Project acronym:	**SACRA**
Project title:	Spectrum and Energy Efficiency through multi-band Cognitive Radio
Project number:	European Commission – 249060
Call identifier:	FP7-ICT-2007-1.1
Start date of project:	01/01/2010 **Duration:** 36 months

Document reference number:	**D5.2**
Document title:	Report on SACRA embedded software library, RF/BB co-design, RF/BB interface, functional and performance validations
Version:	1.0
Due date of document:	Month M24 (12/2011)
Actual submission date:	16 January 2012
Lead beneficiary:	Konstantinos Chatzikokolakis(UoA)
Participants:	Konstantinos Chatzikokolakis(UoA), Spyridon Evaggelatos(UoA), Renaud Pacalet (IT), Sylvain Traverso (TCS).

Project co-funded by the European Commission within the 7th Framework Programme		
DISSEMINATION LEVEL		
PU	Public	**X**
PCA	Public with confidential annex	
CO	Confidential, only for members of the consortium (including Commission Services)	

EXECUTIVE SUMMARY

Energy efficiency and flexibility in the use of spectrum resources are two major research directions towards the development of future wireless communication technologies. The combination of innovative approaches on RF front-end and base band components design with advanced cognitive radio algorithms are the major outcomes that SACRA project plans to deliver and integrate into a demonstrator platform; the design of a flexible base band is one of the key outcomes of SACRA project. The proposed methodology in terms of this WP will be used as input for the demonstration of selected SACRA use cases on a hardware/software platform.

The purpose of Deliverable D5.2, entitled "*Report on SACRA embedded software library, RF/BB co-design, RF/BB interface, functional and performance validations*" is to provide a detailed analysis of the advancements on the development of the software design suite that will be delivered in WP6. Towards this direction, D5.2 covers all the expressed requirements of the SACRA project, as detailed in the D5.1 deliverable, "*Preliminary report on the system requirements, application modelling and embedded software library*". Furthermore, the functional requirements of the radio access technologies considered in the use cases and scenarios as described in D1.1b, ("*SACRA scenario study and system definition*") are taken into consideration. Finally, the validation trial definition and the analysis on integration specification, as thoroughly described in D6.1 ("Validation trial definition") and D6.2 ("Integration Specification") accordingly, have been used as inputs in D5.2.

Deliverable D5.2 presents the embedded software library developed in WP5 and distributed under the terms of a free software licence. The library is the first hardware abstraction level used by programmers of the baseband processor. The principles and guidelines used during its design are presented. **Important note**: most of the report about this topic is delivered as two separate appendices, one with a complete functional description of the library and the other with the complete documentation of the Application Programming Interface (API). For a complete overview please have a look at these two appendices.

Deliverable D5.2 also contains a novel study that highlights the improvement of the linearity performance of a real Power Amplifier (PA) for LTE compliant signals. This approach is based on a combination of a PAPR reduction technique and a Digital PreDistortion (DPD) technique for PA linearization so as to avoid PA non linearities with the OFDM signal. The PAPR reduction technique chosen is the Active Cancellation technique introduced in [9] for a good comprise between complexity and performance. This approach is a WP4/WP5 algorithm that improves the PA power efficiency which is the main power sink of a modem. The proposed DPD technique is enhanced with an indirect learning technique for the computation of the DPD. Description and simulation results are given for the two existing categories of memory polynomial that digital predistortion can be based on.

Deliverable 5.2 briefly presents the system environment of the baseband processor and the different interfaces. As these aspects are already discussed in deep details in deliverables D4.1 and D4.2 and are not repeated here.

Deliverable 5.2 explains the different strategies used to validate the applications developed on the baseband processor on top of the embedded software library. Validations cover the functionality, the performance and the energy estimation. Examples and results are shown for sensing algorithms.

CONTENTS

1 INTRODUCTION

1.1 PURPOSE OF THE DOCUMENT

This document is the deliverable D5.2 entitled “Report on SACRA embedded software library, RF/BB co-design, RF/BB interface, functional and performance validations” of the SACRA project. In section 2 a full description of the software library that is implemented on top of SACRA terminal is given. The software components provide synchronization, data transfer, local and global scheduling and memory management primitives. A detailed analysis on digital/analogue predistortion techniques and digital only PAPR reduction algorithm for OFDMA is given in section 3. In section 4 the interfaces between the Radio Frequency front ends and the base band digital processing units are reported. Section 5 includes validation outcomes of the proposed model, against functional and performance requirements and finally, section 6 concludes the document.

1.2 DOCUMENT RELATION TO TASKS

This document is related to tasks WP5.2, WP5.3 and WP5.5 within the WP5 context. Deliverable D5.2 involves the progress of work in these three tasks during the second year of the project and provides inputs for the upcoming deliverable D5.3. Milestone M5.3 served as a structuring milestone for the description of the embedded software library and milestone M5.4 provided the simulation framework for energy validation. The embedded software library is described in section 2 and validation outcomes based on the simulation framework are given in section 5.

The study of the digital predistortion is at the frontier between the RF and the baseband and is then conducted in the framework of WP4 and WP5. It has been decided to report all algorithmic studies and performance result in D5.2 to have a complete overview of the global co-design in a single document.

1.3 DOCUMENT VERSIONS SHEET

Version	Date	Description, modifications, authors
0.1	28/10/2011	Initial ToC (Konstantinos), IT contribution on section 2
0.2	29/11/2011	UoA contribution on section 5.2(Konstantinos)
0.3	01/12/2011	UoA contribution on section 5.2.2(Konstantinos)
0.4	01/12/2011	Additions in section 5.2 (Renaud)
0.5	02/12/2011	IT contribution in section 5.1 (Renaud)
0.6	02/12/2011	TCF contribution in section 3 (Sylvain)
0.7	02/12/2011	TCF contribution in section 3 (Sylvain)
0.8	05/12/2011	Introduction added, minor modifications in section 5.2.2 (Konstantinos)
0.9	06/12/2011	Executive summary added. Minor editorial modifications in section 3. (Konstantinos)
0.17	09/12/2011	Document ready for internal review
0.19	05/01/2012	Corrections after review Correction of some equations. Add conclusions in section 6.
0.21	10/01/2012	Corrections after review

0.22	10/01/2012	Corrections after Abdel's review and Renaud's comments (Sylvain)
1.0	16/01/2012	Final version is ready. (Konstantinos)

2 EMBEDDED SOFTWARE LIBRARY DESCRIPTION

This chapter first presents the principles governing the design of the embedded software library, the main requirements, the specifications that have been derived from the requirements, the organization of the library, its content and its Application Programming Interface. The two latter are delivered as two separate appendices:

- *ExpressMIMO user guide*, providing the detailed programmer's view of the available digital signal processing functions. The functions are accurately described along with how to launch a processing, provide input data samples and retrieve the output results. After reading this document, a programmer should be able to start designing applications at a very low level, directly on top of the hardware baseband processor.
- *ExpressMIMO baseband library reference manual*, listing and documenting the large collection of macros and functions programmers should use to access the digital signal processing functions and to manage data transfers between processing units. These functions are the hardware abstraction layer of the baseband processor. Thanks to them, application programmers will work at a much higher abstraction level, will safely ignore most hardware-related details and will improve their productivity, the performance and the quality of their applications.

These two documents will be updated every time a new processing block is released or an existing one is enhanced. For a complete overview of the embedded software library please have a look at these two appendices.

2.1 MAIN SPECIFICATION CRITERIA

The embedded software library is a building block of the design framework for baseband applications. It can be seen as the hardware abstraction layer between the target digital baseband processor and the application programmer. One of its main goals is thus to hide as much as possible low-level hardware details to application designers. Of course, some low-level hardware aspects cannot be completely hidden - at least until powerful synthesis tools become available and automatically handle them - like memory management and scheduling of shared resources. Several important requirements, dictated by prior knowledge of the field and SACRA specificities, were considered while specifying the library:

SACRA indicators

The library shall account for the four SACRA indicators: *spectrum occupancy*, *spectrum efficiency*, *energy efficiency* and *reduction of the number of components*; even if the baseband software cannot alone solve these issues, it shall contribute as much as possible.

SACRA requirements

The library shall cover all the expressed requirements of the SACRA project, as detailed in the D5.1 deliverable, "*Preliminary report on the system requirements, application modelling and embedded software library*", plus the functional requirements of the radio access technologies considered in the use cases, in the scenarios and in the WP6 "*Integration, validation and trials*" activities.

Versatility

The library shall be usable for very different purposes, from the design of pure algorithmic, untimed, models, intended for algorithmic performance evaluations by simulations on a regular desktop Personal Computer (PC), to design of the complete application, running on the hardware

platform, with fine grain memory management and shared resources scheduling; the path between these two extreme purposes shall be continuous and smooth.

The library shall offer a mean to access every primitive digital processing feature of the target platform but also to design higher level sequences of operations - like synchronization or channel estimation - and reuse them.

The library shall be usable for manual design and for semi- or fully-automated code generation.

The library shall be usable for functional and performance validations.

User friendliness

The library shall be close from what baseband application designers are accustomed to.

2.2 DESIGN PRINCIPLES

This section details how the preceding requirements are translated into design principles.

SACRA indicators

Spectrum occupancy and efficiency are not directly dealt with by the software library but rather by the use the SACRA project makes of it. As a consequence these two indicators translated into functional requirements: the library has been enhanced whenever needed in order to support the proposed algorithms.

Energy efficiency has been taken into account in two different ways. On the one hand, energy efficiency, at terminal, base station or system level, is an outcome of the proposed SACRA algorithms and strategies. The library supports them as for spectrum occupancy and efficiency by being functionally compatible. On the other hand, the baseband digital processing also contributes *per se* to the overall energy consumption. In order to minimize this contribution, the baseband processor has been designed as a collection of flexible DSP (Digital Signal Processing) units, each being specialized, not in a Radio Access Technology (RAT) but in a class of processing that can be encountered in many RATs. This Software Defined Radio (SDR) approach differs from more classical approaches where each RAT is dedicated a specialized complete baseband processor. In both cases the operation point (clock frequency and supply voltage) of each processor can be dynamically adapted to the workload in order to save as much energy as possible[1]. In the SACRA architecture, this optimization is done across all RATs, sensing techniques, and other baseband services currently running because, for each class of processing, they all use the same shared DSP unit. The savings are thus much closer from the global optimum than with separate baseband processors where the optimization is local to each RAT. The software library supports this by providing means to manage and schedule these shared DSP units in a context where multiple RATs, sensing techniques and other DSP activities run concurrently.

Finally, the reduction of the number of components is achieved by this same baseband architecture which, being capable of handling all RATs and other DSP needs, allows to use the same single processor instead of a collection of specialized ones with almost identical variants of the same basic processing primitives.

SACRA requirements

[1] Of course, in the FPGA based prototyping platforms used in the SACRA project, these dynamic voltage and frequency scaling strategies cannot be deployed. It is only on a final industrial product that they could be. This is the reason why, in SACRA, power consumption of the baseband is estimated by simulation and not by measurements.

The functional requirements have been collected, synthesized and analyzed (deliverable D5.1 "*Preliminary report on the system requirements, application modelling and embedded software library*"). Apart Digital Pre-Distortion (DPD) and Peak to Average Power Ratio (PAPR) reduction, which is discussed in chapter 3 (Rf/BB Co-design), they all have been taken into account and led to functional improvements. The baseband processor and the software library support all expressed requirements.

Versatility

In order to allow pure untimed algorithmic simulations, the library offers a synchronous[2] version. Synchronous programmers ignore all synchronization and parallelisation primitives, along with the underlying operating system, MutekH [38]. They use the functional Application Programming Interface (API), that is, the functions used to launch hardware-accelerated processing, and the data transfers API, that is, the functions used to move data around the baseband processor. Their applications are compiled for a regular desktop PC and linked against an "*emulation*" binary version of the library, designed in C++. This version emulates the processing as it takes place in the real hardware. It is bit accurate but without any notion of processing time. This is thus a way for programmers to accurately evaluate the implemented function on a desktop PC. The calls to the library functions are sequential, as in a classical programming paradigm.

Even if this first kind of application is intended for PC simulations, it can also be cross-compiled and linked against a second binary version of the library in order to run on the hardware platform. The source code is exactly the same, only the compiling and linking tool chain differs. Operations are still purely sequential (we do not take any benefit from the intrinsic parallelism of the baseband processor) but instead of being emulated in software, they are hardware-accelerated by their associated DSP units. Similarly, software memory copies are translated into Direct Memory Accesses (DMA) transfers performed by embedded DMA engines. This possibility has been used to validate the bit-accuracy of the software emulation against the actual hardware: the same application is run on both targets and the results compared. A mismatch indicates a bug either in the emulated version or in the hardware or both. Several difficult to catch bugs were discovered this way.

Once this synchronous version is fully validated on the algorithmic, control flow and memory placement points of view, it can be refined in several steps up to the final version, running on the hardware platform and taking full benefit from its parallelism. The first step consists in inserting parallelization and synchronization primitives exported by the underlying operation system. The goal of this first transform is to express the data dependencies limiting the potential parallelisation. The refined application can again be compiled for a desktop PC and linked against an asynchronous "*emulation*" version of the library. The result is a multi-threaded application, still bit accurate but now parallelized. The benefit is a potential significant speedup factor on a multi-processors / multi-cores desktop PC. The parallelized application can also be validated and dead locks or undesirable race conditions identified. Of course, the target being completely different from the final one, the parallel execution traces differ and there is no guarantee that the obtained traces are a good subset of what could happen in the actual hardware. It is possible that not yet observed bugs are discovered later on the hardware target.

The last refinement step consists in cross-compiling and linking against the full ExpressMIMO binary version of the library. The application can then run on the hardware target. This whole refinement process is really a true refinement because it is the same programmer's source code

[2] Synchronous, in this context, means sequential: operations are executed one after the other, there is no parallelism at all between them. It is the model of computation every programmer knows.

that is reused and transformed along the path. Only the used compilers and library binaries change. Figure 1 illustrates the different uses of the library.

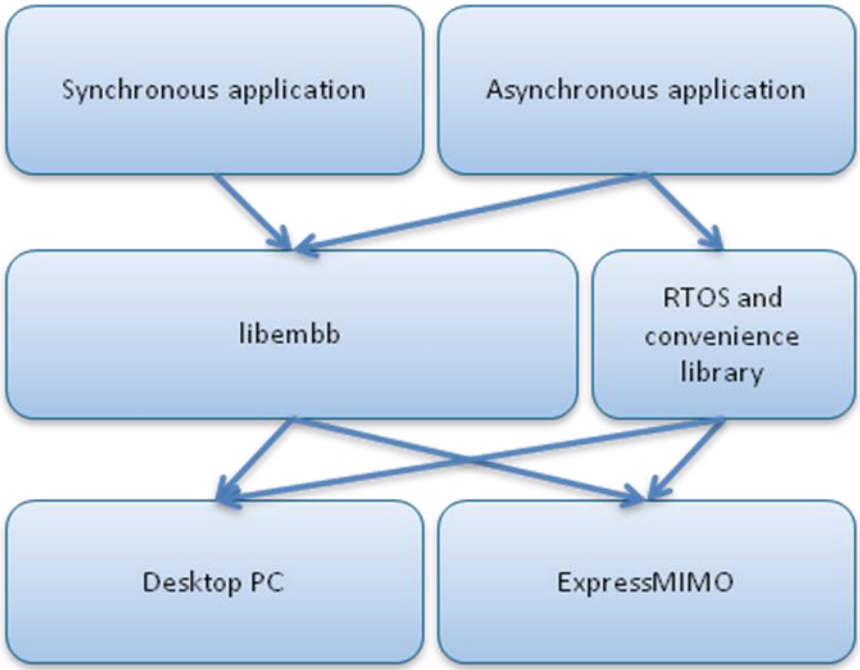


Figure 1: Different uses of the embedded library

This way of designing applications is perfectly compatible with the other versatility requirements:

Frequently encountered processing sequences (e.g. channel estimation) can be coded once and reused as a macro every time it is needed. When such a macro relies on a single DSP unit it can even take benefit from the local micro-controller embedded in each unit[3]. The macro is compiled for the micro-controller, the binary image is loaded in the unit's memory space and the micro-controller chains the operations and data transfers without the main micro-controller (a 32-bits Sparc processor) being involved. The rate of interrupts reaching the main micro-controller is reduced and its load is thus also reduced.

The experimental Unified Modeling Language (UML) -based design framework uses the embedded library as its basic building block to automatically translate UML diagrams to application software source code. The library is thus perfectly usable for semi- or full automatic software generation. Each primitive is accurately specified and documented and its side effects on the hardware is unambiguously known. The main difficulties related to automated software generation are the memory management and the scheduling of shared resources (storage, communication and processing). Until efficient synthesizers are available and capable of solving these issues the source UML diagrams are decorated with manual annotations. These clues are used by the software generators to handle memory management and scheduling.

Finally, as explained above, the same library is used for functional validation thanks to its different versions, either in synchronous or asynchronous mode, either on a desktop PC or on the hardware target. Performance (speed, energy) can be explored either on the hardware target (for speed) or by simulation in the adapted SuperESCalar Simulator (SESC) environment (for speed and energy),

[3] This is not yet supported by the library but will be in a future version.

as will be explained in chapter 5. An intermediate validation platform will also be designed[4] based on the "*emulation*" version of the library: a SystemC (a set of C++ classes dedicated to high level hardware modeling) model of the baseband processor will allow bit-accurate, cycle-approximate simulations.

User friendliness

The API of the embedded software library is 100% ANSI C and its design is oriented towards ease of use. It offers a large, comprehensive and documented collection of functions to setup processing and data transfer parameters. Launching a digital signal processing is as simple as:

```
FEP_CONTEXT fctx;

...

fep_start(&fctx);
```

The structure is very close from other C-based digital signal processing libraries. It also resembles what a Matlab progremmer uses. The algorithmic description should thus be rather straightforward for programmers with DSP programming background. The asynchronous primitives used to control the parallel execution are very classical primitives and any parallel programmer should immediately recognize them:

```
exit_status = fep_wait(&fctx);
```

for instance, is the way to synchronize with the end of the ongoing processing in the FEP (Front End Processor) DSP unit and to get its exit status.

Finally, the ongoing work on the UML-based design framework will offer another way to design applications, a bit like Matlab-Simulink would do, and generate code.

[4] This activity is out of the scope of the SACRA project but will be used if its results are available on time.

3 RF/BB CO-DESIGN

This chapter presents the current status of a novel study that highlights the improvement of the linearity performance of a real Power Amplifier (PA) for LTE compliant signals. This approach is based on a combination of a PAPR reduction technique and a Digital PreDistortion (DPD) technique for PA linearization so as to avoid PA non linearities with the OFDM signal. The PAPR reduction technique chosen is the Active Cancellation technique introduced in [9] for a good comprise between complexity and performance. This approach is a WP4/WP5 algorithm that improves the PA power efficiency which is the main power sink of a modem. The proposed DPD technique is enhanced with an indirect learning technique for the computation of the DPD. Description and simulation results are given for the two existing categories of memory polynomial that digital predistortion can be based on.

3.1 GENERAL IDEA

ENERGY EFFICIENCY

Power Amplifier (PA) hardware non linearity is a well-known problem in all transmitter systems. They cause in-band and out-of-band degradations, which respectively result in received bit error and in adjacent channel pollution. The purpose of our study in SACRA is to improve the linearity performance of a real PA for LTE compliant signals.

Constant envelope signals are not impacted by the PA non linearity, but non-constant envelope signals, and especially signals with high power distribution like OFDM, create linearity damages. OFDM appears as an increasingly popular multi-carrier modulation technique due to its efficiency and robustness under multi-path fading channel as explained in [1]. However one drawback of OFDM is its high dynamic, making this modulation very sensitive to hardware PA non linearity [2].

To avoid PA non linearity, many linearization techniques have been studied like feedback [3], feedforward [4] and baseband digital predistortion [5]. Performance, complexity and adaptation capability criterions indicate to use the Digital PreDistortion (DPD) approach.

Moreover in order to decrease OFDM linearity sensitivity, Peak-to-Average-Power-Ratio (PAPR) reduction techniques are used as clipping, signal addition, coding, and selective mapping. An overview of these techniques is presented in [8]. Among all signal addition techniques, the Active Constellation Extension (ACE) technique introduced in [9] has been chosen for a good compromise between performance and complexity. The advantage of ACE is that it does not cause Bit-Error-Rate (BER) degradations instead of other PAPR reduction techniques. In terms of in-band and out-of-band linearity performance, simulations conducted by Ciochina in [10], have shown that the use of ACE only is not sufficient. A linearization technique and a PAPR reduction technique need to be used together so as to avoid PA non linearities with an OFDM signal.

Few papers consider the association of a PAPR reduction technique and DPD [11], [12], [13], [14] and [15], but none deals with the association of ACE technique with baseband DPD for OFDM. In this study, ACE technique and DPD are jointly used to improve both in-band and out-of-band linearity performance. Performance is estimated in terms of Bit Error Rate (BER) and Error-Vector-Magnitude (EVM) for the in-band linearity and in terms of Adjacent- Channel-Power-Ratio (ACPR) for the out-of-band linearity.

In SACRA, we consider terminal which, by definition, shall be low cost, have a small form factor, implying that the analogue components are imperfect and has several impairments. Possible radiofrequency impairments such as Carrier Frequency Offset (CFO), IQ imbalance [6] and DC-offset [7] have a great impact on the DPD linearization capability: if these impairments are not minimized or compensated, then the DPD is almost inefficient. We propose in SACRA a fully digital

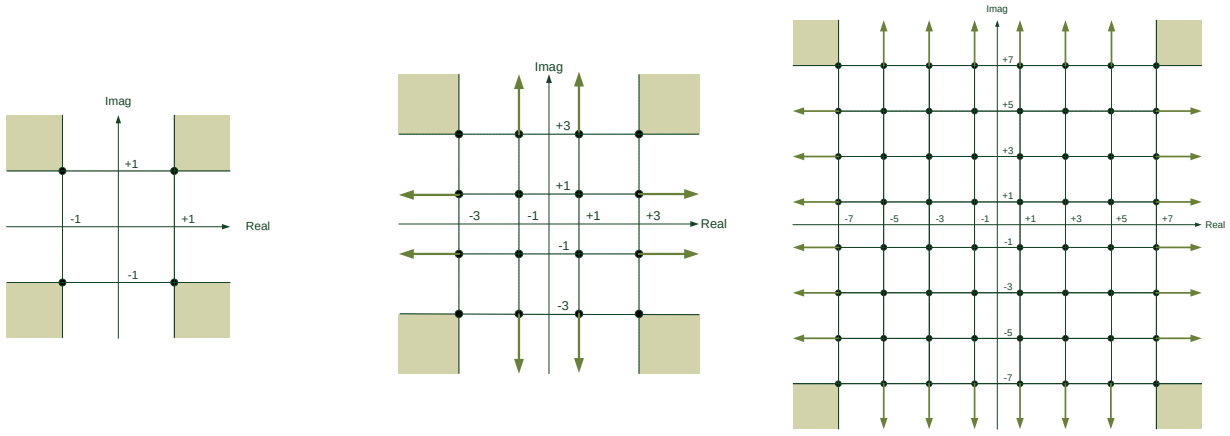
approach that consists in estimating and pre-compensating the CFO, IQ imbalance and DC-offset in baseband.

Section 3.2 presents the ACE algorithm and the simulation results for QPSK, 16QAM and 64QAM. In section 3.3, we describe the DPD approach for SACRA. Our approach consists in first estimating and compensating RF impairments, and then estimating and applying DPD. Several algorithms dedicated to CFO, IQ imbalance and DC-Offset estimations are proposed and described in sections 3.3.1.1 and 3.3.1.2. Then the strategy for DPD itself is addressed in section 3.3.1.3, 3.3.1.4 and 3.3.1.5. The setting of the RF impairments algorithm and DPD parameters is presented in section 3.3.2 thanks to simulations. Global performance assessment of in band and out of band linearity is addressed in 3.4. Finally, conclusions are drawn in section 3.5.

3.2 DIGITAL PAPR REDUCTION ALGORITHM

3.2.1 Active Constellation Extension Presentation

The ACE technique has been introduced by Krongold and Jones in [9]. It uses nonbijective constellations to reduce the PAPR by encoding the data symbols. ACE allows the modification of constellation points on each subcarrier of the OFDM signal, and more specifically extension of the constellation, which results in a PAPR reduction. ACE does not cause any BER degradation; nevertheless a slight increase of the average transmitted power (typically less than 1dB) is necessary. Figure 2 shows in grey the new regions allowed by the ACE algorithm for a QPSK, 16QAM and for a 64QAM modulation.



(a) (b) (c)

Figure 2: Example of (a) QPSK, (b) 16QAM and (c) 64QAM constellation on a subcarrier of an OFDM signal. Shaded regions represent corner-point extension regions, and the dotted, arrowed lines represent the extension paths for side points

The discrete-time domain OFDM symbol can be written as:

$$x^m(n) = \frac{w(n)}{\sqrt{N}} \sum_{k=0}^{N-1} X_k^m e^{j2\pi kn/N}$$

Where:

- $X^m = [X_0^m ... X_{N-1}^m]$ is the complex value QAM symbol vector,

- m the index of the OFDM symbol,
- k the index of the subcarriers,
- $w(n)$ is a discrete-time rectangular window with amplitude 1 over the interval and
- N the number of subcarriers.

A cyclic prefix is appended to cope with interblock interferences and to create non selective parallel frequency channels (inducing very simple frequency domain equalization) [1].

The signal shall be oversampled to $x[n/L]$, where L is the oversampling factor. There is also a filtering operation. Filtering and oversampling allow at the same time to make sure to have no ACPR and no EVM.

The goal of ACE can be summarized as the minimization of $\max_n \left|\overline{x^m}(n/L)\right|^2$ with

$$\overline{x^m}(n/L) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} (X_k^m + C_k^m) e^{j2\pi kn/NL}$$

where $C^m = [C_0^m ... C_{N-1}^m]$ are the selected extension vectors. Indeed, when peak powers are minimized, the PAPR is reduced.

Figure 3 represents the ACE algorithm block diagram based on the adaptive-scaling algorithm introduced in [16].

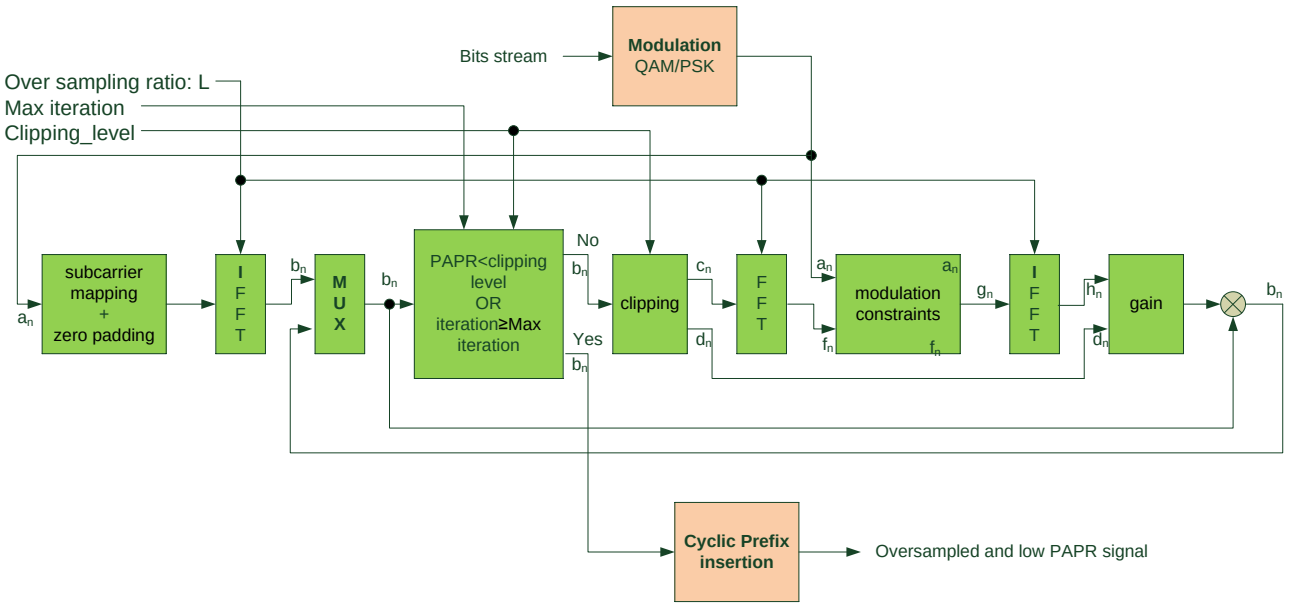


Figure 3: ACE block diagram

The algorithm is composed of 8 steps:

- Step 1: The signal that comes from the “Modulation QAM/PSK” block is oversampled in the time domain thanks to the subcarrier mapping and to the oversized by L IFFT.
- Step 2: The PAPR of the current oversampled OFDM symbol is computed. The algorithm stops the processing if i) the PAPR is lower than a predefined clipping level, or ii) the number of iteration reaches the predefined maximum iteration. The PAPR is defined by

$$PAPR = \frac{\max\left(|b_n|^2\right)}{\frac{1}{L.N_{FFT}}\sum_{n=0}^{L.N_{FFT}-1}|b_n|^2}$$

- Step 3: The time domain OFDM symbol is clipped according to the following formula:

$$c_n = \begin{cases} Clipping\_level.\exp\ j\Phi_n, & |b_n| > Clipping\_level \\ b_{n,} & |b_n| \le Clipping\_level \end{cases}$$

where Φ_n is the phase of b_n.

The resulting time domain OFDM symbol that has been clipped is

$$\tilde{x}^m(n/L) = \frac{1}{\sqrt{N}}\sum_{k=0}^{N-1}(X_k^m + \tilde{C}_k^m)e^{j2\pi kn/NL}$$

where $\tilde{C}^m = [\tilde{C}_0^m ... \tilde{C}_{N-1}^m]$ are the extension vectors created by the clipping.

