WROCŁAW UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF ELECTRONICS

FIELD: Elektronika (EKA)
SPECIALIZATION: Advanced Applied Electronics(AAE)

MASTER OF SCIENCE THESIS

Problems of software implementation of SDR radio in programmable logic structures

Zagadnienia implementacji programowego radia SDR w programowalnych strukturach logicznych

AUTHOR:
Karol Szpila

SUPERVISOR:

Ph.D., D.Sc. Grzegorz Budzyń

Institute of Telecommunications, Teleinformatics and Acoustics (I-28)

GRADE:

WROCŁAW 2019

Nomenclature

ADC	Analog to Digital Converter
ASIC	Application-Specific Integrated Circuit
DAC	Digital to Analog Converter
DBIQIM	Double Branch IQ Imbalance Model
CPU	Central Processing Unit
FIR	Finite Impulse Response
FPGA	Field Programmable Gate Array
IF	Intermediate Frequency
IQ	In-phase Quadrature
LO	Local Oscillator
LSB	Lower Side Band
SDR	Software Defined Radio
PLL	Phase Locked Loop
RF	Radio Frequency
SBIQIM	Single Branch IQ Imbalance Model
SoC	System on Chip
USB	Upper Side Band

Contents

1 Introduction **4**
1.1 Purpose and aim 4
1.2 Thesis outline 4

2 Theoretical background **6**
2.1 Theoretical operation of RF mixer 6
2.2 IQ signal model 8
2.3 IQ imbalance models 11
2.4 Software Defined Radio 14

3 Hardware and tools **15**
3.1 Adalm Pluto and AD tools 15
3.2 libIIO 18
3.3 Zynq and Xilinx tools 19

4 Algorithms **22**
4.1 DC offset correction 22
4.2 IQ Mismatch Correction 24

5 Hardware Implementation **26**
5.1 Data Flow Path 26
5.2 Moving Average Filter 28
5.3 Gaussian Filter 28
5.4 I/Q mismatch compensation 29

6 Performance evaluation **30**
6.1 Simulations 30
6.2 Hardware Implementation 40
6.3 On Chip Algorithm 45

7 Conclusions **51**

Bibliography **51**

Chapter 1

Introduction

Nowadays modern technologies are developing very rapidly often displacing older solution. This creates need for very flexible systems that can adapt to the changes in the future. Such problem can be approached in two ways: by moving processing part of the system form hardware to software, or by create re-configurable hardware. Both approaches have its own advantages and disadvantages, but current trend on the market is system composed of both: fast processing system and re-configurable hardware. This lead to creation of SoC (System on Chip) which is build on the base of both solution. Such approach allows to utilize advantage of one and the other. In RF system problem of easily adaptive system considers also transceiver part. For that reason direct conversion is widely used in modern RF application. Such approach allows to limit hardware part to the necessary minimum and move all further processing to the baseband processing system.

1.1 Purpose and aim

The purpose of this paper is to study various parameters defining RF signal quality and model of IQ imbalance. Research concept of Software Defined Radio, principles of operation of such devices and capability of Xilinx Zynq SoC in such domain. Evaluation of different correction algorithms implementations. Test of their performance in Matlab and hardware implementation together with compenation built in RF agile transceiver present on the board. Measurements includes various type of signals: single tone, multi tone and broadband. The comparison include simulation in Matlab, implementation hardware (FPGA part of ZYNQ SoC) and native correction implemented in RF transceiver hardware.

1.2 Thesis outline

This thesis is divided into following chapters:

- Chapter 1 contains brief introduction to this master thesis, presented problems and considered solutions.
- Chapter 2 explains theory behind IQ signal including: model of the signal, its parameters, imbalance model, theoretical operation of IQ modulator and demodulator and concept of SDR.

- Chapter 3 describes concept of SoC and hardware used for tests.
- Chapter 4 is mathematical introduction to the presented problem of IQ imbalance together with explanation of chosen correction algorithms.
- Chapter 5 presents hardware implementation of presented algortihms.
- Chapter 6 evaluates performance of all algorithhms in Matlab, implemented in hardware. and built in RF agile transceiver present on the board.
- Chapter 7 concludes all performed measurements, results and summarize this master thesis.

Chapter 2

Theoretical background

In this chapter the theoretical operation of basic RF fronted components is explained. Mentioned elements are mixers, quadrature modulators and demodulators. Next widely used in RF communication IQ signal model is described, together with imbalance models. After that Software Defined Radio concept is presented.

2.1 Theoretical operation of RF mixer

The mixer is one of the basics component used in RF communication. It is nonlinear circuit used for signal multiplication. Such process is used virtually in every radio system, cellular base stations, radars and many more devices.

RF mixer is three port devices, which takes as an input two signal and produces signal consisting of the two frequencies on the output. Due to the non linearity of the mixer the new signal on the different frequency is generated. [2]

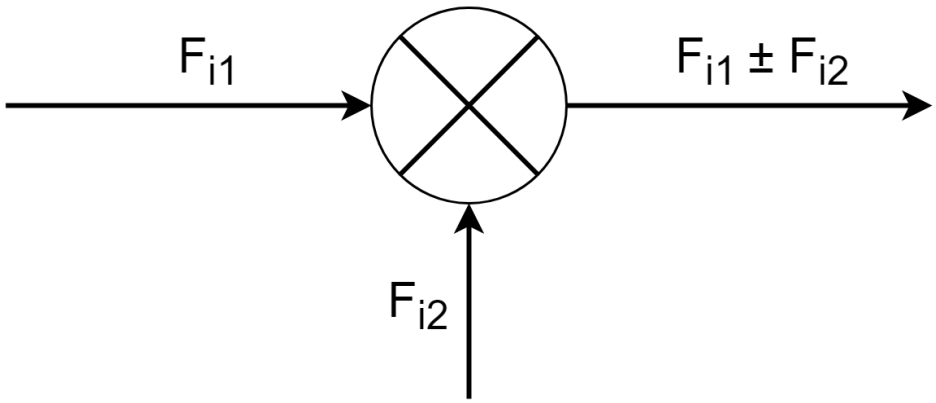

Figure 2.1: *Schematic of RF mixer.*

Where:

- F_{i1} and F_{i2} are input signal that are mixed together.
- $F_{i1} + F_{i2}$ is USB (Upper Side Band) output signal,
- $F_{i1} - F_{i2}$ is LSB (Lower Side Band) output signal.

In every mixing operation USB and LSB signal are generated. Depending on the relation between RF and LO signal one of the band is desired output and other is undesired image.

When $RF < LS$ mixer works as up-converter USB is desired and LSB is undesired.

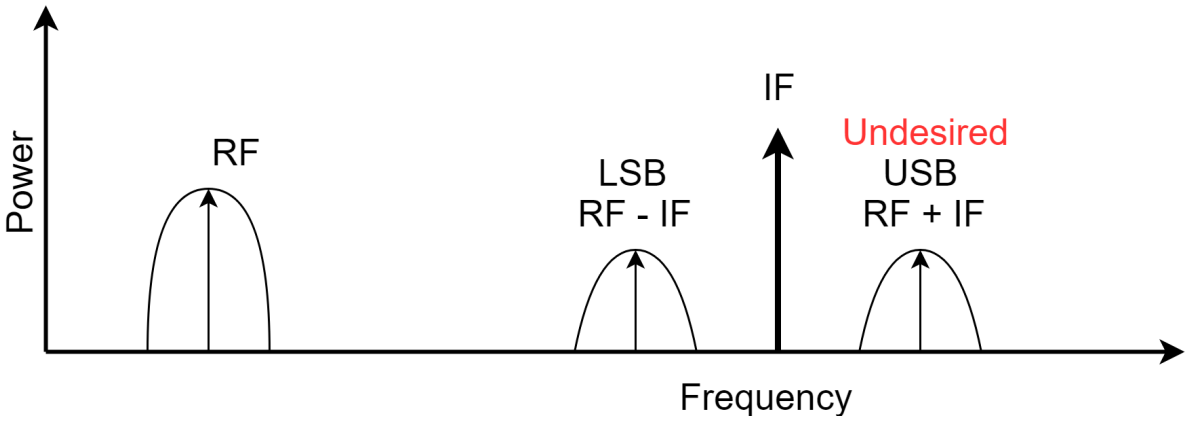

Figure 2.2: *Schematic of RF mixer.*

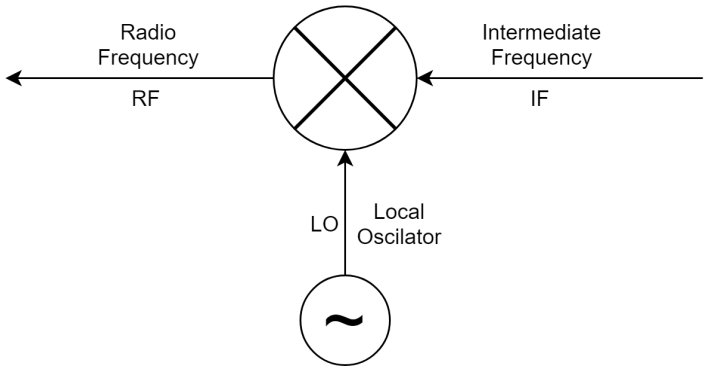

Figure 2.3: *Schematic of RF mixer working as up-converter.*

When $RF < LS$ mixer works as down-converter LSB is desired and USB is undesired..

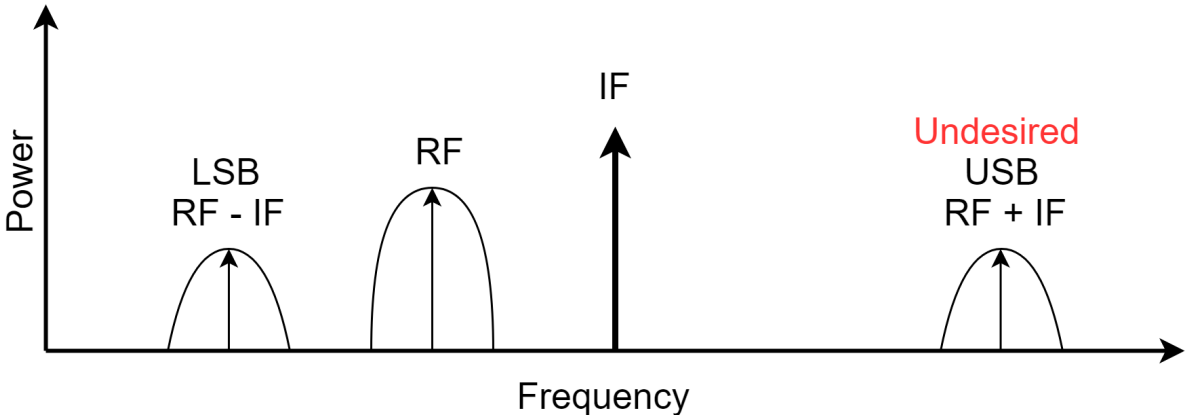

Figure 2.4: *Schematic of RF mixer.*

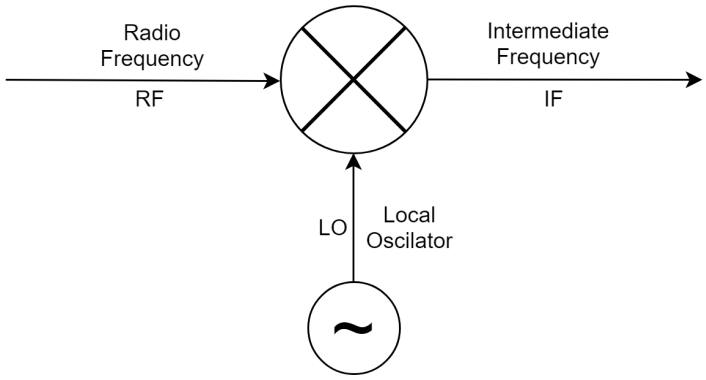

Figure 2.5: *Schematic of RF mixer working as down-converter.*

To remove unwanted band it is necessary to include in the RF mixer design BPFs (Band Pass Filters). This filter attenuates all frequencies outside specified band.

Application for mixers are mostly [2]

- Frequency shifting - The most common application for RF mixer is changing signal frequency with preservation of others parameters such as amplitude and phase. This technology is mostly used in RF transmitter and receivers. Such devices are called up-converters (for mixing signal with carrier) and down-converter (for extracting signal from carrier).
- Phase comparison - It is possible to detect phase difference between two signal using mixer. This RF mixer application is used in PLLs (Phase Locked Loops).

2.2 IQ signal model

The term IQ is an abbreviation for in-phase and quadrature. Signal are considered in-phase when phase of both is equal and quadrature when it differs by 90°. IQ data model shows changes in phase and magnitude of a sine wave. Modification of these parameters allow to encode information upon a sine wave.

Equation of the sine wave is:

$$A cos\left(2\pi f t + \phi\right),$$

where:

- A is amplitude,
- f is frequency,
- ϕ is phase shift

According to equation only amplitude, phase and frequency of the sine wave can be modified. Moreover frequency is first derivative of phase. Therefore it can be collectively referred to as the phase angle. According to these assumptions the instantaneous state of a sine wave can be described in complex plain using magnitude and phase as polar coordinates.

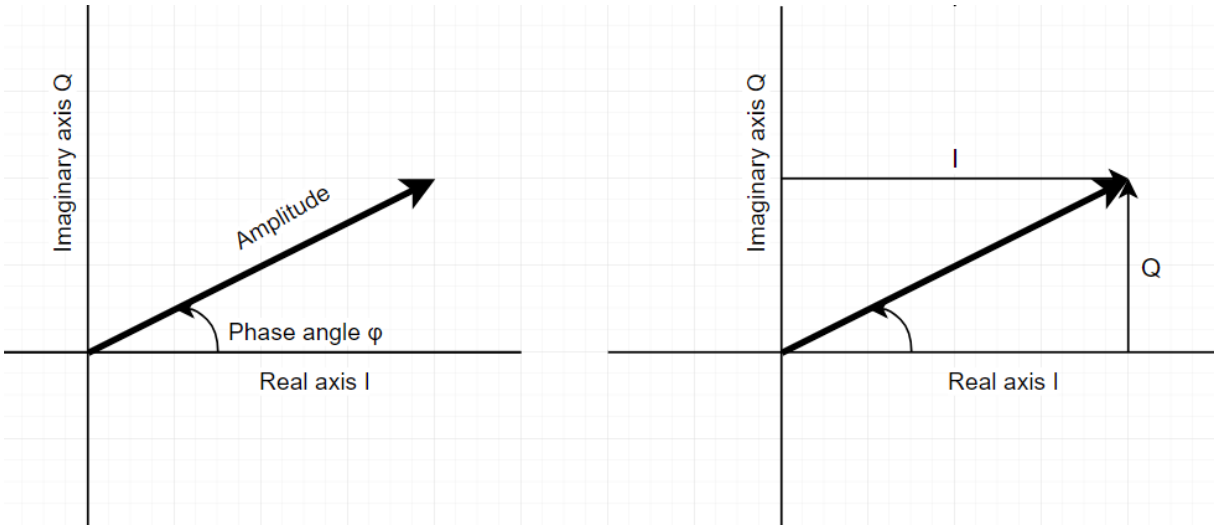

Figure 2.6: *Representation of sine wave in complex plain*

Using trigonometry, the polar coordinates can be converted into I and Q components of the signal using following equations:

- $I = Acos\,(2\pi ft)$,
- $Q = Asin\,(2\pi ft)$,

IQ data model is widely used in RF communication systems. It gives possibility to distinguish type of modulation used on carrier and allows introduce concept of positive and negative frequency. Amplitude and phase angle form seems to be more intuitive, however precisely varying the phase of a high-frequency carrier sine wave in a hardware circuit according to an input message signal is difficult. Therefore such hardware modulators will be expensive and hard to design and build. To avoid direct modulation of RF signal phase signal is decomposed to I and Q components.

According to Ptolemy's identitie for the cosine of sum:

$$cos\,(x+y) = cos\,(x)\,cos\,(y) - sin\,(x)\,sin\,(y)$$

sine wave carrier can be represented as:

$$Acos\,(2\pi ft + \phi) = Acos\,(2\pi ft)\,cos\,(\phi) - Asin\,(2\pi)\,.ft)sin(\phi)$$

Using equation 2.2 following formula is obtained:

$$Acos\,(2\pi ft + \phi) = Icos\,(2\pi ft) - Qsin\,(2\pi ft)\,,$$

where:

- I - is amplitude of in-phase signal,
- Q - is amplitude of quadrature signal.

Using this data samples representation modulation of phase of the RF signal is possible just by modulation of I/Q signals amplitudes and then mixing it with carrier and quadrature of carrier using mixers. Schematics below shows structure of IQ modulator and demodulator.

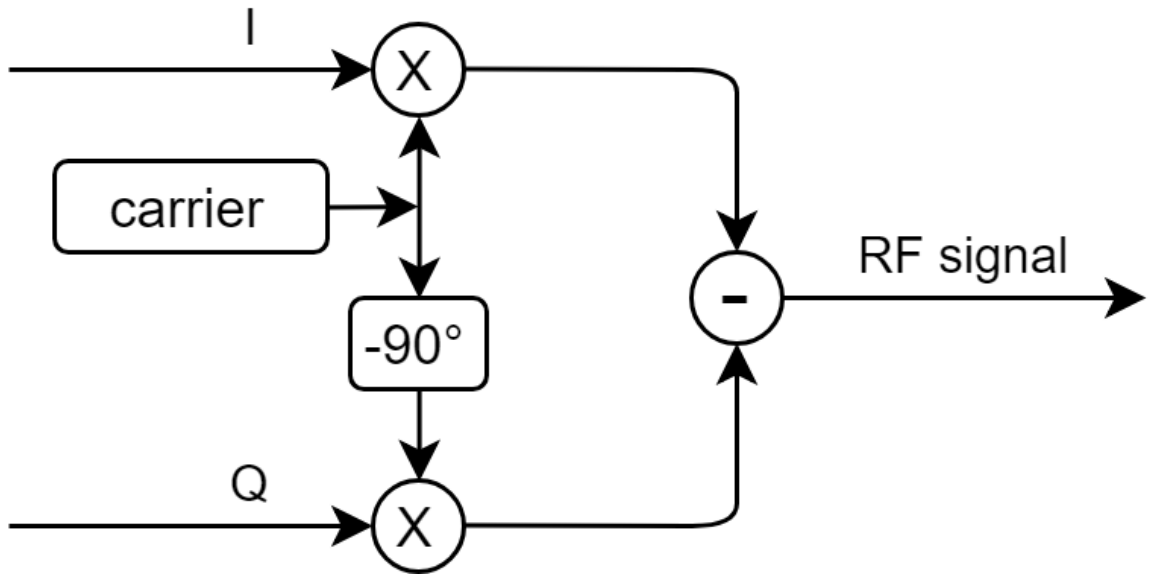

Figure 2.7: *Schematic of IQ modulator*

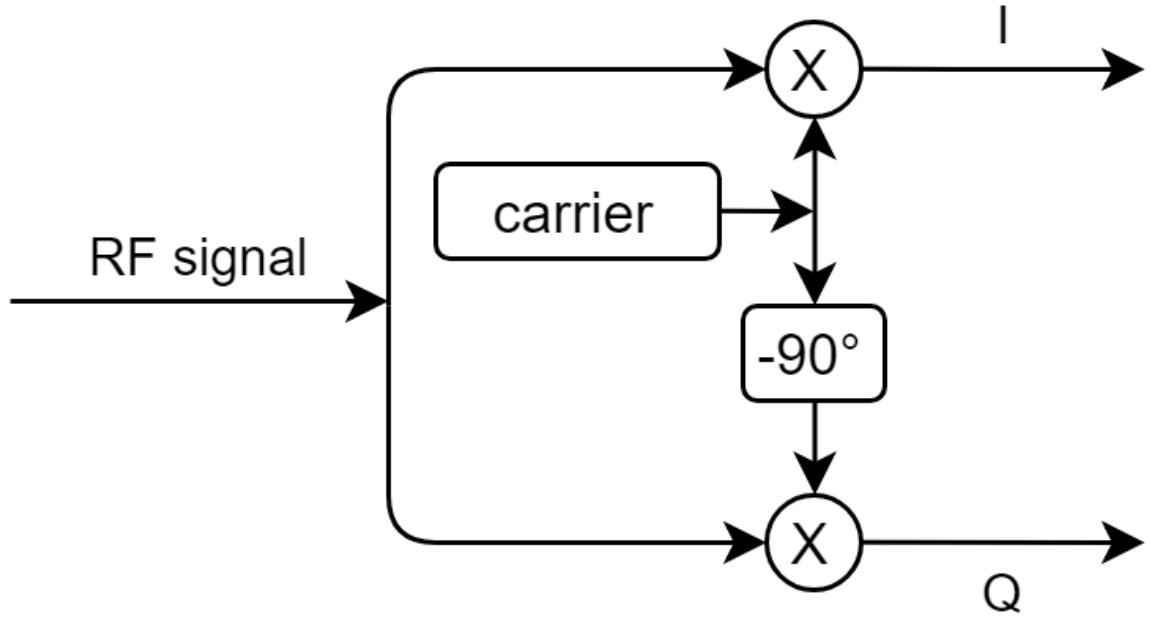

Figure 2.8: *Schematic of IQ demodulator*

The flexibility and simplicity of this solution compared to direct phase manipulations is a reason why I/Q modulators and demodulators are so widely used and popular in RF hardware.

