setup itself are summarized in Table 1.

The PA has been fed up with QPSK signal. The measured AM/AM characteristics are shown on Fig. 3. You can observe a dispersion of points in the characteristics caused possibly by the PA memory effects or incoherency of RX and TX oscillators. With the use of simple polynomial predistorter (memory depth P=0, nonlinearity order K=7), the ACPR improvement of around 12 dB's has been achieved as demonstrated on the spectral plots on right part of Fig. 3. Although such a performance is far from the theoretical values achieved in the pure simulations (20-30 dB) and marketing materials of DPD manufacturers, it has to be stated that till now we did not pay attention to correction of other RF imperfections such as IQ imbalances. As shown in Fig. 4, the temperature-dependent IQ imbalances are present in the used front-end and have to be compensated for to assure the full DPD functionality.

Transmit path		Receive path	
AD9777	Dual channel DAC	HMC174MS8	MMIC RF switch
ADL5385	Mixer	MGA62563	MMIC low-noise amplifier
HMC472LP4	Programmable attenuator	HMC472LP4	Programmable attenuator
GVA-84+	Amplifier	MGA82563	MMIC amplifier
HMC174MS8	MMIC RF switch	ADL5387	Mixer
Tesla 217.33	Power amplifier (to be linearized)	ADA4937-2	IQ baseband amplifiers
		ADS62P44	Dual channel ADC

Table 1: Active devices (chips) in DPD chain



Ref Level 10.00 dBm Offset 30.00 dB RBW 10 kHz
Att 0 dB SWT 234 ms VBW 1 kHz Mode Auto Sweep

Channel Power

Channel	Bandwidth	Offset	Power	
TX1 (Ref)	400.000 kHz		9.23 dBm	
Tx Total			**9.23 dBm**	
Channel	**Bandwidth**	**Offset**	**Lower**	**Upper**
Adj	400.000 kHz	425.000 kHz	-34.45 dB	-36.03 dB

Figure 3: Left: AM/AM characteristics of PA (red), DPD (black) and model of DPD+PA (blue), Right: Measured spectra at PA output without (red) and with (blue) linearization

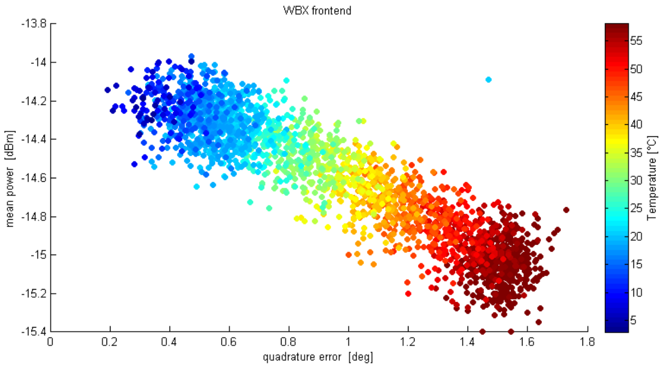


Figure 4: Transmit path temperature dependence.

5. CONCLUSION

The MATLAB-based control of USRP N200 software defined radio has been prepared. It allows to set the parameters such as Tx/Rx gain of this SDR, to transfer a signal created in MATLAB to internal memory of FPGA inside USRP for future playback, and to download received signal from FPGA internal memory back to MATLAB. Created MATLAB framework has been used to experiments with digital predistortion of power amplifiers based on memory polynomials. The improvement of up to 12 dB in term of ACPR has been achieved with the selected power ampliifier working at 800 MHz. The performance is significantly affected by the receiver impairments and further work would be to employ the corresponding compensation methods to further improve the predistorter's performance.

ACKNOWLEDGEMENT

The presented research has been supported by the BUT internal projects FEKT-S-14-2281 and FEKT-S-14-2177.

REFERENCES

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

[2] Marsalek, R., Pospisil, M., Evaluation of digital predistortion using the USRP N200 software defined radio transceiver, In *Proceedings of NORCHIP 2014*, 2014, Tampere, Finland, p. 1-4.

[3] Antilla, L. et al., Recursive learning-based joint digital predistorter for power amplifier and I/Q modulator impairments, *International Journal of Microwave and Wireless Technologies*, 2010, 2(2), p. 173-182.

[4] Intersil, *Adaptive Predistortion Unsing the ISL5239*, Application Note, September 2002

[5] Xilinx, *LogiCORE IP Digital Pre-Distortion v5.0*, August 15, 2011.

[6] Gotthans, T.; Baudoin, G.; Mbaye, A., Influence of delay mismatch on digital predistortion for power amplifiers, in *20th International Conference Mixed Design of Integrated Circuits and Systems (MIXDES)*, 2013, p. 490-493.

[7] Dohl, J.; Fettweis, G., Evaluation of estimation and mitigation algorithms for nonlinearly distorted OFDM signals on a SDR platform, in *13th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2012 , p. 535-539.