The clipped signal is defined by $d_n = b_n - c_n$.

- Step 4: The frequency domain clipped OFDM symbol is computed thanks to an oversized by L FFT.
- Step 5: The "Modulation constraints" are applied for each subcarriers (see Figure 2). This function allows new points being in the shaded areas (or arrows) to remain and replaces the others by the original constellation point. It results in selecting among the $\tilde{C}_k^m$ the C_k^m which responds to the area condition.
- Step 6: The time domain signal h_n of C_k^m is computed.
- Step 7: The time domain ACE OFDM symbol is computed as

$$b_n = b_n + \beta \cdot h_n$$

where the gain β is defined as:

$$\frac{\mathrm{Re}\left[\sum_{n=1}^{N.N_{FFT}} d_n h_n^*\right]}{\sum_{n=1}^{N.N_{FFT}}|h_n|^2}$$

These processing are made for each OFDM symbol, and the algorithm stops after a fixed number of iterations or when the OFDM symbol maximum amplitude value is smaller than the clipping threshold. It is an iterative processing, which requires an extra FFT function in terms of implementation. Both FFT and IFFT functions are also increased by a factor L corresponding to the oversampling ratio. The filtering does not increase the complexity since it simply sets the out of band subcarriers to 0.

3.2.2 Active Constellation Extension simulations

For the past years, PAPR has been widely used as the figure of merit to assess the power variations of non-constant envelope signals. For multicarriers signals, the PAPR upper bound is usually defied as

$$PAPR = 10\log 10(Nu)$$

where Nu is the number of active subcarriers.

The PAPR value is reached if and only if the whole Nu subcarriers align themselves in phase. The main drawback of this figure of merit is that it refers only to the peak power which has a very low occurrence probability, and in consequence PA has a very low impact on linearity at this particular peak power.

In SACRA, we are focused on the 5 MHz channelization. The common simulation parameters and the ACE specific parameters are respectively given in Table 1 and Table 2. We choose to set the oversampling ratio L=4 because It has been proven [17] that the digital PAPR of the OFDM signal oversampled at a rate of L = 4 gives a good approximation of the equivalent continuous-time PAPR. The clipping level setting according to the modulation order (4, 16 and 64) has been chosen empirically in order to provide the best PAPR reduction capabilities.

Sampling Frequency without over sampling (Fs)	2 x 3.84 MHz
IFFT/FFT size without over sampling (N_{FFT})	512
Number of active subcarriers	500
Cyclic Prefix size without over sampling (Ncp)	108
Number of simulated OFDM symbol (Ns)	1000

Table 1: Common simulation parameters for 5 MHz channelization

Modulation	Oversampling ratio (L)	IFFT/FFT size with over sampling ($L \times N_{FFT}$)	Maximum number of iteration	Clipping level
QPSK	4	4 x 512	6	4.86 dB
16QAM	4	4 x 512	6	6 dB
64QAM	4	4 x 512	6	8.5 dB

Table 2: ACE modulation specific parameters

The figure of merit that we propose to use to assess the PAPR reduction is the CCDF (Cumulative Complementary Density Function). CCDF represents the probability of instantaneous power to be higher than a given power P0. Figure 4, Figure 5 and Figure 6 gives respectively the CCDF without ACE and with ACE after 1 to 6 iterations for QPSK, 16QAM and 64QAM.

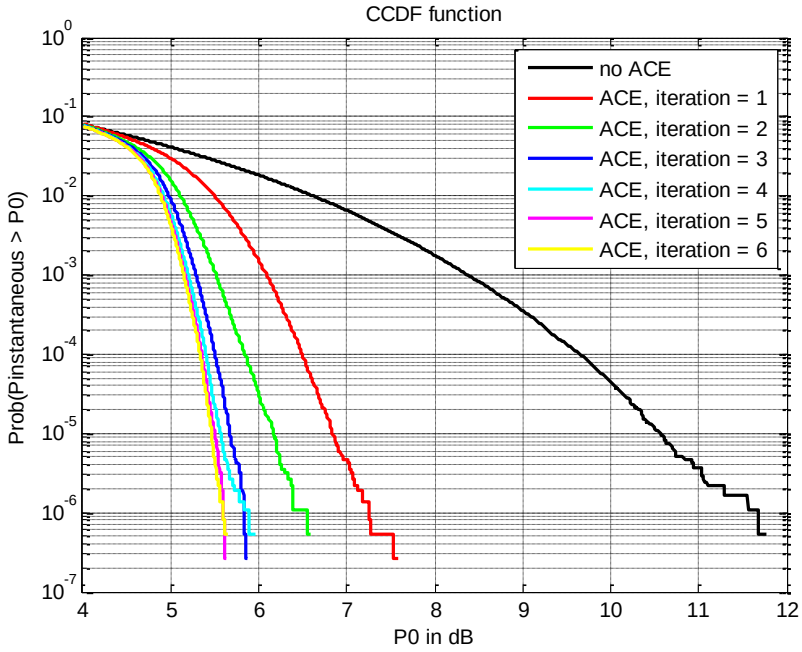


Figure 4: CCDF of the QPSK-OFDM signal according to the number of iteration

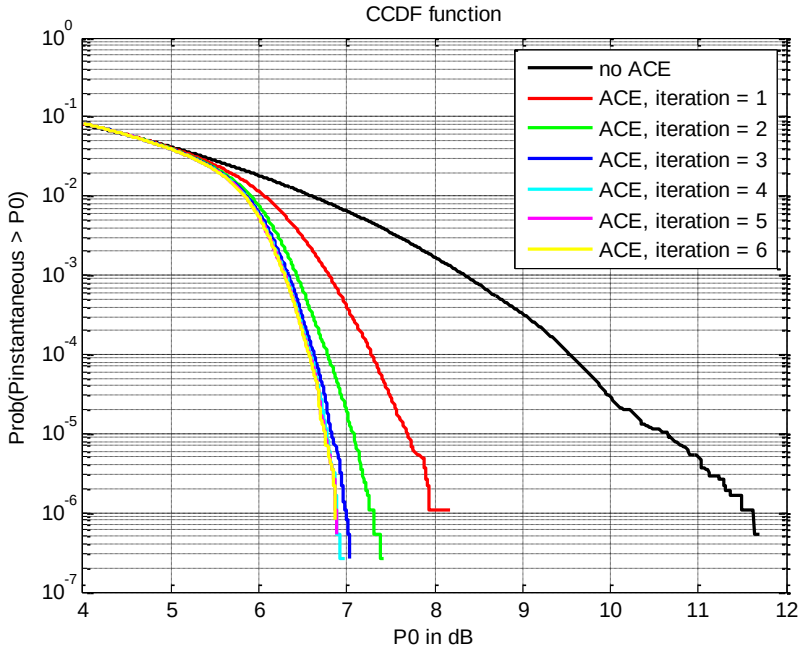


Figure 5: CCDF of the 16QAM-OFDM signal according to the number of iteration

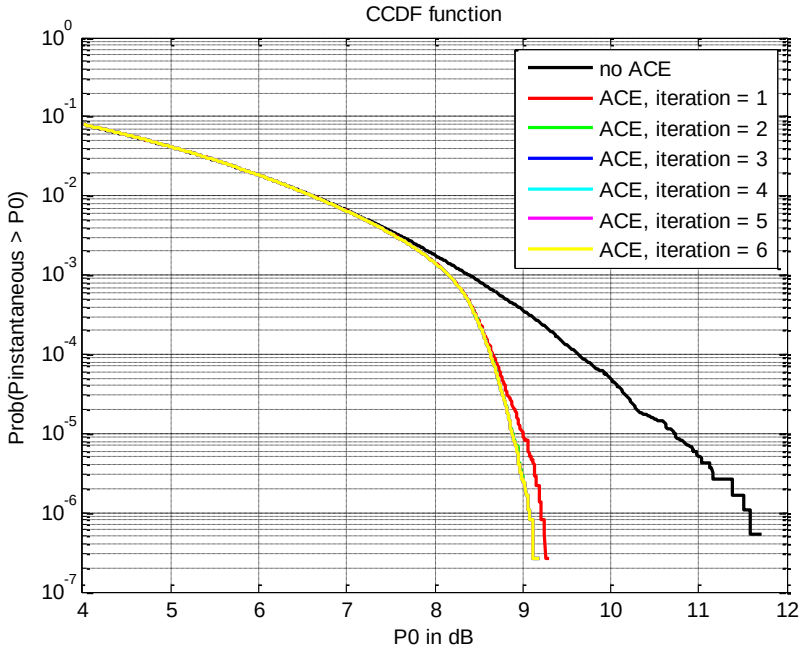


Figure 6: CCDF of the 64QAM-OFDM signal according to the number of iteration

Table 3 presents the PAPR reduction capabilities at CCDF=10^{-6} .We can see that the performance is not that much improved for more than 3 iterations for QPSK and 16QAM, and 1 iteration is enough for 64QAM. One can observe that the PAPR reduction capability decreases with the modulation order. This can be explain by the fact that the modulation constraints (degree of constellation freedom) of high modulation orders are much more constraining than lower modulation orders

Modulation	PAPR without ACE	PAPR with ACE	Number of iteration	Gain
QPSK	11.6 dB	5.8 dB	3	5.8 dB
16QAM	11.5 dB	7 dB	3	4.5 dB
64QAM	11.5 dB	9.2 dB	1	2.3 dB

Table 3: PAPR reduction performance for QPSK, 16QAM and 64QAM @ CCDF=10-6

3.3 DIGITAL PREDISTORTION TECHNIQUE

3.3.1 Proposed DPD

The choices for PA linearization that we propose in SACRA are the following:

- **Digital Predistortion**: great level flexibility and adaptivity.
- **Indirect learning**: direct computation of the predistortion (by postdistortion), avoid the PA modelisation step necessary by the direct learning.
- **Memory Polynomial**: take into account possible memory effects.

The block diagram of the proposed DPD is presented in Figure 7. The DPD is composed of 5 sub-functions or blocks.

The overall process of the DPD is composed of 4 steps (see Table 4). In the chronological order, these steps are:

- Step 1: RF impairments estimation

As presented in section 3.3.3, the presence of the RF impairments at the transmitter does not allow estimating properly the DPD. One option is to design very accurate analogue components to avoid as much as possible impairments. The second option, which has been selected for SACRA, is to tolerate analogue impairments because they can be digitally estimated and compensated. This step is in charge of such purpose.

- Step2: Digital PreDistortion estimation

Once the transmitter RF impairments have been estimated, they can be digitally pre-compensated. If the estimations are accurate enough, then the remaining RF impairments is supposed to be the PA. Step 2 consists in estimating the Digital PreDistortion according to the transmitted signal and to the feedback signal coming from the PA output.

- Step 3: Update Digital PreDistortion

As described in section 3.3.1.4, the DPD module can be based on a polynomial or on one or several LUTs. Step 3 is in charge of updating the Digital PreDistortion module transmitting the polynomial coefficients or filling LUTs.

- Step 4: Apply Digital PreDistortion

In step 4, the RF transmitter RF impairments and the PA are fully digitally pre-compensated.

	RF Impairments Estimation Block	RF Impairments Compensation Block	Digital PreDistortion Computation Block	Digital PreDistortion Update Block	Digital PreDistortion Module Block
Step 1: RF impairments estimation	Activated	Bypassed	Deactivated	Deactivated	Bypassed
Step 2: Digital PreDistortion estimation	Deactivated	Activated	Activated	Deactivated	Bypassed
Step 3: Update Digital PreDistortion	Deactivated	Activated	Deactivated	Activated	Bypassed
Step 4: Apply Digital PreDistortion	Deactivated	Activated	Deactivated	Deactivated	Activated

Table 4: Scheduling of Digital PreDistortion processing

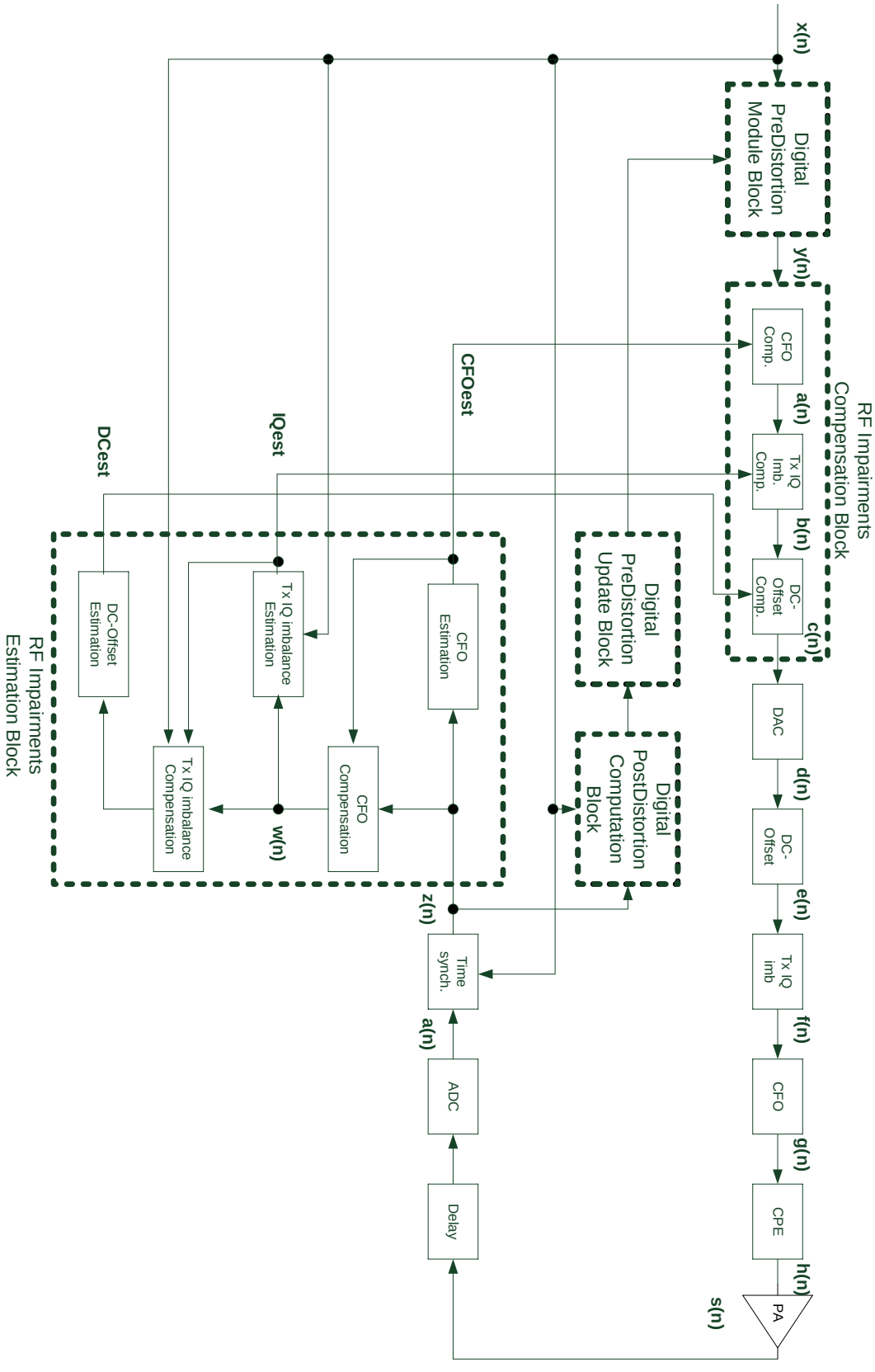


Figure 7: Block diagram of the proposed DPD

3.3.1.1 Radio Frequency Impairments

The DPD is dedicated to the compensation of the PA non linearity and is in charge of estimating its inverse characteristics. DPD does not aim to estimate other Radio Frequency impairments. Unfortunately, the signal that comes from the feedback loop has experienced several others RF impairments such as:

- DC offset
- Transmitter IQ Imbalance
- Carrier Frequency Offset
- Common Phase Error
- Time delay

These impairments usually lead to an inaccurate or poor DPD estimation (see [36] and section 3.3.3) As a result, their impact shall be minimized as much as possible. The goal of the “Radio Frequency Impairments Estimation” block is to estimate RF impairments.

3.3.1.1.1 Radio Frequency origins, models and impacts

Transmitter DC-Offset

Direct-conversion architectures, also called zero-IF architectures, are good candidates for producing low complexity transmitters. However, two drawbacks of the direct-conversion are i) the DC-Offset and ii) the sensitivity to IQ imbalance.

At the transmitter side, two mixers are used to translate the baseband signal to the correct carrier frequency. The DC-offset is due to the finite mixer port to port isolation: the mixer output experience self-mixing of LO leakage and LO signal, resulting in DC-Offset.

The baseband model of a signal impaired by the transmitter DC-Offset is [29]:

$$e(n) = d(n) + DC \qquad (1)$$

Where:

- $d(n)$ is the ideal complex baseband signal and
- DC is the complex DC-offset model.

Note that the real and imaginary parts of the DC-offset model are independent. In the model, we will suppose that DC is constant (time-invariant).

Figure 8 shows the impact of the transmitter DC-Offset on the OFDM spectrum. One can observe that the DC-Offset acts as a sub-carrier with index k=0.

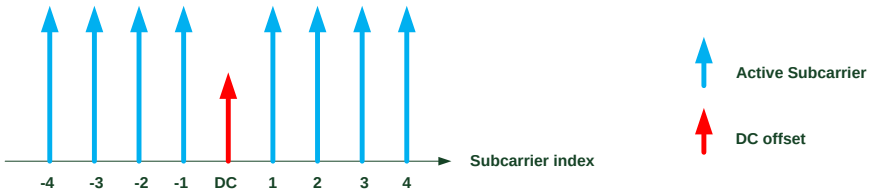


Figure 8: Impact of transmitter DC-Offset on OFDM spectrum. Transmitter DC-Offset acts as a sub-carrier with index k=0

Transmitter IQ imbalance

I/Q imbalance can be considered as non-frequency selective or frequency-selective. In SACRA, we are focussed only on non-frequency selective since the impact of the frequency selectivity on DPD is very low.

The IQ imbalance is caused by a gain mismatch *g* and a phase mismatch ϕ (Figure 9). The gain mismatch is due to a difference of the overall gain between I and Q branches. The phase mismatch is due for example to a non-ideal layout, i.e., the fact that the lines between the mixer and the local oscillator of I and Q branches are not exactly of the same length.

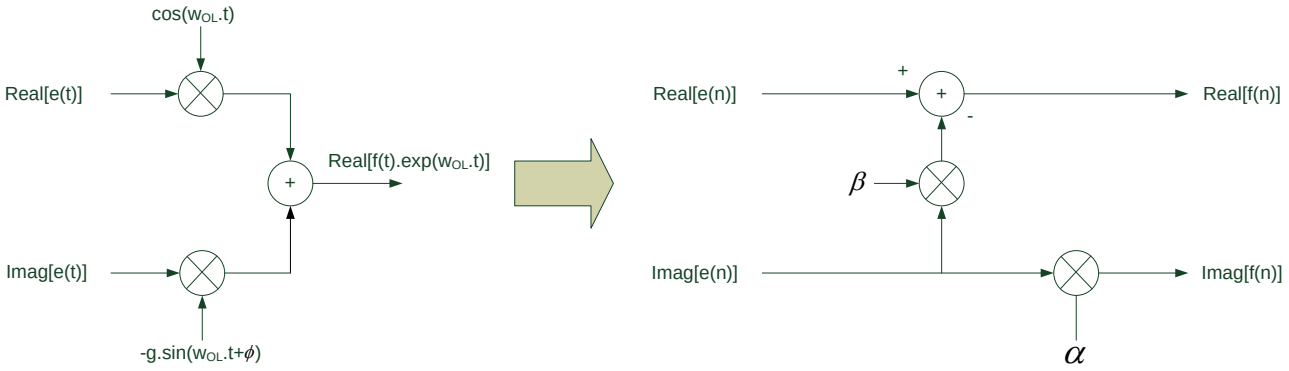


Figure 9: (left) Imperfect direct-conversion architecture caused by gain (g) and phase mismatch (φ); (right) Baseband transmitter IQ imbalance model

The equivalent transmitter I/Q imbalance baseband model is as follows [30]:

$$f(n) = e(n)K_1 + e^*(n)K_2^* \qquad (2)$$

$e(n)$ and $f(n)$ refer respectively to the baseband signal to be transmitted and to the equivalent baseband signal impaired by the transmitter I/Q imbalance. K1 and K2 are the parameters corresponding to the transmitter I/Q imbalance:

$$K_1 = \frac{1 + g\exp(j\phi)}{2} = \frac{1 + \alpha + j\beta}{2}$$

$$K_2 = \frac{1 - g\exp(-j\phi)}{2} = \frac{1 - \alpha + j\beta}{2}$$

with $\alpha = g\cos\phi$ et $\beta = g\sin\phi$.

It is important for the following to note that $K_2 = 1 - K_1^*$.

In the case of OFDM signals or cyclic prefix based signals, the frequency domain baseband signal impaired by the transmitter IQ imbalance can also be rewritten as follows [34]:

$$F_k(n) = E_k(n)K_1 + E_{-k}^*(n)K_2^*$$

The illustration of the previous equation is presented in Figure 10. One can observe that there exists a self-interference between pair-wised subcarriers. The level of this interference depends on the ratio between the parameters K_1 and K_2.

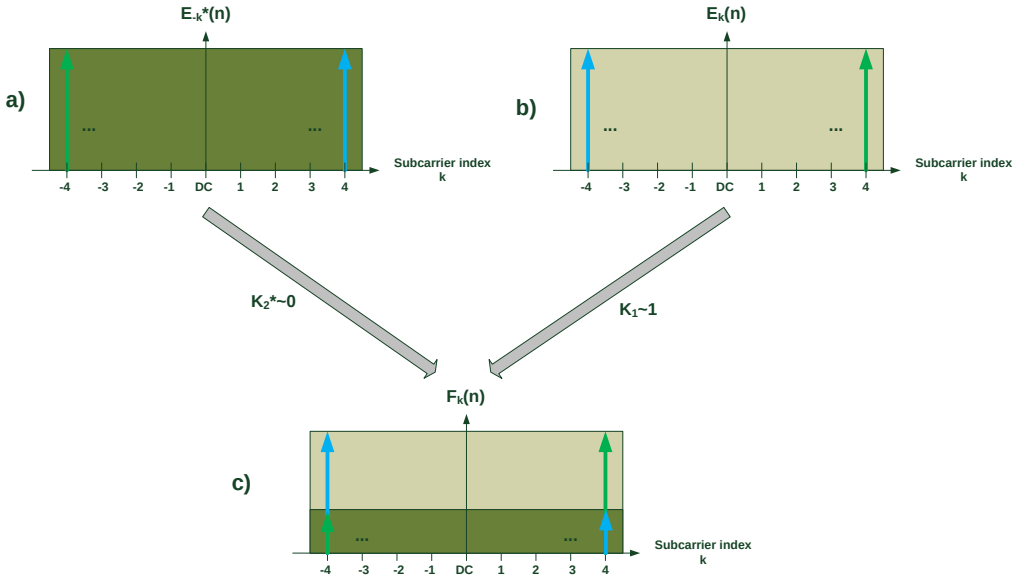


Figure 10: Impact of the transmitter IQ imbalance on OFDM spectrum: a) Spectrum of the complex conjugate signal to be transmitted, b) Spectrum of the signal to be transmitted, c) Spectrum of the transmitted signal impaired by the transmitter IQ imbalance

Carrier Frequency Offset

In SACRA, we choose to use a direct conversion up-conversion and a low-IF down-conversion for the feedback path. As a result, the baseband signal coming from the feedback loop has experienced an analogue up-conversion (IQ modulator) and a mixed analogue/digital down-conversion (Intermediate Frequency transposition + digital IQ demodulator).

The Carrier Frequency Offset (CFO) refers to the overall oscillator mismatch between the transmitter carrier frequency and the sum of the intermediate frequency plus digital carrier frequency. In SACRA, we choose to consider that the intermediate frequency plus digital carrier frequency is the reference carrier. In consequence, the CFO is modelled at the transmitter as follows:

$$g(n) = f(n).\exp\left(-j.2\pi.n.CFO/(F_s)\right) \quad (3)$$

where F_s is the sampling frequency used for DPD.

Figure 11 shows the impact of the CFO on OFDM demodulation. One can observe that the frequency shift due to CFO creates self interference and intercarrier interferences.

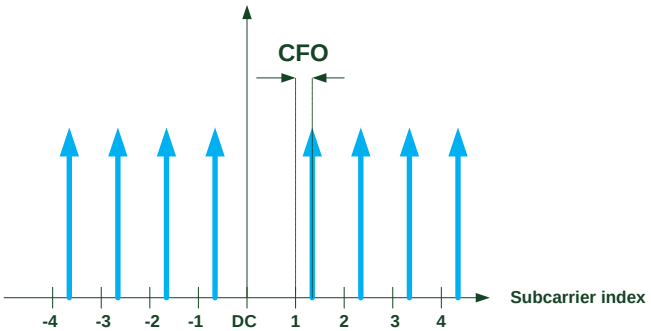


Figure 11: Impact of CFO on OFDM demodulation

Common Phase Error

The CPE is the average phase noise offset also known as the common phase error. It basically refers to the constant phase mismatch between the up and the down-conversion and can be modelled as follows [32]:

$$h(n) = g(n)\exp(jCPE)$$

Time delay

The time delay between the transmitted samples and the samples coming from the feedback loop is due to the group delay of the analogue and digital devices. The delay can be modelled as follows:

$$z(n) = a(n-\tau)$$

where τ is the delay in number of samples.

3.3.1.1.2 RF impairments estimation methodology

The approach that we propose in SACRA is to first estimate the transmitter RF impairments before estimating the DPD. CFO estimation process requires dedicated signal, whereas the other estimations can be done during the transmission of any kind of cyclic prefix based signals. Our proposal in SACRA to estimate the RF impairments is to use a preamble (see Figure 12). The first part of the preamble is specific and is design to estimate the CFO. This part consists in repeating at least two times the same block symbol. The second part of the preamble is dedicated to DC-Offset, I/Q Imbalance and CPE estimations. Note that it is not mandatory to pre-compensate the CPE in the "RF Impairments Compensation" block, but CPE estimation is required for DC-Offset estimation. The signal used for the DC-Offset, IQ imbalance and CPE could be any kind of cyclic prefix based signal.

This preamble is not compliant with LTE due to the repetition of two similar OFDM symbols necessary for the CFO estimation. We believe that it is not an issue since the RF impairments slowly vary with time, thus they can be estimated off-line before regular transmissions. If, for any reasons, it is not necessary to estimate the CFO, a dedicated preamble is not anymore mandatory and RF impairment estimations can be performed during regular transmission.

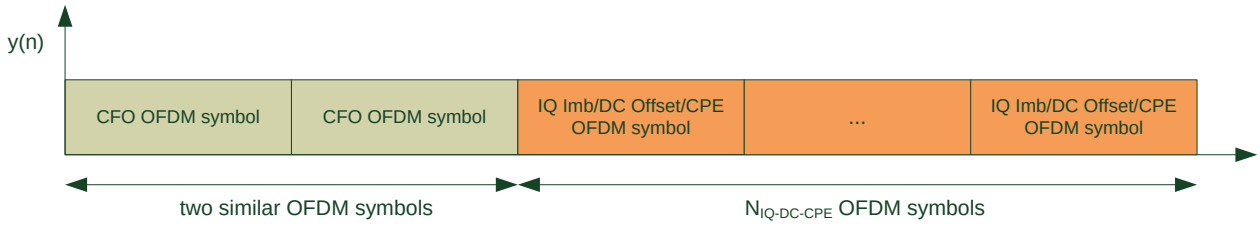


Figure 12: Preamble used for RF impairments estimations

The scheduling of the "RF impairments estimation" block is presented by the Figure 7.

Time synchronization: this function is a usual correlation between transmitted and received signal and is not described in the following. It is important to note that the time synchronization is very robust to realistic transmitter DC-Offset, IQ Imbalance, and CFO.

3.3.1.1.3 Carrier Frequency Offset

Estimation

The carrier Frequency Offset estimation is obtained thanks to a dedicated signal. This signal consists in repeating at least two times a pattern. The CFO estimation is done computing the phase shift between the received two (or more) patterns [33]:

$$CFOest(Hz) = \frac{angle\left(\sum_{l=0}^{N_1.N_{CFO-P}-1} z(l).z^*\left(l + \left(N_{CFO-P} + N_{CFO-PC}\right)\right)\right)}{2\pi.F_s.\left(N_{CFO-P} + N_{CFO-PC}\right)}$$

where:

- $z(n)$ refers to the baseband samples that have been time synchronized,
- N_{CFO-P} is the number of active subcarriers of the OFDM pattern used for CFO estimation,
- N_{CFO-PC} is the number of samples (@ Fs) dedicated to the Cyclic Prefix,
- F_s is the sampling frequency in Hz.

CFO compensation

The CFO compensation consists in correcting the instantaneous phase of each received time domain samples:

$$w(n) = z(n).\exp\left(j.2\pi.n.CFO / F_s\right)$$

3.3.1.1.4 Transmitter I/Q Imbalance estimation

Transmitter I/Q imbalance estimation

The estimation that we propose for SACRA takes advantage of OFDM capabilities in the sense that the transmitted (received) OFDM symbols before IFFT (after FFT) can be written in a very simple way. Another key property of OFDM in the context of I/Q imbalance is that the k^{th} subcarrier is only impaired by the $-k^{th}$ subcarrier, and vice versa. This property is applicable only if the signal does not experience CFO, so that is the reason why the CFO has been compensated in the "RF Impairments Estimation" block The frequency domain baseband signal impaired by the transmitter I/Q imbalance can be written as follows [34]:

$$F_k(n) = X_k(n)K_1 + X^*_{-k}(n)K_2^*$$

where $X_k(n)$ refers to the k^{th} transmitted subcarrier of the n^{th} OFDM symbol.

