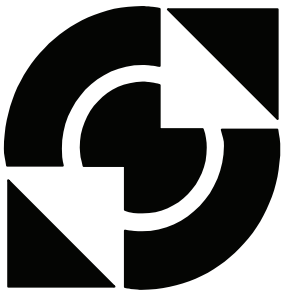
University of Twente

Faculty of Electrical Engineering,
Mathematics & Computer Science



Front-end research for a low-cost spectrum analyser

Receiver system and topology study

Kenneth C. Rovers
MSc Thesis
June 2006

Supervisors:

prof. dr. ir. B. Nauta
dr. ing. E.A.M. Klumperink
ir. F.W. Hoeksema

dr. ir. G.J.M. Smit
dr. ir. A.B.J. Kokkeler

dr. ir. J.W. Eikenbroek
ir. J. Tangenberg

Report number: 67.3163

Chair of Integrated Circuit Design
Chair of Computer Architecture
Design & Test for Embedded Systems
Faculty of Electrical Engineering,
Mathematics & Computer Science

University of Twente
P. O. Box 217
7500 AE Enschede
The Netherlands

University of Twente
Faculty of Electrical Engineering, Mathematics and Computer Science
Integrated Circuits Design group
Computer Architecture Design and Test for Embedded Systems group

Bruco B.V.
Integrated Circuits

Front-end research for a low-cost spectrum analyser

Receiver system and topology study

Kenneth C. Rovers
June 30, 2006

Submitted in partial fulfilment of the requirements
for the Master of Science degrees
Electrical Engineering & Computer Science
University of Twente

Abstract

This report discusses front-end research for a low-cost spectrum analyser. Requirement of the front-end are derived and a topology study is performed, both from an analogue as a digital perspective. Simulations are carried out to confirm the findings.

This master project was initiated by Bruco B.V., and deals with a feasibility study of a low-cost spectrum analyser front-end. Market research shows a demand for low-cost spectrum analysis for various applications, where moderate specifications compared to commercial available spectrum analysers can be acceptable. The aim is to design a spectrum analyser system with a maximum degree of integration, since this provides large cost-savings and a competitive product.

The project is divided into two parts; the analysis and design of an analogue spectrum analyser front-end, and the analysis from a digital perspective.

For the analogue part, typical spectum analyser specifications are examined and relations and trade-offs between them are derived. This results in a balanced set of requirements. Receiver system architectures are explored and a spectrum analyser front-end design is presented. The design consists of a frequency conversion, frequency selection and a measurement sub-block. These are further discussed and their effect on the achievable performance is estimated.

For the digital part, the advantages and disadvantages of the use of digital processing for spectrum analysis are researched. A design is proposed that combines the best of both worlds; the performance of the analogue front-end and the flexibility of digital signal processing. The design is simulated, using MATLAB for the analogue part, while using a specialised reconfigurable architecture for the digital part; the Montium Tile Processor.

Version history

Version 0.1 – 04/02/2005	First layout of contents
Version 0.2 – 14/04/2005	Digital perspective layout
Version 0.3 – 24/07/2005	Updated template and introduction
Version 0.4 – 25/09/2005	Spectrum analyser
Version 0.5 – 09/11/2005	Architecture and review of chapter 1 and 2
Version 0.6 – 01/03/2006	Phase noise and quadrature addition
Version 0.7 – 02/04/2006	Filters, review and division of chapter 3
Version 0.8 – 12/06/2006	Measurement, digital considerations and simulation
Version 0.9 – 26/06/2006	Finished all chapters
Version 1.0 – 30/06/2006	Final review

Thesis committee

prof. dr. ir. B. Nauta
dr. ing. E.A.M. Klumperink
ir. F.W. Hoeksema

dr. ir. G.J.M. Smit
dr. ir. A.B.J. Kokkeler

dr. ir. J.W. Eikenbroek
ir. J. Tangenberg

Acknowledgements

ing. P. Klatser
ir. P.T. Wolkotte

Preface

This project was performed as my master thesis project for the studies electrical engineering and computer science. A background in both electrical engineering and computer science enables an approach from two disciplines to such a diverse subject as a spectrum analyser.

In many ways, the spectrum analyser front-end can be seen as a general radio front-end that needs to support all frequencies in all situations with a pre-determined performance. This limits the use of filtering and the exploitation of signal characteristics and requires a “pure” approach to common problems of receiver systems.

First of all, I would like to thank Bruco B.V. for initiating this project. Jan-Wim Eikenbroek, Jurjen Tangenberg and Paul Klatser have been interested, patient and valuable contributors to this work.

I much appreciate the enthusiasm, ideas and discussions by all of the thesis committee. I especially would like to thank Eric Klumperink for his day-to-day guidance, motivation and willingness to critically review my work. I would also like to thank Gerard Smit for his support, for providing direction and his inspiration. Fokke Hoeksema is much appreciated for his drive and help in explaining the mathematical concepts. I am also thankful to Andre Kokkeler for picking out the many typos, but more so for his constructive criticism and identifying important concepts.

As some of you know, this report has been a challenge; therefore I would like to address this final word to the reader of this work. I wish you success in reading this comprehensive report and hope you will find what you are looking for, in this report as well as in life.

Kenneth C. Rovers
Enschede, June 2006

Contents

Chapter 1

1 Introduction

This chapter will introduce the subject of this thesis, starting with a background on the use of spectrum analysers[1] and available products. This background section can be skipped for readers already knowledgeable about the subject. We continue with an explanation of the original assignment provided by Bruco B.V. [14]. We then identify a set of goals for this project to achieve. After determining our objectives, an approach to achieve these goals is provided, which dictates the structure of this report.

1.1 Background

A spectrum analyser (Figure 1) is a device to measure the spectral content of a signal. A spectrum is the representation of a signal in the frequency domain. The spectral content is the frequency, the amplitude (for a spectrum analyser often as a power spectrum, see section 2.1) and the phase of the signal. Spectrum analysers are highly sophisticated measuring devices [7] useful in many areas. One can for example examine the frequency contents of an audio signal, examine a clock frequency generation circuit or examine signals from space for signs of other intelligent life. However, its main area of use is probably for analysing communication systems.

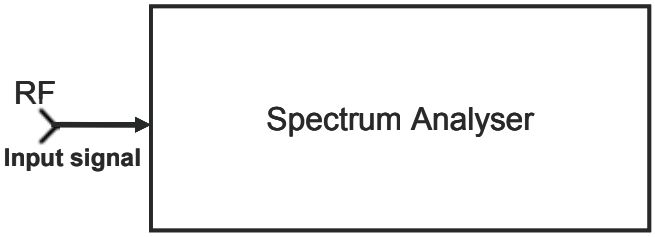


Figure 1. Spectrum analyser with input signal

1.1.1 Communications

Humans are social beings with the need to communicate to share knowledge and experiences. Interpersonal communication comes in many forms and is the subject in a number of disciplines. In order to overcome larger distances than the ear or eye can handle, new methods of communicating were developed, such as the jungle bongs or smoke signals.

Tele-communications is the field of communications over a distance. The elements of a telecommunication system are a transmitter, a medium (and possibly a channel) and a receiver and can be point to point, point to multi-point or broadcast. An early form of telecommunications is the telegraph system and later the telephone.

A main disadvantage of the former two is the reliance on a wired connection between the communication points. A solution to this is found in the form of radio communication in which the communication signal is sent through the air. However, while there can easily be many wires to enable multiple communications at the same time, there is only one atmosphere and it is not practical to isolate parts of the atmosphere. Thus, the communications must be either performed after each other (i.e. time-division), or at different frequencies (i.e. frequency division)[2]. And here we arrive at the need to analyse a spectrum. [13]

[1] As a first note, it is mentioned that the British English spelling of analyser is chosen throughout this report, as the 's' provides a softer appearance hinting at the more "abstract" (as in system level) nature of this project.

[2] Or with orthogonal signals (i.e. code-division). The analogy of using a different language for each group of a conversing crowd is often used [4][13].

1.1.2 Wireless communication systems

Over time, wireless communications has evolved from AM radio to mobile phones and on. However, all are characterised by the limited availability of frequency ranges or channels to use. First, atmospheric absorption and distortion determines which frequency ranges can be used and second technology and practical constraints dictate the useable top frequency and required size of a channel. In the beginning there were only a few radio channels possible, while nowadays millions of people use their mobile phone daily.

From this elaboration, we find that in order to make use of the available frequencies, we must be able to transform the original signal to another frequency, which can be used to transmit, and afterwards the received signal must be translated back to get the original signal. This is called modulation and demodulation.

Thus, we can identify many situations in which we would like to know the frequency contents of a signal. For example to check the frequency components and bandwidth of the original signal, then of the modulated signal, next of the received signal and finally of the demodulated signal to check if the wireless communication system is operating properly. But also to check if a channel is free or check if the device is not generating distortions, since these typically become apparent because of unexpected or larger than expected frequency components. More advanced measurements include checking the type and or performance of a modulations scheme. [7][13]

1.1.3 Available spectrum analysers

At the high-end spectrum analysers, with state-of-the-art specifications and a variety of option packs, can be found from well-known brands such as Agilent (formerly H.P.) [15] or Rohde & Schwarz [16]. They range €10.000 ~ €100.000 or more, almost always differentiated by input range from up to 3 GHz to up to 50 GHz. Each can have a selection of options for advanced measurements in the field of for example GSM, CDMA or WLAN or front-end modules for higher input frequencies. Older generations with lesser specification can be found a bit cheaper, but not much. [15][16]

In mid-range we can find a few spectrum analysers up to 1GHz with adequate performance from €1000 ~ €5000 from Hameg [18], which is a Rohde & Schwarz company. However, only three (9kHz, 120kHz and 1MHz) resolution bandwidths can be selected, which is not very impressive compared to the 1Hz~10MHz bandwidths with at least 7 steps of the high-end devices. [22][24][26].

On the low end there are many amateur projects available on the internet. Most use an audio input of the soundcard to digitise the signal, and use software to convert the time-based signal to a frequency-based one with the Fourier transform (see section 2.1.3) as well as other analysing. Normally a small front-end box is used to enable a wider range of frequencies. The target price is usually in the range of €100. [19][20] It is possible to perform spectrum analysis at such a low price by using of-the-shelf components mostly for televisions and a simple architecture. The result is of course limited specifications, mostly in the selectivity and resolution. Some more well-know examples are the “Poor Man’s Spectrum Analyser” [19] and the “Simple Man’s Spectrum Analyser” [20].

A little bit more expensive and an actual commercial product are the spectrum analysers from Pico [21]. From €100 ~ €1000, they are basically analogue to digital converters with pc interfacing and software performing the spectrum analyser functions. Still the specifications are not very impressive, with low resolution (8 bit), especially low frequencies (up to 200Mhz) and low dynamic range (up to 96dB but only at 166kHz, for high frequencies 50dB). [21]

In general the selection of products in the mid-range and low-end segments is very limited, while at the low end the performance is normally only good enough for amateurs. All spectrum analysers under €10.000 are not able to perform measurements related to recent technologies, such as GSM, Bluetooth or WLAN. Overall, we can see that there is limited choice in the mid-range and no choice in the low end of devices with a performance anywhere near that of high-end devices.

1.1.4 Application

Information content at a certain frequency is not always needed and one might suffice with just an indication of the signal strength. For example, one might only be interested in which frequencies are occupied and which channels are free. A government official checking if a device is transmitting at frequency ranges allowed by regulation, or if a pirate radio channel is broadcasting. Also it is possible that you only want to check if the frequency is correct and as expected.

On another scale, that of a spectrum analyser as part of a device, it can be beneficial and is sometimes needed to scan a certain range of frequencies for free channels, so as to enable channel hopping. It is also beneficial if it is possible to examine signals on circuit to check if they behave as expected. Thus it might be feasible to implement a small and low cost spectrum analyser on chip to examine the signal spectrum at different parts of the chip and make adjustments.

From above it is clear that a lot of opportunities exist for a highly integrated low-cost spectrum analyser with good, but maybe not state-of-the-art specifications. Many possible products and solutions can be thought of; such as a handheld spectrum analyser for use in the field, a low-cost spectrum analyser for hobbyists or companies with smaller budgets and requirements, an easy to integrate spectrum analyser test-module and many more.

It is the objective of this report to provide insight in the feasibility, possibilities and specifications possible in this venture.

1.2 Assignment

As a learning vehicle, Bruco created a low-cost spectrum analyser project. The original project description is provided in appendix A.

The proposed spectrum analyser has to be low-cost with rather modest requirements compared to state-of-the-art analysers. It must be designed with an as high degree of integration as possible, since this provides for large cost-savings and therefore a competitive product.

Market research performed by Bruco shows a demand for low-cost spectrum analysis for various applications. The intended applications include modern wireless communication standards and low-end spectrum analysers do not provide adequate performance for this domain. On the other hand, one can imagine various situations where state-of-the-art performance is not required, while except for niche markets such as spectrum scanners, there are no real low-cost intermediate (mid-range) solutions available.

The main aim of this project is to acquire knowledge of RF systems and to explore new fields of radio applications. The project consists of two parts: 1) analogue RF signal processing; 2) digital signal processing.

The aim of the analogue part is to explore spectrum analyser RF architectures:

- Analyse typical spectrum analyser specifications and derive relations and trade-offs between them, resulting in a balanced set of requirements.
- Explore spectrum analyser front-end architectures and estimate the achievable performance

The aims of the digital signal processing part are:

- Explore the advantages and disadvantages of the use of digital processing for spectrum analysis. Propose an architecture for a spectrum analyser system with digital signal processing.
- Simulate the front-end architecture, using MATLAB for the analogue part, while using a specialised reconfigurable architecture for the digital part; the Montium Tile Processor. This results in a clear overview of the possibilities and limitations of the application of the reconfigurable architecture in the spectrum analyser design.

1.2.1 Goals

With the help of the provided background information and the assignment, a set of goals can be developed to have a reference to work against. We will try to draw conclusions set out against the goals we intent to achieve in this report. The following goals are identified and taken into account during the progress of the project:

- Provide the knowledge needed to assess different spectrum analyser designs for radio frequencies
- Create a balanced, feasible and coherent set of requirements (specifications) with their relations and the trade-offs between them
- Analyse and select the best front-end topology and derive subpart requirements
- Provide further analysis of the subparts especially of the down-conversion architecture with an emphasis on image rejection
- Perform simulations to confirm findings
- Deliver a complete and well-balanced low-cost spectrum analyser front-end design with clear limitations and constraints.

1.2.2 Boundaries

A spectrum analyser is a device with many advanced aspects to be taken into account. Since this project is intended as a final year master's thesis assignment, some boundaries must be set of what can be part of the produced results. Note that information from chapter 2 might be needed to understand this section.

As this is a learning project, the main aim is to perform a system level topology study with corresponding simulations and derive relations and trade-offs. This means architectures and solutions will be provided and simulated on a block level. Some aspects will be further analysed to provide additional relevant information, but this will not include solutions at the transistor level.

The developed architecture will only include the front-end of the spectrum analyser. This is the logical sub-part to start with (going from what we have to what we want) working in a top-down fashion. This means that after determining the specifications and their relations at the top level, we will analyse the front-end with respect to those specifications. The front-end will start at the input of the spectrum analyser and will go up to the output of the frequency selection part. It will also take into account some of the measurement parts, which are normally performed by an analogue to digital converter (ADC). For the digital signal processing, the ADC is moved forward and the frequency selection is performed in the digital domain.

The first part of the spectrum analyser input is generally a low-noise amplifier (LNA). This part of the front-end will be taken into account with respect to the specification it must adhere to and the performance it must have. We base the achievable specifications on typical results for previous designs. The LNA is not further analysed because of time-constraints.

The same can be said for the local oscillator (LO) responsible for producing a range of frequencies. However, as it is a fundamental part of the front-end architecture, the LO requirements will be further analysed than the LNA's. Thus, we will try to shed light on the feasibility of the LO requirements but not go into the possibilities to achieve them.

The chip-technology is only sporadically included as a factor into the design. This is chosen because the chip technology is a relatively low level influence on the design, while the approach is top-down. Most importantly is the integratebility of each part of the design as this is a large influence on the final-cost and the main aim of this project is to be low-cost. This is dependant on the chip-technology and normally the CMOS process is assumed.

Finally, the performed simulations are not exhaustive with respect to each introduced factor of the design, again because of time constraints. Instead the simulations focus on the operation principle and the use of a reconfigurable architecture to perform the digital signal processing.

1.3 Approach

As evident from above this project is performed in a top-down approach. This means, that we can subdivide the above goals or objectives into a logical sequence of sub-objectives.

First we must understand what a spectrum analyser is, what it measures and how this can be done. Thus we have to:

- Determine what a spectrum analyser is and what it must measure
- Determine the different possibilities (operation principles) to achieve these measurements
- Select the operation principles applicable

Next, we must understand how we can quantify the quality and performance of its measurements. These form the specifications so we can compare different spectrum analysers. Thus we have to:

- Identify the specifications to assess spectrum analysers by
- Derive relations and trade-offs between the specifications
- Compare specifications against available spectrum analysers
- Develop a balanced, feasible and coherent set of requirements (specifications)

Then the spectrum analyser front-end is sub-divided into major sub-blocks. Thus we have to:

- Define the major sub-parts that make up the spectrum analyser front-end
- Derive the consequences of dividing the specifications over the sub-parts
- Determine the relevant specifications and trade-offs for each sub-part
- Analyse the front-end topology and derive subpart requirements

The next step is the analysis of the frequency conversion, which is the first major sub-block. Different architectures are analysed and compared, first from a conceptual viewpoint and then going into more detail:

- Determine the different architectures (topologies) possible to implement the frequency conversion.
- Derive relations and trade-offs between the different architectures and the set of requirements
- Select the best architecture according to the relations and trade-offs found
- Further analyse specific problems and provide possible solutions

The following subpart is the frequency selection performed by filters. The different filters needed in the selected architecture are treated:

- Analyse how to transform spectrum analyser specification into filter design criteria
- Determine the filter design criteria for each needed filter
- Analyse possible filter solutions with respect to the achievable filter design criteria
- Select the best filter solution
- Analyse different filter implementations
- Select best filter implementation

The final sub-block is the measurement block. This includes a discussion of the best location for the ADC, both for analogue signal processing as for digital signal processing.

- For the analogue case, analyse compressing the input amplitude range to a logarithmic scale or a more manageable sub-range.
- Discuss an all analogue solution
- Determine the ADC requirements for digitalising the signal at different points after the frequency selection
- For the digital case, select the most flexible but achievable ADC location
- Determine the ADC requirements achievable at that location

After discussing the sub-blocks for the analogue case and selecting an ADC location for the digital case, the front-end is analysed from a digital perspective:

- Define two operating principles, one based on the analogue design, the other based on the FFT
- Discuss advantages and disadvantages of a digital design
- Analyse the implications for the frequency conversion
- Determine the limitations and consequences for the spectrum analyser specifications of using an ADC and discrete digital signals.
- Provide a comprehensive treatment of the different frequency selection options.

Finally, the selected sub-part architectures are combined and simulated.

- Confirm operation by simulating the selected digital architecture in the ideal case without distortions
- Analyse the design with a single input and a real-life input
- Show and discuss the limitations of digital signal processing
- Use a reconfigurable architecture to perform the digital signal processing
- Show and discuss the limitations of using the reconfigurable architecture

A well-balanced high level analogue and digital spectrum analyser front-end is the result. Possible improvements and future work is discussed.

1.4 Structure of the report

From the preceding approach we can easily identify the structure of the report.

After this introduction in chapter 1, we will continue with the analysis of what a spectrum analyser is and does in chapter 2. This includes the treatment of the different specifications and their relations and trade-offs. Chapter 3 will analyse the spectrum analyser topology and its sub-blocks, after which we will continue with the frequency conversion architecture in chapter 4. In chapter 5 we will determine and assess the best filter solutions. Chapter 6 will discuss the measurement of the signal after the frequency selection and the location of the ADC. This gives the design analysed from a digital perspective in chapter 7. We conclude by simulating the resulting front-end design in chapter 8. The last chapter will discuss the achievements with respect to the set goals and derive conclusions. Chapter 9 will also include a discussion of possible improvements and next steps.

Chapter
2

2 Spectrum Analyser

This chapter will first discuss what a spectrum analyser is and the different types that are available. Next, important specifications will be treated with an indication of the wanted performance. Since a spectrum analyser supports up to a high frequency (several GHz), we examine the typical trade-offs for design parameters at radio frequencies[3] (RF). This also identifies the relations between the different specifications. With this knowledge and the suggested specifications we can compare the results with commercially available analysers and arrive at a competitive, coherent and reasonable set of requirements.

2.1 What is a spectrum analyser?

A spectrum analyser is a device to measure the spectral content of an electrical, optical or acoustic waveform, in our case an electrical signal.

The spectrum analyser often shows the power spectrum (the power in small bandwidths for a larger frequency range). At the most basic level, we can see the spectrum analyser as a device that selects a certain frequency, then measures the peak or root-mean-square (RMS) voltage and repeats that over the complete frequency range of interest with the help of a sweep generator. From the voltage, the power is determined with a known load impedance. Optionally the measured signal is processed further to deduce other information about the signal. These functional blocks are shown in Figure 2.

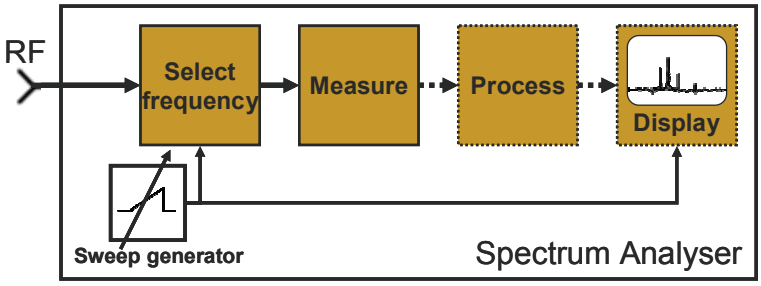


Figure 2. Spectrum analyser composed of functional blocks

Note that selecting a frequency and then further processing the signal is what a radio receiver does. Thus, the front-end (the first two blocks) of a spectrum analyser is a radio receiver front-end, which makes sense as a radio signal is a modulated electromagnetic wave, which becomes an electrical waveform after reception. Most radio receivers use the phase information of a signal, which is not strictly necessary for a spectrum analyser.

2.1.1 Sinusoid

Sinusoids are fundamental signals in physics and are encountered many times in nature, in anything that resonates or oscillates, for example in music or the eigenfrequencies of systems. The function of a sinusoid has the following form:

$x(t) = A \cdot \sin(\omega t + \varphi)$, where

A = peak amplitude

$\omega = 2\pi f = d\theta/dt$ =radian frequency [rad/s, f in Hz]

(2-1) t = time [s]

$T = 1/f$ = period [s]

φ = initial phase [rad]

$\theta = \omega t + \varphi$ = instantaneous phase [rad]

$$P = \int_{t}^{t+T} x(t)^2 \, dt = A_{RMS}{}^2 = \frac{A^2}{2} = \text{power}$$

[3] Radio frequency (RF) is a frequency in the range of the electromagnetic spectrum at which an alternating current can radiate electromagnetic waves by an antenna (if not shielded) and propagate through the atmosphere in turn being able to induce a current. This frequency is then useful for communication and typically in the range of a few kHz to 300 GHz.

The spectrum analyser measures the amplitude in voltage and normalises on the input impedance to get the power spectrum. Note that the frequency of a sinusoid is the derivative of the instantaneous phase with respect to time, which means the initial phase must also be taken into account if it is dependent on time such as for phase noise (section 2.3.7.3).

2.1.2 Complex sinusoid

A complex number is an ordered pair that extends the field of rational numbers into the complex field. A field is a set for which the operations of addition, subtraction, multiplication and division (except division by zero) may be performed. This ordered pair is connected in a way that provides a number of additional useful properties.

The first encounter with numbers came from counting and ordering, which resulted in the set of natural numbers $\mathbb{N}$. Extending this set with negative numbers was a major conceptual advance [1] resulting in the set of integers $\mathbb{Z}$, enabling the quantification of relative sizes between two sets even if the first is smaller than the second. Fractions or the ratio of sizes gives the set of rational numbers $\mathbb{Q}$, which can be extended to include all points on an infinite line with one to one correspondence with the set of real numbers $\mathbb{R}$. Real numbers are used to measure continuous quantities, as some numbers such as $\sqrt{2}$ do not have a rational representation and thus includes everything in between rational numbers on an infinite line. We can extend real numbers with imaginary number (which are no more "imaginary" than a real number [1]), which are numbers whose square give a negative number. Together these orthogonal sets form a complex plane of the set of complex numbers $\mathbb{C}$, with one axis the real axis and the other the imaginary axis.

Sine waves also have a complex representation. Sinusoids are related to complex frequencies by Euler's formula, where the complex exponential is the polar form of coordinates in a complex plane:

$$r \cdot e^{j\theta} = r \cdot \cos(\theta) + j \cdot r \cdot \sin(\theta)$$

(2-2)
$$A \cdot \cos(\omega t + \varphi) = A \cdot \frac{e^{j(\omega t+\varphi)} + e^{-j(\omega t+\varphi)}}{2}$$

$$A \cdot \sin(\omega t + \varphi) = j \cdot A \cdot \frac{-e^{j(\omega t+\varphi)} + e^{-j(\omega t+\varphi)}}{2}$$

The complex exponential is called a complex sinusoid and it follows that a sine or cosine wave consists of two complex sinusoids with the same radius and the same angle with one positive and the other negative. The derivate or rate of change of the angle or phase with respect to time is the frequency of the exponential and this frequency can now also be negative. A positive frequency signifies a counter-clockwise rotation in the complex plane as we can see from Figure 3, while a negative frequency means a clockwise rotation. A negative frequency gives the same result as the positive frequency for a cosine, but a 180° phase shift for a sine.

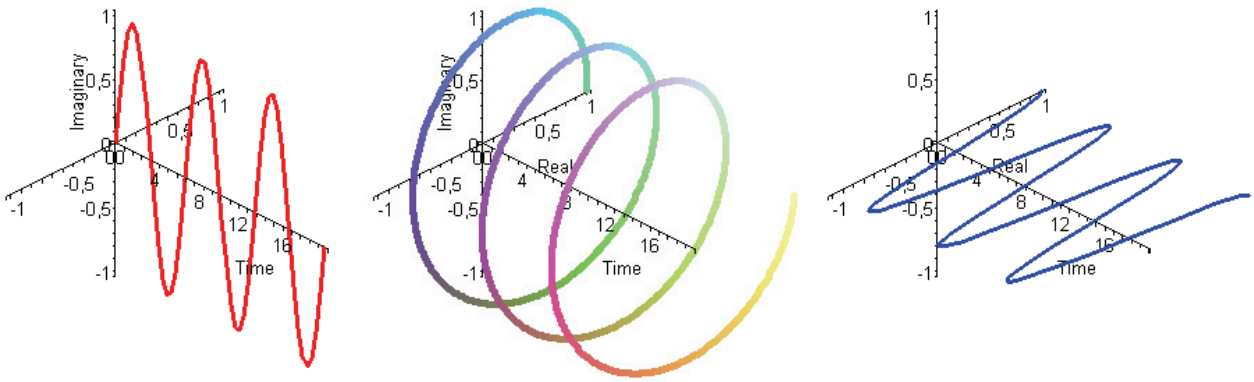


Figure 3. Complex sinusoid with projection on the real and imaginary axis.

From the complex plane, we see that a sine wave is a projection from a circular motion on any straight line, where each line has a different initial phase. Coinciding with the real axis is a cosine wave, and coinciding with imaginary axis is a sine wave being 90° or ½π radians apart. We can also visualize how a cosine wave consists of two complex frequencies with opposite rotation as the projections on the imaginary axis of these two complex exponentials would be exactly opposite and cancel out, leaving only a cosine on the real axis.

2.1.3 Fourier transform

A signal in the time domain is determined by its value or amplitude at each instant in time. For a representation in the frequency domain we perform a Fourier transform:

(2-3) $$X(f) = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt = |F(f)| e^{-j\theta(f)}$$

The result is a function returning a complex number at the provided frequency. This complex number gives an amplitude and phase. The absolute value or magnitude $|F(f)|$ gives the amplitude (for a frequency f), while the complex argument $\theta(f)$ gives the phase shift (for a frequency f) relative to the phase at frequency zero. Over a frequency range these are called the (Fourier) magnitude spectrum and the phase spectrum respectively.

Theoretically the signal must be evaluated over all time, from minus infinity to infinity. Of course this is practically not possible and the signal must be measured over a finite time period, with the start of the measurement at $t=0s$. For relatively constant signals, the results are still acceptable showing the current spectral content of the signal.

There are two sufficient conditions for the Fourier transform to exist [2]; the signal must be absolutely integrable and the signal must be “well behaved” (of bounded variation). Power or energy signals fall into this category [2]. Unfortunately, periodic waveforms with infinite power or random signals do not [4].

2.1.4 Stochastic signals

Because anything can be an input signal to the spectrum analyser, this includes random signals. A precise deterministic description of the signal is then not (easily) possible. In that case, the signals are often best modelled as stochastic signals [6]. The expected value of the stochastic signal is:

(2-4) $$E[x(t)]$$

The Fourier transform of stochastic signals does not exist, but the frequency domain characteristics can be captured in a different function; the power spectral density [5].

2.1.5 Power spectral density

The magnitude spectrum of the Fourier transform gives the amplitude at each frequency. The Fourier transform can thus be approximated by measuring the amplitude at each frequency. Therefore we select a small frequency band (i.e. with a band-pass filter) and then measure the average voltage for that frequency range. Thus, the result is a voltage or power per Hertz.

The instant voltage of a sine wave at the frequency we are measuring could be anywhere between the peak values, but since we do not know at which time in the period we measure, we can not infer the peak or root-mean-square (RMS) value. We could measure the peak value by keeping the largest value over a number of measurements. However, if we have two slightly different frequency components in one frequency band, their amplitudes would sometimes add up and sometimes subtract because of changing phase differences and the peak value would be the sum. This is not an approximation for the amplitude from a Fourier transform. Instead if we would square the value and take the average of a number of measurements we would get the power in the general sense or the mean-square value. We can get the real power from this if we know the input impedance or we could see it as the power with a 1Ω input impedance. We can also get the RMS voltage from the power by taking the root, which is equal to the amplitude of the Fourier transform divided by √2.

The power of a deterministic signal is given by [5]:

(2-5) $$P = \lim_{T\to\infty} \frac{1}{2T} \int_{-\infty}^{\infty} x_T(t)^2\, dt = \lim_{T\to\infty} \frac{1}{2T} \int_{-\infty}^{\infty} |X_T(f)|^2\, df = \int_{-\infty}^{\infty} \lim_{T\to\infty} \frac{1}{2T} |X_T(f)|^2\, df$$

The signal $x_T(t)$ is a truncated version of $x(t)$. Note that the truncated version has finite energy. The area under the integral is called the power spectral density (PSD):

(2-6) $$S_x(f) = \lim_{T\to\infty} \frac{|X_T(f)|^2}{2T},\ X_T(f) = \int_{-T}^{T} x(t) e^{-j2\pi ft}\, \partial t$$

A property of the PSD is [5]:

(2-7) $$S_y(f) = |H(f)|^2 S_x(f)$$

For a band-pass filter with a small bandwidth Δf, around f_c, it then follows:

(2-8) $$|H(f)| = 1$$
$$S_y(f) \cong S_x(f_c) \text{ for } f_c - \frac{\Delta f}{2} \le |f| \le f_c + \frac{\Delta f}{2} \text{ and 0 otherwise}$$

So substituting (2-8) into (2-7) and assuming that $S_x(f)=S_x(f_c)$ around f_c, we find:

(2-9) $$P_y \cong S_x(f_c)\Delta f \text{ and } S_x(f_c) \cong \frac{P_y}{\Delta f}$$

This means the average power in a small bandwidth is an approximation of the power spectral density. The Fourier transform approximation by measuring at multiple small bandwidths is thus actually a PSD approximation.

Interestingly, the Wiener-Khinchin theorem relates the power spectral density to the autocorrelation function via [5]:

(2-10) $$S_x = \Im\{R_{xx}(\tau)\}$$

Thus the PSD is the Fourier transform of the autocorrelation function. The continuous autocorrelation function of a deterministic signal is defined as:

(2-11) $$R_{xx}(\tau) = \int_{-\infty}^{\infty} x(t) \cdot x(t-\tau)^*\, dt$$

The power spectral density of a sinusoid results in the same (discrete) complex Fourier coefficients, only squared. The autocorrelation function of a sinusoid is a sinusoidal function with the same period and the average power of the sinusoid as amplitude [5].

The autocorrelation function for a (wide-sense) stationary random signal is:

(2-12) $$R_{xx}(\tau) = E\left[x(t) \cdot x(t-\tau)^*\right]$$

Instead of using the Fourier transform for a frequency domain representation of the deterministic signal $x(t)$, the power spectral density is used as a frequency domain representation of deterministic and stochastic signals with finite power. The average power is measured for a finite time, and is therefore an approximation. Averaging multiple measurements thus provides a more accurate average power and therefore power spectral density estimation.

Thus instead of the Fourier transform of the input signal, the spectrum analyser determines the Fourier transform of the auto-correlated input signal.

For the discrete Fourier transform (see section 7.4.2), the truncating of the signal causes a convolution with the Fourier transform of the truncation window in the frequency domain. This can be seen as applying a filter to the spectrum for each determined frequency bin, therefore also determining a PSD.

Note that a spectrum analyser normally does not normalise on the bandwidth.

A spectrum analyser approximates a Fourier transform by giving the approximate PSD in a small selectable bandwidth, repeats this measurement for each frequency band in a range and provides the resulting graph. The selectable bandwidth is called the resolution bandwidth, since it determines the resolution (in frequency) of the measurement. Using the voltage (amplitude) instead of the power gives the amplitude spectral density.

2.2 Types

The signal in the frequency domain is completely determined by its frequency, amplitude and phase. Note that before we only discussed measuring the amplitude at a certain frequency. Traditionally, spectrum analysers only were able to measure the amplitude, which is still very useful. Typical measurements that are made are the frequency, voltage or power of a signal, which can be used to determine things as the distortion, noise, modulation or occupied bandwidth.

Modern spectrum analysers can also measure the relative phase (see 2.1.3) of a signal. Practically, this is achieved by doing the Fourier transform in the digital domain. This also enables the measurement of single shot phenomena as well as the signal in the time domain, necessary for example for TDMA, or in the modulation/code domain for GSM or WDMA. Typical measurements include rise and fall times, overshoot and ringing in the time domain and phase error, constellation, trellis and eye diagrams in the modulation or vector signal domain. These analysers are sometimes called (vector) signal analysers, but literally spectrum also includes the phase and spectrum analyser can also be used. Signal analyser is then used to indicate time domain measurement capabilities.

The first spectrum analysers did not have access to digital technology and therefore consisted completely of analogue components. Those are the analogue spectrum analyser types.

While digital technology gradually replaced more and more parts of the spectrum analysers with digital counterparts, their design was still basically the same as the analogue version. This changes when the Fourier transform is performed digitally enabling measurement of the phase. While it is possible to measure the phase with an analogue spectrum analyser, only the digital Fourier transform enables advanced modulation domain analysis. Thus, we call a spectrum analyser which performs a digital Fourier transform a digital spectrum analyser type.

As more and more of a spectrum analyser is digitised, the front-end resembles the concept of software defined radio, which tries to move the ADC as much to the front as possible. This gives as much flexibility as possible as the processing of all (supported) radio standards is then done in software. However, the ADC is not possible at the high frequencies and resolution needed. Thus, today's software defined radio uses a down-conversion step and resembles the vector signal analyser's front-end with emphasis on high flexibility of radio standards.

The following overview of spectrum analyser types also approximately shows the evolution of spectrum analysers, going from analogue RF front-ends to software defined radio. This evolution is also shortly described at the history of spectrum analysers in appendix B.

2.2.1 Analogue

The most basic frequency selection is performed by a fixed band-pass filter followed by an envelope detector to perform demodulation and get the peak voltage. This principle is used by the multi-channel analyser.

The next step is making the band-pass filter tuneable giving a tuned radio frequency (TRF) analyser. The filter is essentially swept through the input range.

The super-heterodyne analyser further advances by using a mixer to do the tuning followed by a fixed high quality band-pass filter and envelope detector. Now, the input range is swept through a fixed filter.

Note that designs that use sweeping need a relatively static signal as a signal at a certain frequency is not measured continuously, but only once every sweep.

2.2.1.1 Multi-channel spectrum analyser

The first spectrum analyser design was a result of using multiple simple band-pass filters in parallel; each tuned at a slightly different frequency (Figure 4). Each output is a channel and can then be used as the Fourier amplitude at the frequency of that filter.

The advantages of this design are that you can achieve good resolution by taking a lot of filters and the quality of the filters can be good because they are at a fixed frequency. Furthermore, since they are all evaluated in parallel the speed is very good, also giving a stable image.

The disadvantages are that the design is not very flexible and can be very expensive as the required resolution and input range and thus the number of filter grows. For example, a 1kHz resolution with a 1GHz input range gives 1 million filters. Also, the filters become much more difficult to make at higher frequencies as the Q ($=f_c/\Delta f_{BW}$) of the filter increases with frequency. Finally, mismatch between channels lowers the amplitude accuracy when using amplitudes from multiple channels.

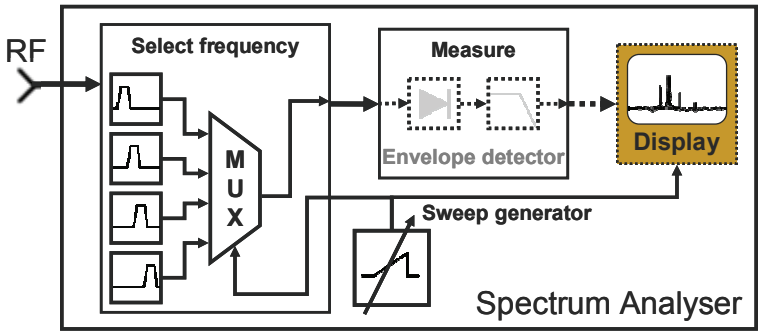


Figure 4. Multi-channel spectrum analyser

2.2.1.2 Swept-tuned or tuned radio frequency spectrum analyser

The tuned radio frequency (TRF) analyser uses a tuneable band-pass filter, which is tuned by the sweep generator to perform the frequency selection as shown in Figure 5.

The advantages of this design are a simple and cheap design with a reasonable frequency range.

The disadvantages are that a tuneable filter is hard to make over a large frequency range, because the frequency poles must change over a large range. This requires (possibly off-chip such as MEMS) component values to be accurately changeable over a large range and/or requires complex circuitry [39]. The values must also be stable; otherwise the filter stability is not very good giving poor frequency stability at the output. Finally the filter bandwidth and therefore the resolution is not constant over the whole tuning range limiting their use (because $Q=f_c/\Delta f_{BW}$).

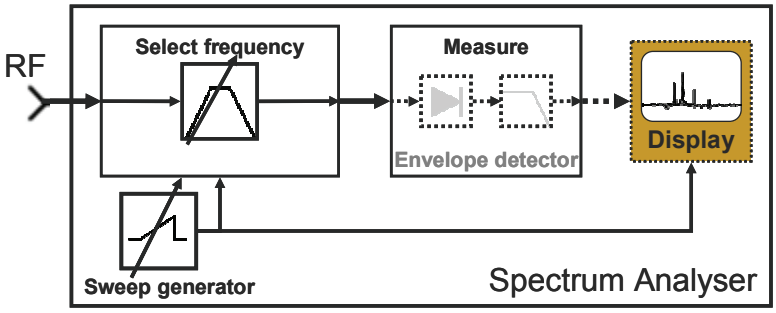


Figure 5. Tuned radio frequency spectrum analyser

A possibility exists to combine this design with the multi-channel analyser to keep the advantages of the latter while lowering the number of required filters.

2.2.1.3 Super-heterodyne spectrum analyser

The super-heterodyne analyser uses a mixer to convert an input frequency (RF) to a fixed intermediate frequency (IF). The local oscillator (LO) that mixes with the input signal can then be used to tune which input frequency is translated to the IF. The intermediate frequency is then band-pass filtered, possibly amplified, measured and displayed.

The advantage of this design is that the band-pass filter and optional amplifier are at a fixed frequency. This means they can be of good quality, especially if a low IF is chosen. This also means everything after the mixer is independent of the input range. This range is therefore determined by the frequency range of the LO and the mixer, which can be very large. The conversion can even be done in multiple stages enabling even higher input ranges and better filtering. Finally, the design is still relatively simple and cheap.

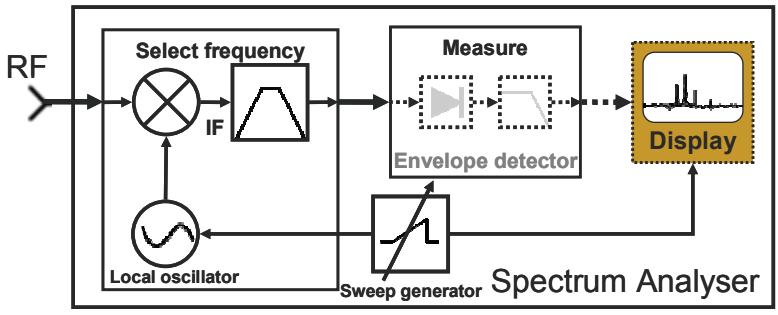


Figure 6. Super-heterodyne spectrum analyser

The disadvantage is that the mixer introduces distortions, depending on the quality of the mixer. In general every stage of the design introduces extra distortions and this is an extra stage. However, mixing gives an especially important distortion because two input frequencies will mix to one intermediate frequency. The distortion or image signal is further discussed in section 4.4. A second disadvantage is that since there is only one band-pass filter, each band in the frequency range of interest must be evaluated sequentially (the input signal must be swept through the IF filter).

Again this disadvantage can be diminished by using more than one IF frequency with corresponding band-pass filters as in the multi-channel analyser.

The advantages of the super-heterodyne spectrum analyser normally outweigh the disadvantages compared to the other two architectures, in almost every application. Therefore, the super-heterodyne architecture is still used today in the vast majority of radio receivers (radio, TV, mobile phones, Bluetooth, Wireless LAN etc.) [4][13].

2.2.2 Digital

As explained we call a spectrum analyser digital if it performs the Fourier transform digitally with a fast Fourier transform (FFT) algorithm. A Fourier analyser will directly digitise the input signal, while a (vector) signal analyser will digitise at an intermediate frequency. A real-time signal analyser adds advanced real-time triggering possibilities.

Almost all modern general purpose spectrum analysers are of the digital type because of their added flexibility and time/code/modulation measurement features [15]-[21]. Most are vector signal analysers, because of their larger input frequency range.

2.2.2.1 Fourier or FFT analyser

A Fourier analyser directly digitises the input signal by using an analogue-to-digital converter (ADC). Optionally an amplifier and/or anti-alias filter is used to optimise the input signal for a maximum ADC resolution. We then have the digitised input signal in the time domain, which is further processed digitally and which we can convert to the frequency domain by a FFT. This architecture is shown in Figure 7.

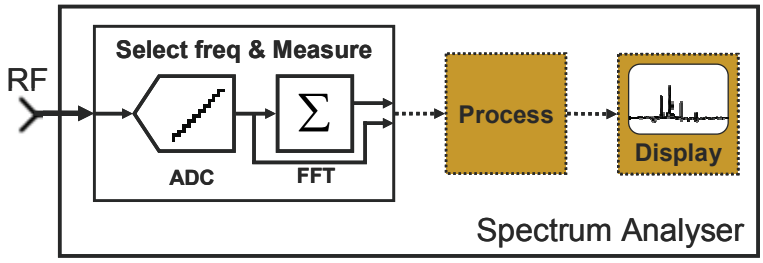


Figure 7. Fourier analyser

The advantage is that time, frequency, modulation and code domain analysis can be performed and that single-shot phenomena can be characterised, since a Fourier transform on a sampled input signal is performed instead of an average voltage measurement.

The disadvantage is that a high speed high resolution ADC is hard to achieve posing limits on the input frequency range and sensitivity. The maximum input frequency of a Fourier analyser is currently (2005) about 40 MHz (based on 12-bit ADC performance [48]), while other super-heterodyne analysers can go up to 50 GHz [15][16].

2.2.2.2 Vector signal analyser

The (vector) signal analyser uses the same design as the super-heterodyne analyser (see section 2.2.1.3), but digitises the input signal at the intermediate frequency. The band-pass filter can now optionally act as a pre-select filter to limit distortions from other frequencies. The filter also enables the use of auto-ranging (adapting the signal to an optimal level to reduce noise and distortions) and possibly can be used as an anti-alias filter for the ADC (to prevent folding of out-of-band frequencies into the band). In this way the advantages of the

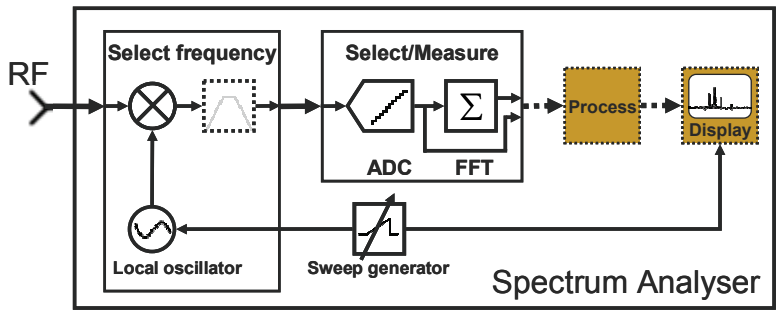


Figure 8. Vector signal analyser

super-heterodyne analyser are combined with the advantages of the Fourier analyser.

The advantages are therefore a very large input range up to 50 GHz with digital processing to enable time, frequency, modulation and code domain analysis.

The disadvantages are the same as for the super-heterodyne design.

The mixer and IF stage can be by-passed for low frequencies that the ADC can handle, giving essentially a Fourier analyser.

2.2.2.2.1 Real-time analyser

While vector signal analysers can perform snap-shot analysis in several domains, their triggering capabilities are relatively limited. Many measured signals however change unpredictably over time. Sometimes called real-time analysers, these devices can trigger on bursts, glitches and other time-varying events and capture and process the following signal. This extension of the vector signal analyser enables capturing important transient events, but this is mostly achieved in software using the same front-end design as the vector signal analyser. [23]

2.3 Specifications

This section describes the functional requirements (except for type and segment) of this project. It deals with the constraints of each specification and provides a wanted performance. Section 2.4 will discuss the different relations and trade-offs between the specifications, while the comparison with other spectrum analysers in section 2.5 provides a balanced final set of specifications.

From the assignment description we can identify the non-functional requirements. The important ones being low-cost and high integrateability, while of secondary concern are a flexible design with high functionality, such as supporting multiple standards and time and modulation domains.

2.3.1 Type and market segment

General (high-end) spectrum analysers are versified by respected brands into middle class and top-class, with further differentiations into application, such as portable or general purpose. However, the aim of this project is to investigate the possibilities of a low-cost integrated solution with moderate requirements compared to high-end devices. The intended applications include modern wireless communication standards and low-end spectrum analysers do not provide adequate performance for this domain. This means considerably higher requirements than available low cost solutions (less than €1000,-). See also section 1.1.3.

2.3.2 Frequency range

The input frequency range is the range between the highest and lowest frequencies that the spectrum analyser can analyse with the provided specifications.

To be effectively used in combinations with modern and future technologies, we would like to include the GSM and UTMS bands as well as Bluetooth and WLAN in the free available 2.4GHz band. Thus, we would at least like to support a maximum frequency of about 3GHz. Note that WLAN IEEE 802.11a and HiperLAN/2 operate at 5GHz, thus supporting up to 6GHz could be advantageous.

The low-end band is less critical as it is in general more easily supported and is only limited by a small frequency range to be used by the device itself. It might be useful to include the AM bands (from 150kHz) as digital radio is introduced on these bands.

2.3.3 Amplitude range

The amplitude range we would like to support is an 80dB range preferably from -80dBm to 0dBm (22.4μV ~ 0.224V at 50Ω), which is most stringent at the maximum resolution bandwidth (see section 2.3.9). This includes most signals of interest; however supporting larger signals up to 30dBm (1W or 7.07V at 50Ω) could be beneficial, but is of lower importance than other specifications. With an amplifier/attenuator (LNA) we can scale the input to the most optimal range for the preceding stages.

2.3.4 Amplitude accuracy

The amplitude accuracy is the accuracy of the measurement of the input signal. The signal at the input has a certain amplitude level, which does not have to remain constant throughout the stages of the spectrum analyser. The signal can for example be amplified or digitised. Since this amplification can not be completely accurate and digitising means rounding the actual value to a discrete value, the final measured value is not exactly the amplitude at the input.

Absolute	**± dB**
Frequency response	0.5 ~ 4
Distortions	0.1 ~ 2
Calibrator	0.2 ~ 1
Relative	**± dB**
Frequency response	0.5 ~ 4
Resolution bandwidth switching	0.1 ~ 1
Reference level (IF gain)	0.1 ~ 1
Scale fidelity	0.5 ~ 2
Band switching	0.5 ~ 1
Input attenuator switching	0.5 ~ 2
Display scale switching	0.0 ~ 1

Table 1. Typical amplitude uncertainty factors

The measurement of the amplitude is either absolute or relative, where relative means the amplitude difference between two measurements. The factors adding to the amplitude uncertainty for a typical spectrum analyser are provided in Table 1. Band switching is caused by the local oscillator, while scale fidelity is caused by non-linearities the measuring part. An input attenuator scaling the signal also has an uncertainty, which can be different at different frequencies. The same holds for display scales in analogue spectrum analysers. The main focus of this thesis is the frequency selection front-end which encompasses the frequency response, distortions, the uncertainty of different resolution bandwidth filters and uncertainty in the reference level compared to the IF level. All the relative factors (except frequency response) are calibrated against a know amplitude at a fixed frequency giving the absolute calibrator uncertainty. Thus absolute measurements are relative to this calibration frequency and the uncertainty depends on the frequency response and distortions as well as the calibration accuracy. For a more thorough explanation see [9].

As the primary objective of a spectrum analyser is to measure the amplitude, we would like the result to be fairly accurate. Therefore, the error in the absolute amplitude accuracy must be smaller than ±2dB in order for the results to be usable in real-life applications.

Note that distortions include noise and linearity distortions. As will be explained later, we want the noise and distortions to fall under the amplitude range. This means that for most of the input range (except for the smallest signals), the noise and distortions are very small compared to a signal of interest and they negligibly distort the amplitude accuracy. Therefore noise and linearity distortions are not taken as part of the amplitude accuracy. Other distortions such as for example gain inaccuracy are.

2.3.5 Sweep time

The sweep time is the time the spectrum analyser needs, to display the currently selected frequency range with the selected resolution.

In an analogue design (no FFT, see section 2.2) the results of a measurement at a certain frequency are measured in turn. If the results are displayed directly as they were for the first spectrum analysers, the results are swept over the screen and the time it takes is therefore the sweep time. Thus, the selection of a smaller bandwidth with the same frequency range results in a longer sweep time as more bandwidths now fit in this range. Nowadays as the screens are self-refreshing and the results are cached, the maximum sweep time is only limited by what the user will tolerate provided that the spectrum analyser measurements will not fluctuate so much that the result will become unusable. The minimum sweep time is even less important as the signal would be averaged or peak-detected or demodulated if it changes too much or too fast to follow.

For a digital design each frequency measurement is done in parallel, thus the sweep time can be much less. However, for an accurate measurement a complete period must have passed, meaning 1s for 1Hz for example, plus some processing time.

As an indication the sweep time will be somewhere between 1ms and 1000s. For normal measurements the sweep time is preferred to stay under 1s for usability reasons. Note that for longer sweep times, temperature variations etc. can limit the accuracy of the measurement.

2.3.6 Resolution bandwidth

A selected frequency band can not be measured with infinite resolution as it would take infinite time to do so. Thus the input range is separated in a number of bands of which the value of each is measured. The smaller this band, the higher the measurement resolution of the selected frequency band is. The bandwidth of these frequency bands is therefore called resolution bandwidth (RBW).

The resolution of the measurement must still be distinguishable on screen otherwise it is of no use. Thus, normally the number of bands in a frequency range is about the same and the bandwidth of the bands is adapted to the size of the range. However, a larger selection of bandwidths means more resolution bandwidths must be implemented by the design.

There are about 500 to 1000 points of resolution on screen and thus bands in a normal spectrum analyser result. The largest frequency range is the complete input range from 0 to 3/6GHz, but it is unlikely one would display the complete range at once at it would be too generic and provide no useful information. Larger bandwidths also have more noise making it more difficult to guarantee the signal to noise ratio. Thus, we choose the largest RBW as 1MHz allowing us about 1GHz on screen and more if we allow more resolution points or frequency gaps between measurements.

The smallest frequency range can be as small as wanted, however the RBW must then also become very small. As the channels used even at low frequencies of a few MHz are still a number of kHz wide, we select 10kHz as the smallest RBW.

The number of RBWs in between determines the flexibility of the design. You would not want to go from a 1MHz RBW, which is just too large for good measurement, to a 10kHZ bandwidth as it would take much too long for the size of the frequency range when at a 1MHz resolution. More RBWs make the design more expensive as each must be implemented. As a compromise we choose seven RBWs of 10, 20, 50, 100, 200, 500 and 1000kHz (steps of 0/0.3/0.7/1 on a log scale).

If we analyse the signals in the digital domain with an FFT, we can easily support much more and smaller RBWs only having influence on the measurement time and required memory. Therefore, the provided limits are mainly useful for analogue designs. However larger bandwidths do set constraints on the digitisation of signals limiting the largest RBW size by the capabilities of the analogue to digital converter (ADC) and the smallest RBW by the available time.

2.3.7 Noise

For spectrum analysers noise indicates indeterminate unwanted signal variations. This noise is super-positioned on the signal of interest, thus if we want to measure the signal of interest reliably we need this noise to be sufficiently small.

Present in all circuits and often dominating is thermal noise [3]. This noise is generated by stochastic variations in the movement of electrons. The electrons move more at a higher temperature and the fundamental thermal noise limit is thus determined by the temperature. This thermal noise is present at the input as part of the input signal and is stochastic and therefore present at all frequencies. The total (thermal) noise is thus dependant on the bandwidth of the RBW, as a higher bandwidth will include more noise. Note that other kinds of noise exist such as flicker noise and shot noise, which are not discussed here.

The final factor determining the noise level when the signal is measured is the noise the stages of the spectrum analyser add. Since a noiseless amplifier amplifies the signal as well as the noise, it is easier to talk about a signal-to-noise ratio (S/N) with S the signal

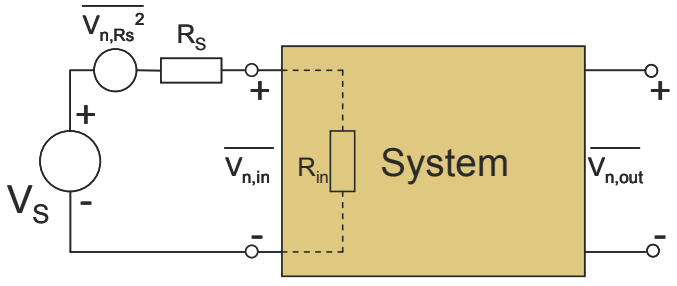


Figure 9. System with input signal, source and input impedance and noise

power and N the noise power.

We can represent a system (i.e. our front-end) as a black box with an input impedance R_{in} and an input and output referred noise power $V_{n,in}$ and $V_{n,out}$. This is shown in Figure 9. Connected to this system is an input signal with a source impedance R_S giving a corresponding (thermal) noise source. If we assume that the source impedance is matched to the input impedance of the system (as it normally is to avoid reflections), we find:

$$\overline{V_{n,Rs}{}^2} = 4kTR_S \cdot \Delta f, \quad R_S = R_{in}$$

(2-13)
$$\overline{V_{n,in}{}^2} = \overline{V_{n,Rs}{}^2} \cdot \left(\frac{R_{in}}{R_S + R_{in}} \right)^2 = 4kTR_S \cdot \Delta f \cdot \left(\frac{1}{2} \right)^2 = kTR_S \cdot \Delta f$$

If we represent these noise signals in power, we find:

(2-14)
$$N_{in} = \left(\frac{\overline{V_{n,in}{}^2}}{R_{in}} \right)_{dB} = \left(kT \cdot \Delta f \right)_{dB} = 10 \log kT + 10 \log \frac{\Delta f}{\Delta f_{ref}}$$

The first term is the thermal noise per Hertz and the second is the bandwidth, in dB referenced to a bandwidth of 1Hz. Note that *10·log(kT)* referenced to 1mW at room temperature is about -174dBm/Hz.

2.3.7.1 Noise level (DANL)

The displayed average noise level (DANL) or noise floor is the total noise level at a certain resolution bandwidth. This internal noise level is the smallest signal the spectrum analyser can measure at that bandwidth before it disappears in the noise. If we assume the noise contribution of the spectrum analyser to be constant, this noise level is only dependent of the bandwidth. At the largest RBW, we find the largest noise level. Thus, we can lower the noise level, by choosing a smaller RBW.

The minimum input level of the spectrum analyser is defined at -80dBm (section 2.3.3). Because noise is a stochastic signal, it is always added to the input signal. With an amplitude accuracy of less than ±2dB, this means the noise level must be lower than about -82dBm. With some extra tolerance, we define a maximum noise level of -84dBm. As also noted at the amplitude accuracy section, the noise at -84dBm is negligible small compared to larger input signals at for example -60 ~ 0dBm. Therefore the noise is not taken as part of the amplitude accuracy. But a small signal levels of -80dBm ~ -70dBm, the amplitude accuracy is significantly degraded and thus worse than ±2dB.

2.3.7.2 Noise figure

The noise figure (NF) of a component is the noise it adds to the signal. Noise figure characterises the signal-to-noise-ratio (S/N) degradation in dB and is defined as the input S/N divided (minus in dB) by the output S/N.