2.3 IQ imbalance models

In this section two models of IQ mismatch imbalance are presented. First model is called Single-Branch IQ Imbalance Model (SBIQMB) and IQ imbalance is modeled as amplitude and phase error in only Q branch. Second is called Double-Branch IQ Imbalance Model (DBIQMB) and models IQ imbalance as amplitude and phase error in both branches.

Single-Branch IQ Imbalance Model

This model is characterized by mismatch existing only in Q branch. Amplitude error is described as g which results in $1-g$ represents total error in amplitude. Phase mismatch is modeled ϕ. This could be written as equation in following form [4]

$$\begin{bmatrix} I(t) \\ Q(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ -gsin(\phi) & gcos(\phi) \end{bmatrix} \begin{bmatrix} I_{RF}(t) \\ Q_{RF}(t) \end{bmatrix} \tag{2.1}$$

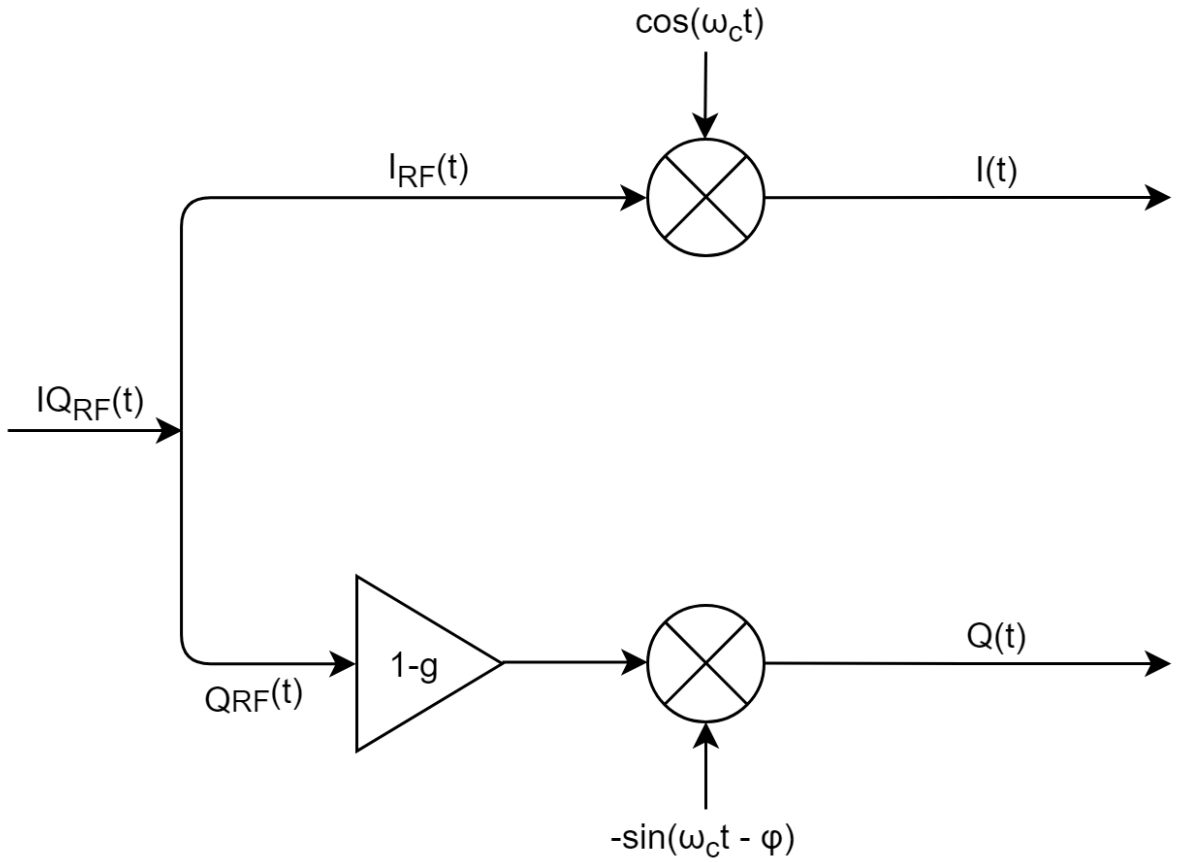

Figure 2.9: *Schematic of SBIQIM.*

where:

- $I_{RF}(t)$ and $Q_{RF}(t)$ IQ components of received signal.
- $I(t)$ and $Q(t)$ IQ components with introduced mismatch.

Double-Branch IQ Imbalance Model

This model is characterized by mismatch existing in Q both. In the I branch mismatch is described as amplitude error 1 + ϵ and phase error ϕ. Respectively in Q branch amplitude error is 1 - ϵ and phase error is $-\phi$ [4]. This can be rewritten as follows:

$$\begin{bmatrix} I(t) \\ Q(t) \end{bmatrix} = \begin{bmatrix} (1+\epsilon)cos(\phi) & (1+\epsilon)sin(\phi) \\ (1-\epsilon)sin(\phi) & (1-\epsilon)cos(\phi) \end{bmatrix} \begin{bmatrix} I_{RF}(t) \\ Q_{RF}(t) \end{bmatrix} \tag{2.2}$$

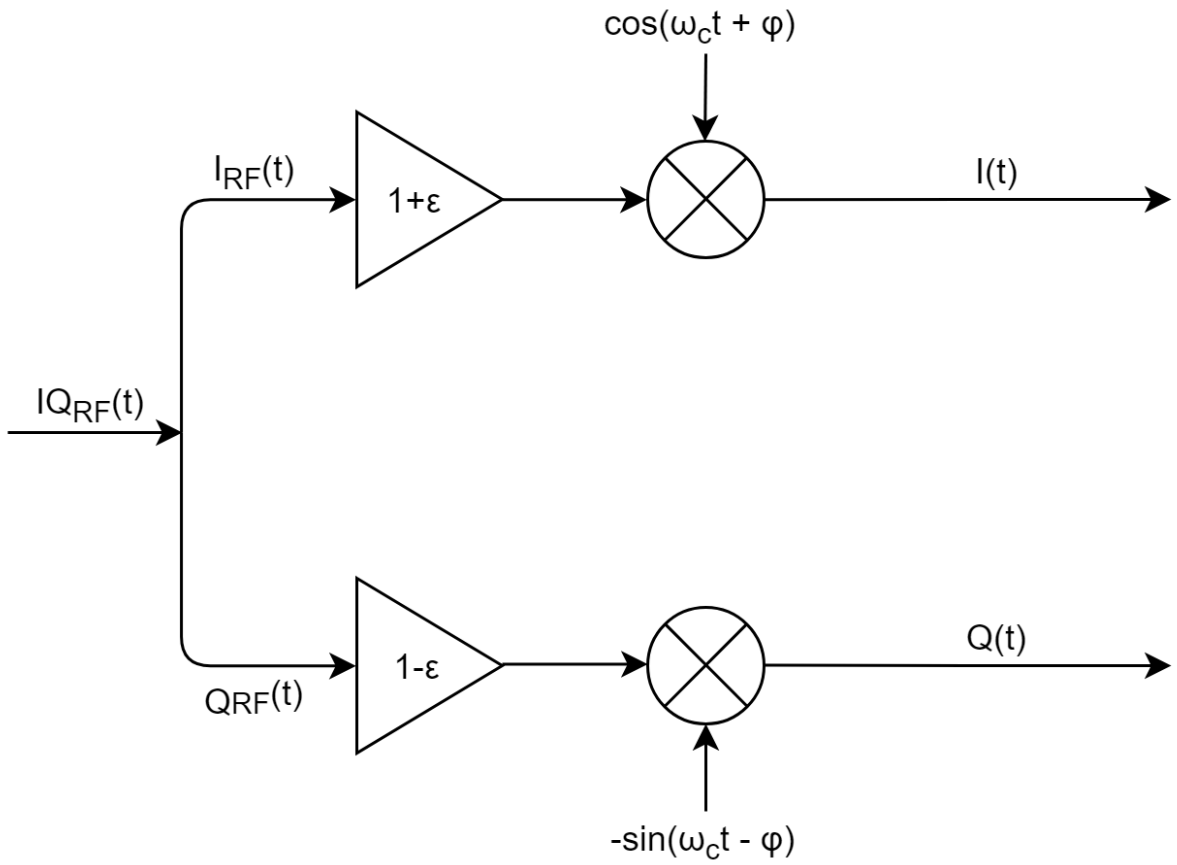

Figure 2.10: *Schematic of DBIQIM.*

where:

- $I_{RF}(t)$ and $Q_{RF}(t)$ IQ components of received signal.
- $I(t)$ and $Q(t)$ IQ components with introduced mismatch.

Relation between SBIQIM and DBIQIM

As mentioned before both models describes IQ mismatch in different ways. DBIQIM is equivalent to SBIQIM processed by two additional matrices. Matrix G that represents gain and matrix R which represents rotation. Equations below shows relation between SBIQIM and DBIQIM [4]

As mentioned earlier:

$$SBIQIM = \begin{bmatrix} 1 & 0 \\ -gsin(\phi) & gcos(\phi) \end{bmatrix}$$

$$DBIQIM = \begin{bmatrix} (1+\epsilon)cos(\phi) & (1+\epsilon)sin(\phi) \\ (1-\epsilon)sin(\phi) & (1-\epsilon)cos(\phi) \end{bmatrix}$$

$$DBIQM = G \cdot SBIQIM \cdot R \tag{2.3}$$

G and R are defined as follows:

$$G = \frac{2}{1+g} \tag{2.4}$$

$$R = \begin{bmatrix} cos(-\frac{\phi}{2}) & sin(-\frac{\phi}{2}) \\ -sin(-\frac{\phi}{2}) & cos(-\frac{\phi}{2}) \end{bmatrix} \tag{2.5}$$

Using equation above transition between models parameters works as follows:

- Calculation of DBIQIM parameters based on SBIQIM.

$$\epsilon = \frac{1-g}{1+g}$$

$$\phi_{DBIQIM} = -\frac{\phi_{SBIQIM}}{2}$$

- Calculation of SBIQIM parameters based on DBIQIM.

$$g = \frac{1-\epsilon}{1+\epsilon}$$

$$\phi_{SBIQIM} = -2\phi_{DBIQIM}$$

2.4 Software Defined Radio

SDR (Software Defined Radio) is a radio communication system where components typically implemented in hardware (e.g. mixers, filters, amplifiers, modulators/demodulators), are instead implemented by the means of software [6].

Below the block diagram of SDR device is presented:

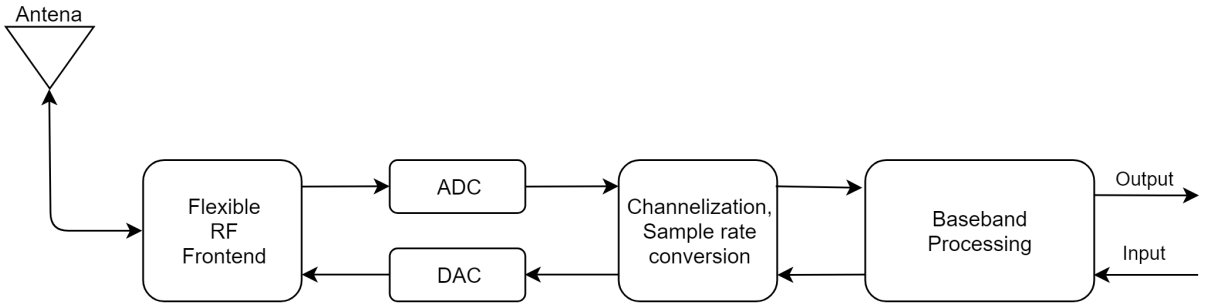

Figure 2.11: *Block diagram of SDR device.*

- **Flexible RF fronted** is a first stage of SDR processing system. This part is configured by the means software to match current requirements for best system performance. This is possible by using numerically controlled components. NCO (Numerically Controled Oscilators) which are used as flexible frequency generators. This component are essential for I/Q modulators/demodulators as flexible carrier generators for mixers. Bandpass filters and equalizers used for signal conditioning.

- **ADC / DAC converters**. This is where captured signal is converted from analog to digital domain and likewise transmited is converted form digital to analog domain. This is a last stage before before purely digital domain of the system.

- **Channelization and Sample rate conversion**. Here the data processed by the baseband system are assigned to respective communication channels, and conditioned for specific channel transmission parameters like frequency and sampling rate.

- **Baseband Processing System** Digital core of the SDR system. This may by either processor, FPGA, ASIC or SoC. Variety of choice in this case allows creation of well-suited system for desired performance and cost range. Mentioned component is responsible with interfacing with external system for data exchange purpose. All logic related to transmission is implemented here in the software including correction algorithms and application layer. Thanks to that system can adapt to changes. This means changing software to improve quality of the processed signal or even complete change of the application layer, by simply providing new software.

SDR devices are characterized by very fast ADC and DAC converters which are necessary for direct conversion. For example in secondary air plain surveillance systems (ADS-B), the carrier frequency is 1.09GHz. By Nyquist law this requires at least two times higher frequency sampling which is 2.18GHz.

Chapter 3

Hardware and tools

This chapter describes chosen hardware, tools and explains reasons behind such decisions.

3.1 Adalm Pluto and AD tools

Adalm Pluto Board

Adalm Pluto is learning module from Analog Devices that can be used to introduce principles of operation and theory behind SDR and RF communication.

Figure 3.1: *Photo of Adalm Pluto device [17].*

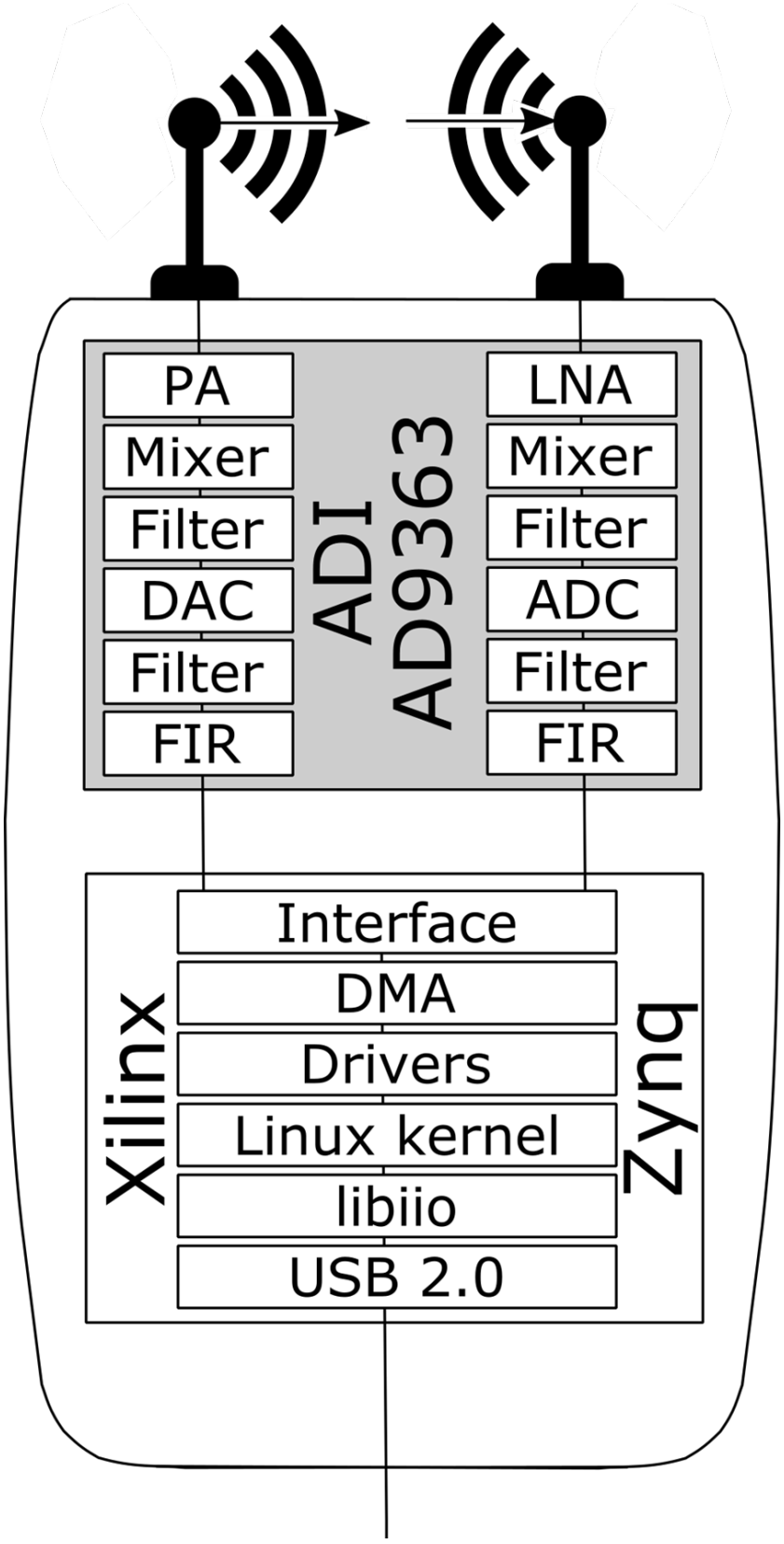

Figure 3.2: *Schematic Adalm Pluto hardware [?].*

Adalm Pluto is already integrated with with MATLAB, Simulink, GNU Radio, and libIIO which allows to interface with the devices using C, C++, C# or Python programming languages. PlutoSDR is open software repository with all software and tools related to Adalm Pluto including: HDL Project in Xillinx Vivado and Buildroot configuration

for embedded Linux. Main reason for choosing this SDR device was relatively low price 136.80USD in comparative to other available solutions on the market. Adalm Pluto core is Zynq7010 SoC which is combination of ARM based processor capable of running Linux and FPGA part which can communicate with the processor via AXI interface.

AD9363 Agile RF Transceiver

The main part of Adalm Pluto SDR is AD9363. AD9363 is a high performance, highly integrated RF agile transceiver designed for use in 3G and 4G femtocell applications.