The received k^{th} subcarrier of the n^{th} OFDM symbol coming from the feedback loop, time synchronized and CFO compensated can be written as follows:

$$W_k(n) = H_k F_k(n) + N_k(n) = H_k\left[X_k(n)K_1 + X^*_{-k}(n)\left(1 - K_1\right)\right] + N_k(n) \qquad (4)$$

where**:**

- H_k refers to the modem frequency domain response and
- $N_k(n)$ refers to the received noise and also to the impact of the PA non linearity.

Note that we assume that the CFO has been perfectly compensated. It is also assumed that the signal coming from the feedback loop has been perfectly time synchronized.

The k[th] received subcarriers coming from the OFDM symbols n_1 and n_2 form a linear equation in two variables K_1 and H_k :

$$\begin{cases} W_k(n_1) &= H_k\left[X_k(n_1)K_1 + X^*_{-k}(n_1)(1-K_1) + N_k(n_1)\right] \\ W_k(n_2) &= H_k\left[X_k(n_2)K_1 + X^*_{-k}(n_2)(1-K_1) + N_k(n_2)\right] \end{cases}$$

If $\frac{X^*_{-k}(n_1)}{X_k(n_1)} \neq \frac{X^*_{-k}(n_2)}{X_k(n_2)}$ and $X_k(n_1) \neq X^*_{-k}(n_1) \cup X_k(n_2) \neq X^*_{-k}(n_2)$, then the solution of this system is:

$$Kest_1 = \frac{W_k(n_2)X^*_{-k}(n_1) - W_k(n_1)X^*_{-k}(n_2)}{W_k(n_1)\left[X_k(n_2) - X^*_{-k}(n_2)\right] - W_k(n_2)\left[X_k(n_1) - X^*_{-k}(n_1)\right]}$$

Note that the expression of $Kest_1$ assumes that the noises $N_k(n_1)$ and $N_k(n_2)$ are negligible. We will show in section 3.3.2.4 dedicated to the transmitter IQ imbalance estimation performance that this assumption is reasonable.

In order to obtain an accurate estimation, it is possible to average the $Kest_1$ estimations over time (use of several OFDM symbols) and frequency (use of all available active subcarriers) [35].

Note that the proposed algorithm does not need transmission of specific signal, and can be performed during regular LTE or prefix cyclic based multicarrier transmissions.

3.3.1.1.5 Transmitter DC-Offset estimation

Since the DC sub-carrier in LTE is always set to zero, the received sub-carrier at frequency index k=0 is zero if the transmitter is not impaired by IQ imbalance. On the other hand, if the transmitter is impaired by the IQ imbalance, the received signal after OFDM demodulation is (see eq. (4) and Figure 13):

$$W_0(n) = H_0 F_0(n) + N_0(n) = H_0\left[DC.K_1 + DC^*.(1-K_1)\right] + N_0(n) \qquad (5)$$

Note that if we assume that the DC-Offset is time-invariant or is very slowly time variant, $W_0(n)$ can be considered as a constant independently of the OFDM symbol n. We assume for the following that the term $N_0(n)$ is negligible. We will show in section 3.3.2.5 dedicated to the DC-offset estimation performance that this assumption is reasonable.

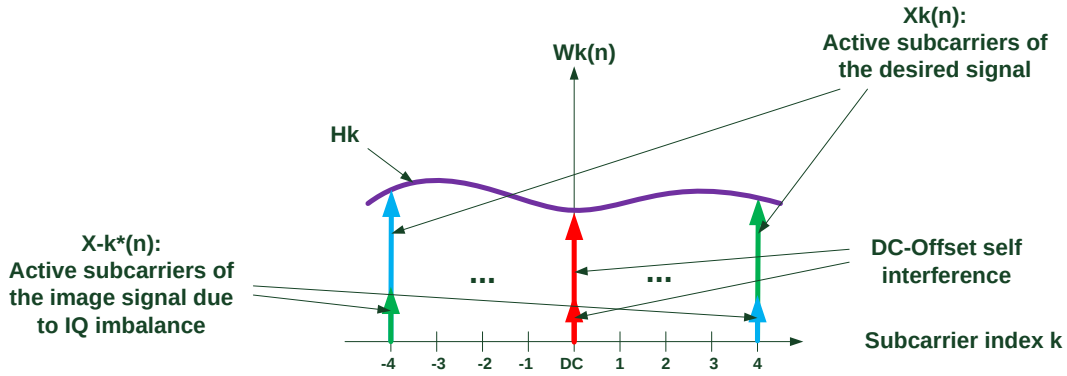


Figure 13: Received signal impaired by transmitter IQ imbalance after OFDM demodulation

Since K_1 has already been estimated, we can see from eq. (5), that it is possible to estimate the DC-Offset if the frequency response of the modem channel is known at the frequency index k=0, in other words, if H_0 is known.

From eq. (4), H_1 and H_{-1} are:

$$\begin{cases} H_1 = \dfrac{W_1(n)}{Kest_1\left(X_1(n) - X_{-1}^*(n)\right) + X_{-1}^*(n)} \\ H_{-1} = \dfrac{W_{-1}(n)}{Kest_1\left(X_{-1}(n) - X_1^*(n)\right) + X_1^*(n)} \end{cases}$$

We assume that the modem channel frequency response has a coherence bandwidth larger than the subcarrier spacing (15 kHz) and we propose to estimate H_0.averaging H_1 and H_{-1}:

$$Hest_0 = \frac{H_1 + H_{-1}}{2}$$

Now the DC-Offset can be estimated equalizing the received signal $W_0(n)$ by $Hest_0$, and then compensating the transmitter IQ imbalance:

$$DCest = \frac{Kest_1^* \dfrac{W_0(n)}{Hest_0} - Kest_2^* \dfrac{W_0^*(n)}{Hest_0^*}}{|Kest_1|^2 - |Kest_2|^2}$$

The reader is invited to refer to section 3.3.1.2.2 for transmitter IQ imbalance compensation description.

3.3.1.2 Radio Frequency Impairments Compensation

The goal of the “Radio Frequency Impairments Compensation” block is to compensate the CFO, transmitter IQ imbalance and DC-Offset according to estimations provided by the “Radio Frequency Impairments Estimation” block.

3.3.1.2.1 CFO compensation

The correction of the CFO at the transmitter consists in modifying the instantaneous phase of the signal to be transmitted:

$$a(n) = y(n).\exp\left(j.2\pi.n.CFO / F_s\right)$$

3.3.1.2.2 Transmitter IQ imbalance compensation

The baseband signal impaired by the transmitter IQ imbalance is

$$f(n) = e(n)K_1 + e^*(n)K_2^* \tag{6}$$

where $x(n)$ is the ideal signal.

Eq. (6) can be re written in the following matrix form:

$$\begin{pmatrix} f(n) \\ f(n)* \end{pmatrix} = \underbrace{\begin{pmatrix} K_1 & K_2^* \\ K_2^* & K_1 \end{pmatrix}}_{K} \begin{pmatrix} e(n) \\ e(n)* \end{pmatrix}$$

If the transmitter IQ imbalance estimation is available, it is possible to pre-compensate the baseband signal. This pre-compensation can be written as

$$\begin{pmatrix} b(n) \\ b(n)* \end{pmatrix} = Kest^{-1} \begin{pmatrix} a(n) \\ a(n)* \end{pmatrix}$$

where $Kest = \begin{pmatrix} Kest_1 & Kest_2^* \\ Kest_2^* & Kest_1 \end{pmatrix}$ is the transmitter IQ imbalance estimation matrix.

We assume that the DC-offset is perfectly estimated and pre-compensated, in other words $e(n) = b(n)$, see Figure 7. The IQ imbalance pre-compensated baseband signal at the IQ modulator output can be written as

$$\begin{pmatrix} f(n) \\ f(n)* \end{pmatrix} = \underbrace{\begin{pmatrix} K_1 & K_2^* \\ K_2^* & K_1 \end{pmatrix}}_{K} \begin{pmatrix} b(n) \\ b(n)* \end{pmatrix} = K.Kest^{-1} \begin{pmatrix} a(n) \\ a(n)* \end{pmatrix}$$

If the IQ imbalance parameters K_1 and K_2 have been perfectly estimated, then the pre-compensated baseband signal at the IQ modulator is $f(n) = a(n)$.

Thus, the relationship between $a(n)$ and $b(n)$ is (see Figure 7)

$$b(n) = \frac{Kest_1^*.a(n) - Kest_2^*.a(n)*}{\left|Kest_1\right|^2 - \left|Kest_2\right|^2}$$

3.3.1.2.3 DC-Offset compensation

The DC-Offset compensation simply consists in subtracting the DC-offset estimation:

$$c(n) = b(n) - DCest$$

where $DCest$ is the complex DC-offset estimation.

3.3.1.3 Digital Predistortion Computation

In SACRA, we choose to use an indirect learning technique for DPD [24] Indirect learning consists in computing first **post**distortion coefficients according to transmitted signal $y(n)$ and signal coming from the feedback loop $z(n)$. Then the **post**distortion coefficients are sent to the **pre**distortion. The goal of the "digital predistortion computation" block is actually to compute the digital **post**distortion.

Figure 14 presents the principle of the indirect learning.

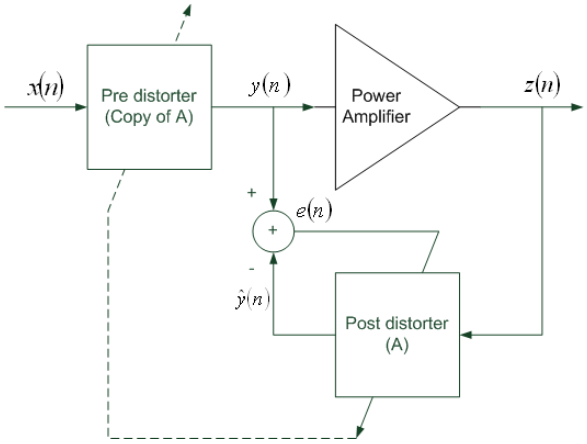


Figure 14: Indirect Learning DPD architecture

There exists two categories for digital predistortion based on memory polynomials namely "traditional memory polynomial" [21] and "orthogonal memory polynomials" [26][27]. Traditional memory polynomials exhibits numerical instabilities when high order terms are included, whereas orthogonal memory polynomials are supposed to alleviate the numerical instability problem in certain conditions. In this document, we provide description and simulation results for both methods.

Polynomial predistortion can be performed using only odd orders or using both odd and even terms. The use of only odd orders allows simplifying implementation since complex computations such as square root are avoided. In the other hand, it has been shown in [25] that the inclusion of even terms in predistortion enhances performance. In this document, we provide description and simulation results for both methods.

3.3.1.3.1 Traditional memory polynomials – Odd terms only

In the case of traditional memory polynomials, the relationship between $\hat{y}(n)$ and $z(n)$ is

$$\hat{y}(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} \beta_{kq} z(n-q) \left| z(n-q) \right|^{2(k-1)} \tag{7}$$

where:

- K is the number of polynomial coefficients and
- Q is the memory depth.

The goal of the postdistorter is to perfectly linearizes the PA, then it is supposed that the samples at the PA output $\hat{y}(n)$ are exactly the same that the samples $x(n)$ that have been transmitted to the PA. In this case, $\hat{y}(n)$ can be substituted to $x(n)$ in the previous equation:

$$x(n) = \sum_{k=1}^{K}\sum_{q=0}^{Q} \beta_{kq} z(n-q)\left|z(n-q)\right|^{2(k-1)} \tag{8}$$

The matrix form of the previous equation is as follows:

$$\vec{x} = \vec{\beta}.Z \tag{9}$$

where:

- $\vec{x} = \left[x(n), x(n-1), \cdots, x(n-N+1)\right]^T$,
- N is the pattern size,
- $\vec{\beta} = \left[\beta_{10},...,\beta_{1Q},\beta_{20},...,\beta_{2Q},...,\beta_{K0},...,\beta_{KQ}\right]^T$,
- $Z = \left[\vec{Z_n}, \vec{Z_{n-1}},...,\vec{Z_i},...,\vec{Z_{n-N+1}}\right]^T$,
- $\vec{z_i} = \begin{bmatrix} z(i), z(i)\left|z(i)\right|^2,...,z(i)\left|z(i)\right|^{2(K-1)}, z(i-1), z(i-1)\left|z(i-1)\right|^2...,z(i-1)\left|z(i-1)\right|^{2(K-1)},... \\ ,z(i-Q), z(i-Q)\left|z(i-Q)\right|^2...,z(i-Q)\left|z(i-Q)\right|^{2(K-1)} \end{bmatrix}$

One can note that $\vec{x} = \vec{\beta}.Z$ is written in a linear way. The Least Square solution for $\vec{\beta}.$ is [28]:

$$\vec{\beta} = \left(Z^H Z\right)^{-1} Z^H \vec{x} \tag{10}$$

where $[.]^H$ denotes the Hermitian transpose.

3.3.1.3.2 Traditional memory polynomials – Odd and Even terms

The inclusion of even terms for traditional memory polynomials slightly modified the relationship between $\hat{y}(n)$ and $z(n)$ described by (7)

$$\hat{y}(n) = \sum_{k=1}^{K}\sum_{q=0}^{Q} \beta_{kq} z(n-q)\left|z(n-q)\right|^{k-1} \tag{11}$$

The estimation of the polynomial coefficients is exactly the same as in (9) except that $\vec{z_i}$ is built as follows

$$\vec{z_i} = \left[z(i), z(i)\left|z(i)\right|,...,z(i)\left|z(i)\right|^{K-1}, z(i-1), z(i-1)\left|z(i-1)\right|...,z(i-1)\left|z(i-1)\right|^{K-1},...,z(i-Q), z(i-Q)\left|z(i-Q)\right|...,z(i-Q)\left|z(i-Q)\right|^{K-1}\right]$$

3.3.1.3.3 Orthogonal memory polynomials – Odd terms only

Numerical instabilities are due to the fact that the condition number of the matrix $Z^H Z$ to be inverted in (9) is very large. Matrix condition number is very important if the implementation of the

matrix inversion is done in fixed-point: inversion of matrix which has a high condition number requires internal variables with very high precision. As a result, decreasing the condition number allows relaxing constraints on fixed-point matrix inversion. In [27], the authors propose an orthogonal basis for complex Gaussian processes. Thanks to the central limit theorem, OFDM signals can be accurately approximated by a complex Gaussian process. The relationship between $\hat{y}(n)$ and $z(n)$ is

$$\hat{y}(n) = \sum_{k=1}^{K}\sum_{q=0}^{Q} \beta_{kq} \Phi_k\left(z(n-q)\right) \tag{12}$$

where $\Phi_k(z) = \sum_{l=0}^{k-1} (-1)^{k-1-l} \frac{\sqrt{k}}{(l+1)!}\binom{k-1}{l} |z|^{2(l-1)} z$

The estimation of the polynomial coefficients is exactly the same as in (9) except that $\vec{z_i}$ is built as follows

$$\vec{z_i} = \left[\Phi_1(z(i)),...,\Phi_K(z(i)),\Phi_1(z(i-1)),...,\Phi_K(z(i-1)),...,\Phi_1(z(i-Q)),...,\Phi_K(z(i-Q))\right]$$

Note that the proposed approach for SACRA is to combined PAPR reduction algorithm with DPD. The PAPR reduction algorithm modifies the power distribution of the signal, and it is quite obvious that the OFDM signal after PAPR reduction is not anymore a complex Gaussian process. The consequence of this power distribution misalignment is addressed in section 3.3.2.6.

3.3.1.3.4 Orthogonal memory polynomials – Odd and Even terms

The inclusion of even terms in orthogonal polynomials is not compatible with the orthogonal basis proposed in [27]. In [26], the authors proposed an orthogonal basis when both odd and even terms are taken into account. The relationship between $\hat{y}(n)$ and $z(n)$ is

$$\hat{y}(n) = \sum_{k=1}^{K}\sum_{q=0}^{Q} \beta_{kq} \psi_k\left(z(n-q)\right) \tag{13}$$

where $\psi_k(z) = \sum_{l=1}^{k} (-1)^{l+k} \frac{(k+l)!}{(l-1)!(l+1)!(k-l)!} |z|^{l-1} z$

The estimation of the polynomial coefficients is exactly the same as in (9) except that $\vec{z_i}$ is built as follows

$$\vec{z_i} = \left[\psi_1(z(i)),...,\psi_K(z(i)),\psi_1(z(i-1)),...,\psi_K(z(i-1)),...,\psi_1(z(i-Q)),...,\psi_K(z(i-Q))\right]$$

Note that the orthogonal basis ψ_k is optimized for uniformly distributed signals. The power distribution the OFDM signal after PAPR reduction is clearly not uniformly distributed process. The consequence of this power distribution misalignment is addressed in section 3.3.2.6.

3.3.1.4 Digital PreDistortion Update

As stated in section 3.3.1.3, the predistorter is a copy of the postdistorter. The relationship between the memory orthogonal predistorter input/output is depends on the selected approach:

- Traditional memory polynomials – Odd terms only

$$y(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} \beta_{kq} x(n-q) \left| x(n-q) \right|^{2(k-1)} \tag{14}$$

- Traditional memory polynomials – Odd and Even terms

$$y(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} \beta_{kq} x(n-q) \left| x(n-q) \right|^{k-1} \tag{15}$$

- Orthogonal memory polynomials – Odd terms only

$$y(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} \beta_{kq} \Phi_k \left(x(n-q) \right) \tag{16}$$

- Orthogonal memory polynomials – Odd and Even terms

$$y(n) = \sum_{k=1}^{K} \sum_{q=0}^{Q} \beta_{kq} \psi_k \left(x(n-q) \right) \tag{17}$$

It is possible to apply predistortion to the samples to be transmitted according to either an explicit polynomial, or to one or more LUTs. Computing $y(n)$ according the eq. (14), (15), (16) or (17) for each sample to be transmitted is very demanding in terms of resource. For this reason, an efficient approach consists in pre computing the formula and saving the results in LUTs.

According to the selected approach, eq. (14), (15), (16) or (17) can be rewritten as follows:

- Traditional memory polynomials – Odd terms only

$$\begin{aligned} y(n) = {} & x(n) \cdot \underbrace{\left[\beta_{10} + \beta_{20} \left| x(n) \right|^2 + \beta_{20} \left| x(n) \right|^4 + \cdots + \beta_{K0} \left| x(n) \right|^{2(K-1)} \right]}_{LUT_1 = g_{NL,0}\left(\left| x(n) \right|^2 \right)} + \\ & x(n-1) \cdot \underbrace{\left[\beta_{11} + \beta_{21} \left| x(n-1) \right|^2 + \beta_{21} \left| x(n-1) \right|^4 + \cdots + \beta_{K1} \left| x(n-1) \right|^{2(K-1)} \right]}_{LUT_2 = g_{NL,1}\left(\left| x(n-1) \right|^2 \right)} + \\ & \cdots \\ & x(n-Q) \cdot \underbrace{\left[\beta_{1Q} + \beta_{2Q} \left| x(n-Q) \right|^2 + \beta_{20} \left| x(n-Q) \right|^4 + \cdots + \alpha_{KQ} \left| x(n-Q) \right|^{2(K-1)} \right]}_{LUT_Q = g_{NL,Q}\left(\left| x(n-Q) \right| \right)^2} \end{aligned} \tag{18}$$

- Traditional memory polynomials – Odd and Even terms

$$y(n) = x(n)\cdot\underbrace{\left[\beta_{10} + \beta_{20}|x(n)| + \beta_{20}|x(n)|^2 + \cdots + \beta_{K0}|x(n)|^{K-1}\right]}_{LUT_1 = g_{NL,0}(|x(n)|)} + \quad (19)$$

$$x(n-1)\cdot\underbrace{\left[\beta_{11} + \beta_{21}|x(n-1)| + \beta_{21}|x(n-1)|^2 + \cdots + \beta_{K1}|x(n-1)|^{K-1}\right]}_{LUT_2 = g_{NL,1}(|x(n-1)|)} +$$

$$\cdots$$

$$x(n-Q)\cdot\underbrace{\left[\beta_{1Q} + \beta_{2Q}|x(n-Q)| + \beta_{20}|x(n-Q)|^2 + \cdots + \alpha_{KQ}|x(n-Q)|^{K-1}\right]}_{LUT_Q = g_{NL,Q}(|x(n-Q)|)}$$

- Orthogonal memory polynomials – Odd terms only

$$y(n) = x(n)\cdot\underbrace{\left[\frac{\Phi_1(x(n))}{x(n)} + \frac{\Phi_2(x(n))}{x(n)} + \cdots + \frac{\Phi_K(x(n))}{x(n)}\right]}_{LUT_1 = g_{NL,0}(|x(n)|^2)} + \quad (20)$$

$$x(n-1)\cdot\underbrace{\left[\frac{\Phi_1(x(n-1))}{x(n-1)} + \frac{\Phi_2(x(n-1))}{x(n-1)} + \cdots + \frac{\Phi_K(x(n-1))}{x(n-1)}\right]}_{LUT_2 = g_{NL,1}(|x(n-1)|^2)} +$$

$$\cdots$$

$$x(n-Q)\cdot\underbrace{\left[\frac{\Phi_1(x(n-Q))}{x(n-Q)} + \frac{\Phi_2(x(n-Q))}{x(n-Q)} + \cdots + \frac{\Phi_K(x(n-Q))}{x(n-Q)}\right]}_{LUT_Q = g_{NL,Q}(|x(n-Q)|^2)}$$

- Orthogonal memory polynomials – Odd and Even terms

$$y(n) = x(n)\cdot\underbrace{\left[\frac{\psi_1(x(n))}{x(n)} + \frac{\psi_2(x(n))}{x(n)} + \cdots + \frac{\psi_K(x(n))}{x(n)}\right]}_{LUT_1 = g_{NL,0}(|x(n)|)} + \quad (21)$$

$$x(n-1)\cdot\underbrace{\left[\frac{\psi_1(x(n-1))}{x(n-1)} + \frac{\psi_2(x(n-1))}{x(n-1)} + \cdots + \frac{\psi_K(x(n-1))}{x(n-1)}\right]}_{LUT_2 = g_{NL,1}(|x(n-1)|)} +$$

$$\cdots$$

$$x(n-Q)\cdot\underbrace{\left[\frac{\psi_1(x(n-Q))}{x(n-Q)} + \frac{\psi_2(x(n-Q))}{x(n-Q)} + \cdots + \frac{\psi_K(x(n-Q))}{x(n-Q)}\right]}_{LUT_Q = g_{NL,Q}(|x(n-Q)|)}$$

One can observe that $y(n)$ can be easily computed thanks to Q+1 LUTs, independently of the polynomial basis and also independently of the fact of taking into account odd terms or not. The complex values of the (Q+1) LUT are respectively addressed by

- $|x(n)|, |x(n-1)|, \ldots, |x(n-Q)|$ .if the odd and even terms are used
- $|x(n)|^2, |x(n-1)|^2, \ldots, |x(n-Q)|^2$ .if the odd terms are only used

Figure 15 presents the LUT DPD approach for even order only or even and odd orders. It is important to note that the size of the LUTs is independent of the polynomial order and the type of polynomial basis ("traditional" or orthogonal).

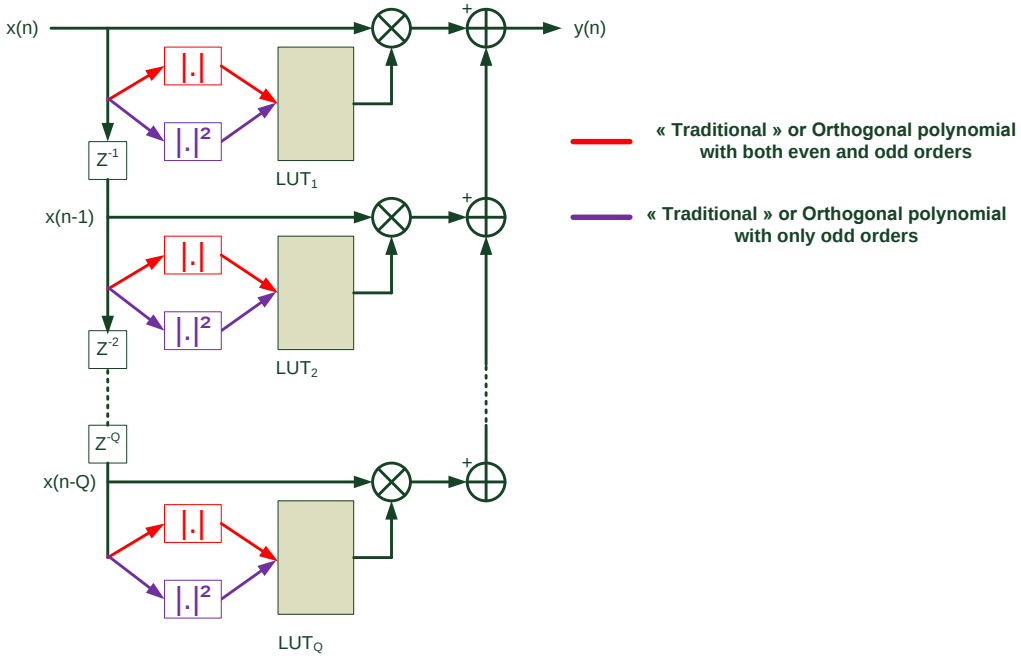


Figure 15: Principle of LUT DPD approach

In conclusion, the "DPD PreDistortion update" block functionality depends on the selected approach:

- If the DPD is performed thanks to an explicit polynomial, the "Digital PreDistortion Update" block sends to the "Digital PreDistortion Module" block the K.(Q+1) coefficients β_{kq}
- If the DPD is performed thanks to LUTs, the "Digital PreDistortion Update" block is in charge of filling the (Q+1) LUTs for the whole entries.

3.3.1.5 Digital PreDistortion Module

The "DPD PreDistortion Module" block is in charge of applying the DPD to the samples to be transmitted. According to the selected approach, there are 2 possibilities:

- If the DPD is performed thanks to an explicit polynomial, the "Digital PreDistortion Module" applies eq. (14), (15), (16) or (17).
- If the DPD is performed thanks to LUTs, the "Digital PreDistortion Module" is performed as described by Figure 15.

3.3.2 Parameters Setting

In this section, we address the DPD and RF impairments estimators' parameter setting. In order to assess accurately algorithm performance, we first build a PA model that is included into simulations.

3.3.2.1 Power Amplifier Model

Pedro in [18] gives an excellent comparative overview of existing PA model. Basically, PA model can be classified into two categories: memory less and memory PA, which is much more general. In this study, we are focussed on a very accurate memory PA model: the more realistic is PA model, the more precise is the parameters setting.

The Volterra Series approximation is known for its good memory effects modelling properties, but the parameters extraction is usually very difficult. To avoid extraction problem, simplified Volterra Series have been proposed, such as for example memory polynomial, wiener, Hammerstein, Parallel Hammerstein [21], Radial Basis Neural Network [19] and Generalized Memory Polynomial (GMP) models [20].

In SACRA, we choose to build a GMP model since this model is a good compromise between complexity, performance and extraction capabilities. The relationship between the PA input-output is as follows:

$$\begin{aligned} s(n) = & \sum_{k\in K_A}\sum_{l\in L_A} a_{kl}\,h(n)\left|h(n-l)\right|^k \\ & + \sum_{k\in K_B}\sum_{l\in L_B}\sum_{m\in M_B} b_{klm}\,h(n-l)\left|h(n-l-m)\right|^k \\ & + \sum_{k\in K_C}\sum_{l\in L_C}\sum_{m\in M_C} c_{klm}\,h(n-l)\left|h(n-l+m)\right|^k \end{aligned}$$

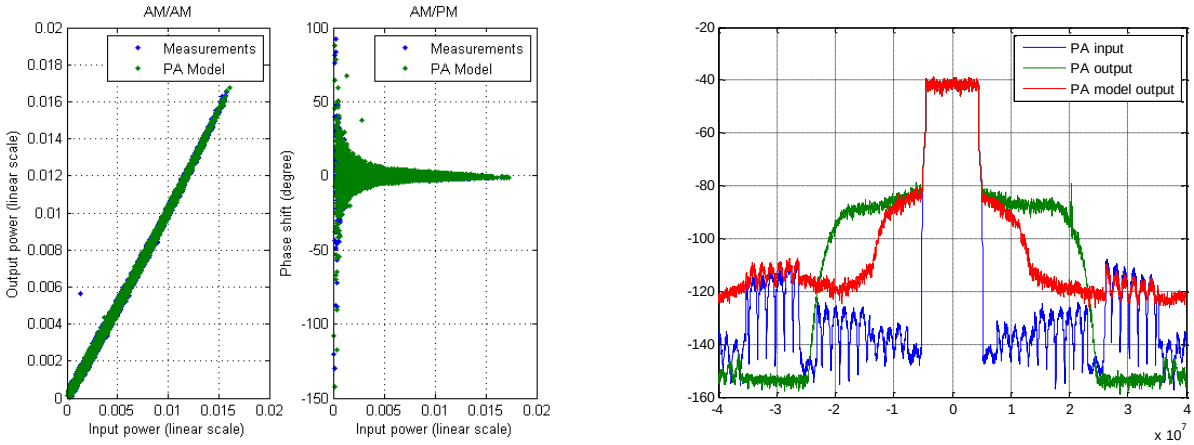
where K_A and L_A are the index arrays for aligned signal and envelope; K_B, L_B and M_B are the index arrays for signal and lagging envelope; and K_C, L_C and M_C are the index arrays for signal and leading envelope.