With the maximum noise level of -84dBm and a largest RBW of 1MHz (=60dB Hz), we find a maximum noise figure at room temperature of:

$$N_{out} = N_{in} + NF_{system}$$

(2-15)
$$N_{out} = \left(kT \right)_{dB} + \left(\Delta f \right)_{dB} + NF_{system} < -84dBm$$

$$NF_{system} < -84dBm - \left(-174dBm/Hz + 60dB\,Hz \right) = 30dB$$

Thus we would like the total noise figure of all the stages of the spectrum analyser front-end to be less than 30dB. Note that if we achieve a lower noise figure, the noise level also becomes lower at the maximum RBW and we can observe smaller input signals.

2.3.7.3 Phase noise

A number of spectrum analyser types contain frequency generators or oscillators. These oscillators are also susceptible to noise. This noise can influence the amplitude as well as the phase of the generated signal (for a phasor representation this means variations in the radius and the angle). The amplitude variations can be seen as part of the noise

figure of the oscillator as they are not dependent on frequency. The amplitude variations must be small enough to keep the amplitude accuracy good enough, however this is normally not a problem because the mixer is implemented with switches passing the input signal without gain. The phase variation due to noise is called phase noise.

Phase noise is commonly specified as the single sideband noise power density in a 1Hz bandwidth at a certain offset from the carrier. This noise power is relative to the carrier signal power, which gives us:

(2-16) $$\pounds(\Delta\omega)\ in\ dBc\left|(\Delta f = 1Hz) = dBc/Hz\ at\ offset\ \Delta\omega\ at\ carrier\ frequency\ \omega_c\right.$$

If we represent the noise sources of a voltage controlled oscillator (VCO) by an equivalent input voltage noise source (Figure 10):

(2-17)
$$\omega_c = K_V \cdot V_{in}$$
$$\omega(t) = K_V\left(V_{in} + V_n(t)\right)$$
$$\varphi(t) = \int_0^t \omega(\tau)\partial\tau = \int_0^t K_V\left(V_{in} + V_n(\tau)\right)\partial\tau$$
$$\varphi(t) = \omega_c t + \varphi_n(t)$$
$$\varphi_n(t) = \int_0^t K_v V_n(\tau)\partial\tau$$
$$V_{out} = A_c\cos(\varphi(t)) = A_c\cos(\omega_c t + \varphi_n(t))$$

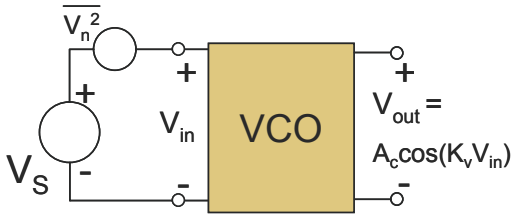


Figure 10. VCO with equivalent input voltage noise.

Thus, the noise sources manifest themselves as variations of the phase of the oscillator frequency. Of course there are also noise sources adding noise to the amplitude of the resulting carrier frequency. We can represent both noise sources where $\varepsilon_n(t)$ is the amplitude noise and $\varphi_n(t)$ is the phase noise:

(2-18)
$$x(t) = \left(A_c + \varepsilon_n(t)\right)\cdot\cos\left(\omega_c t + \varphi_n(t)\right), with\ \left|\varphi(t) \ll 1rad\right|$$
$$x(t) \approx \left(A_c + \varepsilon_n(t)\right)\cdot\left(\cos(\omega_c t) - \varphi_n(t)\cdot\sin(\omega_c t)\right)$$

If the phase variation due to noise are relatively small, this means the phase noise is translated to $\omega_c \pm \omega_n$ or around the carrier. The noise variations are equally likely in any direction for amplitude and phase, thus the noise forms a normal distribution around the carrier.

An oscillator is essentially a resonator, which in turn has a very small bandwidth band-pass filter characteristic with the corresponding falling frequency sidebands. The quality (Q) of the resonator/filter determines the bandwidth or the slope of the fall-off. Phase noise varies the carrier frequency with a normal distribution around the carrier. The amplitude of this carrier frequency undergoes the amplitude filtering with *1/f* frequency dependency of the resonator. The noise power than has a ($1/f^2$) dependence. Thus, the phase noise is shaped by the transfer function of the resonator generating the oscillator frequency. For a simple LC feedback oscillator, we have from Leeson's model[4] [3][4][29], where $\pounds(\Delta\omega)$ is normalised on power:

$$\left|H(j\omega)\right|^2 = 1 + \left(\frac{\omega_c}{2Q}\right)^2\frac{1}{\Delta\omega^2} \approx \left(\frac{\omega_c}{2Q}\right)^2\frac{1}{\Delta\omega^2}$$

(2-19)
$$S_{n,out}(\omega) = \overline{V_n^2} = kTZ\cdot\Delta f\cdot\left|H(j\omega)\right|^2$$
$$\pounds(\Delta\omega) = 10\log\left[\frac{1}{2}\frac{\overline{V_n^2}/Z}{P_c}\right] = 10\log\left[\frac{kT}{2P_c}\cdot\left(\frac{\omega_c}{2Q}\right)^2\frac{1}{\Delta\omega^2}\right]\left|(\Delta f = 1Hz)\right.$$

[4] The approximation is valid close around the carrier, i.e. with small Δω. $S_{n,out}$ is the power spectral density of the noise at the output of the oscillator. The noise factor F has been left out of consideration and the factor ½ comes from the single sideband definition of the phase noise.

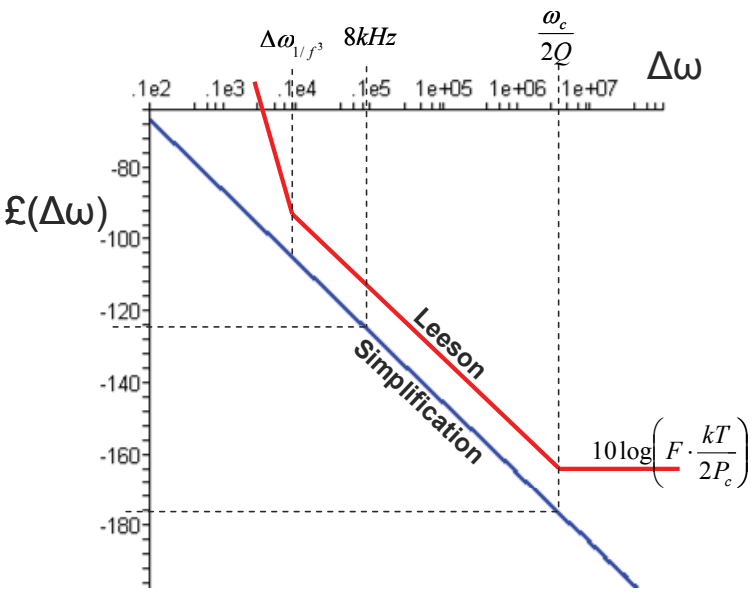


Figure 11. Single sideband phase noise.

The carrier signal power P_c, frequency and quality are fixed, which means the phase noise indeed has a $(1/\Delta\omega)^2$ dependency, with the phase noise power falling 20dB every decade away from the carrier. This is also seen in Figure 11, in which the blue line represents the above phase noise equation. If we look at a fixed offset frequency with fixed carrier power and oscillator quality, the phase noise becomes worse at higher carrier frequencies. Therefore, the oscillator side lobes remain the same relatively, but larger in the absolute sense, the higher the oscillator (carrier) frequency.

Note that the above model is a simplification, which can be extended with a noise factor (F), flicker noise dependencies ($\omega_{1/f}$) close to the carrier and the noise floor far away from the carrier. This model from Leeson is shown as the red line in Figure 11. [29]

The frequency conversion used in the two most promising architectures multiply the input signal(s) with the local oscillator. This results in the side lobes of the local oscillator to be transferred to the translated input signals at the output of the mixer. This is called "reciprocal mixing" and is displayed in Figure 12. The form of the side lobes is the two sided ($1/f^2$) dependence on a linear scale.

The phase noise thus generates side lobes for the input signal. These side lobes are always below the signal itself and therefore don't influence the measured peak amplitude much. However, there can be a small signal in the next resolution band, thus at that point a possibly large input signal's side lobes must be attenuated enough not to interfere with the small signal as can also be seen in Figure 12. The phase noise distortion in the next bandwidth must again be smaller than -84dB (4dB under the smallest input amplitude level of -80dBm). The smallest RBW is 10kHz, however this is distributed around the local oscillator signal. Therefore the next RBW for a single side band starts at a 5kHz offset. For a constant frequency dependence, the noise falls monotonically on a logarithmic scale, which means the average noise level of a 10kHz bandwidth is at about 3kHz. Thus at 8kHz, the side lobes must be below -84dBm with a 0dBm maximum input signal, which means a phase noise specification of 0-84-40[5]= -124dBc/Hz at 8kHz. This is shown in

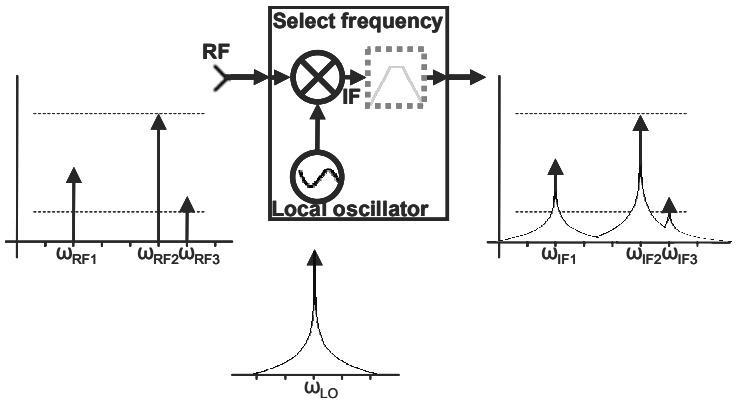


Figure 12. Reciprocal mixing.

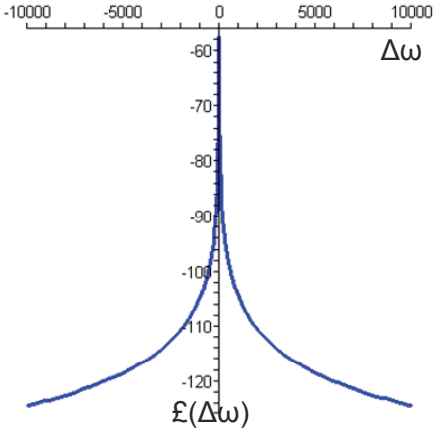


Figure 13. Phase noise frequency side lobes.

[5] The addition of the bandwidth in dB to the phase noise specification is only possible if the phase noise has constant frequency dependence in the corresponding bandwidth. Otherwise proper integration of the phase noise over the bandwidth must be performed.

Figure 13 with a linear scale for $\Delta\omega$ and in Figure 11 with a logarithmic scale for the upper sideband. As an indication, for the maximum RBW of 1MHz, we find a phase noise specification of -144dBc/Hz at 0.8MHz. Because the phase noise drops with 20dB per decade ($1/f^2$), the phase noise requirement at 8kHz is more strict (-124dBc/Hz at 8kHz gives -164dBc/Hz at 0.8MHz).

With $\pounds(\Delta\omega)$= -124dBc/Hz, kT= -174dBm, P_c=0dBm, $\Delta\omega$=8kHz and ω_c=3 or 6GHz, we find:

$$\left[2Q^{-2}\right]_{dB} = \left[\pounds(\Delta\omega)\right]_{dB} - \left(\left[kT\right]_{dB} - \left[P_c\right]_{dB} + \left[\left(\frac{\omega_c}{\Delta\omega} \right)^2 \right]_{dB} \right) \tag{2-20}$$

$$-2 \cdot \left[2Q\right]_{dB} = -124 - \left(-174 - 0 + 2 \cdot \left(94.8 - 40\right)\right) = -59.6 \Rightarrow Q \approx 475$$

Which means Q must be about 475 for a maximum frequency of 3GHz and about 950 for 6GHz. Typical values for Q are in the range of 1~10 for integrated resonators/filters, while external oscillators are in the range of 100~1000. As a first indication, integrated oscillators are not an options and external is already quite difficult. Therefore a PLL loop with an external crystal is normally applied. The crystal has a very high Q, giving a flat sideband close to the carrier and the monotonic fall-off of the VCO further out.

Thus we found that the phase noise requirement is the most strict in the smallest RBW as intuitively obvious. We also found that the phase noise is dependent on the local oscillator signal power (P_c), the quality (Q) and the frequency (ω_c), which must be taken into account when comparing oscillator performance. The phase noise becomes especially problematic at high frequencies, since the needed noise attenuation $\pounds(\Delta\omega)$ at the 8kHz offset $\Delta\omega$ for the smallest RBW stays the same, which means the needed Q is large. Therefore, it is useful to use low local oscillator frequencies when possible.

2.3.8 Linearity

Non-linearity causes several different distortion products. Each distortion product must be small enough so it does not interfere with the original input signal at that frequency. This means that each distortion product must have an amplitude that is smaller than our smallest input signal minus 2dB amplitude accuracy and a small margin. As with the noise level, this is at -84dBm.

Important distortion products, which are discussed below, are harmonics and intermodulation products. Both have distortion products dependent on the amplitude to a certain power, also called the order. Neglecting higher order products, we can identify second and third order performance measures called second-order intercept point (IP_2) and third-order intercept point (IP_3).

2.3.8.1 Harmonics

We can represent a signal as an infinite (power) series expansion. When a pure sinusoid is applied to a non-linear system the output signal is represented as a power series expansion, consisting of the fundamental frequency with higher order distortion products called harmonics (H_x, with x the order). If we assume a memory-less system and that higher than third-order distortion products become very small and can be neglected, we have:

$$y(t) = \alpha_1 x(t) + \alpha_2 x^2(t) + \alpha_3 x^3(t) = \alpha_1 \cdot A\cos\omega t + \alpha_2 \cdot A\cos^2\omega t + \alpha_3 \cdot A\cos^3\omega t$$

$$= \frac{\alpha_2 A^2}{2} + \left(\alpha_1 A + \frac{3}{4}\alpha_3 A^3 \right)\cos\omega t + \frac{1}{2}\alpha_2 A^2 \cos 2\omega t + \frac{1}{4}\alpha_3 A^3 \cos 3\omega t \tag{2-21}$$

We can separate the result in two groups, even-order harmonics including DC and odd-order harmonics. For each group, a harmonic gets amplitude terms from all higher harmonics but not from lower harmonics. Thus third order distortion products result in an amplitude term at the fundamental and a third-order harmonic, while fifth order distortion products results in an amplitude term at the fundamental and third-order harmonic and a fifth order harmonic.

Note that only the wanted term $\alpha_1 \cdot A$ is proportional with the amplitude, while the distortion products are proportional to α_2 and the amplitude squared or α_3 and the

amplitude cubed (etc. for higher order harmonics). Thus, the resulting distortion products depend very much on the amplitude used.

The higher harmonics are indicated in Figure 14 in orange as H_x resulting from two signals at ω_1 and ω_2.

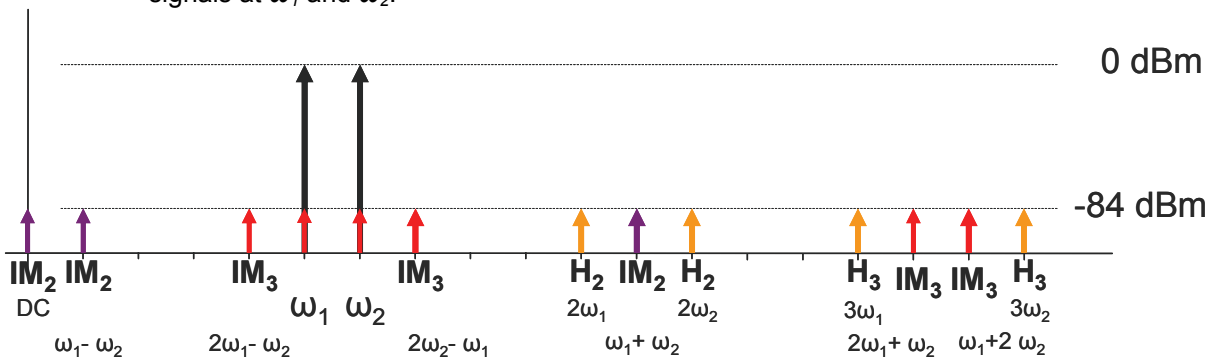


Figure 14. Harmonic and intermodulation distortion products.

2.3.8.2 Intermodulation

When more sinusoid signals are present, they interact to generate signals at different frequencies then the harmonics. This is called intermodulation (IM_x) and generates the following terms:

$$x(t) = A_1 \cos\omega_1 t + A_{\ 2}\cos\omega_2 t$$

(2-22)

$$y(t) = DC : \frac{1}{2}\alpha_2 {A_1}^2 + \frac{1}{2}\alpha_2 {A_2}^2$$

$$\omega_1, \omega_2 : \left(\alpha_1 A_1 + \frac{3}{4}\alpha_3 {A_1}^3 + \frac{3}{2}\alpha_3 A_1 {A_2}^2\right)\cos\omega_1 t + \left(\alpha_1 A_2 + \frac{3}{4}\alpha_3 {A_2}^3 + \frac{3}{2}\alpha_3 {A_1}^2 A_2\right)\cos\omega_2 t$$

(2-23)

$$\omega_1 \pm \omega_2 : \alpha_2 A_1 A_2 \cos(\omega_1 + \omega_2)t + \alpha_2 A_1 A_2 \cos(\omega_1 - \omega_2)t$$

$$2\omega_1 \pm \omega_2 : \frac{3}{4}\alpha_3 {A_1}^2 A_2 \cos(2\omega_1 + \omega_2)t + \frac{3}{4}\alpha_3 {A_1}^2 A_2 \cos(2\omega_1 - \omega_2)t$$

$$\omega_1 \pm 2\omega_2 : \frac{3}{4}\alpha_3 A_1 {A_2}^2 \cos(\omega_1 + 2\omega_2)t + \frac{3}{4}\alpha_3 A_1 {A_2}^2 \cos(\omega_1 - 2\omega_2)t$$

Note that when the amplitudes are about the same we have $½\ \alpha_2\ A^2$ distortion products at DC and $\omega_1 \pm \omega_2$ and $¾\ A^3$ distortion products at the fundamental and at $2\omega_1 \pm \omega_2$ and $\omega_1 \pm 2\omega_2$. Especially the minus terms are of interest because the resulting product can be very close to the fundamental if ω_1 and ω_2 are close.

The different intermodulation products are also shown in Figure 14 in purple for IM_2 products and red for IM_3 products.

2.3.8.3 Second-order intercept point (IP_2)

From the equations above, we find that the second order harmonics and IM_2 products grow with factor $½\ \alpha_2\ A^2$. Note that the IM2 products with $\omega_1 = \omega_2$ give the second order harmonic and DC products as expected. Besides the α factor, the second order distortion products grow twice as fast as the fundamental when the amplitude increases. This is a one times difference in slope on a logarithmic scale. Thus at a certain amplitude it must become equal to the gain ($\alpha_1 \cdot A$) and this depends on the ratio between α_1 and α_2. We can conveniently use this intercept point as in indicator of the second order linearity. This also holds if we use power instead of amplitude, which gives:

(2-24)

$$IP_2 = P_{\max} + \frac{1}{slope}\cdot\left(P_{\max} - P_{IM2,H2}\right) = 0dBm + \frac{1}{1}\cdot\left(0dBm - -84dBm\right) = 84dBm$$

The maximum input of 0dBm is P_{max} and -84dBm is the maximum distortion level.

2.3.8.4 Third-order intercept point (IP_3)

Third-order products grow with factor ¾ A^3 and thus grow three times as fast. The difference in slope is thus 2 on a logarithmic scale. As with the IP_2, we can define a third-order intercept point as in third-order linearity measure. In power this gives:

(2-25) $$IP_3 = P_{\max} + \frac{1}{slope}\cdot\left(P_{\max} - P_{IM3}\right) = 0dBm + \frac{1}{2}\left(0dBm - -84dBm\right) = 42dBm$$

Again with the maximum input of 0dBm as P_{max} and -84dBm is the maximum distortion level.

If we set out the input power against the output power of a system with non-linearities, the gain results in a shifted dependence with slope 1, while the second order non-lineairities have a dependence with slope 2 crossing at the IP2 point and the third-order non-linearities have a dependence with slope 3 crossing at the IP3 point. This is shown in Figure 15. Now we can easily see at each input amplitude and thus power, how large the non-linearities are and that the gap increases quickly with a lower input amplitude.

We can also see by the grey lines that if the linearity distortions need to be a certain amount away, such a the 80dB shown, we can achieve this with lower IP_2 and IP_3 requirements if we lower the maximum input amplitude. The IP_2 and IP_3 requirements lower the same amount as the maximum input power is lowered.

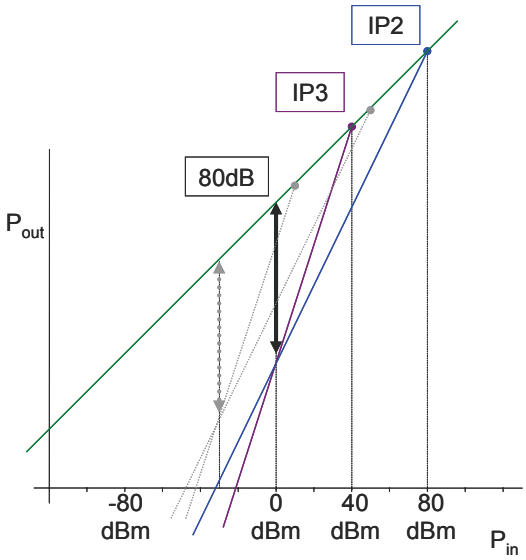


Figure 15. Non-linearities versus input and output power.

2.3.8.5 Interferers

As seen from Figure 14, two large signals can generate a lot of distortion products because of linearity distortions. If we are not interested in these large signals, we call them interferers because they can interfere with the signal of interest. From Figure 15 we found that larger signals cause larger distortion products. Since we limited our input amplitude range, the largest interferers can be 0dBm signals, resulting in the above IP_2 and IP_3 requirements.

If we could reduce the amplitude of the interferers for example with a filter, we would relax the linearity requirements as seen in Figure 15 and Figure 16. For example a 30dB reduction would give an IP_2 requirement of 54dBm and an IP_3 requirement of 27dBm for -84dBm distortions, which is significantly less.

Thus we would like to filter as early as possible, because only after filtering the linearity requirements are relaxed.

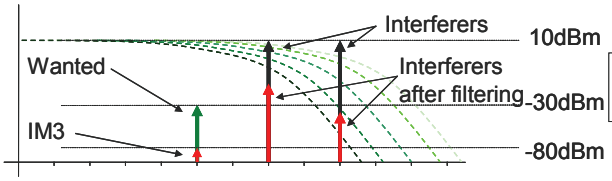


Figure 16. Suppresion of interferers.

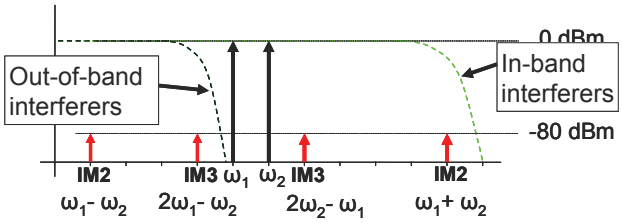


Figure 17. In-band and out-of-band interferers.

The resolution bandwidth filter selects the frequency of interest. We call interferers that fall in the filter bandwidth (light green filter in Figure 17) “in-band” interferers and interferers that fall outside of the filter bandwidth (dark green filter in Figure 17) “out-of-band” interferers.

In case of the resolution bandwidth filter of a spectrum analyser, we assume that everything inside the RBW filter is of interest and therefore, the in-band “interferers” are the wanted signals and not interferers. On the other hand, everything outside of the RBW filter can give distortions inside the RBW filter and is therefore unwanted. These out-of-band interferers however, can be arbitrary close to the filter to the point that they are only suppressed 3dB, which is normally the bandwidth boundary or cut-off frequency. Thus the linearity requirements can not be relaxed with the use of filtering unless a minimum distance that the interferers must be away is specified or that everything under the filter envelope is part of the signal of interest removing the linearity requirement after filtering completely. Since after the RBW filtering we assume that everything left is of interest we apply the latter, but keeping the former in the back of our heads. It is clear however that an extra specific filter just for relaxing linearity requirements is not an option in our case, because of the very large input frequency range and indeterminate frequency of interest.

2.3.9 Spurious free dynamic range

Generally dynamic range (DR) is defined as the ratio of the maximum input level to the noise floor (DANL). However in RF applications the maximum input level is limited by linearity distortions as we saw above. [4] Thus we define the spurious free dynamic range (SFDR) as the ratio between the maximum input level and the maximum resulting distortion products or equivalently as the ratio between the maximum input level without distortion products above the noise floor and the noise floor. This determines the smallest signal we can reliably detect next to a large signal. Thus another way of putting it is that the spurious free dynamic range of a spectrum analyser is defined as the largest power ratio between two signals that the analyser can measure to a specified degree of accuracy.

From now on, dynamic range is used as spurious free dynamic range unless otherwise indicated.

We identified several aspects that limit the spurious free dynamic range. As said, there are two factors limiting the SFDR; noise and linearity distortions.

The noise factor has two components. First there is the noise floor, which is dependant on the selected RBW and thus so is the DR. Second there is the phase noise from the local oscillator, which is not dependent on factors that can be influenced by the user of a spectrum analyser, but by design factors such as the Q and the local oscillator signal power.

The linearity distortions have two components; higher harmonics and intermodulation products. Second-order harmonics and intermodulation products combine into the IP_2 specification and third-order harmonics and intermodulation products into the IP_3 specification. Higher orders are neglected. With lower input levels, these factors fall under the noise levels, but at higher input powers they quickly become dominant.

The noise components can be added to Figure 15, where we set out the input power against the output power of a system for second and third order non-linearities. As the noise floor is fixed, it is not dependent on the power of the input signal, which means a straight line at the output power corresponding to the noise floor level. The phase noise is a specified amount below the signal power of the local oscillator carrier. When the local oscillator signal is multiplied with an input signal at the mixer, both the carrier power as well as the phase noise of the local oscillator signal are scaled by the input signal. The phase noise after mixing is therefore also the same amount below the output signal as it was for the local oscillator signal. Thus the phase noise power follows the input signal power with a fixed distance. The dynamic range is the range from the signal down to the first noise level or distortion level.

While the output power versus input power is very useful to show the dependence of the different factor on the input power, the dynamic range is a little less clear. Therefore we can normalise on the input power. The dynamic range can then be read directly from the y-axis.

With an input range of 80dB (-80dBm ~ 0dBm), we would like to have the complete input range distortion free. This means the noise and non-linearity products must stay below

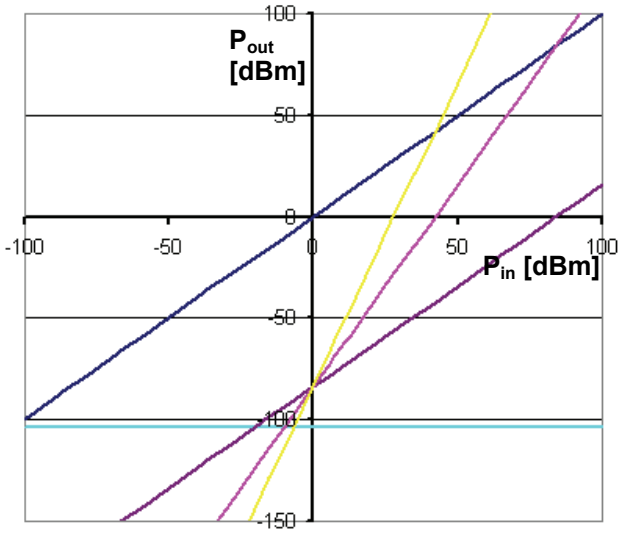


Figure 18. Output power versus input power at a 10kHz RBW.

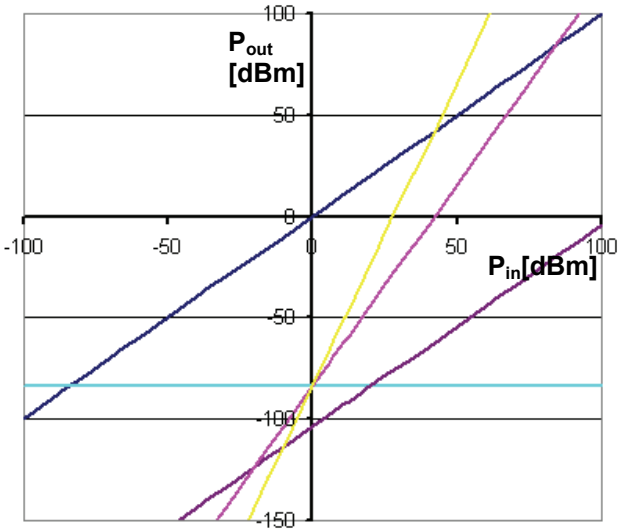


Figure 19. Output power versus input power at a 1MHz RBW.

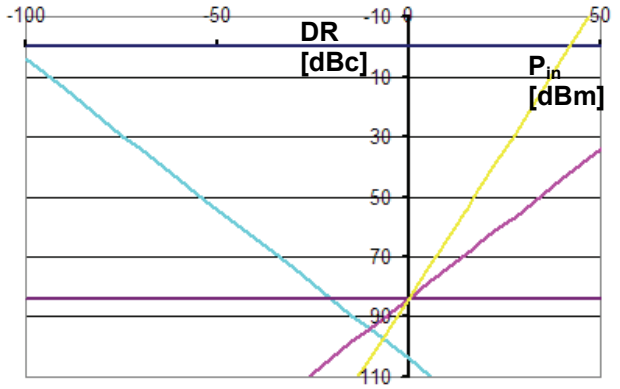


Figure 20. Dynamic range versus input power at a 10kHz RBW.

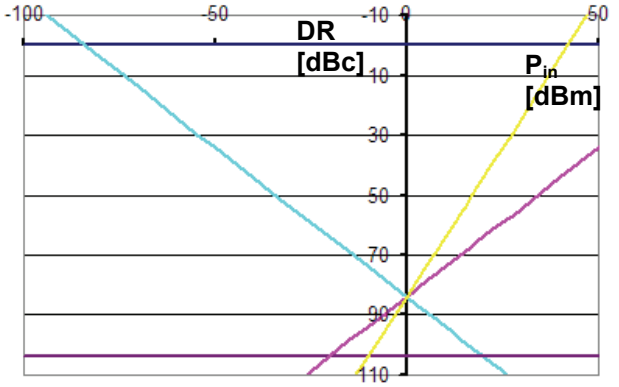
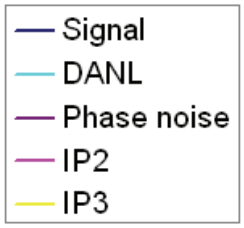


Figure 21. Dynamic range versus input power at a 1MHz RBW.

-84dBm at all resolution bandwidths. From the specifications found, Figure 18 shows the result for a DANL of -144dBm/Hz which corresponds to a total noise figure of the front-end of 30dB, a phase noise of -124dBc at 8kHz, an IP_2 of 84dBm, an IP_3 of 42dBm and a RBW of 10kHz. A RBW of 1MHz results in Figure 19. Note that with -124dBc at 8kHz and a phase noise with a $1/f^2$ dependence, the phase noise at 0.8MHz is -164dBc, while for the dynamic range the phase noise only needs to be -144dBc at 0.8MHz. Thus for lower RBWs, the maximum dynamic range is limited by the phase noise and for higher RBWs by the noise figure, while improving the linearity distortions helps both.



The dynamic range versus input power is shown in Figure 20 for a 10kHz RBW and in Figure 21 for a 1MHz RBW, which also makes the above mentioned limiting factors more obvious.

All the specifications were chosen so to keep limit the noise and distortions under the smallest input signal. Therefore our dynamic range automatically comprises of the complete input range and we will get a dynamic range of -80dBc (a difference of 80 dB down from the signal).

SFDR combines a number of specifications to provide an important indicator of the spectrum analyser performance. Overall, there are three important factors limiting the dynamic range;

- the overall noise floor of the system limited by the temperature and the used bandwidth,
- the phase noise of the local oscillator determined by its Q factor,
- and the linearity distortions of the system.

2.4 RF design parameter trade-offs

From the above specifications, we can identify an RF design hexagon [4] with specifications that can be traded against each other. Note that this is an RF design hexagon, because the spectrum analyser front-end is in essence an RF front-end. We will discuss the trade-offs in the context of the spectrum analyser here.

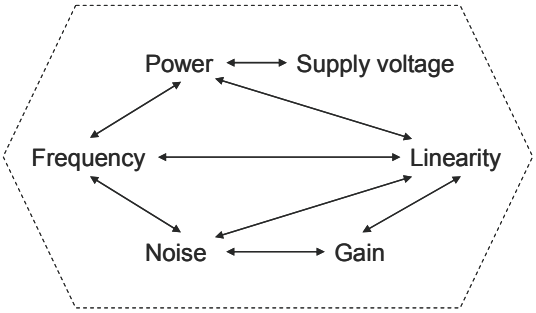


Figure 22. RF design hexagon

2.4.1 Frequency

The complete input frequency range must be supported by the first stages of the front-end requiring a large linearity, thus providing a trade-off between the highest frequency and required linearity.

The resolution bandwidth directly determines the noise floor. The highest RBW gives the highest noise floor and determines the minimum input signal.

2.4.2 Power

Power requirements can be traded against speed or highest supported frequency and linearity. For example making a differential circuit increases the power consumption, but hugely improves the linearity by lowering the even-order terms depending on matching.

Normally power consumption must be limited because of being part of a battery powered device or by the heat the device produces as well as of course environmental concerns. For now the device is not intended for battery use and the other requirements as well as integration requirements limit the spectrum analyser to about 10W. This is more than enough and can be used primarily to improve the linearity.

2.4.3 Supply voltage

The supply voltage is determined by the process used. For CMOS it can widely vary, but 1.5V ~ 3.3V is mostly used. This enables us to use signal up to around 10dBm (0.71V at 50Ω).

Lowering the voltage lowers the power consumption of the digital part of the design significantly, but also lowers the highest possible input signals. Since power should not be an issue and 10dBm should be enough, the supply voltage is also not an issue.

2.4.4 Gain

The gain trades with noise and linearity to determine the maximum dynamic range at a certain bandwidth. The dynamic range is an important specification of a spectrum analyser. If the gain is increased P_{max} increases and the IP_2 and IP_3 requirements change accordingly giving larger linearity distortion products. If the gain is decreased the input amplitude becomes closer to the noise floor and the dynamic range thus decreases.

However, if it is known that there is not a large frequency present in the input signal; gain can be applied to amplify smaller signals into the input range, as long as the largest signal does not become larger than P_{max}. Since any gain amplifies the signal as well as the noise, it can not improve the signal-to-noise ratio.

Attenuation can be used to lower larger input signals into the input range. Thus, we would like to scale the largest input signal down to the point with the highest dynamic range, while the highest dynamic range can be improved by lowering the RBW. With a lower RBW the noise floor lowers, increasing the DR, but only up to the point where the phase noise takes over. Note that with a smaller RBW, the highest DR point lies at a lower input level and the attenuation must be higher.

2.4.5 Noise

The noise floor determines the minimum input signal which can be used. For a certain wanted input amplitude range, this also determines the maximum input range. Lowering

the amplitude of the input signal severely lowers the linearity requirements depending on the order. For example, the IP_3 requirement goes down twice as fast as the input signal and the IP_5 four times as fast.

The identified noise figure of 30dB seems quite high as other spectrum analyser typically specify about 15dB (see section 2.5), we can lower the noise floor to ease the linearity requirements.

The phase noise depends on the local oscillator signal power, the frequency and its quality. The most difficult frequency is the highest frequency of the input range. The identified phase noise requirement of -124dBc/Hz at 8Hz offset is quite difficult [4]. To ease the requirement we can use a lower local oscillator signal power. This reduces the signal level after frequency conversion and thus effectively attenuates the signal trading with the gain parameter. However, with a switching mixer (section 3.2.3.3) the signal power only needs to be enough to drive the switches and does not scale the input signal. The other option is improving the quality of the oscillator, which is normally at the expense of more power usage [4].

When choosing a smaller RBW, the dynamic range improves because of the lower noise floor. However the phase noise requirement becomes more difficult with the next RBW and thus the frequency offset closer. Therefore the DR improvement by choosing a smaller RBW is limited by the phase noise.

2.4.6 Linearity

If we optimise the noise figure and the input range by the gain (or attenuation), the linearity determines the performance of the design. This means that at a certain input level the dynamic range is at its maximum, limited by the linearity distortions for higher input amplitudes and limited by noise for lower input amplitudes. Improving the linearity would then allow larger input amplitudes increasing the dynamic range.

Thus the linearity requirements trade with the noise figure requirement with the optimum input amplitude determined by both.

The identified linearity of IP2=84dBm and IP3=42dBm are very high compared to the linearity distortions of other spectrum analysers (see section 2.5), typical mixers (chapter 3) and filters (chapter 5). More typical specifications are IP2=20~40dBm and IP3= -5~10dBm. This means that in order to ease the linearity requirements we must improve the noise figure.

2.5 Comparison with existing spectrum analysers

To get a feeling of how the intended specification rate against available mid- and high-end spectrum analyser, we selected five other different spectrum analysers and compared them in a table. This is shown in Table 2.

2.5.1 Analyser analysis

We provide a small description of the spectrum analysers and discuss relevant specifications.

2.5.1.1 Bruco LCSA

The Bruco LCSA is a low-cost spectrum analyser with middle-class performance. It is designed to be highly integrated and to bridge the gap between direct digitisation and PC processing with an ADC at the low-end and general purpose middle class spectrum analysers at a much higher price. The Bruco LCSA tries to accomplish this with a simple low-cost design with limited input amplitude and frequency range.

The required distortion and phase noise characteristics are very high in comparison and therefore probably unattainable. This results in a dynamic range which is 10dB higher than the top class devices. The noise figure also seems a bit high. The noise figure and dynamic range can be lowered to easy the linearity requirements as mentioned before. We will derive a more balanced set of requirements in the next section.

	Bruco	Hameg	Tektronix	Rohde & Schwarz		Agilent
Type	LCSA	HM5012-2	RSA2203A Real-Time SA	R&S FSP	FSIQ Signal analyser	PSA
Segment	Low-cost (Integrated)	Mid-range	High-end, middle-class (time, spectrum, modulation)	High-end, middle-class (general purpose)	High-end, top class (time, spectrum, modulation)	High-end, top class (spectrum, modulation)
Price		€ 2,300	€ 21,000 (3GHz)	€ 17,000 (3GHz)	€ 65,000 (3GHz)	€ 43,000 (7GHz)
Frequency range	150kHz ~ 3/6GHz	150kHz ~ 1GHz	0/10MHz ~ 3/8GHz	9kHz ~ 3/7/13/30/40GHz	20Hz ~ 3/7/26GHz	3Hz ~ 7/13/27/44/50GHz
Resolution bandwidth (RBW)	10kHz ~ 1MHz (1Hz ~ 1MHz FFT)	9kHz, 120kHz, 1MHz	1Hz ~ 10MHz	10Hz ~ 10MHz (1Hz ~ 30kHz FFT)	1Hz ~10MHz (1Hz ~ 1kHz FFT)	1Hz ~ 8MHz (1Hz ~10MHz FFT)
Amplitude range (1MHz RBW)	-80dBm ~ 0dBm	-80dBm ~ 10dBm	-80dBm ~ 30dBm	-90dBm ~ 20dBm (30dBm with attn.)	-95dBm ~ 20dBm (30dBm with attn.)	-91dBm ~ 30dBm
Amplitude accuracy	< 2dB	3dB	1.2dB ~ 2.4dB	0.5dB (2dB~4dB > 3GHz)	1dB ~1.5dB (2.5dB > 7GHz)	0.62dB ~ 1.74dB (2.2dB > 7GHz)
Sweep time	1ms ~ 1000s	40ms, 320ms, 1s	20ms ~ 3200s	2.5ms ~16000s	5ms ~ 16000s	1ms ~ 2000s
Noise level (DANL)	-154dBm/Hz	-140dBm/Hz	-142 ~ -151dBm/Hz	-150dBm/Hz	-155dBm/Hz	-143 ~ -153dBm/Hz
Noise figure (NF)	30dB	*31dB*	*24dB*	*20dB*	15dB	*19dB*
Phase noise (1GHz or 3GHz)	-124dBc/Hz at 10kHz -144dBc/Hz at 1MHz		-105dBc/Hz at 10kHz -130dBc/Hz at 1MHz	-108dBc/Hz at 10kHz -118dBc/Hz at 1MHz	-115dBc/Hz at 10kHz -140dBc/Hz at 1MHz (-114/-132 > 7GHz)	-116dBc/Hz at 10kHz -145dBc/Hz at 1MHz
IP_2 (SHI)	> 84dBm	*48dBm* (-75dBc at -27dBm)	*65dBm* (-70dBc at -5dBm)	25dBm ~ 35dBm (80dBm > 1.5GHz)	25dBm ~ 45dBm	42dBm ~ 52dBm (80dBm >1.5GHz) (-82dBc ~ -92dBc at -40dBm)
IP_3 (TOI)	> 42dBm	*10dBm* (-75dBc at -27dBm)	*30dBm* (-74dBc at -7dBm)	5dBm ~ 10dBm (-70dBc ~ -80dBc at -30dBm)	12dBm ~ 17dBm	14dBm ~ 17dBm (-88dBc ~ -94dBc at -30dBm)
Dynamic range (SFDR) (1MHz RBW)	> 80dBc	*60dBc*	*70dBc* *(78dBc without phase noise)*	*58dBc* *(64dBc without phase noise)*	*70dBc*	*70dBc*

Table 2. Spectrum analyser comparison[6]

2.5.1.2 Tektronix RSA2203A

The Tektronix RSA2203A [23] is a high-end middle-class spectrum analyser, which adds transient triggering for real-time time-based signal analysis. It's list price is around €22,000 [17][27].

The Tektronix has good noise and linearity figures, but the dynamic range is limited by its phase noise. Note that the linearity outperforms all others by quite a margin (using a low-IF design followed by an ADC, see chapter 4 and 6).

2.5.1.3 Rohde & Schwarz FSP

The Rohde & Schwarz FSP [24] is a little cheaper than the Tektronix at around €17,000 [16][27] for the FSP3 with an up to 3GHz input frequency.

However, the linearity and phase noise are significantly worse giving a low dynamic range around the same as the much cheaper Hameg. Of course this is compensated for the larger functionality of the FSP.

2.5.1.4 Rohde & Schwarz FSIQ

The Rohde & Schwarz FSIQ [25] is a top class device with the 3GHz version (FSIQ3) at around €65,000 [16][27].

At this price the performance goes up quite a bit with much better phase noise especially at 1MHz for the 3GHz version. Also the noise figure is the lowest of all. The dynamic range is limited a bit by the linearity distortions, but with more typical IP_2 performance of 40dBm the specifications are well balanced.

2.5.1.5 Agilent PSA

The Agilent PSA [26] is also a top class device with better price (around €43,000 [15][27]) and a larger input frequency range (7GHz for the PSA E4443A).

The PSA even has a little bit better linearity and phase noise specifications. However, the noise figure is a little bit higher giving about the same dynamic range as the FSIQ and the Tektronix.

[6] Italic specification values are derived from other specifications

2.6 Summary and Conclusion

This chapter discussed what a spectrum analyser is, its basic operation and what it measures. Furthermore, it defines the specification for the low-cost spectrum analyser front-end design. The trade-offs between the specifications were revealed and discussed. The determined specifications are then set out against commercially available spectrum analysers.

2.6.1 Summary

A spectrum analyser is a device to measure the spectral content of an electrical, optical or acoustic waveform, in our case an electrical signal. The spectrum analyser often shows the power spectrum (the power in small bandwidths for a larger frequency range).

An analogue spectrum analyser uses a band-pass filter to select a small frequency range and determines the voltage or average power. A digital spectrum analyser uses a digital Fourier transform. For both, frequency conversion by a mixer is used to support a large input frequency range at a smaller intermediate frequency.

A spectrum analyser approximates a Fourier transform by giving the approximate PSD in a small selectable bandwidth, repeats this measurement for each frequency band in a range and provides the resulting graph. The selectable bandwidth is called the resolution bandwidth (RBW), since it determines the resolution (in frequency) of the measurement. Using the voltage (amplitude) instead of the power gives the amplitude spectral density.

Two sets of spectrum analyser specification are discussed. One set of specifications state the supported input signals and ranges. The second set determines the spectrum analyser's performance in terms of the supported spurious free dynamic range (SFDR).

The input amplitude and frequency range are chosen. The amplitude range defines the largest and smallest signals. The frequency range defines the minimum and maximum supported frequency. For the amplitude accuracy, we found that especially the frequency response of the filters is important. The resolution bandwidth determines the frequency resolution. More RBWs gives more flexibility but increases the number of needed filters and thus cost. A range of 10kHz to 1MHz in seven steps was chosen. The sweep time can be as low as possible. For normal operation a sweep time under a second is wanted for usability reasons. For a large range with a high resolution, the sweep time is allowed to go up further.

Noise and linearity distortions determine the maximum unwanted signal levels. Combined they give the SFDR.

The displayed average noise level (DANL) is determined by the selected RBW and the noise figure (NF). The NF is the noise the spectrum analyser itself adds. The phase noise only applies to the local oscillator. We found that the phase noise requirement is the most strict in the smallest RBW. The phase noise is dependent on the local oscillator signal power (P_c), the quality (Q) and the frequency (ω_c). The phase noise becomes especially problematic at high frequencies, since the needed noise attenuation at the offset for the smallest RBW stays the same for higher frequencies, which means the needed Q is larger. Therefore, it is useful to use low local oscillator frequencies when possible.

The linearity distortions are characterised in second-order IP_2 and third-order IP_3 specifications, with the order determined how fast the distortions grow for an increasing input signal.

For lower RBWs, the maximum dynamic range is limited by the phase noise and for higher RBWs by the noise figure, while improving the linearity distortions helps both.

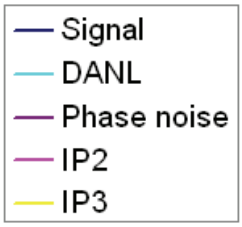
Compared to a number of available spectrum analysers, the required distortion and phase noise characteristics are very high. This results in a dynamic range which is 10dB higher than the top class devices. The noise figure also seems a bit high. The noise figure and dynamic range can be lowered to easy the linearity requirements.

2.6.2 Conclusion

The different types of analysers are grouped into two sets. Analogue spectrum analysers use a resolution bandwidth to select a frequency after which the signal is measured. This is repeated for the frequency band of interest one resolution bandwidth after the other. Digital spectrum analysers use a fast Fourier transform (FFT) to select a number of frequency bands at once and measure the signal.

For both groups it was found that in order to support a large input frequency range with a consistent measurement accuracy, an architecture with frequency conversion must be used, which translates a resolution bandwidth or a FFT bandwidth to a fixed intermediate frequency range. After measurement the next bandwidth is translated and the frequency range of interest is scanned through. Therefore this architecture is called a scanning architecture.

The next step was to derive a well-balanced, coherent and competitive set of specifications for a spectrum analyser front-end. The initial specifications were derived based on the wanted input frequency range and a spurious free input amplitude range. Examination of the relationships between the specifications revealed mainly trade-offs between power, noise and linearity. The dynamic range combines these trade-offs to give an performance indicator and shows how the thermal noise, phase noise and second- and third-order linearity distortions can be balanced to give the optimal spurious free input amplitude range over the complete frequency range.



The initially derived specifications were found to be quite ambitious in comparison with a selection of middle-class to high-end available commercial analysers. Although starting with a somewhat lower frequency and amplitude input range, a dynamic range of 80dB is 10dB higher than what the top-class devices manage at 1MHz. This gives high phase noise and linearity requirements which are probably unattainable.

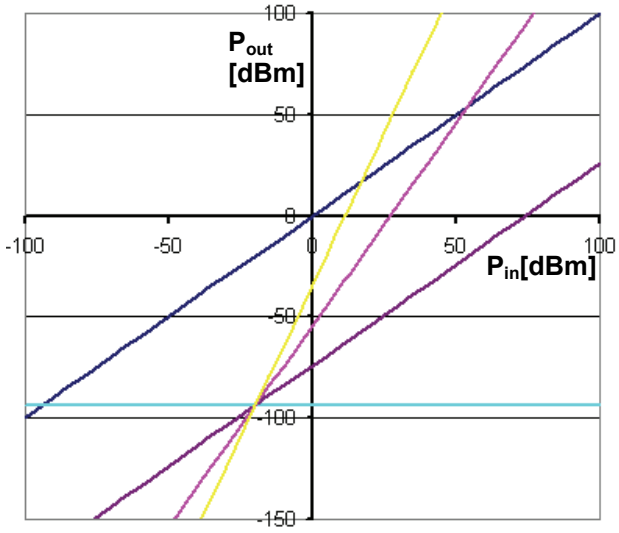


Figure 23. Output power versus input power at a 1MHz RBW and recommended specifications.

A better comprise is found if we lower the dynamic range to 70dBc at an input level of -20dBm. The lower dynamic range eases the phase noise requirement with 10dB as well as easing the linearity requirements. Also, the noise figure requirement is made somewhat stricter at 20dB to ease the linearity requirements. Together, this gives more manageable linearity requirements of IP_2=54dBm and IP_3=17dBm. The phase noise is now specified at -114dBc with a 10kHz RBW and -134dBc for a 1MHz RBW. The output power and dynamic range against the input power for these specifications are provided in Figure 23 and Figure 24 respectively.

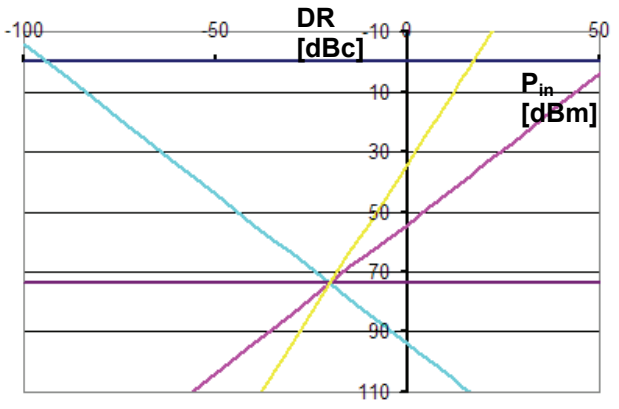


Figure 24. Dynamic range versus input power at a 1MHz RBW and recommended specifications.

The input amplitude range is now 70dB from -90dBm to -20dBm. Higher amplitudes can be allowed optionally, but they introduce spurious responses. For example, a -10dBm has only a 50dBc dynamic range. The linearity distortions are located only at certain frequencies. The location of the distortion products can be determined from the location of the largest signals if they are known. Therefore a larger dynamic range can be allowed for the other frequencies. A much larger input range is not recommended because it must still be measured, which increases the ADC requirements (see chapter 6). However, a problem of this approach is that the device-under-test that is analysed by

the spectrum analyser also typically includes linearity distortion measurements. If inputs larger than -20dBm are allowed, it is not clear if the spurious responses are from the spectrum analyser or the measured device, something a user might not initially recognise. Therefore we will choose to limit the input signal at -20dBm.

A larger or smaller input signal can be attenuatiated or amplified so it falls in the -90 ~ -20dBm range. The presented input range is thus without amplification. With amplification a much larger can be supported, but the dynamic range is still 70dBc.

Note that these recommended requirements are still quite ambitious and therefore competitive compared to the high-end devices available.

In the following table the specifications of a typical middle-class to top-class spectrum analyser derived from section 2.5 are listed against the specification provided as an indication at the start of the project (Bruco First), the specification derived for a distortion free -80dBm ~ 0dBm input range (Bruco Second) and the above recommended specifications.

	Typical	Bruco (First)	Bruco (Second)	Recommended
Type		LCSA v1	LCSA v2	LCSA v3
Segment	Middle-class ~ Top-clas	Low cost, flexible (Integrated)	Low cost, flexible (Integrated)	Low cost, flexible (Integrated)
Frequency range	0 ~ 3/7GHz	100kHz ~ 3GHz	150kHz ~ 3/6GHz	150kHz ~ 3/6GHz
Resolution bandwidth (RBW)	1Hz ~10MHz (1Hz ~ 1kHz FFT)	10kHz ~ 10MHz	10kHz ~ 1MHz (1Hz ~ 1MHz FFT)	10kHz ~ 1MHz (1Hz ~ 1MHz FFT)
Amplitude range (1MHz RBW)	-80dBm ~ 20dBm	-70dBm ~ 10dBm	-80dBm ~ 0dBm	-90 ~-20dBm
Amplitude accuracy	1db ~ 2dB	< 2dB	< 2dB	< 2dB
Sweep time	10ms ~ 5000s	0.1ms ~ 1s	1ms ~ 1000s	1ms ~ 1000s
Noise level (DANL)	-150dBm/Hz	-144dBm/Hz	-154dBm/Hz	-154dBm/Hz
Noise figure (NF)	20dB	30dB	30dB	20dB
Phase noise	-110dBc/Hz at 10kHz -130dBc/Hz at 1MHz		-124dBc/Hz at 10kHz -144dBc/Hz at 1MHz	-114 dBc/Hz at 10KHz -134dBc/Hz at 1MHz
IP_2 (SHI)	40 dBm	0dBm (-60dBc at -30dBm)	> 84dBm	54dBm
IP_3 (TOI)	10 dBm	-15dBm (-60dBc at -30dBm)	> 42dBm	17dBm
Dynamic range (SFDR) (1MHz RBW)	60dBc ~ 70dBc	60 dB	> 80dBc	70dBc

Table 3. Spectrum analyser specifications recommendation

Chapter 3

3 Architecture

In section 2.1, we found that the front-end of a spectrum analyser consists of two major building blocks, the frequency selection and the signal measurement. This is repeated in Figure 25. In section 2.2, we compared different spectrum analyser solutions and found that in order to support our required input frequency range, we need a frequency converting architecture such as a super-heterodyne architecture, repeated in Figure 26. The super-heterodyne architecture primarily defines a way to select a frequency (the first major block), while the measurement and following blocks can also be implemented differently.

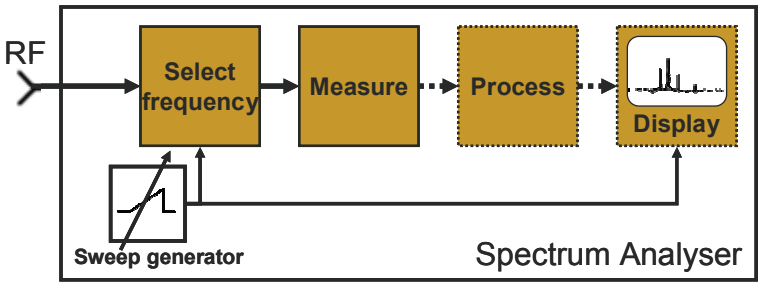


Figure 25. Spectrum analyser composed of functional blocks

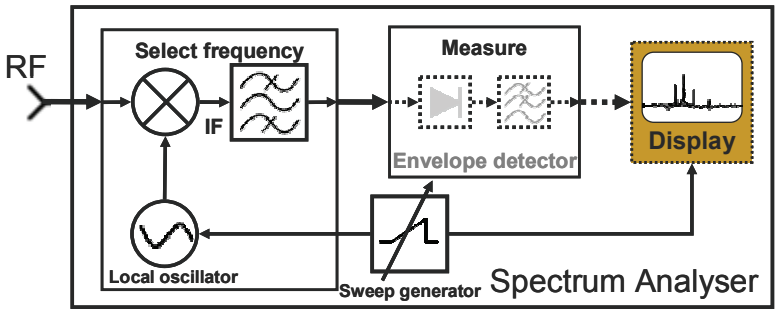


Figure 26. Super-heterodyne spectrum analyser

This chapter will further divide the two major blocks into sub-parts, together with a number of optional front-end parts that were already briefly touched upon, and discuss them one by one in section 3.2. Before that, it is discussed how the specifications can be divided over cascaded blocks. This clarifies how we can translate the requirements of the front-end to requirements of the mostly cascaded sub-blocks of the front-end. We will conclude with the relevant blocks to be further discussed in later chapters

3.1 Cascaded blocks

In chapter 2 we identified two sets of specifications for a spectrum analyser front-end. One set of specifications state the supported input signals and ranges. The second set determines the spectrum analyser's performance in terms of the supported spurious free dynamic range. These factors are required of the overall front-end and must now be translated into requirements of the sub-blocks.

The input amplitude and frequency range are chosen. The amplitude range defines the largest and smallest signals for the sub-blocks. It can be scaled up en down by amplification or attenuation, but the size of the range must stay the same. The frequency range dictates the need for a frequency conversion and thus influences the architecture selection. For the amplitude accuracy, we found that especially the frequency response of the filters is important. This is because the noise and linearity distortions are much smaller than most input signals (except for the smallest) and therefore negligible. The wanted resolution bandwidths specify the required filters. The local oscillator settling time and the slowest transient response of the filters adds up with the number of RBWs in a sweep to give the minimum sweep time.

The amplitude range and accuracy determine the required linearity and noise at a certain RBW. The linearity and noise are relevant to most of the sub-blocks, as will follow from the next section. Thus it is instructive to discuss the combined result of these specifications from a number of cascaded sub-blocks (stages).

First we will treat the noise. We found that phase noise only applies to the frequency generation of the local oscillator and thus we will only discuss the noise figure of a cascade of stages.

The noise figure adds to the noise floor to become the noise floor for the next stage. Amplification amplifies both the signal and the noise floor, we can intuitively see that if a stage amplifies the signal it also "amplifies" the noise figure of previous stages. If we refer this back to the input, this results in [4]:

(3-1) $$NF_{tot} = 1 + (NF_1 - 1) + \frac{NF_2 - 1}{A_{v1}{}^2} + \ldots + \frac{NF_m - 1}{A_{v1}{}^2 \cdot \ldots \cdot A_{v(m-1)}{}^2}$$

This is called the Friis equation [4]. The -1 results because the definition of noise figure includes the ideal transfer of the signal-to-noise ratio from input to output (i.e. 1), while the rest is the amount the stage itself reduces the signal-to-noise ratio. The A_v is squared because the noise figure is a power ratio.