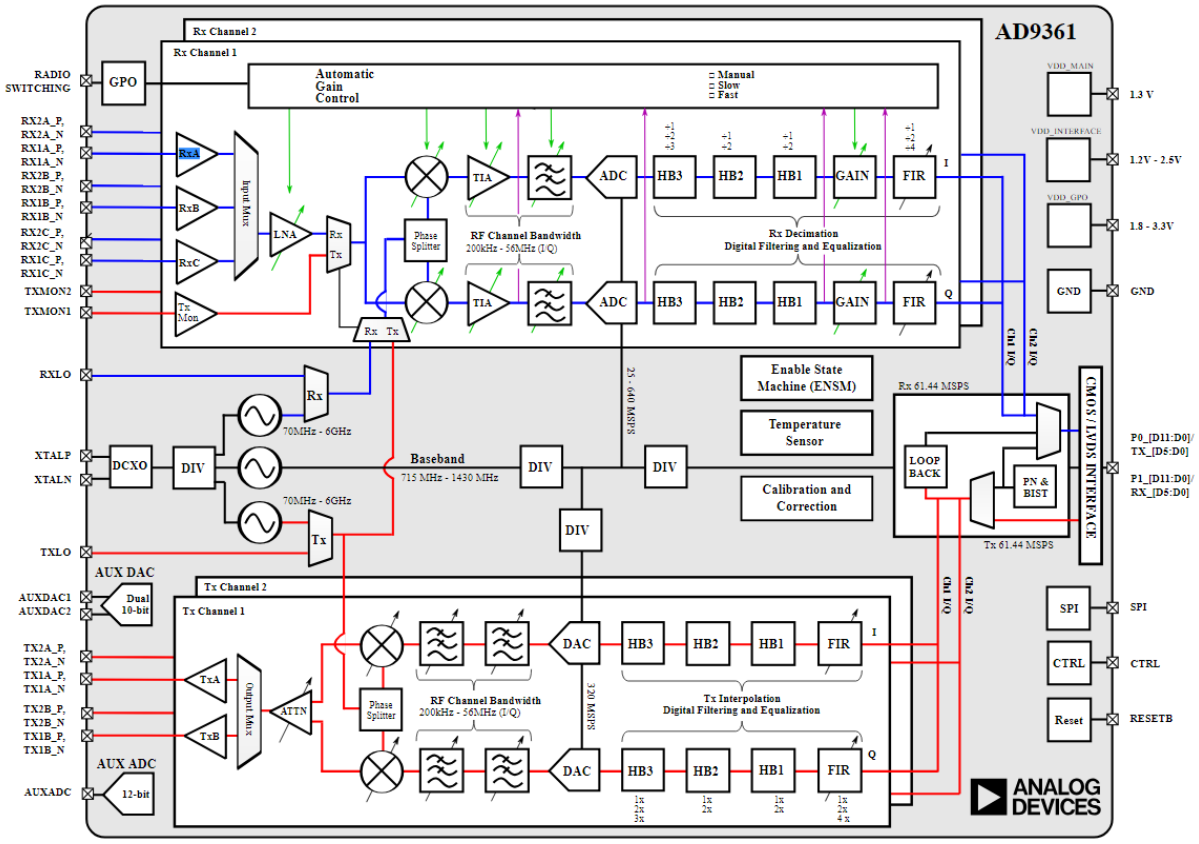

Figure 3.3: *Block diagram of AD9363 RF transceiver [19].*

Ad9363 features:

- Wide bandwidth from 325Mhz to 3.8GHz,
- Tunnable channel bandwidth up to 20MHz,
- Independent AGC (Automatic Gain Control),
- Full duplex communication,
- DC offset and I/Q mismatch tracking.
- Flexible rate, 12-bit ADC and DAC

IIO Osciloscope

The ADI IIO Oscilloscope is application which allows interfacing with different evaluation or custom made board based on AD agile RF Transrecivers. This program allows communication with the device connected with PC using:

- Local or remote network,
- USB 2.0 interface,
- Serial Port.

Application supports full configuration of the device including: configuration receiver frequency and channel bandwidth. Access to DC offset tracking, I/Q correction, internal calibration functionality and all registers. Moreover allows to process revived using filters designed in matlab and four gain control modes:

- manual configuration - gain is selected by user,
- slow attack approach,
- fast attack approach,
- hybrid attack approach,

Program allows to plot received data from selected channels in four different modes:

- time domain as I and Q signals,
- frequency domain with configurable FFT and averaging size,
- constellation as relation between I and Q signals,
- cross-correlation for multi-channel boards.

3.2 libIIO

libIIO is library developed by Analog devices for interfacing Linux IIO (Industrial Input Output) devices.

Library is composed of three main parts:

- local backend - responsible for interfacing with the Linux kernel using sysfs virtual file system,
- network backend - which communicates with the IIOD (IIO Deamon) by network link.

The IIOD is libiio based application that creates libiio context which is using local backend and then connects it to network backend. Described process allows to share local context with the server and makes device accessible via network

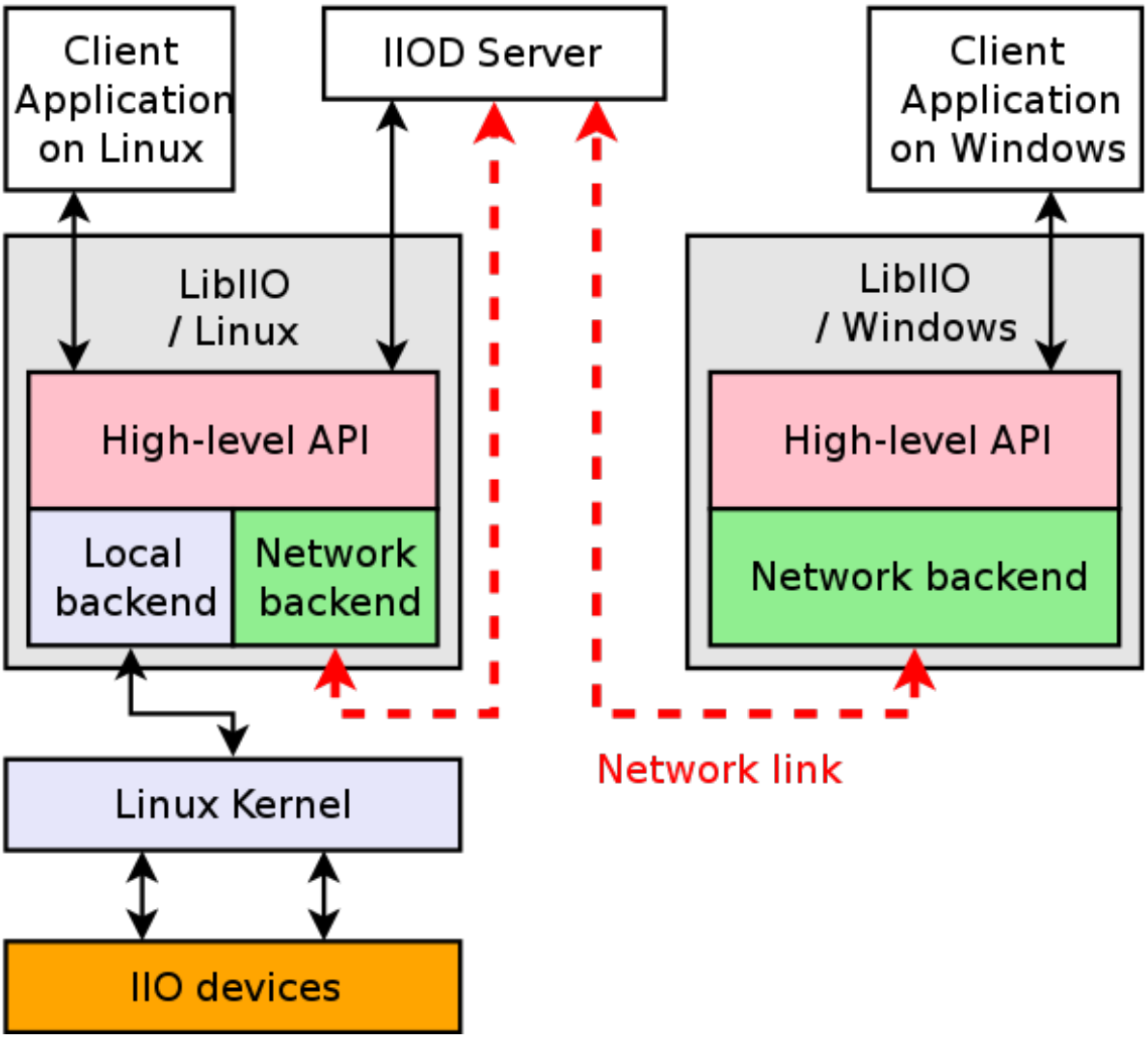

Figure 3.4: *Block diagram of libIIO working principles. [21]*

3.3 Zynq and Xilinx tools

Introduction to SoC

SoC is aberration from System on Chip. Such devices aggregates many types of hardware in single chips including: ADCs, DACs, internal memory, external memory controllers, peripheral interfaces such as SPI, UART, I2C and many others. The main idea is that single silicon chip may be used to implement functionality of entire system. Such approach results in cheaper and faster solution than realizing such functionality on PCB using separate components. In the past such role belonged to ASIC. (Application Specific Integrated Circuits). The major disadvantages of such solution is lack of flexibility and significant development cost and time. This makes ASIC's sustainable only on the high volume market where no future upgrade will be required. These limitations created clear need for more flexible device with faster development time. This need has been long satisfied by a FPGAs. FPGA can be reconfigured as desired which means virtually no risk in deploying solution which may require upgrade based on FPGA. Next step are SoC based solution. SoC is combination of processor and FPGA. This allows to create fast

application dependent on the hardware functionality. Moreover processor can run normal operating system which allows fast development and flexibility of the solution.

Zynq-7000 family SoC

The Zynq is new kind of SoC from Xillinx. It combines both applications processor and FPGA fabric. The Zynq device comprises two sections: PS (Processing System) and PL (Programmable Logic). This sections can be used separately for independent task or together to utilize advantages of both software and hardware. The Zynq devices are meant to use structure of both sections and interface between them. Connection between these parts is provided by AXI (Advanced eXtensible Interface) which is registered under ARM trademark.

Processing System in all Zynq devices has the same architecture, and the base of it is a dual-core ARM Cortex-A9 processor. This is a hard processor which means it is manufactured directly in silicon structure. Zynq allows to use soft processors like PicoBlaze and MicroBlaze which can be implemented in PL sections. Depending on the size and speedgrade of the device ARM processor is accompanied by set of processing resources e.g (MMU, DMA, SRAM, Processing System External Interfaces, cache memory, control registers , etc.) which forms together APU (Application Processing Unit).

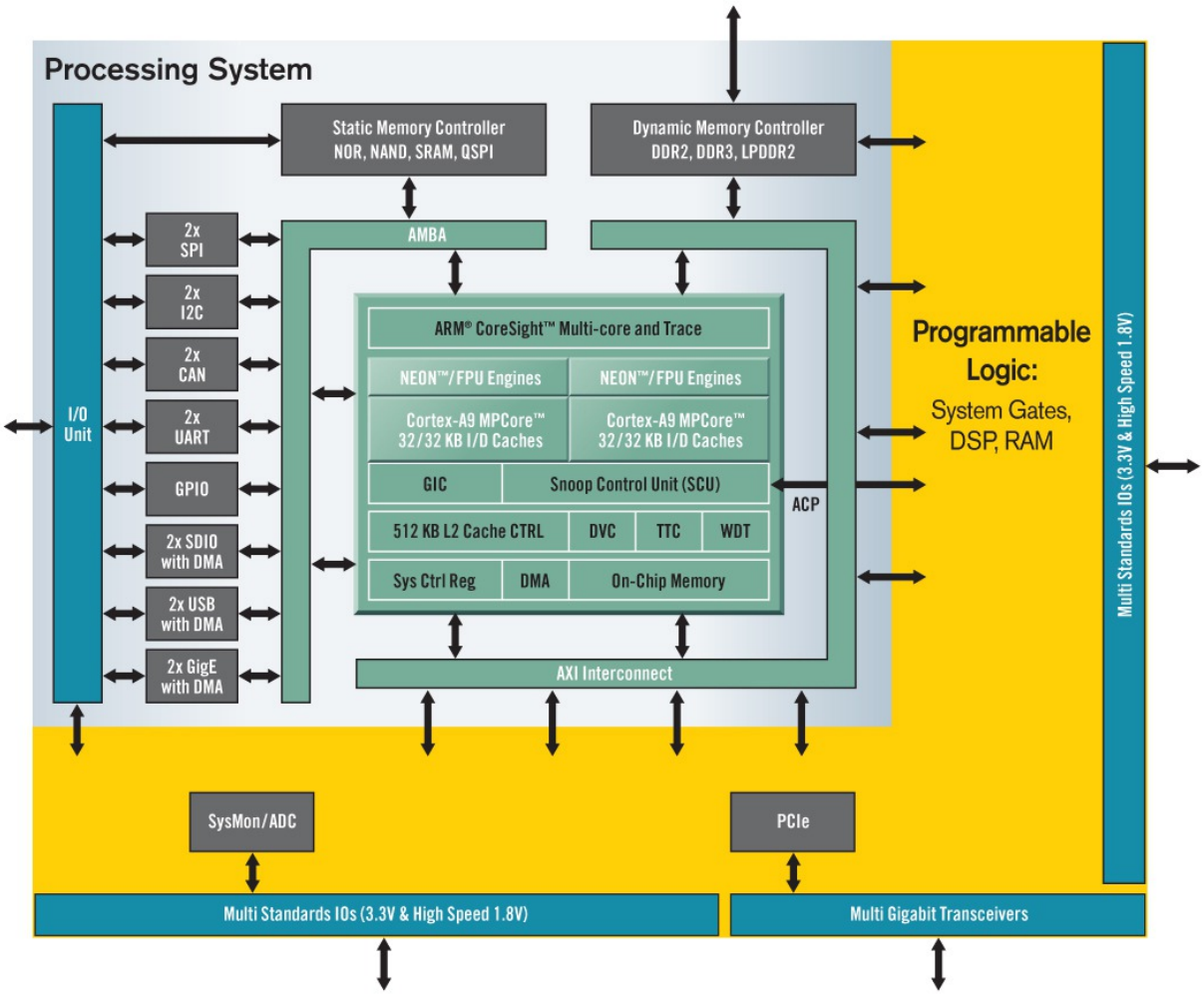

Figure 3.5: *Block diagram fo Zynq 7000 family SoC. [20]*

Programmable Logic is a FPGA based on Atrix-7 and Kintex-7 fabric from Xillinx. PL consist of CLBs (Configurable Logic Blocks), slices, IOBs(Input Output Blocks), LUTs (Lookup tables), fliplops and switch matrices. Additionally there are BRAM (Block Random Access Memory) which are used to store large amout of date in small part of the device which allows fast access to its content. Second additional resource is DSP48E1 slice. This is advanced DSP module that allows many complex mathematical operation in a single clock cycle.

Xillinx Vivado Tools

As a part of Zynq-7000 family SoC design flow. Xillinx Vivado 2018.2 and Xilinx Vivado HLS 2018.2 where choosen. This version of the software is tested and compatible with latest release of the plutosdr-fw repository.

Xillinx Vivado is a tool for FPGA design. This program allows full configuration of PL part of the Zynq-7000 family SoC including: synthesis, implementation, routing, timing validation and generation of the bitstream. Bitstream is used to deploy configuration onto FPGA device.

Xilinx Vivado HLS is a program for High Level Synthesis for Zynq-7000 family SoC. This tool allows to write and test software in C language. After proper validation in the software domain, code can be synthesized to HDL and exported to Xillinx Vivado as an IP Core. This allows much faster design than direct HDL development and much easier testing. These were main reason for choosing HLS environment.

Chapter 4

Algorithms

This chapter describes all algorithms used for signal quality improvement. All of considered algorithms are blind. This means that signal analysis is purely statistical with no prior information about signal. Advantages of this approach is that this algorithms can be used for any type of possible signal.

4.1 DC offset correction

This section describes algorithms used for removing DC offset from revived samples.

Moving Average Filter

Moving Average Filter is simple FIR (Finite Impulse Response) filter. This filter is commonly used for white noise removal and signal smoothing. However when filter order is great enough to account for samples from full period of the signal filters the average represents signal DC offset.

The filter equations is presented below [9]:

$$y[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[n-k], \tag{4.1}$$

where:

- N is order of the filter.
- $y[n]$ is filtered sample at n-th step,
- $x[n]$ is n-th sample from receiver.

DC components of I and Q branches are calculated using equation 4.1, where $x[n-k]$ is substituted with corresponding I or Q sample.

$$I_{mean}[n] = \frac{1}{N} \sum_{k=0}^{N-1} I[n-k]$$

$$Q_{mean}[n] = \frac{1}{N} \sum_{k=0}^{N-1} Q[n-k]$$

Corrected values are calculated as follows:

$$I_{corr}(n) = I(n) - I_{mean}(n),$$
$$I_{corr}(n) = Q(n) - Q_{mean}(n)$$

Algorithm introduces delay by N samples in the signal. It is important to that the number of samples taken into consideration must extend at least one period of the signal. Otherwise filter will works white noise attenuator.

Normalized Gaussian Filter

Normalized Gaussian Filter is filter whose impulse response has shape of Gaussian function with all values in range from 0 to 1.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-x}{2\sigma^2}}$$

where:

- x is
- σ is standard deviation.

The filter equations is presented below [7]

$$y[n] = \sum_{k=0}^{N-1} x[n-k]G(x) \tag{4.2}$$

- N is windows size,

DC components of I and Q branches are calculated using equation 4.2, where $x[n-k]$ is substituted with corresponding I or Q sample.

$$I_{mean}[n] = \sum_{k=0}^{N-1} x[n-k]G(x)$$

$$Q_{mean}[n] = \sum_{k=0}^{N-1} x[n-k]G(x)$$

Corrected values are calculated as follows:

$$I_{corr}(n) = I(n) - I_{mean}(n),$$
$$I_{corr}(n) = Q(n) - Q_{mean}(n)$$

This filter has better performance in frequency domain than Moving Average Filter.

4.2 IQ Mismatch Correction

The IQ imbalance is common problem in RF front-ends that uses analog quadrature down- mixing. The ideal down-converter performs only simple frequency shift. Real down- converters introduces image interference which may be assumed as constant. But amplifiers and filters introduces varying with frequency imbalance.

In ideal case the receiver output is:

$$\begin{aligned} I(t) &= cos(\omega t) \\ Q(t) &= sin(\omega t) \end{aligned} \tag{4.3}$$

where:

- ω is tone frequency.

I and Q components are orthogonal with respect to each other. In real case receiver output is:

$$\begin{aligned} I(t)' &= \alpha cos(\omega t) + \beta_I \\ Q(t)' &= sin(\omega t + \phi) + \beta_Q \end{aligned} \tag{4.4}$$

where:

- α is magnitude mismatch,
- ϕ is phase imbalance,
- β_{IQ}, β_Q is signal DC offset.

Input of phase correction algorithms is signal with DC offset extracted using algortihms from previews section. Hence the signal model is [3]

$$\begin{aligned} I(t)'' &= \alpha cos(\omega t) \\ Q(t)'' &= sin(\omega t + \phi) \end{aligned} \tag{4.5}$$

According to Ptolemy's identitie for the sine of sum:

$$sin\left(\omega t + \phi\right) = sin\left(\omega t\right))cos\left(\phi\right) + cos\left(\omega t\right) sin\left(\phi\right)$$

Equation 4.5 be rewritten in matrix form as:

$$\begin{bmatrix} I(t)'' \\ Q(t)'' \end{bmatrix} = \begin{bmatrix} \alpha & 0 \\ sin(\phi) & cos(\phi) \end{bmatrix} \begin{bmatrix} I(t) \\ Q(t) \end{bmatrix} \tag{4.6}$$

After multiplying both sides of 4.5 by inversion of parameters matrix following set of exuations is obtained:

$$\begin{bmatrix} I(t) \\ Q(t) \end{bmatrix} = \begin{bmatrix} \alpha^{-1} & 0 \\ \alpha^{-1}tan(\phi) & sec(\phi) \end{bmatrix} \begin{bmatrix} I''(t) \\ Q''(t) \end{bmatrix} \tag{4.7}$$

This shows that only α and ϕ must be found to perform I/Q mismatch compensation. According to this paper:

$$< I''(t)I''(t) >= \frac{1}{2}\alpha^2$$

$$\alpha = \sqrt{2 < I''(t)I''(t) >}$$

$$< I''(t)Q''(t) >= \frac{1}{2}\alpha^2 sin(\phi)$$

$$sin(\phi) = \frac{2}{\alpha} < I''(t)Q''(t) >$$

Assuming that $|\phi| \leq \frac{pi}{4}$ $cos(\phi)$ can be obtained directly from sin(ϕ) using following formula:

$$cos(\phi) = \sqrt{1 - sin(\phi)}$$

Using following parameters in we can substitute matrix equations 4.6 with:

$$\begin{bmatrix} I(t) \\ Q(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{\alpha} & 0 \\ \frac{-sin(\phi)}{acos(\phi)} & \frac{1}{cos(\phi)} \end{bmatrix} \begin{bmatrix} I''(t) \\ Q''(t) \end{bmatrix} \tag{4.8}$$

The IQ mismatch correction can be now applied using 4.8 formula.