In SACRA, there is one PA dedicated to the 700MHz band and another PA for the 2.5GHz band, respectively "SKY77707" [21] and "SKY77706" [22]. The PA models are built thanks to PA input/output measurements. The excitation signal is a 10 MHz LTE compliant signal, oversampled by a factor 8. We have built six different PA model referring to 3 different mean input powers (-15, -9 and 1 dBm) for the 2 PAs. Figure 16 a), Figure 17 a) and Figure 18 a) give the AM/AM and AM/PM curves of the "SKY77706" PA and the PA model for respectively -15 dBm, -9 dBm and 1 dBm mean input power. Figure 16 b), Figure 17 b) and Figure 18 b) give the real PA and PA GMP model output spectrums for the three different output powers.

We can observe that the AM/AM and AM/PM curves are almost perfectly modelled. The Normalized Mean Square Error between the real PA output and the PA model output is given in Table 5.

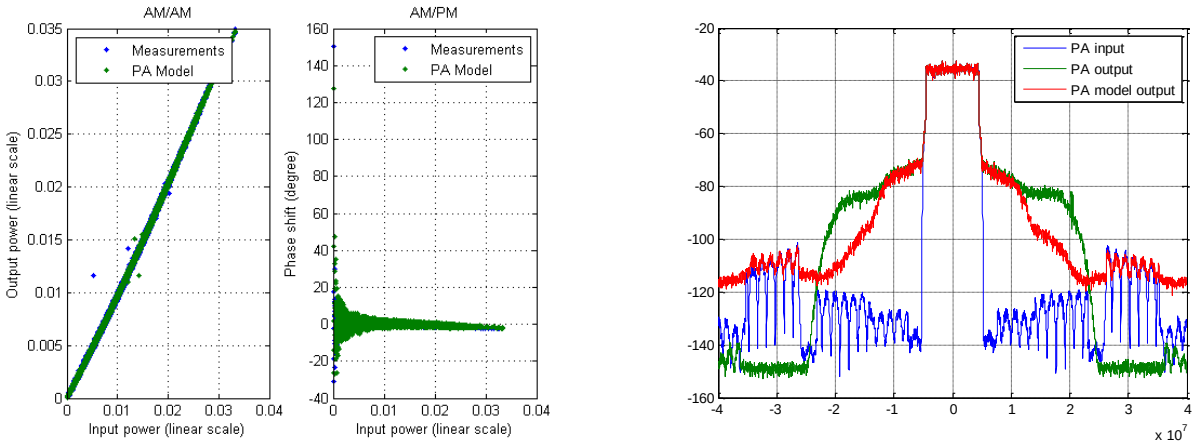
In any case, we can see that there is a misalignment of the spectrums for frequency greater than 20 MHz. This misalignment is due to the fact that the output signal of the measurements has been low pass filtered, which is not the case for the model. This mismatch is not an issue in our study since we are interested in the linearization of the first and second adjacent channels. One can observe that for the mean output powers of -15 and -9 dBm there exists a misalignment of the second adjacent channel. This mismatch is due to the ADC quantization error of the

measurements. Since the model is computed in floating point, this kind of impairments is not modelled. We believe that this second mismatch is not an issue since our model intends only to follow non-linearity's effects.



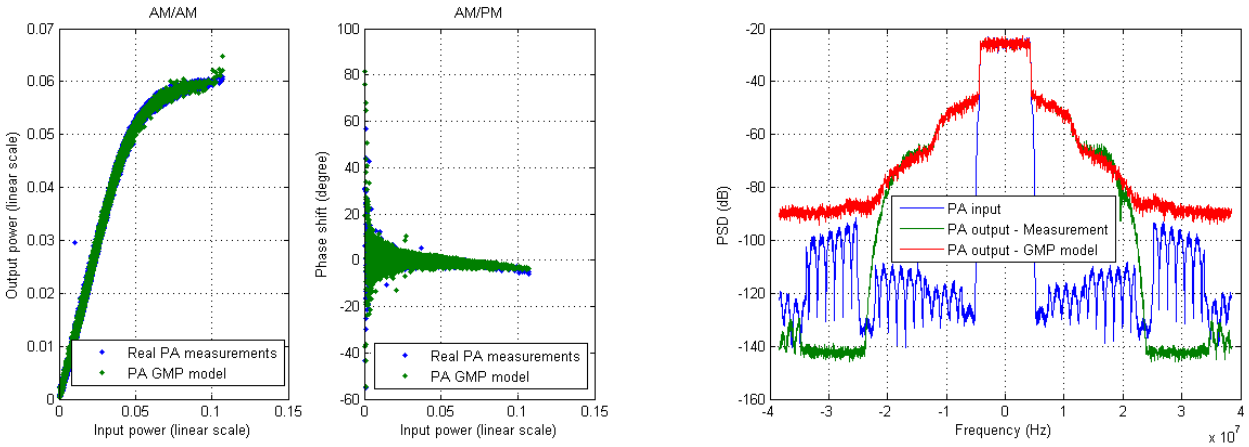
(a) (b)

Figure 16: Mean input power = -15 dBm: a) Comparison of real PA and GMP PA model AM/AM and AM/PM curves, b) Comparison between real PA and GMP PA model output spectrums



(a) (b)

Figure 17: Mean input power = -9 dBm: a) Comparison of real PA and GMP PA model AM/AM and AM/PM curves, b) Comparison between real PA and GMP PA model output spectrums



(a) (b)

Figure 18: Mean input power = 1 dBm: a) Comparison of real PA and GMP PA model AM/AM and AM/PM curves, b) Comparison between real PA and GMP PA model output spectrums

	Pin = -15 dBm	Pin = -9 dBm	Pin = 1 dBm
NMSE (dB)	-37.49	-39.53	-39.77

Table 5: Normalized mean square error of the PA model for -15, -9 and 1 dBm mean input powers

"SKY77707" PA models are quite similar to "SKY77706" models and are not presented in this document.

3.3.2.2 Over Sampling Ratio

The linearization capabilities depend on the oversampling. A rule of thumb is that the bandwidth to be linearized shall be at least equal to the oversampling ratio. In SACRA we aim at linearizing the first and second adjacent channel, meaning that an oversampling of 5 is mandatory. In order to let some room for digital and analogue filtering, we set the oversampling ratio to 6.

The baseband sampling frequency of the LTE 5MHz channelization is 7.68 MHz, meaning that the DPD shall works at a sampling frequency of 46 MHz.

The simulation parameters are presented in Table 6. These values are used for the rest of the simulations.

Parameters	Value
Baseband Sampling frequency	7.64 MHz
OverSampling	6
FFT size	512
Number of active subcarriers	300
Size of the Prefix Cyclic	108
Modulation	QPSK

PAPR reduction algorithm	ACE

Table 6: Simulation parameters

3.3.2.3 Carrier Frequency Offset

In SACRA, we assume that the LTE terminal local oscillators' tolerance is 1 ppm for the 2.5 GHz band. Thus, the corresponding maximum CFO is 2.5 kHz. Table 7 gives the CFO estimator performance for the 3 possible PA input powers when the CFO is set to 2.5 kHz.

It is interesting to note that the best estimation performance refers to the mean input power of -9 dBm. One could expect that when the mean input power of the power amplifier decreases, the PA non linearity's impacts on performance decreases, but the PA used in SACRA experiences a small gain expansion phenomenon at low levels [23] explaining such a behaviour.

For SACRA, we believe that CFO estimation performance is very accurate.

	Pin = -15 dBm	Pin = -9 dBm	Pin = 1 dBm
MSE (dB)	-53	-61	-52
Standard deviation (dB)	9	18	10

Table 7: Normalized mean square error of the PA model for -15, -9 and 1 dBm mean input powers

3.3.2.4 Transmitter IQ Imbalance

For terminal equipment, we believe that a phase imbalance of 3° and a gain imbalance of 0.25 dB are state of the art realistic values.

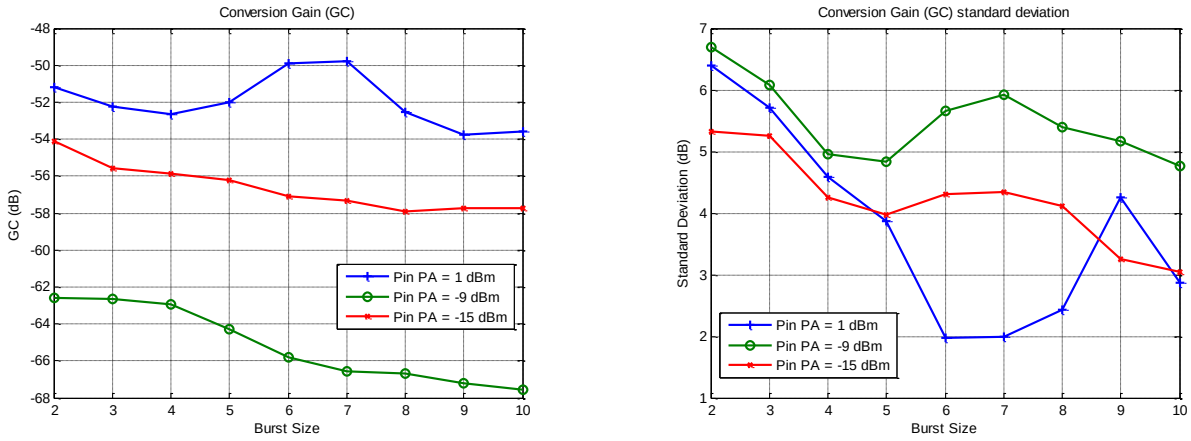
We must define a figure of merit in order to assess IQ imbalance estimation performance. In [31], the authors propose the conversion gain (GC), defined as the power of the undesired complex up-conversion divided by the power of the desired complex up-conversion:

$$GC(dB) = 10\log 10\left(\left|\frac{K_2}{K_1}\right|^2\right)$$

The initial GC with 3° and 0.25 dB of phase and gain imbalance is about -30 dB. Figure 19 gives the resulting mean and standard deviation of GC after estimation and compensation according to the number of OFDM symbols used for estimations, and according to the PA input power.

One can observe that increasing the number of symbols does not necessarily greatly improve the estimation performance. As noted in section 3.3.2.3 and for the same reasons, the best performance is given for the medium mean input power of -9 dBm.

In SACRA we recommend to estimate the IQ imbalance using only 2 OFDM symbols since the trade-off between complexity and performance is very good.



(a) (b)

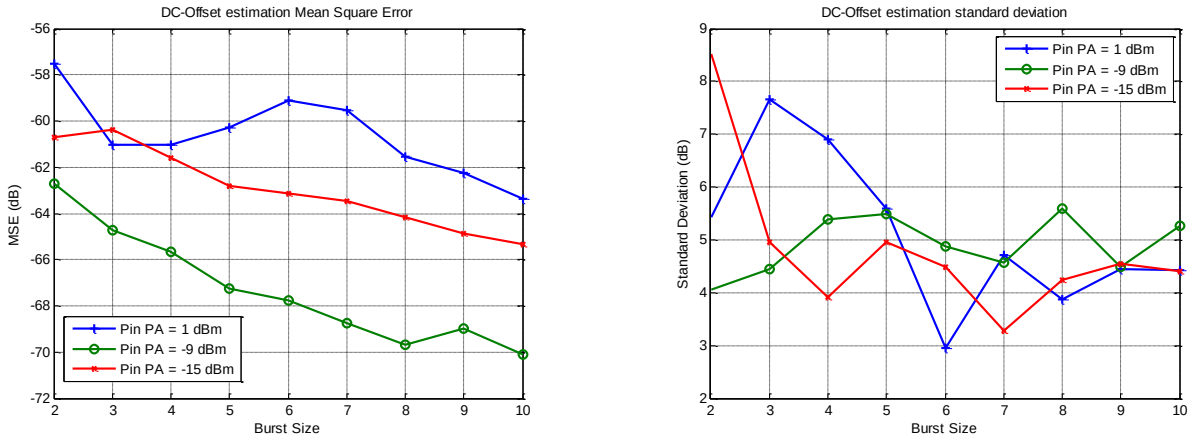
Figure 19: (a) Mean and (b) standard deviation of the conversion gain after IQ imbalance compensation according to the number of OFDM symbols and to the PA model

3.3.2.5 Transmitter DC-Offset

We have set the real (imaginary) part of the DC-Offset 10 dB below the mean amplitude of the real (imaginary) part of the signal to be transmitted. Figure 20 gives the mean square error and standard deviation of the DC-Offset estimator according to the number of OFDM symbols used for estimations, and according to the PA input power.

One can observe that increasing the number of symbols slightly improves the performance. As noted in section 3.3.2.3 and for the same reasons, the best performance is given for the medium mean input power of -9 dBm.

In SACRA we recommend to estimate the DC-Offset using only 2 OFDM symbols since the trade-off between complexity and performance is very good.



(a) (b)

Figure 20: (a) Mean Square Error and (b) standard deviation of the DC-Offset estimation according to the number of OFDM symbols and to the PA model

3.3.2.6 Polynomial Order "K" and Memory Depth "Q"

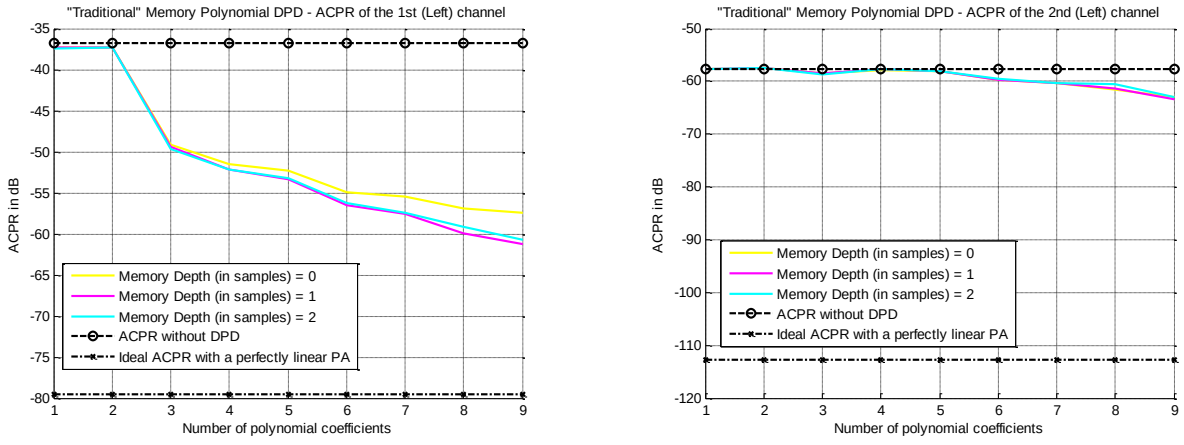
In this section, we address the polynomial order and memory depth setting according to the use of traditional or orthogonal memory polynomial, and also according to the fact of taking into account only odd terms or odd and even terms. The RF impairments are assumed to be perfectly estimated and compensated.

We choose to assess performance only for the higher PA input power of 1 dBm because it is usually for powers close to saturation that the PA exhibits most of its non linearity, and where predistortion is the most useful.

We have parameterized the ACE algorithm such as the OFDM signal used for the predistortion computation has a PAPR 1 dB greater than the signal to be predistorted. This allows to be sure that there exists in inverse for the whole range of instantaneous powers.

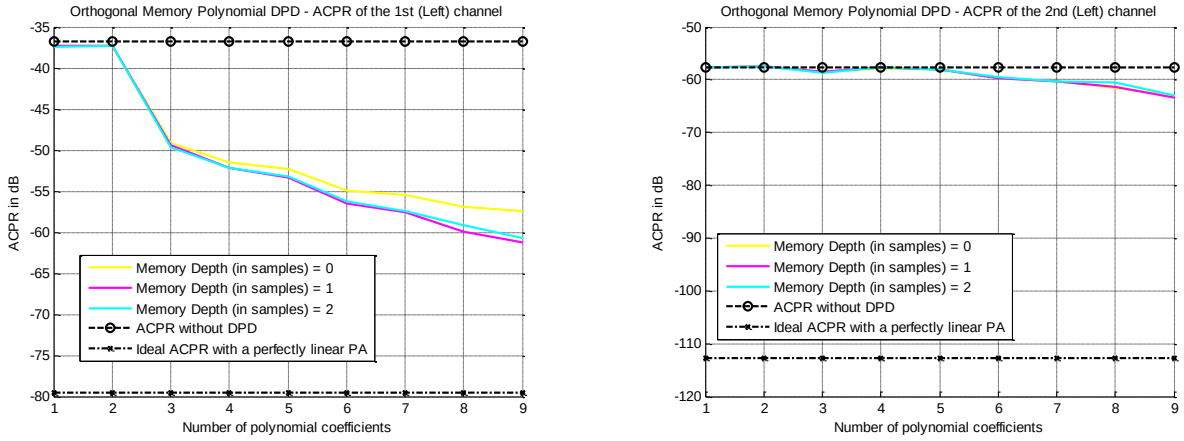
Figure 21, Figure 22, Figure 23 and Figure 24 show the ACPR of the first and second left channels for respectively i) traditional polynomials with odd terms only, ii) orthogonal polynomials with odd terms only , iii) traditional polynomials with both odd and even terms, iv) orthogonal polynomials with both odd and even terms. One can observe that:

- For a given order approach technique (odd terms only or both odd and even terms), traditional and orthogonal polynomials give exactly the same performance.
- The inclusion of even orders allows improving slightly the performance of the first and second adjacent channels. The reason for such behaviour is the fact that the PA has been accurately modelled using both odd and even orders.
- The PA experience very low memory effects. Taking into account the memory effects of only 1 sample allows decreasing by about 5 dB the ACPR performance of the first adjacent channel.
- In order to reach -50 dBc of ACPR, only 3 polynomial coefficients are required for the odd orders approach, when 4 coefficients are required for the odd and even orders approach.
- The second adjacent channel is only very slightly improved.



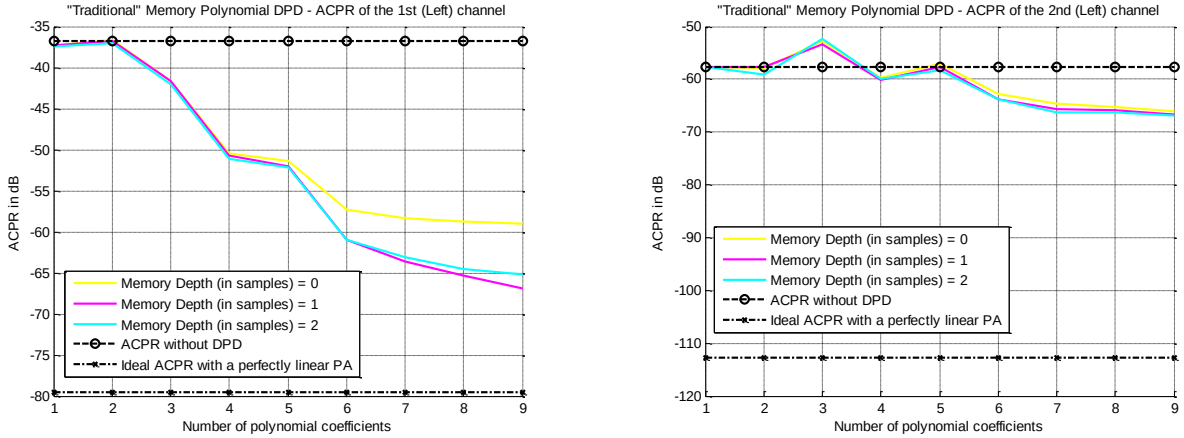
(a) (b)

Figure 21: ACPR of (a) the first left adjacent channel and (b) the second left adjacent before and after DPD according to the number of coefficients and memory depth for traditional memory polynomial with odd terms only



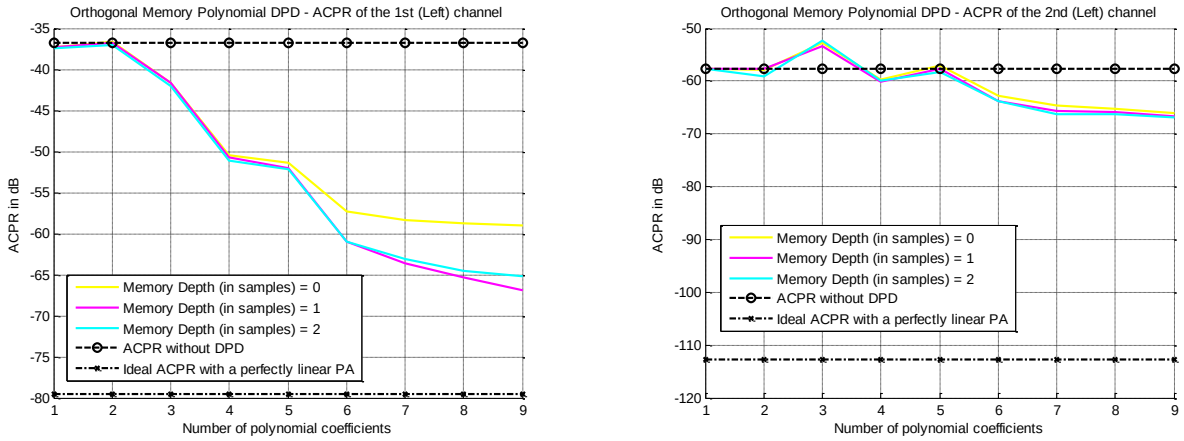
(a) (b)

Figure 22: ACPR of (a) the first left adjacent channel and (b) the second left adjacent before and after DPD according to the number of coefficients and memory depth for orthogonal memory polynomial with odd terms only



(a) (b)

Figure 23: ACPR of (a) the first left adjacent channel and (b) the second left adjacent before and after DPD according to the number of coefficients and memory depth for traditional memory polynomial with both odd and even terms

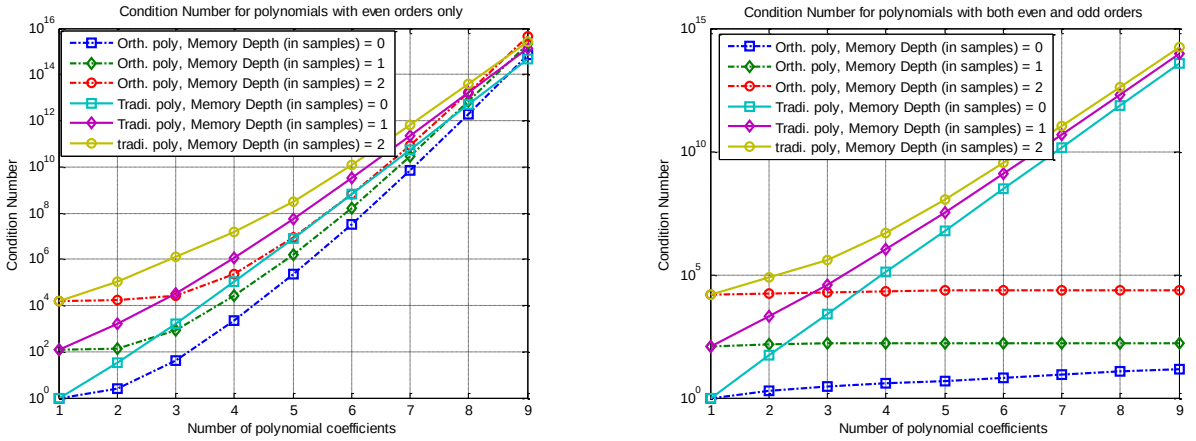


(a) (b)

Figure 24: ACPR of (a) the first left adjacent channel and (b) the second left adjacent before and after DPD according to the number of coefficients and memory depth for orthogonal memory polynomial with both odd and even terms

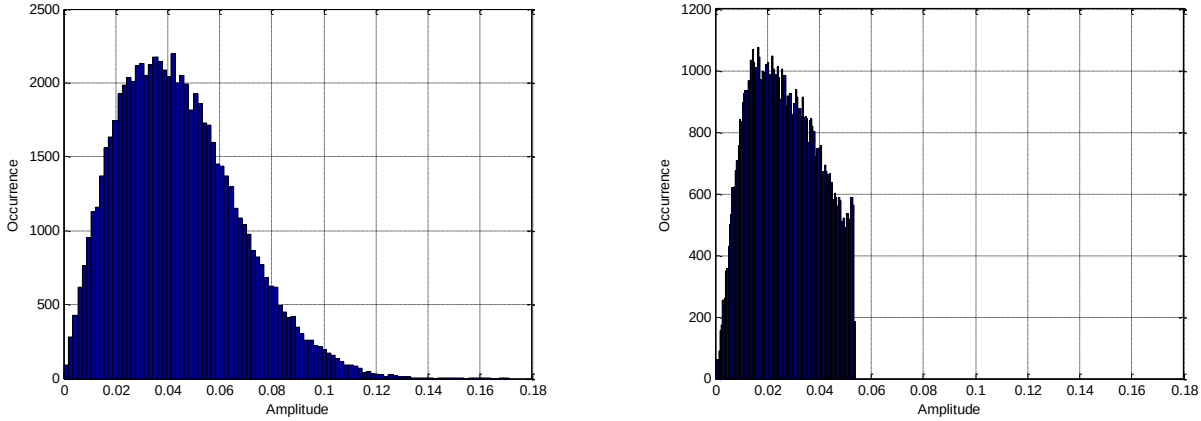
The condition number of the matrix to be inverted (see eq. (10)) according to the polynomial number of coefficients and memory depth are shown by Figure 25 (a) for odd terms only. We can see that the condition number is decreased thanks to orthogonal polynomials but is still important. For example, for a polynomial number of coefficients of 6 and a memory depth of 1 sample, the condition number for traditional polynomial is 3.3×10^{-9} whereas it is 1.6×10^{-8} for orthogonal polynomial. In [27], the authors have shown that the condition number could be very close to 1 at the condition that signal strictly follows a complex Gaussian process. Figure 26 presents the distribution of the amplitude of (a) an OFDM signal and (b) an OFDM signal that has been PAPR reduced by ACE, and at the output of a PA. The amplitude distribution of a complex Gaussian process is a Rayleigh distribution. We can see that it is the case for the OFDM signal without PAPR reduction, and simulation have shown that the condition number of the matrix built with this signal is very close to 1. In SACRA, the signal at the PA outputs clearly does not follow a Rayleigh distribution because i) the compression/saturation area of the PA modifies the distribution, ii) the PAPR has been reduced. For these reasons, we can explain why the condition number of orthogonal polynomials with odd terms only is far from 1.

The condition number of the matrix to be inverted (see eq. (10)) according to the polynomial number of coefficients and memory depth are shown by Figure 25 (b) for both odd and even terms. We can see that the condition number is drastically decreased thanks to orthogonal polynomials. For example, for a polynomial number of coefficients of 6 and a memory depth of 1 sample, the condition number for traditional polynomial is 1.2×10^{-9} whereas it is 164 for orthogonal polynomial.



(a) (b)

Figure 25: Condition number of traditional and orthogonal memory polynomials for (a) odd terms only and (b) both odd and even terms according to the number of polynomial coefficients and to the memory depth



(a) (b)

Figure 26: Amplitude distribution of (a) an OFDM signal at the output of a linear PA and (b) a PAPR reduced OFDM signal (ACE) at the output of a real PA with compression

3.3.2.7 Least Square Estimation Size

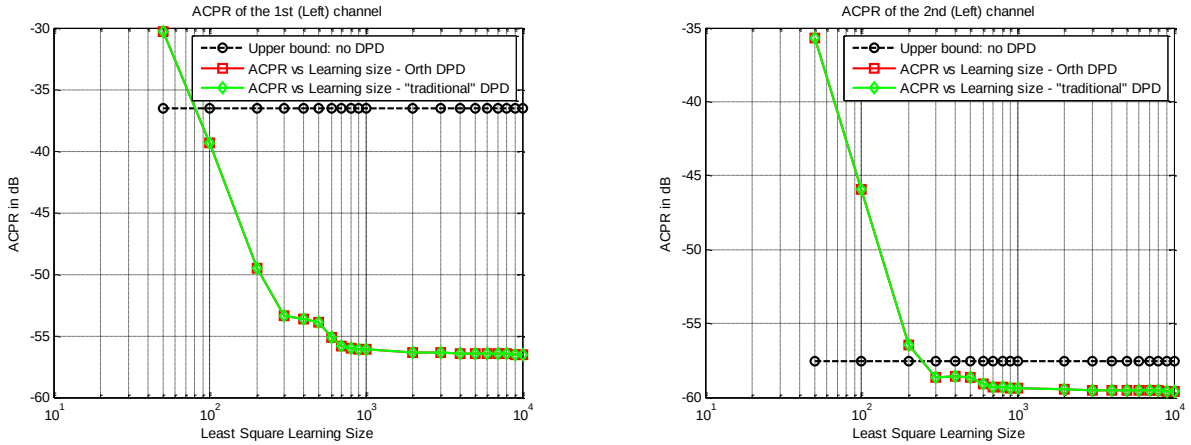
The size of the pattern used for the DPD estimation is critical since it fixes the complexity of the LS matrix operations. The simulation setup is the same as in section 3.3.2.6. The number of polynomial coefficients is set to 6 and the memory depth to 1 sample.