The preceding equation indicates that the noise figure of the first few stages is the most critical as it is "amplified" by the following stages. Or consequently if we have attenuation, the noise figure of the last few stages is the most critical as the noise figure of the preceding stages is "attenuated".

The supported input amplitude range is 70dB, with the maximum signal and the optimum dynamic range at -20dBm. It was already found that amplifying the input signal provides no advantages with respect to noise. The input range was specified to have the lowest signal levels with an 80dB range to ease the linearity requirement. The smallest signal is limited by the DANL. Therefore, the signal can not be further attenuated without losing the smallest signals. Thus when the input signal is in the input range from -90dBm to -20dBm, the amplification factors of the following blocks can be taken as 1, and the noise figure of the stages all simply add together.

The same result holds for the amplitude accuracy.

For the linearity distortion products of two cascaded stages we find [4]:

(3-2) $$\begin{aligned} y(t) &= \alpha_1 x(t) + \alpha_2 x^2(t) + \alpha_3 x^3(t) + \ldots \\ z(t) &= \beta_1 y(t) + \beta_2 y^2(t) + \beta_3 y^3(t) + \ldots \\ z(t) &= \beta_1\alpha_1 x(t) + \left[\beta_1\alpha_2 + \beta_2\alpha_1{}^2\right]x^2(t) + \left[\beta_1\alpha_3 + \beta_2 2\alpha_1\alpha_2 + \beta_3\alpha_1{}^3\right]x^3(t) + \ldots \end{aligned}$$

Here and following we will neglect the fourth and higher order terms. Note that the second and third order amplification factors are typically much smaller than α_1.

For the amplitude of two input signals with A_1=A_2, we have for the wanted terms:

(3-3) $$x(t) = A_1 \cos\omega_1 t + A_{\ 2}\cos\omega_2 t$$

(3-4) $$\omega_1, \omega_2 : \left(\alpha_1 A_1 + \frac{3}{4}\alpha_3 A_1{}^3 + \frac{3}{2}\alpha_3 A_1 A_2{}^2\right) = \alpha_1 A + \frac{9}{4}\alpha_3 A^3$$

For the amplitude of the second order terms we found:

(3-5) $$\begin{aligned} 2\omega_1, 2\omega_2 &: \frac{1}{2}\alpha_2 A^2 \\ \omega_1 \pm \omega_2 &: \alpha_2 A^2 \end{aligned}$$

And for the amplitude of the third order terms we found:

(3-6) $$\begin{aligned} 3\omega_1, 3\omega_2 &: \frac{1}{4}\alpha_3 A^3 \\ 2\omega_1 \pm \omega_2,\ \omega_1 \pm 2\omega_2 &: \frac{3}{4}\alpha_3 A^3 \end{aligned}$$

With $\alpha_1 \gg 9/4\ \alpha_3$ and if we take the largest term, we find for the amplitude at which the second order products is as large as the wanted term (the IP_2 point):

(3-7) $$A_{IP2} = \sqrt{2\left|\frac{\alpha_1}{\alpha_2}\right|}$$

And for the largest third order product:

(3-8) $$A_{IP3} = \sqrt{\frac{4}{3}\left|\frac{\alpha_1}{\alpha_3}\right|}$$

If we combine these with the result for the cascaded stages, we find:

(3-9) $$A_{IP2} = \sqrt{2\left|\frac{\alpha_1\beta_1}{\beta_1\alpha_2 + \beta_2\alpha_1^{\,2}}\right|}$$

(3-10) $$A_{IP3} = \sqrt{\frac{4}{3}\left|\frac{\alpha_1\beta_1}{\beta_1\alpha_3 + \beta_2 2\alpha_1\alpha_2 + \beta_3\alpha_1^{\,3}}\right|}$$

We can simplify these by squaring and inverting to find:

(3-11) $$\frac{1}{A_{IP2}^{\,2}} = \frac{1}{2}\frac{\beta_1\alpha_2 + \beta_2\alpha_1^{\,2}}{\alpha_1\beta_1} = \frac{1}{A_{IP2,x}^{\,2}} + \frac{\alpha_1}{A_{IP2,y}^{\,2}}$$

(3-12) $$\frac{1}{A_{IP3}^{\,2}} = \frac{3}{4}\frac{\beta_1\alpha_3 + \beta_2 2\alpha_1\alpha_2 + \beta_3\alpha_1^{\,3}}{\alpha_1\beta_1} = \frac{1}{A_{IP3,x}^{\,2}} + \frac{3\beta_2\alpha_2}{2\beta_1} + \frac{\alpha_1^{\,2}}{A_{IP3,y}^{\,2}}$$

With $\beta_1 \gg \alpha_2\beta_2$, we can generalise these to:

(3-13) $$\frac{1}{A_{IP2}^{\,2}} = \frac{1}{A_{IP2,1}^{\,2}} + \frac{\alpha_1}{A_{IP2,2}^{\,2}} + \frac{\alpha_1\beta_1}{A_{IP2,3}^{\,2}} + \ldots$$

(3-14) $$\frac{1}{A_{IP3}^{\,2}} = \frac{1}{A_{IP3,1}^{\,2}} + \frac{\alpha_1^{\,2}}{A_{IP3,2}^{\,2}} + \frac{\alpha_1^{\,2}\beta_1^{\,2}}{A_{IP3,3}^{\,2}} + \ldots$$

We would like A_{IP2} and A_{IP3} to be a large as possible. This means that for stages with amplification ($\alpha_1,\beta_1>1$) that the influence of the later stages is “amplified” by the gain. This is intuitively clear as the input signals are increased with gain, increasing the linearity distortions to the second or third power. With attenuation the first few stages are the most critical. Again in our case we have no gain ($\alpha_1,\beta_1=1$) and the non-linearity distortions of the stages simply add up.

The NF is in dB and the linearity distortions are not. Thus the NF adds in dB as the signal passes through the blocks and the linearity distortions add normally. If we assume large signals are relatively scarce, large distortions are also. So it is unlikely to have multiple distortions at one frequency becoming larger than specified dynamic range allows, as long as all block have a somewhat better linearity than the total front-end.

3.2 Spectrum analyser building blocks

The functional blocks are further subdivided. For the frequency selection block we already introduced the super-heterodyne architecture to support a large input frequency range. We also introduced the resolution bandwidth filter which selects a “single” frequency. Thus, we can subdivide the frequency selection block into a frequency conversion and a resolution bandwidth selection part. First however, two optional parts are introduced. To extend or relocate the input amplitude range an optional amplifier/attenuator can be used, as was already mentioned before. The second part is a low-pass filter to remove frequencies higher than the supported input frequency range. This results in the sub-division of Figure 27. Finally the measure block is discussed.

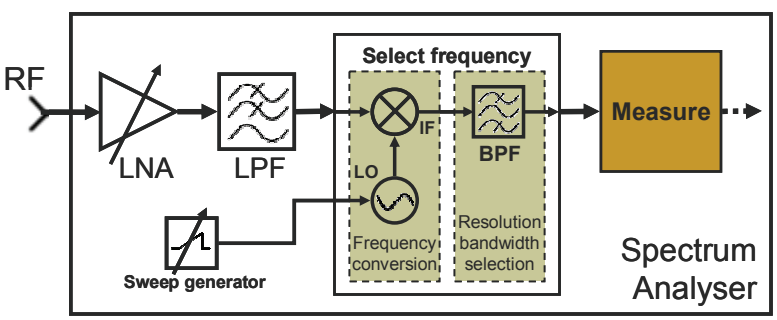


Figure 27. Spectrum analyser front-end sub-division

For each sub-block we will discuss the relevant specifications and their consequences for the overall front-end.

3.2.1 Low noise amplifier or attenuator

As became evident from the derivation of the specifications, it is sometimes necessary to amplify/attenuate a signal to the optimum level with the highest dynamic range or to amplify/attenuate the signal enough so it falls within the amplitude input range. Thus, it prevents overload and distortion. It was also found that amplifying the signal also increases the noise level with at least the same amount, while it also makes the distortion products larger to a power of the amplitude. Therefore, it is best to amplify the signal only up to the point that the largest signals stay below -20dBm so as to not introduce linearity distortions, as long as it is the amplification is sufficient for the signal of interest to fall within the input range.

After the low noise amplifier/attenuation (LNA), the signal amplitude range is conditioned to the specified amplitude range. The discussion of section 3.1, about dividing the specifications over the different block then applies. The implications of a wider input amplitude range on the specifications of the LNA and the influence on the specifications of the rest of the front-end blocks are not further analysed. However, it can be mentioned that as the LNA is the first block, that in case of the amplification the NF is critical and in case of attenuation the linearity is critical. Thus the LNA must at least (more for larger inputs) achieve an IP_2 performance of over 54dBm and an IP_3 performance of over 17dBm for the complete input frequency range. We look at the noise figure from the perspective of the signal after the LNA. For the conditioned input after the LNA, the LNA is assigned a noise figure of 2dB (see section 3.2.5). This is thus separate from possible noise and distortion already in the input signal before the LNA and which are thus amplified or attenuated by the LNA (which is inevitable).

Typical LNA characteristics ([4], p.167) are a noise figure of 2dB and an IP_3 of -10dBm to 10dBm [32], the latter therefore being a problem. As the LNA falls outside of the scope of this thesis, we will not go into further detail.

3.2.2 Low-pass filter

It will become clear from section 3.2.3 that in order to achieve frequency conversion, one must perform a multiplication of two sinusoids. As a consequence frequency components are up and down converted. This means that frequencies higher than the frequency input range can be translated into the frequency input range. These frequency components can interfere with the signal we want to measure. In that case it is useful to filter these higher frequencies with a low-pass filter (LPF). As with the LNA it must adhere to all the specifications set. We allow the frequency response to distort the amplitude accuracy by ±0.5dB. The mixer will take a significant portion of the NF, thus for the filter we would like a low NF of 2dB. The filter requirements are further discussed in chapter 5. Note that this filter is placed after the LNA, to ensure the input amplitude range is defined and limited.

3.2.3 Frequency conversion

Section 2.2 shows, that we need a frequency conversion to support a large frequency input range. This is because most parts are easier to design with the requested specifications at a lower frequency. Even if they could support the input range as defined now, we could greatly enlarge the input range by moving even higher frequencies into the supported range, which would make the design more competitive.

Frequency conversion of sinusoids is done by multiplication with a single (pure) sine:

$$x_1(t) = A_1\cos(\omega_1 t + \varphi_1) = A_1\left[\frac{e^{j\varphi_1}e^{j\omega_1 t} + e^{-j\varphi_1}e^{-j\omega_1 t}}{2}\right]$$

$$x_2(t) = A_2\cos(\omega_2 t + \varphi_2) = A_2\left[\frac{e^{j\varphi_2}e^{j\omega_2 t} + e^{-j\varphi_2}e^{-j\omega_2 t}}{2}\right]$$

(3-15)

$$x_1(t)x_2(t) = \frac{A_1A_2}{2}\left[\begin{array}{l} e^{j(\varphi_1+\varphi_2)}e^{j(\omega_1+\omega_2)t} + e^{-j(\varphi_1+\varphi_2)}e^{-j(\omega_1+\omega_2)t} \\ \qquad + e^{j(\varphi_1-\varphi_2)}e^{j(\omega_1-\omega_2)t} + e^{-j(\varphi_1-\varphi_2)}e^{-j(\omega_1-\omega_2)t} \end{array}\right]$$

$$x_1(t)x_2(t) = \frac{A_1A_2}{2}\left[\cos\left((\omega_1+\omega_2)t + (\varphi_1+\varphi_2)\right) + \cos\left((\omega_1-\omega_2)t + (\varphi_1-\varphi_2)\right)\right]$$

Thus, the results are the sum and difference of the original frequencies. If ω_{RF} is the input signal, we can move it up or down by ω_{LO}, which is called the local oscillator frequency. The resulting frequency is the intermediate frequency (ω_{IF}). Note that not only the frequency determines the resulting signal, but also the phase relation. Section 4.2 will go further into this. Also the intermediate frequency is not yet specified. The selection and resulting trade-offs are discussed in chapter 4.

3.2.3.1 Sweep generator

The sweep generator drives the local oscillator over the selected frequency range that must be measured. It determines which frequency the local oscillator must generate, which in turn determines which input frequency is converted to the intermediate frequency.

The sweep generator normally generates a saw-tooth signal conditioned to have the local oscillator generate an increasing frequency starting from the lowest to the highest in the selected frequency band. It is therefore said that the input signal is swept through the resolution bandwidth filter. For a continuous sweep the resulting signal at the intermediate frequency is measured at a fixed interval after which one RBW has passed through the filter (with a linear sweep).

With a linear sweep the input frequency that is converted to the intermediate frequency is constantly changing. Because the RBW filter needs time to settle, it does not accurately represent the signal in one RBW, especially if the signal is changing fast (having distinct frequency components). A better result is achieved with a stepping sweep. The sweep generator than steps through the selected frequency range one RBW at a time. After a step, it waits for the filters to settle after which the signal is measured and the sweep generator takes the next step. From now on, we will assume the sweep generator is stepping unless otherwise indicated.

Because the sweep generator only drives the local oscillator, there are no specifications to adhere to other than correctly generating the driving signal. If this signal is not stable, the local oscillator will generate a slightly varying frequency, which corresponds to phase noise.

3.2.3.2 Local oscillator

The local oscillator generates the converting frequency. While the design of the local oscillator falls outside of the scope of this thesis, the implications of the local oscillator are sometimes discussed. For example, the phase noise requirement directly applies to the local oscillator.

One problem with the local oscillator (LO) originates if the local oscillator frequency is not a pure sinusoid. Making a tuneable local oscillator that produces a pure sinusoid over a large range is difficult. If the generated signal is not a pure sinusoid, it contains higher harmonics and it is as if you are multiplying with multiple frequencies. This means that a number of frequencies (one for each harmonic) are being translated to the intermediate frequency resulting in distortion. The local oscillator must therefore preferably be designed so that the higher harmonics are at least 74dB below the fundamental (the specified spurious free dynamic range plus 4dB of headroom, taking the component resulting from LO higher harmonics as spurious distortions).

As the local oscillator is a source, it must provide a range of frequencies without higher order distortions, but the other noise and linearity requirements do no apply as the input signal does not pass through this block.

3.2.3.3 Mixer

Multiplication of signals is done with a mixer, which must support the input frequency range as well as a possibly higher local oscillator range. After the mixer, the frequency range is limited to the intermediate frequency.

The frequency conversion block is another stage in the spectrum analyser front-end adding noise and linearity distortion products. A typical mixer has a noise figure of about 12dB and an IP_3 at 5dBm ([4], p.181), again lower than the required 17dBm. For a good state-of-the-art mixer 17dBm might be possible (not easy), otherwise the dynamic range

must be allowed to reduce. A little lower noise figure of 10dB allows some room for the other blocks as it is already halve of the allowed noise figure.

The frequency conversion discussed above assumes a linear multiplier. Such a mixer is difficult to implement with these specifications. Therefore the mixer is normally implemented as a switching circuit, which connects the input directly with the output for half of the frequency period and a crossed with the output for the other half. If this switching has a “hard” on-off behaviour, it is as if you are mixing with a square wave no matter what you drive the switches with. Thus in that case the local oscillator signal (which drives the switches) might as well be a square wave also, as this also keeps the time of switching well defined. However, in that case the higher harmonics are large, which is problematic for the SFDR.

Thus if the frequency conversion must be linear, the local oscillator must generate a pure sinusoid and the mixer must be linear (for example with “soft” linear switches that track the local oscillator sinusoid for their transfer, which means the input is connected both straight and crossed part of the time).

The design of the local oscillator and the problem of higher harmonics are outside the scope of this thesis because of time constraint. But, this remains a significant problem.

3.2.4 Frequency selection

The frequency of interest is translated to the intermediate frequency, but all other input frequencies are also translated to frequencies around the intermediate frequency. As explained in section 2.3.6, we would like to filter everything but a very small bandwidth, conceptually representing one frequency. Note that this bandwidth is selectable and determines the resolution of the measurement. In case of a vector signal analyser, the intermediate frequency actually consists of two overlapping frequency ranges later to be divided into two separate ranges and different frequencies in the digital domain.

3.2.4.1 Resolution bandwidth filter

After the resolution bandwidth filter (which is a band-pass filter (BPF) for reasons that will become clear in chapter 4), the other frequencies are severely attenuated. Since the RBW bandwidth is conceptually one frequency, everything in this bandwidth is the signal of interest and if there are actually two signals with modulation products in this bandwidth is not of interest. Thus, in-band interferers are not interferers but signals of interest. As the other frequencies are now out-of-band interferers which are attenuated, their contribution to the distortion products is limited depending on the amount of attenuation, which in turn depends on how far away they are.

As an estimate let’s say the RBW filter may add 2dB of noise. With a passive implementation there are no linearity problems (linear system), otherwise the filter must also have an IP_2 of over 54dBm and an IP_3 of over 17dBm. For the low-pass filter we allowed the frequency response in the pass-band to vary ±0.5dB at max adding to the amplitude inaccuracy. For the resolution bandwidth filter, the selectivity of the filter is important as it separates two frequencies next to one another. As will become clear in chapter 5, the selectivity can be improved if we allow a larger frequency response variation. Therefore we would like a maximum of ±1dB of frequency response variation in the pass-band for the resolution bandwidth filter. The properties of the resolutions bandwidth filter are also further discussed in chapter 5.

3.2.5 Signal measurement

After an input frequency is translated to the intermediate frequency and all other frequencies are filtered away, we would like to measure the amplitude of that signal. Of the 20dB specified for the noise figure, we assigned 10dB to the mixer and 2dB to the LNA, the LPF and the RBW filter each. Thus we have about 4dB left for the signal measurement and following stages.

As explained above, the out-of-band interferers are filtered away relaxing the linearity requirements. While non-lineairities can no longer add interference of out-of-band interferers to the signal of interest after the RBW, the signal amplitude can be distorted. If two large in-band signals generate an intermodulation product at the position of the

largest signal in the RBW, instead of the amplitude of this largest signal, the amplitude plus the intermodulation product is measured. Since we would like a maximum amplitude distortion of less than 2dB, the distortion must be smaller. We assigned ±0.5dB of maximum amplitude inaccuracy to the LPF and ±1dB to the RBW filter, which means we have ±0.5dB left for the measurement block and following stages. If we take for example 20dB of separation between the maximum signal in the RBW and the in-band distortions generated by two other large signals (interferers), then the non-linearity distortions will influence the amplitude accuracy negligibly (1%) leaving the ±0.5dB (about 12%) for other causes of inaccuracy. With a maximum signal of -20dBm and interferers of also -20dBm, this means an IP_2 requirement of 0dBm and IP_3 of -10dBm for the following stages. This is reasonable and if achieved, the amplitude distortion is negligible.

Note that after measurement we can correct for the amplitude changes introduced in the preceding stages, as long as they are known and consistent (for example by calibration).

The division of the measurement block into sub-parts as well as a comparison between analogue measurement and digital measurement can be found in chapter 6.

3.3 Summary and Conclusion

This chapter discusses the division of the spectrum analyser front-end into sub-parts. The specifications are also divided over the blocks when relevant.

3.3.1 Summary

First we discussed how the front-end requirements are translated to requirements of the sub-parts. The specifications were separated into two sets and are treated one by one.

The amplitude input range is the same for all blocks the input signal passes through. This range can be amplified or attenuated, but we found that after conditioning the input signal with a low noise amplifier or attenuator, we would like this range to be fixed at -90dBm ~ -20dBm with a 1MHz RBW. The frequency range determines the need for a frequency conversion block. Up to this block the complete frequency range must be supported by every block. After this block, only the frequency range of the selected resolution bandwidth filter must be supported. The amplitude accuracy is largely influenced by the filters, but other blocks also diminish it. Finally the sweep time is determined by the stabilisation time of each block, which is mainly defined by the transient response of the filters and the settling time of the local oscillator.

The second set includes the noise and linearity. The phase noise is only relevant to the local oscillator, but the noise figure and linearity are relevant to all the blocks the input signal passes through. It is found that with amplification between the blocks, the first stages are most critical for the noise and the last stages for the linearity. For attenuation it is the other way around. In our case we have no amplification or attenuation between the blocks and the noise and linearity figures all add up equally.

The low noise amplifier/attenuator scales the input amplitude range to a defined range. The following low-pass filter filters out frequencies higher than the input frequency range so they can not interfere with the rest of the front-end. The input signal is than passed through the frequency conversion block. It is multiplied by the frequency generated by the local oscillator, which is driven by the sweep generator. The sweep generator must step through the selected frequency range with RBW sized steps in a saw-tooth way. This allows the filters to stabilise their response. The resulting signal that drives the local oscillator to generate a certain frequency must be stable otherwise the frequency will not be stable. The local oscillator itself must also make sure the generated frequency is correct and stable. If it is not, this results in phase noise. Other specifications do not apply. The multiplication is done with a mixer. The linearity of this combination is problematic as higher harmonics can also convert to the intermediate frequency distorting the signal. Conceptually “one” frequency is then selected by the RBW filter after which it is measured.

Next the specifications for the sub-parts were defined. Concerning the frequency range, the complete range must be supported from the input up to the frequency selection after which only the RBW must be supported. The amplitude input range is first scaled by the LNA to the -90dBm ~ -20dBm range. After the LNA this range must be supported by all blocks the input signal passes through at least up to the measurement. The amplitude accuracy of 2dB is divided among the low-pass filter, the resolution bandwidth filter, the non-linearity after the filter and the other blocks. The first was assigned ±0.5dB, the second ±1dB and the third should add a negligible amount. Therefore the rest is allowed ±0.5dB of amplitude inaccuracy.

For the noise, it was found that the phase noise applies to the sweep generator and local oscillator. The noise that adds to the input signal is divided over the blocks the input signal passes through. Because of its switching operating, the mixer is normally noisy and is allocated 10dB of the 20dB of noise figure. The low noise amplifier and both filters are each assigned 2dB of noise figure. This leaves 4dB for the measurement block and following stages. Concerning the linearity, this is divided equally over the block the input signal passes through up the frequency selection, each thus needing an IP_2 of over 54dBm and an IP_3 of over 17dBm. Although it is noted, that with a passive filter implementation this is most likely not a problem leaving less block to divide it over. After the frequency selection, out-of-band interferers giving linearity distortions are all filtered out. In-band interferers distort the amplitude accuracy, but with an IP_2 of 0dBm and an IP_3 of -10dBm this is negligible (1%).

The low noise amplifier/attenuator, the sweep generator and the local oscillator is not discussed any further throughout this thesis because of time constraints. The frequency conversion architecture is further discussed in chapter 4. The low-pass and resolution bandwidth filters are further discussed in chapter 5 and the measurement of the signal is further discussed in chapter 6.

3.3.2 Conclusion

The supported input amplitude range is 70dB, with the maximum signal and the optimum dynamic range at -20dBm. Amplification of this range is not possible because of linearity distortions. Attenuation only lowers the dynamic range. After the input signal is conditioned by the LNA to correspond to these boundaries, no amplification or attenuation is applied for the following blocks. The amplitude range is therefore the same for each block. In that case, the NF and linearity distortions of each stage add up equally to the total NF and linearity distortion of the complete front-end.

The “select frequency” block is sub-dived into a frequency conversion and frequency selection block. The frequency conversion consists of a mixer and a local oscillator, which is multiplied with the input signal. A pure sinusoid LO is assumed with a linear mixer. The problem of higher harmonics is left as future work. The frequency selection consists of the RBW filter.

The division of specifications over the block is indicated in Figure 28. The amplitude accuracy is indicated by AA and the phase noise by PN.

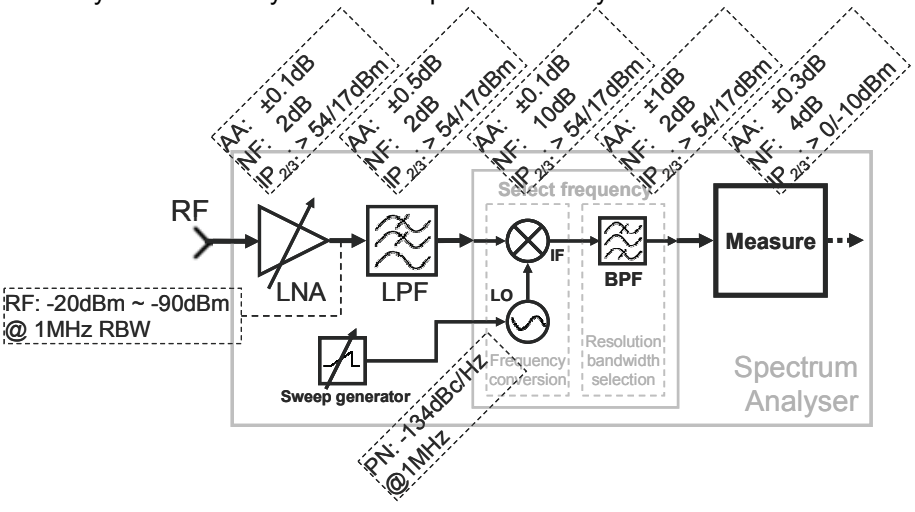


Figure 28. Division of the specifications over the spectrum analyser's sub-blocks.

Chapter 4

4 Frequency conversion

After determining the wanted specifications in chapter 2, we developed a top-level front-end architecture from the general principle of selecting a frequency and measuring a signal in chapter 3. We also determined we need frequency conversion in order to support the wanted input frequency range. Frequency conversion is done by a multiplication of two signals, the input signal (RF) and the local oscillator signal (LO), with the sum and the difference of their frequencies as the result.

Before discussing the implication of frequency conversion, the mathematical and graphical representation used throughout this chapter is introduced in section 4.1. The multiplication is performed by the mixer. In section 4.2 we will discuss a limitation of this front-end architecture with a normal mixer: the amplitude can not be determined with a zero intermediate frequency. A solution is presented by performing the multiplication of the input signal with two signals which are 90° out of phase (quadrature signals) with a quadrature mixer. An additional advantage of the quadrature architecture is the possibility to distinguish positive and negative frequencies, which can be exploited to perform image rejection. The frequency conversion results in two frequencies being converted to the intermediate frequency (IF), one frequency is the wanted signal (RF), the other is the unwanted or also called the image signal (IM). This is further discussed in section 4.3. Rejection of this image is discussed in section 4.4. Thus far the conversion to the IF, the location of images and the rejection of the image is discussed, but not the location of the intermediate frequency. This is discussed in section 4.5. It will turn out that the selected intermediate frequency has an unwanted component at DC. Section 4.6 will discuss the causes and possible techniques to remove this DC signal.

4.1 Mathematical and graphical representation

The spectrum analyser architecture discussed this far uses the multiplication of the input signal with a local oscillator signal. The result of the multiplication of two sinusoids is repeated from section 3.2.3 below:

$$x_{RF}(t)x_{LO}(t) = \frac{A_{RF}A_{LO}}{2}\left[\begin{array}{l}\cos\left(\left(\omega_{RF}+\omega_{LO}\right)t+\left(\varphi_{RF}+\varphi_{LO}\right)\right)\\ \qquad +\cos\left(\left(\omega_{RF}-\omega_{LO}\right)t+\left(\varphi_{RF}-\varphi_{LO}\right)\right)\end{array}\right] \tag{4-1}$$

Since this representation is quite extensive, we present the following conceptual simplification:

$$\begin{array}{l|c|}
 & \omega_1 \\ \hline
A\cos(\omega_1 t+\varphi_1) \Rightarrow \cos(\omega_1 t) \Rightarrow \cos & +\cos \\
A\sin(\omega_1 t+\varphi_1) \Rightarrow \sin(\omega_1 t) \Rightarrow \sin & +\sin
\end{array} \tag{4-2}$$

$$\begin{array}{l|cc}
 & \omega_1+\omega_2 & \omega_1-\omega_2 \\ \hline
\cos(\omega_1)\cdot\cos(\omega_2)=\cos(\omega_1+\omega_2)+\cos(\omega_1-\omega_2) \Rightarrow \cos\cdot\cos & +\cos & +\cos \\
\cos(\omega_1)\cdot\sin(\omega_2)=\sin(\omega_1+\omega_2)-\sin(\omega_1-\omega_2) \Rightarrow \cos\cdot\sin & +\sin & -\sin
\end{array}$$

Thus we leave the initial phase out and note that the phase behaves in the same way as the frequencies. Also we leave the amplitude out as we are primarily interested in the resulting frequencies. Note that the difference between a sine and a cosine at the same frequency and initial phase, is a 90° phase shift.

If we are only interested in the frequencies of a signal or a certain bandwidth and not in particular in the amplitude and phase, the signals can represented graphically as in Figure 29. This is called a single-sided spectrum.

Figure 29. Representation of signal frequencies and bandwidth.

Mathematically a pair of numbers can be represented as a complex number (section 2.1.2). Sinusoids are a set of two complex exponentials. The amplitude and instantaneous phase can be represented as the polar coordinates in the complex plane. The frequency is the rate of change of the phase. The sign of the frequency indicates the direction of the phase rotation in the complex plane. From section 2.1.2:

$$\omega = \frac{\partial \theta}{\partial t}$$

(4-3)
$$A\cos(\omega t + \varphi) = A \cdot \frac{e^{j\varphi}e^{j\omega t} + e^{-j\varphi}e^{-j\omega t}}{2}$$

$$A\sin(\omega t + \varphi) = j \cdot A \cdot \frac{-e^{j\varphi}e^{j\omega t} + e^{-j\varphi}e^{-j\omega t}}{2}$$

Graphically with *t=0* and *φ=0*, a cosine is then represented as in Figure 30 and a sine as in Figure 31.

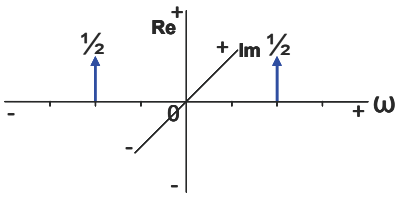


Figure 30. Cosine in a complex plane versus frequency.

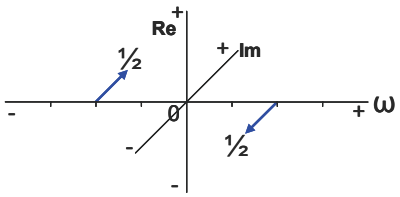


Figure 31. Sine in a complex plane versus frequency.

Multiplication of two exponentials also results in an exponential. The exponents of the two original numbers are added. Thus, the multiplication of complex exponentials in the time domain is equivalent to shifting on a frequency scale. Multiplying with $e^{j\omega t}$ means shifting the frequencies to the right and multiplying with $e^{-j\omega t}$ to the left. Since a sine or cosine consists of two complex exponentials, multiplying with them in the time domain, means convolving the spectrum with two complex exponentials in the frequency domain. This convolution results in shifting the spectrum down by the complex exponential with a negative frequency and up by the complex exponential with the positive frequency. This is shown in Figure 32 for a multiplication with a cosine.

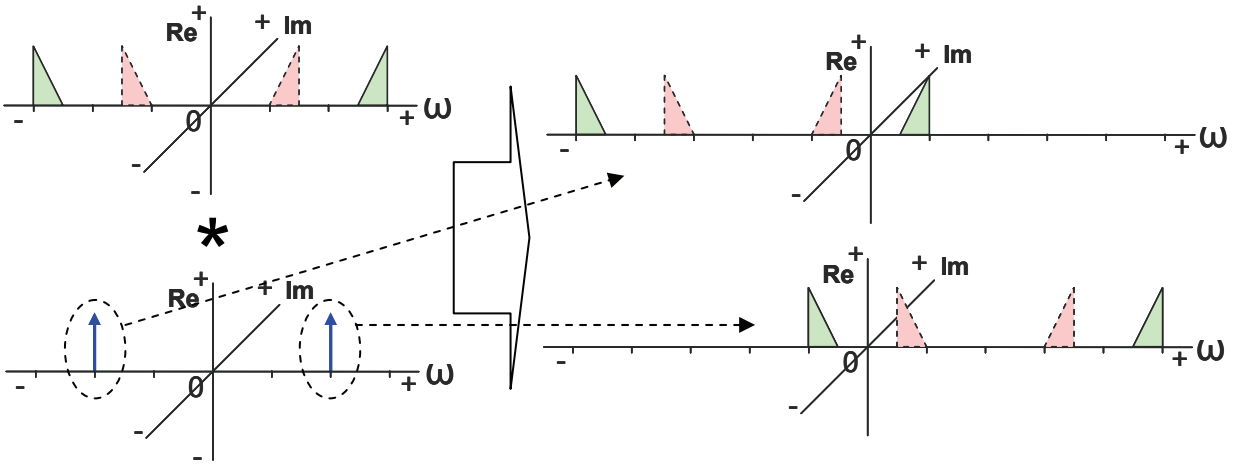


Figure 32. Multiplication of an input spectrum with a cosine.

The two shifts from the complex exponentials are kept in separate figures for clarity, but **they represent one spectrum**. From the figure we see that indeed two signals end up at the same frequency, one the wanted signal, the other the image.

In Figure 32 the input spectrum and output both consist of cosines. Sometimes we are not interested in the amplitude and phase of the input signal, but we are interested in the direction of the phase rotation. This is because the shifted spectrum can give a result with the same frequency but opposite direction of rotation at a certain frequency. This direction is important as two signals with the same frequency with opposite direction of rotation can be distinguished as will follow in section 4.2. These signals are thus more easily represented by also using negative frequencies as in Figure 33. This is called a double-sided spectrum. Note that a physical real signal always has an even spectrum.

Figure 33. Frequency bands represented using positive and negative frequencies.

Each representation represents the same thing with more or less information about the signal. The single sided spectrum of Figure 29 only shows the existence of a signal at a certain frequency. Figure 33 also indicates the direction of rotation in the complex plane by using negative frequencies, while Figure 32 includes the phase relation (the angles in the complex plane) between signals.

4.2 Simple and quadrature architecture

The architecture of chapter 3 uses a mixer with the RF input signal and the local oscillator signal as inputs and the intermediate frequency as output. A frequency range of interest is swept through an intermediate frequency one RBW by one. The primary objective of the spectrum analyser is to measure the amplitude or magnitude of the translated input signal at the intermediate frequency.

We identified the most basic architecture to perform this function. The amplitude and phase at the IF can be determined by analysing a period of the signal at the IF. The measured maximum determines the peak amplitude and the zero crossings determine the phase (in reference to a time origin). However, this "simple" architecture has a problem when the RF and LO frequencies are the same, because the signal at the IF is not varying anymore. In this case, the amplitude and phase can not be determined from the period. This is explained in section 4.2.1.

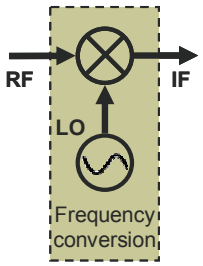


Figure 34. Simple mixer

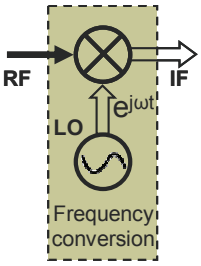


Figure 35. Quadrature mixer.

A quadrature architecture creates two signals from the input signal, which are 90° out of phase. The use of quadrature signals has two important advantages. The first is that the amplitude and phase can be determined instantaneous by using Pythagoras's formula. The second is that positive and negative frequencies can be differentiated, which is useful for the selection of a single sideband or for image rejection. This is explained in section 4.2.2. Because literature about quadrature signals is relatively scarce and often confusing, we will provide a somewhat extensive treatment of the quadrature operation principle illustrated with figures and the resulting advantages, with an emphasis on the use for our front-end.

Two signals in quadrature can be generated by the mixer. A mixer that performs this is called a quadrature mixer. We will call a normal mixer, a simple mixer. The graphical distinction is shown in Figure 34 and Figure 35. Note that mixing with quadrature signals mathematically corresponds to multiplying with a single complex exponential as indicated in Figure 35. Furthermore, the open arrow represents two signals which are 90° out-of phase (in quadrature) and together represent a complex signal.

4.2.1 Simple architecture

The "simple" spectrum analyser architecture of Figure 36 uses a simple mixer for the frequency conversion. The multiplication of two sinusoids results in sinusoids at the sum and difference frequencies. One of those will be the intermediate frequency.

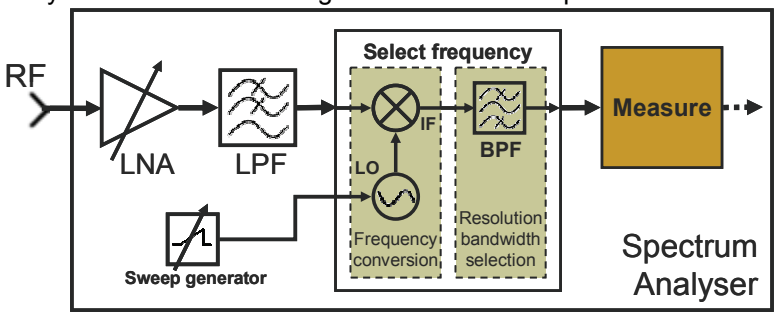


Figure 36. "Simple" spectrum analyser front-end architecture.

4.2.1.1 Amplitude and phase

Assume we are interested in the difference frequency. This is:

$$x_{IF}(t) = \frac{A_{RF} A_{LO}}{2}\left[\cos\left(\left(\omega_{RF} - \omega_{LO}\right)t + \left(\varphi_{RF} - \varphi_{LO}\right)\right)\right] \tag{4-4}$$

The resulting signal after the resolution bandwidth filter at the IF is a single known frequency (because all the other signals are filtered and the IF is known).

The amplitude $A_{RF}A_{LO}/2$ of the signal can not be determined directly from the measured instantaneous amplitude, because the phase φ_{RF}-φ_{LO} is unknown. The amplitude can be found by measuring the peak value. The signal must therefore be tracked sufficiently long, so that the peak amplitude is encountered at least once (i.e. at least half a period of the intermediate frequency). The amplitude measurement therefore also takes time adding to the sweep time and depending on the intermediate frequency.

The phase φ_{RF}-φ_{LO} can be determined by relating the zero crossings of the signal to a time reference. However to relate the found phase of the current RBW to previous or following RBWs, the initial phase of the local oscillator must be known or stay the same between measurements. For example, starting with the same initial phase after a RBW step. Or determining the phase shift of the LO signal of the previous step to the current step. Because of the LO is not a perfect sinusoid at the correct frequency instantly and because of phase noise, this is not practical. Therefore, the phase is not available with a simple architecture.

4.2.1.2 Measurement problem

From (4-4) we can see that the intermediate frequency at the difference frequency is:

$$\omega_{IF} = \omega_{RF} - \omega_{LO} \tag{4-5}$$

We can identify a problem when the difference is used with equal frequencies (i.e. $\omega_{RF}=\omega_{LO}$). The intermediate frequency is then zero and if we filter out the sum frequencies we find from (4-1) or (4-4):

$$x_{IF}(t) = \frac{A_{RF} A_{LO}}{2}\left[\cos\left(\varphi_{RF} - \varphi_{LO}\right)\right] \tag{4-6}$$

The signal at IF has frequency $\omega_{IF}=0$ and is therefore not varying with time. Unfortunately we do not know before hand what the phase relation between the local oscillator and the input signal is. Therefore $A_{RF}A_{LO}$ can not be determined. Because the instantaneous phase is not varying, the instantaneous (measured) amplitude is not varying and we can not determine the peak amplitude by taking the maximum measured amplitude. The measured value could just as well be the peak amplitude with the phase close to 0 radians as a large amplitude with a phase close to ½π radians.

A possible solution if we want to use this zero intermediate frequency is to perform multiple measurements at different times (after the LO has been set at other frequencies and thus φ_{LO} has been "reset"). If these measurements are performed at (quasi) random time intervals we can assume the phase difference to be uniformly distributed. Therefore the rectified average value will be the RMS value of the amplitude. Of course this is at the cost of extra time as multiple sweeps must be carried out. Also the phase relationship of input signals can not be determined.

4.2.2 Quadrature architecture

With the use of quadrature signals we can instantly determine their amplitude and phase, even in case of a zero intermediate frequency (see section 4.2.1.2). Another important use is the ability to distinguish two signals which end up at the same frequency after a frequency conversion. If one signal is the wanted signal and the other the unwanted image, we can select the wanted signal and thus perform image rejection. For a frequency bandwidth close to the LO frequency at IF, this is known as single sideband selection and rejection.

First we will explain what quadrature signals are in section 4.2.2.1. Next, we will explain how we can get two signals which are in quadrature by applying the Hilbert transform to a signal. The Hilbert transform performs a 90° phase shift for all frequencies.

The use of quadrature signals to determine the amplitude and phase of the signals is discussed in section 4.2.2.3. The use of quadrature signals to distinguish signals after frequency conversion is discussed in section 4.2.2.4. Finally it is explained how to get quadrature signals combined with frequency conversion by using a quadrature mixer.

4.2.2.1 Quadrature signals

In section 2.1.2, we found that a sine wave consists of two complex exponentials with the same value but opposite sign. This was the result of the complex exponential being a coordinate pair in the complex field. Because the sine and cosine are two orthogonal signals in the complex plane of which the complex exponential is the polar form, we have:

$$r \cdot e^{j(\omega t+\varphi)} = r \cdot \cos(\omega t+\varphi) + j \cdot r \cdot \sin(\omega t+\varphi)$$
$$r \cdot e^{-j(\omega t+\varphi)} = r \cdot \cos(\omega t+\varphi) - j \cdot r \cdot \sin(\omega t+\varphi) \quad \text{(4-7)}$$

Thus a complex exponential consists of a sine and a cosine of the same frequency, with the same amplitude and initial phase. Two orthogonal signals and thus a sine and cosine of the same frequency are 90° out of phase in a Euclidean space. Two signals which are 90° out of phase are said to be in quadrature. The sine lagging or leading the cosine can be mathematically seen as a positive or a negative frequency.

The real part of a complex exponential corresponds to the cosine. The imaginary part corresponds to the sine. The real and imaginary parts are the Cartesian coordinates in the complex plane.

If we have two signals in quadrature then together they represent a complex exponential. The complex value is represented as a pair of two physically real values, one the real part, the other the imaginary part.

4.2.2.2 Hilbert transform

So how can we get two signals which are 90° out-of-phase. We know that a sine and a cosine of the same frequency are in quadrature, which we can represent as:

$$A \cdot \cos(\omega t+\varphi) = A \cdot \frac{e^{j(\omega t+\varphi)} + e^{-j(\omega t+\varphi)}}{2} = \frac{A}{2} \cdot e^{j(\omega t+\varphi)} + \frac{A}{2} \cdot e^{-j(\omega t+\varphi)}$$
$$A \cdot \sin(\omega t+\varphi) = j \cdot A \cdot \frac{-e^{j(\omega t+\varphi)} + e^{-j(\omega t+\varphi)}}{2} = -j \cdot \frac{A}{2} \cdot e^{j(\omega t+\varphi)} + j \cdot \frac{A}{2} \cdot e^{-j(\omega t+\varphi)} \quad \text{(4-8)}$$

The cosine and sine are represented as two complex frequencies in a complex plane in Figure 37 and Figure 39. We can see from the equations and figures that a 90° phase decrease corresponds to multiplying the positive frequency with –j (or -90° in the complex plane, being counter-clockwise) and the negative frequency with j (or +90° in the complex plane, being clockwise) for any ω. If this is done for all frequencies at once, all frequencies are shifted 90° in phase. This is called the Hilbert transform and shown in Figure 38. Applying the Hilbert transform to a cosine gives a sine. Note that performing the Hilbert transform on the sine gives *–½j· -j= -½* for positive frequencies and *½j·j= -½* for negative frequencies, which corresponds to a negative cosine as expected.

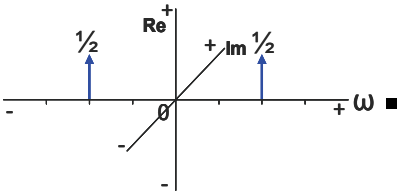


Figure 37. Cosine in a complex plane versus frequency.

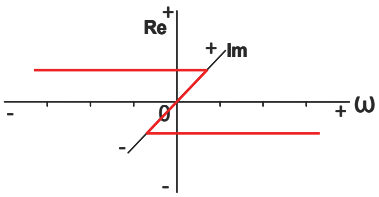


Figure 38. Hilbert transform in a complex plane versus frequency.

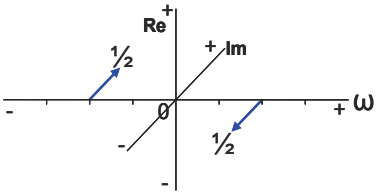


Figure 39. Sine in a complex plane versus frequency.

The Hilbert transform can be implemented or approximated in several ways, two of which will be discussed in section 4.4.

4.2.2.3 Amplitude and phase

A second solution to the problem of not being able to determine the amplitude with a zero intermediate frequency (section 4.2.1.2) exists, if we find a way to have the measured signal in quadrature, i.e to duplicate the input signal and implement and perform the

Hilbert transform on one of them. If we take one signal to be the cosine of an angle θ, then the other signal is the sine (because they are in quadrature) and from Pythagoras (or a coordinate transformation from Cartesian to polar circular system) we find:

$$A\cos\theta = a\ (adjacent), \quad A\sin\theta = o\ (opposite)$$

(4-9)

$$A = \sqrt{a^2 + o^2}, \quad \theta = \arctan\frac{a}{o}$$

The cosine corresponds to the $x_{IF}(t)$ from equation (4-4) in section 4.2.2.1, which gives $A=A_{RF}A_{LO}/2$ and $\theta=\varphi_{RF}-\varphi_{LO}$. We can determine both instantly by measuring the amplitudes of the quadrature signals at any time.

Of course the intermediate frequency does not have to be zero to be able to use this. For the general case we have for the intermediate frequency:

$$x_{IF}(t) = A_{IF}\cos\theta_{IF}$$

(4-10)

$$A_{IF} = \frac{A_{RF}A_{LO}}{2}, \quad \theta_{IF} = \omega_{IF}t + \varphi_{IF}, \quad \omega_{IF} = \omega_{RF} \pm \omega_{LO}, \quad \varphi_{IF} = \varphi_{RF} \pm \varphi_{LO}$$

Thus, the difference between determining the amplitude and phase in a quadrature architecture to a simple architecture is, that the amplitude and phase can be determined instantly at any time, instead of waiting for a peak amplitude or zero crossing.

Relating the phase to previous measurements at different frequencies is not practical as discussed in section 4.2.2.1.

4.2.2.4 Single sideband selection

Thus, the use of quadrature signals allows us to instantly measure the amplitude and phase. However, the use of quadrature signals has another important advantage. It is possible to distinguish negative and positive frequencies. This is useful is the wanted signal is on one side and the image on the other side of the double-sided spectrum; i.e. we can reject the image. In combination with down-conversion, the selection of positive or negative frequencies corresponds to selecting the upper or lower sideband around the LO frequency. Note that down-conversion is always needed to move the image to a negative frequency.

Looking back at Figure 37 and Figure 39 of the complex representation of a cosine and a sine, we see that a cosine has positive components for both the complex exponential, while the sine has a component of *-j* for the positive frequency and *+j* for the negative frequency. If we take the cosine as the real part of a complex signal and the sine as the imaginary part, then together they represent a single exponential. If the imaginary part is added we get the exponential with a positive frequency and if it is subtracted we get the exponential with a negative frequency. This is shown in Figure 40.

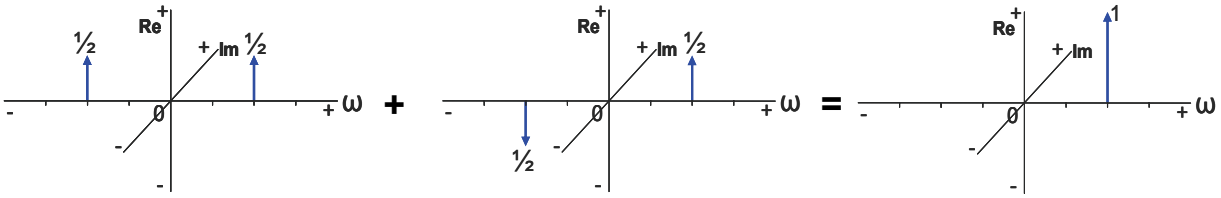


Figure 40. $\cos(j\omega t) + j\cdot\sin(j\omega t) = e^{j\omega t}$.

We can generalise this if we have two signals in quadrature. Because the signals are 90° out of phase for all frequencies, the above result for a cosine and sine holds for all frequencies. Thus together they represent all the positive frequency components or all the negative ones:

$$\cos(j\omega t) + j\cdot\sin(j\omega t) = e^{j\omega t}$$

$$\cos(j\omega t) - j\cdot\sin(j\omega t) = e^{-j\omega t}$$

(4-11)

$$x_a(t) = x(t) + j\cdot\hat{x}(t)$$

$$\overline{x_a}(t) = x(t) - j\cdot\hat{x}(t)$$

The signal $\hat{x}(t)$ is the Hilbert transform of $x(t)$. It will follow in section 4.2.2.5, that $x(t)$ is also labelled as in-phase (I) and $\hat{x}(t)$ as quadrature (Q). The signal $x_a(t)$ is called an analytic signal [41]. An analytic signal has no negative frequency components. Its complex conjugate $\overline{x_a}(t)$ only has negative frequencies and no positive ones.

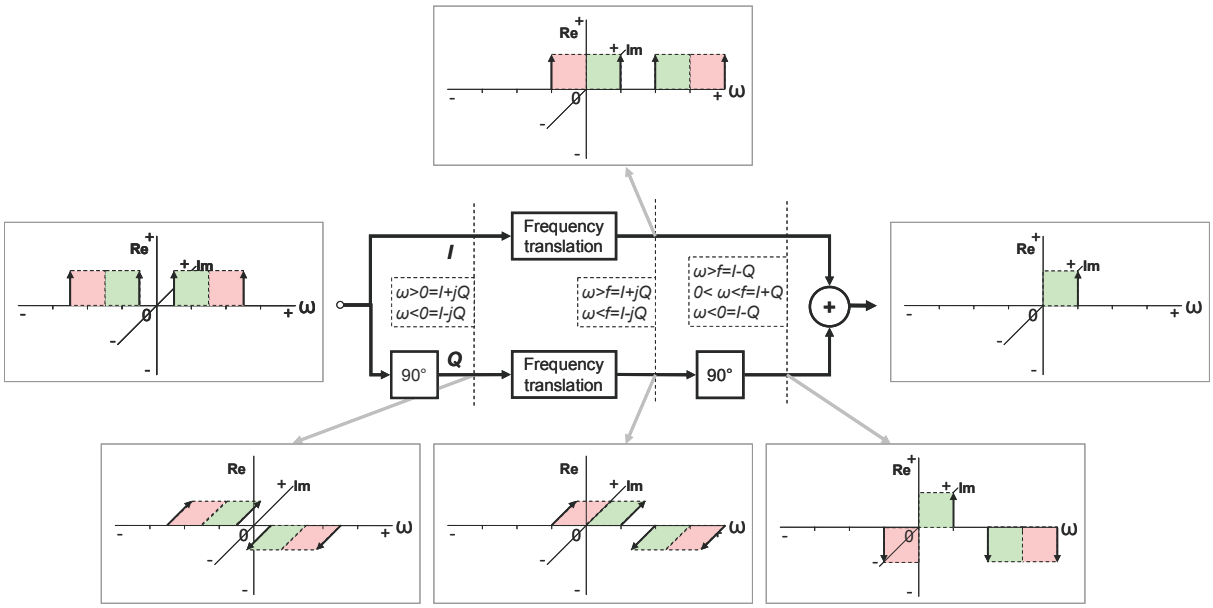


Figure 41. Single sideband selection with quadrature signals.

A physically real signal has an even double sided spectrum. Therefore being able to represent a single side of the spectrum (around zero, thus just the negative frequencies or just the positive frequencies) is not particularly useful for us, as for each side the unwanted images are still part of the spectrum (it is if you want to perform single sideband modulation). However, combined with a frequency translation it is useful.

Quadrature signals are created by duplicating the input signal and performing the Hilbert transform on one of them (see Figure 41). For quadrature signals, the positive frequencies are represented by *I+jQ* and the negative by *I-jQ*. However, something interesting happens when we perform a frequency translation. Let's translate both the *I* and *Q* frequencies up by a frequency of *f*. Since both signals are translated, nothing has changed relative to each other. But this means *I+jQ* now represents frequencies higher than *f* and *I-jQ* represents frequencies lower than *f* including the negative frequencies. Another (second) Hilbert transform results in multiplying with $j \cdot j = -1$ for positive frequencies and with $j \cdot j = -1$ for negative frequencies. But what about the negative frequencies that where translated to the positive side? Those frequency components were multiplied with *j* at the first Hilbert transform and are multiplied with *-j* for the second resulting in: $j \cdot -j = +1$. Added to the in-phase signal, all frequencies cancel each other out, except for those. This corresponds to the selection of the lower-sideband of the original (real) signal. If instead the signal subtracted from the in-phase signal, the negative frequencies that were translated to the positive side cancel each other out. Now the upper sideband is selected.

Possibly easier to follow is the use of a -90° phase shift ($\cdot -j$) for positive frequencies in the complex plane and a +90° phase shift ($\cdot j$) for negative frequencies. If we perform another Hilbert transform on the quadrature signal, normally the negative frequency components have changes -90° twice and the positive frequency components +90° twice. All frequencies thus have had a 180° phase shift and added to the in-phase signal they cancel each other. The translated frequency band first had a +90° phase shift as a negative frequency and then a -90° a positive frequency. Thus that frequency band has had a 0° phase shift.

Note that in Figure 41 only shows the up-translation of the signals, while the translation down must also be taken into account. For the down-translation, the in-phase signal is mirrored on the real-axis, and the quadrature signals are mirrored on the real and imaginary axes. The final results is the negative frequency band of the wanted signal. Thus making the spectrum symmetrical again for the physically real output signal.

Thus, by using a quadrature architecture, we are able to make a selection between the upper or lower side band. One contains the wanted signal, the other the image and is rejected. Combined with frequency conversion, we can choose the origin for the frequency selection, where we can choose all frequencies above or below that origin.

4.2.2.5 Quadrature mixer

In order to generate an input signal in quadrature, the input signal must be split into two paths and one or both paths must be shifted in phase so that they are 90° out of phase. This phase shift introduces distortions and noise. The large amplitude range of the input signal makes it difficult to guarantee the wanted performance over this complete range. Since the input signal frequency is converted to an intermediate frequency by a mixer, we can also move the phase shift to the local oscillator path instead of the input signal path.

We compare the RF signal in quadrature multiplied by a cosine LO, with a cosine RF multiplied by quadrature LO signals. The operation *cos·cos* is communative:

$$\cos(\omega_{RF}t+\varphi_{RF})\cdot\cos(\omega_{LO}t+\varphi_{LO})=\frac{1}{2}\cos((\omega_{RF}\pm\omega_{LO})t+(\varphi_{RF}\pm\varphi_{LO})) \tag{4-12}$$

But *sin·cos≠cos·sin*, with *sin·cos* corresponding with the quadrature RF signal multiplied by the LO signal, and *cos·sin* corresponding with the RF signal multiplied by the quadrature LO signal. The difference is:

$$\sin(\omega_{RF}t+\varphi_{RF})\cdot\cos(\omega_{LO}t+\varphi_{LO})=\left[\begin{array}{l}\frac{1}{2}\sin((\omega_{RF}+\omega_{LO})t+(\varphi_{RF}+\varphi_{LO}))\\ \qquad+\frac{1}{2}\sin((\omega_{RF}-\omega_{LO})t+(\varphi_{RF}-\varphi_{LO}))\end{array}\right] \tag{4-13}$$

$$\cos(\omega_{RF}t+\varphi_{RF})\cdot\sin(\omega_{LO}t+\varphi_{LO})=\left[\begin{array}{l}\frac{1}{2}\sin((\omega_{LO}+\omega_{RF})t+(\varphi_{LO}+\varphi_{RF}))\\ \qquad+\frac{1}{2}\sin((\omega_{LO}-\omega_{RF})t+(\varphi_{LO}-\varphi_{RF}))\end{array}\right]$$

The sum frequencies are equal but the difference frequencies are not:

$$\begin{aligned}\frac{1}{2}\sin((\omega_{LO}-\omega_{RF})t+(\varphi_{LO}-\varphi_{RF}))&=\frac{1}{2}\sin(-(\omega_{RF}-\omega_{LO})t-(\varphi_{RF}-\varphi_{LO}))\\&=\frac{1}{2}\sin((\omega_{RF}-\omega_{LO})t+(\varphi_{RF}-\varphi_{LO})+\pi)\end{aligned} \tag{4-14}$$

Thus, the difference is a 180° phase shift of the initial angle. A 180° phase shift of the initial angle does not change the phase relationships. Thus a 90° phase shift in the signal path or in the LO path makes no difference. A 90° phase shift in the local oscillator signal path is easier because the local oscillator amplitude is fixed. Thus, quadrature LO signals are preferred.

A mixer that combines frequency conversion with the generation of quadrature signals is called a quadrature mixer. The quadrature mixer implements two normal mixers one for the in-phase signal and one for the quadrature signal. The RF input signal is routed to both mixers and the *I* signal of the LO to one and the *Q* to the other. Quadrature signals are represented by a double lined (block) arrow. This results in the architecture of Figure 42, which we call a quadrature architecture.

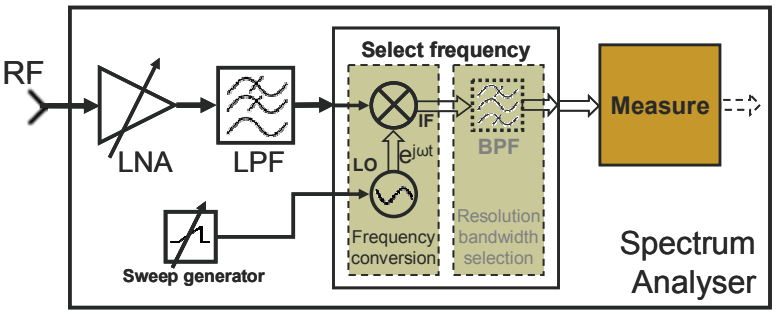


Figure 42. "Quadrature" spectrum analyser front-end architecture.