Chapter 5

Hardware Implementation

This chapter explains structure of hardware implemented in Zynq PL section. This includes data flow from PS to AD9361 RF transceiver, including correction algorithms for RF signal quality improvement.

5.1 Data Flow Path

Diagram below shows data path for I/Q samples from Zynq PS system to PL logic.

- *AXI_AD9361* - IP core provided by Analog devices. This block is responsible for interfacing with AD9363 RF transceiver placed on board.

- *FIR Decimator* - down-sampling block with additional filters to block unwanted images created by change of sampling frequency in receive path.

- *FIR Interolator* - up-sampling block with additional filters to block unwanted images created by change of sampling frequency in transmit path.

- *DC Offset Removal* - implementation of dc offset removal algorithm. Depending on test bench Moving Average or Gaussian Filter.

- *I/Q Correction* - realization of phase and magnitude correction in the revived signal.

- *Zynq Processing System* - CPU part of Zynq SoC. Mentioned block is responsible for communication between PL and PS part using AXI interface. Zynq PS section perform task related to data aquisition and interfacing with PC using USB 2.0 interface.

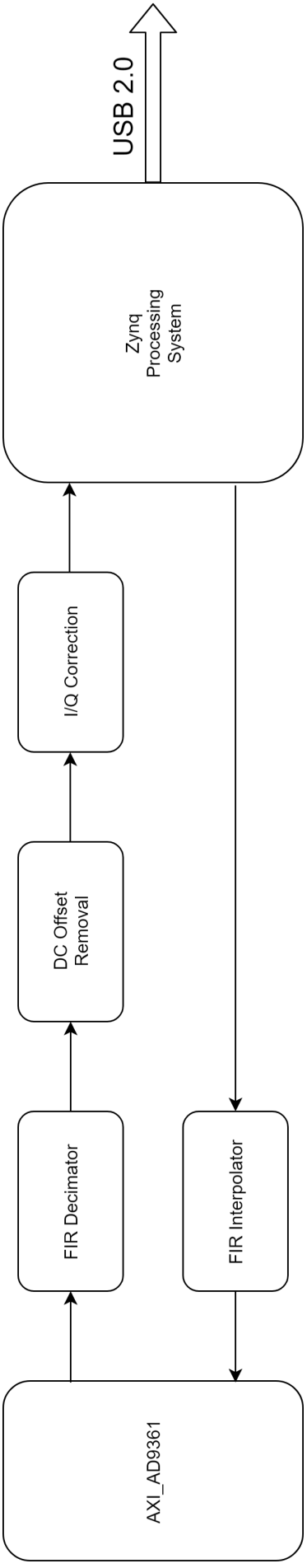

Figure 5.1: *Block diagram of the PS logic structure.*

5.2 Moving Average Filter

Table 5.1

Clock	Period	Estimated	Uncertanity
100MHz	10ns	8.358	1.25

Table 5.2: Timing estimation of C synthesis for Moving Average Filter.

Table 5.3

Latency		Interval	
min	max	min	max
1	1	2	2

Table 5.4: Latency estimation of C synthesis for Moving Average Filter.

Table 5.5

BRAM_18K	DSP48E	FF	LUT
2	0	102	418

Table 5.6: Resource utilization for Moving Average Filter.

5.3 Gaussian Filter

Table 5.7

Clock	Period	Estimated	Uncertanity
100MHz	10ns	8.358	1.25

Table 5.8: Timing estimation of C synthesis for Gaussian Filter.

Table 5.9

Latency		Interval	
min	max	min	max
1	1	2	2

Table 5.10: Latency estimation of C synthesis for Gaussian Filter.

Table 5.11

BRAM_18K	DSP48E	FF	LUT
2	5	599	1770

Table 5.12: Resource utilization for Gaussian Filter.

5.4 I/Q mismatch compensation

Table 5.13

Clock	Period	Estimated	Uncertanity
100MHz	10ns	8.429	1.25

Table 5.14: Timing estimation of C synthesis for IQ mismatch compensation.

Table 5.15

Latency		Interval	
min	max	min	max
119	119	2	2

Table 5.16: Latency estimation of C synthesis for IQ mismatch compensation.

Table 5.17

BRAM_18K	DSP48E	FF	LUT
6	20	6145	10412

Table 5.18: Resource utilization for IQ mismatch compensation.

DC offset removal algorithms are relatively cheap in terms of resource utilization. Gaussian Filter takes additional DSP48E units. IQ mismatch cimpensation algorithm is the most expensive due to its complexity.

Chapter 6

Performance evaluation

This chapter presents results of simulation, hardware implementation and built in functionality of AD9363 in IQ imbalance compensation. For all of these approaches three different cases are considered:

- single tone - signal consisting of only one frequency,
- multi tone - signal consisting of two different frequencies,
- broadband - signal with band of specified width.

In case of testing real hardware, device is transmitting signal to itself through antennas included with the Adalm Pluto.

6.1 Simulations

In this section results of the algorithms implemented in the Matlab are tested in two cases: without additional noise and with noise of normal distribution and amplitude of 0.1. Algorithms were tested in the following conditions:

- Amplitude - 1,
- DC offset in I branch - 0.1,
- Phase mismatch - 3°.
- Moving Average Filter Order - 256.
- Gaussian filter $\sigma = 5$, window size = 30

DC offset removal

In this section simulations of DC offset correction algorithms are compared for signle tone signal of frequency $f = 1kHz$ with and without additional noise.

Scenario 1: Noiseless signal

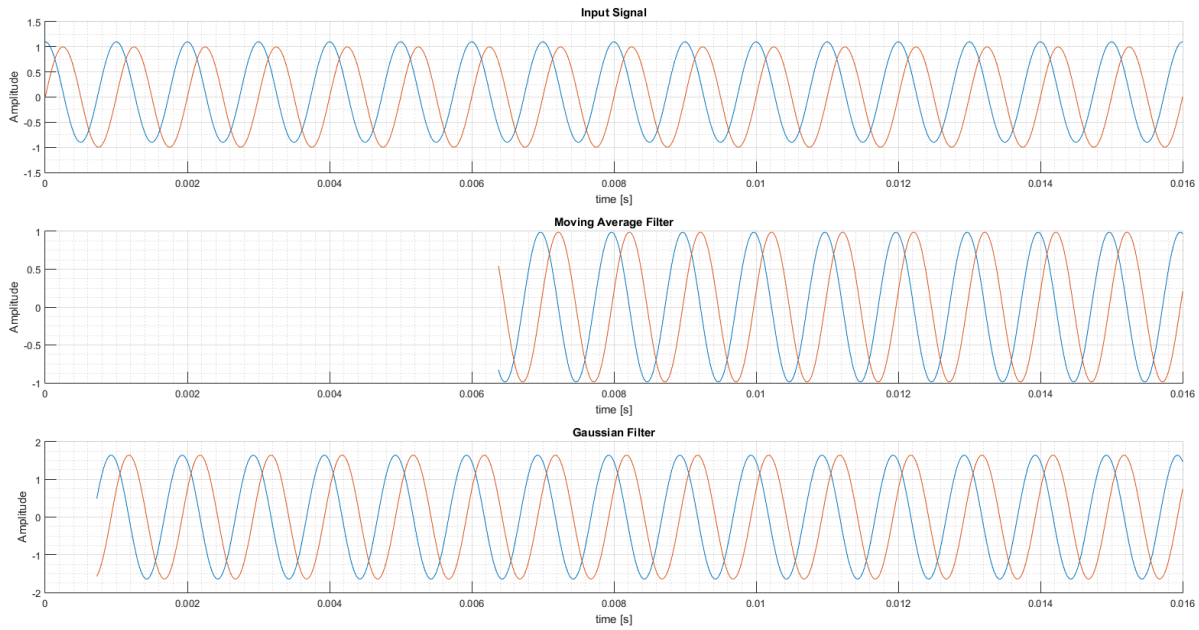

Figure 6.1: *Comparison of DC offset algorithms outputs in time domain.*

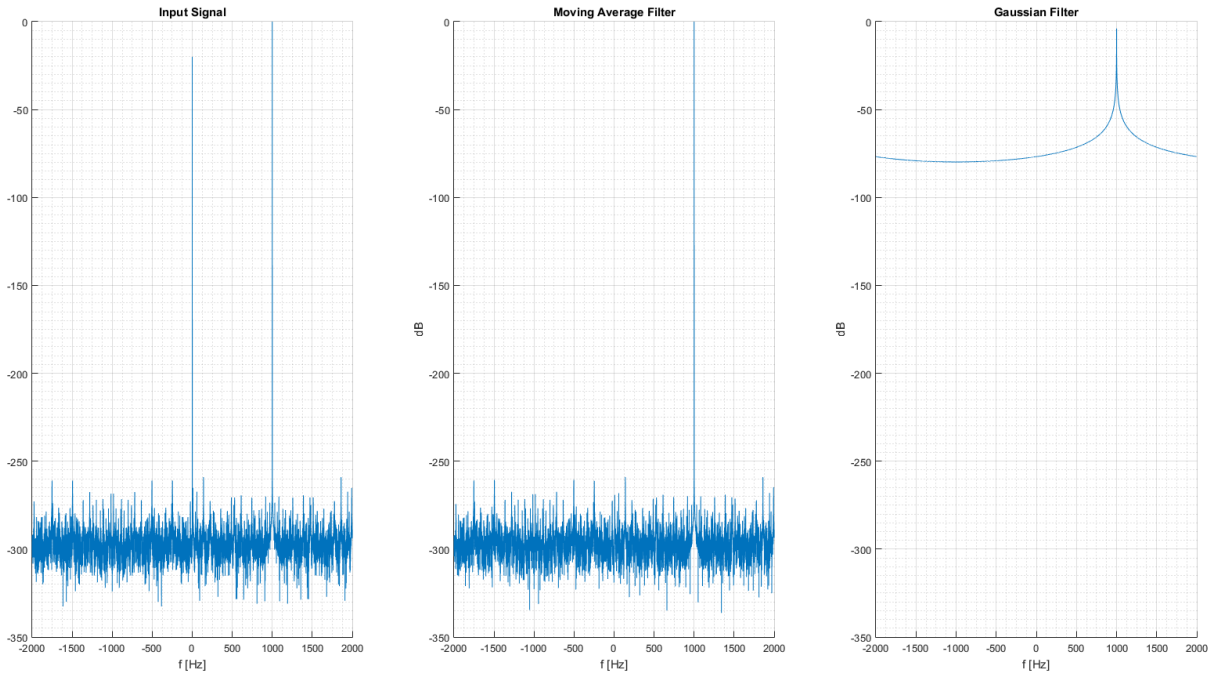

Figure 6.2: *Comparison of DC offset algorithms outputs in frequency domain.*

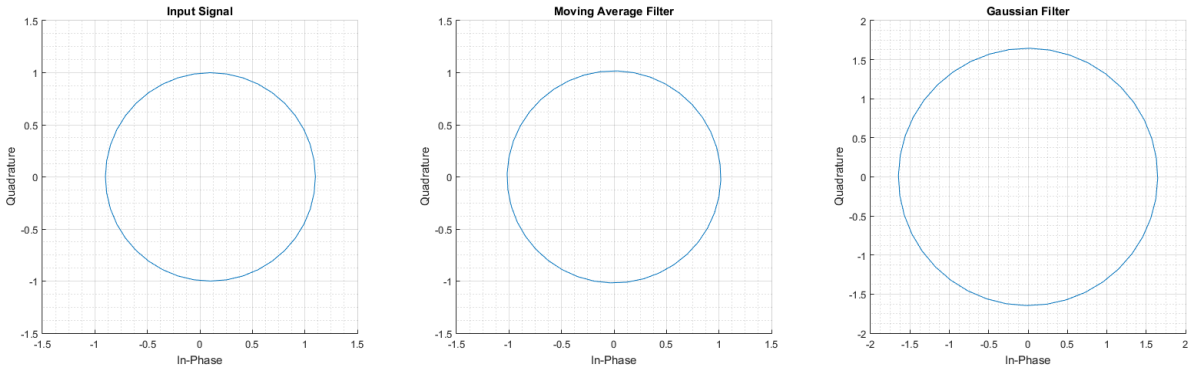

Figure 6.3: *Comparison of constellation diagram for tested algorithms.*

Scenario 1: Noisy signal

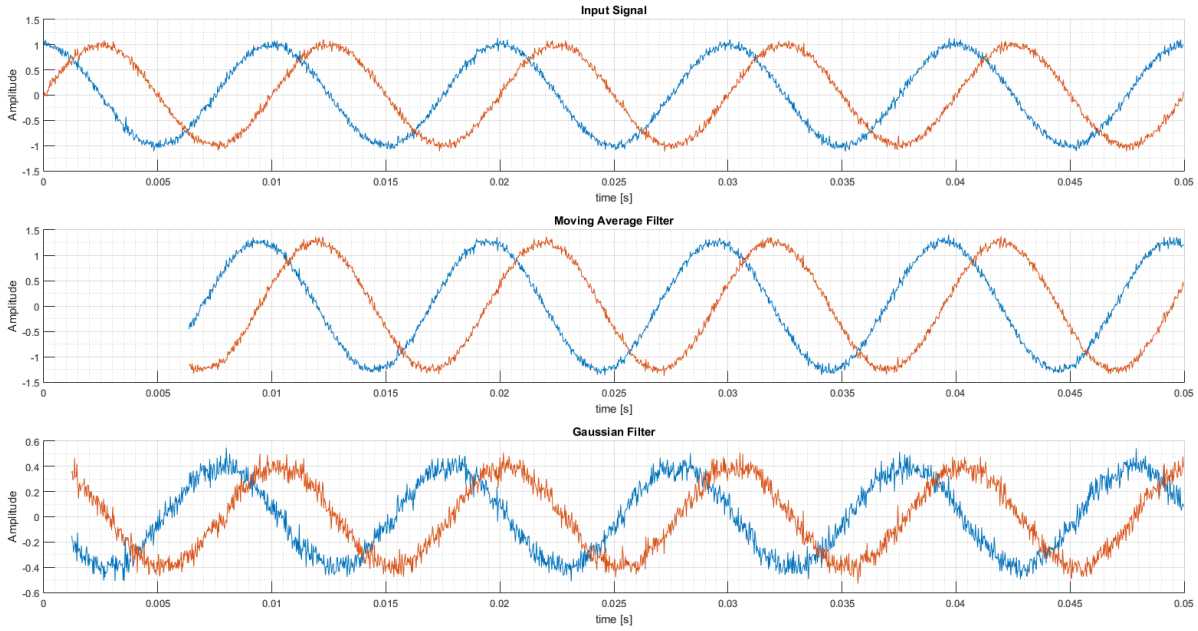

Figure 6.4: *Comparison of DC offset algorithms outputs in time domain.*

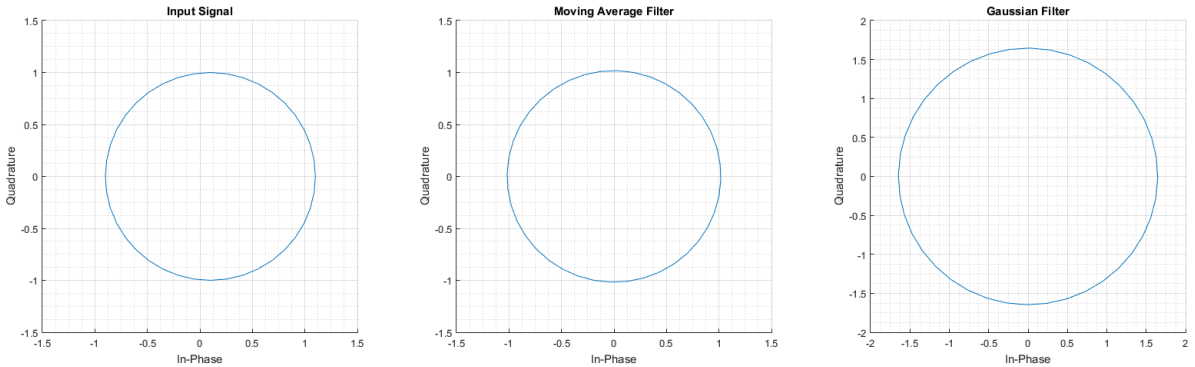

Figure 6.5: *Comparison of constellation diagram for tested algorithms.*

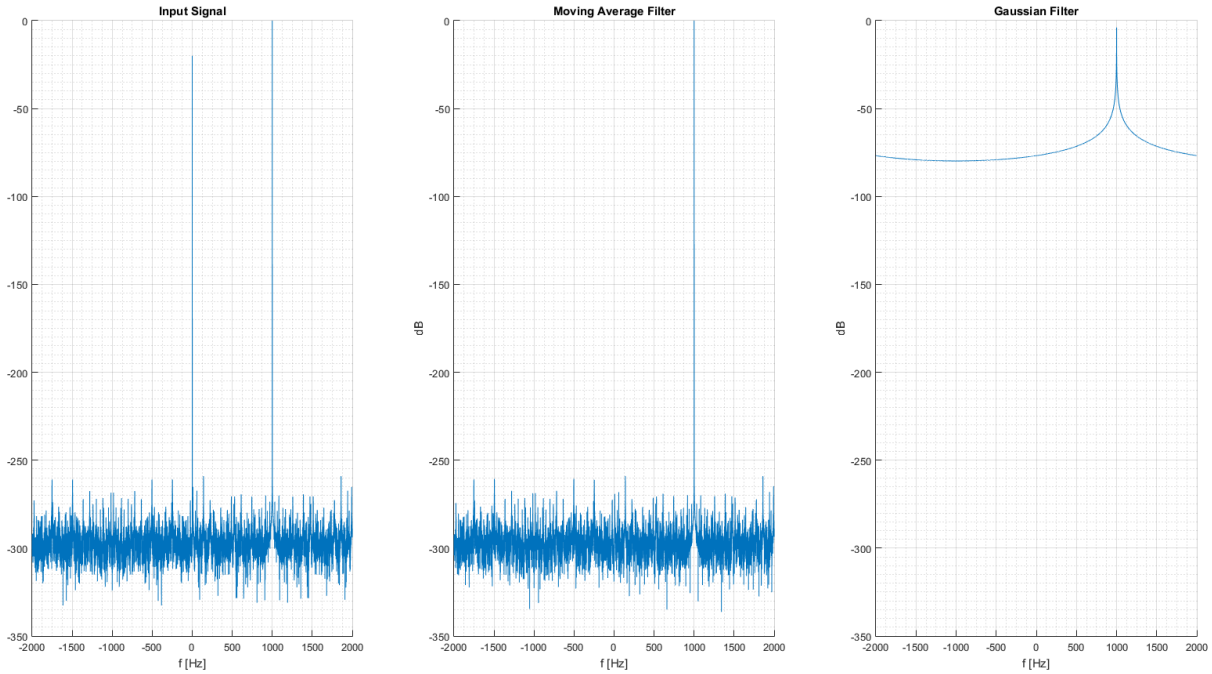

Figure 6.6: *Comparison of DC offset algorithms outputs in frequency domain.*

Gaussian filters needs almost three times lower number of samples to work correctly in comparison to Moving Average Filter. It means lower delay in processing path. Gaussian filter distort magnitude of the signal, but this problem will be resolved by Magnitude and Phase correction algorithm.