Figure 27 and Figure 29 show the ACPR according to the learning size of the first and second left channels for respectively polynomials with odd terms only, and polynomials with both odd and even terms. Figure 28 and Figure 30 give the standard deviations. One can observe that:

- For a given approach (odd orders only, or both odd and even orders), the performance of the traditional and orthogonal polynomials are equivalent

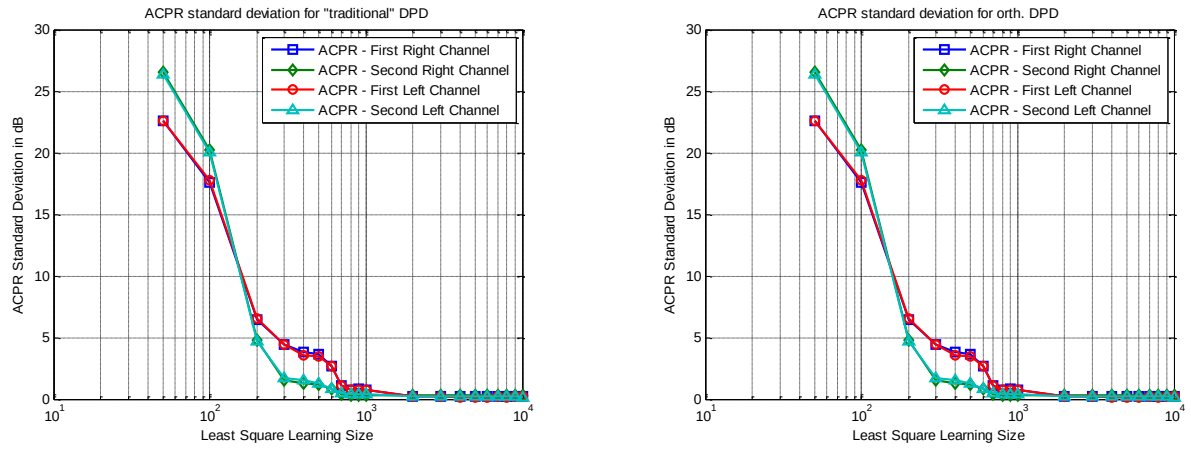
- For pattern size below 200 samples, the instantaneous PA input/output power statistics can be considered as poor and the predistortion is not capable to provide accurate estimations: The linearization performance is not good.
- ACPRs reach a floor for learning size equal or greater than 400 samples. The floors of the first and second adjacent channels are 5 dB lower for the approach which takes into account both odd and even terms.
- Second adjacent channel ACPRs with DPD are worst than the ACPR without DPD for learning size respectively lower than 300 and 100 samples for respectively odd orders only and odd and even orders.
- Standard deviations are very low (less than 3 dB) for learning size greater than 400 samples.

In SACRA we recommend to set the learning size to 400 samples.



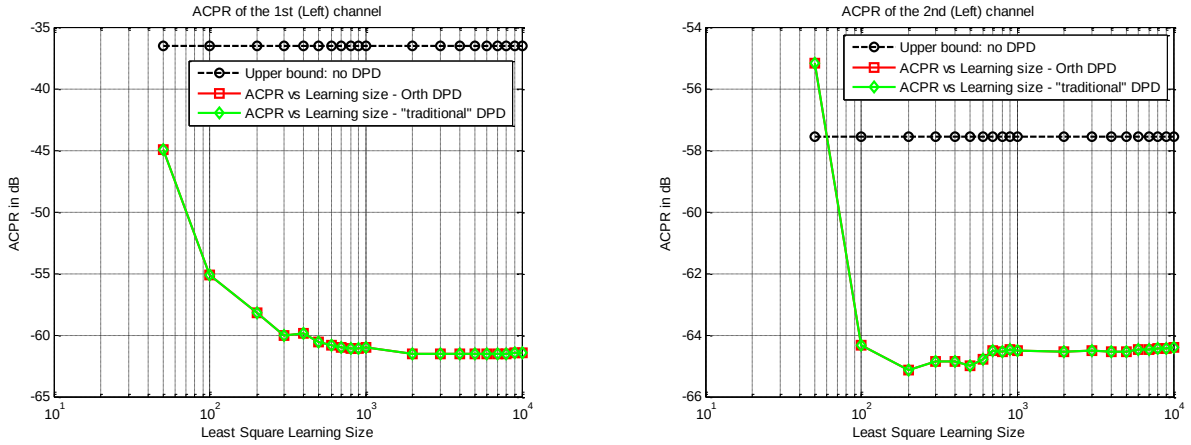
(a) (b)

Figure 27: ACPR of (a) the first left adjacent channel and (b) the second left adjacent before and after DPD according to the learning size used for Least Square DPD estimation. Traditional and orthogonal memory polynomial with odd terms only



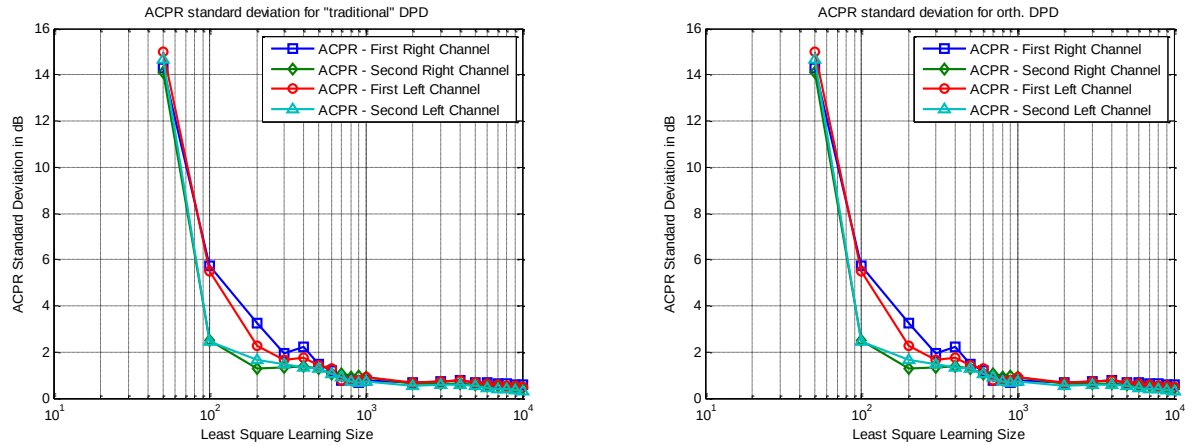
(a) (b)

Figure 28: Standard deviation of the ACPR of (a) the traditional memory polynomials and (b) the orthogonal memory polynomials after DPD according to the learning size used for Least Square DPD estimation. Polynomials with odd terms only



(a) (b)

Figure 29: ACPR of (a) the first left adjacent channel and (b) the second left adjacent before and after DPD according to the learning size used for Least Square DPD estimation. Traditional and orthogonal memory polynomial with both odd and even terms



(a) (b)

Figure 30: Standard deviation of the ACPR of (a) the traditional memory polynomials and (b) the orthogonal memory polynomials after DPD according to the learning size used for Least Square DPD estimation. Polynomials with both odd and even terms

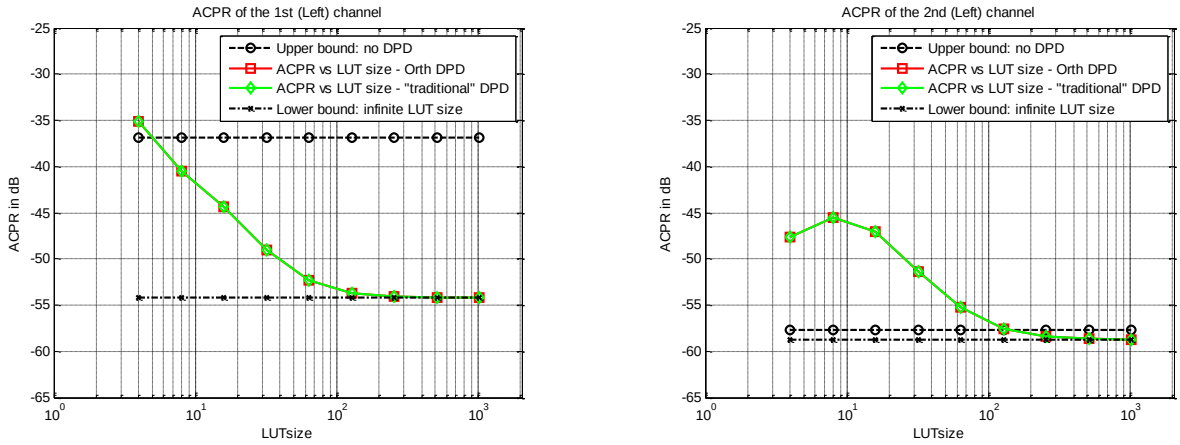
3.3.2.8 LUT size

In this section, we assess the LUTs size impact on ACPR performance. The simulation setup is the same as in section 3.3.2.7. The learning size is set to 400 samples. Since we have set the memory depth to 1 sample, the number of LUTs is 2.

Figure 31 and Figure 33 show the ACPR according to the LUTs size of the first and second left channels for respectively polynomials with odd terms only, and polynomials with both odd and even terms. Figure 28 and Figure 30 give the standard deviations. One can observe that:

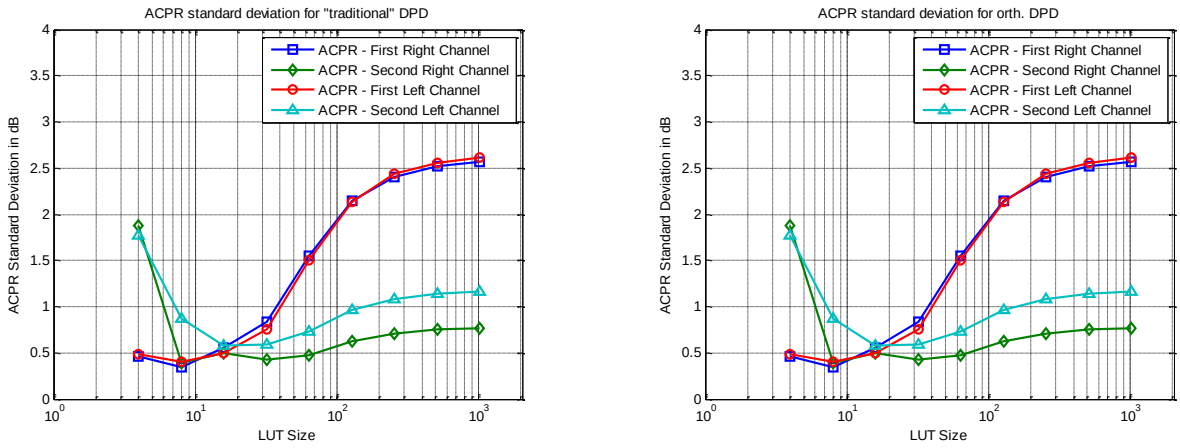
- For a given approach (odd orders only, or both odd and even orders), the performance of the traditional and orthogonal polynomials are equivalent
- Second adjacent channel ACPRs with DPD are worst than the ACPR without DPD for LUTs size respectively lower than 256 and 64 for respectively odd orders only and odd and even orders.
- ACPRs drop very quickly when LUTs size increases. LUTs size of 256 entries or greater has a very low (<1dB in terms of ACPR) with regard to impact on optimal performance (infinite LUTs size).

In SACRA, we believe that a good compromise between performance and complexity is to use 2 LUTs with 256 entries.



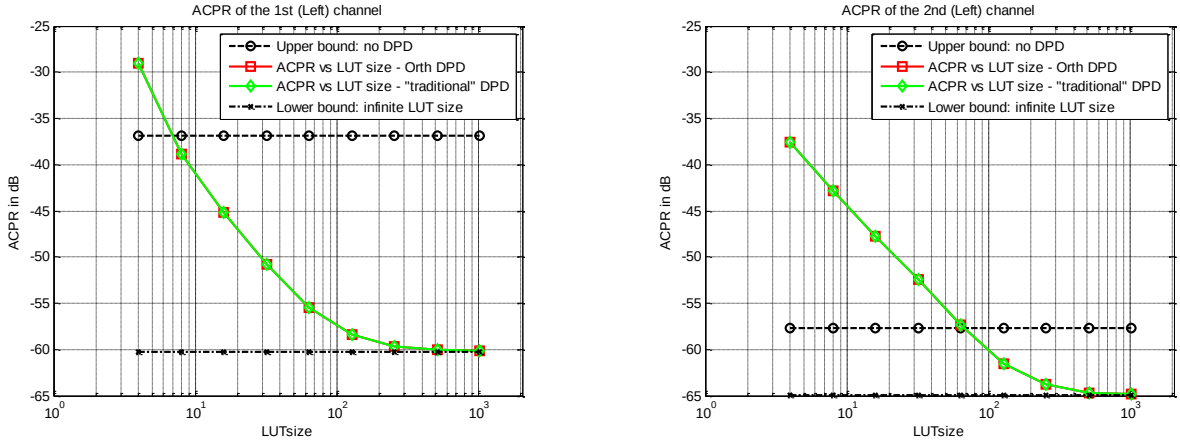
(a) (b)

Figure 31: ACPR of (a) the first left adjacent channel and (b) the second left adjacent after DPD terms according to the LUTs size. Traditional and orthogonal memory polynomial with odd terms only



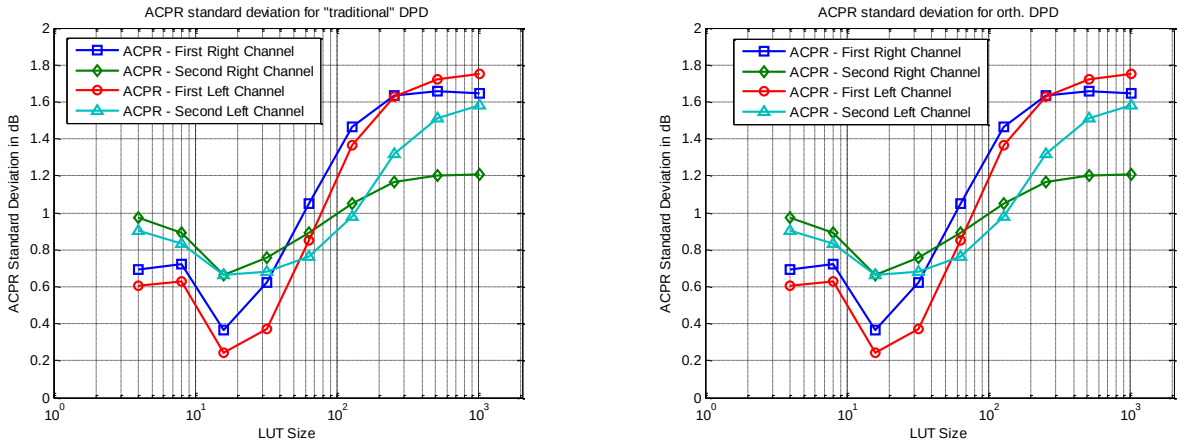
(a) (b)

Figure 32: Standard deviation of the ACPR of (a) the traditional memory polynomials and (b) the orthogonal memory polynomials after DPD according to the LUTs size. Polynomials with odd terms only



(a) (b)

Figure 33: ACPR of (a) the first left adjacent channel and (b) the second left adjacent after DPD terms according to the LUTs size. Traditional and orthogonal memory polynomial with both odd and even terms



(a) (b)

Figure 34: Standard deviation of the ACPR of (a) the traditional memory polynomials and (b) the orthogonal memory polynomials after DPD according to the LUTs size. Polynomials with both odd and even terms

3.3.3 Impact of RF impairments on DPD

In this section, we address the impact of the RF impairments on DPD capabilities and performance. Figure 35 refers to the AM/AM and AM/PM curves of the PA, of the DPD and of the association of the DPD with the PA when the RF impairments are perfectly digitally compensated. In consequence, this figure is the reference and will be compared with the curves obtained in sections 3.3.3.1, 3.3.3.2 and 3.3.3.3 when RF impairments are not compensated.

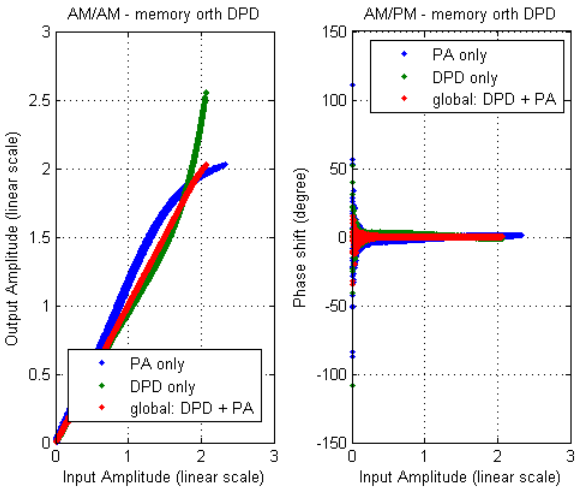
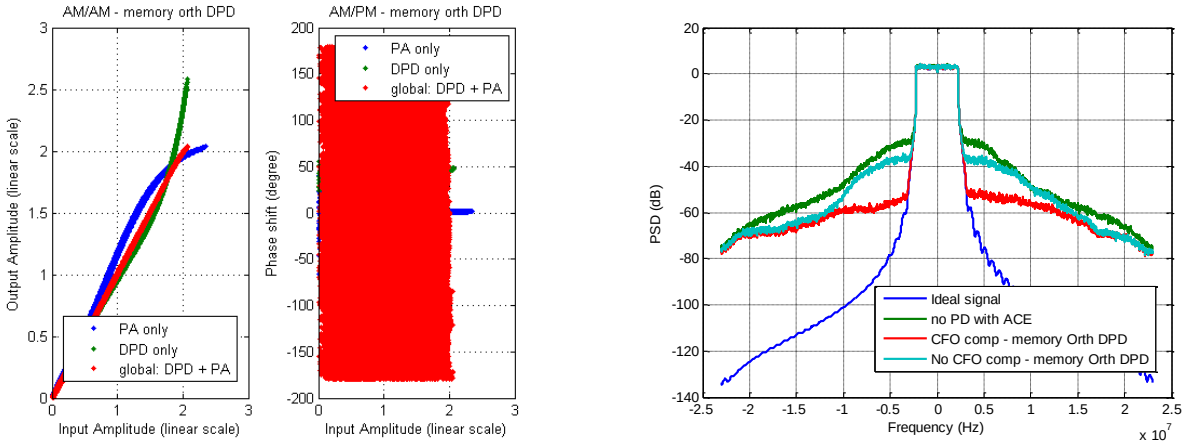


Figure 35: (a) AM/AM and AM/PM curves for ideal CFO, IQ imbalance and DC-Offset compensations

3.3.3.1 CFO

We take the example of a terminal which is impaired only by a CFO of 1.5 kHz (referring to a tolerance of 0.6 ppm at 2.5 GHz). We can see in Figure 36 (a) that the DPD is capable of compensating the modulus buts is unable to maintain a constant phase modulus according to the input power. As pointed out by eq. (3), the CFO creates a phase shift at the PA input which varies with time. In other words, for a given input power, the CFO creates a phase shift at t=t1 which is different of the phase shift at t=t2. Figure 36 (b) shows the impact of the phase variation on PSD. The PSD of a DPD without CFO compensation is less 10 dB below the PSD of a non-linearized PA. The compensation of the CFO allows improving by 20 dB the linearity in this example.

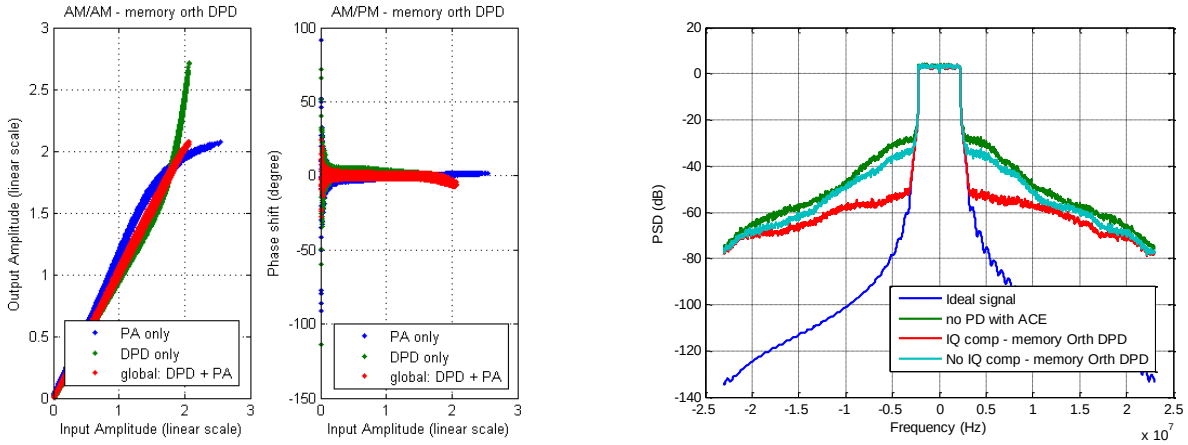


(a) (b)

Figure 36: (a) AM/AM and AM/PM curves without CFO estimation/compensation, (b) Impact on Power Spectral Density of the CFO compensation

3.3.3.2 Transmitter IQ imbalance

We assume that the terminal is only impaired by a phase imbalance of 3 degrees and a gain imbalance of 0.25 dB. Figure 37 shows the impact of the IQ imbalance on AM/AM and AM/PM curves, and PSD. From eq. (2), we can observe that the phase and modulus of the input power don't depend on $e(n)$ but depend on a linear combination of $e(n)$ with $e(n)^*$. As a result, the AM/AM and AM/PM curves of the resulting combination of the DPD with the PA are thicker than in Figure 35 where the IQ imbalance is perfectly compensated. One can observe also that the AM/PM is not strictly linear, especially for the greatest instantaneous input powers. The PSD of an uncompensated IQ imbalance DPD only improves by 5 dB the linearity of a non-linearized PA.

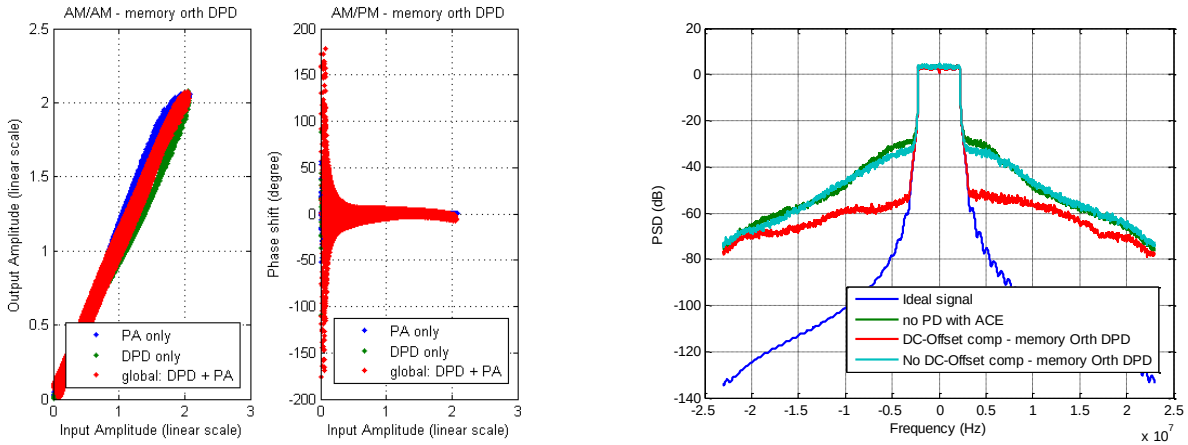


(a) (b)

Figure 37: (a) AM/AM and AM/PM curves without IQ imbalance estimation/compensation, (b) Impact on Power Spectral Density of the IQ imbalance compensation

3.3.3.3 Transmitter DC-Offset

We assume that the terminal is only impaired by a DC-Offset 10 dB below the mean amplitude of the real (imaginary) part of the signal to be transmitted. Figure 38 shows the impact of the DC-Offset on AM/AM and AM/PM curves, and PSD. From eq.(1), we can observe that the DC-Offset modify the phase and modulus of the input power. As a result, the AM/AM and AM/PM curves of the resulting combination of the DPD with the PA are thicker than in Figure 35 where the DC-Offset is perfectly compensated. One can observe also that the AM/PM is not strictly linear, especially for the greatest instantaneous input powers. The PSD of an uncompensated DC-Offset DPD only improves by 3 dB the linearity of a non-linearized PA.



(a) (b)

Figure 38: (a) AM/AM and AM/PM curves without DC-Offset estimation/compensation, (b) Impact on Power Spectral Density of the DC-Offset compensation

3.4 GLOBAL PERFORMANCE ASSESSMENT

In this section, we address the global impact of the association of the ACE with DPD on the performance. We distinguish two types of performance:

- The in band linearity by means of EVM (section 3.4.2), uncoded BER (section 3.4.3) and total degradation (section 3.4.4)
- The out of band linearity by means of ACPRs (section 3.4.5)

3.4.1 Parameters

The parameters used for the following simulations are given in Table 6 and Table 8.

Parameters	Value
Polynomial type	Orthogonal
Polynomial order	Odd terms only
Number of polynomial coefficients (K)	6
Memory depth (Q)	1
LUTs size	128
PA model	SKY77707, Input power = 1 dBm

Table 8: Global performance assessment parameters

The BackOff is defined as the mean PA output power deviation from the PA instantaneous output power saturation. The DPD computation is not possible if part of the instantaneous power falls into the PA saturation area. If one wants to compute and apply the DPD for OFDM signals without PAPR reduction, the BackOff shall be large due to the fact the corresponding CCDF of an unmodified OFDM signal is also very large. If the DPD is used with ACE, the BackOff can be decreased according to the PAPR reduction capabilities of QPSK, 16QAM and 64QAM. In order to fairly compare the performance, we have performed simulations when the DPD is disabled. In

these cases, we have chosen to set the same BackOff value as the one defined when the DPD and ACE are jointly used. The BackOff setting is given in Table 9 below:

	PA only		DPD and PA	
	Without ACE	With ACE	Without ACE	With ACE
QPSK	6 dB	6 dB	10.7 dB	6 dB
16QAM	7 dB	7 dB	10.7 dB	7 dB
64QAM	9.5 dB	9.5 dB	10.7 dB	9.5 dB

Table 9: BackOff setting

3.4.2 Error Vector Magnitude

The error vector magnitude is defined as the gap between the transmitted QPSK, 16QAM or 64QAM points and the ideal points:

$$EVM(\%) = 100.\left(\frac{\sum_{n=[1;N\max]} \sum_{k=[-Nu/2;-1]\bigcup[1;Nu/2]} \left| \frac{W_k(n)}{H_k} - X_k(n) \right|^2}{\sum_{n=[1;N\max]} \sum_{k=[-Nu/2;-1]\bigcup[1;Nu/2]} \left| X_k(n) \right|^2} \right)$$

In SACRA, the EVM is computed at the PA output, with an ideal receiver which perfectly compensates the modem channel H_k. Table 10 presents the EVM with/without ACE, with/without DPD according to QPSK, 16QAM and 64QAM. Since the ACE PAPR reduction is obtained thanks to constellation modifications, we have included two EVM definitions: the first one uses the constellation before the ACE modifications as a reference, and the second uses the constellation after ACE modifications. When the DPD is not used, we can observe that the EVM is larger for lower constellation orders. This can be explained by the fact that the BackOff decreases with the modulation order, and as a result, a larger part of the signal falls into the PA compression and saturation areas. When the DPD and ACE are activated, EVMs are almost perfectly compensated, but one can observe that it remains a small level of EVM of about 0.5% when ACE is not used. A reason for that is the fact that the probability that the PAPR of an OFDM signal without ACE is greater than the 10.7 dB of BackOff is not null.

	PA only			DPD and PA		
	Without ACE	ACE (ref. without ACE)	ACE (ref. with ACE)	Without ACE	ACE (ref. without ACE)	ACE (ref. with ACE)
QPSK	4.55 %	14.89 %	2.56 %	0.57 %	14.93 %	0.15 %
16QAM	3.15 %	15.35 %	1.61 %	0.43 %	14.89 %	0.07 %
64QAM	2.00 %	2.65 %	1.91 %	0.51 %	1.78 %	0.31 %

Table 10: EVM with/without ACE, with/without DPD according to QPSK, 16QAM and 64QAM

3.4.3 Bit Error Rate

In this section, we present the uncoded BERs over AWGN channel obtained with/without DPD and with/without ACE for QPSK, 16QAM and 64QAM, see respectively Figure 39, Figure 40 and Figure 41.

The constellation extension of the ACE can be seen as an adding signal technique: for a given uncoded BER, the use of ACE requires a slight signal power increase (typically less than 1 dB for the worst case). Nevertheless, the BER is not degraded thanks to the fact that the ACE does not decrease the Euclidean distance of the constellation. The effect of a real PA on performance is larger without the use of ACE than with the use of ACE. Finally, one can observe that the DPD slightly improves the in band linearity.