Thus, we can generate a quadrature signal with a quadrature mixer. The quadrature signal represents the original signal in the complex domain, with which we can determine the amplitude and phase easily. We can also select the frequency range above or below a certain frequency, which is determined by the origin of the frequency translation.

If the input signal is a cosine, then the results after the quadrature mixer are:

$$I(t) = A_{RF}\cos(\omega_{RF}t)\cdot\cos(\omega_{LO}t) = \frac{1}{2}A_{RF}\cdot\left[\cos\left((\omega_{RF}+\omega_{LO})t\right)+\cos\left((\omega_{RF}-\omega_{LO})t\right)\right]$$

(4-15)

$$Q(t) = A_{RF}\cos(\omega_{RF}t)\cdot\sin(\omega_{LO}t) = \frac{1}{2}A_{RF}\cdot\left[\sin\left((\omega_{RF}+\omega_{LO})t\right)+\sin\left((\omega_{RF}-\omega_{LO})t\right)\right]$$

I(t) is called the in-phase (*I*) component as it is in phase with the original input signal, while *Q(t)* is called the quadrature (*Q*) component as it is 90° out-of-phase with the original input signal. A single input frequency multiplied with a local oscillator frequency in quadrature thus gives quadrature sum and difference frequencies. Note that:

$$I(t) \pm jQ(t) = A_{RF}\cos(\omega_{RF}t)\cdot e^{\pm j\omega_{LO}t} \quad \text{(4-16)}$$

This was used in section 4.2.2.4.

4.3 Image frequency location

In section 3.2.3 we found that the frequency of the RF input signal is converted to an intermediate frequency by multiplying it with a local oscillator frequency. We also found that multiplying two sinusoids results in their sum and difference frequencies. The IF can thus be at the sum of the RF and LO signals. This is called up-conversion. The IF can also be at the difference of the RF and LO signals. This is called down-conversion.

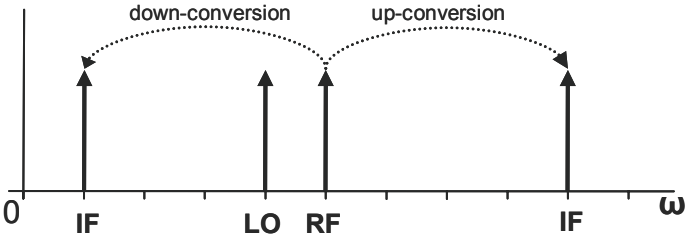


Figure 43. Signals resulting at sum and difference frequencies.

Figure 43 shows the resulting signals for a single RF sinusoid. All other sinusoids in the RF input signal are also up and down-converted. This also means that if we select a certain IF frequency, two different input frequencies are converted to the same intermediate frequency. One of these signals is the wanted signal; the other is unwanted or also called the image (IM) signal.

The image frequency location depends on the chosen IF. The IM frequency can be close to the RF in case of a low IF, or as close or far as you would like in case of up-conversion. The choice of the actual IF for the spectrum analyser front-end is postponed until section 4.5. First we will analyse the image frequencies for each possible IF. This allows us to determine the possibilities for image rejection.

For the multiplication of two cosines with frequencies ω_{RF} and ω_{IM} with the LO, we find:

(4-17)

	$\omega_{RF}+\omega_{LO}$	$\omega_{RF}-\omega_{LO}$	$\omega_{IM}+\omega_{LO}$	$\omega_{IM}-\omega_{LO}$
$\cos\cdot\cos$	$+\cos$	$+\cos$	$+\cos$	$+\cos$
$\cos\cdot\sin$	$+\sin$	$-\sin$	$+\sin$	$-\sin$

If we choose the IF at $\omega_{RF}+\omega_{LO}$ this corresponds to up-conversion of the RF signal. The term $\omega_{IM}+\omega_{LO}$ can not result in a signal at IF if $\omega_{IM}\neq\omega_{RF}$, but $\omega_{IM}-\omega_{LO}$ can. At the same time, if the IF is chosen to be at $\omega_{RF}+\omega_{LO}$, there is an image frequency of which $\omega_{IM}+\omega_{LO}$ also results at the IF.

Note that when the LO is a cosine, the results all become cosines with the same sign, but when the LO is a sine, the sums become positive sines and the differences become negative sines. This will be exploited by a quadrature architecture (section 4.2 and 4.4).

4.3.1 Sum frequency

When the sum frequency is selected as IF, the RF is up converted and the IM is down converted to the IF. The RF and IM signal are therefore two times the LO frequency apart:

$$\begin{aligned}\omega_{RF}+\omega_{LO} &= \omega_{IF}\\ \omega_{IM}-\omega_{LO} &= \omega_{IF}\\ \hline \omega_{RF}-\omega_{IM} &= -2\cdot\omega_{LO}\end{aligned} \quad \text{(4-18)}$$

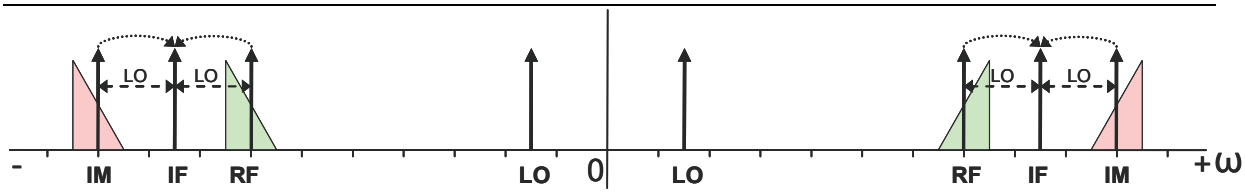


Figure 44. Up conversion of the RF with the IM two times the LO frequency away.

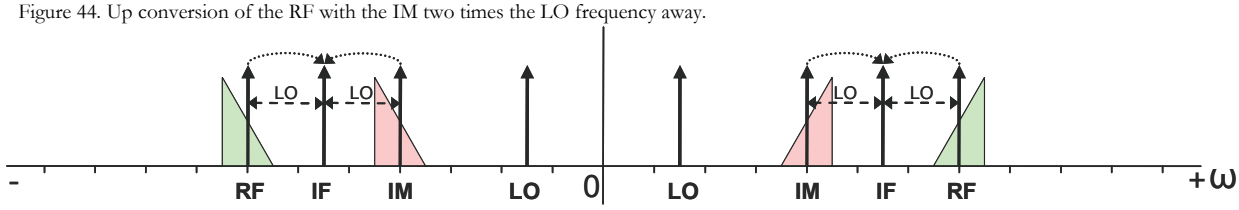


Figure 45. Down conversion of the RF with the image two times the LO frequency away.

This can also be seen in Figure 44. If the frequency of LO becomes larger, the IF will be higher and the IM also will be at a higher frequency.

4.3.2 Difference frequency

The difference frequency can also be chosen as IF. This is shown in Figure 45. For the frequency difference of RF and IM we find:

$$\omega_{RF} - \omega_{LO} = \omega_{IF}$$

(4-19) $$\underline{\omega_{IM} + \omega_{LO} = \omega_{IF}}\ -$$

$$\omega_{RF} - \omega_{IM} = 2 \cdot \omega_{LO}$$

The image is still two time the LO frequency away, only now at the other side of the RF signal.

If the frequency of the LO becomes larger, the IF becomes one time the LO frequency lower and the IM two times. In the case that $2 \cdot \omega_{LO} > \omega_{RF}$, the IM will be at a negative frequency. This is shown in Figure 46. When the image frequency has become negative, its frequency will increase instead of decrease with an increasing LO frequency. As long as $\omega_{LO} < \omega_{RF}$, the IM frequency will be less than the RF. When the LO frequency becomes closer to the RF, the IM frequency will also become closer. For the frequency difference of RF and IM we find:

$$\omega_{RF} - \omega_{LO} = \omega_{IF}$$

(4-20) $$\underline{-\omega_{IM} + \omega_{LO} = \omega_{IF}}\ +$$

$$\omega_{RF} - \omega_{IM} = 2 \cdot \omega_{IF}$$

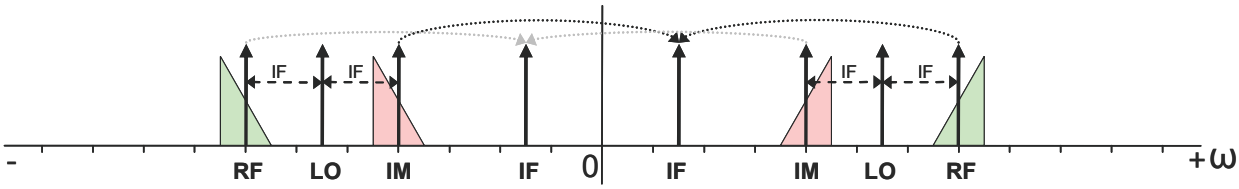


Figure 46. Down-conversion to a low IF, the image is now two times the IF away.

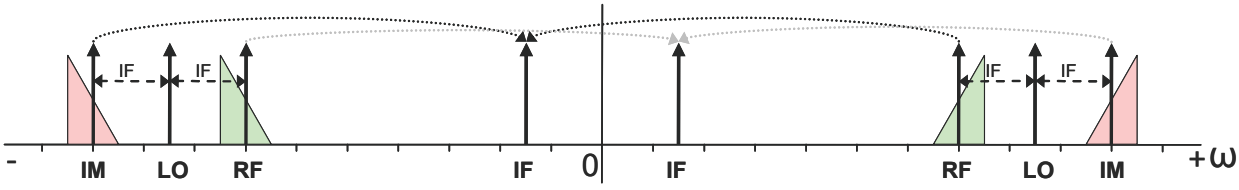


Figure 47. Down-conversion to a negative low IF, the image is still two times the IF away only at the other side of the RF band.

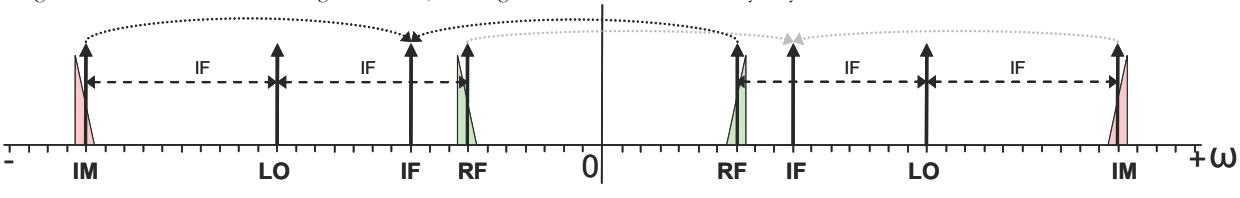


Figure 48. Up-conversion by using the frequency difference with a high LO frequency., the image is still two times the IF away.

Now the IM frequency is two times the IF away from the RF signal, instead of the LO frequency. This is also indicated in Figure 46. The closer IF becomes to zero, the closer the IM frequency will become to the RF. If we compare Figure 45 and Figure 46, we can see the locations of the IF and LO frequency have switched.

The situation of Figure 46 is quite common when a signal is modulated by a carrier. The RF signal is the original signal and the LO is the frequency of the carrier. Because the carrier is normally a high frequency, while the IF is low to allow signal processing, the LO frequency is close to the RF and $2 \cdot \omega_{LO} > \omega_{RF}$. Multiplying the RF and IM signal with the LO signal can be seen as translating the positive frequencies down and the negative frequencies up with RF and IM around DC (section 4.1). The RF band is called the upper-sideband and the IM frequency band is called the lower sideband.

The LO frequency can also be made larger than the RF ($\omega_{LO} > \omega_{RF}$). The RF is then translated to a negative IF:

$$\begin{aligned} &\omega_{RF} - \omega_{LO} = -\omega_{IF} \\ &\underline{-\omega_{IM} + \omega_{LO} = -\omega_{IF}}_{+} \\ &\omega_{RF} - \omega_{IM} = -2 \cdot \omega_{IF} \end{aligned} \qquad (4\text{-}21)$$

Again the image is still two times the IF away, only now at the other side of the RF signal. This is shown in Figure 47. The RF band is now the lower sideband and the image the upper sideband.

There is one final situation. If the LO frequency is increased further and $\omega_{LO} > 2 \cdot \omega_{RF}$, then $\omega_{IF} > \omega_{RF}$. Thus although the difference frequency is used, the IF is above the RF. This is shown in Figure 48. Nothing changes for equation (4-21), but because the IF is high, the IM frequency is also high.

Thus, looking from the RF signal, if the sum of the RF and LO frequency is taken as IF, the RF is up-converted and the image is located $2 \cdot \omega_{LO}$ higher. If the difference is taken, with $\omega_{IF} > \omega_{RF}/2$ then the RF is down-converted and the image is located $2 \cdot \omega_{LO}$ lower.

If $\omega_{IF} < \omega_{RF}/2$ then RF is down-converted and the image is located $2 \cdot \omega_{IF}$ lower. The RF signal is the upper sideband in this case. With $\omega_{IF} < 0$ the RF is the lower sideband and the image is located $2 \cdot \omega_{IF}$ higher. Finally if $\omega_{IF} < -\omega_{RF}$, the RF is up-converted again, but now the image is located $2 \cdot \omega_{IF}$ higher.

This is illustrated in Figure 49.

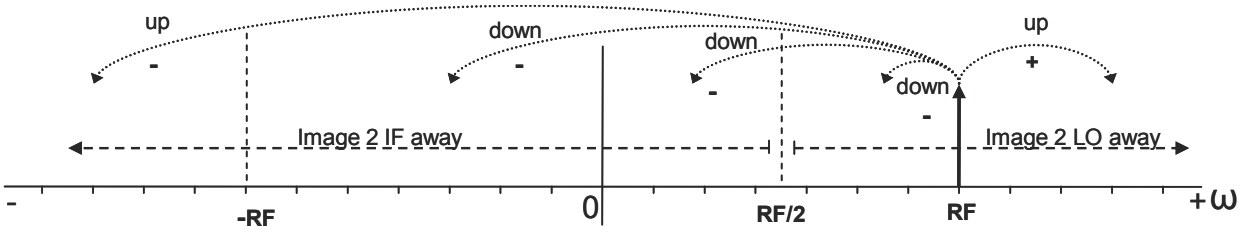


Figure 49. The five different IF sections.

4.4 Image rejection

The frequency that is intended to be translation to the intermediate frequency is called the wanted signal, while the other is called the image signal. We would like to suppress this unwanted image so as to not distort the wanted signal. This is called image rejection.

There are two possible solutions to suppress the image. The first is filtering the image out before the frequency conversion (as afterwards it is combined with the wanted signal). Depending on the frequency conversion scheme used this is a viable solution. The frequency conversion schemes are further discussed in section 4.5 and include treatment of the possibility of a filter. The second solution is to use quadrature signals to select a single sideband as discussed in section 4.2.2.

4.4.1 Filtering

The filtering of all frequencies other than the signal of interest is shown in Figure 50. The filter attenuates possible images as well as other large signals that can for example cause linearity distortions. The filtering of everything but the signal of interest is therefore very attractive, especially if the signal of interest is at a fixed frequency. That is the case for most radio receivers where pre-filtering is often used. However, the filtered frequency band often consists of many channels, which makes further filtering or other measures necessary.

In case of a spectrum analyser, the input frequency range is very large (0 ~ 3/6GHz). The resolution bandwidth selects only a small part of this range, which causes the image to fall in the input frequency range. Furthermore, the wanted signal can be located anywhere in the input range before it converted to the IF. The filter must thus be tuneable, which can be difficult to implement (especially on-chip) in case the image is close.

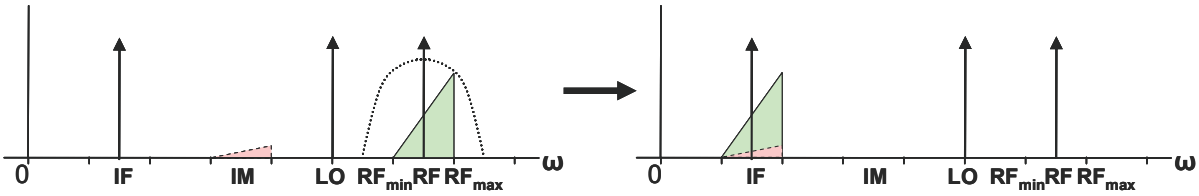


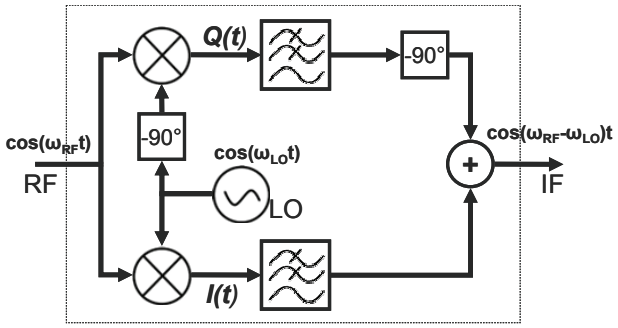
Figure 50. Filter image (displayed attenuated) before frequency conversion.

4.4.2 Quadrature image reject architectures

Section 4.2.2 showed how quadrature signals can be used to select a sideband around the centre frequency before conversion. The quadrature architecture is further developed by three different image rejection architectures.

4.4.2.1 Hartley

Figure 51 shows the Hartley image reject frequency conversion [4]. The Hartley architecture uses a local oscillator in quadrature as a quadrature mixer.



	$\omega_{RF}-\omega_{LO}$	$\omega_{RF}+\omega_{LO}$
$\cos\cdot\cos$	$+\cos$	$+\cos$
$\cos\cdot\sin$	$-\sin$	$+\sin$
$\cos\cdot\sin/90°$	$+\cos$	$-\cos$
$c\cdot c+c\cdot s/90°$	$+\cos$	

Figure 51. Hartley image reject frequency conversion.

The complex signal (I/Q) after the quadrature mixer is first low-pass filtered to filter out the sum frequency (and possibly channel selection). Note that this is not strictly necessary as will follow from the following treatment where we will not perform this filtering. As in section 4.2.2 the quadrature architecture is used to select the upper or lower frequencies relative to the first frequency conversion (i.e. the local oscillator). The $\omega_{LO}+\omega_{IF}$ and $\omega_{LO}-\omega_{IF}$ frequency bands are called the sidebands of ω_{LO}. With the quadrature architecture we can select one or the other. Focussing on image rejection, selecting only one means we can differentiate the wanted signal from the image.

Multiplication with the in-phase local oscillator signal shifts the input signal up and down as was shown in Figure 32 and repeated in Figure 52. For the quadrature local oscillator signal, we get the 90° phase shifted version as shown in Figure 53. This means multiplying with a quadrature mixer performs a frequency translation and a Hilbert transform in one.

The quadrature signal next undergoes another 90° phase shift (Figure 54), which corresponds to the second Hilbert transform. The in-phase (Figure 52) and phase shifted quadrature signal (Figure 54) are then added to get the lower single sideband (Figure 55).

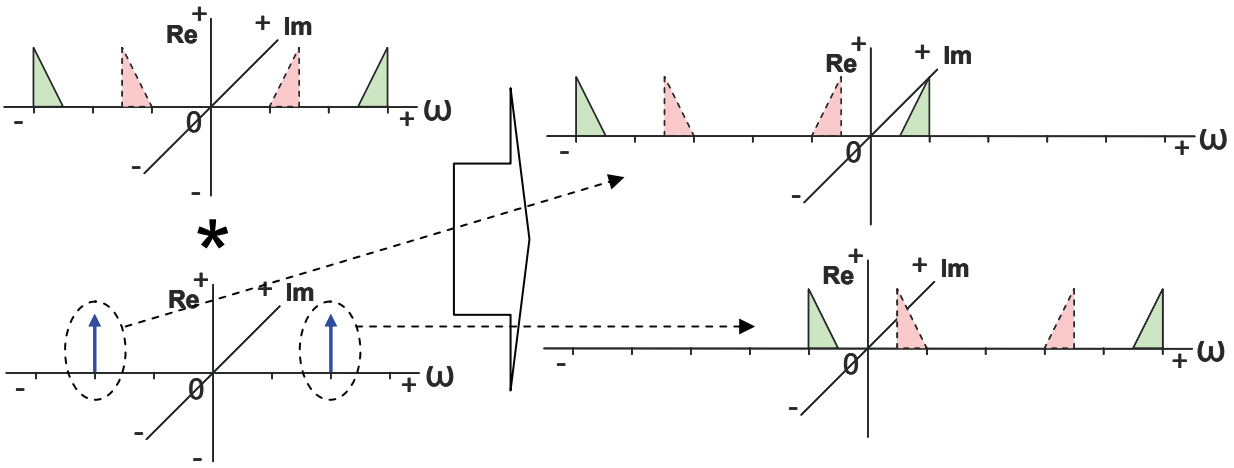


Figure 52. Multiplication of an input spectrum with a cosine (producing I).

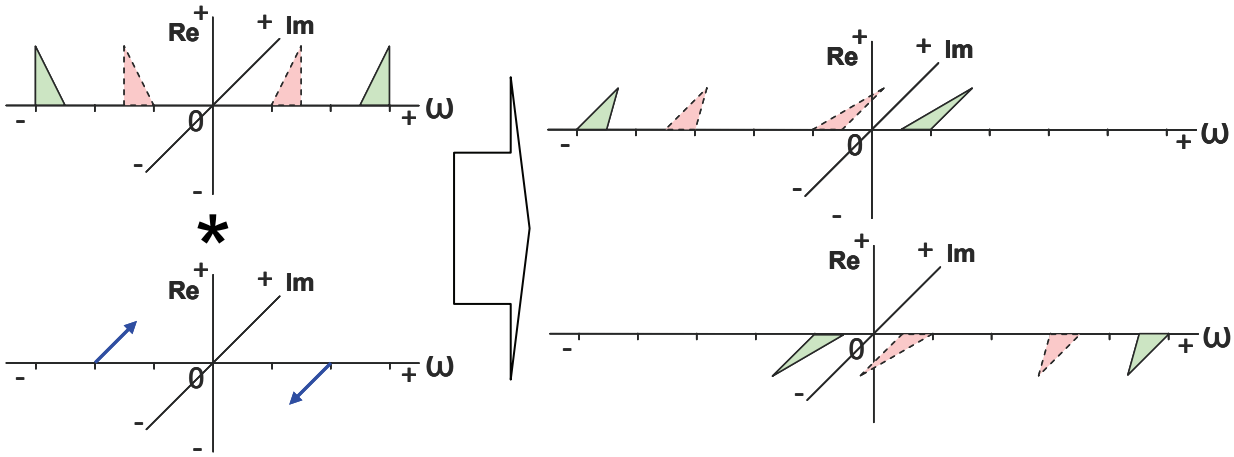


Figure 53. Multiplication of an input spectrum with a sine (producing Q).

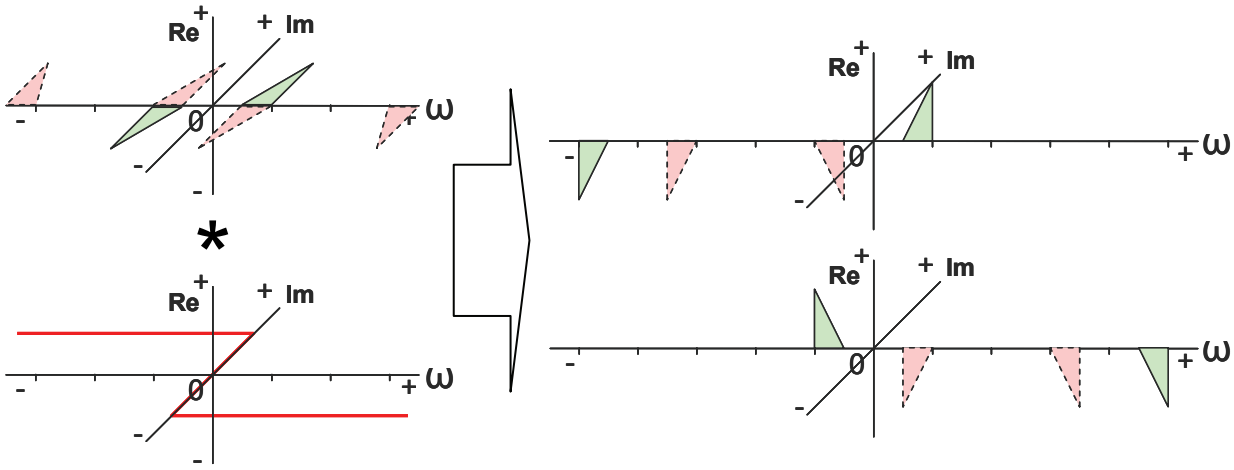


Figure 54. Hilbert transform of the quadrature signal.

The implementation of the frequency generation and the 90° phase shift are outside of the scope of this thesis. However, the accuracy of the phase difference between the in-phase and quadrature signal, determines how well the unwanted signals are removed. This is discussed in section 4.4.2.5 as it applies to all quadrature architectures.

Figure 55. Added in-phase signals and phase-shifted quadrature signals.

4.4.2.2 Weaver

In the previous section we found that the quadrature mixer performs a frequency translation and Hilbert transform in the quadrature path and only a frequency translation in the in-phase path. Therefore we can also implement the second Hilbert transform as another quadrature mixer if we allow a second frequency translation. This is the Weaver architecture [4] and is shown in Figure 56. Again the low-pass filters are for filtering out the higher sum frequencies and are not strictly necessary.

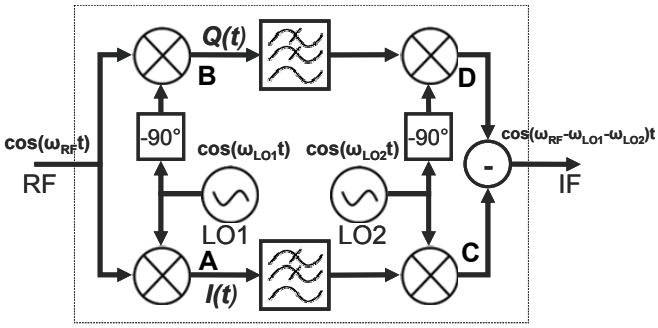


Figure 56. Weaver image reject frequency conversion.

	$\omega_{RF} - \omega_{LO1} - \omega_{LO2}$	$\omega_{IM1} - \omega_{LO1} + \omega_{LO2}$	$-\omega_{IM2} + \omega_{LO1} - \omega_{LO2}$	$-\omega_{IM3} - \omega_{LO1} - \omega_{LO2}$
$\cos\cdot\cos\cdot\cos$	$+\cos$	$+\cos$	$+\cos$	$+\cos$
$\cos\cdot\sin\cdot\sin$	$-\cos$	$+\cos$	$+\cos$	$-\cos$
$c\cdot c\cdot c - c\cdot s\cdot s$	$+\cos$			$+\cos$

The Weaver architecture uses two quadrature mixers both performing a frequency conversion. Before we identified that two input frequencies (the wanted and image signal) convert to the same output frequency causing interference. With two cascaded mixers this happens twice. Indeed we now have four frequencies to take into account that can all convert to the same intermediate frequency as shown in Figure 57.

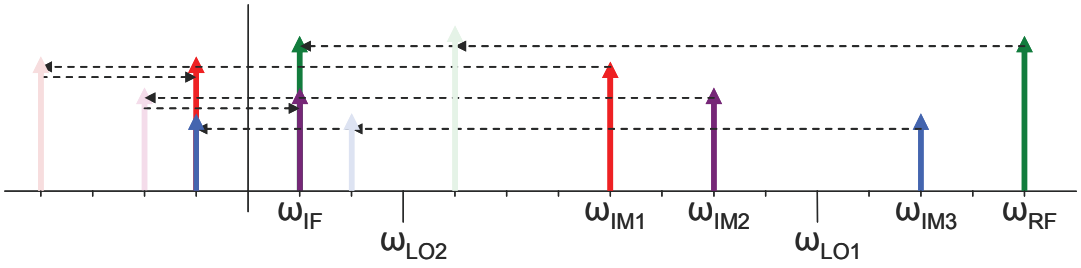


Figure 57. Wanted and image signal for two down-conversion stages (dual-IF).

Next we will analyse the input and output signal as well as the signals at positions A, B, C and D. Our input signal shown in Figure 58 now shows four frequencies to take into account, the four of Figure 57. The results of the first stage are the same as *I(t)* and *Q(t)* of the Hartley architecture, shown with four frequencies as *A* and *B* in Figure 59. Again note that the signal representations are separated for clarity but they represent one combined signal.

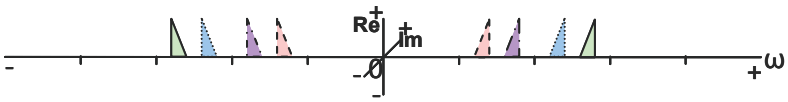


Figure 58. Wanted input signal with three image signals.

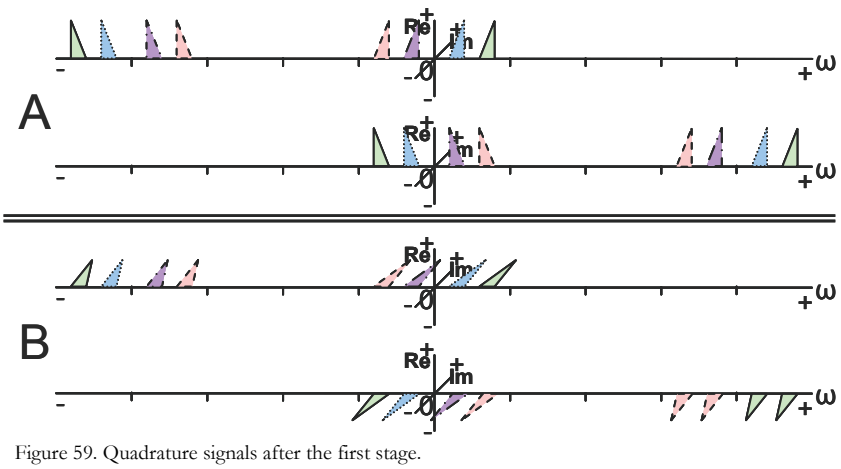


Figure 59. Quadrature signals after the first stage.

The second stage performs another multiplication of *A* with a cosine and *B* with a sine. The resulting signals *C* and *D* are shown in Figure 60

When D is subtracted from C, we can see that the second and third line cancel

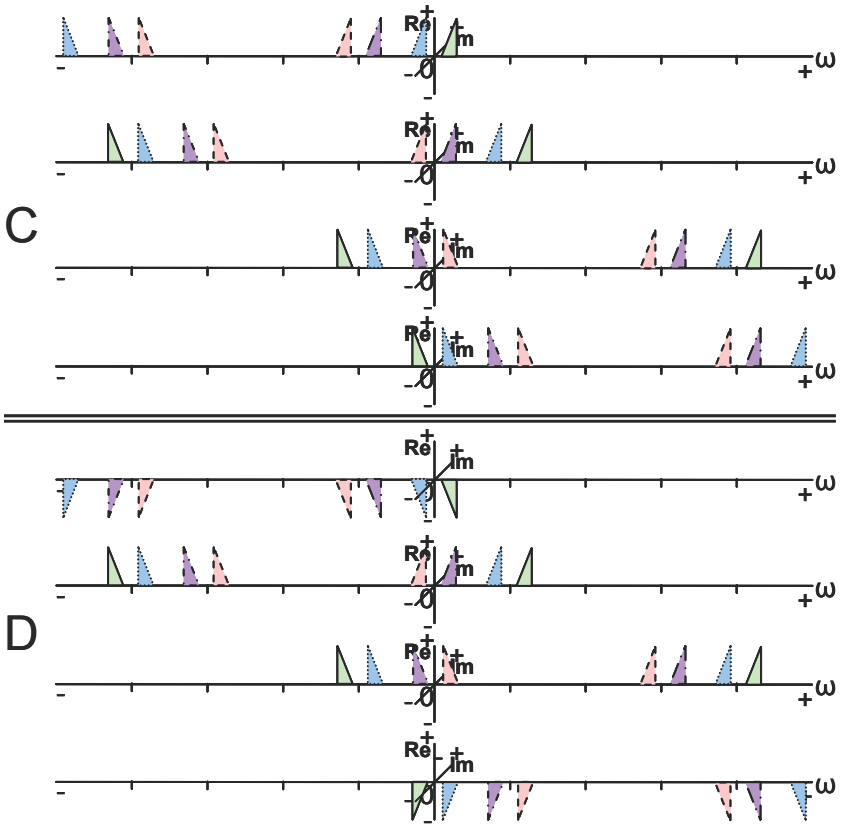


Figure 60. Quadrature signals after the second Weaver stage (without low-pass filter).

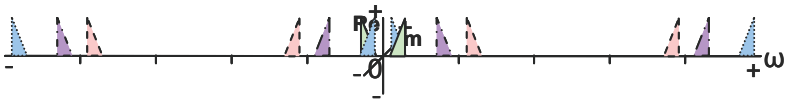


Figure 61. Output signal after subtracting D from C.

out, while the first and fourth lines add up. Therefore the resulting output signal is the one shown in Figure 61. There are higher frequency signals which could also be present with the Hartley architecture but just were not shown. These higher frequencies can be filtered. There is also an image signal (ω_{IM3}) at the location of the wanted signal. This is also the result from the mathematical analysis shown above. Thus although the image frequencies on the left of ω_{LO1} are cancelled, the frequencies on the right are not. This confirms with our frequency selection discussion of a quadrature architecture in section 4.2.2.4. Note that the image that is cancelled corresponds to the image from the first conversion stage, while the one that is not corresponds to the image of the second conversion stage.

Thus, the second stage of the Weaver architecture allows cancellation of the image of the first stage, but also introduces a secondary image from the second stage, which is not cancelled. Another stage could be added to cancel the secondary image, but this would introduce a tertiary image and so on. The Weaver architecture is therefore only useful, if it is known beforehand that there are no frequencies at the secondary image position and there is no way to convert the wanted frequency directly to the final location (i.e. with one stage).

Because the intermediate frequency after the first stage is at fixed location, we can also utilize a filter to between the stages to filter out the secondary image. The final intermediate frequency determines how far away the secondary image is and together with the bandwidth of the wanted signal determines the needed selectivity of the filter.

4.4.2.3 Generic (6-mixer)

We showed that the Weaver architecture can cancel the primary image of the first stage, but introduces a secondary image from the second stage. At the request of the thesis committee, the second stage was further analysed to see if the secondary image can be addressed with two extra mixers at the second stage. This brings the total mixer count up to six

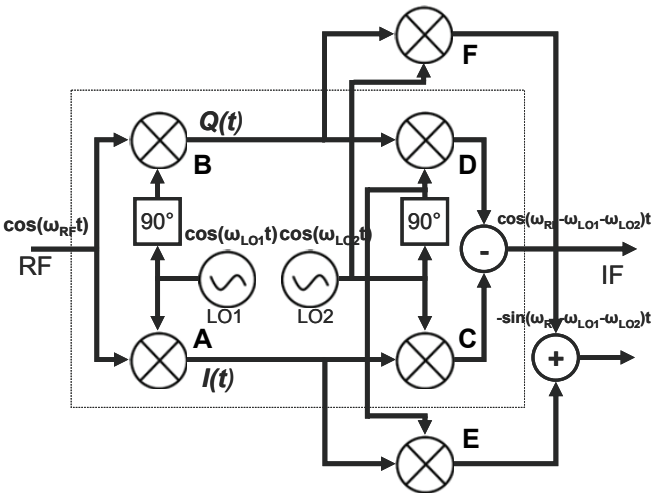


Figure 62. 6-mixer image rejection archicture.

and is therefore called the 6-mixer architecture. This architecture is shown in Figure 62.

The striped box corresponds to the Weaver architecture without low-pass filters as discussed in the previous section. The signals at *RF*, *A*, *B*, *C*, *D* and *IF* therefore corresponds to the signals in Figure 58, Figure 59, Figure 60 and Figure 61.

	$\omega_{RF} - \omega_{LO1} - \omega_{LO2}$	$\omega_{IM1} - \omega_{LO1} + \omega_{LO2}$	$-\omega_{IM2} + \omega_{LO1} - \omega_{LO2}$	$-\omega_{IM3} - \omega_{LO1} - \omega_{LO2}$
$\cos \cdot \cos \cdot \cos$	$+\cos$	$+\cos$	$+\cos$	$+\cos$
$\cos \cdot \sin \cdot \sin$	$-\cos$	$+\cos$	$+\cos$	$-\cos$
$\cos \cdot \cos \cdot \sin$	$-\sin$	$-\sin$	$+\sin$	$+\sin$
$\cos \cdot \sin \cdot \cos$	$-\sin$	$+\sin$	$-\sin$	$+\sin$
$c \cdot c \cdot c - c \cdot s \cdot s$	$+\cos$			$+\cos$
$c \cdot c \cdot s - c \cdot s \cdot c$	$-\sin$			$+\sin$

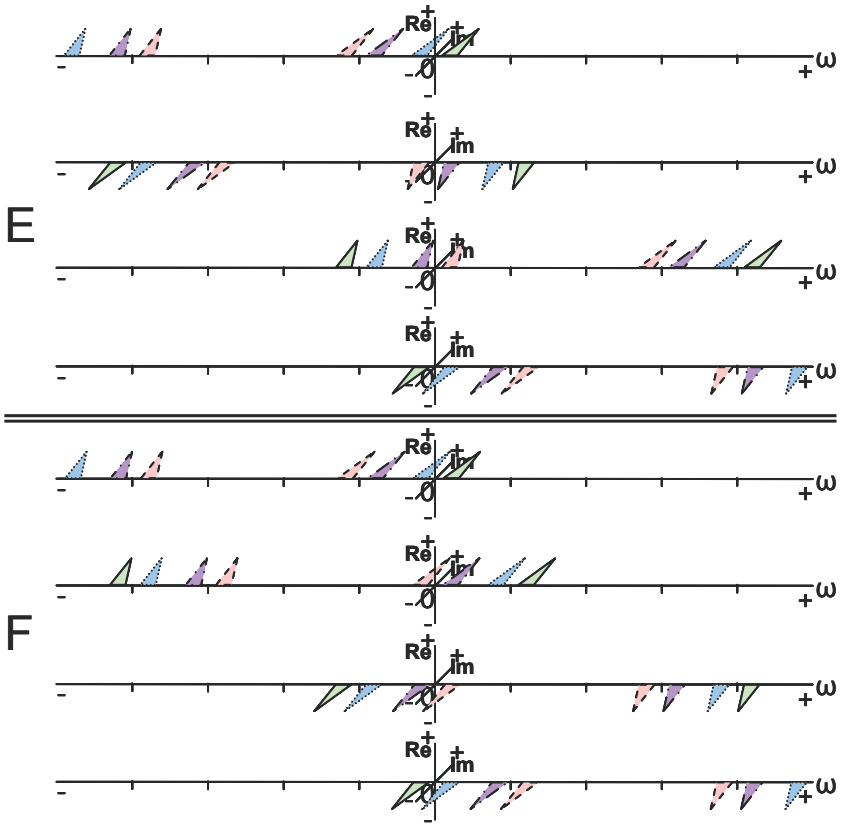


Figure 63. Quadrature signals of the added two mixers.

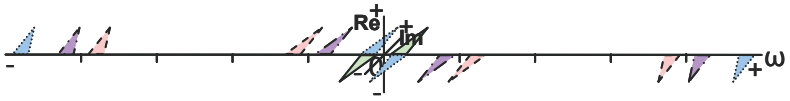


Figure 64. Output signal after adding F to E.

The missing signals *E* and *F* are shown in Figure 63. Again the second and third line cancel each other out, although this time after addition of the signals. The resulting output signal is shown in Figure 64.

Also again there is the secondary image located at the wanted frequency. This time, however, the wanted signal and image are of opposite sign, which is also evident from the mathematical analysis. Unfortunately these are sine wave and therefore 90° out of phase with the output of the Weaver architecture. One of the two outputs must therefore be phase shifted 90°. But then the circuit has the same operating principle as the Hartley architecture for the secondary image, in which case the whole first Weaver stage is unnecessary. Another mixer stage instead of the phase shift will add another image frequency as with the Weaver architecture. Therefore the extra two mixers provide no added value and nothing is gained with the 6-mixer architecture.

It is interesting to note however, that the two outputs of the 6-mixer architecture are quadrature signals. The four mixers in the second stage are differential [35], therefore the matching is less critical giving a better imager rejection (section 4.4.2.5).

4.4.2.4 Digital quadrature IF

A possible quadrature architecture for a digital spectrum analyser is the digital quadrature IF architecture (Figure 65). As will follow from chapter 6, the digital spectrum analyser is most flexible with the analogue-to-digital converter (ADC) as far to the front as possible. Limitations of the ADC dictate its placement after the frequency conversion.

In a quadrature architecture, after the quadrature mixer, we have two signal paths. Thus the ADC can be placed after the quadrature architecture, or two ADCs can be used.

The latter is used. Note that also two anti-alias low-pass filters are needed (see chapter 7).

The resulting digital signals represent the input signal in quadrature. Further processing can be performed on these signals. Both a Hilbert transform can be performed digitally as a second quadrature mixing stage, preceded by an image filter.

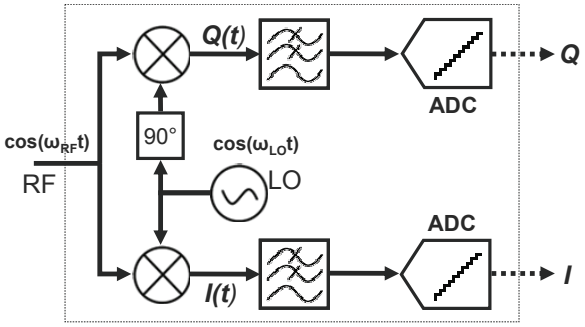


Figure 65. Digital quadrature IF architecture.

4.4.2.5 Image rejection performance

The image is cancelled by adding or subtracting the two signals of a quadrature architecture. The wanted signal is then in-phase and combines and the image is 180° out-of-phase and cancels. The image rejection depends on the mismatch between the gain in each signal path and the phase mismatch, which is determined by the accuracy of the phase difference after the quadrature mixer and the second 90° phase shift.

Thus, the amount of rejection (i.e. image rejection ratio (IRR)) for a quadrature architecture depends on the accuracy of the gain (ΔA/A) and phase shifts (Δθ) [4][34]:

(4-22) $$IRR = \frac{4}{(\Delta A / A)^2 + \theta^2}$$

For typical matching of an integrated design, the image rejection is limited to about 40dB [4][35][36]. This corresponds to about a 0.2dB gain mismatch and a 1° phase imbalance.

The image rejection can be improved by applying gain and phase calibration. For example, [34] uses phase calibration applied to a Weaver architecture to achieve an IRR of 57dB. They expect the IRR to go to over 70dB when adding gain calibration. This would make the quadrature architecture good enough to achieve the spectrum analyser's specified SFDR. However this must be further analysed.

The 6-mixer architecture can be used to decrease the matching requirements [35][36]. It can be shown that using both RF and LO signals in quadrature gives an architecture that is much less sensitive to gain and phase mismatch [36]. It is then used a double-quadrature architecture with poly-phase filters as Hilbert transforms. This improves the image rejection to about 60dB.

Another option is digitizing the quadrature signals and performing the second 90° phase shift in the digital domain with a digital Hilbert transformer. The performance of the Hilbert transformer depends on the number of taps (see section 7.4). For example, a 45-tap FIR Hilbert transformer has a 0.1dB ripple in the frequency response [51]. If a better performance is required, the number of taps can be increased at the cost of extra hardware and time.

4.4.3 Architecture selection

The major problem of the Hartley architecture for use in the spectrum analyser front-end is the second 90° phase shift. This Hilbert transform is difficult to design for a large frequency range. It is normally implemented with combined RC sections giving a polyphase filter [36]. With the use of RC section for the phase shift, the gain mismatch is directly determined by the component mismatch, which is typically relatively large for integrated circuits [4]. The frequency range for the Hilbert transform at IF is 1MHz at maximum. This should be reasonable to achieve [36].

The Weaver architecture or in fact every design with multiple frequency conversion stages has the problem of a secondary image. The secondary image is at a fixed location, which makes it possible to filter it out at the second stage. For example, with the first IF at 20MHz and a second IF at 1MHz, the filter quality must be:

(4-23) $$Q = \frac{f_c}{\Delta f_{BW}} = \frac{20 \cdot 10^6}{2 \cdot 10^6} = 10$$

Typical values for Q are in the range of 1~10 for integrated filters, while external filters are in the range of 100~1000. So this is quite difficult for an integrated filter. The use of a polyphase filter as Hilbert transform is therefore preferred over the Weaver architecture.

The digital IF can improve performance by either a digital Hilbert transformer of which the performance can be controlled or a digital filter which can be very selective but slow. Which of the solutions need more taps and thus time must be further analysed.

4.5 Intermediate frequency selection

Frequency conversion is used to translate the input signal to a fixed intermediate frequency. Without frequency conversion, the entire input range must be supported by the spectrum analyser front-end, which reduces its performance. With frequency conversion, we can choose the location and bandwidth of the intermediate frequency and optimise the following blocks of the front-end. For example, the RBW filter now has a fixed position, which is easier to design and has a better performance than a tuneable bandwidth filter, while the frequency conversion allows a larger input range than a tuneable filter with a 0.5dB noise figure and 70dB dynamic range could support.

The bandwidth of the intermediate frequency is determined by the selected resolution bandwidth. This leaves us with the intermediate frequency location to choose.

A number of possible ranges to choose for the IF can be identified, relative to the RF input range. We can choose the IF above the input range, which is called a high-IF architecture. We can choose the IF in the input range, which we call a mid-IF architecture. The IF can also be chosen below the input range[7]; a low-IF architecture. Finally, the IF can be chosen at DC, the zero-IF architecture. The first three architectures are all super-heterodyne architectures with different IF locations, while the last is also called a homodyne or direct-conversion architecture.

4.5.1 High-IF

In a high-IF architecture, the intermediate frequency is chosen above the input range. The RF input range for the spectrum analyser goes from about DC up to 3 or 6 GHz. We choose the IF at 4 or 8 GHz, We will use the 3GHz input frequency range case from here on. From section 4.3, we know we have two choices in which the intermediate frequency is located above the input range; at the sum and at the difference frequency.

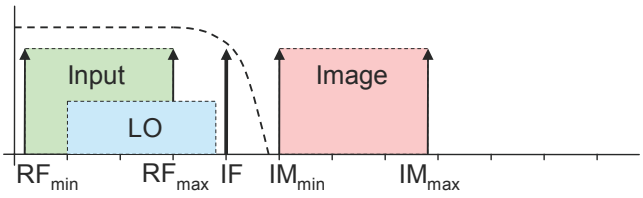


Figure 66. High-IF with the IF at the sum of the RF and LO frequency.

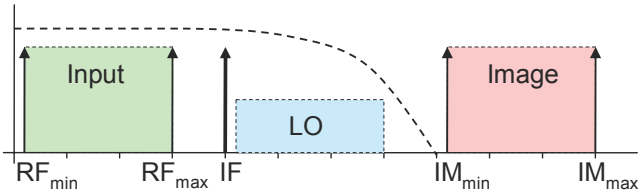


Figure 67. High-IF with the IF at the difference of the RF and LO

The LO must run from 4 to 1GHz, for the IF at the sum of the RF input signal and the LO frequency. The image is two times the LO frequency away ($2\omega_{LO}$) from the RF. As the LO varies, the frequency distance between the RF and its image varies from 2GHz to 8GHz. The image frequency range is thus from 8 to 5GHz. This is shown in Figure 66.

For the IF at the difference of the RF input signal and the LO frequency, the LO must run from 4 to 7 GHz. This is shown in Figure 67. Now the image is two times the IF frequency away ($2\omega_{IF}$) from the RF, which is always

[7] Assuming the input range does not start before the intermediate frequency. If the input range does start before the intermediate range, essentially the mid-IF and low-IF architectures are the same. For more typical radio front-ends (i.e. a mobile phone front-end), the input range is small and at a high frequency. Frequency conversion is then needed to translate the signal of interest to a manageable lower frequency; otherwise the signal could be handled directly without the need for any frequency conversion. Thus a mid-IF solution is not applicable there. For a spectrum analyser front-end, the input range is much larger, therefore making frequency conversion a necessity as directly handling the range is impractical (see section 2.2). Now part of the input range could be handled directly, but still needs to be translated because the intermediate frequency was chosen elsewhere. This also means part of the input signal could be at the same frequency as the intermediate frequency, not requiring frequency conversion for that part of the input signal. In that case we have a mid-IF architecture.

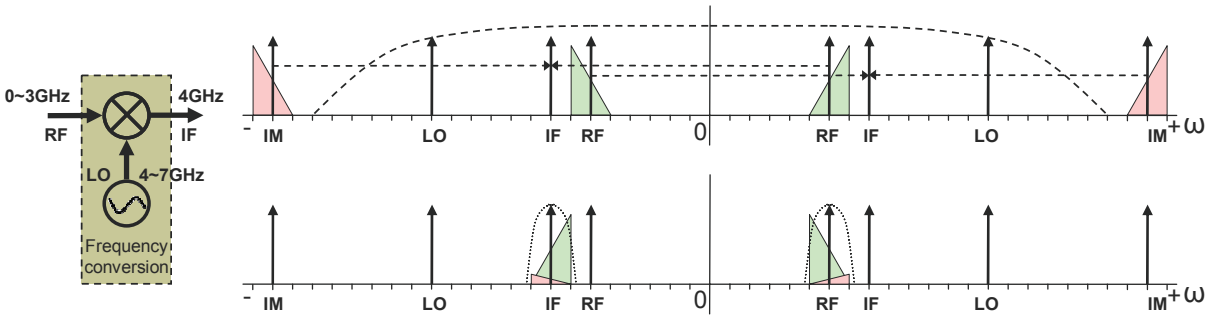


Figure 68. Operation of the high-IF architecture with the IF at the diffence of the RF and LO frequency.

8GHz. The image frequency range is thus from 8 to 11GHz. Note that for these high LO frequencies, the tuning range is relatively small (half a decade).

A big advantage of the high-IF architecture is that the images lie outside of the input range and can therefore be filtered out before the frequency conversion, eliminating the image problem of the previous section. This is the reason many high-end spectrum analysers with an input range up to 3GHz use this approach [7][10].

The IF at the difference frequency has the advantage that the image is always 8GHz away instead of varying. This also gives a larger frequency range between the largest RF and the smallest IM frequency. This makes the filtering of the IM easier. It also has the advantage of a smaller tuning range for the LO, although at higher frequencies. A more important advantage is that the LO higher harmonics give no distortions at the IF. For example, for a LO of 7GHz, the RF is at 3GHz, the IM at 11GHz and the first higher harmonic of the LO is at 14GHz and converting the signals at 10GHz and at 18GHz to the IF. Filtering the IM thus also filters the images of the higher harmonics. (For the IF at the sum, the LO at 1GHz, the RF at 3GHz, the IM at 5GHz and the images of the first higher harmonic at 2GHz and 6GHz for example is a problem). Therefore the IF at the difference frequency of the RF and LO frequency is recommended.

The situation of filtering the image followed by converting the RF to the IF is shown in Figure 68. Next, we will approximate the needed performance of the low-pass filter.

For a fixed low-pass filter, in order not to distort the input range, the corner frequency of the filter needs to lie above 3GHz. We would like a 70dB SFDR. At 8GHz, the start of the images, the attenuation must therefore be at least 74dB to ensure an image falls a 4dB margin under the minimum signal power. This filter is also shown in Figure 67. Note that lowering the RBW, lowers the DANL, but not necessarily the spurious responses, such as images. Assuming a 20dB attenuation per decade per order far away from the corner frequency (see chapter 5), we find:

$$\left(\log \omega_{IM_{\min}} - \log \omega_{RF_{\max}}\right) \cdot 20dB/decade/order = 8.5dB/order$$

(4-24) $$74dB/8.5dB/order \approx 9^{th}\, order$$

$$Q = \frac{f_c}{\Delta f_{BW}} = \frac{3 \cdot 10^9}{5 \cdot 10^9} = 0.6$$

A 9th order filter is fairly large and complex to implement because of stability issues among other things, but not impossible [42].

The filter also partly suppresses feed-through, because of finite isolation, of the input to the intermediate frequency. For a fixed filter, this attenuation is 22dB (*(log 4 - log 3)·20·9*). The isolation then needs to be at least 52dB for a 74dB SFDR.

For a tuneable filter as shown in Figure 69, the image is always 8GHz away. This means the worst case is for the highest frequency:

$$\left(\log \omega_{IM} - \log \omega_{RF_{\max}}\right) \cdot 20dB/decade/order = \left(\log 11 - \log 3\right) \cdot 20 = 11.3dB/order$$

(4-25) $$74dB/11.3dB/order \approx 7^{th}\, order$$

$$Q = \frac{f_c}{\Delta f_{BW}} = \frac{3 \cdot 10^9}{8 \cdot 10^9} = 0.4$$

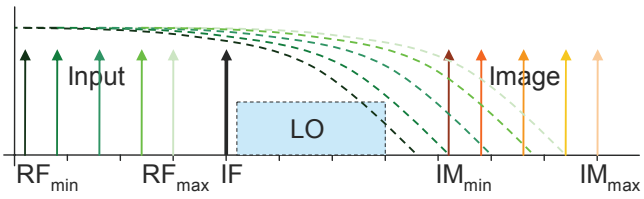


Figure 69. High-IF architecture with a tuneable filter.

A 7th order filter is not much better and it also needs to be tuneable. For the feed-through, the worst case scenario is at a maximum input frequency, which only gives a 10dB (*(log 4 – log 3)*18*4*) attenuation. The needed isolation is therefore at least 64dB.

For a high-end spectrum analyser, the filter requirements are a small price to pay for the solution to the image problem as well as the higher harmonics problem. High quality external filters can be used to achieve the required performance. The high LO frequencies, but at the same time strict phase noise requirements also make it likely that an external LO is needed.

For a low-cost integrated design, this is a problem. Either the filter and LO can be integrated with much difficulty or external components must be used increasing the cost.

The high-IF also has another disadvantage. The signal at the IF can not be processed further without first being down-converted. This down-conversion also has an image, although at a fixed location, because the IF is at a fixed location. It still requires another image filter. If the second IF is at a low frequency, the filter at the second stage needs to be very selective to filter the image. Another option is using multiple down-conversion stages. Both also increase the cost of the design.

4.5.2 Mid-IF

In case of a mid-IF architecture, the IF is chosen somewhere in the input range. The mid-IF solution is attractive, because the frequencies are lower than in the high-IF case, making integration of components easier. But on the other hand, the IF is still large enough that images are reasonably far away. Assume we choose the IF at 1GHz for the spectrum analyser.

One option is to up-convert the input range from 0~1GHz and down-convert the input range from 1~3GHz. However now we have two different situations. For the up-converted frequencies, the image is $2\omega_{LO}$ distance away, but LO becomes 0 when the input frequency reaches 1GHz (the LO is from 1~0GHz). The image therefore increasingly becomes closer as the LO frequency becomes lower. For the down-converted frequencies, the LO is from 0~2GHz. For the input frequency range up to 2GHz, we have $2\cdot\omega_{LO}<\omega_{RF}$. The wanted signal frequency range from 1~2GHz was the image for the 0~1GHz range and now it is the other way around. For frequencies larger than 2GHz, the image is $2\omega_{IF}$ away. Clearly this is not a good solution.

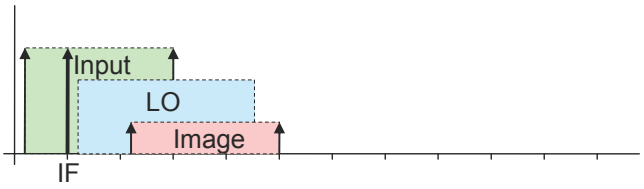


Figure 70. Mid-IF architecture at the difference frequency.

The other option is to always use the difference frequency, with the LO from 1~4GHz (Figure 70). The image is at $2\omega_{IF}$ from the RF. The IM frequency range is 2~5GHz. Because part is in the input range a tuneable filter is needed. For the rest the operation is mostly the same as for the high-IF case, as shown in Figure 71.

A possible problem is the feed-through of the input signal to the IF. The isolation must therefore be larger than 74dB or a pre-filter must be used. However, this filter must be disabled when a signal frequency range of interest is around the IF, again giving the feed-through problem. The pre-filter is therefore not a very elegant solution.

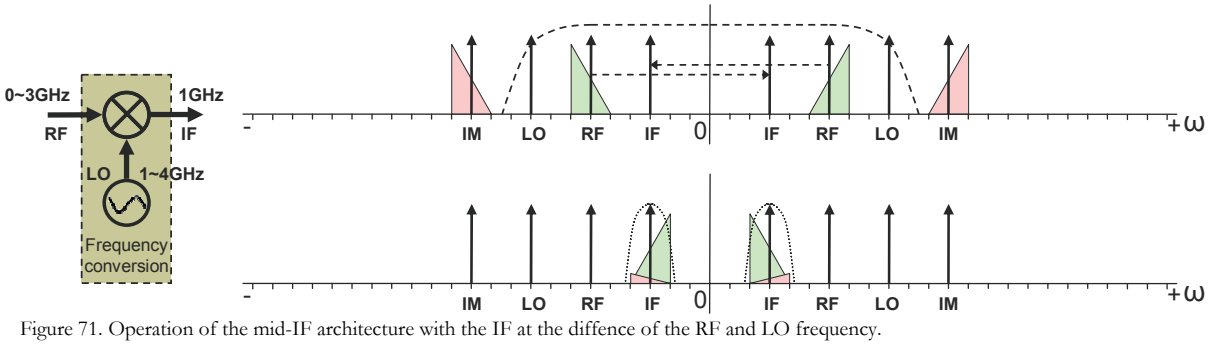


Figure 71. Operation of the mid-IF architecture with the IF at the diffence of the RF and LO frequency.

As with the high-IF case, the signal at IF must be down-converted for processing, requiring more down-conversion stages with image filters.