Single Tone Signal

In this case IQ mismatch compensation with two DC offset removal algorithms are compared for single tone signal with frequency $f = 1kHz$ with and without additional noise.

Scenario 1: Noiseless signal

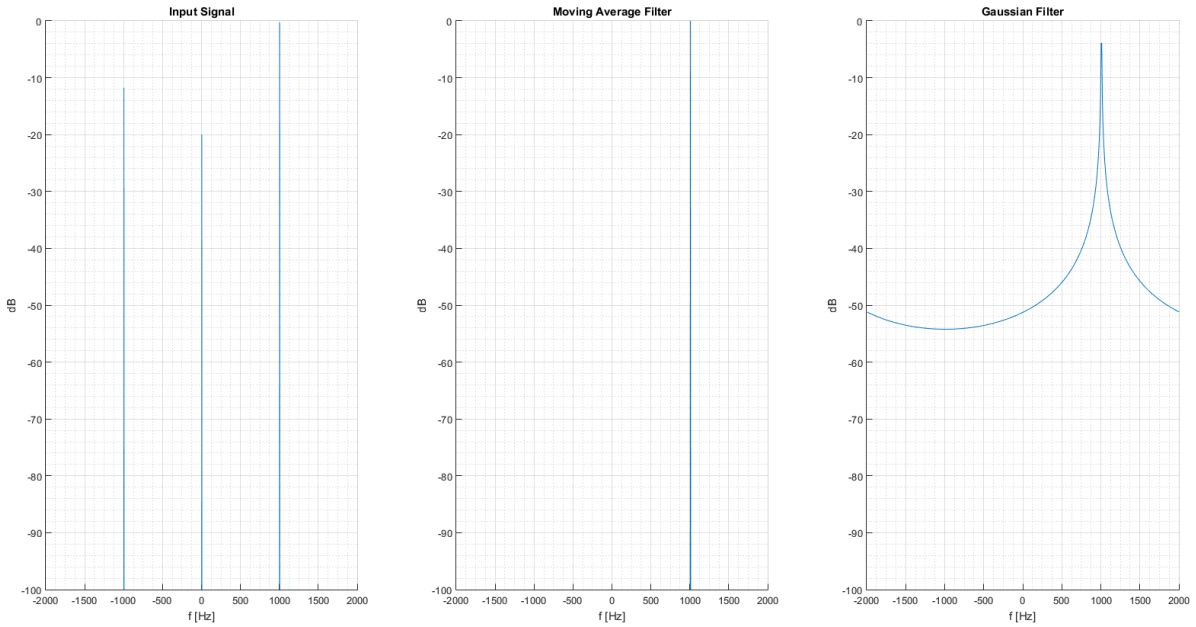

Figure 6.7: *Comparison of spectrum for IQ mismatch compensation algorithms for signle tone signal.*

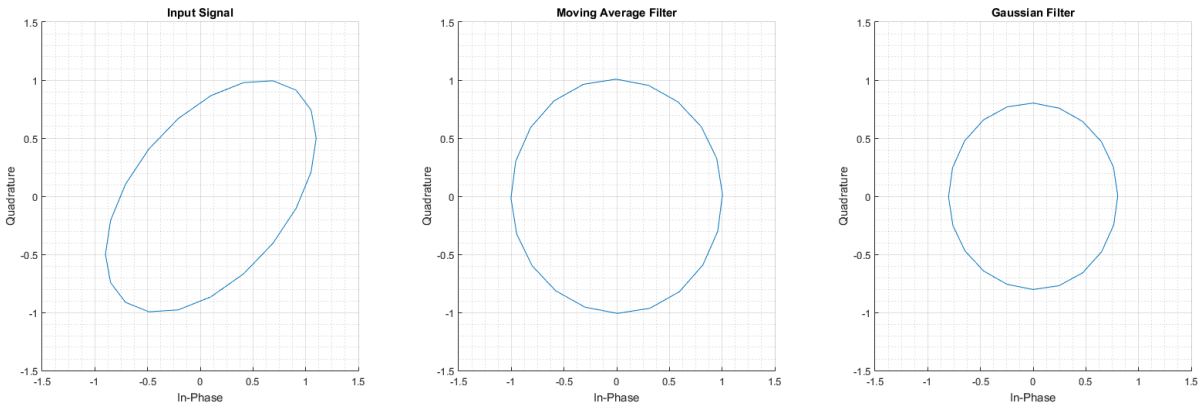

Figure 6.8: *Comparison of constellations for IQ mismatch compensation algorithms for signle tone signal.*

Scenario 2: Noisy signal

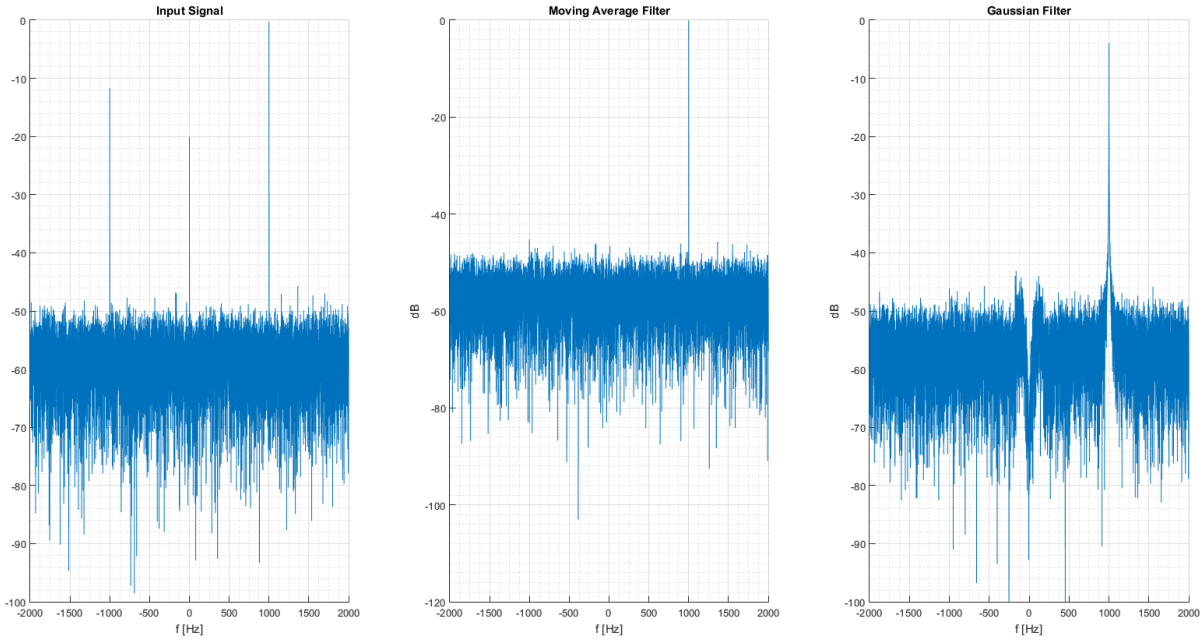

Figure 6.9: *Comparison of spectrum for IQ mismatch compensation algorithms for signle tone signal with noise.*

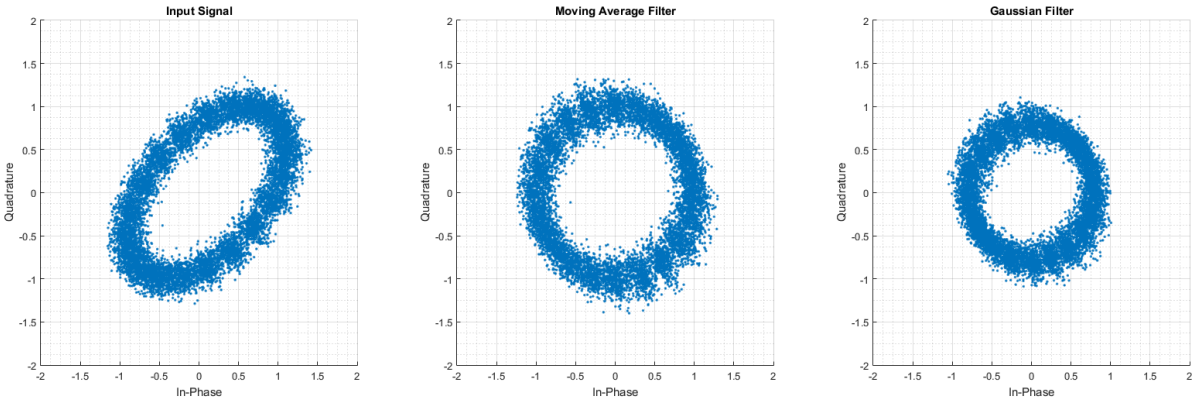

Figure 6.10: *Comparison of constellations for IQ mismatch compensation algorithms for signle tone signal with noise.*

In both cases unwanted image in $-1kHz$ and DC component were removed. Gaussian Filter introduced some spectral leakage, but compensation was still working correctly. Constellation diagram after mismatch correction is visibly shaper closer to circle than before.

Multi tone

In this case IQ mismatch compensation with two DC offset removal algorithms are compared for multi tone signal with frequencies $f1 = 1kHz$ and $f2 = 0.5kHz$ with and without additional noise.

Scenario 1: Noiseless signal

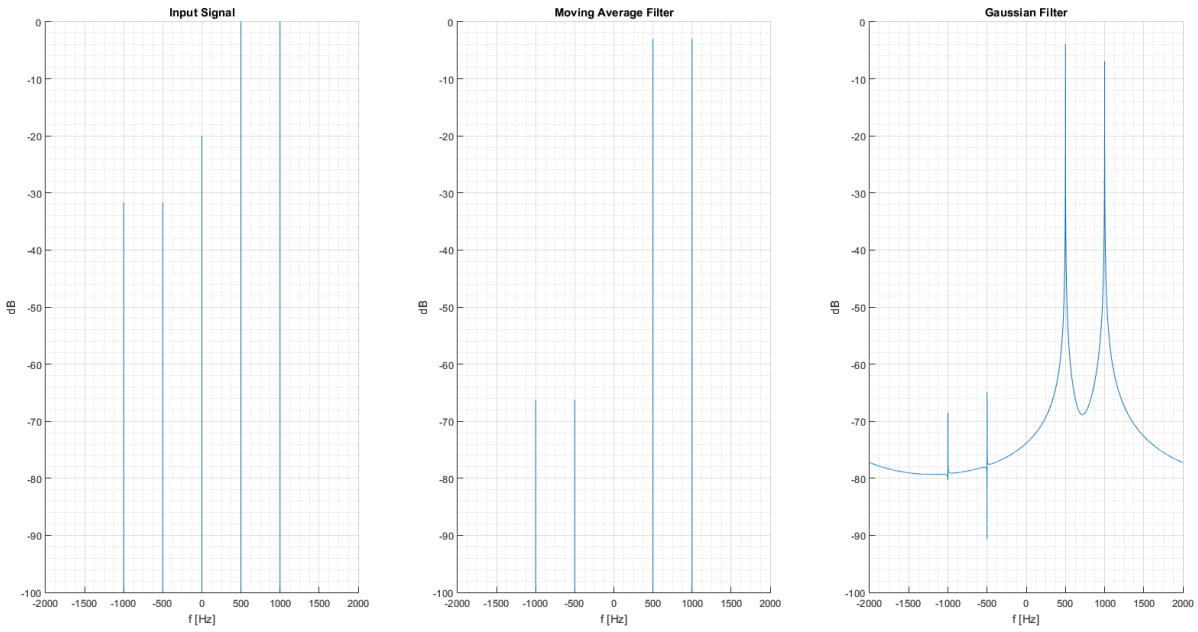

Figure 6.11: *Comparison of spectrum for IQ mismatch compensation algorithms for multi tone signal.*

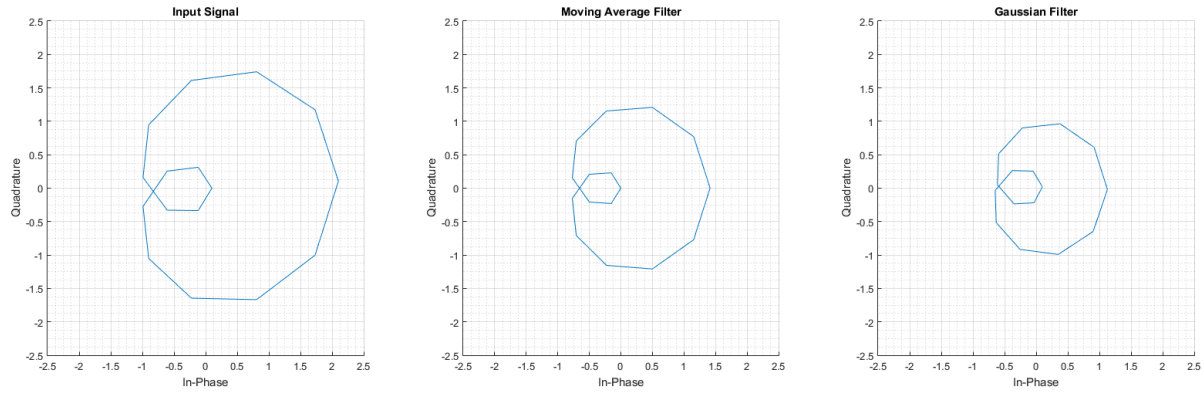

Figure 6.12: *Comparison of constellations for IQ mismatch compensation algorithms for multi tone.*

Scenario 1: Noisy signal

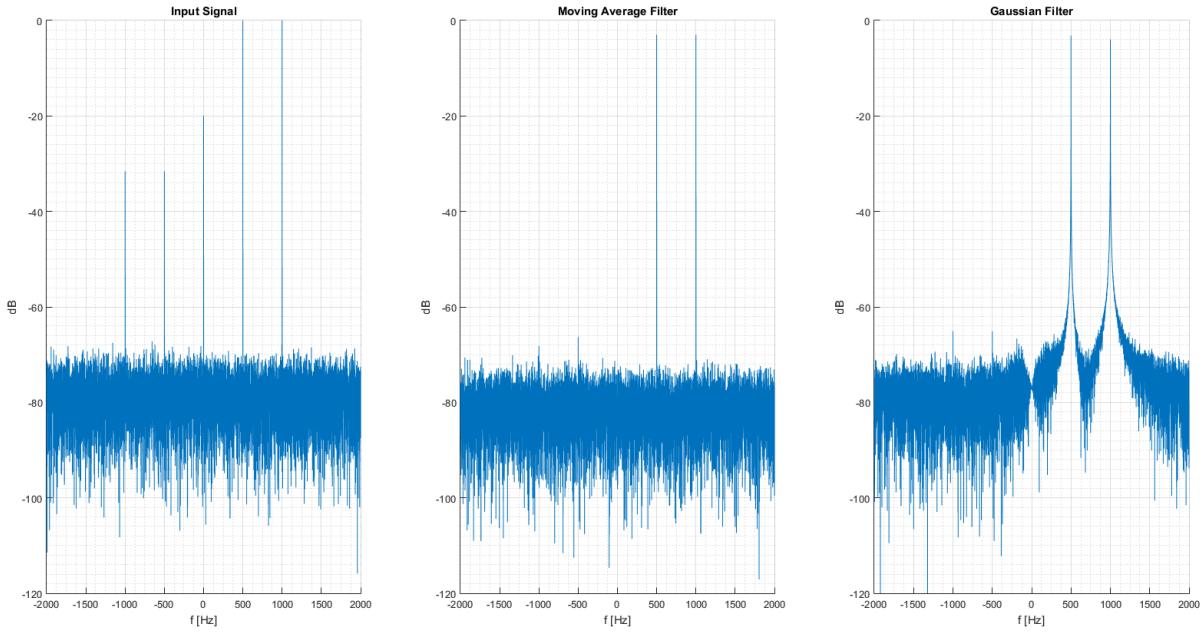

Figure 6.13: *Comparison of spectrum for IQ mismatch compensation algorithms for multi tone signal with noise.*

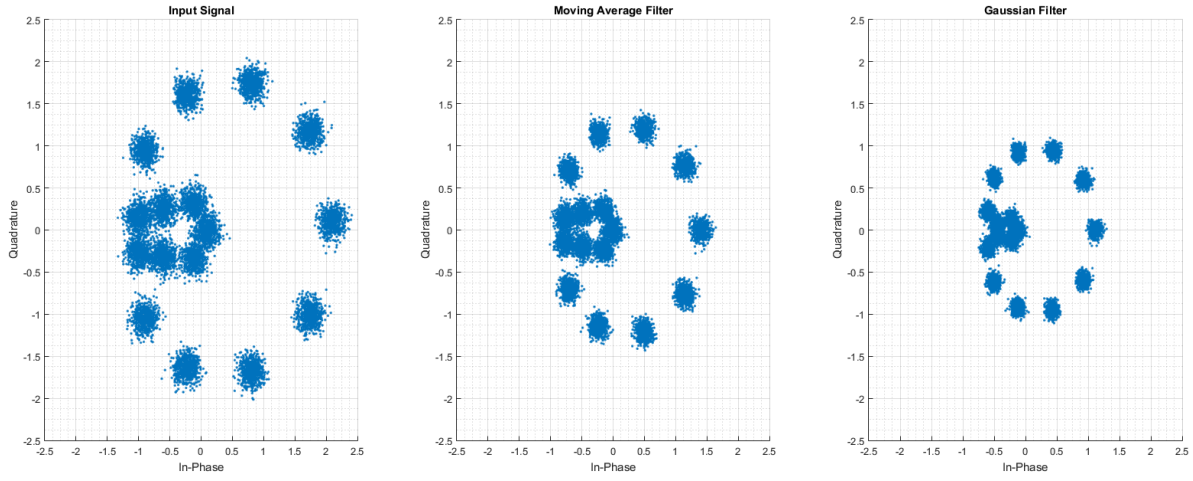

Figure 6.14: *Comparison of constellations for IQ mismatch compensation algorithms for multi tone signal with noise.*

Signal image and DC offset were successfully removed by correction algorithms. Again, Gaussian filter introduced some spectral leakage. In this case constellation diagram is composed of two circles each representing different frequency in the signal.

Broadband

In this case IQ mismatch compensation with two DC offset removal algorithms are compared for broadband signal in frequency range from $f_s = 0.5kHz$ to $f_e = 1kHz$ with and without additional noise.