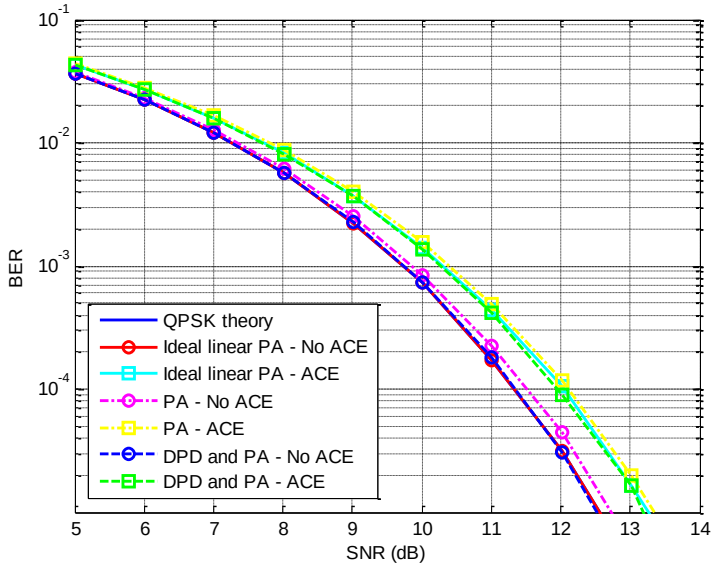


Figure 39: Uncoded BER for QPSK with/without ACE, with/without DPD for AWGN channel

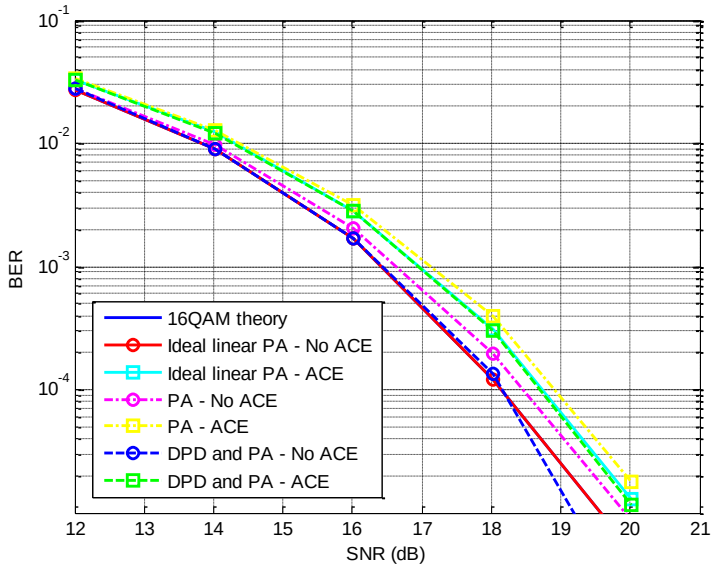


Figure 40: Uncoded BER for 16QAM with/without ACE, with/without DPD for AWGN channel

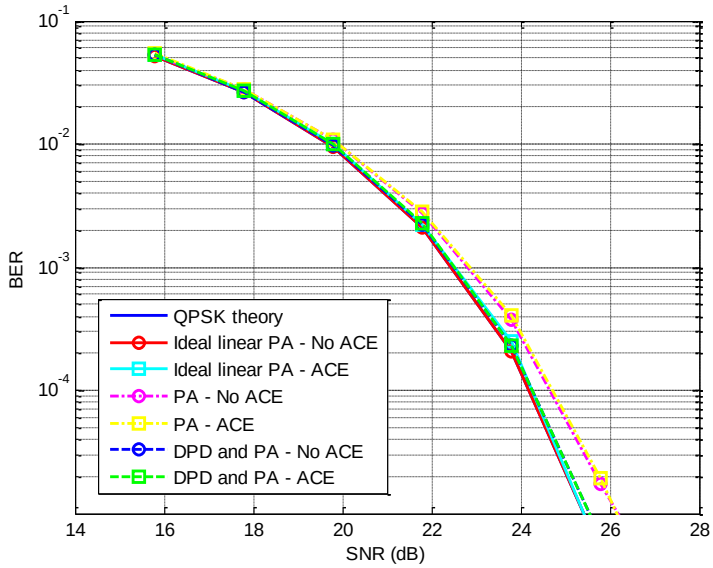


Figure 41: Uncoded BER for 64QAM with/without ACE, with/without DPD for AWGN channel

3.4.4 Total Degradation

The total degradation is defined as the sum (in decibels) of the amplifier BackOff and of the increment ΔSNR in the ratio SNR required to maintain a given BER with respect to the situation of perfectly linear PA:

$$TotalDegradation(dB) = BackOff(dB) + \Delta SNR(dB)$$

Total degradation basically refers to the maximum sustainable path loss degradation

Table 11 and Table 12 respectively refers to the total degradation for BER=10^{-2} and BER=10^{-5}. One can observe that the total degradation improvement obtained thanks to the association of the ACE and DPD is roughly the PAPR reduction capabilities.

		PA only		DPD and PA	
		Without ACE	With ACE	Without ACE	With ACE
QPSK	Back Off	6 dB	6 dB	10.7 dB	6 dB
	ΔSNR	0 dB	0.4 dB	0 dB	0.4 dB
	Total Degradation	6 dB	6.4 dB	10.7 dB	6.4 dB
16QAM	Back Off	7 dB	7 dB	10.7 dB	7 dB
	ΔSNR	0 dB	0.4 dB	0 dB	0.4 dB
	Total Degradation	7 dB	7.4 dB	10.7 dB	7.4 dB
64QAM	Back Off	9.5 dB	9.5 dB	10.7 dB	9.5 dB
	ΔSNR	0.1 dB	0.1 dB	0 dB	0 dB
	Total Degradation	9.6 dB	9.6 dB	10.7 dB	9.5 dB

Table 11: Total Degradation for uncoded BER = 10^{-2} with/without ACE, with/without DPD for AWGN channel according to QPSK, 16QAM and 64QAM

		PA only		DPD and PA	
		Without ACE	With ACE	Without ACE	With ACE
QPSK	Back Off	6 dB	6 dB	10.7 dB	6 dB
	ΔSNR	0.8 dB	1.4 dB	0 dB	1 dB
	Total Degradation	6.8 dB	7.4 dB	10.7 dB	7 dB
16QAM	Back Off	7 dB	7 dB	10.7 dB	7 dB
	ΔSNR	0.2 dB	0.8 dB	0 dB	0.5 dB
	Total Degradation	7.2 dB	7.8 dB	10.7 dB	7.5 dB
64QAM	Back Off	9.5 dB	9.5 dB	10.7 dB	9.5 dB
	ΔSNR	0.3 dB	0.3 dB	0 dB	0 dB
	Total Degradation	9.8 dB	9.8 dB	10.7 dB	9.5 dB

Table 12: Total Degradation for uncoded BER = 10^{-5} with/without ACE, with/without DPD for AWGN channel according to QPSK, 16QAM and 64QAM

From a pure total degradation point of view, one can observe that the best performance is obtained without DPD and with ACE. But in these case, and as pointed out in section 3.4.5, the out of band linearity is not acceptable.

The joint use of ACE with DPD allows to drastically improving the out of band linearity with almost similar total degradation than without any processing.

3.4.5 Adjacent Channel Power Ratio

In this section, we address the out of band linearity. We use the ACPR as a factor of merit. Table 13 gives the ACPR of the first and second adjacent channel with/without ACE and with/without DPD according to QPSK, 16QAM and 64QAM.

		PA only		DPD and PA	
		Without ACE	With ACE	Without ACE	With ACE
QPSK	First Channel	-36 dBc	-37 dBc	-50 dBc	-55 dBc
	Second Channel	-57 dBc	-58 dBc	-57 dBc	-58 dBc
16QAM	First Channel	-38 dBc	-40 dBc	-50 dBc	-55 dBc
	Second Channel	-57 dBc	-58 dBc	-58 dBc	-59 dBc
64QAM	First Channel	-41 dBc	-41 dBc	-50 dBc	-56 dBc
	Second Channel	-57 dBc	-57 dBc	-60 dBc	-60 dBc

Table 13: ACPR of the first and second channel with/without ACE, with/without DPD according to QPSK, 16QAM and 64QAM

3.5 CONCLUSIONS

3.5.1 Performance and energy efficiency

One of our concerns in SACRA is the linearity and the energy efficiency of the terminal (also called user equipment) without degradation of the global link budget of the communication.

It is known that the use of the DPD without PAPR reduction algorithm has low to moderate efficiency in terms of linearity for OFDM signals which have inherently a very large power distribution. The use of a PAPR reduction algorithm with DPD allows ensuring that the DPD is capable to sustain a high linearity until a certain level of backoff. This minimum reachable backoff refers to the resulting PAPR after PAPR reduction. The less is the backoff, the greater is the PA efficiency, and as a result the overall transmitter efficiency. In this deliverable, we have shown that the ACE algorithm used as a PAPR reduction for OFDM signals decreases significantly the PAPR for QPSK and 16QAM constellations which greatly improved the efficiency, nevertheless the PAPR reduction capabilities is slight for 64QAM modulation.

The DPD that we propose for SACRA is adaptive in order to be capable to track the PA behaviour variation due for example to aging, temperature, frequency, etc… The approach that we have chosen for the DPD has been selected having in mind to reduce as much as possible the complexity while keeping excellent performance: we propose a DPD based on LUTs with relatively low number of entries (256), orthogonal memory polynomial with only odd orders. Since the amount of PA memory effects is low, only few additional coefficients (with regard to the memory less case) are necessary to cope with these memory effects.

In SACRA, we consider terminal which, by definition, shall be low cost, have a small form factor, implying that the analogue components are imperfect and has several impairments. We have shown that some of the possible radiofrequency impairments such as CFO, IQ imbalance and DC-offset have a great impact on the DPD linearization capability: if these impairments are not minimized or compensated, then the DPD is inefficient. We propose in SACRA a fully digital approach that consists in estimating and compensating the CFO, IQ imbalance and DC-offset in baseband. The algorithms that we have developed take benefit of i) the feedback loop that is already required for adaptive DPD, ii) the functions already present in baseband (for instance FFT). As a result, we believe that the additional complexity required is not prohibitive. Finally, as the DPD, these algorithms are adaptive and can track behaviour modifications: they can be activated as soon as the re-estimation is necessary. Further studies on PAPR reduction and DPD will be conducted in the following of the project to address implementation issues.

3.5.2 Feasibility analysis of digital PAPR reduction

The proposed PAPR reduction strategy has been analysed and checked against the functional and performance characteristics of the target baseband processor. All involved digital signal processing operations that can be hardware-accelerated are vector operations (component wise additions, products, etc.) or Fourier transforms. These operations can thus benefit the hardware acceleration of the Front End Processor (FEP) DSP unit of the baseband processor. Its performance is quite classical and can thus be considered as a realistic comparison point. Fourier transforms are computed using a radix-4 Fast Fourier Transform (FFT), two radix-4 operations per clock cycle (cc). The other vector operations process two vector components per cc. The conclusions could be slightly different with another target processor. In order to stay generic, we assume in the baseband processor with the following characteristics:

- 2^n Fourier transforms are computed in $A\times\lceil n/2\rceil\times 2^n$ cc (A=1/8 with FEP). This simply means that the target processor computes Fourier transforms using a radix-4 FFT, which is a very

reasonable assumption. Some processors could use a radix-2 FFT instead but the results would not be very different ($A \times n/2 \times 2^n$ instead of $A \times \lceil n/2 \rceil \times 2^n$).

- All other vector operations on N-components vectors are computed in $B \times N$ cc (B=1/2 with FEP). This is again a very reasonable assumption.
- The available vector operations are equivalent to that of FEP. This assumption is a very strong one because FEP offers several powerful features that are usually not available). As a consequence, our estimated cc counts shall be considered as minimum counts.
- The overheads due to initialization intervals between operations and other side effects are negligible. This assumption is usually true as long as the processed vectors are long enough, which is the case in PAPR reduction because of the significant oversampling factor.

The different involved processing steps fall in one of three kinds:

- Operations that are natively available as one of the built-in operations of the FEP. Most operations fall into this category.
- Operations that cannot be hardware accelerated by the baseband processor (square roots, inverts, ...) but are very low rate and can thus be handled in software by the main micro-controller. They are not taken into account in the performance analysis because of their very low rate.
- Demanding operations with significant processing power that cannot be hardware accelerated by the baseband processor. Step 5 (modulation constraints) is the only such operation. It consists in a 2D value comparison followed by a selection.

In the following we explain how the different steps of the PAPR reduction would be implemented and what the overhead (in cc) would be, compared to a classical LTE transmission without PAPR reduction. 2^n is the vector size without oversampling (n=9 for 5MHz bandwidth), L is the selected oversampling factor (L=4 in the preceding performance simulations), $2^N = L \times 2^n$ is the vector size with oversampling (N=11 for 5MHz bandwidth and L=4) and I is the number of iterations (I=3 in the preceding performance simulations with QPSK and 16QAM modulations). Step by step detailed analysis:

- Step 1 is implemented by the built-in FFT. The largest computable Fourier transform is 4096 points in the current version of the baseband processor. With the value of L retained for the simulations (L=4), this limits the application of PAPR to bandwidths less or equal 10MHz. The overhead is only due to the oversampling factor: $A \times \lceil N/2 \rceil \times 2^N - A \times \lceil n/2 \rceil \times 2^n$ cc.
- Step 2 (computation of PAPR of oversampled OFDM symbols) is based on the Component-Wise square of Modulus (CWM) operation of FEP, plus a few simple extra software operations. Max and sum of the $|b_n|^2$ vector are computed on the fly, thanks to the Sum, Min-max, Argmin-argmax (SMA) feature of FEP. Overhead: one 2^N CWM per iteration = $I \times B \times 2^N$ cc.
- Step 3 (clipping operations) is implemented by a Component-Wise Lookup (CWL) on the $|b_n|^2$ vector, followed by Component-Wise Product (CWP) and a Component-Wise Add (CWA) for the computation of the d_n. Note: the CWL operation filters the components of a real vector through a look-up table, with linear interpolation. This is one of the advanced features of FEP that are usually not available on more classical Digital Signal Processing (DSP) or vector processors. In this case it is used to compute the inverse of the $|b_n|$ from the $|b_n|^2$. This operation being tabulated, it introduces a loss of accuracy which impact on the algorithm's performance has not yet been evaluated (more generally, the impact of the

fixed point representation of data samples has not been evaluated). Overhead: one 2^N CWM, one 2^N CWP and one 2^N CWA per iteration = $3\times I\times B\times 2^N$ cc.

- Step 4 uses the built-in FFT. Overhead: one 2^N FFT per iteration = $I\times A\times\lceil N/2\rceil\times 2^N$ cc.
- Step 5 (applying the modulation constraints) cannot be hardware accelerated with FEP and would have to be computed in software or a new dedicated operation would have to added to FEP. There are no budgeted resources in the SACRA project to design such a new operation but, for overhead estimation, we assume that it is a regular vector operation. The actual overhead would be significantly larger without such an optimization as the software implementation would be much slower. Overhead: one such custom operation per iteration = $I\times B\times 2^N$ cc.
- Step 6 uses the built-in FFT. Overhead: one 2^N FFT per iteration = $I\times A\times\lceil N/2\rceil\times 2^N$ cc.
- Step 7 is computed with one built-in 2^N CWM to compute the sum of the $|h_n|^2$, one built-in 2^N CWP to compute the β factor, one built-in 2^N CWP and one built-in 2^N CWA to update the b_n vector, plus a few simple software operations, per iteration. That is, four 2^N vector operations per iteration. Overhead = $4\times I\times B\times 2^N$ cc.

Conclusion: on the functional point of view the proposed PAPR reduction algorithm could be implemented on the target baseband processor but one of the computation intensive steps would have to be computed in software.

The total overhead is significant, even with optimistic assumptions about the available operations: $(2\times I+1)\times A\times\lceil N/2\rceil\times 2^N - A\times\lceil n/2\rceil\times 2^n + 9\times I\times B\times 2^N$ cc. With a FEP-like processor (A=1/8 and B=1/2) and with the retained parameter values in the preceding simulations for QPSK and 16QAM modulations (n=9, L=4, N=11, I=3), it would be about 38000 cc, while a processor running at a 500 MHz clock frequency would have only 36000 cc to completely process a LTE OFDM symbol (normal cyclic prefix, 7 OFDM symbols per 0.5 ms slot). The throughput of vector operations could be increased for operations involving regular memory accesses (that is, not CWL-like operations, for which B=1/2 is probably a minimum), up to a point where the bandwidth between the samples memory and the processing engine would become unrealistic with today's technology. We estimate this upper limit to about B=1/8. With these new figures, the overhead would become about 20000 cc (55% of the total processing power of a 500 MHz vector processor). Of course, things are even worse at larger bandwidths (85000 cc = 240% for a 20 MHz bandwidth). It must however be noted that the overhead is on the vector processor only, not on other DSP units (channel coder or decoder, interleaver, etc.).

Nowadays, the PAPR reduction is thus probably realistic in base stations where this optimization is really benificial, the available processing power is sufficient and the extra power budget spent in the extra digital signal processing is a small fraction of the total power consumption. Things are different on the user equipment side because these conditions are not yet met. In the mid-term range, however, it seams reasonable to anticipate digital vector processors running at 1 to 2 GHz and offering the above-mentioned features, with a high level of parallelism and total power consumption that will also become a small fraction of the total. The overhead would then become acceptable and PAPR reduction algorithms could then be implemented on terminals, at least for small to medium bandwidths.

The baseband processor used in the SACRA project is based on FPGAs for practical reasons and the maximum reachable clock frequency is in the 100-200 MHz range depending on the FPGA brand and version. As a consequence, PAPR reduction cannot be implemented.

3.5.3 Feasibility analysis of Digital Pre-Distortion (DPD)

The same feasibility analysis and overhead estimation has been conducted on the proposed digital pre-distortion technique. The analysis considers only the functional constraints and the required extra processing power of the digital baseband processor. Other impacts like the existence of the feedback path, the extra analogue-to-digital conversions or the increased conversion sampling rates of ADCs and DACs are out of scope of the digital baseband activities.

Among the different proposed operations some are low rate (RF impairments estimations, DPD computation and update) while others are expected to run at sample rate (application of DPD and RF impairments compensation). The former are all functionally compatible with the target baseband processor. We do not detail how each estimation would be implemented; most computations are hardware-accelerated, some very low rate operations are implemented in software, thanks to the embedded general purpose micro-controller. In other architectures, a classical DSP processor could take these low rate operations in charge. The total processing power overhead is negligible (and so is also the corresponding energy budget).

The RF impairment compensation is already built-in in the DSP unit that interfaces the baseband processor with the ADCs and DACs: the Pre-Processor (PP). PP is responsible for digital re-sampling (this feature is thus used to apply the DPD-related oversampling), CFO compensation, IQ imbalance compensation and DC offset compensation (which is applied simultaneously with IQ imbalance compensation). The RF impairment compensation thus relies on PP. The only source of computation overhead is due to the oversampling factor. A specific parameters setting of the compensation can be implemented or not depending on:

- The maximum throughput of the RF impairment compensation chain, expressed in cc per sample.
- The maximum running clock frequency.
- The number of transmission and reception channels simultaneously active and sharing the same RF impairment compensation chain.

In the SACRA baseband processor the PP is shared among all transmission and reception channels, the throughput is one sample per cc and the maximum clock frequency is 100 MHz on the ExpressMIMO board. In other baseband architectures with one dedicated PPs per channel, or different throughputs and clock frequencies, results could of course be different. With the parameters selected for the preceding simulations, and assuming one transmission channel at 46 Ms/s and one reception channel at 7.64 Ms/s, about 50% of the available processing power of PP is used on an ExpressMIMO target and 10% on a 500 MHz running user equipment. This part is thus perfectly realistic, on the digital signal processing point of view, both for base stations and user equipments.

The application of the DPD itself cannot be hardware-accelerated with the existing baseband processor. Its function could be computed by the FEP with a series of built-in operations (CWL, CWP and CWA) but there is no communication path between the output of the PP retiming filters and the FEP, nor between the FEP and the RF impairment compensation chain. Moreover, even if possible, this solution would be terribly inefficient, both in terms on useless data transfers between PP and FEP and in terms of used FEP processing power. As the application of the DPD is rather demanding, its software implementation would not be realistic.

A specific, lookup table-based, hardware accelerator would have to be added on front of the RF impairment compensation chain. This is perfectly realistic with the proposed parameters (2 tables, 256 entries each). The new module would have the same throughput and clock frequency characteristics as the RF impairment compensation chain so the conclusions about processing

power are the same: the most demanding supported configuration depends on the peak throughput and clock frequency. With the same assumptions as before and with this extra module added, the complete proposed DPD technique, including RF impairment compensation and application of DPD, would use 50% of the available processing power of PP on an ExpressMIMO board and only 10% on a 500 MHz running user equipment.

4 RF/BB INTERFACES

This chapter briefly presents the system environment of the baseband processor and the different interfaces. These aspects are already discussed in deep details in deliverables D4.1 and D4.2 and are not repeated here.

The overall system developed in the WP4 of SCRA is illustrated in the figure below:

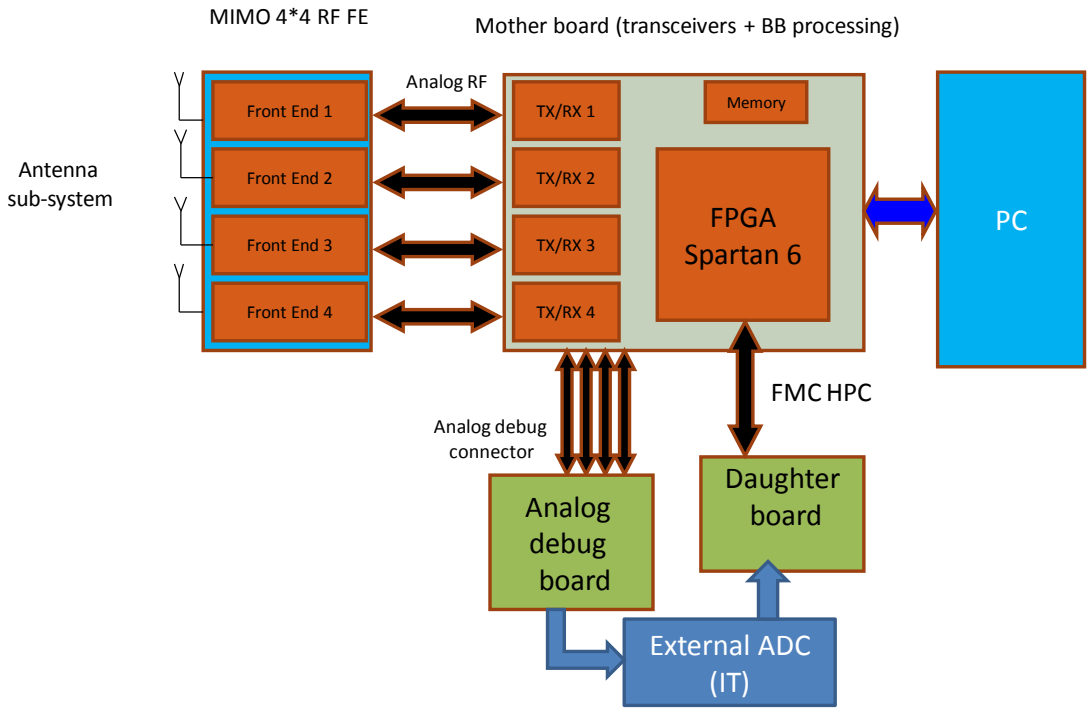


Figure 42: WP4 modem

Hardware parts can be seen as a full modem, able to perform a wireless communication in various bands and according to various Radio Access Technologies (RATs).Figure 42 shows a simplified interaction of all components in the overall platform architecture.

The overall platform architecture comprises:

- 4 antennas, i.e. 4 radiating elements, in order to perform Multiple Inputs Multiple Outputs (MIMO) processing. MIMO is a well known technique for spectrum efficiency [39]. Spatial and polarization diversity can be combined.
- A Radio Frequency Front End board, which comprises 4 Analogue Front End (AFE), for both Transmission and Reception (RX/TX).
- A motherboard that includes a SPARTAN6 LX150T from Xilinx and 4 transceivers. The transceivers can address a frequency range from 0.3 to 3.8 GHz, and with an extension up to 6.5 GHz.

- A daughter board, connected to the motherboard through a FPGA Mezzanine Connector (FMC) with a High Pin Count (HPC). This daughter board is used to debug the FPGA and it will be used to connect the output of an external Analogue to Digital Converter (ADC). The FMC connector can also be used to connect another FPGA-based daughter board in order to extend the limited processing capabilities of the SPARTAN6 FPGA.
- An analogue debug board interfaced to the motherboard through a dedicated connector. This board allows us to debug the analogue parts of the motherboard. It also provides analogue baseband (BB) signals for the transmission and for the reception. On the reception side, it is possible to use the analogue BB signal to feed an external ADC.
- A Personal computer (PC) running a real time Operating System (OS) based on Linux. The interface with the Motherboard is PCI-Express.

The overall modem is in development, and the general specifications are given as indicative. The overall objectives are very ambitious in term of functionalities, as illustrated in the following list:

- Real time, with processing split between the SPARTAN6 and the PC
- MIMO 4*4 : 3GPP Long Term Evolution Advanced (LTE-A) compliant for the baseband and analogue parts
- Frequency Division Duplex (FDD) and Time Division Duplex (TDD) are possible
- Frequency range between 300 MHz to 6.5 GHz with some restrictions on antennas
- We also target typical terminal RF performance such as a Noise Figure of 7 dB and a maximum transmitted Power of +23 dBm (for a 10 MHz LTE signal, 33 dB of ACPR)
- We target a Maximum bandwidth of 28 MHz
- The transceivers are controlled individually, so that it is possible to configure the system in a MIMO 4*4, MIMO 2*2 on one band and MIMO 2*2 on another band.
- Finally, the system supports a feedback loop in order to support Dirty RF mechanisms (Digital Pre Distortion, Inphase and Quadrature imbalance, Local Oscillator leakage compensation).

The components of the WP4 modem, as well as the interfaces, are extensively described in D4.1 and D4.2 of SACRA. After a first analysis of the WP4 modem, WP5 partners have made the following remarks:

- The Spartan 6 FPGA has very limited performance, both in terms of total available resources and maximum reachable clock frequency
- The existing ExpressMIMO platform is based on Virtex 5 FPGAs and there are significant differences between the Spartan 6 and the Virtex 5 architectures. Porting the existing baseband processor to Spartan 6 would require a significant amount of extra effort.

It has thus been decided to keep using the existing ExpressMIMO hardware target and to connect it to the WP4 modem through the FMC HPC connector.

The resulted integration is given in the figure below:

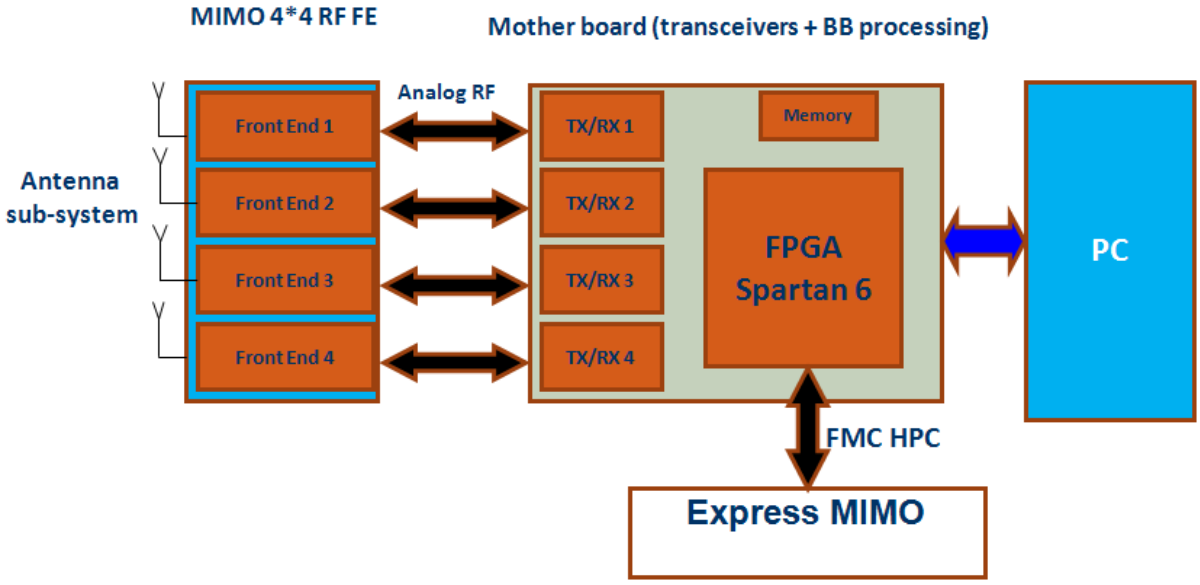


Figure 43: Integration of ExpressMIMO in the WP4 modem

Hence, the technology target o the baseband processor and its companion software layers developed in WP5 remain unchanged (Virtex 5-based ExpressMIMO board). The integration in WP6 will consist in connecting the motherboard to the ExpressMIMO board through their respective expansion connectors.

Therefore, we will have 2 settings for WP6:

- A setting with external ADC (Figure 42)
- A setting with ExpressMIMO (Figure 43).

In WP6, WP2 algorithms will be implemented in ExpressMIMO thanks the library, and WP3 algorithms will be implemented on the PC, since they are a part of higher layers (L3 and above).

5 OVERALL SYSTEM VALIDATION

In this chapter a detailed analysis of the simulation strategies that were followed for the validation of the applications developed on the baseband processor on top of the embedded software library is given. This section covers the validation of the overall system in terms of functionality, performance and energy estimation. More specifically a functional validation study that is conducted in four abstraction levels is provided in section 5.1. A complete version of this analysis will be delivered in 2012 as part of the WP5.4 results. Section 5.2includes simulation results of selected sensing algorithms that are implemented on top of the library that was described earlier in section 2. Performance and energy figures for those algorithms are given for different sizes of input length per operation executed in the Front End Processor component. These results highlight the benefit of SACRA's approach, as it is compared to a scenario where all the operations would be executed in the main processor.

5.1 FUNCTIONAL VALIDATION

Functional validations can be conducted at different abstraction levels:

1. Purely algorithmic, non bit-accurate, validations. WP5 activities do not cover this very classical abstraction level which relies on well known tools and languages (Matlab, C/C++, etc.). Computations are usually performed in floating point precision and in an untimed manner, without any concurrency between elementary processing. This abstraction level is the fastest one and also the most user-friendly for application designers but it suffers three drawbacks:
    a. It produces approximate results because of the floating point computations that will finally be converted into fixed point at lower abstraction levels.
    b. The coded application cannot be directly ported on the target hardware baseband processor because there is no direct relation between the used processing primitives and the hardware-accelerated ones and because the memory management is completely different from the final one.
    c. As it is purely sequential, it cannot be used to validate the correctness of the parallel behaviour.
2. Purely algorithmic, bit-accurate, validations. The synchronous version of the `libemmbb` library is used to design baseband processing applications. As with the previous abstraction level the execution is untimed, without any concurrency between elementary processing. This abstraction level is already implemented and has been used to validate several sensing and communication applications. It is slightly slower than the previous one because hardware-related details are taken into account. But it guarantees the portability on the target and, as the library is bit true with respect to the hardware implementation of the target baseband processor, computations are performed in fixed point precision and the results are bit accurate.
3. Bit-accurate, time-approximate, validations. A complete SystemC simulator of the baseband processor, including a cycle-approximate Instruction Set Simulator (ISS) of the main micro-controller, is used to run the application almost as it would run on the target. Of course, in order to take full benefit of this abstraction level, the application has to be first parallelized, thanks to the parallelization primitives offered by the underlying embedded Operating System: MutekH [38]. The observed timings are not 100% accurate because the ISS is not and also because the SystemC models of the DSP units of the baseband processor are just wrappers around `libemmbb` primitive calls. Each processing is executed

in zero-time and the surrounding wrapper applies a variable, approximate, delay, computed from the processing parameters (vector lengths, ...) in order to emulate the actual hardware latency. This level of abstraction is not yet fully available but a prototype version is currently running and debugged. A complete version will be delivered in 2012 as a part of the WP5.4 results. The main advantage is the possibility to simultaneously validate the algorithmic aspects and the functional aspects related to parallel processing (deadlocks, real time deadlines, data buffer underflows / overflows, etc.) Another advantage is the possibility to run and debug exactly the same application as the final one. The drawbacks are:

a. The simulation speed is about 20 times less than the actual target, which is acceptable for functional debugging of the application's parallelism but becomes a serious problem for very long simulation runs.

b. Because the ISS and the SystemC models are not 100% cycle-accurate, it could be that functional defects related to the parallel behaviour are not caught: in some rare corner cases, a few clock cycles more or less make the difference between a correct and a bogus result.