For the worst case at a 3GHz input frequency:

$$\left(\log\omega_{IM} - \log\omega_{RF_{\max}}\right)\cdot 20dB/decade/order = \left(\log 5 - \log 3\right)\cdot 20 = 4.4dB/order$$

(4-26) $$74dB/4.4dB/order \approx 17^{th}\,order$$

$$Q = \frac{f_c}{\Delta f_{BW}} = \frac{3\cdot 10^9}{2\cdot 10^6} = 1.5$$

We can see the needed filter order is quickly increasing. This filter might still be feasible, but for higher orders, filtering is not an option. In that case only the quadrature architecture is left as an option for image rejection. However, one might as well choose the low-IF solution, as the mid-IF provides no advantages in that case.

4.5.3 Low-IF

In case of a low-IF architecture, the IF is chosen below the input range. Ideally, the spectrum analyser's input frequency range start at zero. The IF is therefore chosen to be very low. This also makes it possible to use the signal at the IF directly for further processing instead of requiring multiple conversion stages.

The maximum RBW is 1MHz, so let's use an IF of 1MHz with the RBW centred around it. For an IF of 1MHz, the $2\cdot\omega_{LO}=\omega_{RF}$ is reached at a 2MHz input frequency. Thus the low-IF architecture always uses the difference of the RF and the LO frequency as IF. For an LO from 2MHz ~ 3.001GHz, the RF is translated to the negative IF. The image is always 2MHz away, from 3MHz to 3.002GHz. Alternatively, the RF is started from 2MHz and the LO is from 1MHz ~ 2.999GHz. The RF is translated to the positive IF and the IM is from DC~2.9998GHz. This is shown in Figure 72.

It is clear that filtering at a low-IF is not viable. For an IF of 1MHz, the quality factor of the filter at 3GHz must be:

(4-27) $$Q = \frac{f_c}{\Delta f_{BW}} = \frac{3\cdot 10^9}{2\cdot 10^6} = 1500$$

Besides even being problematic for an external filter, it also needs to be tuneable over the complete input frequency range.

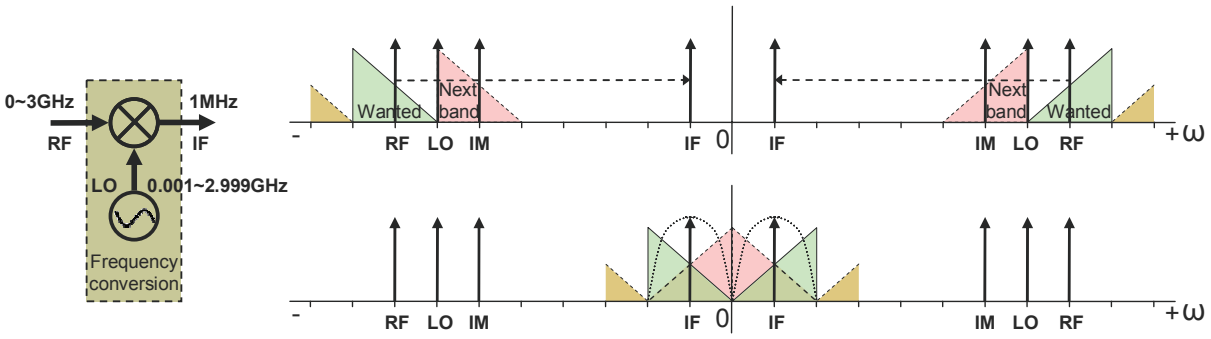


Figure 72. Low-IF architecture with an image problem. Signal before conversion on top, after conversion at the bottom.

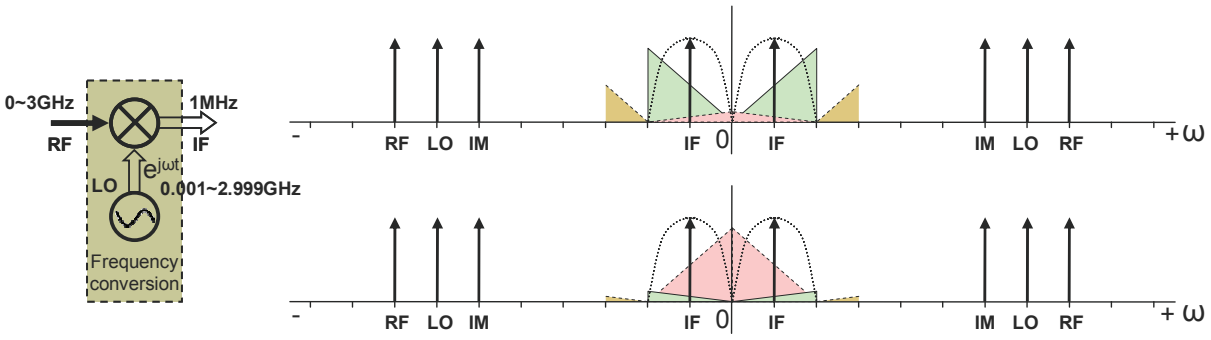


Figure 73. Low-IF quadrature architecture with image rejection. Selection of the upper sideband on top, the lower sideband at the bottom.

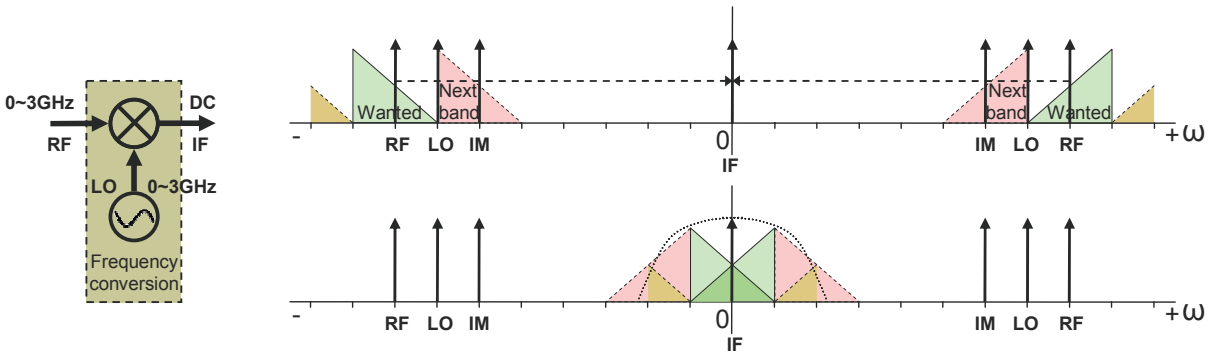


Figure 74. Zero-IF architecture with no image problem, but overlapping sidebands.

Instead a quadrature architecture is used for image rejection in case of a low-IF architecture. Unfortunately, the image rejection is limited by matching to about 40dB, possibly improvable to about 60dB (section 4.4.2.5). This is not enough for the required 70dB dynamic range.

The operation of the quadrature low-IF architecture is shown in Figure 73. Note that the upper-sideband or lower-sideband or both can be chosen. Thus although we need two mixers and a second 90° phase shift, we also have two bandwidths at once.

4.5.4 Zero-IF

In case the IF is zero, the LO frequency is equal to the RF, because $\omega_{IF}=\omega_{RF}-\omega_{LO}$. The LO frequency range is thus from 0~3GHz, the same as the input range. As the RF and LO frequency are the same, $2\cdot\omega_{LO}$ is always greater than ω_{RF} and the image is at $2\omega_{IF}$. But the IF is zero and thus is the image the same as the wanted signal. This is the main advantage of the zero-IF architecture; there is no image problem.

However, this is the case for a single RF. For a frequency band around RF, the two sidebands still overlap as shown in Figure 74.

In case of a spectrum analyser the frequency band around RF is the RBW and represents a “single” frequency in the resulting spectrum. Therefore the overlap of the sidebands is not a problem.

4.5.5 Choice of IF

As explained the high-IF architecture is not very suitable for integration, especially because of the high LO frequencies. Furthermore, the need for (probably multiple) down-conversion stages after the IF make the high-IF architecture costly. It is therefore not recommended.

The mid-IF architecture trades a higher order image rejection filter for lower LO frequencies. This makes integration easier, but the IF in the input range gives feed-through interference problems. Also further down-conversion is still needed.

For a low-IF architecture filtering of the image is not an option and a quadrature architecture must be used. Unfortunately, the quadrature architecture can not provide enough dynamic range for the spectrum analyser front-end requirements.

The zero-IF architecture comes out preferable, because it has no image problem. This is because in the case of an (analogue) spectrum analyser, the IF represents a “single” frequency of which we would like to know the power.

For a zero-IF architecture, the RF is translated to DC and the RF band is translated around DC. Unfortunately, there are several causes of distortions at and around DC. These are further discussed in section 4.6.

Note that the zero-IF architecture requires a LO that is tuneable from 0~3GHz, therefore over many decades. Furthermore, the step size must be as small as 10kHz for the smallest RBW. Together with the high phase noise requirements, the design of the LO therefore remains a challenge.

4.6 DC offset

It was found that a zero-IF architecture provides the significant advantages of manageable local oscillator frequencies without an image problem. This comes at the cost of DC offset problems. The frequencies close to zero are now part of the signal at the intermediate frequency. This distortion signal can mask a small signal of interest. This smallest signal we want to measure is -90dBm (at a 1MHz RBW), the DC distortion must therefore be under -94dBm in order not to distort the (smallest) signal of interest. If we select a smaller RBW, we would like the noise floor to drop, so we can measure smaller signals (although this is limited by the phase noise close-by). Therefore we would like the DC offset to drop as well with a smaller RBW. For the smallest 10kHz RBW, the noise level must stay under -114dBm and thus the DC offset preferably also.

There are several causes that generate distortion products around DC, which we will introduce first. Next we will discuss a number of possible solutions.

Note that heterodyne architectures (low-, mid- and high-IF) also generate DC offset distortions. For example interferers at the local oscillator frequency if this frequency is in the input range down-convert to DC. Other causes also still contribute. However, as the intermediate frequency is now further away from DC, this offset is not part of the signal to be measured and can be filtered more easily.

4.6.1 Causes

Two main causes of DC offsets are due to the finite isolation of the two mixer paths. As found in section 3.2.3 and 4.2, two multiplied signals generate sum and difference frequencies and two equal frequencies therefore include a DC signal. This is the case if a signal is multiplied with itself, also called self-detection. A mixer multiplies two signals with each other and thus we can have leakage of the LO signal into the RF path and leakage of the RF signal into the LO path as shown in Figure 75.

Besides self-detection there are also DC offset components because of noise and because of non-linearities.

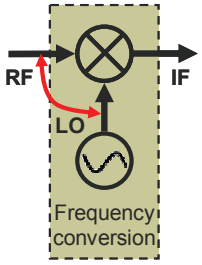


Figure 75. Self-detection

4.6.1.1 LO self-detection

The local oscillator signal consists of a fixed amplitude single frequency. Mixed with itself (*LO·LO*) it will generate a single DC component. These DC component will have a fixed amplitude as long as the local oscillator amplitude does not change and the LO leakage remains constant.

4.6.1.2 RF self-detection

The RF input signal is a wide-band signal ranging from zero to 3/6 GHz with varying amplitudes. Multiplied with itself (*RF·RF*), this results in a broadband signal with twice the bandwidth. If we assume approximate equal amplitudes across the RF band, we see from Figure 76 that most RF self-detection energy is located around DC.

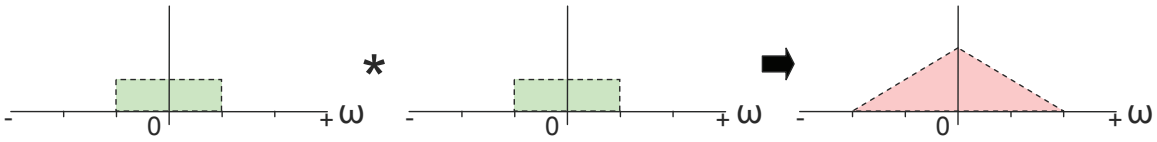


Figure 76. Convolution of the input spectrum with itself.

4.6.1.3 Thermal noise

Thermal noise causes small signal variations present at all frequencies including DC. In section 2.3.7 we found the noise to be at least -174dBm/Hz to which the noise of the front-end is added. The noise level was specified to remain under the input amplitude range at all used bandwidths and therefore the DC offset because of thermal noise also remains under the input amplitude range.

4.6.1.4 Flicker (1/f) noise

A noise source that was not discussed in section 2.3.7 is flicker noise, also called 1/f noise, which as the name implies is dependent on the frequency. This noise becomes larger with lower frequencies, with the actual value depending on the process technology and design.

Flicker noise arises from random trapping of charge at the oxide-silicon interface of MOSFETs (as well as other mechanisms) [3][4] and can be represented as:

(4-28) $$\overline{V_n^2} = \frac{K}{WLC_{ox}}\frac{1}{f}$$

With *W*, *L* and C_{ox} MOSFET parameters, *K* a process-dependant parameter and *1/f* the frequency dependence.

The 1/f noise is often specified as the corner frequency at which the flicker noise intersects with the thermal noise floor. The flicker noise corner frequency typically lies about 30dB under the maximum supported frequency (The corner-frequency is about 500kHz to 1MHz for sub-micron devices [4]. The maximum frequency of the process is about 10 times higher than the maximum “workable” (supported) frequency in order to achieve some gain etc.). Not all blocks of the spectrum analyser operate at a high frequency: only the LNA, LPF and mixer do, the IF after the mixer is a relatively low frequency. The maximum frequency of the first block is 3GHz giving a 1/f corner of 3MHz. For the spectrum analyser front-end the 1/f noise of the mixer will be the most problematic. This is because 1/f noise generated before the frequency conversion is converted to a different frequency and everything in the input range above 3MHz (which can be converted to DC) has no noticeable 1/f noise. The mixer however, operates at a high frequency, but the IF frequency band at the output of the mixer is from DC to the selected RBW corner-frequency. The maximum frequency at IF is 0.5MHz for a 1MHz RBW. The 1/f corner frequency is then at 0.5 kHz. Note that if we assume passive filters are used, then they contain no transistors and generate no 1/f noise.

4.6.1.5 Second order distortion

In section 2.3.8, we found that second order distortion products also include DC components. The second order linearity of the front-end must therefore also be good enough to reduce the DC offset. We already found that the IP_2 requirement is over 54dBm for distortion products (including DC) to remain under the input amplitude range.

4.6.2 DC removal

As the noise and non-linearity distortions are specified to remain below the input amplitude range, the DC offset distortion depends on the isolation of the local oscillator and input signal paths. If this isolation is good enough to reduce the signal to below -94dBm (at 1MHz RBW), this is preferred. If not other solutions must be applied.

There is the exception of the 1/f noise, which was not specified, but also is not dependent on the isolation. This is discussed separately.

4.6.2.1 Isolation

The preferred method to reduce the DC offset is by improving the isolation of RF and LO signal path, because this method does not alter the signal of interest. This isolation can be done by proper design and layout. For example, integrating the frequency generation on-chip improves the isolation and/or using on-chip amplifier for the LO signal close to the mixer [32][33], but this is not always possible. Orthogonal RF and LO signal wiring also reduces coupling [32].

Typical LO-to-RF isolation for receivers is 50 to 70dB [32][33]. Typical LO power is in the range from -10dBm to 10dBm [32]. A -10dBm LO signal results in about -70dBm distortion at the input. The LO signal possibly can be smaller, depending on the implementation of the mixer. This is not further discussed. A maximum -20dBm RF signal gives a distortion of about -80dBm at the LO signal path. With design and layout optimization only, the result can be improved result by about 30dB [32], giving a low enough distortion at the RF input of -98dBm [32][33] in case of a 1MHz RBW.

An attenuation of 60dB only because of isolation is also suggested by [4] (p.132), [31]. The performance of typical receivers includes high-pass filtering besides isolation and should therefore achieve an isolation better than 60dB, but it is still only 50 to 70dB. This is because those receivers typically also include about 30dB of gain before the mixing, which also amplifies the DC signal and is counterbalanced by the filtering.

4.6.2.2 High-pass filter

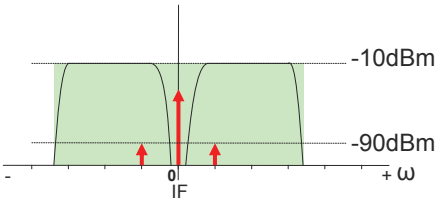


Figure 77. High-pass DC offset filtering.

With a stepping local oscillator frequency (when sweeping over an input frequency range) and a relatively stable RF, it is known that the DC offset because of self-detection does not change at least until the next step. Therefore, it is known the DC offset is not varying and can be filtered with a high-pass filter. A high-pass filter is an obvious solution to the DC offset problem (Figure 77). The wanted signal is also centred around DC in case of a zero-IF architecture or close to DC in case of low-IF. The corner frequency must therefore be very low giving slow transient and long settling times. Note that always part of the wanted signal is removed because of the filtering.

We can estimate how long we have to wait until the distortion signal is attenuated enough. Next we will discuss the loss of part of the wanted signal.

4.6.2.2.1 Measurement time

For a simple first order RC low-pass filter, the impulse response is:

(4-29) $$V(t) = V_0 e^{\frac{-t}{RC}}$$

This exponential decaying response is shown in Figure 78. If we assume a LO-RF isolation of at least 60dB (see section 4.6.2.1), we need to wait for the filter to reduce the signal power the remaining 44dB (-10dBm LO signal and -114dBm noise level at 10kHz needs 104dB isolation).

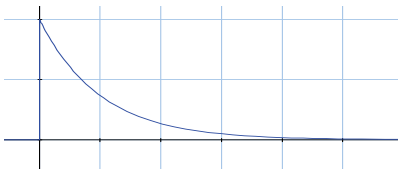


Figure 78. First order low-pass filter impulse response.

For a zero-IF architecture, the smallest RBW of 10kHz is centred around DC, which means a somewhat above 5kHz corner-frequency. Let's allow 1/10th of the bandwidth for DC removal. The corner frequency of the high-pass filter is therefore at 0.5kHz.

The time before the signal falls 44dB is therefore:

(4-30) $$10^{-44/20} = 1e^{-t\cdot\tau} \Rightarrow t \approx 5.1\tau$$
$$\tau = 2\cdot 10^{-3} \Rightarrow t \approx 0.010$$

Thus, each measurement needs 10ms settling time for the high-pass filter. For a sweep with a 1000 measurement points, this adds 10 seconds to the sweep time. As the high-pass filter can be combined with the resolution bandwidth filter, only the processing time adds to this. Looking at the sweep time results of available spectrum analysers (section 2.5), the shortest is much shorter at typically 1ms probably at the largest RBW, while the longest is as much as a 1000s, but probably for much smaller RBWs in the Hz region. A 1MHz RBW for our spectrum analyser front-end would give a 100ms sweep time (a 100 times less), while a 10Hz RBW gives 1000s. A 10 second sweep time at a 10kHz can therefore be acceptable.

4.6.2.2.2 Signal loss

As said, the filtered DC signal also includes part of the wanted signal. This filtered part might be important, containing the largest signal in the bandwidth. In case of a spectrum analyser, the RBW represent conceptually "one" frequency of which we primarily want the largest signal power. We can exploit the inherent frequency instability (phase noise) of the local oscillator to ensure the signal that would convert to DC is distributed around DC. Thus if the largest signal happens to be at DC, part of this signal still ends up in the pass-band, although attenuated, as shown in Figure 79.

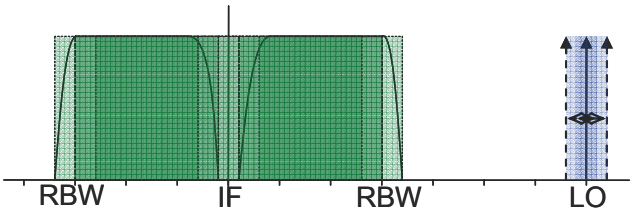


Figure 79. Frequency instability distributed filtered signal band of interest.

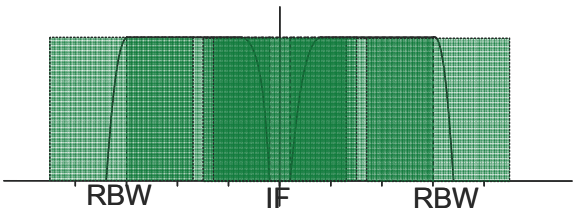


Figure 80. Combined result of three measurement of the signal of interest, each halve the RBW apart.

If the frequency instability is limited (good local oscillator), a large part of the signal of interest that ends up at DC is excluded from the measurement as shown in the figure. As an alternative one could use a LO steps of halve the RBW and combine the results of three measurements as shown in Figure 80. The darkness of the green indicates how many times that part of the original signal is accounted for in the combined result of three measurements, each halve the RBW apart. As seen the area around DC is measured un-attenuated twice, while the dark green parts are measured three times and the light green parts once. It is clear that also part of the signal outside of the RBW is included in the result. Thus the signal at DC is taken into account better than in Figure 79, but at the cost of including part of the signal of the previous and following RBWs.

4.6.2.3 Frequency planning

Another solution is to use multiple local oscillator frequencies, which convert the signal down to DC in multiple steps (fractions of the total wanted down-converted frequency). However, as found in section 4.4, each stage introduces new images, which must be filtered or rejected with a quadrature architecture. This defeats the purpose of choosing a zero-IF architecture.

With a single down-conversion step, we can use a local oscillator frequency which is a multiple of the intended frequency (i.e. double or triple), together with a frequency divider close to the mixer. This increases the isolation as the final local oscillator frequency only exists close to the mixer in a small area. As an added benefit the divider can be used to generate 90° phase shifted I and Q local oscillator signals for a quadrature mixer [32]. Of course, this requires higher LO frequencies; a multiple of the already high (up to 3GHz) input frequency range. This makes the local oscillator more difficult (also see section 2.3.7.3 for the phase noise requirement).

Another option that works by using different LO frequencies, is using a dual mixer architecture with complex square waves [33]. This architecture also uses multiple down-conversion steps. However instead of using fractions of the wanted frequency, it uses complex square waves with energy at different frequencies except the wanted and which combine after multiplication to form a square wave of the wanted frequency [33]. This solution is likely to give the same image problems as the fractional one. The result may be better because one could use complex waves with many frequency components each having less energy than the frequencies of a fractional approach. Under the assumption there are only a few interferers giving an image problem, the average contribution giving a problematic image is less likely.

4.6.2.4 Calibration

In case the DC offset can not be removed adequately, it can be compensated for by using calibration. For example, the DC offset can be measured without an input signal and be subtracted from the measurement result with an input signal [31]. Of course this is at the cost of a longer measurement time needed. Related to this is common-mode rejection by using offset compensation in a feedback path, which can be made faster by “predicting” the DC offset.

4.6.2.5 1/f noise

For a spectrum analyser, we want the noise floor to drop when we choose a smaller RBW. This corresponds to a fixed non-frequency-dependent noise figure. If we see 1/f noise as part of the noise figure, this means the 1/f noise of the front-end must also drop

for a smaller RBW. Thus the 1/f noise is most stringent for the smallest RBW as this is closest to DC[8]. The 1/f noise adds to the thermal noise and noise figure. However this means the noise floor increases when getting closer to DC in the RBW. In case a high-pass filter is used for DC removal, the 1/f noise will not influence the result if the 1/f corner frequency is below the cut-off frequency of the high-pass filter. With the cut-off again at 1/10th of the RBW, this gives a low 1kHz 1/f corner frequency for a 10kHz RBW. This is all shown in Figure 81.

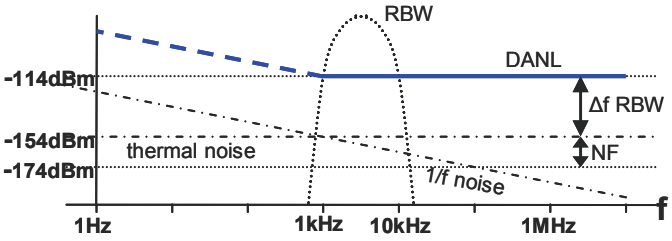


Figure 81. Noise floor with 1/f noise at low frequencies, 1/f noise filtered by band-pass RBW filter.

With a 10kHz RBW and a 1kHz high-pass filter corner frequency filtering the 1/f noise, the noise level is *kT+NF+RBW = -174dBm+20dB+40dB = -114dBm*.

Because the needed 1/f corner-frequency is low, it is probably not attainable. What happens if the 1/f noise is not completely filtered by the high-pass filter? It increases the DANL. If the high-pass corner frequency is always chosen as 1/10th (-10dB) of the RBW corner frequency, the amount of 1/f noise that is added depends on the selected RBW. The 1/f noise at 1/10th of the RBW bandwidth is the highest, because the 1/f noise of lower frequencies is filtered out and for higher frequencies it falls with 1/f. We use the highest 1/f noise as a worst-case figure for the whole bandwidth.

Now if we select a smaller RBW, the 1/f noise is thus larger. Section 2.3.7 explained the noise level is dependent on the bandwidth of the signal, which is determined by the bandwidth of the RBW filter. This dependence is proportional on the frequency, while the 1/f noise is proportional to the inverse of the frequency. They therefore cancel each other out **when selecting a smaller RBW bandwidth**. This means the DANL no longer drops when selecting a smaller RBW.

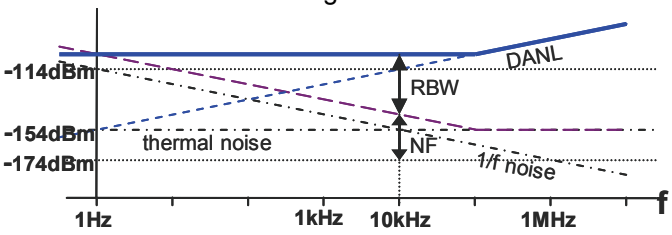


Figure 82. Selection of RBW filter's bandwidth giving the noise floor.

Figure 82 shows the DANL versus the selected RBW frequency on a logarithmic scale. The blue striped line shows the DANL without 1/f noise, which drops for lower RBW frequencies. The thermal noise normalised to a 1Hz bandwidth with the NF added (20dB without 1/f noise) is also shown. This noise is not frequency dependent. Let's assume we have achieved a certain 1/f noise corner frequency of 10kHz. The 1/f noise with a corner of 10kHz is frequency dependent. However, the figure shows the frequency of the selected RBW. As explained above, the 1/f noise is highest at 1/10th of the RBW. The 1/f noise line is therefore shifted 10dB to the right in the figure and shown in the purple line, where it is combined with the thermal noise.

Thus, every RBW below 10 times the achieved 1/f corner frequency will have a fixed noise level and this noise level will therefore not become smaller by selecting a smaller RBW. The resulting noise floor (DANL) is then indicated by the blue line.

The 1/f noise is dependent on the process technology used as well as the highest needed frequency, because this dictates the needed transistor size, which in turn gives a certain 1/f noise. Larger, slower transistors have less 1/f noise. [4] Another technique that exists is turning the transistor off periodically, thereby "resetting" the noise and lowering the 1/f noise about 8dB [37][38].

Note that if multiple frequency conversion steps are used, only the last mixer's 1/f noise ends up around DC. If this is chosen to be a small frequency compared to previous steps, a lower 1/f corner frequency can be achieved giving less 1/f noise. [33]

[8] Note that in contrast to phase noise which also has a 1/f dependence, flicker noise is not dependant on the maximum signal power. Therefore we would like the flicker noise to scale down with a smaller RBW to view a smaller signal, while the phase noise will already be at least the dynamic range smaller than the signal we want to measure (assuming that the signal is the maximum signal in the RBW).

The DC offset because of finite isolation is preferable so small no further DC removal is needed. The same applies to the 1/f noise. In the likely case this is not achievable, DC removal must be used. Frequency planning, except for using a multiple of the LO, is not the best option because it defeats the purpose of using a zero-IF architecture. It is also only useful for the DC offset resulting from self-detection. The use of calibration must be further analysed.

An obvious choice is a low-pass filter. A high-pass corner frequency at $1/10^{th}$ of the RBW bandwidth is proposed. This comes at the cost of a longer sweep time. If a 1/f noise corner frequency of 1kHz is achieved, the DANL will drop when going from 1MHz to a 10kHz RBW. Otherwise, the DANL will be fixed at an RBW lower than 10 times the 1/f corner frequency.

4.7 Summary and Conclusion

This chapter discusses the frequency conversion needed to support the complete input frequency range. Frequency conversion is performed by multiplying the input signal (RF) with a tuneable local oscillator (LO) frequency. Unfortunately, multiplication of two sinusoids results in components at the sum and difference frequencies. One is the wanted signal, the other is called the image.

4.7.1 Summary

A mathematical notation of the multiplication of sinusoids is presented, based on sines and cosines with their sign versus frequency. In effect the notation shows the frequency and a 0°, 90°, 180° or 270° phase. Also, three graphical representations were presented which show an increasing amount of information about the signal. The single sided spectrum of only shows the existence of a signal at a certain frequency. The double sided spectrum also shows the direction of rotation in the complex plane, while the complex plane versus frequency includes the phase relation between signals.

Next, quadrature signals are introduced, which are two signals which are 90° out of phase. A Hilbert transform perform a 90° phase shift for all frequencies. The use of quadrature signals has two important advantages. The first is that the amplitude and phase can be determined instantaneous by using Pythagoras's formula. The second is that positive and negative frequencies can be differentiated, which is useful for the selection of a single sideband or for image rejection.

Quadrature signals are created by duplicating the input signal and performing the Hilbert transform on one of them. The input signal is called the in-phase (I) signal, the 90° phase shifted signal is called the quadrature (Q) signal. For quadrature signals, the positive frequencies of the spectrum are represented by $I+jQ$ and the negative by $I-jQ$. The spectrum of a real signal is symmetrical around zero and the origin of the Hilbert transform is also. However, if we perform a frequency shift of the quadrature signals, the origin shift to a positive or negative frequency. If shifted up, frequency components that were negative now lie at the positive side. For a second Hilbert transform the origin is again at zero. Normally the negative frequency components have changes -90° twice and the positive frequency components +90° twice. All frequencies thus have had a 180° phase shift and added to the in-phase signal they cancel each other. The translated frequency band first had a +90° phase shift as a negative frequency and then a -90° a positive frequency. Thus that frequency band has had a 0° phase shift. When the two signals are added, the 0° phase shift frequency band remains. If they are subtracted this frequency band cancels and the rest remains.

Thus, by using a quadrature architecture, we are able to make a selection between the upper or lower side band. One contains the wanted signal, the other the image and is rejected. Combined with frequency conversion, we can choose the origin for the frequency selection, where we can choose all frequencies above or below that origin.

Two signals in quadrature can be generated by the mixer. A mixer that performs this is called a quadrature mixer. The mixer also performs a frequency translation. The quadrature mixer thus combines the frequency translation with the generation of quadrature signals by a 90° phase shift between the two signals.

We can identify a number of different situation of the relation between the RF and LO frequency. Looking from the RF signal, the sum or difference of the RF and LO frequency is taken as the IF. The different possibilities are indicated in Figure 49.

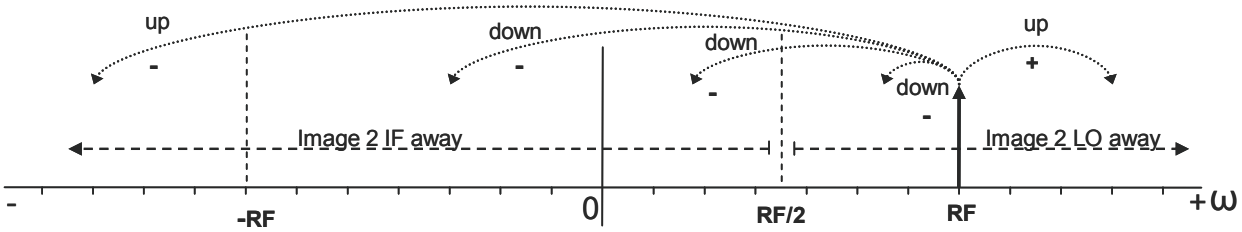


Figure 83. The five different IF sections.

Image rejection can be achieved by filtering or by a quadrature architecture. They cab also be used both. Four different quadrature architectures are analysed. The Harley uses a quadrature mixer followed by a Hilbert transform in the quadrature signal path. The Hilbert transform can be difficult to implement. The Weaver architecture uses two cascaded quadrature mixers. The second quadrature mixer stage gives a secondary image problem. This image must be filtered, which results in a difficult filter with a Q of 10 in case of the spectrum analyser. The 6-mixer architecture adds two mixers to the secondary stage, which try to address the secondary image. Unfortunately, it is shown that they can not, but the resulting double-quadrature mixer in the second stage does relax the matching requirements. Typical quadrature architectures achieve a 40dB image rejection limited by the matching. The use of calibration or the double-quadrature mixer improves the image rejection to 60dB.

The I and Q signals can be digitized after the quadrature mixer instead of after the recombination of the signals. This gives a digital quadrature IF architecture. The second 90° phase shift or quadrature mixer is then performed in the digital domain. The performance in the digital domain can be controlled by the allowed hardware and time.

The knowledge of the quadrature architectures and the image locations allows us to choose an IF location. The IF above the input range is called a high-IF architecture. The IF in the input range is called a mid-IF architecture. The IF can also be chosen below the input range; a low-IF architecture. Finally, the IF chosen at DC is the zero-IF architecture.

The high-IF architecture is not very suitable for integration, especially because of the high LO frequencies. Furthermore, the need for (probably multiple) down-conversion stages after the IF make the high-IF architecture costly. It is therefore not recommended.

The mid-IF architecture trades a higher order image rejection filter for lower LO frequencies. This makes integration easier, but the IF in the input range gives feed-through interference problems. Also further down-conversion is still needed.

For a low-IF architecture filtering of the image is not an option and a quadrature architecture must be used. Unfortunately, the quadrature architecture can not provide enough dynamic range for the spectrum analyser front-end requirements.

The zero-IF architecture comes out preferable, because it has no image problem. This is because in the case of an (analogue) spectrum analyser, the IF represents a “single” frequency of which we would like to know the power.

For a zero-IF architecture, the RF is translated to DC and the RF band is translated around DC. Unfortunately, there are several causes of distortions at and around DC, which are LO self-detection, RF self-detection, thermal noise, 1/f noise and second order distortion products.

As the noise and non-linearity distortions are specified to remain below the input amplitude range, the DC offset distortion depends on the isolation of the local oscillator and input signal paths. Typical isolation is 50~70dB, which is not good enough to reduce the signal to below the DANL of -94dBm (at a 1MHz RBW). However design and layout optimization can improve this.

There are different methods for DC removal, which are briefly explained. A high-pass filter can be used. Assume $1/10^{th}$ of the RBW bandwidth is used for DC removal. This gives a 10ms delay for the filter’s impulse response to provide enough attenuation. For a 1000 measurements and assuming the LO settling time and processing time is much less, this gives a 10 second sweep time for a 10kHz RBW. That is acceptable. The sweep

time is 100ms for a 1MHz RBW. Phase noise of the local oscillator would probably ensure that a possible large input signal ending up at DC is distributed to the pass-band of the high-pass filter. Other methods of DC removal are frequency planning and calibration.

Unfortunately 1/f noise is not dependent on the isolation. The 1/f noise is often specified as the corner frequency at which the 1/f noise intersects with the thermal noise floor. The corner frequency typically lies about 30dB under the maximum frequency. For the spectrum analyser front-end the 1/f noise of the mixer will be the most problematic.

The DC offset because of finite isolation is preferable so small no further DC removal is needed. The same applies to the 1/f noise. In the likely case this is not achievable, DC removal must be used.

An obvious choice is a low-pass filter. If a 1/f noise corner frequency of 1kHz is achieved, the DANL will drop when going from 1MHz to a 10kHz RBW. Otherwise, the DANL will be fixed at an RBW lower than 10 times the 1/f corner frequency.

4.7.2 Conclusion

Frequency conversion is always based on mixing, but reveals different aspects to consider depending on the choice of IF and mixer.

A quadrature signal represents the original signal in the complex domain. With the use of quadrature signals we can instantly determine their amplitude and phase, even in case of a zero intermediate frequency. Another important use is the ability to select the upper or lower sideband around the LO frequency. With the wanted signal in one and the image in the other, we can perform image rejection. Quadrature signals are 90° out of phase. The Hilbert transform perform a 90° phase shift for all frequencies.

We can generate a quadrature signal with a quadrature mixer. The quadrature mixer combines frequency conversion with the generation of quadrature signals. This allows us to choose the origin for the frequency selection, where we can choose all frequencies above or below that origin.

The frequency location of the image is determined by the relation between the RF and LO frequency. If the LO frequency is higher than half the RF, the image is a distance of two times the LO frequency from the RF. If the LO frequency is lower, the image is a distance of two times the IF from the RF.

Image rejection can be achieved by filtering or by a quadrature architecture. They can also be used both. The Hartley uses a quadrature mixer followed by a Hilbert transform in the quadrature signal path. The Weaver architecture uses two cascaded quadrature mixers. The digital quadrature IF architecture digitizes the signal after the first quadrature mixer stage. The Hartley architecture is preferred because a Hilbert transform over 1MHz, is easier than the needed filter for the secondary image of the Weaver architecture. The digital architecture gives more control over the achieved image rejection.

The image rejection is typically 40dB because of mismatch and can be improved to 60dB. The image rejection is not enough for the required 70dB SFDR.

For the choice of the IF location, the zero-IF architecture comes out preferably, because it has no image problem. For low-IF, a quadrature architecture is needed which does not provide enough dynamic range. For the mid-IF and high-IF solutions, the high frequencies and the need for multiple down-conversion stages is prohibitive.

The zero-IF is distorted by DC offsets caused by LO self-detection, RF self-detection, thermal noise, 1/f noise and second order distortion products. Thermal noise and second order distortion products are specified to remain below the input amplitude range. Typical isolation provided 50~70dB of attenuation for the self-detection products. The remainder and the 1/f noise are best removed by a high-pass filter. The high-pass corner frequency is chosen at $1/10^{th}$ of the RBW bandwidth. Possible input signals that are filtered out are likely to be distributed into the pass-band by the LO phase noise. For the smallest 10kHz RBW, the sweep time with DC removal is about 10s. If a 1/f corner frequency of 1kHz is achieved, the DANL will drop when going from 1MHz to a 10kHz RBW. Otherwise, the DANL will be fixed for an RBW lower than 10 times the 1/f corner frequency.

Chapter 5

5 Filters

In the architectures of chapter 3 we identified a number of filters and presented some preliminary specifications for these filters. In this chapter we will go into further detail, especially in the case of the important resolution bandwidth filter.

Note that we will only discuss passive filters solutions in this chapter as they give a good linear performance without much noise since only linear devices such as capacitors and inductors are used. Active filter solutions are left to be evaluated as future work.

We will first evaluate an ideal filter in section 5.1 and find that it is not realisable. The ideal filter can be approximated. We identify some filter characteristics in section 5.2, which define how well an actual filter approximates the ideal filter. Together with the requirements from chapter 2, we will describe which performance characteristics are important for the low-pass pre-filter and the resolution bandwidth filter from the front-end architecture of chapter 3, repeated in Figure 84.

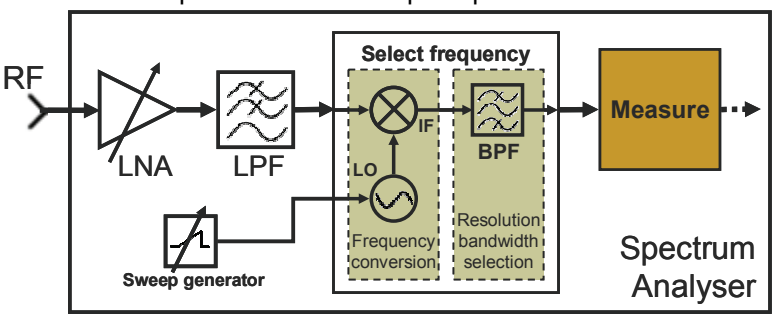


Figure 84. Spectrum analyser front-end architecture.

A number of common higher order passive filter designs exist of which we will determine their applicability for our filters in section 5.5. As will turn out, there is a selectivity versus speed trade-off with an optimum for a Gaussian filter. Therefore the Gaussian shape is analysed in section 5.6.

5.1 Ideal

A filter is used to separate elements depending on criteria. An electronic filter performs a selection based on the frequency of signals. Certain frequencies are passed while others are stopped. We call a range of frequencies a band. The band at which frequencies are passed is the pass-band and when the frequencies are blocked it is the stop-band. The frequency separating the bands is the corner (or cut-off) frequency. The width of a specified band of a filter (mostly the pass-band) is of course called the filter's bandwidth.

Refer to section 2.1 for a discussion of sinusoids, complex frequencies and the Fourier transform.

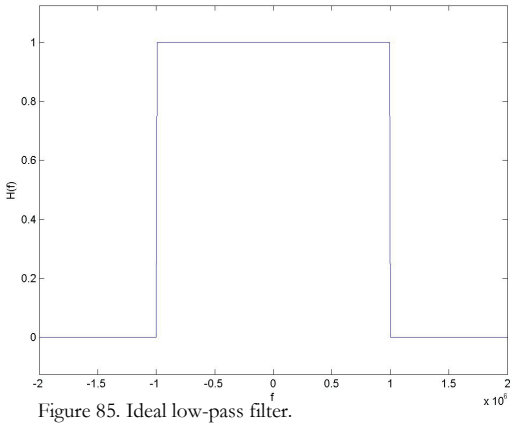


Figure 85. Ideal low-pass filter.

5.1.1 Frequency domain

For an ideal filter, we would like the transfer of all frequencies in the pass-band, to be the same, which means a flat (unity transfer) frequency response preferably without amplification. In the stop-band we would like to attenuate the signal as much as possible, ideally an infinite attenuation. For an ideal low-pass filter this means:

$$H(f)=1, \quad -f_0 < f < f_0 \quad \left(|H(f)|=1, \angle H(f)=0\right) \tag{5-1}$$

$$H(f)=0, \quad |f| > f_0 \quad \left(|H(f)|=0, \angle H(f)=0\right)$$

This is shown in Figure 85 with f_c, the corner frequency at 1MHz. Note that the angle is zero for all frequencies and the phase is thus unchanged for the output signal.

5.1.2 Time domain

We can show the response of this filter in the time domain by performing a Fourier transform:

$$h(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H(f)\cdot e^{j2\pi ft}\,df$$

$$h(t) = \frac{1}{2\pi}\int_{-f_0}^{f_0} e^{j2\pi ft}\,df \qquad \text{(5-2)}$$

$$h(t) = \frac{\sin(2f_0 t)}{\pi\cdot t} = 2f_0\,\mathrm{sinc}(2f_0 t)$$

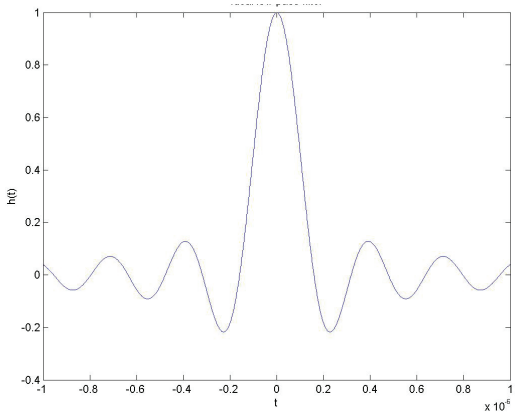


Figure 86. Ideal low-pass filter impulse response

This transient (impulse) response, shown in Figure 86, has the obvious problem of being non-causal, meaning it depends on a response in the past (i.e. negative time, before the signal actually occurs). Since this is impossible to achieve physically, an ideal filter can not be made and we must approximate it. Also shown is that the signal takes a long time to become steady, which means we must wait a long time before the signals after the filter settles to the ideally wanted value (0 or 1). This is called the "ringing" of the filter. This time delay will increase the sweep time of a spectrum scan.

The Paley-Wiener criterion relates causality in the time domain to the frequency response [47]:

- The magnitude response of the filter can not be zero for a finite frequency band, but can at a finite number of distinct frequencies.
- The magnitude response can not decrease faster than exponentially.

Network function are rational functions that approximate the exponential decrease. Note that zeros can be used to introduce sharp minima.

5.2 Filter design criteria and specifications

The ideal filter can be approximated. For an analogue passive filter we can use a capacitor or inductor as frequency dependent components. We can combine a number of frequency dependent components to achieve a better approximation. As capacitors and inductors are dual components they can provide a frequency dependent transfer function in different configurations. Each of them can give a frequency dependence in the numerator or the denominator of the transfer function (giving a zero or a pole respectively). A frequency dependence in the numerator gives a increasing transfer function for increasing frequency, while a frequency dependence in the denominator gives a decreasing transfer function for increasing frequency. Combined they give a flat transfer function. Note that zeros give a zero or infinitely attenuated frequency response, while poles give resonation or an infinite frequency response.

The order of a filter is the number of poles and zeros. At much higher frequencies than the pole and zero position, the frequency response is falling with $1/f^n$, with n the number of poles minus the number of zeros. Note that the response is increasing if there are more zeros than poles. The $1/f^n$ dependence is a straight line on a logarithmic scale. If we assume there are no zeros, the response falls with -10dB/decade/order for the power. This is -20dB/decade/order for the voltage transfer.

In this section we will introduce a number of performance criteria, which we can use to evaluate the performance of different filter designs of the same order, but with different transfer functions and thus a different approximation and different characteristics.

After these criteria we will discuss the applicability of the specifications from chapter 2 for filters in general and provide numbers for a typical filter.

5.2.1 Filter design criteria

In order to evaluate different filter designs with respect to each other, we identify a number of basic parameters characterising the performance of a filter, following from the analysis of an ideal filter.

The non-causality of the ideal filter is caused by the abrupt change from pass-band to stop-band. Causal filter have a transition range, which means the amplitude transfer begins to fall slowly close to the edges of the pass-band and faster and faster the further away from the pass-band. Thus the frequency response of the pass-band is not completely flat and the selectivity of frequencies in the pass-band and frequencies in the stop-band is limited. Also the phase response varies in the transition range.

From the time domain analysis of an ideal filter, we found that the transient response can take a long time, while time may be limited.

5.2.1.1 Amplitude accuracy (H(jω))

The transfer function of the filter specifies the frequency response, which is the amplitude versus frequency of the output signal divided by the input signal. The amplitude accuracy is very important for a spectrum analyser as this is primarily measured. Thus, the response of the pass-band must be as stable as possible, i.e. the frequency response must be flat, only starting to fall close to the corner frequency and as fast a possible past the corner frequency.

5.2.1.2 Selectivity (fall-off rate)

The selectivity of a filter is determined by its fall-off rate or roll-off after the corner frequency, i.e. how fast the frequencies above and/or below the pass-band are suppressed. Further away from the corner frequency the roll-off is always -20dB/decade per filter order which is -6dB/octave/order (without zeros). However, the difference is in the transition area from pass-band to -20dB/decade/order. The faster the frequency response falls with -20dB/decade/order after the corner frequency, the better the selectivity of the filter. With a sharper corner frequency a certain attenuation can be reached at a up to 5 times lower frequency.

As said we would like the frequency response to fall as fast as possible past the corner frequency, since this gives us a better selectivity.

5.2.1.3 Group delay ($\partial\varphi(\omega)/\partial\omega$)

A transfer function has an amplitude response as well as a phase response, with the amplitude being the magnitude of the complex transfer function and the phase response being the angle of the complex transfer function. The amplitude response provides the separating function of a filter. Sometimes what happens to the phase of the signal is not important at all. For example, the spectrum analyser normally shows the magnitude of the signals. However, for modulated signals or signals in the time domain used in vector signal analysers, the phase can be very important. Therefore, it can be important not to distort the phase of the signals. Ideally this would mean that the phase after the filter is exactly the same as before the filter. The phase of the transfer function of the filter is thus zero for all frequencies (at least in the pass-band). This is unfortunately not possible with a causal filter [40].

An alternative is when the phase changes the same amount (of time) for all signals with respect to its frequency. This is, the time delay is the same for all signals and it is just as if we measured a little later in time. With a fixed time and non-changing frequencies, the phase changes linearly with respect to frequency (i.e. a higher frequency component with the same time delay has a larger phase shift $\omega\Delta t$ than a lower frequency component). In other words, all spectral components experience the same delay. Thus the transfer function should have a linear phase response.

The group delay is the rate of change or the derivative of the phase with respect to frequency. If the phase is linearly dependent on the frequency, the group delay is a constant. Therefore, the phase relation between signals is not distorted.

Thus, we would like the angle of the transfer function of the filter to be zero or linearly dependent on frequency, which corresponds to a constant group delay.

5.2.1.4 Transient response (h(t))

The transient response shows the response in the time domain and indicates how long it takes for the output signal to stabilise and give the correct value. We already found that this response must be causal or being zero for negative time.

In order to evaluate the transient response, normally the impulse response or step response is used. We will use the impulse response with a unit impulse at time 0. The signal power of the impulse is thus 1, so we identify the time at which the impulse response falls under -74dB (the dynamic range) as the time after which the signal is valid.

5.2.2 Specifications

The specifications of chapter 2 are also relevant to filters. From the above filter criteria we found the frequency response influences the amplitude accuracy, while the transient response influences the sweep time as the filter must stabilise for each RBW.

For the other specifications, we find that phase noise only applies to the local oscillator. The specific frequency range and amplitude range are dependent on the filter, as well as the noise and linearity. The noise of each filter adds to the noise figure of the front-end just as any other block, while the filter must also be linear enough to not add any distortion products. A passive analogue filter is in principle a linear time-invariant system, but must be made by non-ideal components, which introduce non-linearities.

Finally, the frequency response of a filter determines the amplitude accuracy of the signal after the filter. The response in the pass-band can result in an amplifications or attenuation of the input signal (at different frequencies). As we found for the mixer, the filter can be used to adjust the gain of the input signal. As we have a dedicated LNA for this purpose, we take the gain of the filter in the pass-band to be 1 or 0dB.

State-of-the-art filter boundaries (2002)		
Filter order	3 ~6	6
Corner frequency	2 ~ 6 MHz	2.1 MHz
Gain	0 ~ 45 dB	18 dB
Amplitude accuracy	0.2 ~ 3 dB	0.2 dB
Stop-band rejection	45 ~ 100 dB	70 dB
Noise figure	30 ~50 dB	42 dB
IP_3	11 ~ 50 dBm	51.4 dBm
Vdd	0.3 ~ 3.3 V	2.7 V
Power	2 ~ 6 mW	1 mW

Table 4. Typical state-of-the-art filter upper boundaries.

Typical values for a state-of-the-art integrated filter gathered from [43] are shown in Table 4. Note that these filters have a low corner frequency compared to the input frequency range. This is because we will focus on RBW filters for reasons that will become clear in the next two sections. The last column shows a filter possibly suitable for the RBW and will be discussed further in section 5.4.

Looking at the specifications, the amplitude accuracy, stop-band rejection, linearity and voltage and power requirements are good enough for our use (chapter 2 and 3). The numbers from Table 4 are for active filters however, which explains the high noise figure. Gain can be applied to lower the noise figure, but the gain is about 5 to 20dB below the noise figure. We assigned a 2dB noise figure to the filters. For passive filters this is not a problem.

5.3 Low-pass pre-filter

The low-pass pre-filter is the second part in the general front-end. Its aim is to pre-filter high frequency signal outside of the input frequency range to prevent them to down-convert and distort a signal of interest. For a high-IF architecture this filter serves as a image rejection filter. In that case the filter must have good selectivity (high order) and the specifications of chapter 2. In our case of zero-IF, the image is not a problem. The filter now only filters out higher frequencies for the following front-end blocks. The filter is therefore not essential to the operation of the spectrum analyser. This means it must improve (condition) the input signal for the following parts, and not add any distortions.

The filter must pass the complete input frequency range and block everything above, thus we have a fixed -3dB corner frequency ω_c a little above 3GHz or 6GHz. The amplitude accuracy, of which the frequency response is part, is very important for a spectrum analyser and since this is an optional part, we would like the amplitude accuracy to vary ±0.5dB at most. At the corner frequency the response has fallen to -3dB and therefore the corner frequency must lie a little higher than the input range. The noise

figure must be as low as possible (less than 2dB), especially since this is an optional part. Finally, a high selectivity is nice, but since this is a pre-filter it is not absolutely necessary and must not come at a high price.

5.4 Resolution bandwidth filter

The resolution bandwidth filter is much more important and essential as the frequency selecting component for analogue spectrum analysers. It can also act as an anti-alias filter, which is discussed in section 7.3.2.

In the analogue case there are actually multiple resolution bandwidth filters, which differ in corner frequency f_c from 5kHz to 0.5MHz, but otherwise have similar requirements (for zero-IF, the RBW bandwidths centre around DC, making the corner frequency half the RBW bandwidth). We will use a corner frequency a little over 0.5MHz, assuming the largest frequency is the most difficult.

We would like an amplitude accuracy of ±1dB (section 3.2.4). The selectivity must be as good as possible, ideally completely rejecting the out-of-band signals at least 74dB (SFDR), which is of course not possible. Depending on the fall-off rate, a large signal a few RBWs away is not completely filtered out and already shown partly and more and more as it comes closer. Therefore the signal on screen will follow the filter shape. The group-delay in the pass-band preferably is flat so the phase can be accurately determined. As the spectrum analyser's main task is measuring the signal power, this is a secondary objective. However as a consequence of the transient response requirement, the group delay will be mostly flat (see section 5.5). For the transient response, the output signal of the RBW must have stabilised each time a new RBW is measured. The spectrum analyser's input signal changes each time the LO frequency is changed to select the next RBW (assuming a stepping LO with RBW sized steps). For a continuously sweeping LO frequency, the frequency sweep over one RBW must be so slow that a frequency peak that occupies the complete RBW is correctly measured. This means the sweep over one RBW must take longer than the filter needs to stabilise. The total sweep time is the stabilising time of one RBW measurement times the number of measurements.

As for the specifications; the RBW filter is still at the front of the design. Only after the RBW filter the linearity requirements are lowered (as there are no more out-of-band interferers, see chapter 3). According to Table 4 the linearity requirement should not be a problem for the filter (a 54dBm IP_3 and 17dBm IP_3 is needed). Being after the frequency conversion the input frequency range is limited to the intermediate frequency range, which is zero to halve the selected RBW for zero-IF. The complete amplitude input range is at the input of the filter and at the output representing a single frequency. The noise in this single frequency at the output of the filter thus has the bandwidth of the RBW filter. As found in section 2.3.7, this adds 60dB to the noise level for the largest RBW (or actually 57dB for the 0.5MHz for zero-IF). Therefore the largest RBW is the most critical noise wise with smaller bandwidths having a lower noise level. This characteristic is normally exploited when using a spectrum analyser and the noise level as well as the other specifications should preferably be the same for all the RBW filters. The noise figure that the filter may add was assigned to be only 2dB in chapter 3.

5.5 Filter designs and families

First-order passive filter designs consist of a capacitor or inductor together with resistive losses which is a pole in the transfer function and thus has a fall-off rate of -20dB/decade.

Second order implementations consist of a capacitor and an inductor (as two capacitors or inductors add up to act as one and thus a first-order filter). This filter therefore has a fall-off rate of -40dB/decade when the two poles combine their fall-off.

Higher order filters designs alternate capacitors and inductors to achieve a higher fall-off. By varying the component values and thus the location of the poles, different filter characteristics can be achieved.

Since the ideal filter from section 5.1 is not possible, there are a number of polynomial approximations resulting in higher order filter designs [42], each optimised for different characteristics. The polynomial approximations determine the component values. These designs can be classified into different families based on major characteristics. We will evaluate each approximation for a low-pass filter according to our requirement criteria. From this low-pass filter one can easily create different types of filters with the same characteristics by performing filter transformations.

For each filter design a third order filter with a corner frequency of 1MHz is simulated. The magnitude transfer function and envelope delay are generated with the student edition of the Elsie electronic filter design and analysis program [44]. The impulse response is generated with Maple [45].

5.5.1 Chebyshev family

The Chebyshev family [42] of filters are all optimised primarily on their fall-off rate. As we will see this has consequences for the other performance parameters. These filters can be developed along similar lines and can revert to one another under certain circumstances. We will start with the filter design with the fastest fall-off and gradually work towards more compromising designs.

5.5.1.1 Cauer

The Cauer filter or also called elliptical filter is optimised for a maximal fall-off with a defined pass-band ripple and stop-band attenuation depth (stop-band ripple). The more distortions in the frequency response and group delay are allowed the steeper the fall-off is. The first pole is responsible for the corner frequency fall-off and is quickly followed by a LC resonator in the forward path, which block the frequencies close to the resonator frequency and is responsible for the sharp dip in the frequency response. The next pole ensures the frequency response coming back up from the dip has a maximum which is the stop-band attenuation depth. The closer the stop-band (i.e. a larger fall-off and selectivity) or the deeper the attenuation, the larger the ripple is in the pass-band. Thus there is a limit to the steepness and attenuation set by the pass-band ripple allowed (which results in an input amplitude inaccuracy).

The high selectivity comes at the cost of a distorted group delay nearing the corner frequency and a long time before the ringing fades out especially before the impulse response becomes below -74dB from the peak value.

Filter	Amplitude accuracy	Selectivity	Group delay	Transient response
Cauer	--	++	--	--

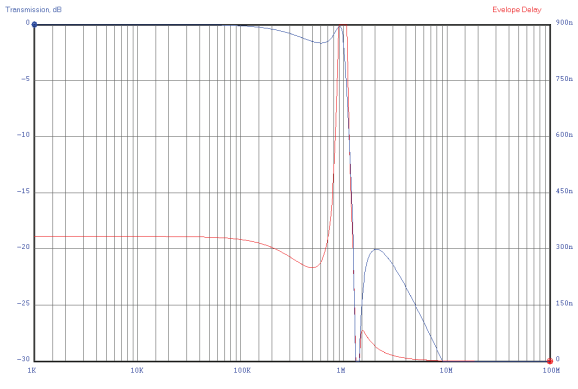


Figure 87. Cauer filter frequency response and group delay.

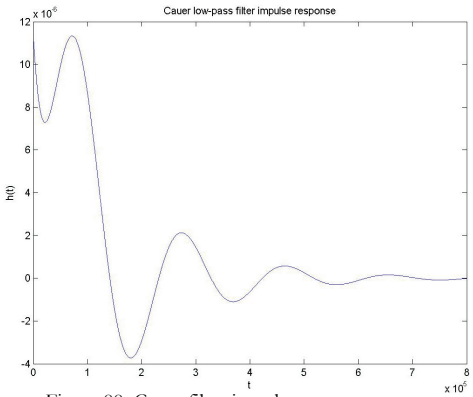


Figure 88. Cauer filter impulse response.