Scenario 1: Noiseless signal

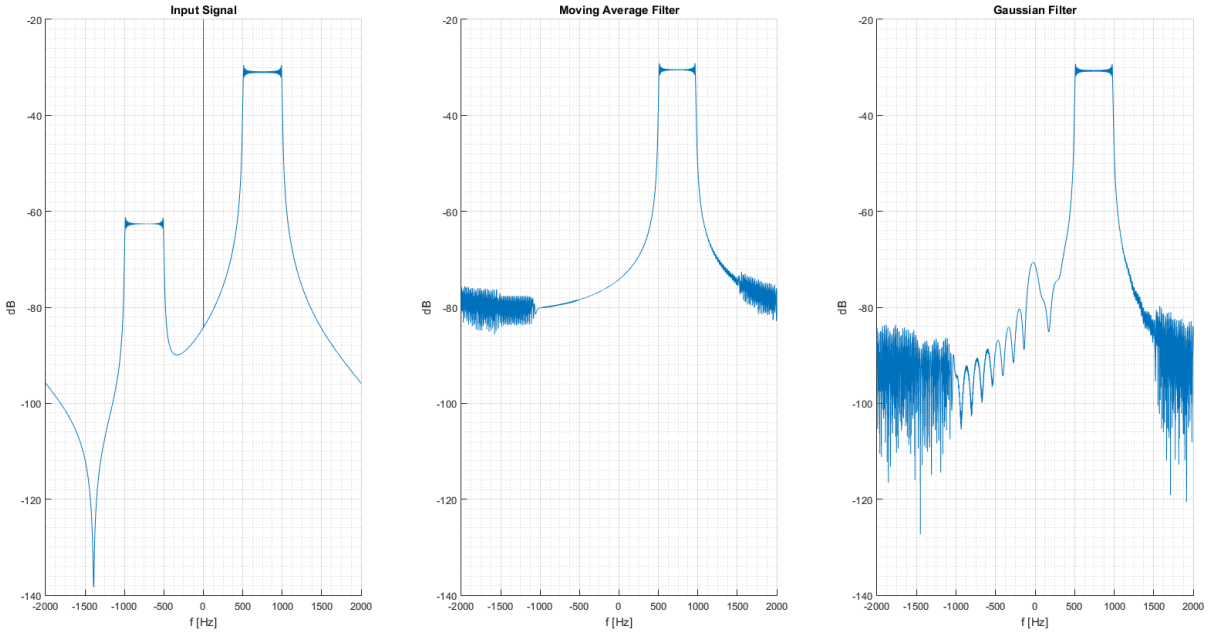

Figure 6.15: *Comparison of spectrum for IQ mismatch compensation algorithms for broadband signal.*

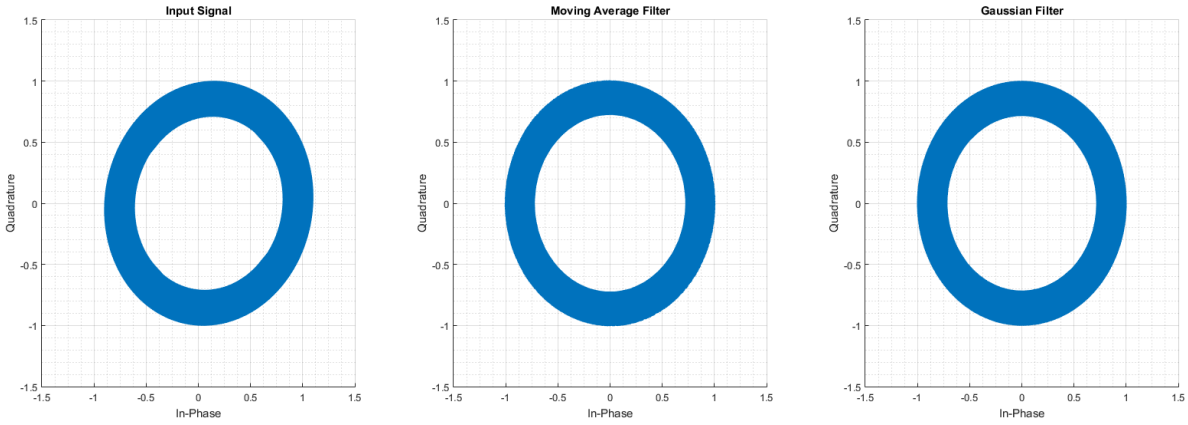

Figure 6.16: *Comparison of constellations for IQ mismatch compensation algorithms for broadband signal.*

Scenario 1: Noisy signal

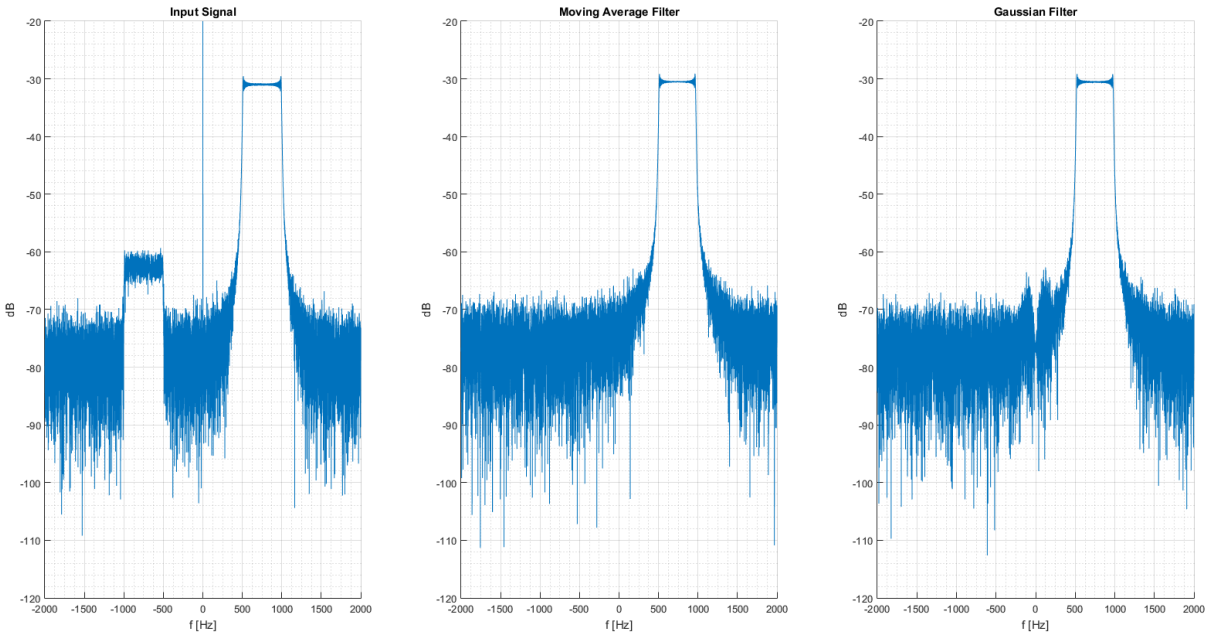

Figure 6.17: *Comparison of spectrum for IQ mismatch compensation algorithms for broadband signal with noise.*

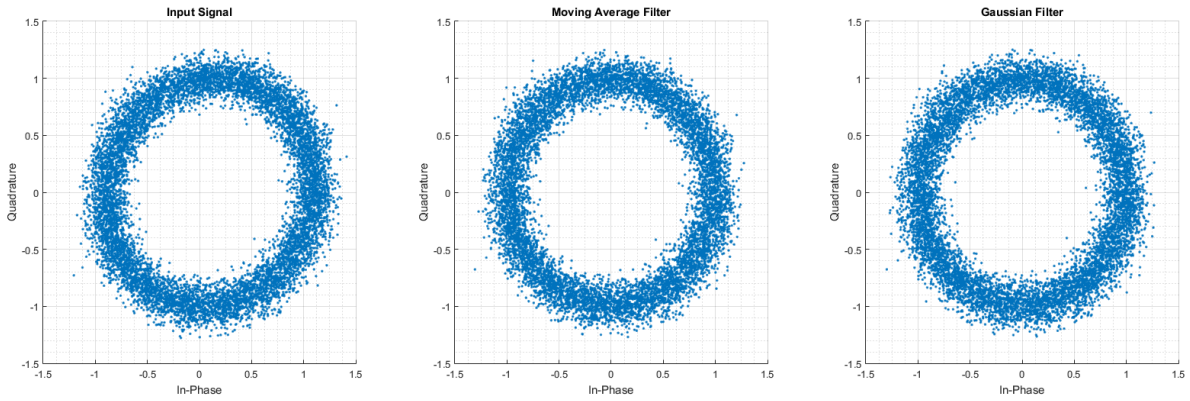

Figure 6.18: *Comparison of constellations for IQ mismatch compensation algorithms for broadband signal with noise.*

Signal image and band from $-1kHz$ to $-0.5kHz$ and DC offset were successfully removed. Constellation diagram shows improvement in signal orthogonality. In this case constellation diagram is a ring which starts at lower band limit and ends at upper band limit.

6.2 Hardware Implementation

This section contains description of all algorithms implemented in Zynq PL section and its performance evaluation. All test were performed using IIO Oscilloscope program. Transmit and receive path are configured as follows:

- sampling rate - 30.719998 MSPS,
- bandwidth - $18MHz$.,
- TX LO frequency - 2799.999998 MHz,
- RX LO frequency - 2799.999998 MHz.

and in following conditions:

- Moving Average Filter Order - 256.
- Gaussian filter $\sigma = 5$, window size = 30
- Amplitude - 1,
- DC offset in I branch - 0.1,
- Phase mismatch - 3°.

DC offset removal

In this case built in DC offset correction algorithm is tested for signle tone signal of frequency $f = 1kHz$.

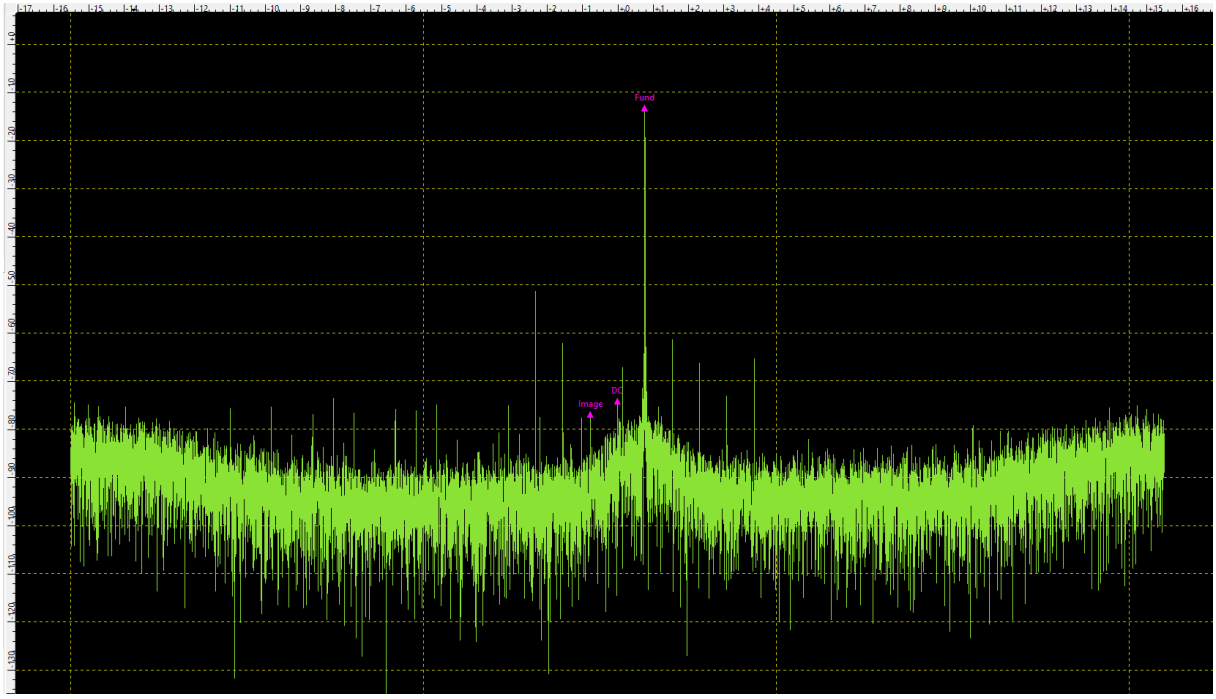

Figure 6.19: *Received signal spectrum with Moving Average Filter DC offset removal implemented in Zynq PL section.*

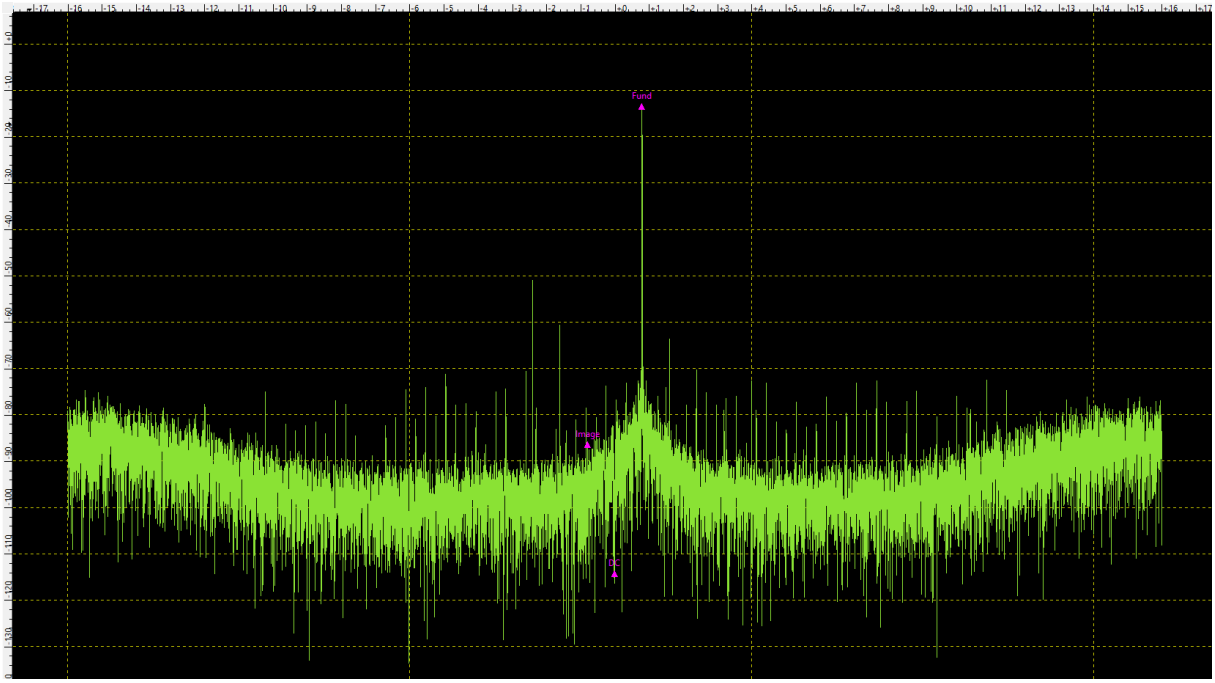

Figure 6.20: *Received signal spectrum with Gaussian Filter DC offset removal implemented in Zynq PL section.*

Single Tone Signal

In this case implemented mismatch compensation is tested for single tone signal with frequency $f = 1kHz$.

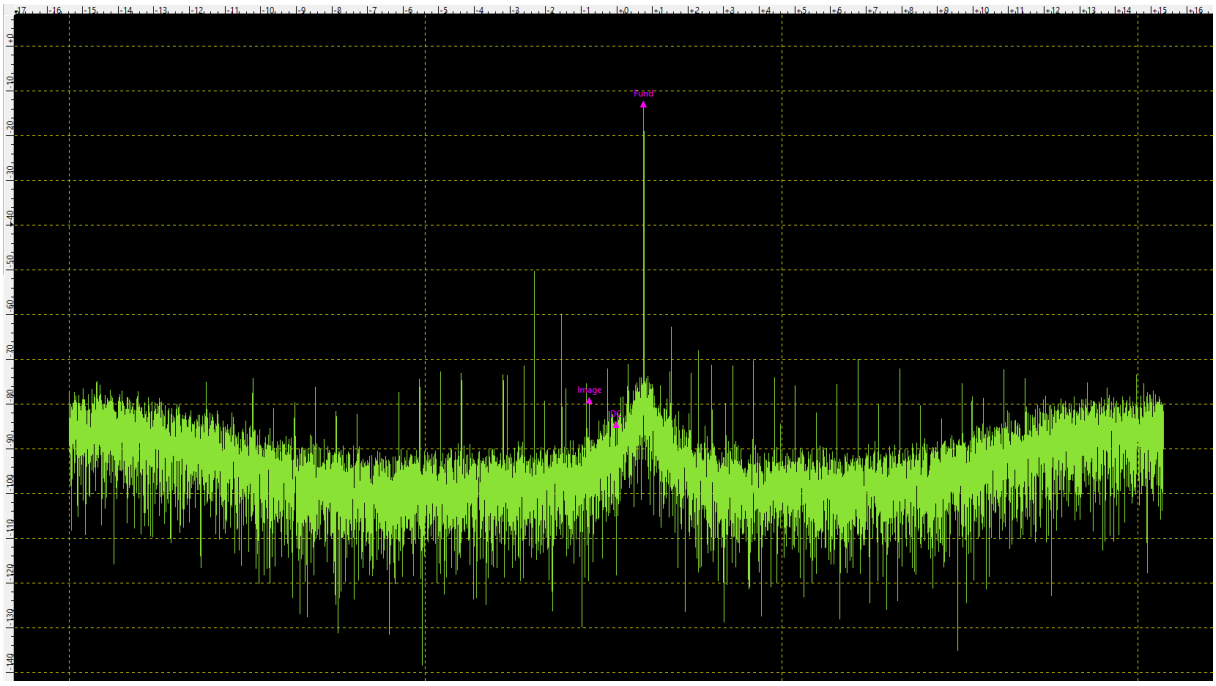

Figure 6.21: *Received single tone signal spectrum with IQ mismatch compensation with Moving Average Filter.*

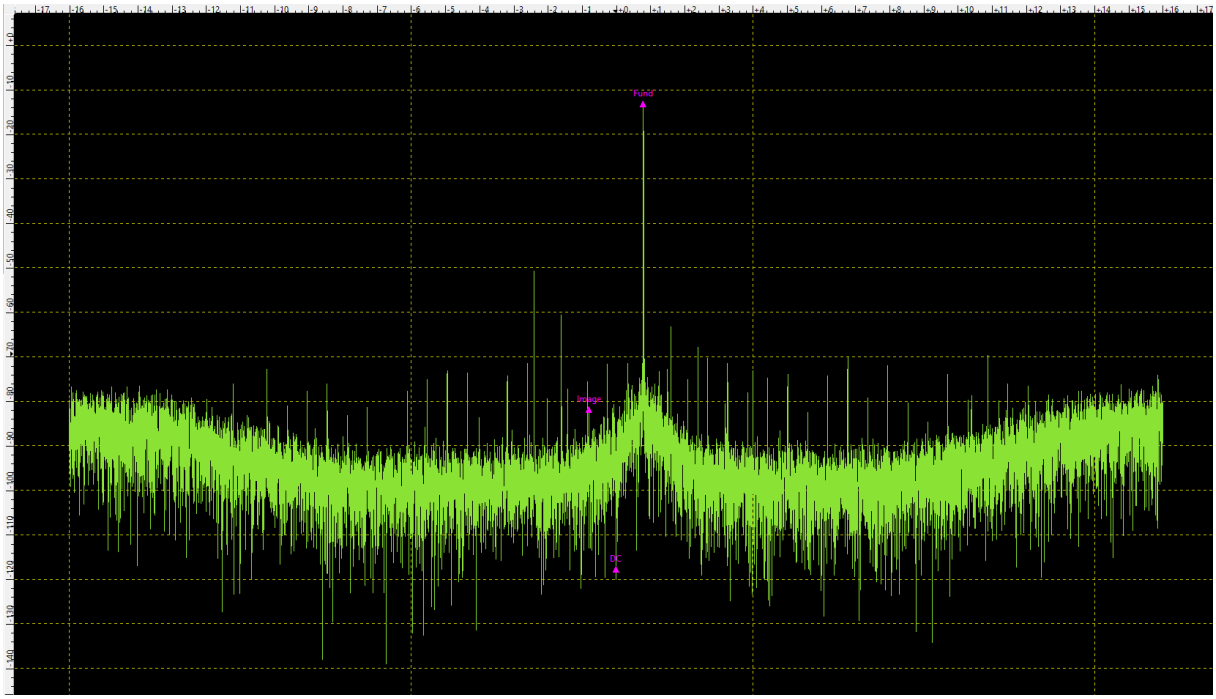

Figure 6.22: *Received broadband signal spectrum with IQ mismatch compensation with Gaussian Filter.*

Multiton Signal

In this case implemented mismatch compensation is tested for multi tone signal with frequencies $f1 = 1kHz$ and $f2 = 0.5kHz$.