4. In order to speed-up the simulations at the previous abstraction level, a last environment is used in which the ISS of the main micro-controller is replaced by a native software process running the whole application but the calls to the hardware-accelerated operations. Its memory accesses in the addresses space of the DSP units of the baseband processor are trapped and passed to the SystemC simulator of the baseband processor. As the control software runs directly on the simulating PC, it is much faster than the ISS (and even than the target). A speed-up factor of at least 100 times is expected, compared with the ISS-based solution. The main counterpart is that only the hardware-accelerated operations are time-approximate. The time taken by the main controller when running the control software is ignored and the parallel behaviour can thus be different from what takes place on the actual target. This last level is currently under development and will be delivered, if successful and fast enough, in 2012 as a part of the WP5.4 results.

These four different abstraction levels can all be used in a very classical design flow:

- The algorithmic aspects of the application are first captured at the first abstraction level (e.g. using Matlab code) and simulated. The model is modified until the obtained algorithmic performance is satisfactory.
- The same application is re-coded in C, on top of the `libemmbb` library. The explicit memory management is manually added. Using the second abstraction level, the new algorithmic aspects due to fixed point computations are evaluated. The data placement in memory is also checked. The model is reworked until both aspects are validated.
- Synchronization primitives from the OS are added to the previous C-coded application. The new version is cross-compiled for the target micro-controller and linked with the OS and the target version of the `libemmbb` library. It is then run on the SystemC simulator (third abstraction level) in order to debug the parallel processing aspects.
- Once debugged, the final application is re-compiled for the desktop PC using the last abstraction level. Longer and much faster simulations can then be run with a mixture of native execution (for the control parts) and SystemC simulation (for the hardware-accelerated operations).

5.2 PERFORMANCE AND ENERGY VALIDATION

In this section a series of experiments for energy validation are shown. They were conducted on a modified version of the SuperESCalar Simulator (SESC). More specifically, specific examples that

are based on Energy Detector algorithm and the Welch periodogram based detector are examined in terms of energy consumption. These detectors are sensing algorithms that will be executed in the Front End Processor (FEP) component of the proposed Express MIMO architecture. As it will be shown in this section significant gains in terms of total cycle count are achieved due to the execution of these algorithms to components that are modelled as Hardware Accelerators. As a consequence energy consumption is expected to be reduced. In section 5.2.1 a detailed analysis of the Energy Detector algorithm is given, as well as the validation outcomes of the conducted simulations. Such an analysis is also given for the Welch periodogram based detector in section 5.2.2.

The performance and energy consumption are expressed in terms of total number of clock cycles and quantized energy consumption. The clock cycle counts are accurate because the processing times are deterministic and have been characterized by simulation of low-level hardware models. But, in the current version of the modified SESC simulator, the application is run in a pure sequential way (no parallelism between hardware-accelerated operations, data transfers, etc.) The cycle counts are thus approximations of what would take place in the actual target if it was running a synchronous version of the application.

The FPGA target technology used in the SACRA project for practical reasons is not a realistic one on the power consumption point of view. As a consequence, and in order to allow power comparisons between different applications or different implementations of a same application, each basic operation (adds, multiplies, memory loads and stores, etc.) has been associated an abstract energy quantum. The presented results are expressed in this abstract energy unit. In order to translate the abstract quanta into nano-Joules, an actual target manufacturing process of the baseband processor would have to be selected, the processor would have to be implemented up to and including place and route stages and the basic operations would have to be characterized by detailed power simulations for different operating conditions. These steps are out of scope of the SACRA project. Despite the fact that it does not provide accurate absolute energy figures, the very high level of abstraction used for energy validation is very useful to quickly compare relative energy figures. It is thus an efficient and fast tool to estimate and optimize the energy consumption of the digital baseband processing.

5.2.1 Energy detection algorithm

As mentioned afore, Energy Detector is a sensing algorithm that will be executed in the Front End Processor (FEP) component of the Express MIMO architecture. Energy detection is a common method for spectrum sensing due to its low complexity [37]. The ED compares the received energy during a time interval to a predefined threshold. The algorithm, during this finite time interval will execute vector operations on the input vector and a series of DMA transfers between the FEP component and the LEON3 microcontroller.

In this section, three series of simulation results will be presented. Firstly, the impact of input's length on the performance and energy consumption of the Energy Detection algorithm is examined. So, a set of total cycle count and quantized energy consumption for different value of input length will be presented. Secondly, it will be shown how the number of cycles and the energy consumption are distributed on each operation (vector operations, Fourier transformations and DMA transfers) for a specific length of input vector. Afterwards, it is examined if different thresholds values will influence the execution of the algorithm. Finally, a comparison that shows the significant gains due to the immigration of the execution of the algorithm from the LEON3 microcontroller to the Front End Processor of ExpressMIMO architecture [37] is described in detail.

Figure 44 and Figure 45 show the outcomes of the first series of simulation. More specifically, Figure 44 illustrates the total number of cycles needed for the execution of the Energy Detection on the coprocessor while Figure 45 shows the energy consumed during this execution. Various

values of the length of the input vector are examined and as it is shown the cycle count and consequently the energy consumption are reduced as the length of the input vector is increased. As mentioned before, although accurate energy figures are not available, the very high level of abstraction used for energy validation is very useful to quickly compare relative energy figures.

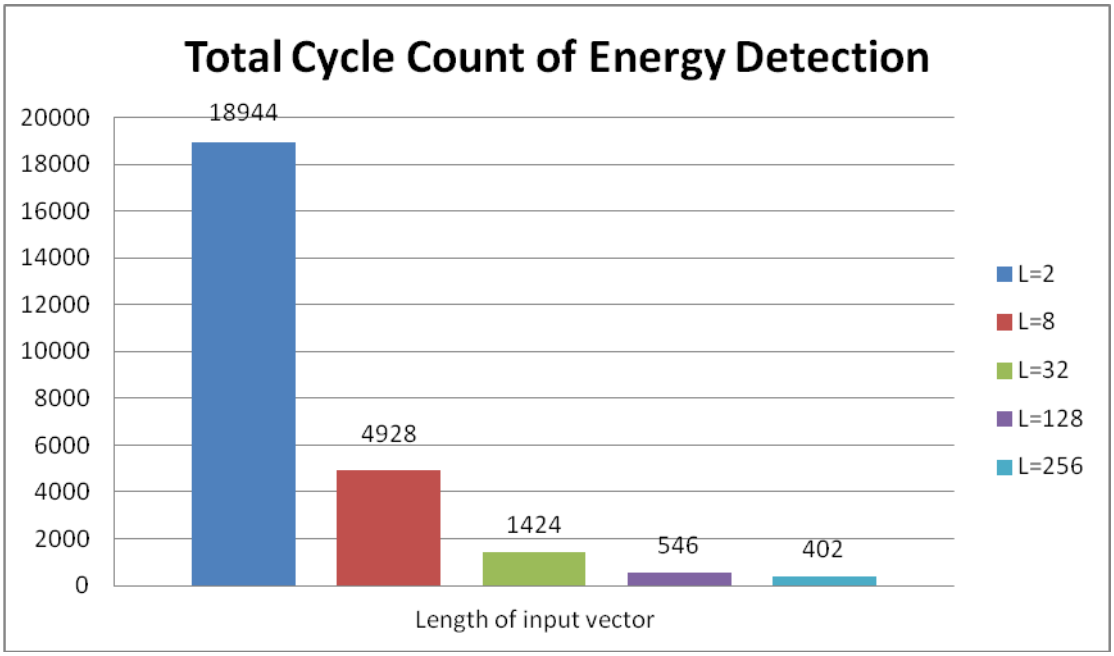


Figure 44: Cycle count of Energy Detection for various input vector lengths

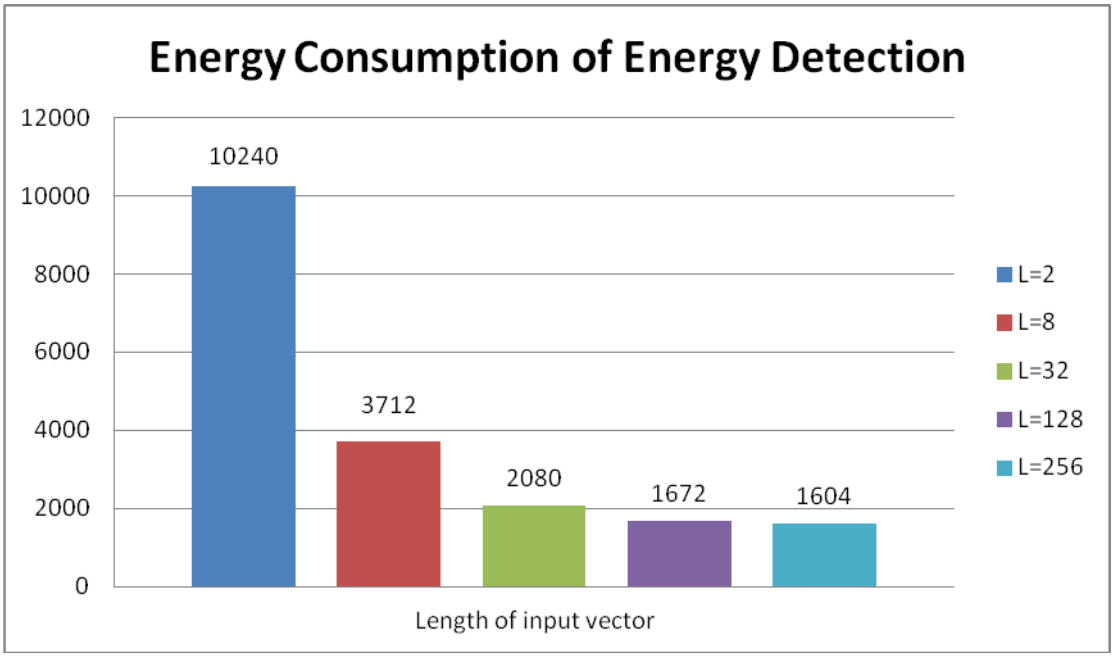


Figure 45: Energy Consumption of Energy Detection for various input vector lengths

Figure 46 to Figure 49 show the outcomes of the second series of simulation. More specifically, Figure 47 illustrates the number of cycles needed for the execution of the Energy Detection on the

coprocessor for each operation needed in Energy Detection, while Figure 47 shows the energy consumed during the execution of each operation. These results correspond to input vector length L=8.

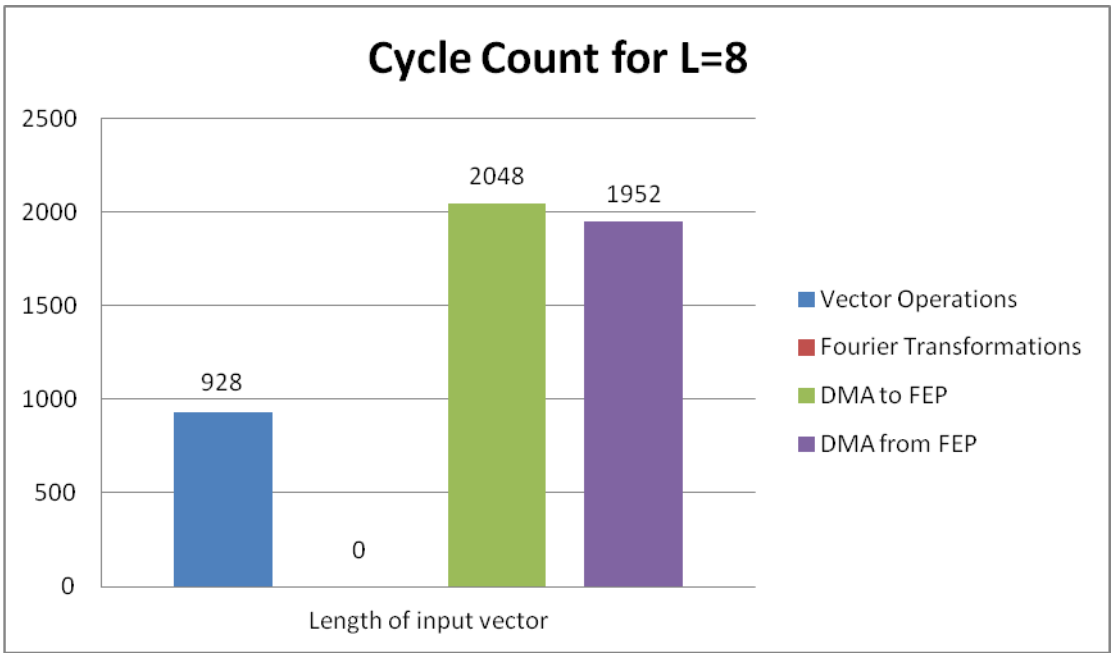


Figure 46: Cycle count per operation for L=8

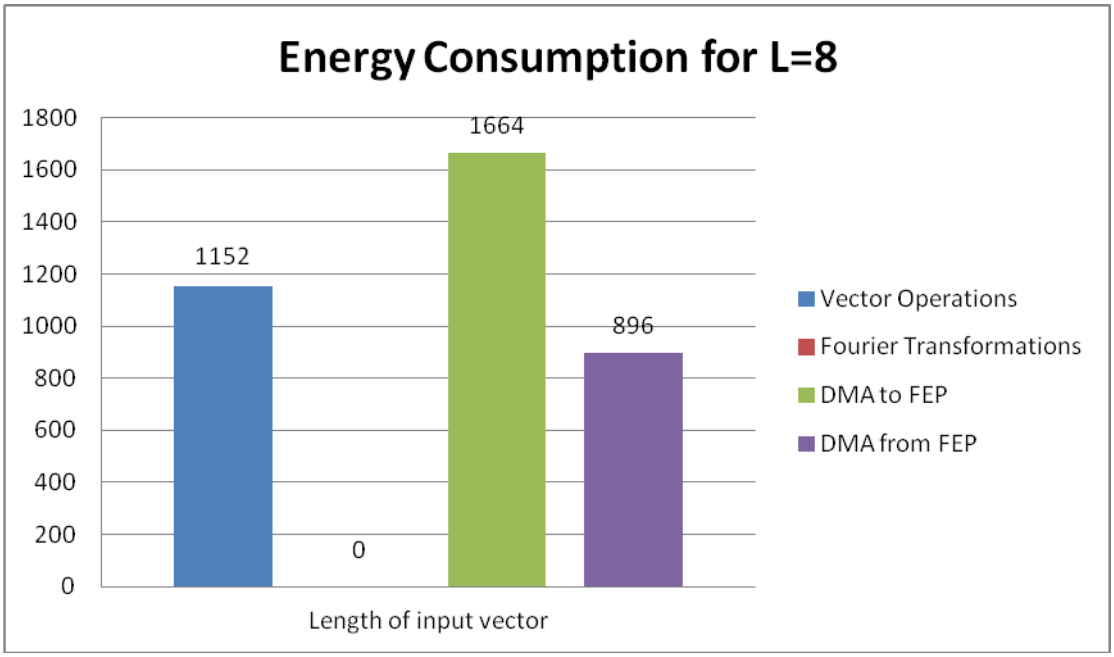


Figure 47: Energy Consumption per operation for L=8

Figure 48 and Figure 49 illustrate the same simulation pattern as Figure 46 and Figure 47, but for length of input vector L=128. As already mentioned, energy validation outcomes are useful in the essence that they provide an efficient and fast tool to estimate and optimize the energy consumption of the digital baseband processing.

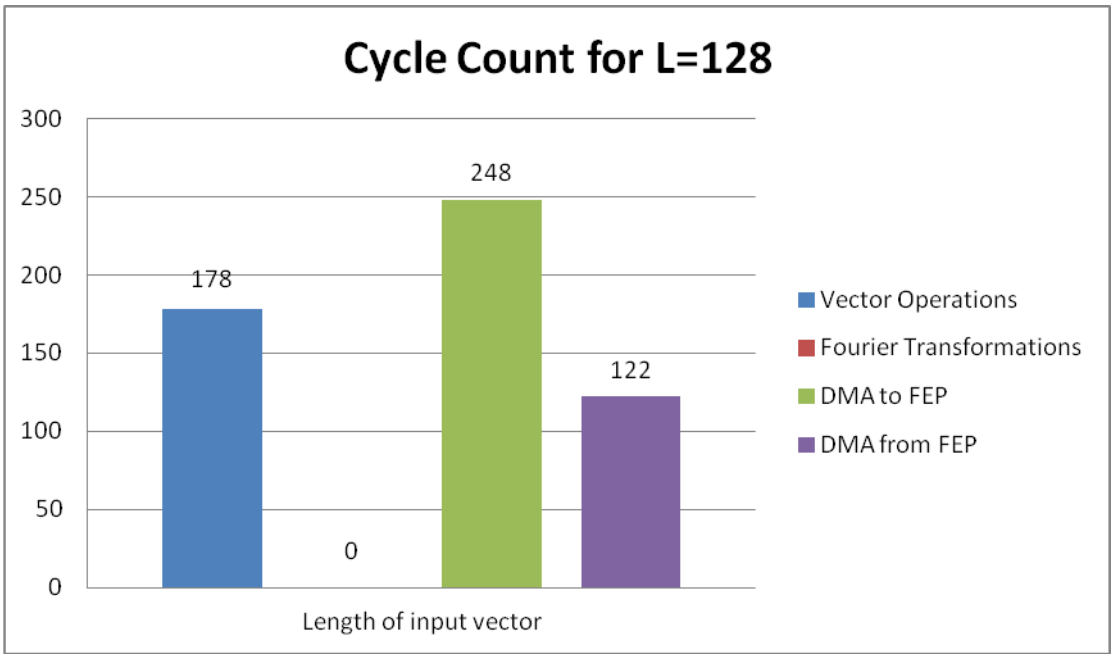


Figure 48: Cycle Count per operation for L=128

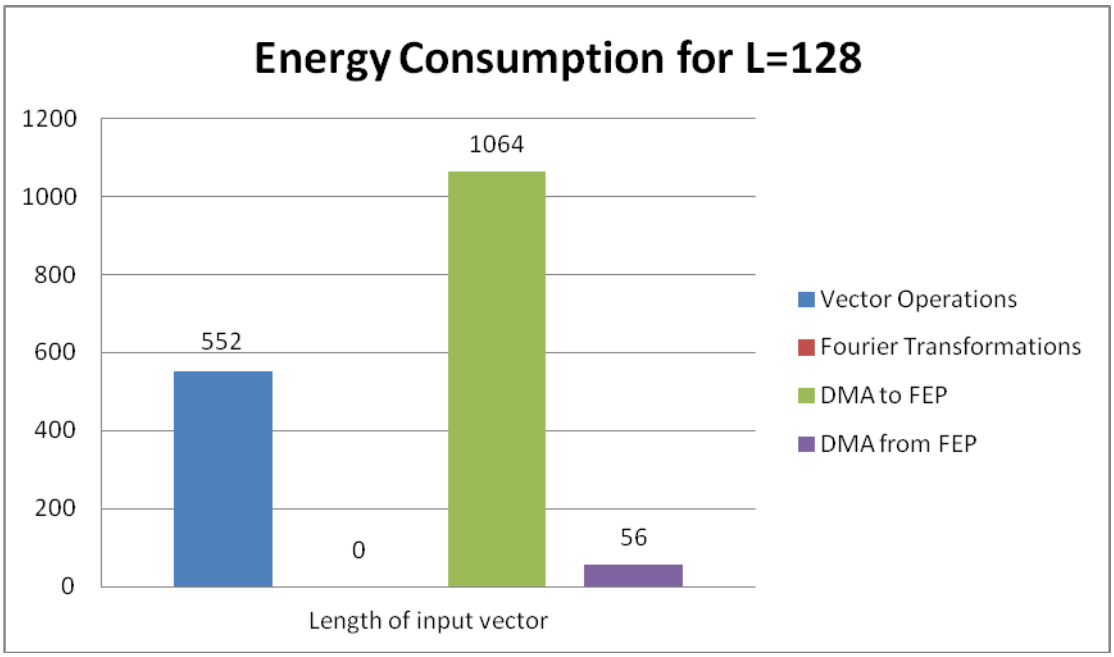


Figure 49: Energy Consumption per operation for L=128

Figure 50 and Figure 51 show the outcomes of the third series of simulation. More specifically, these two figures illustrate the comparison of the execution of the algorithm for different values of energy threshold. Figure 50 shows the number of cycles needed for the execution of the Energy Detection on the coprocessor for each operation needed in Energy Detection, while Figure 51 shows the energy consumed during the execution of each operation. These results correspond to input vector length L=128.

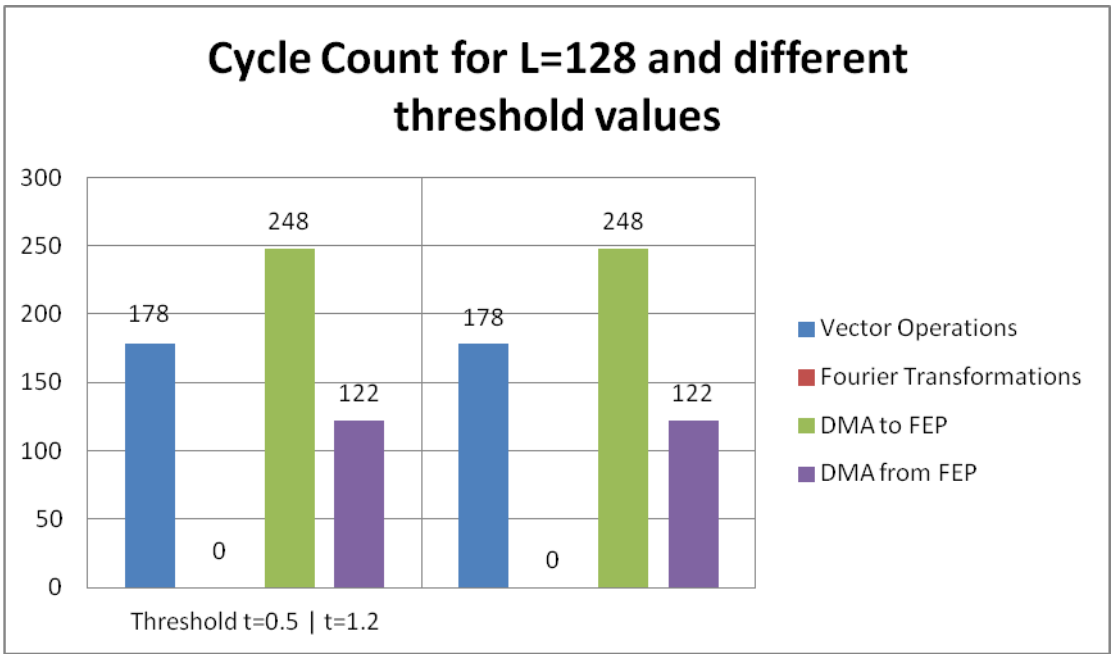


Figure 50: Number of cycles for different values of threshold

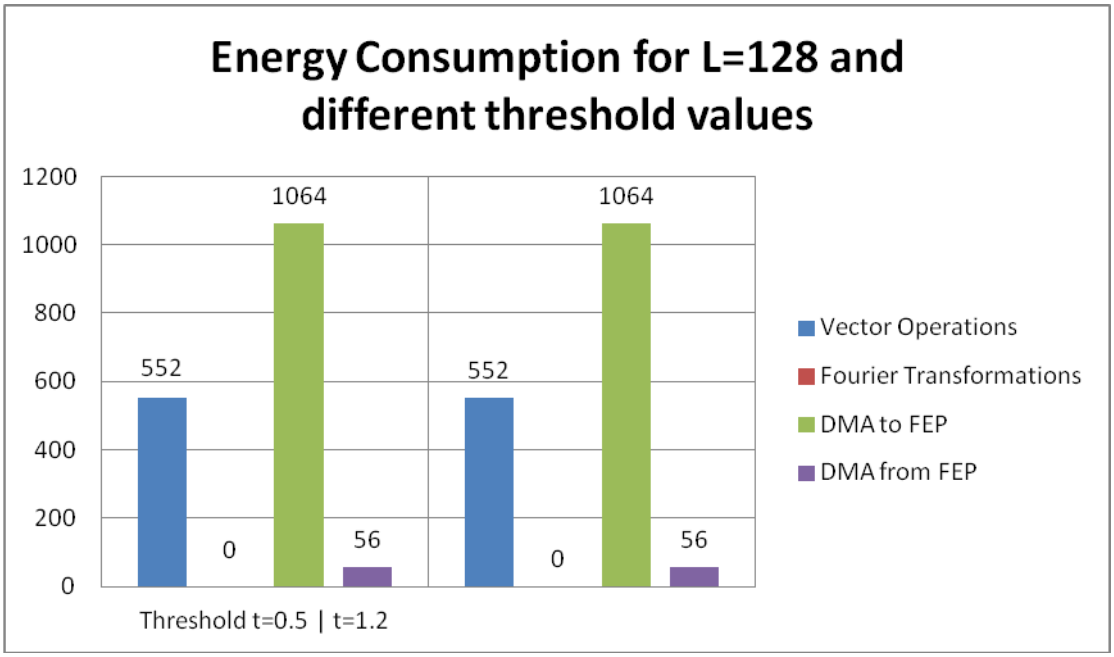


Figure 51: Energy Consumption for different values of threshold

Finally, Table 14 illustrates the difference between, a hypothetical scenario where all the operations would execute on the LEON3 microcontroller and the adopted approach, where the main computational part is executed on the Front End Processor which is a dedicated FPGA that can execute complicated instruction in few cycles. Of course, the cycles on the hardware accelerator cannot be directly compared with cycles on the main processor (e.g. different architectures, operation frequencies, etc), but these results show the significant gains of SACRA's approach as the total number of cycles on the main processor are reduced to 15% of their original value.

	LEON 3	LEON3 + FEP
LEON 3 cycles	186713255	2744541
Hardware Accelerator Cycles	0	840
Total	186713255	2747081

Table 14: Comparison between LEON 3 only and LEON 3 and Front End Processor

5.2.2 Welch periodogram algorithm

Similar analysis compared to the Energy Detection performance validation for the Welch periodogram is given below. In Welch periodogram algorithm an average value over L samples in frequency domain is compared to a predefined threshold to test the existence of the primary signal. Further analysis regarding the mathematical context and the use cases that this algorithm fits into were described in detail in [37].

As in energy detection algorithm, this algorithm's performance and energy consumption is tested in terms of total number of cycles and quantized energy consumption accordingly. As before, three series of simulation results will be presented. Firstly, the impact of input's length on the

performance and energy consumption of the Welch periodogram algorithm is examined. So, a set of total cycle count and quantized energy consumption for different value of input length will be presented. Secondly, it will be shown how the number of cycles and the energy consumption are distributed on each operation (vector operations, Fourier transformations and DMA transfers) for a specific length of FFT size. Finally, a comparison that shows the significant gains due to the immigration of the execution of the algorithm from the LEON3 microcontroller to the Front End Processor of ExpressMIMO architecture [37] will be presented.

Figure 52 and Figure 53 show the outcomes of the first series of simulation. More specifically, Figure 52 illustrates the total number of cycles needed for the execution of the Welch periodogram based algorithm on the coprocessor while Figure 53 shows the energy consumed during this execution. Various values of the length of the input vector are examined and as it is shown the cycle count and consequently the energy consumption are reduced as the length of the input vector is increased.