5.5.1.2 Chebyshev

The Chebyshev filter is optimised for a maximal fall-off with a defined pass-band ripple (type I) or a defined stop-band attenuation depth (stop-band ripple) (type II or inverse Chebyshev). The fall-off is less than that of a Cauer filter, but it has also less group delay variations in the vicinity of the corner frequency and less ringing. The inverse Chebyshev filter has a slower fall-off and requires some extra components than a type I filter, but the advantage of no pass-band ripple.

Filter	Amplitude accuracy	Selectivity	Group delay	Transient response
Chebyshev	-	+	-	+-
Inverse Chebyshev	+-	+	+-	-

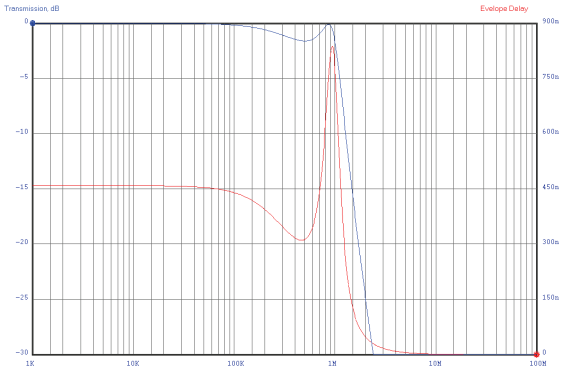


Figure 89. Chebyshev filter frequency response and group delay.

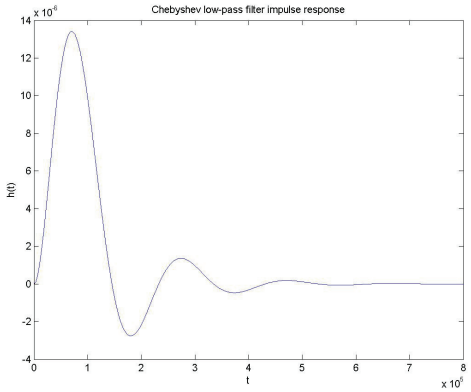


Figure 90. Chebyshev filter transient response.

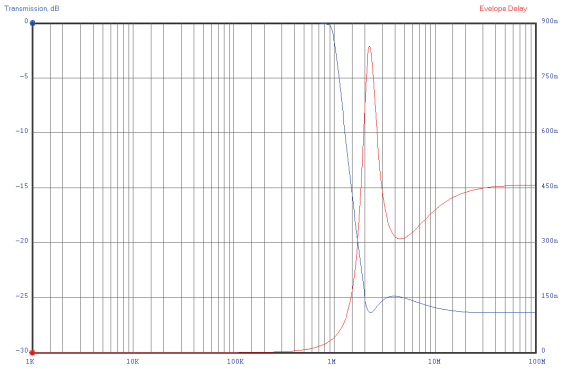


Figure 91. Inverse chebyshev filter response and group delay.

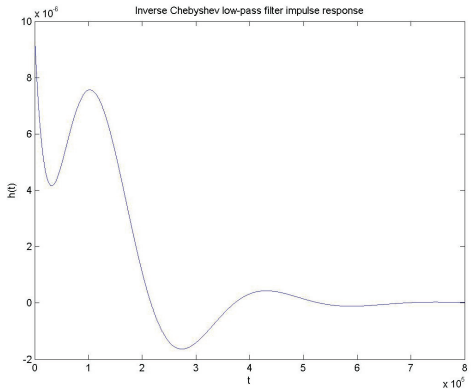


Figure 92. Inverse Chebyshev transient response.

5.5.1.3 Butterworth

The Butterworth or maximally flat magnitude filter has a maximal fall-off with a flat or monotonic pass-band. The Chebyshev filter becomes a Butterworth filter when no ripple is allowed. The flat pass-band gives a better amplitude accuracy, but at the cost of a lower selectivity. However, the group delay and transient response are also better. It becomes intuitive that a distortion of the pass and/or stop-band allows for a better selectivity but also takes longer to stabilise. The Butterworth having no pass- and stop-band ripples, the Cauer having both and the Chebyshev having one of them.

Filter	Amplitude accuracy	Selectivity	Group delay	Transient response
Butterworth	++	+-	+	+

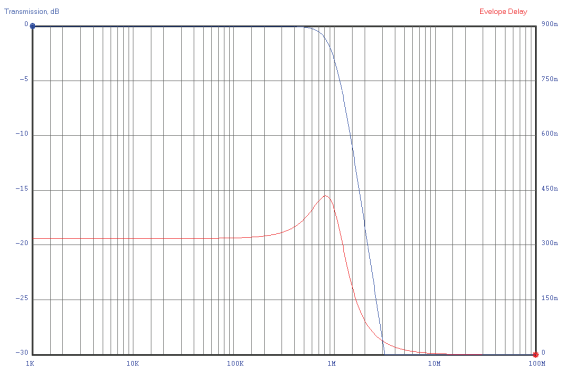


Figure 93. Butterworth filter frequency response and group delay.

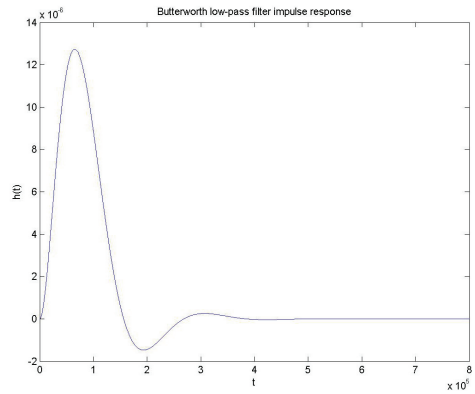


Figure 94. Butterworth filter transient response.

5.5.2 Gaussian family

From the Butterworth transient response one can still see a relatively large ringing effect, which can take a long time after the first peak to dampen out enough. One can image the need for a fast response without ringing (or overshoot for a step response). The filters that are optimised for transient or group-delay characteristics all approximate a Gaussian pulse and are therefore in the Gaussian family. Another advantage of this approximation is symmetry around the first impulse response peak, instead of a fast ramp up and slower ramp-down with ringing of the Chebyshev family filters.

5.5.2.1 Bessel

The Bessel filter is optimised for a maximal flat group delay. The filter therefore has a maximally linear phase response. This improved group delay comes at the price of a lesser selectivity. An expected result as we could already see the group delay flatten out for the Butterworth filter compared to more selective filters. The frequency response as well as the transient response are more or less approximating a Gaussian shape.

Filter	Amplitude accuracy	Selectivity	Group delay	Transient response
Bessel	++	-	++	+

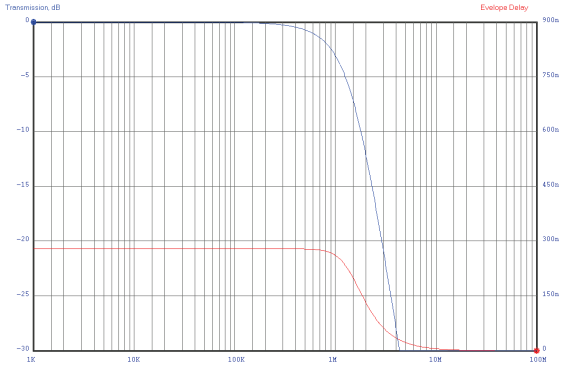


Figure 95. Bessel filter frequency response and group delay.

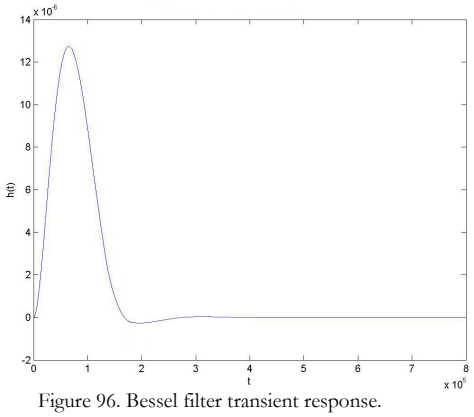


Figure 96. Bessel filter transient response.

5.5.2.2 Gaussian magnitude

The Gaussian magnitude filter has a maximal fall-off without transient response ringing. The magnitude and transient response both approximate the Gaussian shape as much as possible for the filter order. The result is a flat group delay and no ringing while having even lesser selectivity. An even less steep fall-off than Gaussian give a faster response, but since selectivity is the filters main function, the Gaussian filter is seen as the best compromise between frequency and time domain behaviour.

Filter	Amplitude accuracy	Selectivity	Group delay	Transient response
Gaussian	++	--	++	++

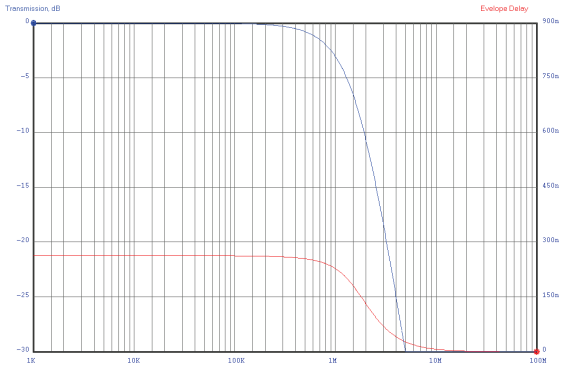


Figure 97. Gaussian magnitude filter response and group delay.

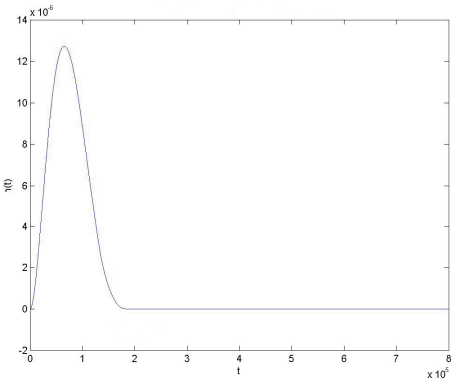


Figure 98. Gaussian magnitude transient response.

5.5.3 Transitional and other filters

There are a number of filter designs in the region between the good selectivity of the Chebyshev family and the good group delay and transient response behaviour of the Gaussian family. These will be discussed briefly.

5.5.3.1 Legendre

From the Chebyshev filter we can see that allowing even a very small ripple in the pass-band has a large effect on the selectivity. If we allow no ripple in the pass-band at all, the Chebyshev filter degenerates to a Butterworth filter. The Legendre filter uses a different approximation than the Chebyshev family to get a filter with a pass-band frequency response which is not as flat as the Butterworth filter, but still has no ripple. This gives the Legendre filter a selectivity, group delay and transient response in between the Butterworth and Chebyshev filter.

5.5.3.2 Linear phase with a defined ripple

Just as we found that allowing a defined ripple in the frequency magnitude response of a Butterworth filter in exchange for a better selectivity of the Chebyshev filter, we can allow a ripple in the flat group delay response of the Bessel filter to achieve better selectivity. This results in a linear phase filter with a defined ripple. As said its selectivity is a little better as the Bessel filter, but with a ripple in the group delay and more ringing in the transient response.

5.5.3.3 Gaussian to 6/12dB

An ideal Gaussian magnitude filter has a series approximation of infinite terms (see section 5.6.3.1). A finite number of terms approximate the ideal filter up to a certain level down from the peak and after this level has a normal (-20dB/decade/order) filter fall-off. Two common approximations are Gaussian down to 6dB and down to 12dB. The fall-off rate of the ideal Gaussian filter steadily increases (Figure 99), while the approximation's levels off to a fixed value. The approximation is therefore a little less selective and has a little faster transient response than an ideal filter.

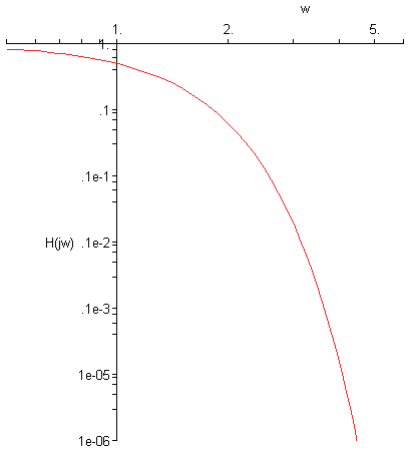


Figure 99. Ideal Gaussian magnitude filter fall-off rate on a double logarithmic scale (Bode plot).

5.5.3.4 Synchronously tuned

The synchronously tuned filter (also see section 5.6.3.2) is a simple cascade of single poles. All poles are tuned to the same frequency, but the exact placement is not critical. Each stage has a lower Q (quality factor, also see section 2.3.7.3) and therefore selectivity than the over-all filter, which combine to give the larger Q and selectivity. As it is simply a cascade of first-order filters with the same corner frequency, the selectivity is worse than the other higher-order filters, which place their poles in order to optimise certain filter characteristics. However, because of the Central Limit theorem[9], the frequency response of the synchronously tuned filter will approximate that of the Gaussian filter. The simplicity of the stages, combined with the non-critical pole placement and therefore component values, while still approximating a Gaussian response, give this filter its appeal, even though the selectivity is less than that of the Gaussian filter.

[9] The Central Limit theorem states that the sum of equally distributed independent stochastic values (each synchronously tuned filter stage) with finite variance approximates a normal (also called Gaussian) distribution (the complete synchronously tuned filter response) [46]. Also see section 5.6.1.1.

5.5.4 Selection of the RBW filter type

Most of the above filter characteristics are grouped together in Table 5, Figure 100 and Figure 101.

Filter	Amplitude accuracy	Selectivity	Group delay	Transient response
Cauer	--	++	--	--
Chebyshev	-	++	-	+-
Inverse Chebyshev	+-	+	+-	-
Legendre	+	+	+	+-
Butterworth	++	+-	+	+
Bessel	++	-	++	+
Gaussian magnitude	++	--	++	++
Synchronously tuned	++	---	++	++

Table 5. Filter criteria comparison of common higher-order filters.

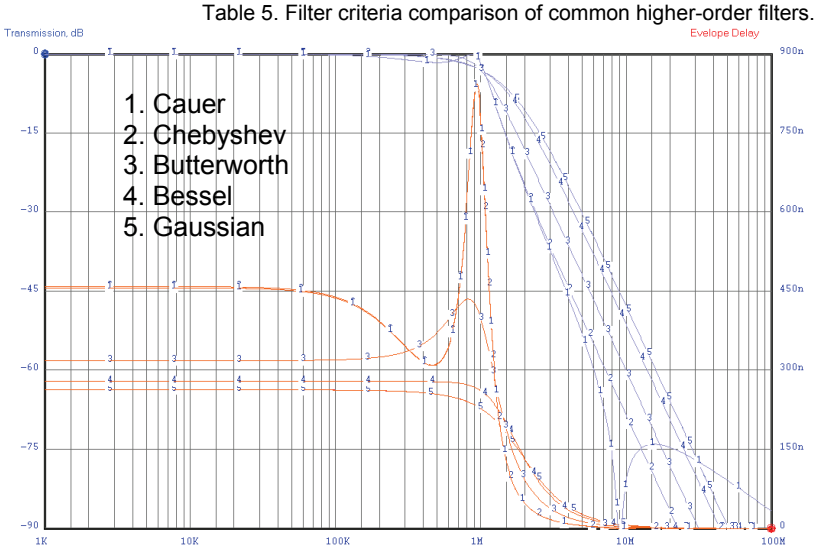


Figure 100. Frequency response and group delay of common higher-order filters.

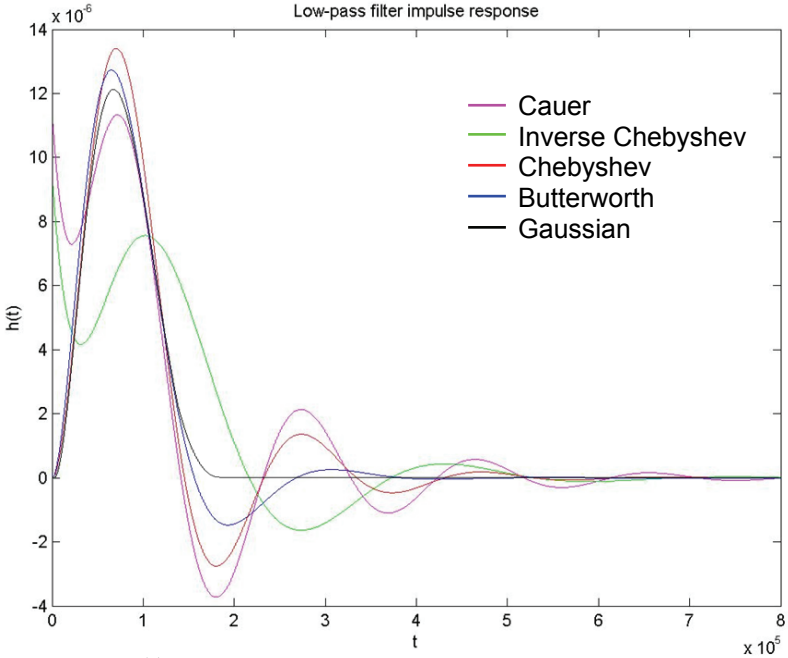


Figure 101. Transient response of common higher-order filters.

The frequency response and group delay in Figure 100 are numbered 1 to 5 and correspond to the Cauer, Chebyshev, Butterworth, Bessel and Gaussian magnitude filter respectively. The Legendre filter is between the Butterworth and Inverse Chebyshev responses. The Gaussian to 6dB and 12dB are a little more selective than the Gaussian response and the synchronously tuned a little less.

The transient response is shown for the Cauer, Inverse Chebyshev, Chebyshev, Butterworth and Gaussian filters in Figure 101. The Legendre, Gaussian to 6dB and Gaussian to 12dB have response close to the Gaussian's but with a little bit of ringing which would not be clearly visible in the figure and is therefore left out. The synchronously tuned filter has a little faster response than the Gaussian.

From the filter characteristics, we can observe a few trends.

First of all there is a trade-off between the fall-off rate and the group delay and transient response. A filter with a higher selectivity approximates the ideal filter better, but this also results in time behaviour more like a sync pulse with a long stabilisation time (ringing). This is the selectivity versus speed trade-off. This is a consequence of the inverse relationship between time and frequency and is also called the uncertainty principle [2].

For a spectrum analyser the sweep time and thus the time to wait for a stabilised frequency response is limited. However if the selectivity is not good enough, we must select a smaller RBW to get a good enough resolution to be able to separate two signals

close to each other. With a smaller RBW and the same frequency range, the measurement time is also longer. Thus we have to wait because of more measurements because of a smaller RBW or because of a slower filter with higher selectivity.

The second observation is that the first response peak is always there. Thus there is always a time delay because of the first pulse inherent of the filtering, which was also found during the discussion of the group delay in section 5.2.1.3. The peak becomes narrower and closer to the origin when the fall-off is less steep, again the selectivity-speed trade-off. This will be shown more clearly during the discussion of the synchronously tuned filter in section 5.6.3.2.

Thus we must select a balance between selectivity and speed, which translates to resolution and sweep time for a spectrum analyser. But the exact balance is not important as a higher selectivity takes more time per measurement while a lower selectivity needs more measurements to achieve the same resolution. The optimum would thus be when the selectivity and speed combined give the highest resolution for a certain sweep time. From the Cauer to the Gaussian filter frequency response, we have about a four times improvement with the -74dB point at 8MHz for the Cauer filter and at 40MHz for the Gaussian filter. For the transient response, the Cauer filter still has ringing at least 4 times longer than the Gaussian filter, while we can see the first pulse is not getting much smaller for filters with lower selectivity. So the speed times selectivity is basically equal for the two. However, the amplitude accuracy and group delay of the Gaussian filter are much better. Thus filters of the Gaussian family are preferred. Filters with less selectivity than the Gaussian filter improve the transient response much slower because of the first pulse. This is no coincidence as will follow from the more detailed treatment of a Gaussian shape in the next section.

Because of its well balanced time and frequency domain behaviour, a filter of the Gaussian family is preferred. However, the Bessel filter which still has some ringing, takes longer to stabilise to -74dB than the Gaussian magnitude filter without ringing, while the selectivity improves marginally. Thus an approximation with less selectivity and no ringing is preferable over an approximation with more selectivity and a little ringing.

Note that higher selectivity means worse amplitude accuracy, slower transient response (more ringing) and more phase distortion and that the group delay (phase distortion) flattens out before the ringing of the transient response stops with increasingly lower selectivity.

5.6 Gaussian shape

We determined that the frequency response of the Gaussian magnitude filter is the preferred shape for the resolution bandwidth filter because of its well balanced selectivity and speed. Therefore we will further analyse the Gaussian shape and identify why the Gaussian filter has this balance and how we can approximate the shape.

5.6.1 Analysis

The Gaussian or also called normal distribution [46] is a symmetrical probability distribution around an average value μ (the mean). This equals the likelihood of the occurrence of a random value around the average value μ. The further away from the average value the less likely it is to find an occurrence (i.e. the probability is less). This is a frequently occurring situation in biology, engineering and economics. The variance σ^2 is a measure for the distribution around the average value, with σ the standard deviation. The standard deviation is a scaling factor; for a larger σ the probability of a large deviation with respect to μ is higher. The normal distribution is written in shorthand as $N(\mu, \sigma^2)$. The standard normal distribution is N(0, 1) and has a bell shape. It is therefore also called the bell curve. At one standard deviation, the response is about 60% of the peak value and the area under the curve up to one standard deviation is about 68% of the total area. This is also the bending point of the graph.

The Gaussian distribution is provided mathematically with x the stochastic variable around μ with standard deviation σ and $f(x)$ the probability of its occurrence as:

(5-3) $$f(x)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}} \text{ or } f(x)=e^{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}-\log\sigma-\frac{1}{2}\log 2-\frac{1}{2}\log\pi}=e^{ax^2+bx+c}$$

For a Gaussian filter, the frequency response has a Gaussian distribution shape. As said, we would like the peak response of the filter to be one[10] and therefore the Gaussian distribution is scaled by a factor $\sqrt{2\pi\sigma^2}$. We would like the amplitude accuracy to be ±1.0dB in the pass-band, so the frequency response must be at least that flat. For the largest RBW, the response at 0.5MHz is then -1dB. Since the fall-off rate is fixed by the Gaussian distribution shape (i.e. a higher order filter improves the approximation, but the maximum fall-off is limited by the order, while the ideal response has an ever increasing fall-off rate), we can calculate the -3dB point to be at about 0.87MHz and σ at 0.74MHz. This is shown in Figure 102.

Figure 102 does not look much like a normal frequency response, because the frequency response is normally displayed on a double logarithmic scale. Figure 99 already showed the frequency response on a double logarithmic scale in section 5.5.2.2 and is repeated in Figure 103.

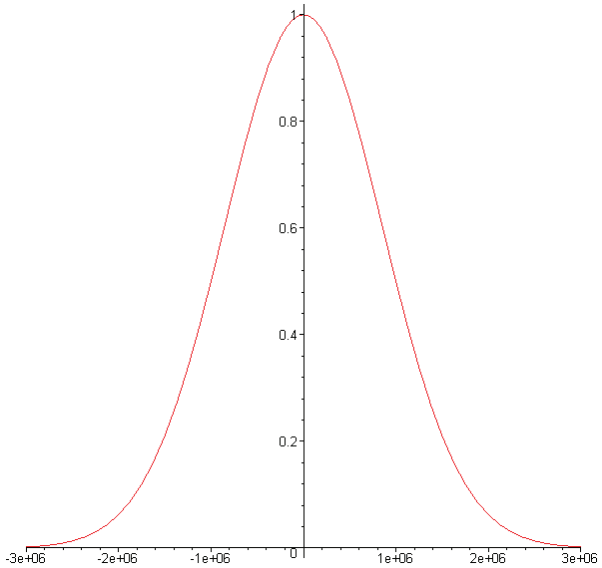


Figure 102. Gaussian distribution around 0 with a $0.74\cdot10^6$ standard deviation and scaled to a peak response of 1.

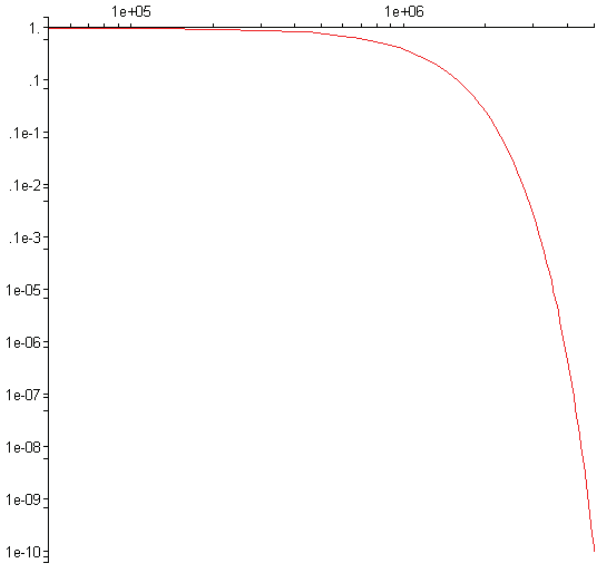


Figure 103. $N(0, (0.74\cdot10^6)^2)$ distribution on a double logarithmic scale (bode plot).

5.6.1.1 Central limit theorem

The common occurrence of the Gaussian distribution shape is because the sum of random values all with equal likelihood converges to the normal distribution. This can be demonstrated for example with dice. If we take one dice and throw it a large number of times, each number on the dice is thrown approximately the same number of times, i.e. it is uniformly distributed. If we take two dice and sum the result, it is obvious that the total will equal 6 more than 1 or 12. If we take more dice, the result will resemble the bell shape more and more. This is called the central limit theorem.

The Central Limit theorem states that the sum of equally distributed independent stochastic values with finite variance approximates a normal (also called Gaussian) distribution [46].

[10] Actually, since we know the response will not be higher than the peak response, it is advantageous to set the peak the maximum allowable amplitude error above one. The response can then fall to the maximum allowable amplitude error below one at the wanted frequency. This therefore improves the selectivity of the filter and evens-out the average value in the pass-band better.

5.6.2 Fourier transform

If we have a Gaussian shaped frequency response, we can derive the transient response by performing an inverse Fourier transform [2]. This derivation is provided in appendix C.

The result is that the Fourier transform of a Gaussian frequency response is a Gaussian transient response. Because a Gaussian does not exhibit ringing, it provides the well-balanced trade-off between frequency selectivity and stabilisation time we found in the previous section. It has this behaviour because the shape of the response is the same in the frequency and time domain.

The Gaussian and its Fourier transform also show the duality of a signal and its Fourier transform, because the standard deviation in the time domain is the inverse of the standard deviation in the frequency domain (or the other way around). This corresponds to the scaling theorem, which states that the shorter the pulse, the broader the spectrum and the longer the pulse, the narrower the spectrum [2].

5.6.3 Approximation

The transient response of a Gaussian filter is centred around *t=0*. This time origin can of course be shifted to move it to some positive value. In order for the filter to be causal, however, the infinitely expanding side lobes of the Gaussian shape must be truncated up to *t=0*. How far the peak of the response is moved away from the time origin determines how well the ideal Gaussian shape is approximated, but also results in a longer delay. Also one can not wait infinite time for the response to stabilise, so the positive side lobe is also truncated.

The result is the Gaussian shape is approximated. We will discuss two approximations of the Gaussian response; the Gaussian magnitude implements the series expansion of the Gaussian pulse, and the synchronously tuned approximates the Gaussian shape because of the Central Limit theorem.

5.6.3.1 Gaussian magnitude

The frequency response is also approximated. An exponential can be represented as the following convergent infinite series found from the Taylor series expansion:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots \tag{5-4}$$

Which for a Gaussian magnitude filter corresponds to:

$$H(j\omega) = e^{-\frac{\omega^2}{2\sigma^2}} = \sum_{n=0}^{\infty} \frac{1}{n!}\left(-\frac{\omega^2}{2\sigma^2}\right)^n = \sum_{n=0}^{\infty} \frac{1}{n!}\left(-\frac{1}{4}\right)^n \left(\frac{\omega}{\sigma}\right)^{2n}$$

$$H(j\omega) = 1 - \frac{1}{4}\left(\frac{\omega}{\sigma}\right)^2 + \frac{1}{32}\left(\frac{\omega}{\sigma}\right)^4 - \frac{1}{384}\left(\frac{\omega}{\sigma}\right)^4 + \ldots \tag{5-5}$$

The Gaussian magnitude filter implements this approximation up to a certain level as it can only implement a finite number of terms. An *n*-order filter can implement (*n+1*) terms of the equation [42].

The placement of the filter poles is critical for a good approximation of the Gaussian shape. High quality components must be used as small errors in their value introduce errors in the approximation. As the Gaussian filter is the fastest response before introducing ringing, these errors can result in not completely linear phase and ringing in the time domain.

5.6.3.2 Synchronously tuned

As said the synchronously tuned filter approximates the Gaussian filter's response because of the Central Limit theorem. The selectivity of the synchronously tuned filter is not as good as the normal Gaussian magnitude approximation. However, the filter stages are much simpler, because they are simple first order RC or RL sections. The transfer function of a simple RC low pass filter (Figure 104) section is:

(5-6)
$$H(j\omega) = \frac{1/j\omega C}{R + 1/j\omega C} = \frac{1}{1 + j\omega RC}$$
$$H(s) = \frac{a}{(s+a)}, \; with \; a = \frac{1}{RC}$$

Thus for a synchronously tuned filter:

(5-7)
$$H_{st}(s) = \frac{a^n}{(s+a)^n}$$, with n the filter order.

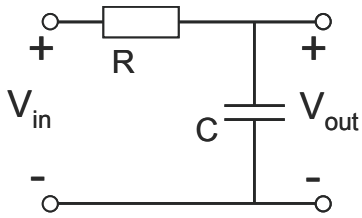


Figure 104. First order RC low-pass filter.

While the synchronously tuned filter has a lower selectivity than a Gaussian magnitude filter of the same order, each section is identical for the synchronously tuned filter and more importantly the exact pole location and thus the accuracy of the component values is not critical. It is therefore very easy to increase the filter order to improve the result.

5.7 Summary and Conclusion

In this chapter frequency selection by filters is discussed. The main focus is on the resolution bandwidth filter. Using four performance criteria, a filter design is selected and further analysed.

5.7.1 Summary

After analysing an ideal filter, it is shown that this filter is not realisable because it is not causal and that it exhibits a lot of ringing in the time domain, causing the filter to take a long time to stabilise. The ideal filter must therefore be approximated. We identified four design criteria to evaluate filter performance against; the amplitude accuracy or frequency response, the selectivity or fall-off rate, the group delay or linear phase response and the transient or impulse response. These factors can be optimised at the cost of one or more of the others during the filter design. General specifications of a filter include the linearity distortions it introduces and the noise it adds to the signal. Typical values for these design criteria and specifications set out against the specification of the spectrum analyser front-end show that amplitude accuracy and linearity should provide no problems. The noise of active filter designs is too high, thus passive solutions must be used.

More detailed discussion of the low-pass pre-filter show that the pre-filter is not allowed to introduce any distortions or delays, because it is not an essential front-end part. Therefore the group delay must be flat and the transient response fast. An amplitude accuracy of ±0.5dB and a maximum noise figure of 2dB are allowed. Because of these requirements the selectivity of the filter is of second concern.

The resolution bandwidth filter is essential and we would like a good selectivity, preferably blocking the next RBW -74dB (the dynamic range), as well as a fast response so the sweep time is short. The amplitude accuracy needs to be ±1dB. A flat group delay is important for accurately determining the phase, however this is a secondary objective. As the RBW represents one frequency that will be measured, the noise after the filter is the noise of the signal. Because noise is frequency dependent a wider bandwidth adds more noise. This could allow for a higher noise figure for filters with smaller bandwidths. However, since this property is normally also exploited to lower the noise level when using a spectrum analyser (i.e. a higher resolution is chosen, taking a longer time but giving more information and less noise), the noise figure of all RBWs must be the same 2dB allocated for the filter. The IP_2 and IP_3 must be higher than 54dBm and 17dBm respectively. According to the typical filter values the linearity is not a problem, but the noise of active filters is and passive solutions must be used.

The next step is to select a filter design, primarily for the resolution bandwidth filter. Several common higher order filter designs are evaluated according the filter criteria we defined. This includes the Cauer, Chebyshev, inverse Chebyshev, Butterworth, Bessel, Gaussian magnitude, Legendre, linear phase with a defined ripple, Gaussion up to 6 or 12dB and synchronously tuned filters. It is found that the impulse response quickly

becomes a number of times larger than the first main response if a filter is more selective than the Gaussian filter is used. Also the group delay is distorted. For example, for the Gaussian filter, the -74dB point is reached at 40MHz and for the Butterworth filter at 23MHz, for a 1MHz corner-frequency. But the impulse response takes about 2.5 times longer to settle. However, for filters less selective than the Gaussian filter, the impulse response improves marginally. Thus, during the evaluation, the selectivity versus speed trade-off is identified being a fundamental property of the time and frequency domain relationship (the scaling theorem and uncertainty principle of Fourier transforms).

It is found that a filter with a (nearly) Gaussian frequency response provides the best trade-off, because of its well balanced time and frequency domain behaviour. An approximation with less selectivity and no ringing is preferable over an approximation with more selectivity and a little ringing. Note that higher selectivity means worse amplitude accuracy, slower transient response (more ringing) and more phase distortion and that the group delay (phase distortion) flattens out before the ringing of the transient response stops with increasingly lower selectivity.

The Gaussian shape is therefore further analysed and it is shown that the Fourier transform of a Gaussian is also a Gaussian. This explains the well-balanced frequency and time behaviour. An ideal Gaussian shape can not be implemented. Therefore the series expansion is shown which the Gaussian magnitude filter implements with a finite number of terms.

The synchronously tuned filter also approximates a Gaussian response because of the Central limit theorem. The selectivity is less and the approximation worse than the Gaussian magnitude approximation for the same filter orders. However, its design is much simpler and the component values less critical.

5.7.2 Conclusion

The analogue spectrum analyser front-end contains filters at two positions. The low-pass pre-filter and the RBW filter. Analogue filter designs are evaluated on four criteria; the amplitude accuracy or frequency response, the selectivity or fall-off rate, the group delay or linear phase response and the transient or impulse response.

The low-pass filter may not introduce any distortions or delays. An amplitude accuracy of ±0.5dB and a maximum noise figure of 2dB are allowed. For a zero-IF architecture, it is optional.

The resolution bandwidth filter is essential and we would like a good selectivity, preferably blocking the next RBW -74dB (SFDR), as well as a fast response so the sweep time is short. The amplitude accuracy needs to be ±1dB. According to the typical filter values the linearity is not a problem, but the noise of active filters is and passive solutions must be used.

An ideal filter (with respect to selectivity) is not realisable because it is not causal. It also exhibits a lot of ringing in the time domain. Filters designs approximate the ideal response according to different criteria. During the evaluation, the selectivity versus speed trade-off is identified being a fundamental property of the time and frequency domain relationship (the scaling theorem or uncertainty principle of Fourier transforms). It is found that a filter with a (nearly) Gaussian frequency response provides the best trade-off, because of its well balanced time and frequency domain behaviour. An approximation with less selectivity and no ringing is preferable over an approximation with more selectivity and a little ringing.

The Fourier transform of a Gaussian is also a Gaussian, which explains the well-balanced frequency and time behaviour. An ideal Gaussian shape can not be implemented. The Gaussian magnitude filter implements the series expansion with a finite number of terms. The synchronously tuned filter also approximates a Gaussian response because of the Central limit theorem. The selectivity is less and the approximation worse than the Gaussian magnitude approximation for the same filter orders. However, its design is much simpler and the component values less critical.

Chapter
6

6 Signal measurement

In this chapter we will further discuss the measurement block. In chapter 3 we found that the amplitude accuracy of the measurement must be less than ±0.5dB and the noise figure of the measurement block must be less than 4dB. The out-of-band interferers are filtered by the RBW filter. As the in-band signals are conceptually a ”single” frequency we would like to measure, the linearity requirements are relaxed to an IP_2 of 0dBm and an IP_3 of -10dBm.

The measurement block must determine the signal power of the currently selected frequency band, so that the spectrum analyser can generate the power spectral density, which approximates the power spectrum as explained in section 2.1.5. However, practically the spectrum analyser measures the signal’s amplitude. With this amplitude and a known impedance, the signal power can be determined.

A spectrum is the quantification of something versus frequency (section 2.1). This quantity is normally the signal power for a spectrum analyser, but this is not necessarily so. For example, the Fourier transform determines the amplitude spectrum, which could also be an option on the spectrum analyser.

The analogue spectrum analyser measures the signal amplitude and determines the spectrum. The signal can be digitised at various stages of the front-end. We will still call the architecture an analogue spectrum analyser, if the signal is merely digitised and processed digitally in the same way as a traditional analogue spectrum analyser. However, when this digitisation causes changes in the capabilities or the operating principle of the design, we will call it a digital spectrum analyser.

For an analogue spectrum analyser, there are a number of options to measure the signal, after the frequency selection. These are discussed in section 6.1. For a digital spectrum analyser, the analogue to digital conversion (ADC) can be seen as the measurement of the signal. The ADC requirements are discussed in section 6.2. After digitalisation a number of options exists, of which the implications are discussed in chapter 7.

In both the analogue and digital case, the signal after measurement should not be susceptible to noise, interference or other introduced errors anymore, when handled with care (because of the reduced dynamic range in the analogue case and the digitalisation in the digital case).

6.1 Analogue

In case of an analogue spectrum analyser, the Fourier transform is performed in the analogue domain. The RBW represents one frequency component and we would like to measure the average or peak amplitude. Other types of measurement (sample, quasi-peak) might also be of interest and are further explained in [6] (p.26).

Unfortunately it is possible that the resolution bandwidth is too large and there are actually two or more[11] frequency components in the RBW. If the amplitudes are in the same range, the measured amplitude is slowly time-varying instead of static. This is a limitation of all swept-tuned analogue spectrum analysers. Because the input signal is swept it can not constantly monitor a changing amplitude. In order to separate the frequencies a smaller RBW must be used. With multiple measurements (sweeps) of the same frequency, the average can be taken of this slowly changing amplitude.

[11] As the bandwidth can be chosen infinitely small, an infinite number of frequency components are possible and actually exist, because noise contains all frequencies. However, signals below a certain level don’t noticeably influence the measured amplitude.

6.1.1 Linear and logarithmic amplifier

The input amplitude range of the signal is quite large. We can use a linear amplifier to select a smaller amplitude range and scale it to the display range. Some signals can be small, while others are large, which if displayed on a linear scale would leave most of the smaller signals invisible. Thus we would like to have the option to compress this large range by displaying the measurements on a logarithmic scale.

The amplitude accuracy of 0.5dB, determined in section 3.2.5, must be divided over the linear or log amplifier and the amplitude detection. Therefore, we allow 0.25dB for each. Of the 4dB noise figure left, we allow 2dB for the amplifier and 2dB for the rest. If we put the amplifiers before the signal measurement, the amplifiers must achieve a 2dB NF and the 0dBm IP_2 and -10dBm IP_3 determined in section 3.2.5.

The linear amplifier scales a smaller input range for the following block, the signal measurement. The smaller amplitude range is shown linearly, for example a 20dB range with the largest signal a 100 times larger than the smallest. This means a small signal is amplified to a higher range the amplitude measurement supports. In case a large signal was already present, this can also cause an overflow. We assume signals do not have to be attenuated. In that case, the signal as well as the noise is amplified and the signal-to-noise ratio is not changed, except for the noise figure the amplifier itself adds. The following blocks therefore have 2dB NF left.

The amplification of the signal to a logarithmic scale is done with a logarithmic amplifier (log amp). With an 70dB input range (-90dBm to -20dBm) and 0.5dB of accuracy (display resolution), we have 140 levels, with -90dBm corresponding to 1/140 or -42.9dB of the maximum signal. We would like the noise level to remain below -94dBm. The highest DANL is -96dBm at a 1MHz RBW (there is 2dB NF left for the measurement). After the log amplifier this corresponds to 1/152 or -43.6dB. With -94dBm corresponding to 1/148 or -43.4dB, the NF of the following blocks must be less than 0.2dB because of the log amp. The added noise of the following blocks must therefore remain -57.9dB under the maximum signal after the log amp. For smaller RBWs the DANL is lower and thus the added noise can be larger (for example -50.8dB under the maximum signal for a 100kHz RBW). We can scale this maximum signal to make sure this is easily achievable.

Note that the log amp rectifies the signal as the logarithm of a negative number does not exist. We can track the sign or use an offset, but for the amplitude measurement this is not necessary.

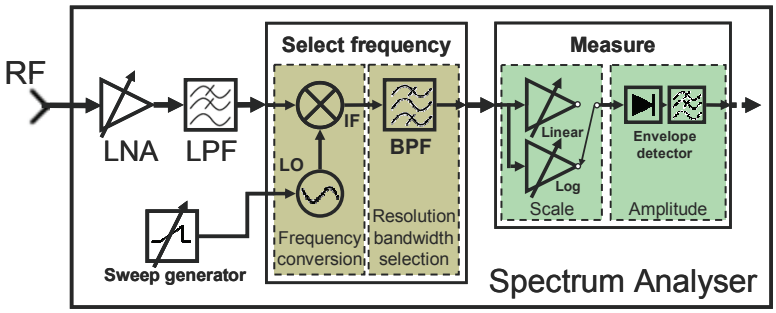


Figure 105. Analogue spectrum analyser front-end

Noise and distortions should not be visible anymore after the linear or log amp as they should be much smaller than the amplified or compressed signal range that is displayed.

If the amplifiers are put after the signal measurement, the amplification is done for display purposes and is therefore done just before displaying. The signal processing after the measurement can than use a linear non-amplified (uncompressed) signal. But the requirements are not relaxed in that case and the measurement and following blocks together must achieve a ±0.5dB amplitude accuracy and 4dB NF, besides the 0dBm IP_2 and -10dBm IP_3 determined in section 3.2.5.

6.1.2 Envelope detector

In order to measure the amplitude we can rectify the signal (with a diode) followed by a low-pass filter. This is called an envelope detector and is shown in Figure 105. The resulting value is the average (RMS) amplitude of the signal. The amplitude accuracy must be 0.25dB (see above).

6.1.3 ADC

As an alternative for the signal measurement, we can also use an analogue-to-digital converter (ADC) to measure the signal as shown in Figure 106. This allows us to further process the signal in the digital domain. After measurement we can for example take the signal off-chip, but the signal is also more robust against noise and distortions.

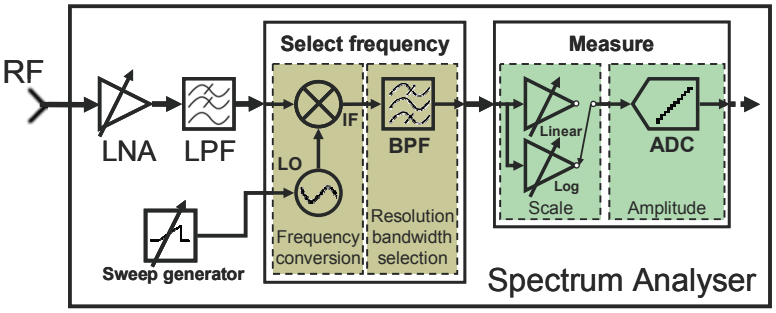


Figure 106. Analogue Fourier transform, digital measurement.

In this case again a limited range is measured with a linear amplifier or the complete range compressed after the log amplifier. The resolution and accuracy of the conversion is limited and adds to the noise and absolute amplitude uncertainty.

We will first analyse the needed resolution of the ADC. It will turn out we can not use the full number of ADC levels together with the log amp, because the quantisation noise is too large, which increases the noise level above -94dBm. This can be remedied if we use a larger range for the log amp.

If we choose 0.5dB of accuracy for an 70dB input amplitude range with a log amplifier, we need 140 measurement or quantisation levels, which means an 8-bit ADC (because the dB range is now linear after the log amp). The least significant bit (LSB) is the distance between adjacent quantization levels. The measurement or quantisation error is half an LSB at maximum [49] (also see section 7.3). This error can be seen as random noise with a 0.29 LSB standard deviation [49]. An 8-bit ADC has 256 levels. If the 70dB input range after the log amplifier is distributed over the full 8-bit, then zero corresponds to -90dBm or -48.0dB under the maximum input of the ADC and one LSB corresponds to 0.27dB. For the 1MHz RBW, the noise level is at -96dBm or 0.93 LSB or -48.8dB from the maximum. The error level is at -58.9dB. This is low enough according to the previous section, but then the DANL was at 1/140 and now it is at 1/256. Indeed, the measurement error increases the noise to $\sqrt{(0.93^2+0.29^2)}=0.97$ or -92.1dBm. Thus, the noise level has become too high. One way to solve this is instead of distributing the 256 levels over 70dB, using 0.5dB as LSB giving a range of 128dB. This corresponds to the 140 levels for 70dB of the previous section. As the quantisation noise level is still at -58.9dB, this should be low enough. Indeed, the noise level at -96dBm then corresponds to 1.48 LSB or -44.7dB, which is increased to 1.52 LSB or -44.55dB or -94.4dBm by the quantisation noise. This also has the added benefit of being able to measure the signal down to -20dBm – 128dB = -148dBm. The DANL drops by selecting a smaller RBW and is at -114dBm for the smallest RBW of 10kHz. These small signals are therefore worthwhile being able to measure.

With the linear amplifier and an 8-bit ADC, we can measure a 48dB sub-range (6dB/bit [48]) of the complete input range. The complete input range being at least -90dBm to -20dBm, but preferable also down to -114dBm.

The maximum RBW is 1MHz, which is a 0.5MHz corner-frequency for the zero-IF case, so the sampling rate must be over 1MHz according to the Nyquist-Shannon sampling theorem. A 1MHz 8-bit ADC should not be a problem (see section 6.2.1). Note that the resolution bandwidth filter also acts as an anti-alias filter (to prevent folding of out-of-band frequencies into the band) as it filters out everything but the frequency (frequencies) of interest [49].

Instead of using a log amp or measuring only part of the amplitude range, we can also measure the complete (uncompressed) input amplitude range if the ADC is precise enough. The linear and logarithmic amplifiers are then unnecessary and the ADC is moved more up to the front. The frequency selection is still performed by the analogue RBW filters, after which the signal is digitalised. The requirements for an 70dB input range are a 16-bit 1MHz ADC as discussed in section 6.2.1. The ±0.5dB amplitude

accuracy and 4dB NF then only apply to the ADC, while the RBW filters reduce the IP_2 and IP_3 requirements to 0dBm and -10dBm respectively.

However, it is preferable to perform the RBW filtering in the digital domain (see chapter 7). This is discussed in the next section.

The signal before rectification in the analogue case also still contains the phase relationship. However, now it is dependent on the local oscillator phase (see section 4.2.1.1). We can use the instantaneous signal amplitude of the in-phase and quadrature channels of the quadrature architecture to determine the amplitude and phase of the signal as discussed in section 4.2. Without a quadrature architecture, the phase is not available. The price is thus that we have to measure two signals.

6.2 Digital

The digital spectrum analyser digitises the signal as early as possible to achieve as much flexibility as possible. Therefore it must support the complete input amplitude range as said. This determines the ADC resolution. Current technology only allows a certain maximum sampling rate however. According to the Nyquist-Shannon sampling theorem, signal with a frequency higher than half the sampling rate can not be measured correctly. Unfortunately the maximum sampling rate is nowhere near the maximum frequency of the input frequency range (3/6GHz), as we will see in the next section. Therefore, we can not move the ADC all the way to the front, which would give a Fourier analyser (section 2.2.2.1). We can place the ADC directly after the frequency conversion. Performing the frequency selection in the digital domain will allow for a lot of flexibility, discussed in chapter 7. The resulting architecture is shown in Figure 107.

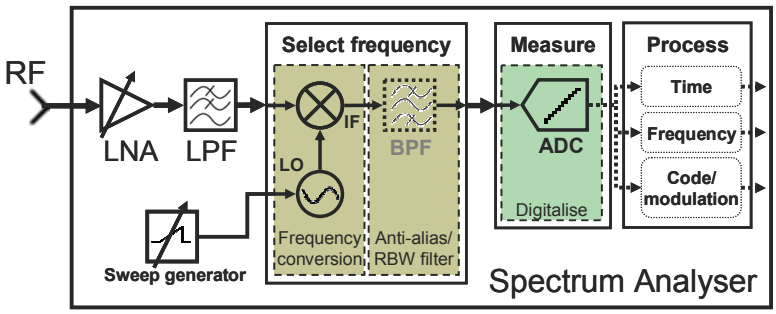


Figure 107. Digital spectrum analyser front-end.

Compared with the analogue spectrum analyser with an ADC to measure the signal, the different RBW filters (7 in total) are not needed anymore, as they can be performed digitally. One analogue filter is still necessary to prevent the folding of higher frequencies into the lower frequency range. This is called the anti-alias filter. For more information see [49] and section 7.3. This anti-alias filter at the same time can act as a RBW filter, with the ADC only performing signal measurement as in the analogue case. Other RBW filters can be implemented digitally. If the ADC operates at the maximum possible sampling frequency, only RBW filters with a smaller bandwidth can be implemented digitally. Higher RBWs can still be implemented the "analogue way" however.

The limitations of the digital spectrum analyser are thus determined by the limitations of the ADC. The ADC requirements and boundaries are discussed next.

6.2.1 ADC

The signal before digitisation still contains the complete range of frequencies of the input signal as we are processing them in the digital domain. However, there is a limit in the sampling speed that we can achieve with an ADC. As discussed in section 2.2.2, the input frequency range is therefore separated in smaller frequency bands that the ADC can handle and which are converted one by one to an intermediate frequency as in the analogue case. The size of this band is to be chosen by us, however first we look at the needed resolution.

As the signal to digitise can contain multiple frequencies with different amplitudes, we can not limit the amplitude range with something like a log amp as above. If we would use a log amp, a small signal would be lost in the conversion whenever the current amplitude of a larger signal is not close to the current amplitude of the smaller signal. In other words, if

the larger signal is not close to a zero crossing in the time domain, smaller signals would be lost with a log conversion. Thus we need enough resolution for the complete 70dB range at all times. A -20dBm sinusoid has a maximum amplitude of 0.032V (at 50Ω). However, note that multiple signals of -20dBm in a RBW can combine to give a larger amplitude than 0.32V. On the other hand, we would also like to see the DANL as a reference. Therefore an 80dB range over -95dBm to -15dBm is proposed. With 6dB of signal-to-noise ratio (S/N) per bit for an ideal ADC, this results in the need for a 14-bit ADC. A 14-bit ADC has an 84dB range from -99dBm to -15dBm.

For the ideal ADC, the S/N is limited by the white thermal noise. In actual ADC there are also other sources of distortion which can be characterised as white noise limiting the S/N with about 1.5 bits on average [48]. Thus the effective number of bits is about 2 bits less. Therefore for the wanted 80dB of resolution we need a 16-bit ADC. Note that this mostly concerns state-of-the-art ADCs, the previous 8-bit ADC from above should be possible with close to ideal performance.

The sampling rate influences the sweep time, as slower ADCs need smaller bandwidths and thus more bands in the same frequency range. Therefore the wanted sampling rate is the fastest reasonably achievable.

State-of-the-art ADC boundaries (1999)		
Resolution	14 bit	16 bit
Sampling rate	20 MHz	5 MHz
S/N, SFDR degradation	2 bit	2 bit
Input voltage	2~10V	2~10V
Power dissipation	0.5~10W	0.5~10W

Table 6. State-of-the-art ADC upper boundaries

Table 6 lists the performance of state-of-the-art ADCs in 1999 for 14~16 bit resolution [48]. With the given resolution, all ADCs at [48] achieve a SFDR of less than 2 bits below the stated number of bits. This means the linearity distortions are below the 14-bit resolution needed, however these are specified at an input voltage of 2~10V, which is 20~33dBm (at 50Ω) thus some amplification might be necessary. It is also stated that the performance improves about 1.5bits of resolution every 8 years for a given sampling rate. The state-of-the-art sampling speed of 5MHz should therefore nowadays (2006) be reasonable to achieve. Again, the maximum RBW is 1MHz, which is a 0.5MHz corner-frequency for the zero-IF case, so the sampling rate must be over 1MHz according to the Nyquist-Shannon sampling theorem. The needed anti-alias filter is then essentially the same as the maximum RBW filter in the analogue case. In section 7.2, we will argue for a 2MHz sampling rate, while over-sampling might also be employed for reducing aliased frequencies (section 7.3.2).

It is possible to subdivide the measured signal further in frequency digitally. If this is wanted, the digitalised signal contains multiple frequency bands, with possible interferers. Therefore, the linearity requirements are not relaxed and the ADC must achieve an IP_2 of over 54dBm and an IP_3 of over 17dBm. Because the spurious free dynamic range (SFDR) degradation is less than 2 bits with input voltages of 2~10V (20dBm~33dBm) [48], this is achievable. For a 16-bit ADC with 14 effective bits (84dB), this means the IP_2 is at least about 100dBm and the IP_3 at least about 60dBm, but this also means we must amplify the maximum input signal of -15dBm with about 35dB before the ADC. This increases the IP_2 requirement to 89dBm and the IP_3 requirement to 52dBm. An ADC design with 14 effective bits is thus always good enough as long as the largest input signal of the ADC is equal to or above -15dBm.

A 16-bit ADC has a quantisation noise of -107dB or -123dBm with a maximum input signal of -15dBm. This is far below the -98dBm DANL (is without the 4dB NF assigned for the measurement block) or 2.61 LSB at a 1MHz RBW. The quantisation noise only increases the DANL to -97.95dBm. This means the ADC still has almost the full allocated 4dB left for other parts of the ADC or even improving the spectrum analyser specifications. However, note that with 14 effective bits, we can only measure an 84dB range instead of the 128dB range with a linear or logarithmic amplifier in front of the ADC as in section 6.1.

6.3 Summary and Conclusion

This chapter discussed the signal measurement following the frequency conversion and frequency selection of the input signal.

6.3.1 Summary

The measurement block was discussed in further detail.

The analogue spectrum analyser handles the signal measurement in the analogue domain. The frequency conversion and RBW select conceptually one frequency. Because only a limited amplitude range can be displayed because of a limited screen resolution and size, a linear amplifier is used to select a smaller part of the amplitude range or a log amp is used to display the range on a logarithmic scale. These amplifiers must achieve a ±0.25dB amplitude accuracy, 2dB NF, 0dBm IP_2 and -10dBm IP_3.

The amplitude of the signal is then measured and displayed. The amplitude accuracy of this measurement must be ±0.25dB (of the ±2dB allowed, ±0.5dB is taken by the low-pass filter and ±1dB by the RBW or anti-alias filter leaving ±0.5dB for the amplification and measurement). Furthermore, the requirements of the signal measurement are a 2dB NF, 0dBm IP_2 and -10dBm IP_3. The measurement can for example be done with an envelope detector or a 1MHz 8-bit ADC. With a linear amplifier, this ADC can measure a 48dB range. With a log amp a 128dB range with 0.5dB precision is preferred for keeping the quantisation noise low enough. The ADC can also be moved up to the front after the RBW filters. A higher resolution ADC measures the entire amplitude range with the frequency selection performed by analogue RBW filters. In that case, the ADC must be 16-bit (14 effective bits), 1MHz sampling rate, ±0.5dB amplitude accuracy, 4dB NF, 0dBm IP_2 and -10dBm IP_3.

The digital spectrum analyser uses an ADC to digitise the complete input amplitude range and moves it even further to the front to just after the frequency conversion. However, an anti-alias filter after the frequency conversion is still needed. This is only one filter instead of multiple RBW filters. The frequency selection can now be performed digitally. An analogue anti-alias filter is still needed, but this is only one filter instead of multiple RBW filters. Without a resolution bandwidth filter before the ADC, the digitised signal now possibly contains a range of frequencies instead of conceptually one frequency. Therefore the ADC must achieve an IP_2 of over 54dBm and an IP_3 of over 17dBm. An 80dB range from -95 ~ -15dBm is proposed including headroom and the DANL. The 80dB input range needs 14 bits (6dB/bit), however we need 2 extra bits because of added noise and SFDR degradation. State-of-the-art ADCs achieve a 5MHz sampling rate at 16bits in 1999. Nowadays (2006), the needed 1MHz 16-bit ADC should be reasonable to achieve. The ADC must achieve a ±0.5dB amplitude accuracy and 4dB NF of which almost nothing is lost to quantisation noise.

6.3.2 Conclusion

An analogue spectrum analyser uses a RBW filter, which relaxes the linearity requirements of what follows. The measurement block consists of a linear or logarithmic amplifier which scales or compressed the input range, and signal measurement by a envelope detector or 8-bit 1MHz ADC.

For a digital spectrum analyser, the ADC can be seen as the measurement part, which is placed after the frequency conversion in front of the frequency selection. A single anti-alias filter is needed. The measured signal's frequency range can be subdivided further digitally, which therefore can contain interferers. This means the linearity requirements are not relaxed, however the resolution and linearity requirements can be met reasonably up to a 5MHz sampling rate. At least a 16-bit (with 14 effective bits) 1MHz ADC is needed.

Because of the added flexibility, while still being able to perform the spectrum analyser functionality with the same specifications as the analogue architecture, the digital spectrum analyser implementation is recommended. This comes at the cost of an ADC and digital signal processor (DSP) needed, which is further discussed in chapter 7.

Chapter 7

7 Digital considerations

This chapter focuses on the low-cost spectrum analyser front-end from a digital perspective. Starting from the basic building blocks repeated in Figure 108, we determined the need for signal conditioning to limit the input amplitude and frequency ranges and a subdivision into a frequency conversion block and a frequency selection block (Figure 109). Following from chapter 6, the frequency selection can be performed in the analogue or digital domain, which differ in how far the ADC is placed to front.

We determined that with a digital spectrum analyser flexibility is gained if the signal is digitalized before the frequency selection. With the current ADC performance, however, frequency conversion is needed to support the complete input frequency range and the ADC can not be moved further forward.