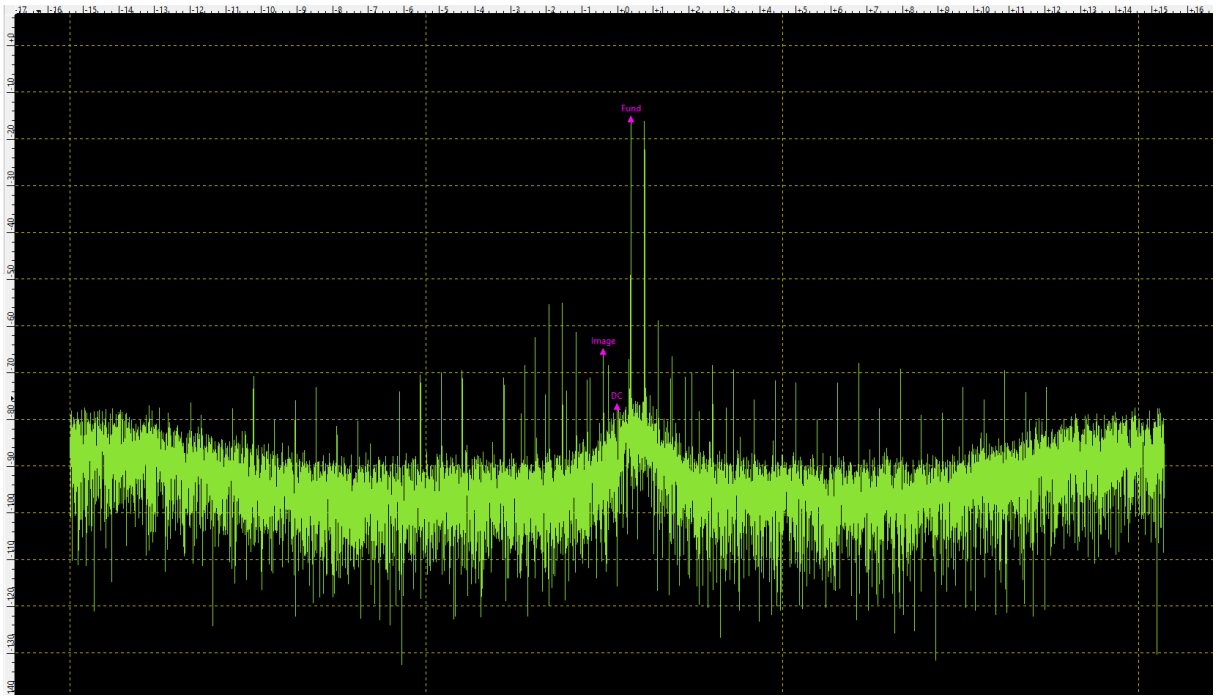

Figure 6.23: *Received multi tone signal spectrum with IQ mismatch compensation with Moving Average Filter.*

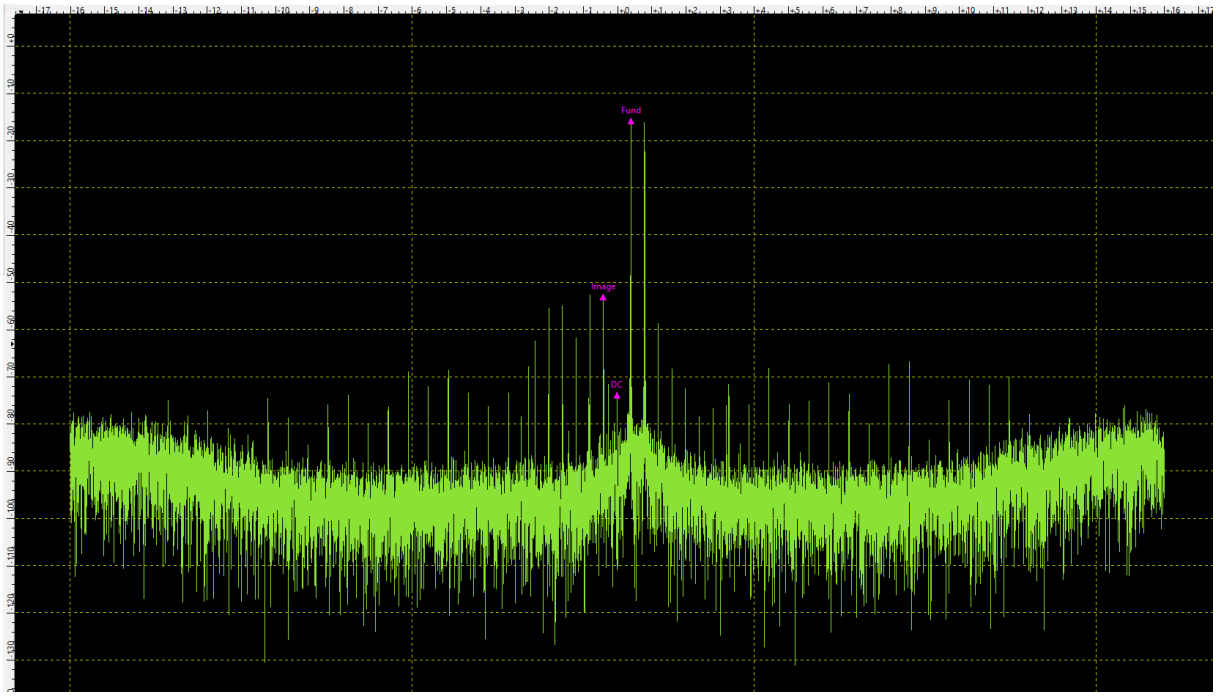

Figure 6.24: *Received broadband signal spectrum with IQ mismatch compensation with Gaussian Filter.*

Broadband Signal

In this case implemented mismatch compensation is tested for broadband signal in frequency range from $f_s = 0.5kHz$ to $f_e = 1kHz$.

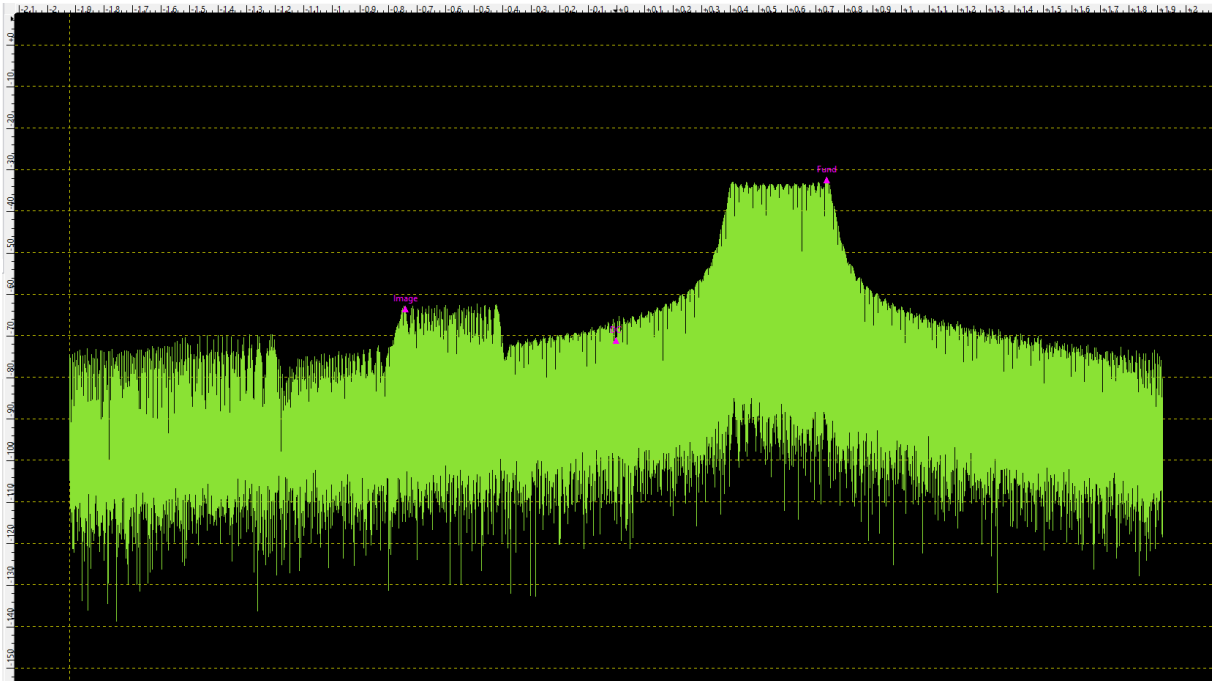

Figure 6.25: *Received broadband signal spectrum with IQ mismatch compensation with Moving Average Filter.*

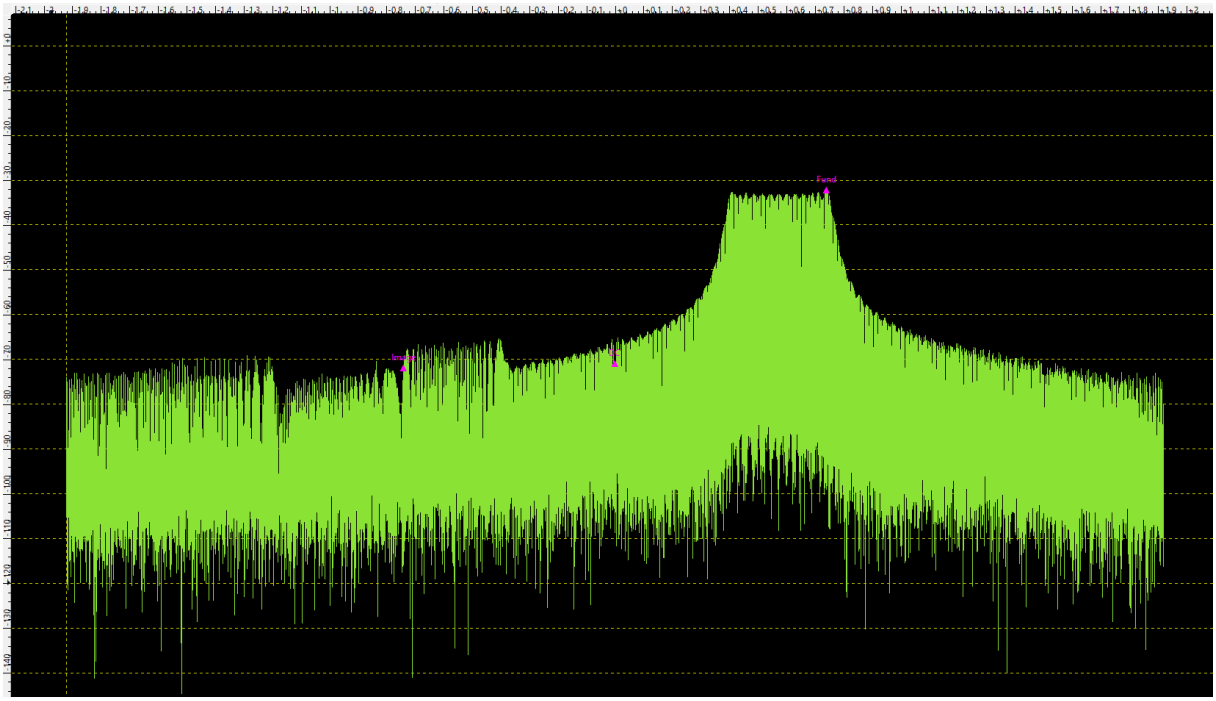

Figure 6.26: *Received broadband signal spectrum with IQ mismatch compensation with Gaussian Filter.*

In all cases DC offset was successfully removed. Gaussian filter introduce additional spectral leakage close to the tone frequencies. Algorithms perform the worst in multi tone signal case. For single tone and broad band images are successfully reduced to almost level of the noise.

6.3 On Chip Algorithm

This section contains performance evaluation of compensation algorithms built in AD9363.

All test were performed using IIO Oscilloscope program. Transmit and receive path are configured as follows:

- sampling rate - 30.719998 MSPS,
- bandwidth - $18MHz.$,
- TX LO frequency - 2799.999998 MHz,
- RX LO frequency - 2799.999998 MHz.
- Moving Average Filter Order - 256.
- Gaussian filter $\sigma = 5$, window size = 30
- Amplitude - 1,
- DC offset in I branch - 0.1,
- Phase mismatch - 3°.

DC offset removal

In this case built in DC offset correction algorithm is tested for signle tone signal of frequency $f = 1kHz$.

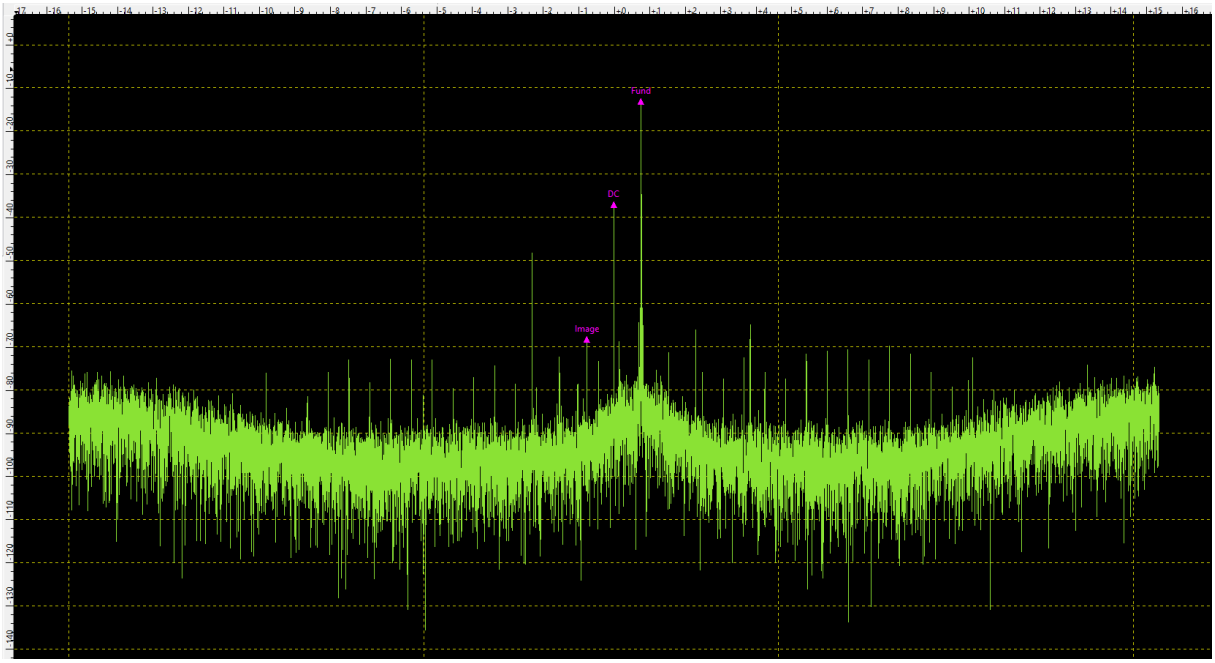

Figure 6.27: *Received signal spectrum without DC offset removal.*

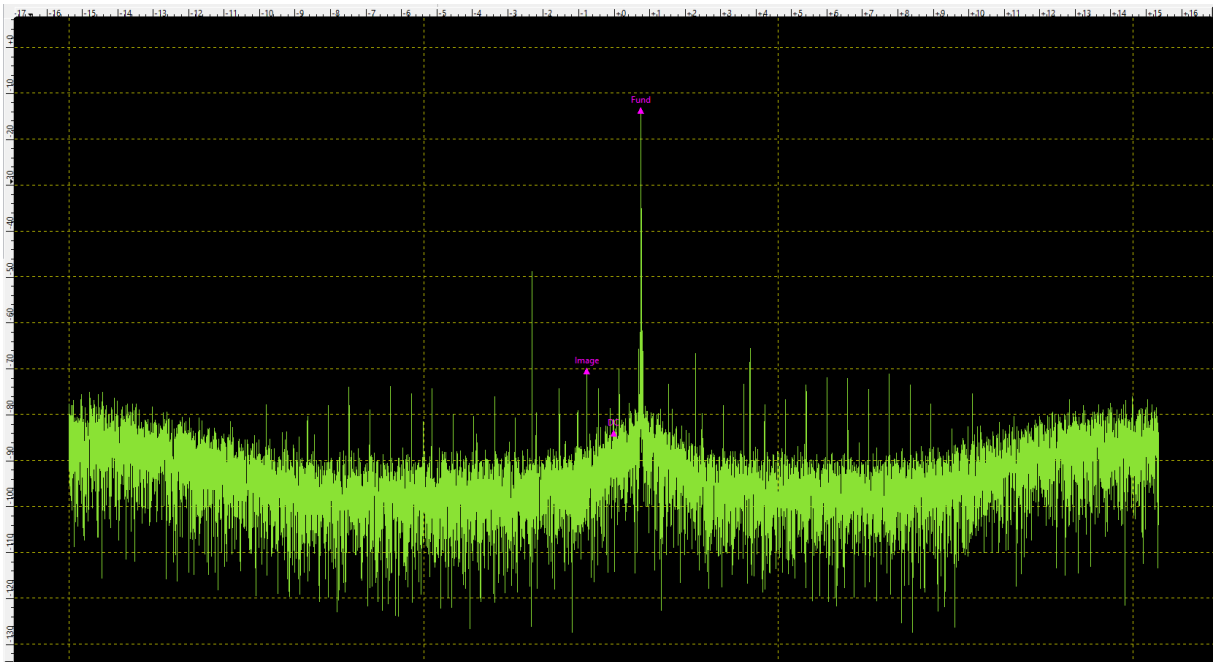

Figure 6.28: *Received signal spectrum with AD9363 built in DC offset removal.*

Single Tone Signal

In this case built in IQ mismatch compensation and DC offset removal algorithms are tested for single tone signal with frequency $f = 1kHz$.