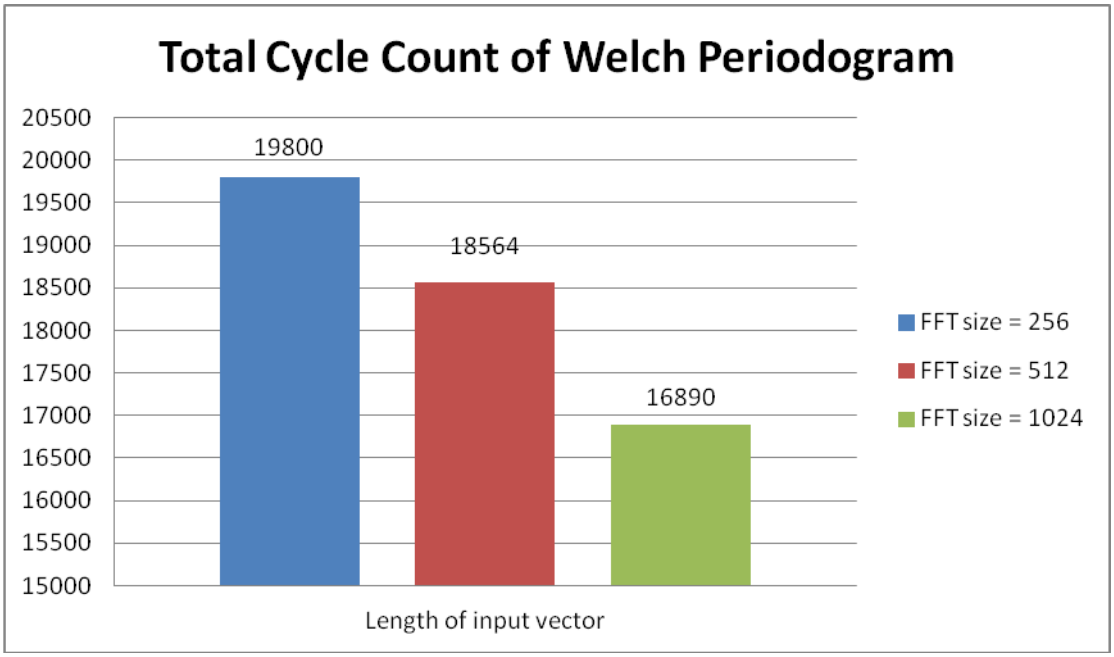


Figure 52: Cycle count of Welch Periodogram based detector for various input vector lengths

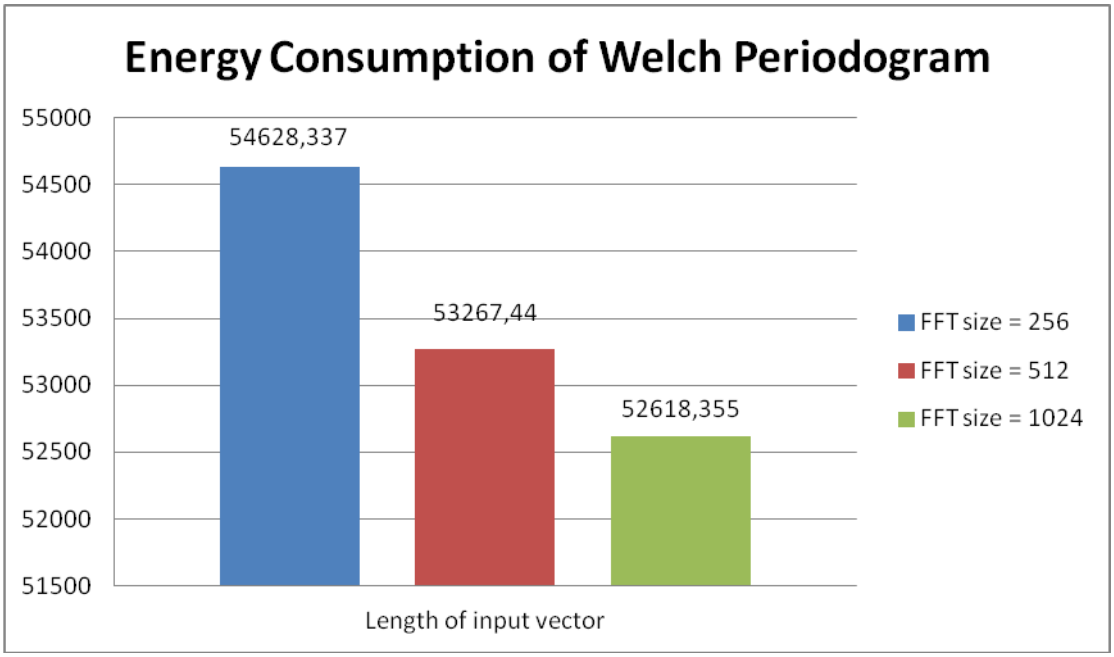


Figure 53: Energy Consumption of Welch Periodogram detector for various input vector lengths

Figure 54 to Figure 59 show the outcomes of the second series of simulation. More specifically, Figure 54 illustrates the number of cycles needed for the execution of the Welch Periodogram based algorithm on the coprocessor for each operation needed in this algorithm, while Figure 55 shows the energy consumed during the execution of each operation. These results correspond to FFT size = 256 and number of used carriers=59. The first corresponds to the input vector length for FFT operations in Welch periodogram based algorithm, while the latter corresponds to the input vector length of vector operations needed in this sensing technique.

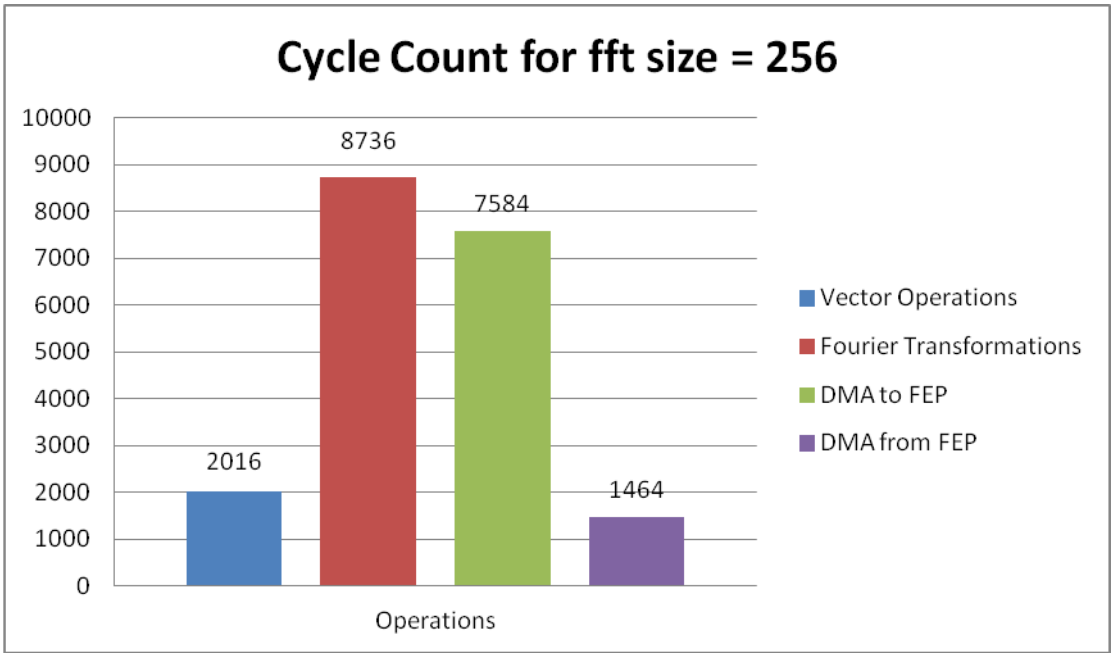


Figure 54: Cycle count per operation for FFT size=256

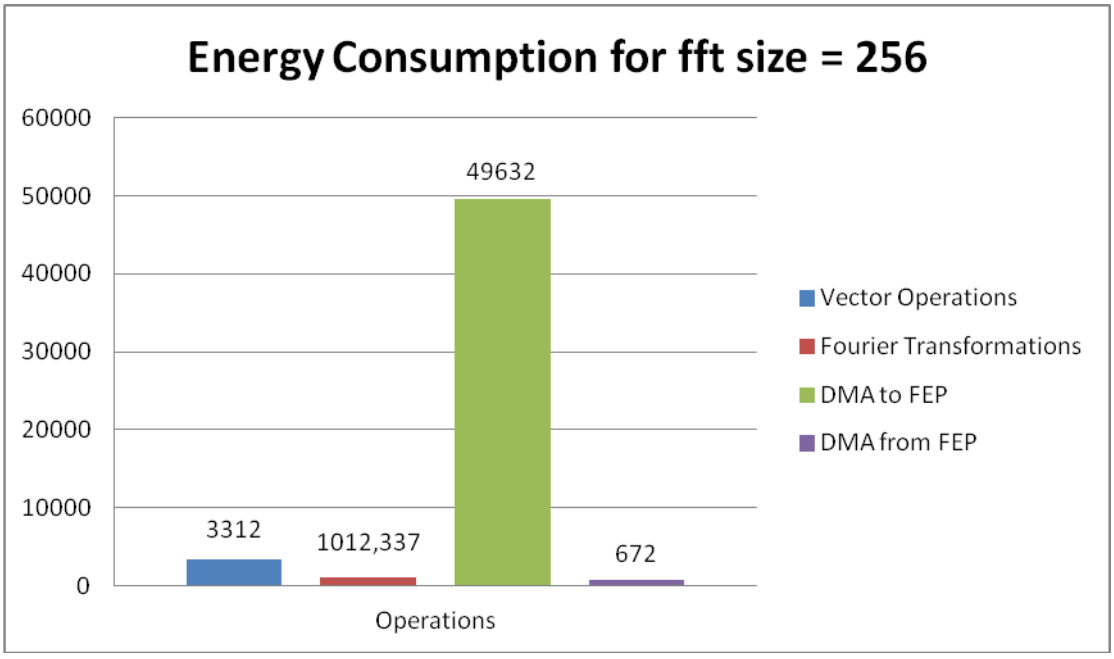


Figure 55: Energy Consumption per operation for FFT size=256

Figure 56 and Figure 57 illustrate the same simulation pattern as Figure 54 and Figure 55, but for length of input vector equal to 512 for FFT operations. In this case the number of used carriers is 115.

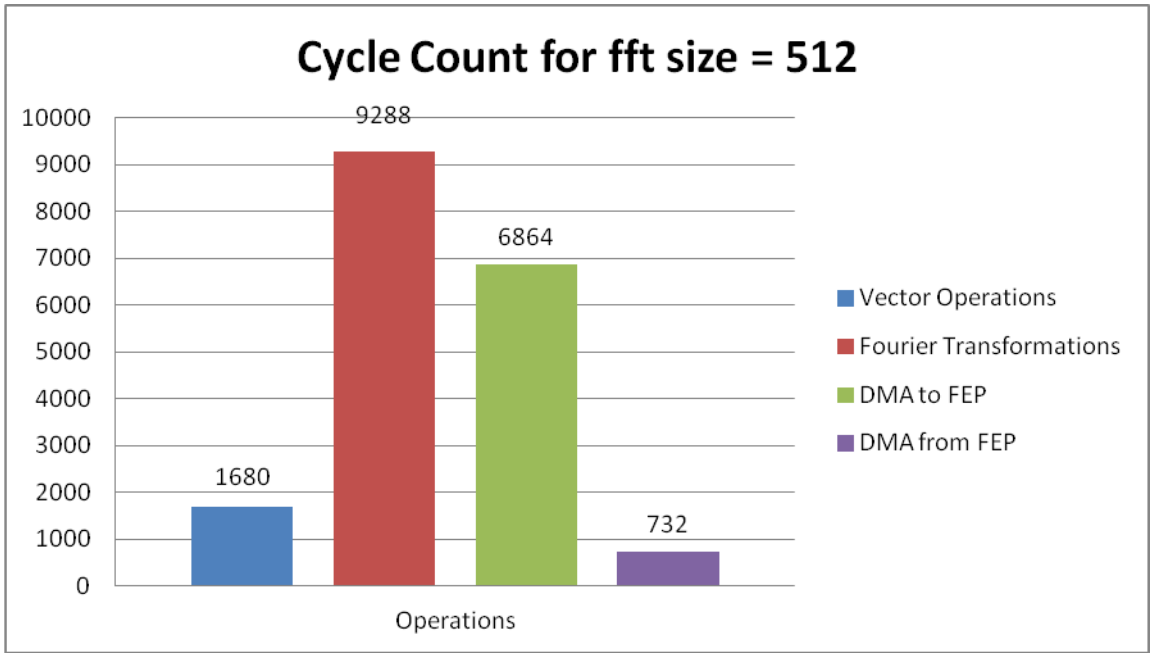


Figure 56: Cycle count per operation for FFT size=512

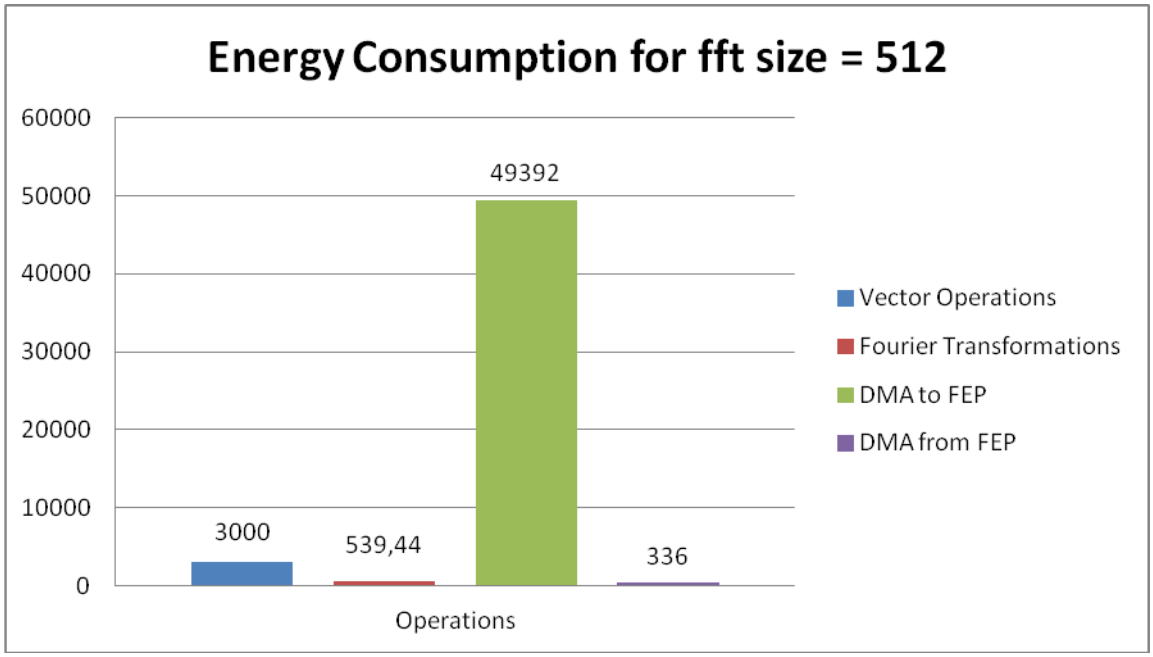


Figure 57: Energy Consumption per operation for FFT size=512

As before, Figure 58 and Figure 59 illustrate the same simulation pattern, but for length of input vector equal to 1024 for FFT operations. In this case the number of used carriers is 231.

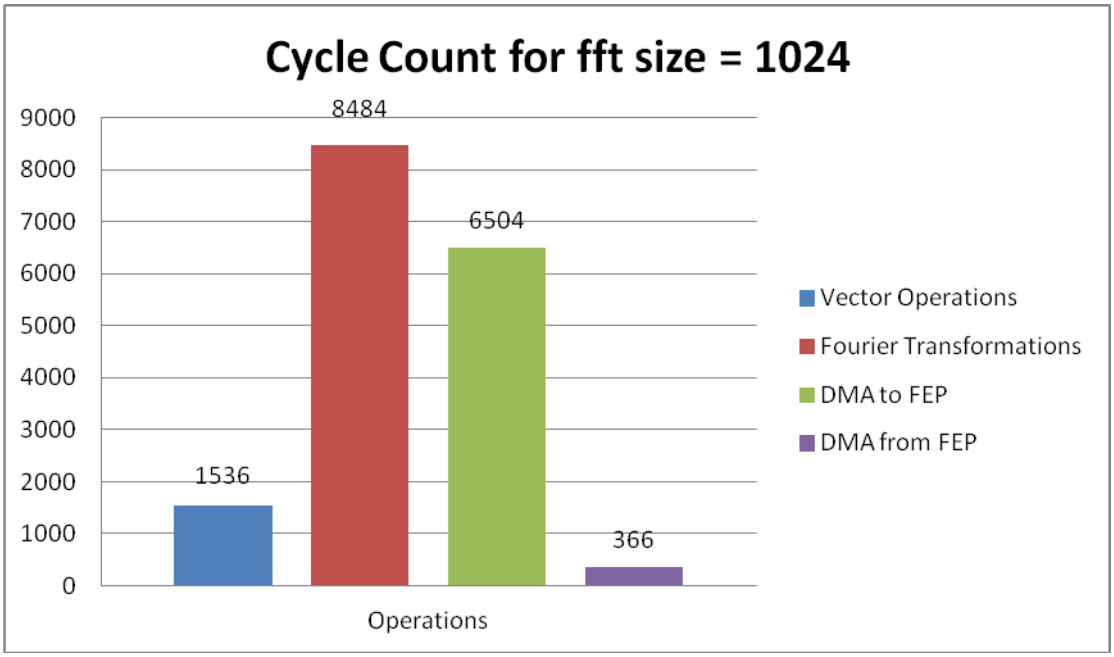


Figure 58: Cycle count per operation for FFT size=1024

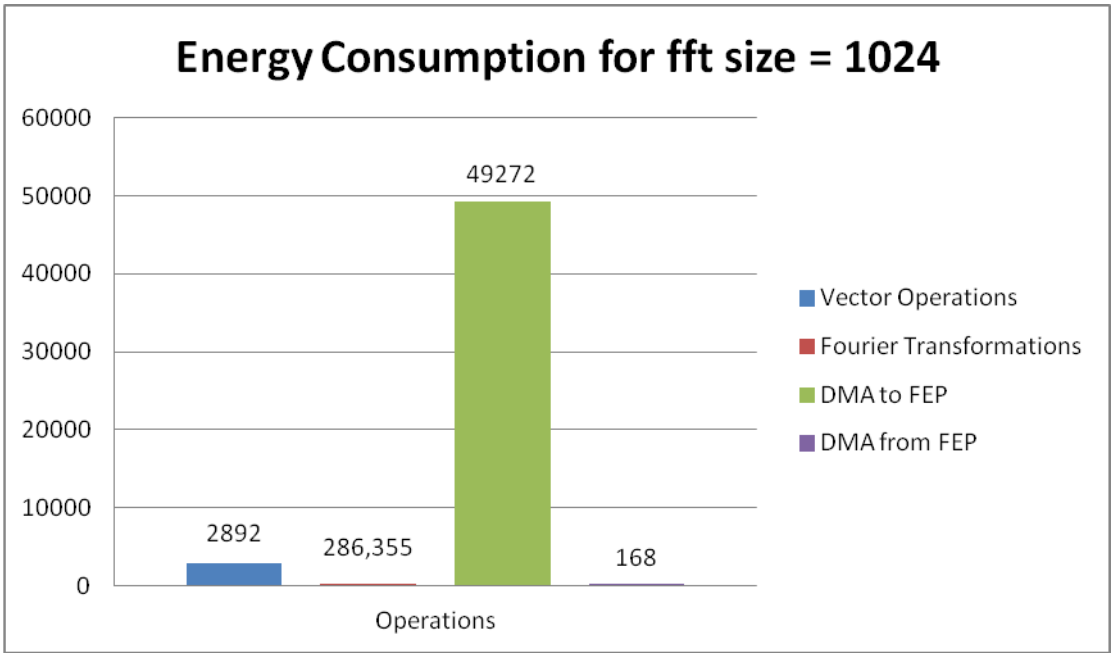


Figure 59: Energy Consumption per operation for FFT size=1024

As shown before in Energy Detection, different values of energy threshold do not influence the performance, or the energy consumption of the algorithm. So, testing the algorithm for different values of threshold would not bring any interesting validation results.

Finally, as in case of Energy Detector a hypothetical scenario where all the operations would execute on the LEON3 microcontroller is compared to the adopted approach, where the main computational part is executed on the Front End Processor which is a dedicated FPGA that can execute complicated instruction in few cycles. Table 15 illustrates the results of this comparison, but as mentioned before the cycles on the hardware accelerator cannot be directly compared with cycles on the main processor (e.g. different architectures, operation frequencies, etc). In case of Welch periodogram based algorithm these results show the significant gains of SACRA's approach as the total number of cycles on the main processor are reduced to 7% of their original value.

	LEON 3	**LEON3 + FEP**
LEON 3 cycles	1339448918	94129031
Hardware Accelerator Cycles	0	18564
Total	1339448918	94147595

Table 15: Comparison between LEON 3 only and LEON 3 and Front End Processor

6 CONCLUSION

At the time of delivery of this deliverable, a complete design and validation environment is available for baseband applications. A very detailed study of RF/baseband co-design strategies has also been conducted, including algorithmic performance evaluations and feasibility analysis. The outputs of this deliverable will be further enhanced in the terms of this work package. Milestone M5.4 will report the complete software design suite that will be integrated in WP6 demonstrator. Finally, Deliverable D5.3 will include the description of the adapted base band processor, the full software development kit (OS, libraries, compilers, linkers, debuggers), the SystemC-based virtual prototype, the embedded software generators and the integration in the TTool design framework.

7 ACRONYMS

Term	Description
API	Application Programming Interface
BB	Base Band
DAC	Digital-to-Analogue Converter
DPD	Digital PreDistortion
DSP	Digital Signal Processing (or Processor)
ISS	Instruction Set Simulator
MIMO	Multiple Input Multiple Output
OS	Operating System
PA	Power Amplifier
PAPR	Power Amplifier Power Reduction
PC	Personal Computer
RAT	Radio Access Technology
RF	Radio Frequency
SDR	Software Defined Radio
SESC	SuperESCalar Simulator
UML	Unified Modeling Language
AM/AM	Amplitude to Amplitude Modulation
AM/PM	Amplitude to Phase Modulation
IQ	In Phase / In Quadrature Phase
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
LUT	Look-Up Table
LS	Least Square
PSD	Power Spectral Density
CCDF	Cumulative Complementary Density Function
CFO	Carrier Frequency Offset
CPE	Common Phase Error
DC	Direct-Current
ACPR	Adjacent Chanel Power Ratio
OFDM	Orthogonal Frequency Division Multiplexing

Term	**Description**
LTE	Long Term Evolution
ACE	Active constellation Extension
BER	Bit Error Rate
FEP	Front End Processor
DMA	Direct Memory Access
FPGA	Field-Programmable Gate Array
FFT	Fast Fourier Transformation
EVM	Error Vector Magnitude

8 BIBLIOGRAPHY

[1] J.A.C. Bingham, “Multi-carrier Modulation for data transmission: An Idea Whose Time Has Come”, IEEE Communications Magazine, vol. 28, no. 5, PP. 5-14, May 1990.

[2] P. Ramjee, OFDM for Wireless Communications Systems. Artech House, Incorporated, 2004.

[3] H.S. Black, “Stabilised feedback amplifiers”, The Bell system Technical Journal, Jan. 1934.

[4] A. Smith and J. Cavers, “A wideband architecture for adaptive feedforward linearization”, IEEE 48th Vehicular Technology Conference, VTC’98, vol. 3, pp. 2488-2492, May. 1998.

[5] J.K Cavers, “Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements”, IEEE Transactions on Vehicular Technology, pp. 374-382, Nov. 1990.

[6] J.K Cavers, “The effect of quadrature modulator and demodulator errors on adaptive digital predistorters for amplifier linearization”, IEEE Transactions on Vehicular Technology, pp. 456-466, May 1997.

[7] R. Qiming; I. Wolff, “Effect of demodulator errors on predistortion linearization“,IEEE Transactions on Broadcasting, pp 153-161,Jun. 1999.

[8] J.H. Lee and S.H. Han, “An overview of peak-to-average power ratio reduction techniques for multicarrier transmission”, IEEE Wireless Communications, pp. 56, Apr. 2005.

[9] B.S. Krongold and D.L. Jones, “PAR Reduction in OFDM via Active Constellation Extension”, IEEE Transactions Broadcasting, pp. 258-268, Sep. 2003.

[10] C. Ciochina, F. Buda and H. Sari, “An Analysis of OFDM Peak Power Reduction Techniques for WIMAX Systems”, IEEE International Conference On Communications, pp. 4676-4681, Jun. 2006.

[11] N. Aizawa, O. Muta, Y. Akaiwa and M. Sawahashi, “Effect of Peak Power Suppression and Adaptive Predistortion on Power Amplification of an OFCDM Signal”, IEEE Transactions On Vehicular Technology Conference, pp. 1783-1787, Dec. 2005.

[12] A. Bo, Y. Zhi-xing, P. Chang-yong, Z. Tao-tao and G. Jian-hua, “Effects of PAPR Reduction on HPA Predistortion”, IEEE Transactions On Consumer Electronics, vol. 51, Issue 4, pp. 1143-1147, Nov. 2005.

[13] H. Ryu, T.P. Hoa, K.M. Lee, S. Kim and J. Park, “Improvement of Power Efficiency of HPA by PAPR Reduction and Predistortion”, IEEE Transactions On Consumer Electronics, vol. 50, Issue 1, pp. 119-124, Feb. 2004.

[14] M. Heloui, S. Boumaiza, A. Ghazel and F.M. Ghannouchi, “Power and Efficiency Enhancement of 3g Multicarrier Amplifiers Usung Digital Signal Processing with Experimental Validation”, IEEE Transactions on Microwave Theory and Techniques, vol. 54, pp. 1396-1404, Jun. 2006.

[15] C. Nader, et al, “Peak-to-average power ratio reduction versus digital pre-distortion in OFDM based systems”, IEEE Int. Microwave Symposium Digest, Aug. 2011.

[16] L. Wang and C. Tellambura, “An Adaptive-Scaling Algorithm for OFDM PAR Reduction Using Active Constellation Extension”, IEEE Transactions On Vehicular Technology Conference Fall, pp. 1-5,Sep. 2006.

[17] J. Tellado, Multicarrier Modulation with Low Peak to Average Power Applications to xDSL and Broadband Wireless, Boston/Dordrecht/London: Kluwer Academic Publishers, 2000.

[18] J.C. Pedro and S.A. Maas, "A comparative overview of microwave and wireless power amplifier behavioral modeling approaches," IEEE Trans. Microw. Theor Techn, Vol. 53, No. 4, April 2005, pp. 1150–1163.

[19] M. Isaksson, D. Wisell, and D. Rönnow, “Wideband dynamic modeling of power amplifiers using radial basis function neural networks,” IEEE Trans. Microw. Theory Tech., vol. 53, no. 11, pp. 3422–3428, Nov. 2005.

[20] D. Morgan, Z. Ma, J. Kim, M. Zierdt, and J. Pastalan, “A Generalized Memory Polynomial Model for Digital Predistortion of RF Power Amplifiers,” IEEE Trans. on Signal Processing, vol. 54, pp. 3852–3860, Oct 2006.

[21] SKYWORK 77107 Power Amplifier [Online] http://www.skyworksinc.com/uploads/documents/201227a.pdf

[22] SKYWORK 77106 Power Amplifier [Online] http://www.skyworksinc.com/uploads/documents/201225a.pdf

[23] C. Blanco, "Gain expansion and intermodulation in a MESFET amplifier,” Electron. Lett., vol. 15, no. 1, pp.31 - 32 , 1979.

[24] L. Ding, G. T. Zhou, D. R. Morgan, Z. Ma, J. S. Kenney, J. Kim, and C. R. Giardina, "A robust digital baseband predistorter constructed using memory polynomials," IEEE Trans. on Communications, vol. 52, no. 1, pp. 159-165, January 2004.

[25] L. Ding, G. T. Zhou, “Effects of Even-Order Nonlinear Terms on Power Amplifier Modeling and Predistortion Linearization,” IEEE Trans. on Vehicular Technology, vol. 53, no. 1, pp. 156-162, January 2004.

[26] R. Raich, H. Qian, and G. T. Zhou, "Orthogonal polynomials for power amplifier modeling and predistorter design," IEEE Trans. on Vehicular Technology, vol. 53, no. 5, pp. 1468-1479, September 2004.

[27] R. Raich, and G. T. Zhou, "Orthogonal Polynomials for Complex Gaussian Processes," IEEE Trans. on Signal Processing, vol. 52, no. 10, pp. 2788-2797, October 2004.

[28] S. M. Kay, “Fundamentals of Statistical Signal Processing: Estimation Theory”, Englewood Cliffs, NJ: Prentice-Hall, 1993.

[29] K. Yanagisawa, N. Matsuno, T. Maeda, S. Tanaka, “A New DC-Offset and I/Q-Mismatch Compensation Technique for a CMOS Direct-Conversion WLAN Transmitter,“ in Proceedings of IEEE MTT-S, pp 85-88, June 2007.

[30] M. Windisch and G. Fettweis, “Adaptive I/Q Imbalance Compensation in low-IF Transmitter Architectures,” in Proceedings of IEEE Vehicular Technology Conference, Sept. 2004, pp. 2096–2100.

[31] M. Windisch and G. Fettweis, “Standard-independent I/Q imbalance compensation in OFDM direct-conversion receivers,” in Proc. 9th International OFDM Workshop, Sept. 2004, pp. 57–61.

[32] A. Armada, “Understanding the Effects of Phase Noise in Orthogonal Frequency Division Multiplexing (OFDM),” IEEE Trans. on Broadcasting, vol. 47, no. 2, pp 153-159, June 2001.

[33] T. M. Schmidl and D. C. Cox, “Robust frequency and timing synchronization for OFDM,” IEEE Trans. Commun., vol. 45, pp. 1613–1621, Dec. 1997.

[34] T. C. W. Schenk, E. R. Fledderus and P. F. M. Smulders, "Performance impact of IQ mismatch in direct-conversion MIMO OFDM transceivers", Proc. IEEE Radio Wireless Symposium 2007, pp. 329-332, Jan 2007.

[35] S. Traverso, M. Ariaudo, I. Fijalkow, J-L Gautier, and C. Lereau, “Decision Directed Channel Estimation and High I/Q Imbalance Compensation in OFDM Receivers,” IEEE Transactions on Communications, vol. 57, no 5, pp 1246-1249, Feb. 2009.

[36] Y. Kim, E. Jeong, T. Noh, and Y.H. Lee, "Joint Adaptive Compensation for Amplifier Nonlinearity and Quadrature Modulation Errors", in Proc. VTC Spring, 2006, pp.2290-2293.

[37] Deliverable of SACRA Project, “Preliminary report on the system requirements, application modeling and embedded software library”.

[38] MutekH, a free and portable operating system for embedded platforms. [Online] http://www.mutekh.org/

[39] S. N. Diggavi, “On achievable performance of spatial diversity fading channels,” IEEE Trans. Inform. Theory, vol. 47, no. 1, pp.308-325, Jan. 2001.