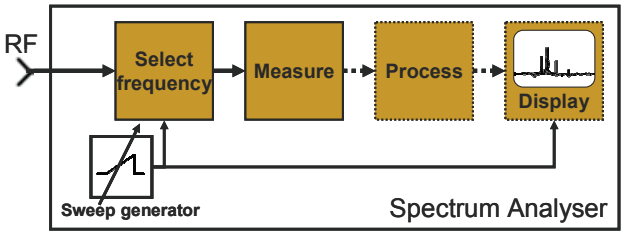


Figure 108. Spectrum analyser building blocks.

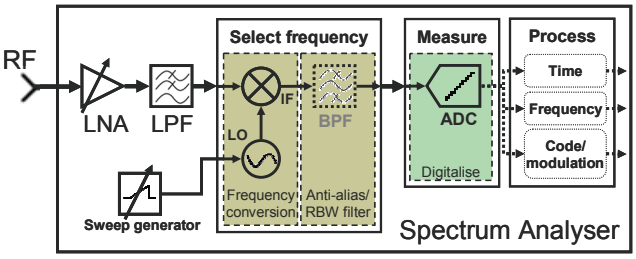


Figure 109. Signal conditioning and select frequency subdivision.

A digital architecture can provide the frequency selection with digital RBW filters flexibly. But in section 2.2.2, we also introduced the Fourier analyser and the vector signal analyser as a Fourier analyser with a frequency conversion front-end. Those digital spectrum analysers use a digital Fourier transform as operating principle. Therefore, we have two operating principles, which are further discussed in section 7.1.

Each operating principle has different consequences for the requirements of the previous front-end blocks. Before we focussed on the RBW based front-end from an analogue perspective. Therefore, each building block is revisited and discussed from a digital perspective. Section 7.2 discusses the frequency conversion architecture, followed by the signal measurement in section 7.3, while section 7.4 discusses the frequency selection. The result is a recommended front-end for a digital spectrum analyser in section 7.5.

7.1 Two operating principles

The two operating principles were already identified in section 2.2. The main operating principle of an analogue spectrum analyser is selecting a frequency with a resolution bandwidth filter. This has been taken as the default case in the preceding chapters. The main operating principle of a digital spectrum analyser is performing a Fourier transform with an FFT. Note there is no fundamental difference, only a conceptual difference (both determine the power spectral density (PSD) over a bandwidth).

7.1.1 RBW based

A spectrum analyser transforms a time based signal to the frequency domain. Mathematically this is performed by a Fourier transform. As explained in section 2.1.3, this transform can be approximated by selecting a small frequency band and measuring the signal power in that band. If measured for an infinite time, this gives the PSD of the input signal. A spectrum analyser thus approximates a Fourier transform by giving the approximate PSD in a small selectable bandwidth, repeats this measurement for each frequency band in a range and provides the resulting graph. The selectable bandwidth is called the resolution bandwidth (RBW), since it determines the resolution (in frequency) of the measurement. This operating principle is therefore called RBW-based.

The RBW-based operating principle uses a resolution bandwidth filter to select a small frequency range and interpret it as a single frequency. With a mixer a frequency range of interest is swept through an intermediate frequency one RBW by one. This is shown in Figure 110.

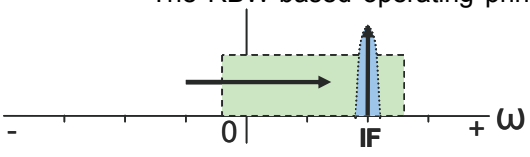


Figure 110. Sliding the input range through the RBW filter.

The different RBW filters can be analogue. With a digital spectrum analyser the ADC is moved so far upfront, that the RBW filters can also be (partly) implemented digitally. The digital filter implementations can achieve a more ideal Gaussian response with an easier implementation (see section 7.4.1). We still need an analogue anti-alias filter, but only one instead of multiple RBW filters. The resulting architecture is shown in Figure 111.

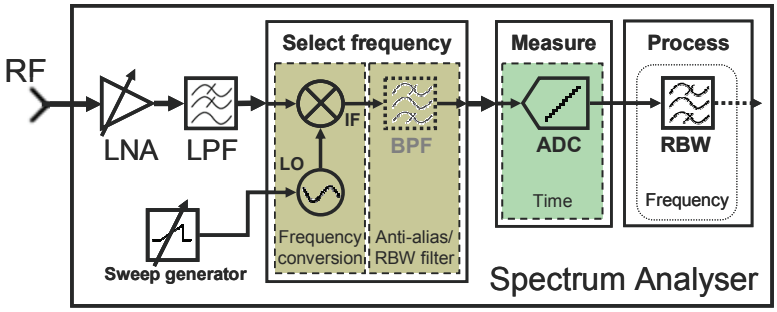


Figure 111. RBW based digital spectrum analyser front-end.

7.1.2 FFT based

The Fourier transform for aperiodic signals defines a continuous frequency spectrum with an integral over infinite time. This means an aperiodic signal is composed of an infinite number of orthogonal sinusoids. The RBW based operating principle approximates the Fourier transform by taking a finite number of frequency bands and measuring for a finite time. Another option is calculating the Fourier transform using a computer or digital signal processor (DSP). Because a computer or DSP uses discrete signals, the Fourier transform is approximated with a discrete Fourier transform (DFT). An optimised implementation of the DFT with respect to computational complexity is the fast Fourier transform (FFT). This results in the architecture of Figure 112.

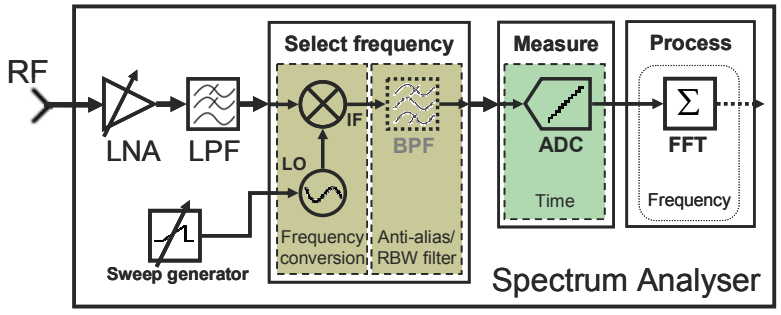


Figure 112. FFT based digital spectrum analyser front-end.

The DFT calculates the amplitude and phase of a sampled signal at a specific frequency point determined by the number of samples and the sampling rate. With an infinite number of samples, there would be an infinite number of frequency points and thus a continuous spectrum. With a finite number of samples, the resolution is limited. The limited number of samples is a discrete, limited time view of the actual signal, called a window. The calculated amplitude at each frequency point then comprises of a frequency band, the shape of which is determined by the shape of the window.

The FFT can thus also be seen as a number of parallel filters corresponding to the number of samples, with their frequency response determined by the window shape. An analogue implementation of this corresponds with the multi-channel spectrum analyser of section 2.2.1.1. The analogue implementation of the FFT based operating principle therefore uses a number of filters in parallel to approximate a spectrum, while the analogue RBW based operating principle, steps through the range one filter at a time to approximate the spectrum. The digital implementation in effect is no different.

The implications of using a DFT are further discussed in section 7.4.2.

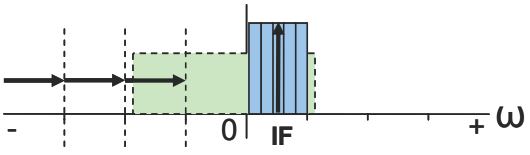


Figure 113. Stepping the input range through the FFT.

The maximum frequency the FFT can resolve is determined by the sampling rate of the ADC. In chapter 6 we found a maximum sampling rate of about 5MHz for the required amplitude range. This means a frequency conversion architecture is still needed to down-convert the signal of interest to the range that can

be digitalised. Then the complete 1MHz band can be transformed by the FFT. Thus for a larger frequency band than 1MHz we must perform multiple FFTs and combine the results. The LO then steps with 1MHz steps for each FFT. This is shown in Figure 113.

7.2 Frequency conversion

For the RBW-based analogue spectrum analyser front-end, a zero-IF architecture was found to be preferred. This is because a high-IF architecture needs to generate high LO frequencies (4~7 GHz for a 0~3GHz input signal), which is difficult on-chip; especially with respect to phase noise. Also the first IF filter is located at a high frequency (above the input range at 4/8GHz), but still needs enough selectivity to filter out the images of the first down-conversion step. Such a filter is difficult to design and integrate.

A low-IF filter, on the other hand, does not have these problems because the signals are down converted to lower frequencies. However, we found the image in that case is a frequency distance of two IFs away. As the IF is low, this is very close and the filter must be very selective, which becomes difficult especially near the upper end of the input frequency range. More importantly, the image is in the input range and changes for each input signal, which means the filter must also be tuneable over the complete input range.

A zero-IF architecture down-converts to DC, which means the image is the signal itself. The image is therefore no problem and a complex filter is not needed, while the advantage of using lower frequencies also applies. Note that although the frequencies the LO generates are lower, they occupy many decades from 0 to 3GHz, while the tuning range in case of high-IF is only a quarter of a decade (4~7GHz).

The RBW-based operating principle selects a resolution bandwidth, which in the case of zero-IF is located around DC. Everything in the bandwidth is conceptually “one” frequency and the image is indeed the signal itself. The spectrum analyser’s resolution can be increased by choosing a smaller RBW, still centred around DC, with smaller LO steps. If the signal is first digitalised or not before the RBW filter does not make a difference.

The FFT-based operating principle first digitalises the signal. The FFT operation transforms the digital signal from the time domain to the frequency domain. Frequencies up to half the sampling rate of the ADC can be differentiated. Higher frequencies are first eliminated by low-pass filtering. Also see section 7.3. The resolution of the result is determined by the number of samples used by the FFT.

The FFT provides the amplitude and phase of a number of frequency bands from DC to half the sampling rate instead of one frequency band around DC as in the RBW case. Each frequency band is not centred around DC but around equally spaced steps from DC to half the sampling rate. Alternatively we can see the band from DC to half the sampling rate as centred around an IF of a quarter of the sampling rate. The RBW and FFT situations are illustrated in Figure 114 and Figure 115. Note that the red areas are not useable because of DC offset problems (see section 4.6.2 and the end of this section).

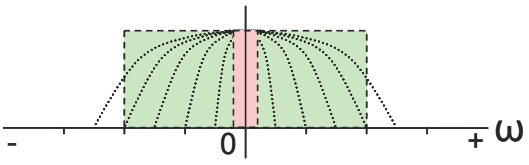


Figure 114. Zero-IF architecture with different RBW filters.

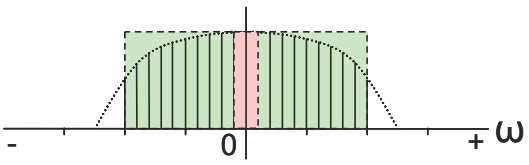


Figure 115. Low-IF architecture with an FFT.

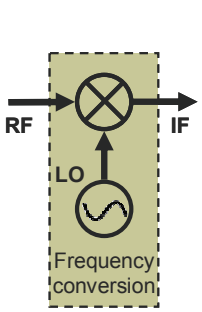


Figure 116.
Normal mixer

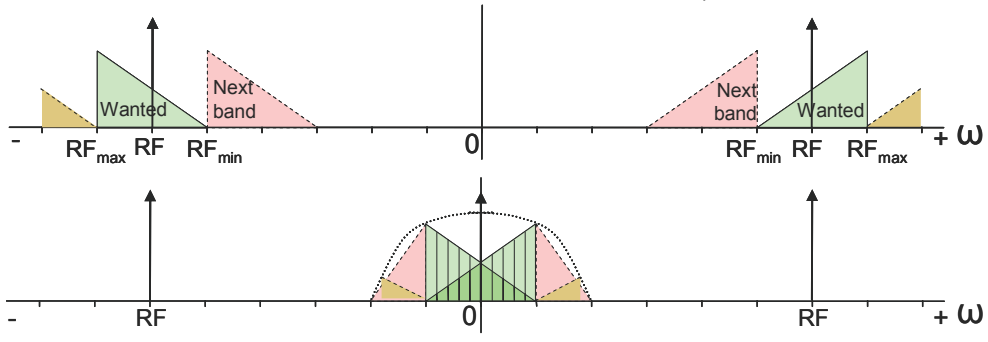


Figure 117. Zero-IF architecture with simple mixer.

The FFT-based operating principle thus uses a low-IF architecture. This means this solution also has the image problem described above. This is illustrated in Figure 117. The upper figure shows the different frequency bands before down-conversion. The lower figure shows, that although the wanted signal ends up under the indicated RBW filter response, the signal overlaps itself with the upper sideband over the lower sideband. For an RBW-based solution, this is not a problem, since the wanted signal band represents one frequency and only the amplitude is needed. For a FFT-based solution this is a problem, since for the indicated frequency bands of the FFT, the upper and lower sidebands overlap. This explains why in effect the architecture is a low-IF architecture with the wanted frequency band starting at DC. Also shown is, that parts of the surrounding frequency band also end up under the frequency response, which can distort the signal, although attenuated.

One of the sidebands becomes the image and can be filtered before down-conversion. But because a highly-selective (therefore probably external) tuneable filter is not an option, a quadrature architecture is recommended as a means of image rejection (see section 4.5). Unfortunately, the quadrature architecture only achieves an image rejection of 40~60dB, which is not enough for the specified 70dB dynamic range.

A quadrature architecture (see chapter 4) can be used to separate the signal into two signals with half the bandwidth of the original signal. These two signals can be used to select positive or negative frequencies, corresponding to the upper sideband or the lower sideband, or to measure amplitude and phase. A quadrature mixer introduces two 90° phase shifted versions of the input signal as well as providing the needed frequency conversion. A second 90° phase shift is needed in one of the two signal paths. Depending on whether the two signal paths are subtracted or added, the upper or lower sideband is selected. The other sideband is the image. This is illustrated in Figure 119 and Figure 120.

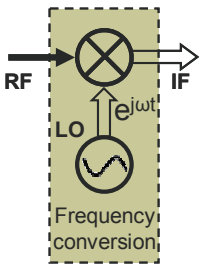


Figure 118.
Quadrature mixer.

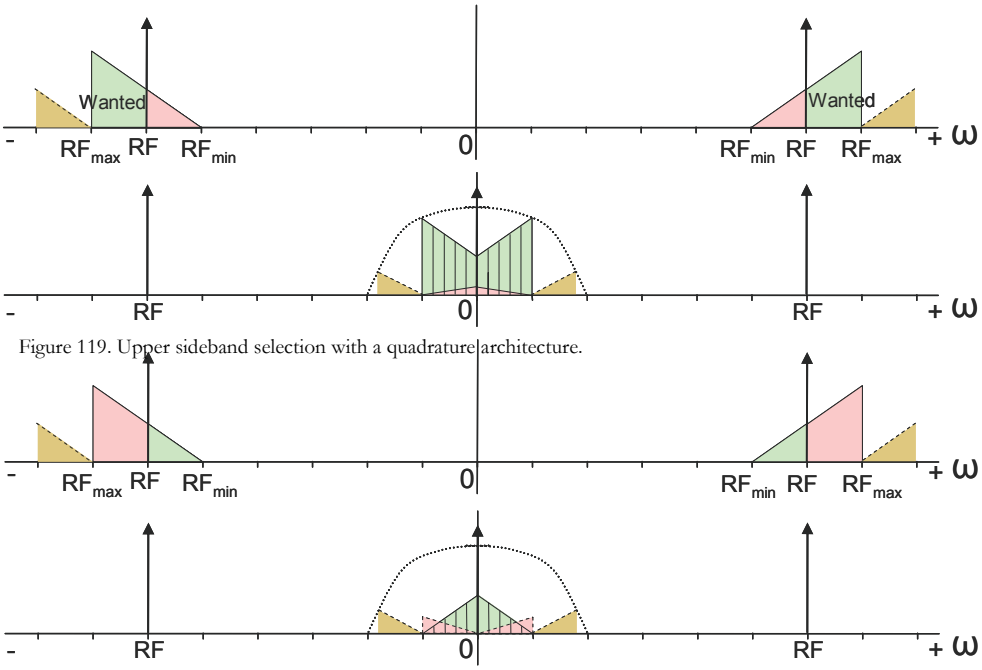


Figure 119. Upper sideband selection with a quadrature architecture.

Figure 120. Lower sideband selection with a quadrature architecture.

When the quadrature signals are combined in the analogue domain, the resulting selected sideband of the signal can be digitalised with half the bandwidth of the original input signal. Another option is to perform the second 90° phase shift in one of the signal paths in the digital domain (for example by using a Hilbert transformer). This means that two ADCs are needed, each half the bandwidth of the input signal. However, the image rejection becomes better because the matching is better and both the upper and lower sideband can be determined in the digital domain.

In Figure 114 and Figure 115, red areas indicated unusable frequency bands, because of DC offset problems. These DC offset are removed, but could contain frequencies of

interest. As discussed in section 4.6.2, for the RBW case part of the signal at DC is distributed back in-band because of local oscillator phase noise or by overlapping measurements. Although this is partly the case for the FFT principle, we are still left with a hole in the spectrum. When combined with previous and next frequency band, they also have a hole. Therefore, we must perform extra FFTs that overlap these holes. As we have to perform extra measurements anyway, we might as well take the opportunity to throw away a larger part around DC, which is distorted by DC offsets. In effect the intermediate frequency is moved to a little higher frequency.

On the other side of the spectrum, it will become clear in the next section that because of finite attenuation of the anti-alias filter, part of the signal of the following frequency band will fold into the current frequency band. This is largest at the high end of the spectrum. Therefore part of the signal at that end is also distorted and best not used.

It is proposed to increase the anti-alias filter corner-frequency from 0.5MHz to 1MHz and the ADC sample rate from 1MHz to 2MHz. For the RBW principle, this means an extra larger RBW of 2MHz, while the previous 1MHz will be implemented digitally. This also means the DANL increases with 3dB because of the doubled bandwidth. However, from chapter 6, we found that 4dB of NF is still available for the ADC. For the FFT-principle, the first and last 0.25MHz are thrown away, leaving a 0.5MHz usable band (Figure 121). By using both the upper-sideband as the lower-sideband, this gives a 1MHz frequency band as result each measurement. The measurements are interleaved to cover the holes as illustrated in Figure 122. The LO steps are 0.5MHz apart, each time measuring two steps and then skipping two steps. The IF is at 0.5MHz. Of course this scheme can also be used with the original 0.5MHz corner frequency.

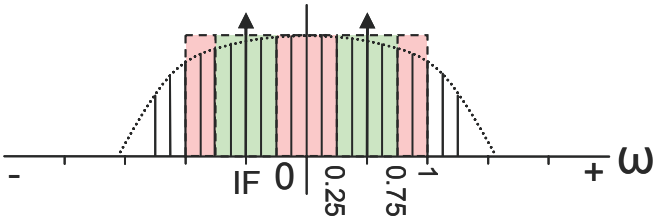


Figure 121. Usable and unusable FFT bands.

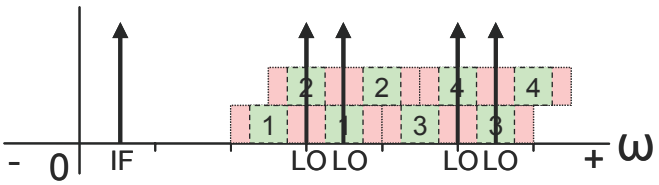


Figure 122. Interleaving of LO steps.

So the spectrum analyser front-end up to the frequency selection can be used as a zero-IF architecture with digital RBW filters or the same architecture can be used as a low-IF architecture with a FFT. The high-IF architecture is not suitable for a low-cost (highly integrated) spectrum analyser front-end. For an FFT-based operation, the zero-IF is not possible, which leaves us with the low-IF solution. This comes at the cost of a reduced dynamic range. Interleaving is used to make the DC offset and aliasing problems easier to handle.

7.3 Measurement

For a digital spectrum analyser, one can argue that the signal is measured when it is converted from analogue to digital (ADC). Each sample represents after all the instantaneous amplitude at the time the sample is taken.

In section 6.2 we determined the ADC is best placed as far to the front as possible, to provide the most flexibility. The result is the placement right after the frequency conversion. Further frequency selection is performed by signal processing in the digital domain. This is discussed in the next section.

The input range of 80 dB (-95 ~ -15dBm) determines the need of 14 effective bits. The SFDR is specified at 70dB, which corresponds with 12 effective bits. The extra 10dB is for headroom and to measure the noise level Because the S/N degradation as well as the SFDR degradation is about 2 bits (section 6.2), a 16 bit ADC is needed to support the input amplitude range. The maximum frequency sampling rate with these specifications is found to be at about 5MHz (in 1999, see section 6.2.1).

As found above, without a RBW filter before the ADC, the digitised signal now contains a frequency range instead of conceptually one frequency. This also means that

the linearity requirements of the ADC are not lowered because of filtering as was done with the RBW filter. Thus the ADC must achieve an IP_2 of over 54dBm and an IP_3 of over 17dBm just as the block before it, as also discussed in section 6.2.

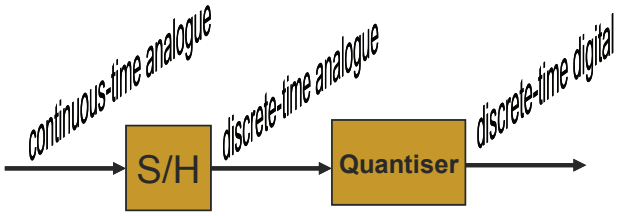


Figure 123. Continuous-time analogue to discrete-time digital.

The conversion of a continuous-time analogue signal to a discrete-time digital signal is performed in two steps. The conversion from continuous-time to discrete-time is done by the sample-and-hold (S/H) block. The conversion from an analogue signal to a digital signal is performed by the quantiser. Analogue to digital conversion thus strictly taken only consists of the the quantisation, but in practice does both.

7.3.1 Sample and hold

An ADC first samples a continuous-time analogue signal, which turns it into a discrete-time analogue signal. The Nyquist-Shannon sampling theorem shows that the maximum frequency that can be identified after sampling is half the sampling rate. It can also be shown that higher frequencies appear as lower frequencies after sampling, which is called aliasing (see [2] and [49] for example). Frequencies from half the sampling rate up to the sampling rate fold to half the sampling rate back to zero with 180° phase shift. From the sampling rate up to one and a half times the sampling rate becomes zero to half the sampling rate with 0° phase shift after sampling. From one and a half to two times the sampling rate, the range is folded again and so on.

Sampling can be modelled by multiplying the input signal with an impulse train. In the frequency domain the signal's spectrum is convolved with the impulse train. Because of time-frequency duality, close spaced impulses in the time domain are far away in the frequency domain. The spectrum thus consists of duplicated spectra the sampling frequency apart. If the spectrum of the input signal is wide and the sampling frequency is not high enough, these duplicated spectra can overlap resulting in the discussed aliasing. The input signal thus can not contain frequencies higher than half the sampling rate in order to prevent aliasing. The higher frequencies must be filtered out with the anti-alias filter before the ADC.

7.3.2 Anti-alias filter

As evident from chapter 5, the attenuation outside the pass-band of a filter is limited. This means the sampled signal at a certain frequency is distorted by a small amount because of aliased frequencies.

The smallest signal we would like to measure distortion free is 70dB below the largest signal (the SFDR). For the filter we found a Gaussian (approximation) filter is preferred because of the well-balanced time and frequency domain behaviour. Because the digital spectrum analyser still uses a stepping LO, this is still applicable as the trade-off is between the number of steps needed for the selected band and the time to wait for the transients to settle. We can see in Figure 103 in section 5.6, that it takes about six RBWs for the signal to fall under 70dB. It is also discussed that in order for the frequency response in the pass band to add a maximum of 1dB of amplitude inaccuracy, the -3dB corner-frequency of the filter must lie at 1.74 times the RBW bandwidth.

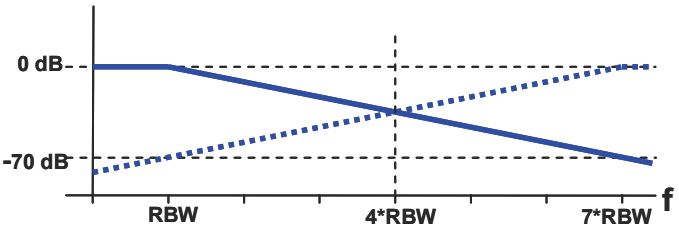


Figure 124. Distortion because of aliasing with a Gaussian filter and four times over-sampling.

One option to improve the selectivity is to sample at a higher sampling rate than double the RBW and take a higher resolution FFT. The frequencies above the RBW corner frequency are not used and can be distorted by aliasing. The folding of the frequencies from half the sampling rate (=RBW) ensures that if the sampling rate is 4

times higher, only the frequencies (4-1)*2=6 times the RBW away fold into the range of interest, as shown in Figure 124.

However, with a four times higher sampling rate, this means four times as many LO steps are needed for the same frequency band as would be needed if we could use the complete FFT result. Thus again higher selectivity comes at the cost of longer sweep times. The total time needed to sample each step remains the same, because both the sampling rate and the number of samples have gone up.

One can imagine an analysis of the trade-off between a filter with a higher selectivity and a longer settling time, versus a shorter settling time with more steps. For example, the Gaussian filter settling time plus the sampling time must be at least less than half the time the settling and sampling with a sharper filter, with 70dB attenuation two RBWs away, takes. In the latter case we only need a two times higher sampling rate instead of four and thus halve the steps, but the 70dB attenuation must be reached at a three times lower frequency. This seems to point to the advantage of the Gaussian filter solution, because each doubling of the sampling rate and thus sweep time gives four extra RBWs of distance between the frequency range of interest and aliased signals. This is however not further analysed in this report.

This extra selectivity at the cost of sweep time can of course be made optional. A possibility would be to select a certain attenuation of the aliased distortion signals when using an FFT. With the option off and thus without over-sampling, the result would be the same as in the case of using a RBW filter; the filter shape is visible as the signal is stepped through it.

7.3.3 Quantiser

The discete-time analogue signal is turned into a discrete-time digital signal by the quantiser. A quantiser divides a certain range which is defined for the ADC into a number of discrete levels. The maximum input signal corresponds to the largest digital value. The smallest signal corresponds to a digital zero. The distance between two quantisation levels is the peak input value divided by the number of levels, also called the least-significant-bit (LSB). An analogue signal at a certain instance in time is somewhere between two quantisation levels and thus a maximum of half a quantisation distance away. The input signal normally has equal-likelihood of being anywhere between two levels. The quantisation error is than uniformly distributed with a standard deviation of about 0.29 divided by the number of quantisation levels [49] or 0.29 LSB. Quantisation thus can be seen as adding 0.29 LSB of random noise.

This random noise is the noise level of the ADC. For a 16-bit ADC, this is $10\ log(0.29/2^{16})^2$ = -107dB. This noise level is added to the noise of the signal. The smallest supported input signal results in a digital zero as well as all the signals smaller than that. Thus smaller signals can not be measured.

For an ADC with 14 effective bits, with the maximum input signal normalised to 1 (0dB), the lowest spurious and noise free signal level is 1/16384 or -84.3dB. The quantisation noise is $10\ log(0.29/2^{14})^2$ = -95dB, because the 2-bit signal to noise ratio (S/N) degradation of the 16-bit ADC increases the noise level by 12dB. The maximum DANL before the ADC is -98dBm, which is -83dB (with a -15dBm maximum ADC input) for the 1MHz RBW case or -80dB for the 2MHz anti-alias/RBW filter case. The quantisation noise increases these to -82.7dB and -79.9dB. Thus the NF is 0.3dB or 0.1dB. As we have 4dB of NF specified for the measurement, this is very good.

The amplitude range to be digitalised must probably be scaled to the range the ADC expects with a linear amplifier. This amplifier can have a maximum NF of 3.7dB, assuming no further noise is added. This should be no problem (section 3.2.1). Furthermore, the amplifier and ADC together must achieve an amplitude accuracy of 0.5dB.

One advantage the linear amplifier adds is the option to amplify smaller signals than the -99dBm ~ -15dBm to the 84dB ADC range. As discussed in section 6.1, the amount of amplification of the digitised signal depends on the largest signal in the ADC bandwidth, as too large signals can cause an overflow. This concerns the entire digitised bandwidth, which is possibly larger than the selected (digital) RBW. If there are no large signals in the bandwidth, selecting a smaller (digital) RBW lowers the noise level and allows one to

analyse smaller signals, possibly going as far down as -134dBm for a 100Hz RBW. For the FFT principle, the noise level depends on the number of bins used and the quantisation noise. Because the noise level is already close to smallest signal level at the ADC bandwidth, the FFT noise level quickly drops with a higher resolution and becomes limited by the quantisation noise.

7.4 Frequency selection or transformation

In a digital spectrum analyser architecture, the signal measurement consisting of digitalising the time-domain signal is moved forward, so that the frequency selection or transformation to the frequency domain can take place in the digital domain. This provides significant advantages in flexibility, speed and resolution.

7.4.1 RBW filter

For the RBW-based operating principle, the RBW filter performs the frequency selecting action. This RBW filter can be implemented digitally.

Digital filters work by implementing the filter's impulse response and use it to calculate the resulting output of the digitalised input signal. This can be done in two different ways; by convolution or by recursion, which are discussed next.

7.4.1.1 FIR filter

The input signal can be convolved with the impulse response to determine the output, which means weighting the effect of each input sample for a certain output sample. The filter's impulse response is called the filter kernel. After passing an input samples through the calculation a finite number of output samples results after which the result remains zero. These filters are therefore called finite impulse response (FIR) filters. Any possible linear filter can be implemented by their impulse response. However, for a long impulse response, a long convolution has to be performed with many input samples. The number of impulse response samples determine the performance of the filter and the number of calculations needed to calculate it. For each sample one filter-"tap" is needed consisting of a multiply-accumulate (MAC) and a delay. Since the output of a digital filter implementation is the same (except for it being sampled) as for an analogue implementation with the same impulse response, the same speed-selectivity trade-off exists.

An important advantage, however, is that the impulse response does not have to be causal. This is because the signal is first digitalised and then passed through the filter. The origin can be somewhere in the middle, with the first samples representing negative time. The *sinc* function of section 5.1 can therefore be applied to an input signal, resulting in a rectangular window in the frequency domain. Of course the *sinc* function extends to negative and positive infinite time and must be truncated. The approximation can still achieve exceptional selectivity if the impulse response is long enough. This corresponds to the *windowed-sinc* digital filter, with very large roll-off and stop-band attenuation. A selective filter has a long ringing impulse response and needs many input samples before settling and providing a stable output signal.

As said, because many input samples are needed, it takes a long time for the filter's output to "settle". The convolution is calculated for each output sample with the contributions of each input sample, or one output sample at a time with a number of input samples corresponding to the impulse response size (see [49]). The convolution operation is therefore very computationally extensive and the execution can be slow.

The phase response is also determined by the impulse response. We found in section 5.2.1, that we want the phase response to be zero or linear. Zero or linear phase is very easy with digital filters, by simply making the impulse response symmetrical. [49]

7.4.1.2 IIR filter

A FIR system such as a FIR filter only depends on its input. If a single impulse is provided as input, it will pass through the system and produce an output. But the output becomes and staya zero after the impulse is gone through all the samples of the impulse response,

making the impulse response finite. Therefore there exists no feedback in the system and the system is inherently stable. The transfer function of such a system has all its poles located at zero (in a z-plane).

In the analogue domain, the FIR filter has no counter-part as the system's output always depends on previous outputs (the state of the system, i.e. the voltage in a capacitor). Therefore such a system depends on its inputs as well as on previous outputs. The impulse response of the system can therefore theoretically extend to infinity, which is why it is called an infinite impulse response (IIR) system. As the output depends on previous outputs, the system has feedback. This can be a problem because the finite precision of digital system causes the rounding errors to add and because of stability. The transfer function has both variable zeros and poles. If the poles are located outside of the unity circle (in a z-plane), the system is unstable.

A second digital filter method is thus by using both the input samples and the previously calculated output samples, giving an infinite impulse response (IIR) filter. The result is a set of recursion coefficients, which are called that because they can be used recursively for infinite time, even though there may no longer be an input signal. This is analogous with feedback and the corresponding zeros and poles in the analogue domain. For example, a single pole can be represented with two coefficients. Their advantage is long output responses can be calculated with only a limited number of input samples and calculations. Therefore they execute much faster than FIR filters. Their disadvantage is they can become unstable (FIR filters are inherently stable) just as their analogue version and they have less performance and flexibility [49].

Recursive filters use past input and output values and are therefore causal[12]. This gives the same limitations in selectivity as analogue filters have. In fact analogue filters are very easily implemented as digital IIR filters by entering their parameters as coefficients. The performance of the digital version is much better, because the digital values are more accurate then component values for analogue filters. This inaccuracy limits the order before becoming unstable in the analogue domain. In the digital domain the order is limited by inaccuracy because of round-off errors. The causality caused non-linear phase because the impulse response is not symmetrical, which can be resolved by passing the signal through the filter twice; once normally and once reversed. The selectivity however remains limited compared to the *windowed-sinc* filter. [49]

7.4.1.3 RBW shape

A digital filter can perform much better than an analogue one, but still has the same fundamental speed-selectivity trade-off. Therefore a Gaussian filter is still recommended for the RBW-based operating principle as in chapter 5.

An ideal Gaussian filter can be approximated by a FIR filter. The number of impulse response samples determines the accuracy. Simulation with MATLAB shows that at least 15 taps are needed for a Gaussian response down to -74dB. Note that this is an ideal Gaussian response with shape factor[13] 4.43 instead of 5.7 for a 15th order synchronously tuned analogue filter approximation, thus giving a better selectivity than a comparable analogue implementation.

Hence, digital filters can be much more selective if you are willing to wait for it. Also their phase response can be perfectly linear. Furthermore, the used coefficients are much more accurate and are limited by round-off errors depending on the precession of the digital values used. Most importantly however, digital filters are very easily implemented. The filter can be changed by using a different filter kernel or recursion coefficients. While normally Gaussian filters would be used because of their speed-selectivity balance, highly selective *windowed-sinc* filters can be made optional when such a high precision is needed. The bandwidth of the RBW filter can also be very flexible. Many RBW choices can be easily provided or even calculated from the provided bandwidth "on-the-fly".

[12] Non-causal IIR filters can be implemented by separating the transfer function in a causal and anti-causal part. The input signal is then passed through the causal part, time reversed and passed through the anti-causal part and finally time-reversed again. Note that the signal must be truncated to perform time reversal.

[13] The shape factor is the -3dB to -60dB bandwidth ratio (i.e. BW_{-3dB}:BW_{-60dB}), which is 4.43 for an ideal Gaussian shape.

7.4.2 FFT

The fast Fourier transform (FFT) is a fast implementation of the discrete Fourier transform (DFT). The DFT is the discrete version of the Fourier transform, although when using a finite number of frequencies (results) only periodic signals can be transformed. In that case it strictly is the discrete version of the Fourier series. An infinite number of frequencies is needed to represent an aperiodic signal. Because the number of DFT results is always finite to be useful, the aperiodic signal is approximated by a periodic signal with a long enough period to capture the essential signal part.

The Fourier transform is discussed in section 2.1.3. An analogue signal must be sampled to get a digital signal. This has an effect on the spectrum of the signal. Furthermore, the Fourier transform is an integral over infinite time, and the therefore the discrete counter-part being a summation over infinite time. This is approximated by taking a finite number of samples. This snapshot of the signal for a certain amount of time is called a window. By assuming the samples outside of the window are zero, we can calculate the DFT. The window also has an effect on the spectrum of the signal.

The effect of sampling the signal is the introduction of duplicated spectra of the input signal spectrum separated by the sampling frequency. These aliases can overlap as was discussed in section 7.3.

Next a number of samples must be taken to compute the DFT with. If the samples are taken as is, this can be seen as a multiplication by a rectangular window with amplitude one inside and zero outside, selecting a subset of all the possible input signal samples. As we know from section 5.1, the Fourier transform of a rectangular window is a *sinc* function. The sampled signal spectrum is thus convolved with the *sinc* function. A different window shape has a different Fourier transform and convolves the signal spectrum with a different window spectrum. This window spectrum determines how much of the surrounding frequencies are taken into account when determining the amplitude and phase at a certain frequency and therefore determines the ability to differentiate two close frequencies. The window is further discussed below.

The DFT transforms the windowed time-domain signal to the frequency domain. The result of N input samples is N complex values representing amplitude and phase at N equally spaced apart frequencies. This can be represented by multiplication of the continuous frequency domain of the windowed sampled signal with an impulse train of half the sample rate divided by N apart. Sampling the spectrum results in a periodic signal in the time-domain, because the continuous spectrum has become discrete (see Fourier series versus Fourier transforms in [2]). The window determines the period and resulting DFT represents the input signal inside the window is continuously repeated.

7.4.2.1 Spectral leakage

A signal can be measured for only a limited amount of time, resulting in a finite number of samples, which can be seen as a rectangular window on the original signal. But also, a finite discrete number of results are calculated. As aperiodic signals have an infinite number of frequency components, while only a finite number are calculated, the input signal is assumed to be periodic by the DFT. The finite number of samples are taken as one period of a periodic signal. However, this means that if a sinusoid does not fit in the window with an exact number of periods, a discontinuity is created when concatenating a duplicate of the window's content. A discontinuity has a very wide frequency spectrum (the Fourier transform of an impulse is 1 for all frequencies). This causes the sinusoid's energy to spread out into all other frequencies. The convolution of this spectrum with the Fourier transform of the window function, causes side-lobes around the original sinusoids frequency. This is called spectral leakage as some of the frequencies energy leaks into surrounding frequencies.

Spectral leakage is thus caused by the finite measurement time and the assumption the signal is periodic. The signal-to-noise ratio will degrade because of it and frequency components can be masked under the sidebands.

If all the sinusoids in a signal all fall exactly into the window with some multiple of their periods, then no discontinuity is created by concatenating the window with itself. The original signal is then accurately represented in the digital domain for all time. With no

discontinuity giving energy at all frequencies, the convolution with the window transform does not smear out the frequencies in the signal and clear frequency components result from the DFT.

A sinusoid may fall exactly into the window with multiple periods, but still generate unwanted higher harmonic frequency components because of quantisation. This is because for each period of sinusoids, the sampling takes place at the same position. For example, if we have four periods and 32 samples, each period is sampled 8 times, but these 8 samples are at the same position for the following periods. Normally, this is not a problem, but when the signal is quantised, the quantisation error is the same for each period and thus repeats 4 times. The quantisation noise thus is correlated and has the same period as the sinusoid, but since it is not a pure sinusoid it also contains higher harmonics. These harmonics then appear in the resulting spectrum.

7.4.2.2 Window

It is very unlikely that any real-word signal will fit exactly into the window. Another solution for the spectral leakage problem would be to make sure that the signal at each end falls to zero. This can be done by scaling the input samples depending on their position; i.e. applying a different window.

Note that these windows can also be applied to obtain a better filter kernel for a FIR filter, instead of truncating the impulse response with a rectangular window.

The FFT can be seen as performing many filters in parallel, with the filter shape determined by the window in the time domain. The window in the time domain therefore determines the resolution (ability to distinguish two closely spaced frequencies). An *N*-point FFT provides the amplitude and phase at frequencies half the sampling rate divided by the number of points apart. Just as with RBW-based approach, the frequencies in between also contribute to the result because of the convolution with the Fourier transform of the window shape. The filter shape causes surrounding frequencies to contribute to the result at one frequency (the inverse view of contributing itself to surrounding frequencies, being spectral leakage).

The FFT result is therefore also a spectral density. One frequency result is hence called a frequency bin. A frequency bin conceptually represents one frequency, just as the RBW does. Consequently we would like the surrounding frequency bins not to affect the result of the current frequency bin. For a 70dB spurious-free dynamic range, the window spectrum at the next frequency bin must be at least 70dB attenuated, 74dB to provide a margin. Otherwise, the filter shape becomes visible, again as in the RBW case.

Many different window functions can be devised that fall to zero at the edges. These result in different window shapes in the frequency domain. In the frequency domain, the main lobe is centred around zero or DC and must pass the frequencies unchanged. The width of the main lobe determines the resolution, i.e. the ability to differentiate two closely spaced frequencies. Away from the main lobe are the side lobes, which pass the unwanted frequencies. Therefore, the attenuation of the main lobe to the first side lobe determines the spurious free dynamic range. This is because surrounding frequencies at the side lobe location become part of the signal of interest with that much attenuation. We would like the dynamic range to be as high as possible. In the case of the spectrum analyser the SFDR is specified at 70dB. The total area under the side lobes determines how much the signal of interest is distorted by other frequencies. Of course a higher attenuation is achieved at the expense of a wider main lobe and therefore a lower resolution.

The following side lobes also add to the distortion at the signal of interest, for some windows with a higher attenuation, for others with the same attenuation. If we design the side lobes at -74dB, all the side lobes combined can only increase the distortion 4dB before the SFDR becomes too low. Because only a small number of large signals is assumed, this is feasible (small signals are attenuated to such small levels they are negligible).

Next, a number of common window functions are provided and analysed. They are generated using the "Window Design & Analysis Tool" of MATLAB. Each window is 32 samples of length and is shown on the left. The frequency response on a semi-logarithmic scale is displayed on the right on a normalised frequency scale, with each bin corresponding to 1/32=0.03.

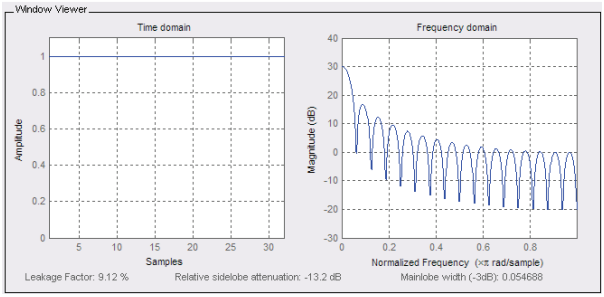


Figure 125. Rectangular window function.

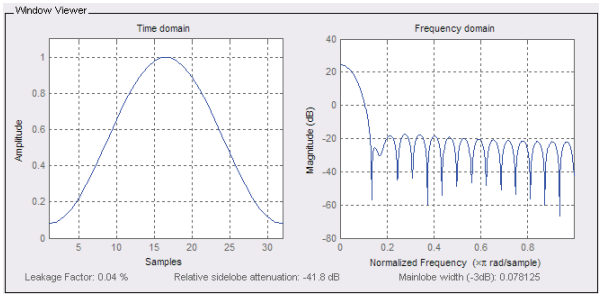


Figure 126. Hamming window.

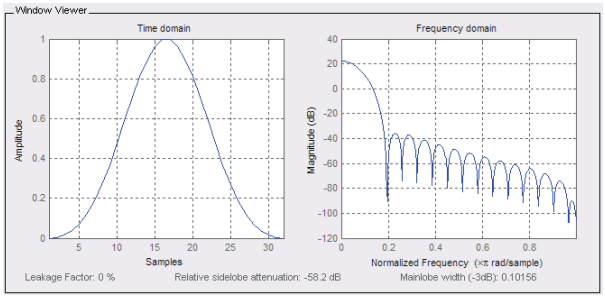


Figure 127. Blackman window function.

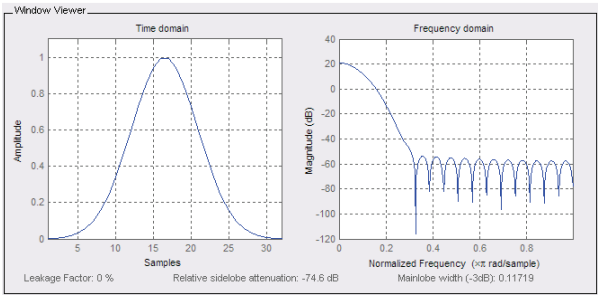


Figure 128. Gaussian window function (α=3.62).

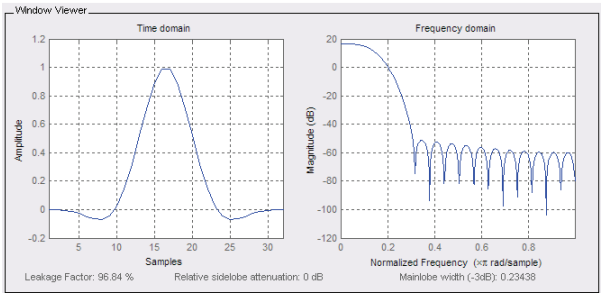


Figure 129. Flat top window function.

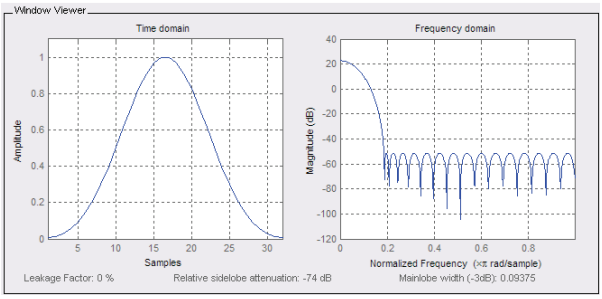


Figure 130. Chebyshev window function (74dB attenuation).

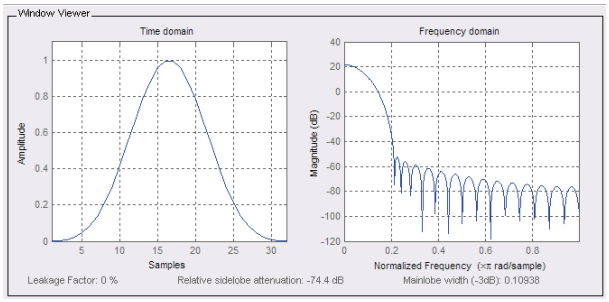


Figure 131. Kaiser window function (β=10).

The first window is the rectangular window. This window has the smallest main lobe width (-3dB point) of 0.05. However, the side lobe attenuation is also smallest at -13.2dB and falling to about -30dB.

The Hamming window increases the attenuation to a constant -41.3dB at the cost of a 0.08 main lobe width. The Blackman window continues this trend with a -58.2dB attenuation and falling at a 0.10 main lobe width. The flat top window is optimised to have a flat pass and response at the cost of a significantly wider main lobe of 0.23 at a reasonable attenuation of 70dB.

All the previous windows do not provide the wanted attenuation of -74dB however. The Gaussian, Chebyshev and Kaiser window function all have a parameter that allows one to specify the amount of attenuation. At α=3.62, the Gaussian window gives -74.6dB of attenuation with a 0.11 wide main lobe, but the response falls slowly. The Chebyshev window improves this with a 0.09 wide main lobe at -74dB.

The main lobe width versus side lobe area trade-off can be optimised. The Kaiser window provides a near optimal result [6]. For β=10, the attenuation is -74.4dB, but at a

larger width of 0.11 than the Chebyshev function. This is because the attenuation keeps falling, making the total area under the side lobes less than the Chebyshev window. The Kaiser window is equivalent to many other windows at different β factors [6]. For large β, the window becomes Gaussian.

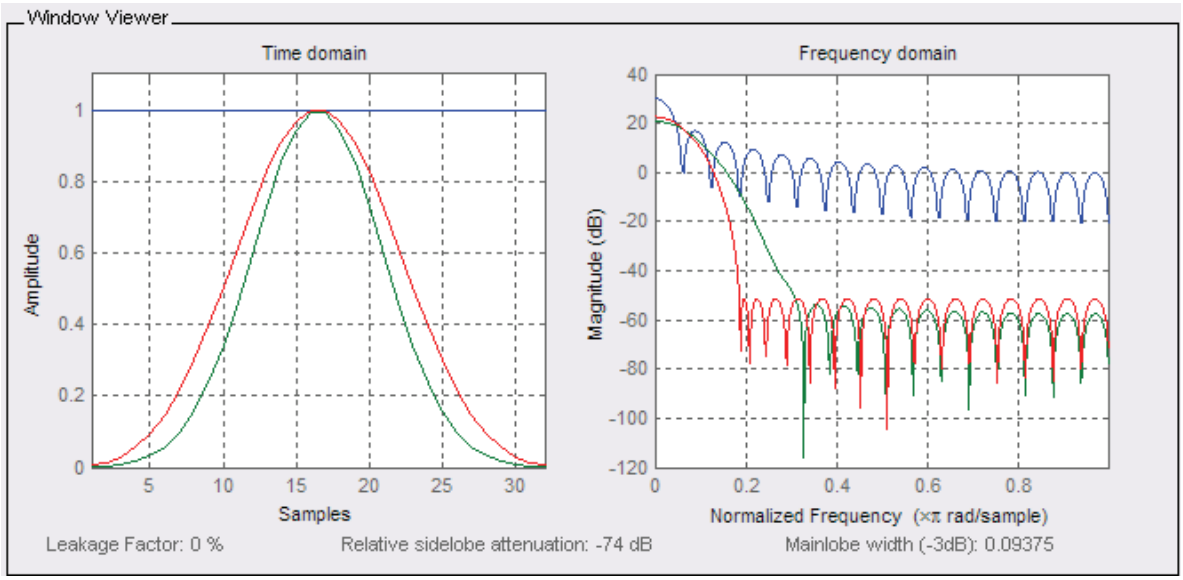


Figure 132. Rectangular versus Gaussian versus Chebyshev window functions.

Figure 132 shows the rectangular, Gaussian and Chebyshev window functions in one figure. The rectangular window clearly does not provide enough attenuation. The Gaussian and Chebyshev windows do, but at a wider main lobe. The Chebyshev window does this at a much better resolution. Note that the main lobe of the Chebyshev window is about three times wider than the rectangular window.

The Kaiser window is more optimal with respect to the total area under the main lobe. It is a good alternative for the Chebyshev, but because the number of large frequencies is expected to be limited, the Chebyshev is preferred for it's better resolution.

Note that the selectivity versus speed or time trade-off of analogue filters does not apply to the windows because the application of the window to the input samples can be performed almost instantly in the digital domain. One does not have to wait for the result to settle. Thus the time needed to perform the FFT is not affected. However, this trade-off is replaced by a resolution versus attenuation one.

7.4.2.3 Specifications

The signal bandwidth after the ADC is 1MHz. By using both the upper and lower sidebands with two ADCs, the total bandwidth is 2MHz. A separate FFT is performed for each side band. The number of samples or points *N* determines the resolution. With a 1MHz resolution, only a 2-point FFT is needed, giving the amplitude and phase of the 1MHz bandwidth. The number of points can easily be increased to a 1024-point FFT for a resolution of about 1Hz. The computational complexity of the FFT roughly corresponds to *N log N*. With today's processing power, high resolutions are possible. For a low-cost spectrum analyser however, a 1Hz resolution would be enough. This resolution is much better than the 10kHz from the analogue case.

The computation of the FFT can be performed as soon as enough samples are available and can be much faster than the RBW principle. Thus the sweep time is limited by the needed complexity of the FFT and the needed number of 1MHz bandwidths (or 2MHz bandwidths with two ADCs). If interleaving is required for DC offset and aliasing problems, this sweep time doubles, but is still much less than the accumulated settling time of the RBW filter for a large number of LO steps.

Besides the amplitude at a certain frequency, the FFT also returns the phase which can be useful in the code/modulation domain for example.

7.5 Summary and Conclusion

This chapter discussed the spectrum analyser from a digital perspective. The digital spectrum analyser gains a lot of flexibility by digitalising the signal as early as possible. This digital spectrum analyser architecture is shown in Figure 133.

7.5.1 Summary

The digital spectrum analyser provides many advantages. The most prominent are the flexibility, higher resolution and higher speed. Furthermore, time and code/modulation domain signals can also be handled by the analyser. The disadvantage is the need for an ADC and DSP.

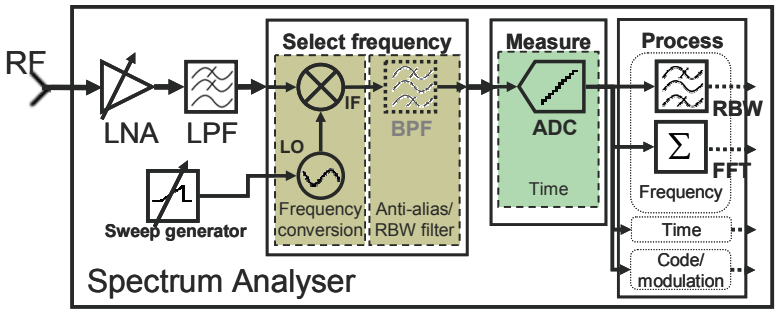


Figure 133. Digital spectrum analyser architecture.

Two operating principles were identified, which both approximate the power spectral density of the input signal (and are therefore not fundamentally different). The RBW-based operating principle uses a filter to select a small frequency band conceptually representing one frequency. Of this band, the amplitude and possibly phase is measured. The frequency band of interest is stepped through the filter, one RBW at a time. For the digital spectrum analyser, the signal is digitalised after the frequency conversion and the RBW filter is applied digitally.

The FFT-based operating principle applies a fast implementation of the discrete Fourier transform to the time-based digitalised signal. The FFT operates on a frequency band at once. The resolution of the result is determined by the number of samples used.

A single front-end can be used for both principles by placing the ADC in front of the frequency selection. The spectrum analyser front-end up to the frequency selection can be used as a zero-IF architecture with digital RBW filters or the same architecture can be used as a low-IF architecture with an FFT. The RBW-based zero-IF solution has a high dynamic range of 70dB, but is slower and has less resolution. The first because of the parallel processing of the FFT versus the sequential stepping of the RBW-based solution. The second because of the limit in the smallest RBW filter because of DC offset.

The high-IF architecture is not suitable for a low-cost (highly integrated) spectrum analyser front-end. For an FFT-based operation, the zero-IF is not possible, which leaves us with the low-IF solution. The FFT-based low-IF solution is faster and has a higher resolution, but has a lower dynamic range of 40~60dB. This is because the low-IF has image problems. The images are very close to the wanted signal at the input spectrum, because the IF is low. Therefore a quadrature architecture is needed for image rejection. This architecture however is limited because of matching to 40~60dB of attenuation. Interleaving is used in case of the FFT to make the DC offset and aliasing problems easier to handle.

The needed ADC is 14 effective bits for an 84dB (-99dBm to -15dBm) range and a 2MHz sampling rate. An ADC consists of a sample-and-hold, which turns the continuous signal into a discrete signal, and a quantiser, which turns the analogue signal into a digital signal.

The sampling can be seen as multiplication of the input signal with an impulse train, which causes duplicated aliased spectra, each a sampling rate apart, in the frequency domain. If the bandwidth is too large, these spectra can overlap and must therefore be filtered by a 1MHz bandwidth anti-alias filter.

The quantisation causes the noise floor to increase, but within limits. A linear amplifier optionally can be used to scale a wider amplitude range into the 84dB range of the ADC, as long as the larges signal in the band does not cause an overflow.

The frequency selection is performed by a digital RBW filter or by an FFT.

A digital filter can perform much better than an analogue one, but still has the same fundamental speed-selectivity trade-off. Therefore a Gaussian filter is still recommended for the RBW-based operating principle as in chapter 5. The digital Gaussian filter has more ideal frequency response with a shape factor of 4.43 instead of 5.7 for a 15^{th} order synchronously tuned analogue filter approximation. Also their phase response can be perfectly linear.

Digital filters are very easily implemented. The filter can be changed by using a different filter kernel or recursion coefficients. While normally Gaussian filters would be used because of their speed-selectivity balance, highly selective *windowed-sinc* filters can be made optional when such a high precision is needed. The bandwidth of the RBW filter can also be very flexible. Many RBW choices can be easily provided or even calculated from the provided bandwidth “on-the-fly”.

The FFT calculates the discrete Fourier transform of the digital input samples. The Fourier transform is defined for infinite time. A signal can be measured for only a limited amount of time, resulting in a finite number of samples, which can be seen as a rectangular window on the original signal. But also, a finite discrete number of results are calculated. As aperiodic signals have an infinite number of frequency components, while only a finite number are calculated, the input signal is assumed to be periodic by the DFT. The window represents one period. Discontinuities because of concatenating the window, causes spectral leakage; the smearing of the signal frequency to other frequencies. This can be seen as convolving the input spectrum with the spectrum of the window. The FFT can also be seen as performing many filters in parallel, with the filter shape determined by the window in the time domain. The filter shape causes surrounding frequencies to contribute to the result at one frequency (the inverse view of contributing itself to surrounding frequencies, being spectral leakage).

With no discontinuity giving energy at all frequencies, the convolution with the window transform does not smear out the frequencies in the signal and clear frequency components result from the DFT. Many different window functions can be devised that fall to zero at the edges. While no selectivity versus transient response exists as for analogue filters, because the window can be applied instantly, a higher side lobe attenuation comes at the cost of a wider main lobe and therefore less resolution. For the spectrum analyser a 70dB SFDR range is specified, which means the attenuation must be at least that much. The Chebyshev window is preferred because it has the highest resolution with a 74dB side lobe attenuation.

7.5.2 Conclusion

A digital spectrum analyser design is presented which can both perform frequency selection by stepping the input range through a digital RBW filter as by performing an FFT.

Both operating principles can use the same front-end if a quadrature architecture is used. The ADC must be placed after the frequency conversion. The frequency selection is performed in the digital domain. The ADC must have 14 effective bits and a 2MHz sampling rate. The FFT can thus analyse a 1MHz band. If a larger bandwidth is wanted, the FFT must be performed multiple times with a stepping LO.

The RBW operating principle provides a higher dynamic range of 70dB, because the image problem is alleviated by choosing the zero-IF architecture. However, the RBW solution is slow, especially at small bandwidths. For each frequency it must wait for the RBW filter to settle. For smaller bandwidths the transient response becomes slower. If the DC offset is removed by high-pass filtering, the design becomes even slower. Therefore a Gaussian filter is normally used to limit the settling time. The selectivity of digital RBW filter can be much better however. With enough time a nearly ideal rectangular frequency response can be implemented.