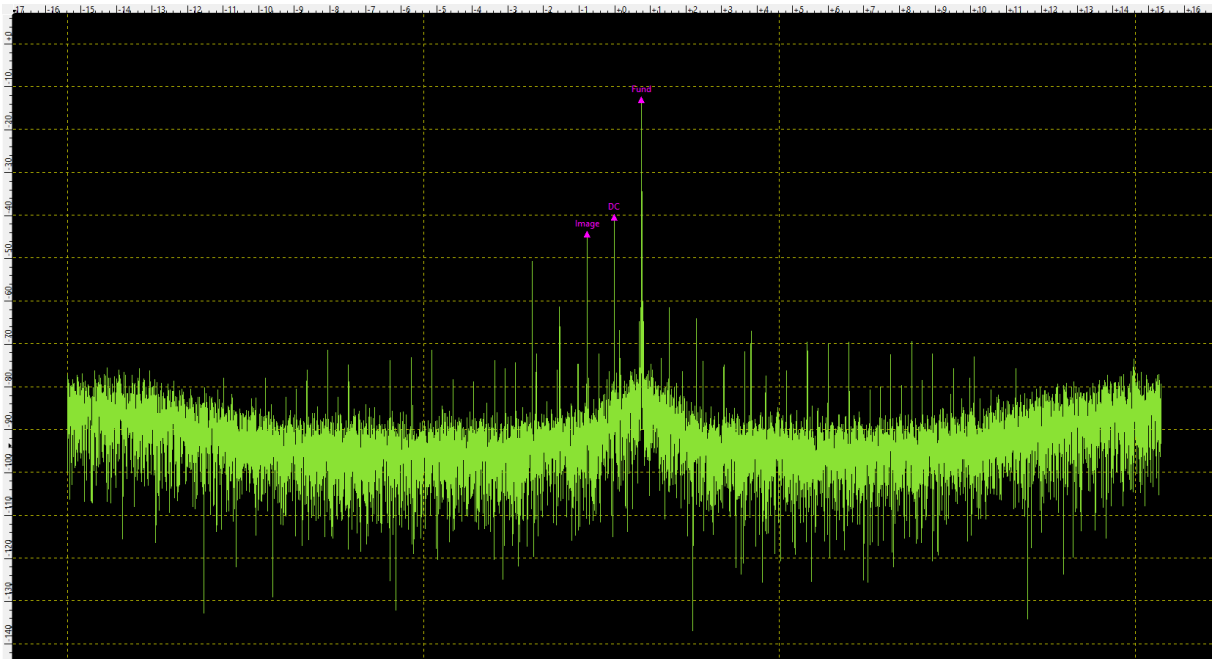

Figure 6.29: *Received single tone signal spectrum without IQ mismatch compensation.*

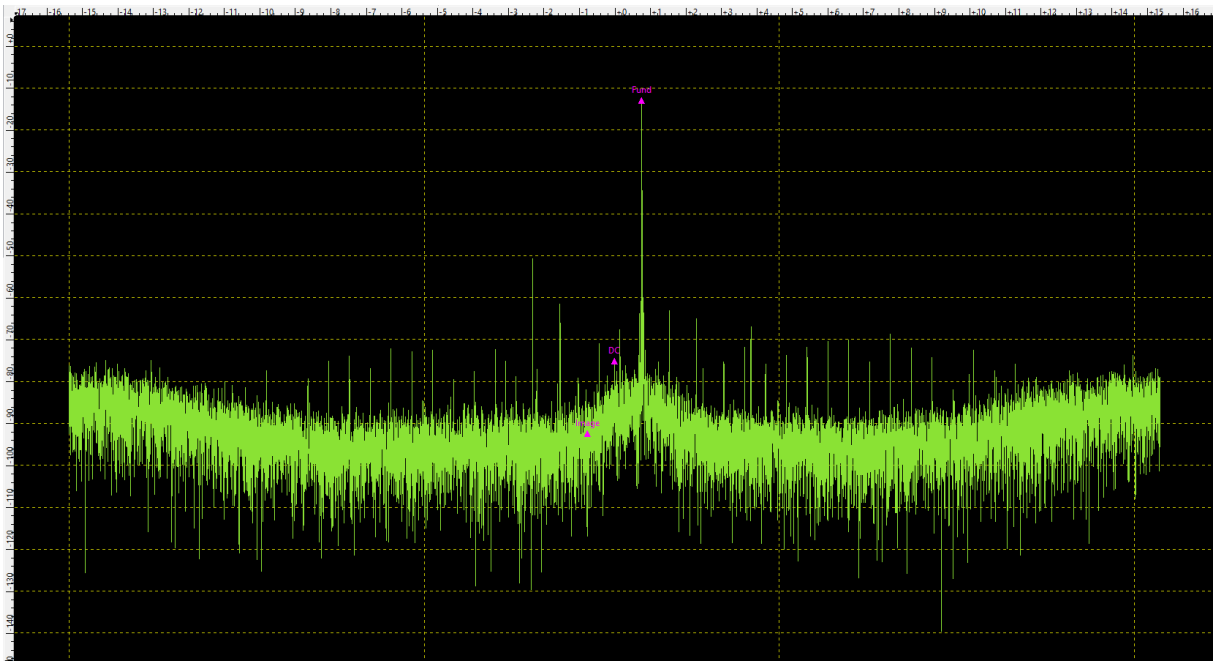

Figure 6.30: *Received single tone signal spectrum with IQ mismatch compensation.*

Multiton Signal

In this case built in IQ mismatch compensation and DC offset removal algorithms are tested for multi signal with frequencies $f1 = 1kHz$ and $f2 = 0.5kHz$.

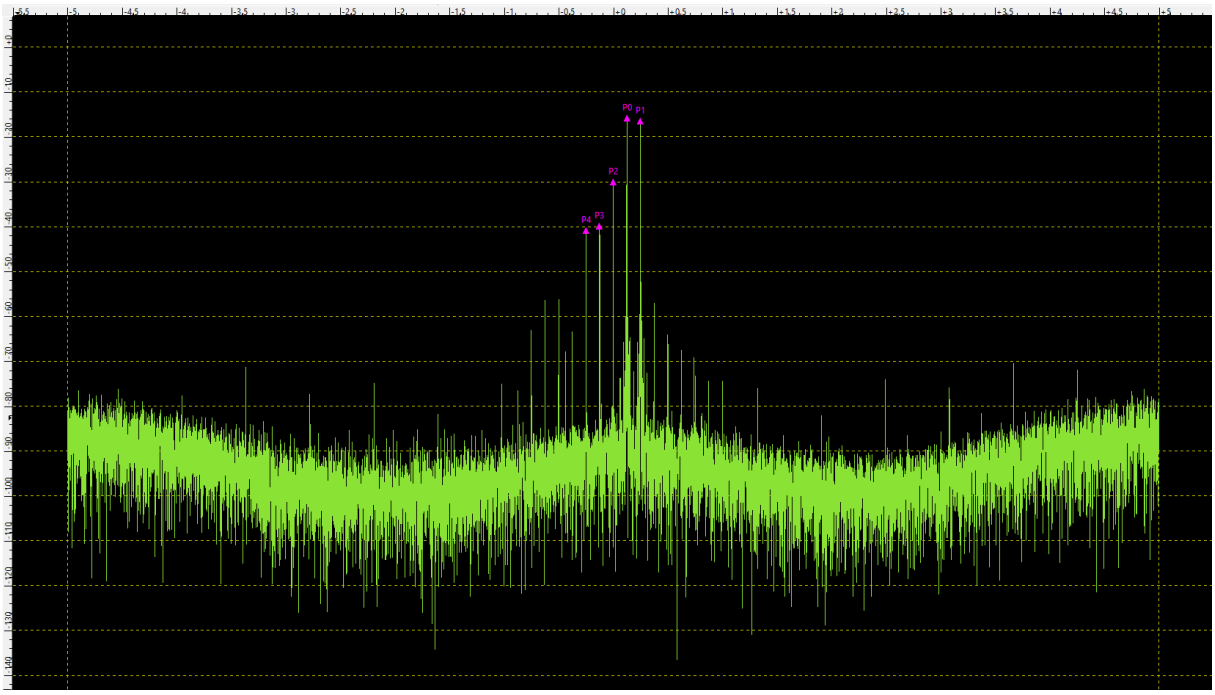

Figure 6.31: *Received multi tone signal spectrum without IQ mismatch compensation.*

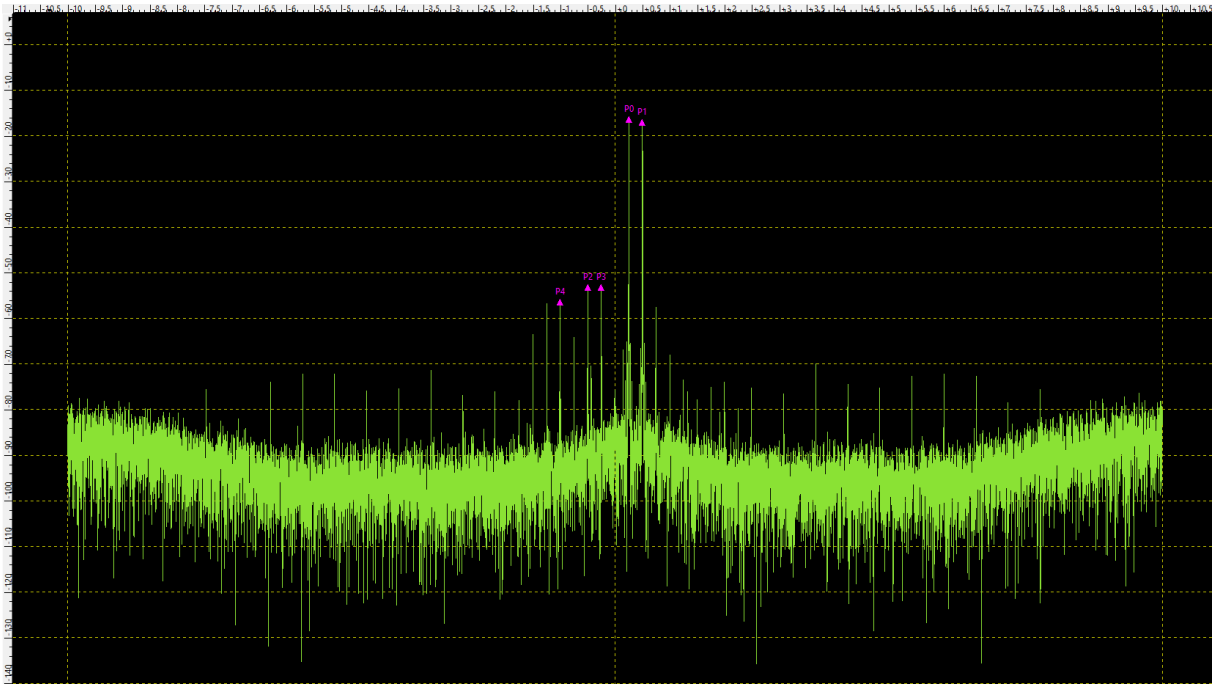

Figure 6.32: *Received multi tone signal spectrum with IQ mismatch compensation.*

Broadband Signal

In this case built in IQ mismatch compensation and DC offset removal algorithms are tested for broadband signal in frequency range from $f_s = 0.5kHz$ to $f_e = 1kHz$.

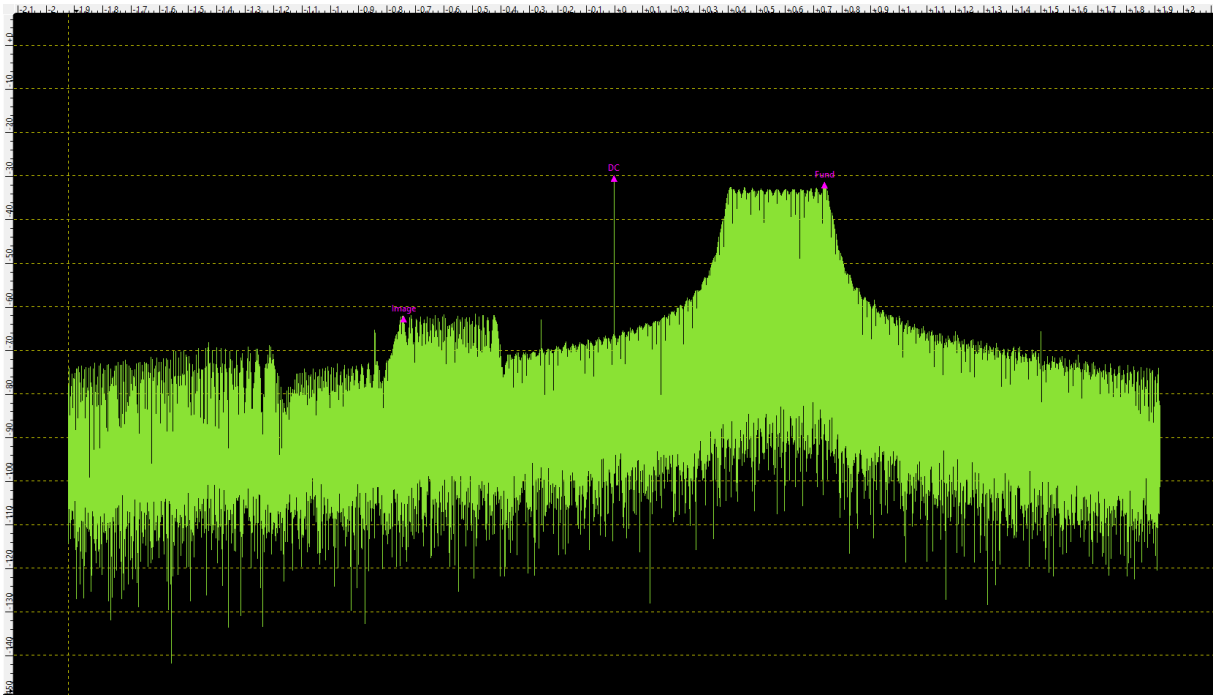

Figure 6.33: *Received broadband signal spectrum without IQ mismatch compensation.*

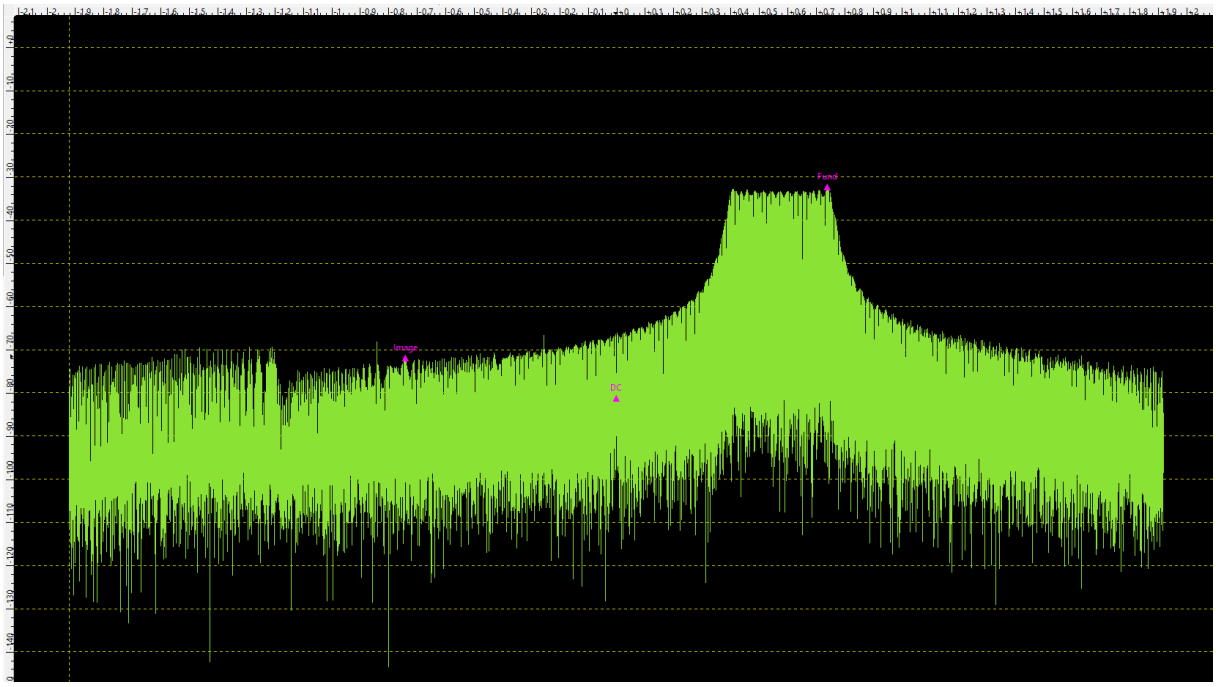

Figure 6.34: *Received broadband signal spectrum with IQ mismatch compensation.*

In all cases DC offset component was successfully removed. For multi tone singal test case markers P0 and P1 represents $1kHz$ and $0.5kHz$ tons in the signal. P3 and P4 are images of P0 and P1. P2 marker is DC component in the signal. In this case built in algorithms performance was the worst. Image fringes were reduced by only 10dB. For broadband signal image was reduced to the level of noise.

Chapter 7

Conclusions

Performance of presented algorithms was very satisfying. In all cases test cases DC component was successfully removed form the signal. Gaussian filter takes more resources than Moving Average Filter, but requires almost 9 times lower number of samples to operate (256 samples in case of Moving Average Filter and 30 for Gaussian Filter). Gaussian filter introduced additional spectral leakage close to signal tone frequency. Gaussian Filter worked better for multi tone signal than Moving Average but was worse in case of broadband signal. In all cases unwanted images were significantly reduced. Constellation diagram shapes are closer to circle which means orthogonality of the I and Q component. In case of single tone signal constellation diagram changes it shape form ellipse to circle. For multi tone two circles are present. One for lower and one for higher frequency tone. Broadband signal constellation diagram is a ring with lower bound at lowest frequency in the band ad upper bound in the highest. Introduction of the noise confirmed that algorithms are able to work in real case. However performance of both implemented and built in algorithm for multi tone signal case was the worst. Performance of algorithms implemented in Zynq PL section was better than built in AD9363 transceiver for single and multi tone, but worse for broadband signal.

Algorithms tested in this thesis shows that it is possible to significantly improve RF signal quality using both implementation approaches: in software and in hardware. However hardware approach allowed to perform effective IQ correction in real time. Devices such as SoC shows potential in field of SDR, thanks to mixed architecture of CPU and FPGA.

Bibliography

[1] R.G Lyons *„Wprowadzenie do cyfrowego przetwarzania sygnałów, WKiŁ, 1999*

[2] Li, Richard Chi Hsi *RF Circuit Design, Hoboken, NJ, USA: John Wiley Sons, Inc.*

[3] S.W. Ellingson. *Correcting I-Q Imbalance in Direct Conversion Receivers, February 10, 2003.*

[4] Lauri Anttila, Mikko Valkama, Markku Renfors *Frequency-Selective I/Q Mismatch Calibration of Wideband Direct-Conversion Transmitters, https://ieeexplore.ieee.org/abstract/document/4476448*

[5] Guanbin Xing, Manyuan Shen, Hui Liu. *Frequency Offset and I/Q Imbalance Compensation for Direct-Conversion Receivers. IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 4, NO. 2, MARCH 2005.*

[6] Hüseyin Arslan, Joseph Mitola III *Cognitive radio, software-defined radio, and adaptive wireless systems, 2007 John Wiley Sons, Ltd.*

[7] T. W. Parks, C. Sidney Burrus *Digital Signal Processing and Digital Filter Design, Wiley-Interscience; 1 edition (August 1987)*

[8] François Horlin, André Bourdoux *Digital Compensation for Analog Front-Ends. England: 2008 John Wiley Sons, Ltd*

[9] Steven W. Smith, Ph.D. *The Scientist and Engineer's Guide to Digital Signal Processing. https://www.analog.com/media/en/technical-documentation/dsp-book/dsp_book_Ch15.pdf*

[10] Louise H. Crockett Ross A. Elliot Martin A. Enderwitz Robert W. Stewart *The Zynq Book Embedded Processing with the ARM® Cortex®-A9 on the Xilinx® Zynq®-7000 All Programmable SoC, Department of Electronic and Electrical Engineering University of Strathclyde Glasgow, Scotland, UK 1st Edition*

[11] Xilinx Inc., *Vivado Design Suite User Guide - High-Level Synthesis. https://www.xilinx.com/support/documentation/sw_manuals/xilinx2017_4/ug902-vivado-high-level-synthesis.pdf*

[12] Xilinx Inc., *Xilinx Vivado High Level Synthesis: Case studies, https://digital-library.theiet.org/content/conferences/10.1049/cp.2014.0713*

[13] Xilinx Inc., *Zynq-7000 SoC Data Sheet, https://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf*

[14] Xilinx Inc., *Zynq-7000 SoC Technical Reference Manual, https://www.xilinx.com/support/documentation/user_ guides/ug585-Zynq-7000-TRM.pdff*

[15] Analog Devices Inc., *AD9363 Datasheet, https://www.analog.com/media/en/technical-documentation/data-sheets/AD9363.pdf*

[16] Xilinx Inc., *Zynq-7000 All Programmable SoC Software Developers Guide, https://www.xilinx.com/support/documentation/user_ guides/ug821-zynq-7000-swdev.pdf*

[17] *https://www.analog.com/-/media/analog/en/evaluation-board-images/images/adalm-pluto-web.gif?h=270thn=1hash=A9C5189745767A0191C4536A230C498745F14619*

[18] *https://www.analog.com/-/media/analog/en/evaluation-board-images/images/adalm-pluto_ medium_ block_ diagram.png?h=270hash=1A0512F2EC9F8DAA40A1C954A2394FE4E8BA34B3*

[19] *http://analogdevicesinc.github.io/libad9361-iio/doc/AD9361.svg*

[20] *https://www.cnx-software.com/wp-content/uploads/2012/03/xilinx_ zynq-7000_ EPP_ block_ diagram_ large.jpg*

[21] *https://wiki.analog.com/_ detail/resources/tools-software/linux-software/libiio_ diagram.png?id=resources%3Atools-software%3Alinux-software%3Alibiio*