The FFT operating principle is much faster. The FFT can be calculated as soon as enough samples are measured. The FFT also can achieve a much higher resolution, only being limited by the amount of acquired samples and therefore the time to wait and the available processing power. The FFT additionally provides the phase, which can be easily related to previous LO steps by slightly overlapping the frequency bands. For the FFT, the front-end is used as a low-IF architecture. Therefore, the FFT solution has image

problems. For the FFT principle a quadrature architecture is necessary for image rejection. For the RBW principle the quadrature architecture provides the advantage of instant amplitude and phase determination. The image rejection of a quadrature architecture is limited by the matching and is typically 40~60dB. This is not enough for the specified spurious free dynamic range of 70dB.

For the digital spectrum analyser, the RBW solution can be used if a high dynamic range or very high selectivity is needed or the FFT if high speed and/or resolution is needed. The range of possible RBW filters is also much more flexible, both in terms of bandwidth as well as frequency response.

Chapter
8

8 Simulation

As a proof of concept, the designed digital front-end is simulated using “MathWorks MATLAB & Simulink Release 14” [51] A second objective is to evaluate the use of a coarse-grained reconfigurable processor, the Montium [50] for the processing after digitalisation. Note that the front-end up to the ADC is the same as for the analogue front-end.

First the ideal front-end is simulated in which we will focus on the FFT-based approach. The correct operation and claimed specifications will be verified. Next, the limitations of running the FFT on the fixed-point reconfigurable Montium processor versus the “ideal” floating-point MATLAB version is analysed and simulated.

8.1 Library

In order to provide a framework for evaluating different implementations of the front-end blocks, a library was set up. Different front-end designs can be evaluated by selecting appropriate component implementations from the main categories. But also, when updating a certain library block, all front-end designs using this block are automatically updated. For example, the ideal implementation uses many of the same blocks as the implementation comparing the ideal result with the results using the Montium. The same applies, for example, when adding noise or non-linearities to certain components.

The current library is shown in Figure 134, providing an easily extendable and updateable starting point for further simulations. Note that not all library blocks are used or implemented at the moment.

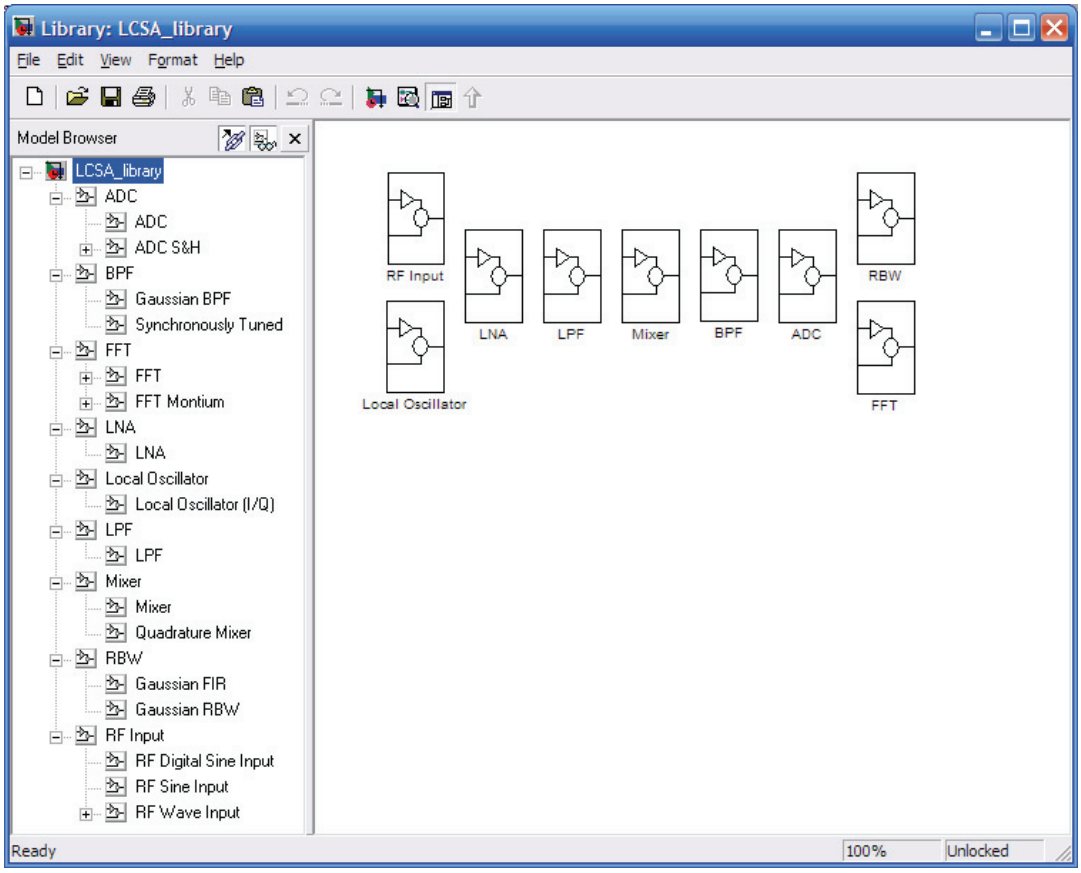


Figure 134. Low-cost Spectrum Analyser library for Simulink.

8.2 Ideal font-end

Two operating principles were identified, which use an identical front-end up to the frequency selection.

For the frequency selection, the RBW-based operating principle uses smaller, single frequency, local oscillator steps with RBW filters. After the filter the signal's amplitude and phase can be determined after half a period. A quadrature front-end can be used to instantly determine amplitude and phase. However, tracking the phase relation between measurements can be difficult (see section 4.2).

The FFT-based operating principle uses larger steps of a frequency band on which an FFT is performed. The FFT determines the amplitude and phase shift of a number of frequency bins in the band. By slightly overlapping the frequency band, the phase shifts between LO steps can be tracked more easily than the RBW case.

The proposed digital analyser uses both operating principles. This enables the selection of the RBW-based principle in case a high dynamic range or a high selectivity is needed, or the FFT-based principle if a high speed, high resolution or the phase is needed. This model is discussed in section 8.2.1, while the discussion of different simulation results is found in section 8.2.2.

8.2.1 Front-end

The front-end sub-parts such as presented in chapter 7 are implemented as Simulink blocks. This is shown in Figure 135. This figure also shows sample time colours, to indicate the sample rate at which the blocks operate. Everything up to the ADC in black is continuous. These blocks use the *ode45* continuous signal solver with automatic step size. Yellow blocks are hybrid and have multiple or varying sampling rates. The ADC samples the continuous signal. The ideal output in red is at the highest sampling rate to simulate an ideal signal compared to the other output in green. The digital RBW filter does not change the sample rate, but the FFT buffers the samples into frames before processing the frame at once. In this case, in frames of 512 samples at the lowest sampling rate in blue.

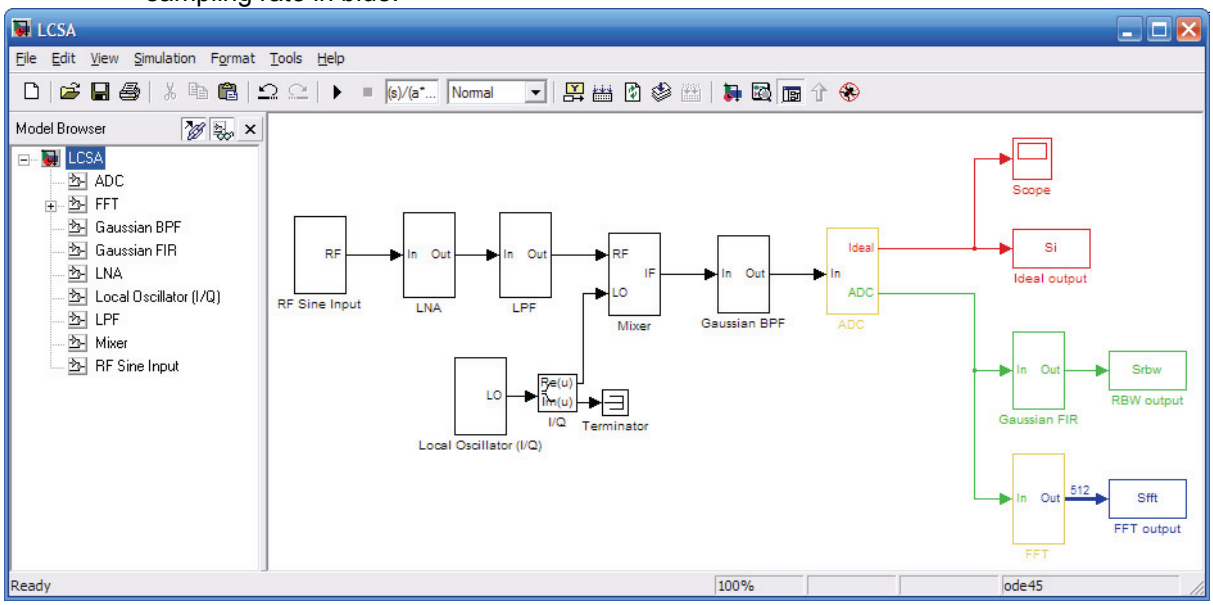


Figure 135. Ideal digital spectrum analyser front-end.

The RF, LO and IF frequencies are kept relatively close together, in order to keep the simulation time acceptable. The IF is from DC up to 1MHz with a 2MHz sample rate for the ADC as discussed in section 7.2. The RF input range is positioned slightly above this from about 5 to 13MHz. The LO is therefore also from about 5 to 13MHz. This gives a zero-IF architecture with a usable range of DC up to the RBW corner-frequency for the RBW output. For the FFT the usable range is 0.25MHz to 0.75MHz, for DC removal and alias suppression purposes (section 7.2).

The RF sine input block consists of two added sine waves. One sine can represent the signal of interest, while the other can for example represent an image or distortion frequency. For a more real-world simulation, the RF sine input block can be replaced by something like a sound wave input.

The LNA and LPF are not implemented and simply pass the signal. Therefore the input signal is not conditioned and care must be taken that the signal amplitude and frequency range is known and gives the expected result. Note that out-of-bound signals can be used on purpose to verify expected linearity and noise limitations for example. Because the analogue front-end is still implemented as ideal, no limitations except for the finite floating-point precision exist.

The local oscillator consists of two ideal sine waves with the same frequency and amplitude but a 90° phase shift; thus providing quadrature signals. The I and Q signals are combined as a complex signal. Their amplitudes are set to 2V, so that the resulting sum and difference frequencies both have the same amplitude as the input signal.

The mixer is implemented as a simple ideal multiplier. Therefore only one of the quadrature LO signals is used as input. Because, this is not a quadrature architecture, we expect to see a single sine input twice for the low-IF case; once as a positive frequency and once as a negative frequency.

8.2.1.1 Gaussian BPF

The Gaussian anti-alias band-pass filter is also a DC removal filter and the largest RBW filter. DC removal is not implemented, because in the ideal case there is infinite isolation between the RF and LO signal paths and no 1/f noise (see section 4.6.2). Also noise and non-linearities are not simulated, which means no DC offset is generated. Hence, the filter can become a low-pass filter. The Gaussian response is approximated with an 8^{th} order analogue Bessel filter, because it is readily available in Simulink. A synchronously tuned implementation would be better, but this is left for future implementation.

Because the Bessel filter still shows some ringing in the transient response, the time to wait for the signal to settle is twice as long as strictly necessary with a Gaussian filter (section 5.5.2). The sweep time will therefore also be slightly longer.

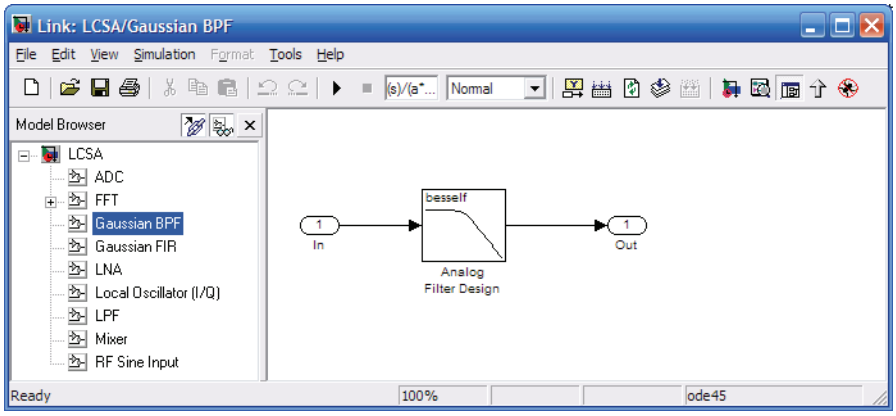


Figure 136. Gaussian filter approximated with an 8th order Bessel filter.

In order to keep the amplitude inaccuracy under ±1dB in the pass-band, the -3dB corner frequency must be at 1.74 times the RBW/anti-alias bandwidth, which is a corner frequency of 1.74MHz for a 1MHz RBW corner frequency (section 5.6).

8.2.1.2 ADC

The ADC is implemented by first sampling the signal to make it discrete and then quantise it to make it digital. The sample and hold (S&H) block is responsible for converting a continuous signal into a discrete signal. This block is available as part of the Signal Processing Blockset. Sampling occurs at the rising and/or falling edge of the trigger input and the value is held until the next trigger. This enables a varying sampling rate as well as starting and stopping the sampling. The signal type of the trigger signal determines the signal type of the S&H output, thus care must be taken the trigger pulse is not continuous as the S&H output will also remain continuous.. Therefore the trigger is a discrete signal of the desired sample rate, changing between zero and one at each sample time, and the S&H react to rising and falling edges.

The sampling can also be implemented by a zero-order-hold (ZOH) block. This block takes the analogue signal amplitude at the sample time and holds it for the sample period until the next sample time. Therefore only a fixed sampling rate can be used. A hold

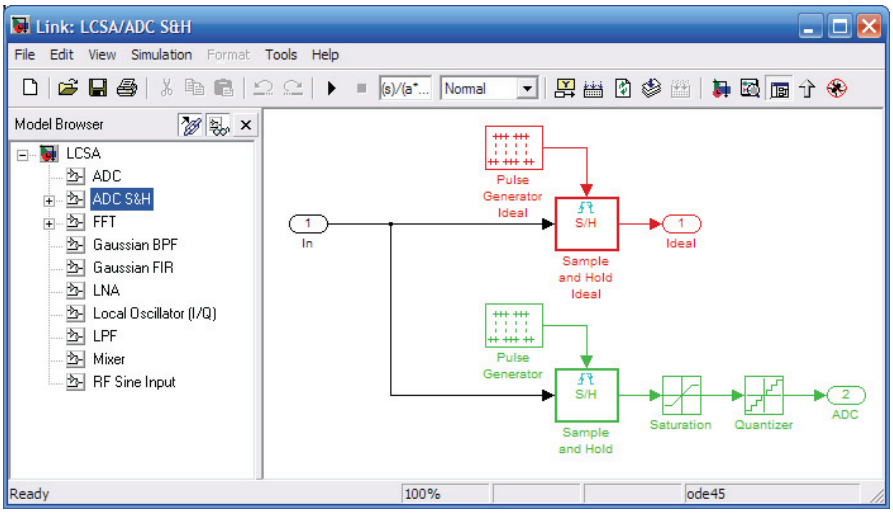


Figure 137. ADC implementation with a sample & hold.

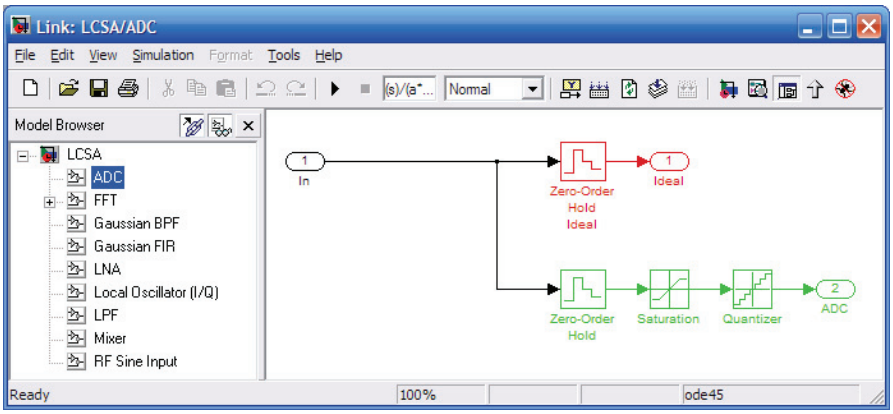


Figure 138. ADC implementation with ideal output at a 50 times higher sample rate.

signal from the ZOH is represented by one sample value and the signal therefore becomes discrete. This can be seen in Figure 138 from the sample time colours as red and green indicate discrete signals.

The ZOH block is normally used for making discrete signals analogue by holding the digital value until the next discrete value is provided. In this case the following block must be continuous. Although the S&H is principally more correct, when using the same sampling time, the result is exactly the same (confirmed numerically).

The ideal output of the ADC has a 50 times higher sample rate then the normal output. The output is discrete so further sample based processing such as an FFT can be performed, but the output is not saturated or quantised. This output is provided because the signal after the anti-alias filter and sampling is not always easily recognisable as a waveform. For example, of frequencies close to halve the sample rate, only two or three samples of a complete period are available. On the scope this does not resemble a sinusoid anymore, but with 50 times more samples it does. Furthermore, for the frequency spectrum, the FFT can be taken up to a 50 times higher frequency, so no aliases should be visible in the range of interest.

Of course, this also increases the simulation time. Therefore when the ideal signal output is not of interest, it is possible to lower the over-sampling down to the same sampling rate as the ADC output.

Note that in MATLAB, although the time-based signal is quantised, the FFT returns floating point results with a much higher precesision, Thus, the results are not quantised.

The normal sample rate of the ADC is 2MHz. The maximum input signal of -15dBm corresponds to a peak voltage of 0.06V. However, for the simulation a maximum input signal of -10dBm is used. The ADC is therefore saturated outside -0.1V~0.1V.

As the ADC is ideal, the signal is quantised with 14 effective bits and a LSB of $0.1/2^{14}$ = 6.10µV. This is an 84dB range, with the quantisation noise at -95dB.

8.2.1.3 FFT

The first samples after an LO step are distorted by the transient of the anti-alias filter. A Gaussian filter has a transient taking about 2 periods of the -3dB corner-frequency (section 5.5.2). The corner frequency is at 1.74MHz. With a sample rate of 2MHz, this means the first 3 samples must be discarded. The Bessel filter needs 4 periods according to section 5.5.2 and the simulation results, which is 5 samples. This seems a small increase, but for example with a low resolution 8-point FFT, this means a 18% increase in

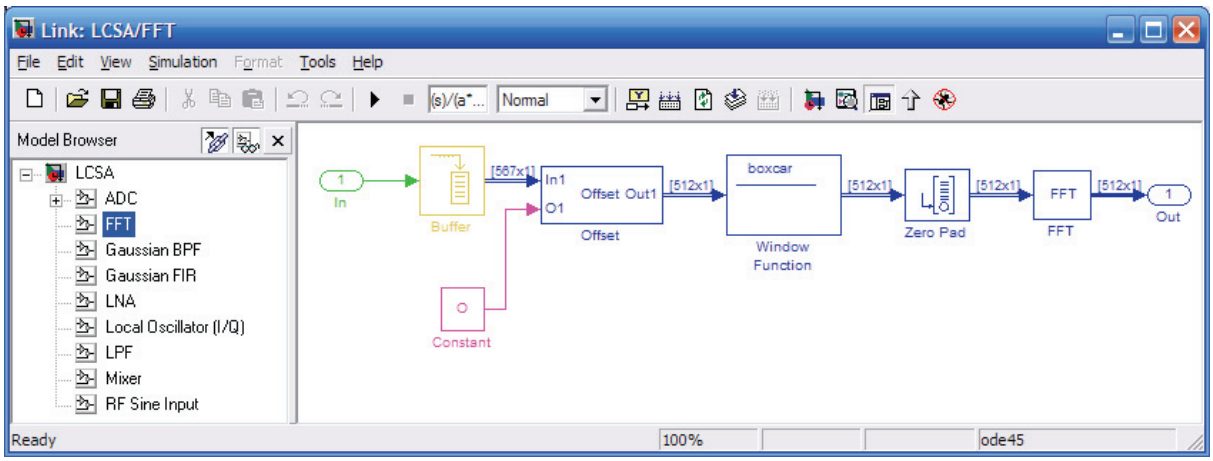


Figure 139. Fast Fourier transform with transient removal, window and padding.

sweep time (without LO settling time), while the Bessel filter provides almost no difference in selectivity compared to a Gaussian filter.

The FFT implementation first buffers the samples to discard (*O*) plus the number of samples needed for the FFT (*N*) into a frame. Note that only one frame needs to be converted (to achieve the fastest sweep time), after which the LO steps to the next frequency band. The simulation therefore only has to run the time needed to fill the buffer. The offset block removes the first distorted samples. Then a window is applied to the frame. Results are provided with a rectangular window (no window) and with a Chebyshev window with indicated side-lobe attenuation. The Chebyshev window gives the highest selectivity with a specified side-lobe attenuation (section 7.4.2.2).

The next block pads the samples with zeros to the next power of two. This is needed because the FFT only accepts frames with a power-of-two number of samples. Note that zero padding does not increase the resolution, although it may seem like that because of the higher number of samples used; the magnitudes of the unpadded signal result spread out over more FFT output values in the padded case, but this does not increase the ability to differentiate two closely spaced signals [6]. However, a power of two FFT does execute faster in most cases. With the spectrum analyser, we can measure enough samples for a power of two and the zero-padding is not really used.

Finally the FFT is computed of the *N* samples in the frame resulting in *N* complex values. These complex values are outputted to a MATLAB variable and converted to amplitude (magnitude) and phase. Next the amplitudes are converted to power in dBm. The result is then plotted or added to the results of previous LO steps.

Note that interleaving is not applied to the ideal design. This is because, as discussed above, no DC offsets are generated in the ideal case. The folding effect will be evaluated.

8.2.2 Results

In order to evaluate the proper operation of the front-end, we will simulate the implementation with a single sinusoid and evaluate the result at different frequencies without a window and with a Chebyshev window. Next we will enlarge the measured frequency band by stepping the local oscillator and combining measurements. Finally, the results of using two sinusoids and a real-life waveform containing many frequencies are evaluated.

8.2.2.1 Window

For the first simulation, the RF input is a single -10dBm, 5.5MHz sine wave. The LO is set at 5MHz. The signal is therefore expected at 0.5MHz in the IF band. Only the FFT results are provided at the IF band from DC to 2MHz. This range therefore includes the negative frequencies for the normal 2MHz sampled ADC output. A 512-point FFT is used for a high resolution The settling wait time is set to 8 filter periods, so that no transients affect the results. The result for the ideal (50 times higher sampling rate) ADC output is shown in Figure 140 and the result for a 2MHz sampling rate is shown in Figure 141.

For a 0.5MHz sinusoid sampled at 2MHz, each period is sampled 4 times. The 512 sample FFT therefore contains exactly 128 periods (and the ideal output 50 times more).

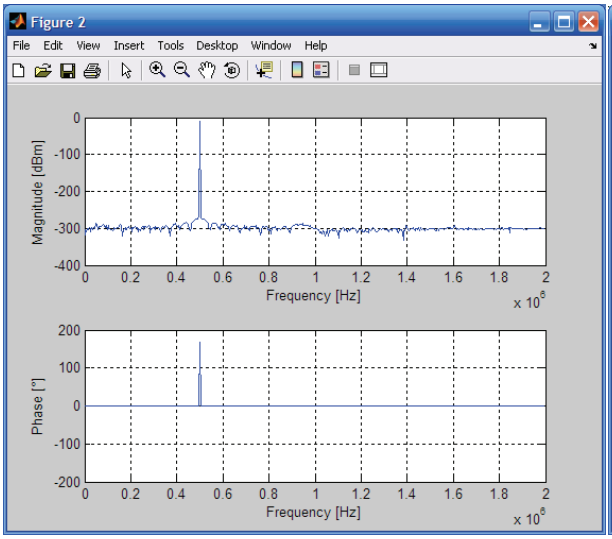


Figure 140. Single sine input and no spectral leakage (ideal result).

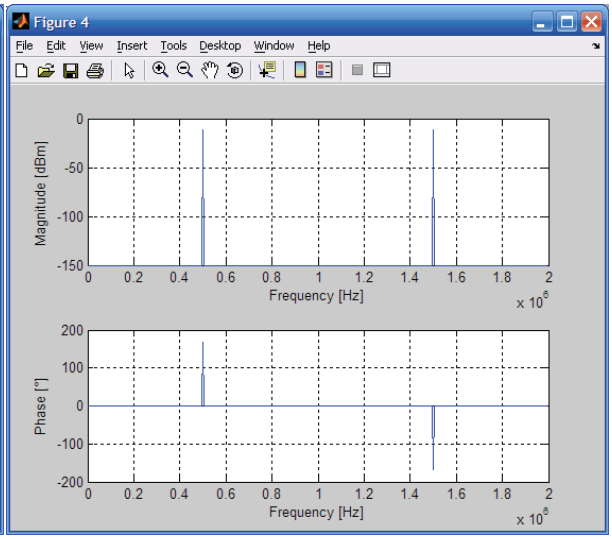


Figure 141. Single sine input and no spectral leakage (normal result).

Thus no spectral leakage occurs and the result is a sharp peak at 0.5MHz (see section 7.4.2.1). The rest of the results are limited by the floating point precision, which is at about -300dBm as can be seen in the figure of the ideal result. The normal ADC output is quantised with the smallest signal level at about -84dB or -94dBm. The quantisation noise is about -95dB or -105dBm. Therefore, the results other than the signal peak become zero. Zero is minus infinity on a log scale, which can not be displayed. In this case the result is set to -150dBm.

Another problem when the results hit the floating point limit, is that the phase results are meaningless. This is because the phase results from the arctan of the imaginary results of the FFT divided by the real result. The division of two round-of noise signal can give any result. The phase then results anywhere in between -180° and 180° and looks like noise. Therefore, if the magnitude is smaller than (or equal to) -250dBm in the ideal case or -150dBm in the normal case, the phase is made zero.

The peak value is a little less than -10dBm as expected because of the anti-alias filter. This is confirmed by simulating with a higher filter order and a more selective filter, which both give a result closer to -10dBm.

An interesting result appears if the sine wave fits into the window with an exact number of periods, but higher harmonics also fall into the FFT band. This is shown in Figure 142 and Figure 143.

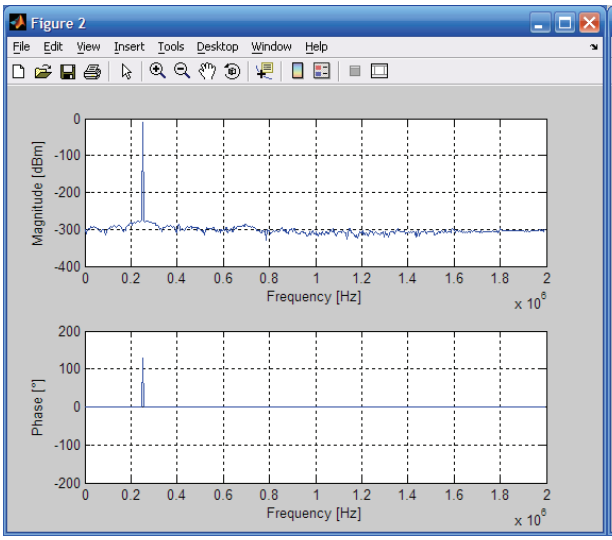


Figure 142. Lower single frequency result without quantiser.

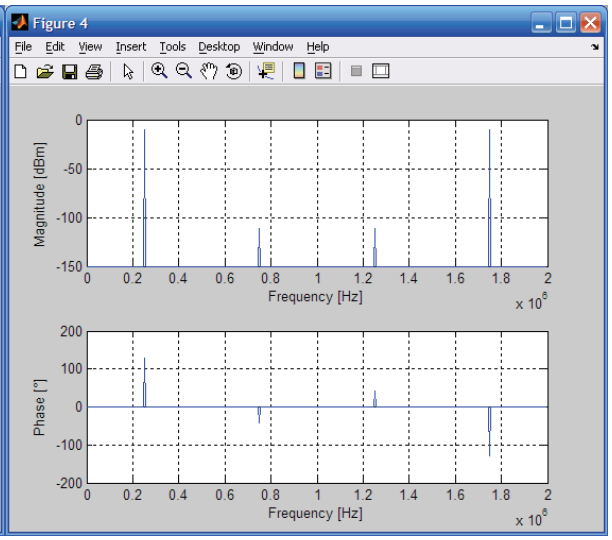


Figure 143. Result with quantiser giving rise to higher harmonics.

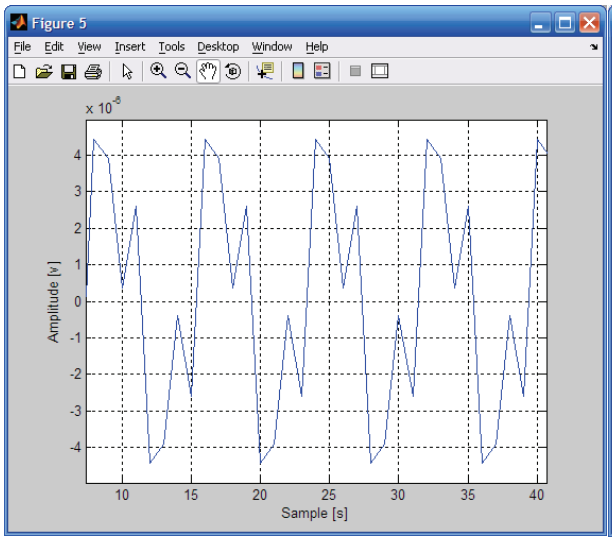


Figure 145. Quantisation error for a 0.25MHz sine with a 14-bit 2MHz ADC)

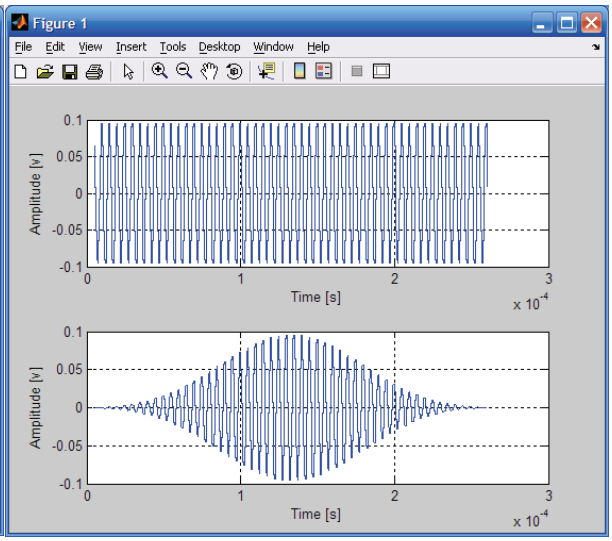


Figure 144. Time based IF signal with rectangular (top) and Chebyshev (bottom) window.

The ideal ADC output does not quantise the signal and shows a single sinusoid at the expected 0.25MHz. The normal ADC output is quantised. As explained in section 7.4.2.1, the quantisation noise is the same for each sampled period and is therefore periodic itself. Because the quantisation noise is not a pure sinusoid, this gives rise to higher harmonics. A few periods of the quantisation error are shown in Figure 144. Being of a sampled sine, the noise has a half-wave odd symmetry, which results in only odd higher harmonics at 0.75MHz, 1.25MHz etc.

The sine waves above were selected to fit exactly into the window. Therefore duplicated concatenated windows do not create a discontinuity. As explained in section 7.4.2.1, it is more likely that the sinusoid will not fit exactly in the window (at least not all of them). This results in spectral leakage.

Figure 146 and Figure 147 show the simulation of a single sine wave with an 5.434MHz frequency. Now we have side lobes (spectral leakage) which only go down to about -70dBm. Note that in the ideal case, the side lobe attenuation is a bit higher because the energy of the discontinuity spreads out over a larger frequency range. The dynamic range is limited by the spectral leakage to about 60dB, clearly not good enough. We would like the side lobes to remain under -94dBm (section 7.4.2.2).

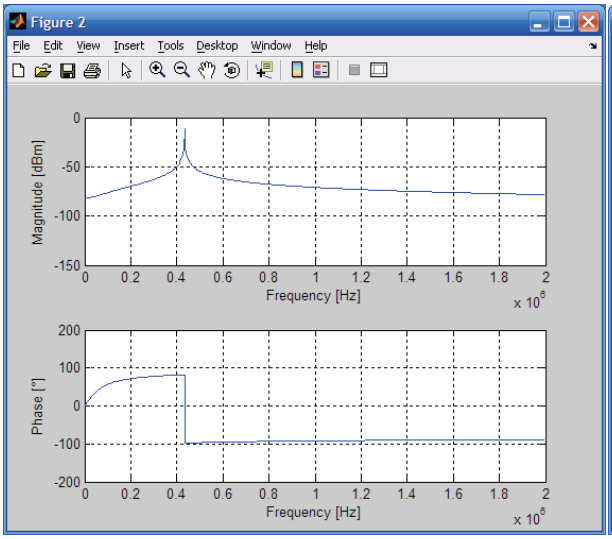


Figure 146. Spectral leakage with a 100MHz sampling rate.

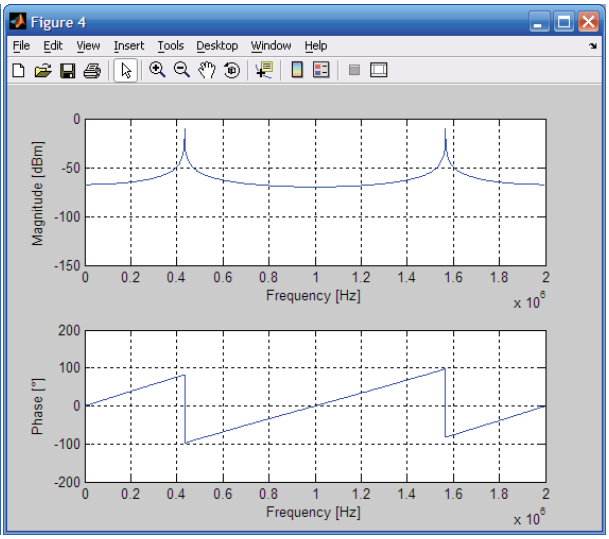


Figure 147. Spectral leakage with a 2MHz sampling rate.

To achieve side lobes below -94dBm, a Chebyshev window with 84dB attenuation is applied. The result is shown in Figure 148. Note that all the windows lower the peak level, because the energy is spread out over more frequency bins. The peak is at -18dBm instead of -10dBm. This must therefore be corrected for. How much correction is needed

is not further analysed. The attenuation is 84dB down from the (lower) peak, so this is no problem. Unfortunately the phase is also distorted; especially near the frequency peaks, where it is not a clear 180° phase shift as it was without a window. This is the case for most windows (less problematic with lower side lobes attenuation). Thus when phase is important it is better to apply no window.

For an evaluation of the peak width, the peaks are zoomed-in in Figure 149. It is clear that the rectangular window has a much sharper peak and much less attenuation. As expected, the peak bandwidth is two to three times wider with the Chebyshev window. The complete width is about 3 frequency bins for the rectangular window and about 8 frequency bins for the Chebyshev window.

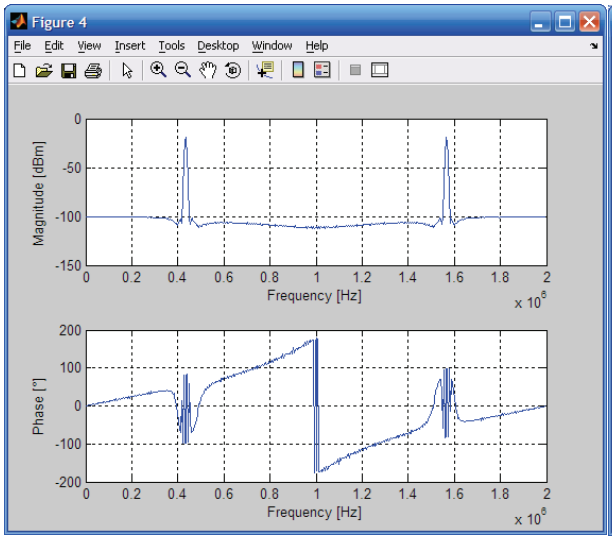


Figure 148. Single sine with Chebyshev window.

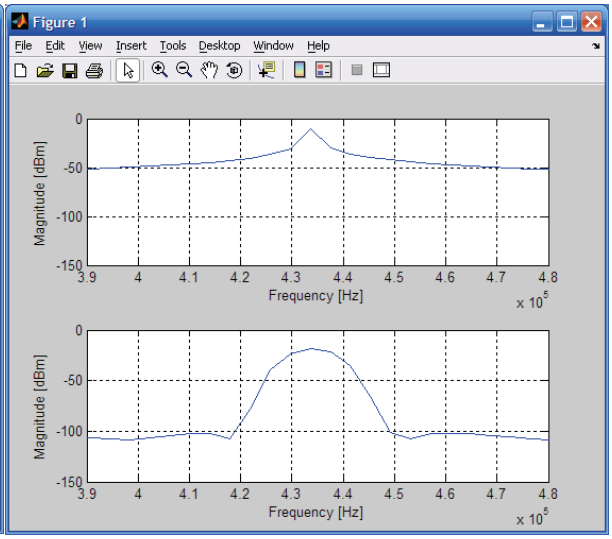


Figure 149. Peaks with rectangular and Chebyshev window.

Finally, the settling wait time was set back to the intended 4 filter periods. The result is almost the same, except for a slightly different response because the samples are shifted. Using no offset and therefore no settling wait time raises the side lobes about 10dB in case of no window. For the Chebyshev window, there actually is no difference at all, because the window function scales the first samples to zero, thus diminishing the transient distortion. Even with a Cauer or Chebyshev filter as anti-alias filter, the frequency response with a Chebyshev window is largely unaffected. Without a window the side lobes are raised about 20dB.

These results can be explained because the transient distortion spreads out over the whole frequency range in case of a rectangular window and the transient distortion is for the large part removed by the window in case of a Chebyshev window.

Of course this is only when a large number of samples is used, i.e. a high resolution FFT. For example, with a 16-point FFT, the side lobes are raised significantly (30dB) by the transient distortion, also with a Chebyshev window. With anything larger than 4 filter periods of settling wait time, this distortion is removed. Therefore, 4 periods is sufficient and will be used for the remaining simulations.

8.2.2.2 Stepping LO

Until now, we only looked at one RBW band at the IF over which the FFT was taken. For the spectrum analyser design to be useful, it must also operate over a larger frequency band by combining multiple FFTs with a stepping LO. This is evaluated in this section.

A single sinusoid is generated at 8.34MHz. The LO is stepped from 5MHz to 13MHz with 1MHz increments. Because these are 8 steps, a smaller 128-point FFT is used to achieve a 1024-point FFT, double the display resolution as the figures above. The result with a rectangular window is shown in Figure 150 and with a Chebyshev window in Figure 151. Obviously many peaks appear in the spectrum instead of one at 8.34MHz. Almost all of these are aliased frequencies. The anti-alias filter has a high corner-frequency to limit the frequency response in the pass-band to ±1dB, and the filter roll-of is limited by the speed-selectivity trade-off. Therefore higher frequencies are only partly attenuated and alias into the IF band after digitalisation (see section 7.3.2). For example, if the LO is set

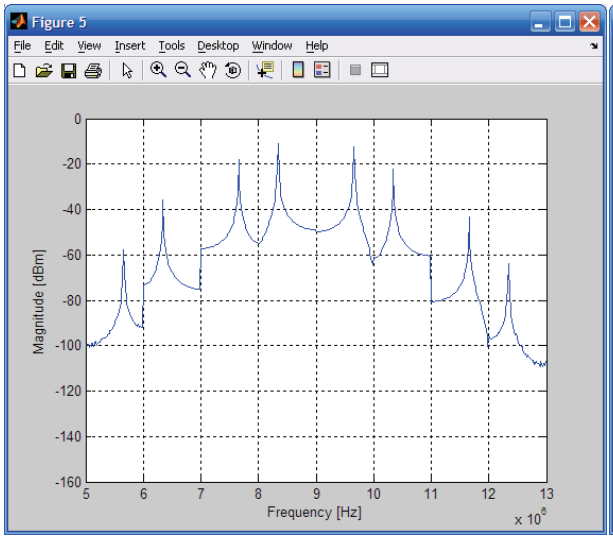


Figure 150. Stepping LO with a single sine and rectangular window.

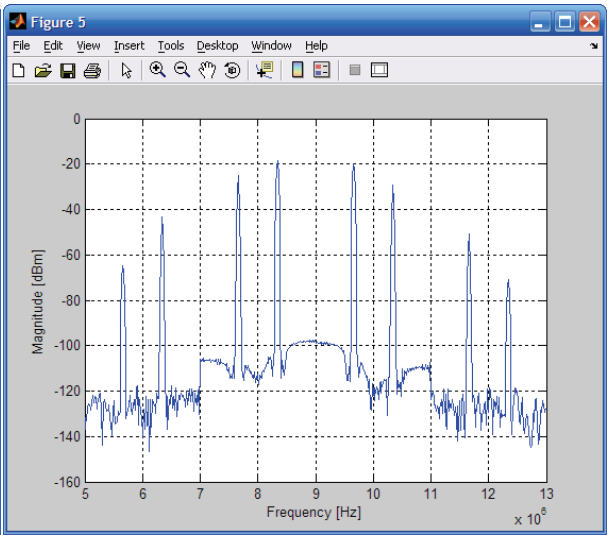


Figure 151. Stepping LO with a single sine and Chebyshev window.

to 5MHz, the sinusoid is down-converted to 3.34MHz. At that point, the attenuation is about 45dB as can be seen from the peak at 5.66MHz (as higher frequencies are folded into the ADC bandwidth, higher frequency ranges are inverted every other one). This results in the anti-alias filter response becoming visible by imaging a line over the peaks.

Possible solutions are a sharper anti-alias filter, although the filter can never be so sharp as to prevent any aliasing. Another option is over-sampling as explained in section 7.3.2. A combination of both is also an option and possibly an optimum can be found. We will analyse the over-sampling option. Figure 152 show a two times over-sampling at 4MHz and Figure 153 three times at 6MHz. For a 4MHz ADC (two times over-sampling) with the LO at 5MHz, the range 5~6MHz is wanted and the range from 6~7MHz also falls under the Nyquist frequency and those FFT results can be discarded. The range 7~8MHz folds back to 7~6MHz and 8~9MHz to 6~5MHz. Thus frequencies 3MHz away from the LO are the first to interfere. That's why the peak still shows in Figure 152. With four times over-sampling, frequencies 7MHz away from the LO are the first to fold into the range of interest. At that point the frequencies should be attenuated enough (section 7.3.2) not to distort the signal.

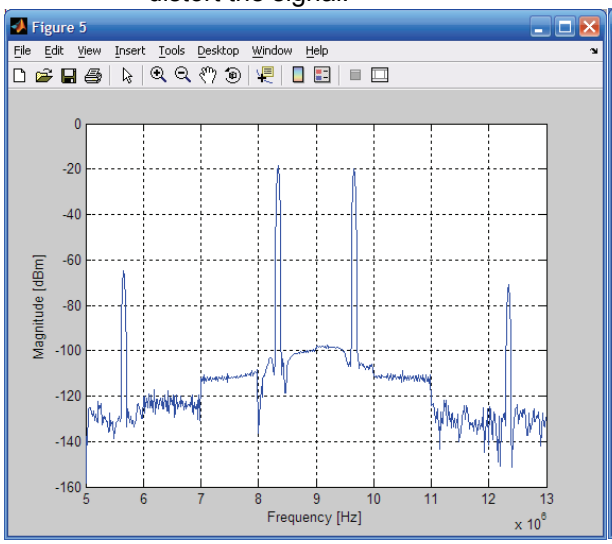


Figure 152. Stepping LO with two times over-sampling.

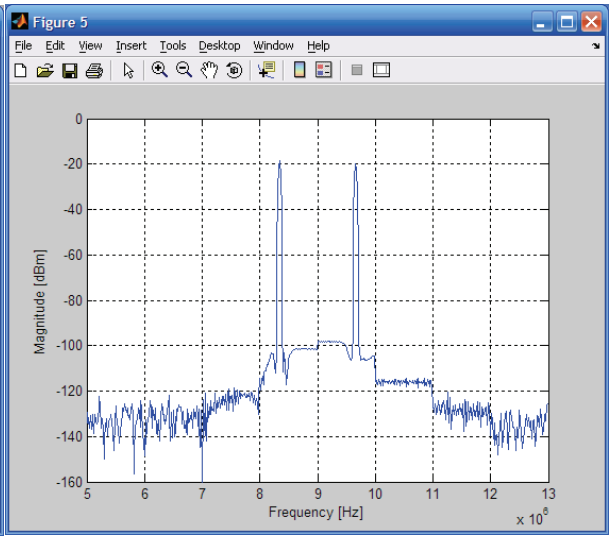


Figure 153. Stepping LO with four times over-sampling.

However, Figure 153 still shows two frequencies. This is the image problem of a low-IF architecture (chapter 4). If the LO is set to 9MHz. The wanted range is 9~10MHz, but the image band is 9~8MHz. Thus the 8.34MHz sine results as an image at 9.66MHz.

The RBW-based principle uses a zero-IF architecture and therefore it has no image problem. We can also simulate the RBW case by using an FFT and the resolution so that we get one result at the IF from DC to 1MHz. This result is the amplitude and phase of a

2MHz band at RF (a 2MHz band at RF down-converted to DC has a 1MHz bandwidth, see chapter 4). By scaling the anti-alias/RBW filter bandwidth and the LO steps, the current implementation can be used as an analogue RBW-based spectrum analyser with the measurement performed by the FFT.

To show the zero-IF has no image problem, we compare the low-IF FFT-operating principle with the zero-IF RBW operating principle. Four times over-sampling is used to remove alias distortions. The 5~13MHz range is measured with a display resolution of 256 points. For the FFT principle this means 8 LO steps of 1MHz with 32 FFT points in the band 0~1MHz for each step (thus a 256-point FFT for the 4 times over-sampled 8MHz sampling rate). Note that the higher frequencies, including the negative frequencies are discarded each step. The result is shown in Figure 154. For the RBW principle, the RBW/anti-alias filter corner frequency is set 32 times lower at 54.375kHz, and the LO makes 256 steps of 31.25kHz. An 8-point FFT is used after the settling wait time, giving 1 point in the 0~1MHz band. Again the higher frequency results are discarded. This is shown in Figure 155.

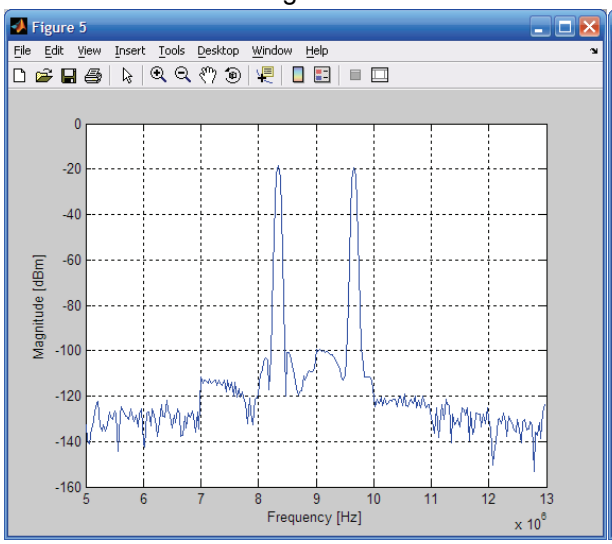


Figure 154. 32-point FFT with 8 LO steps.

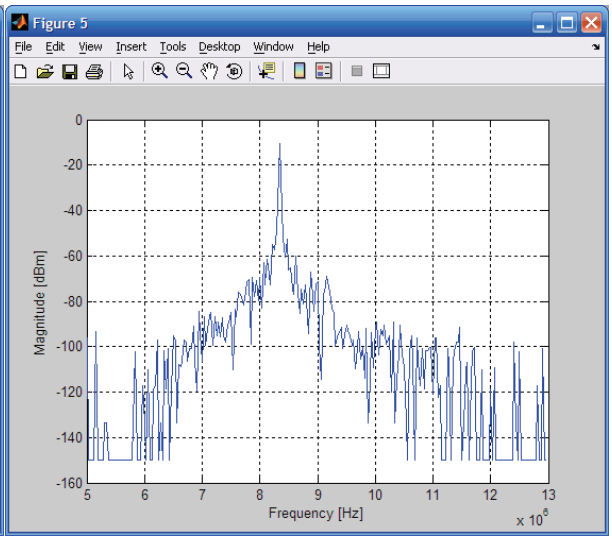


Figure 155. 1-point FFT with 256 steps.

Figure 154 clearly shows the 8.34MHz sine being an image at 9.66MHz. It also shows relatively wide peaks with high attenuation. Figure 155 shows only one peak at the expected 8.34MHz. Thus the zero-IF indeed has no image problem. The peak is much sharper. However, the side bands show less attenuation starting at about -70dBm and falling to about -100dBm. This is because with the frequency of the sinusoid close to the RBW corner-frequency, the attenuation by the RBW filter is small. The attenuation still starts at about 50dB because of the over-sampled ADC: although the sine is close, the frequencies are at FFT results that are discarded. Further away the attenuation by the RBW filter is enough and the results hit the quantisation noise level at about -100dBm.

8.2.2.3 Real-life input signal and use

Although a single sinusoid clearly shows what is happening with the signal, it is not very good at representing real-life use of a spectrum analyser. Therefore we first simulate the design with two sinusoids in some different situations, followed by the simulation of the spectrum analysis of an audio signal.

First a sinusoid of 8.34Mhz and of 5.49MHz is analysed using 8 LO steps with 32 FFT point in the 0~1MHz band, giving a 256-point resolution. Four filter periods settling time and four time over-sampling is used. The two sinusoids give four peaks as shown in Figure 156, because of the low-IF architecture. Clearly with multiple frequencies, the result quickly becomes indecipherable and a quadrature architecture is needed.

Next, the two sinusoids are located in the same ADC bandwidth by giving the second a frequency of 8.49MHz. This is shown in Figure 157. Because of the lower resolution, the peaks are not clearly separated. In this case, you would choose a smaller frequency band with about the same display resolution and sweep time, but with a higher frequency

resolution. Two LO steps with 128 FFT points per step show two separated signals as shown in Figure 158. At this point one might choose to amplify the input signal, because a third very small signal is expected to be at about 8.6MHz. However, now the larger signal at 8.34MHz becomes larger than -10dBm resulting in an overflow of the ADC input. This severely distorts the signals over a large range as shown in Figure 159. Therefore an overflow is best detected by the spectrum analyser and reported back to the user.

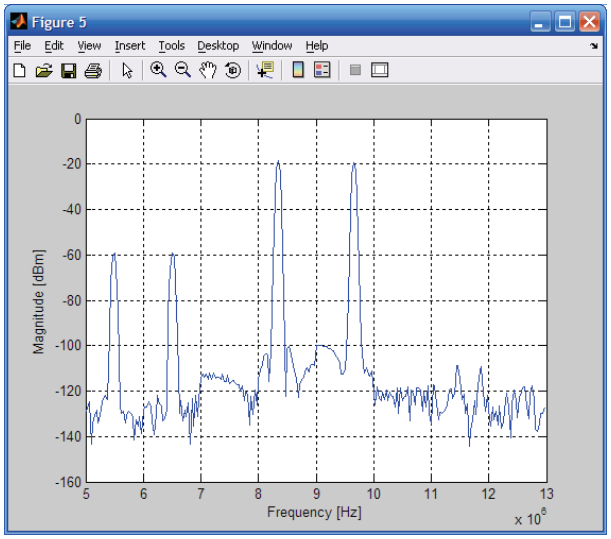


Figure 156. Two sinusoid inputs with different amplitudes.

The figures that contain multiple LO steps clearly show that each sinusoid ends up at its specified frequency and as an image frequency for the next LO step. It is also clear from section 4.3, that two RF input signals end up at the IF, with the image at a distance of two times the IF.

An 8.34MHz signal is at an IF of 0.34MHz. The image is thus 0.68MHz away at 7.66MHz. Indeed Figure 160 shows that our -10dBm signal at 8.34MHz, which after the window should end up as -20dBm, is distorted by a -13dBm signal at 7.66MHz. The resulting signal is 10dB smaller than it should be.

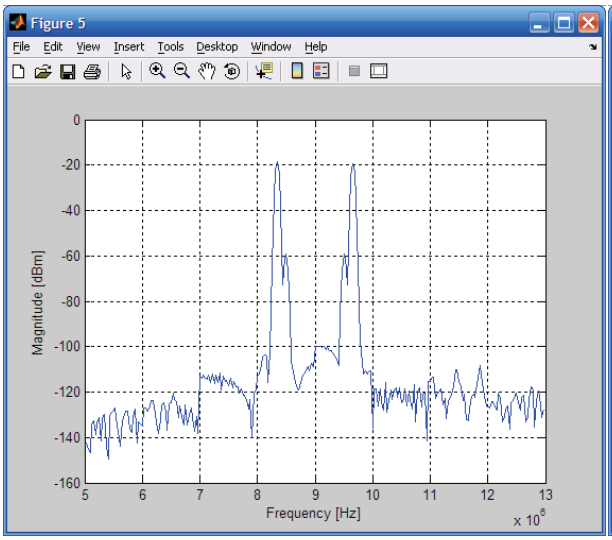


Figure 157. Two sinusoids close together with not enough resolution.

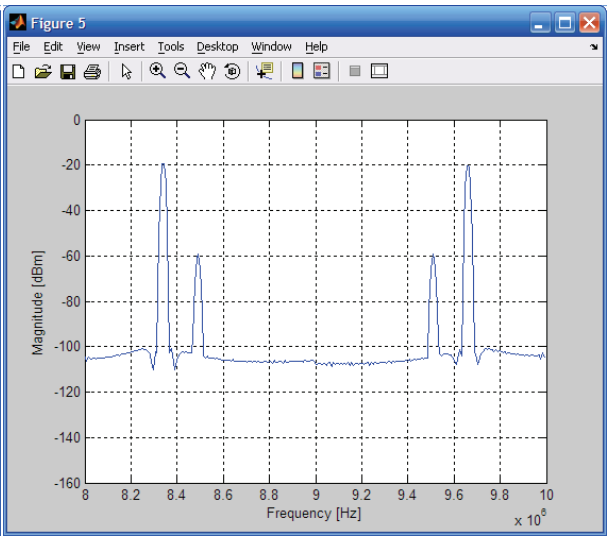


Figure 158. Measurement with a higher frequency resolution.

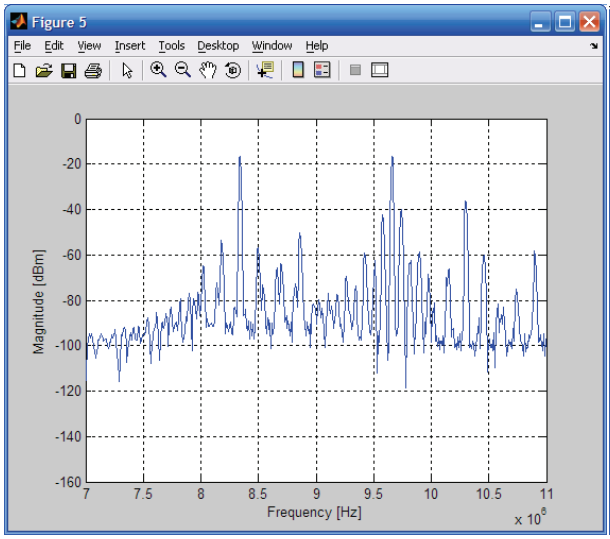


Figure 159. Distorted measurement because of an overflow.

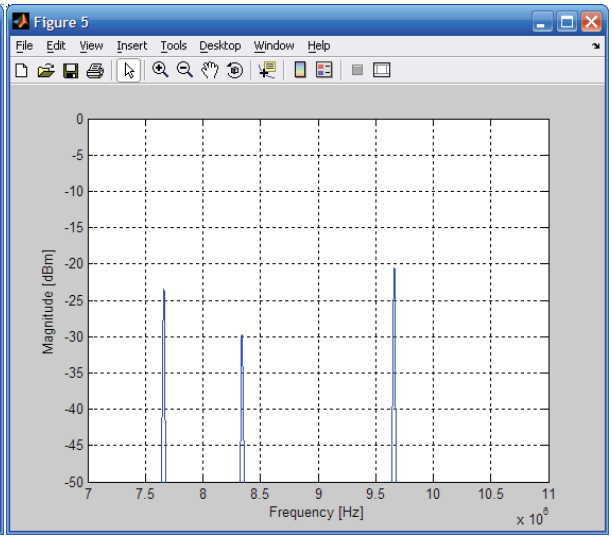


Figure 160. Wanted signal distorted by an image.

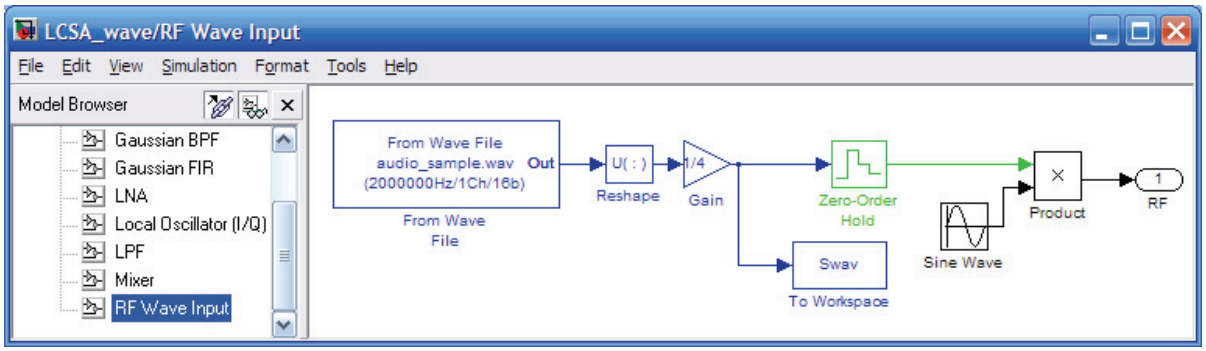


Figure 161. Wave input of 2MHz sample rate, scaled and upconverted with 8.0005MHz to RF.

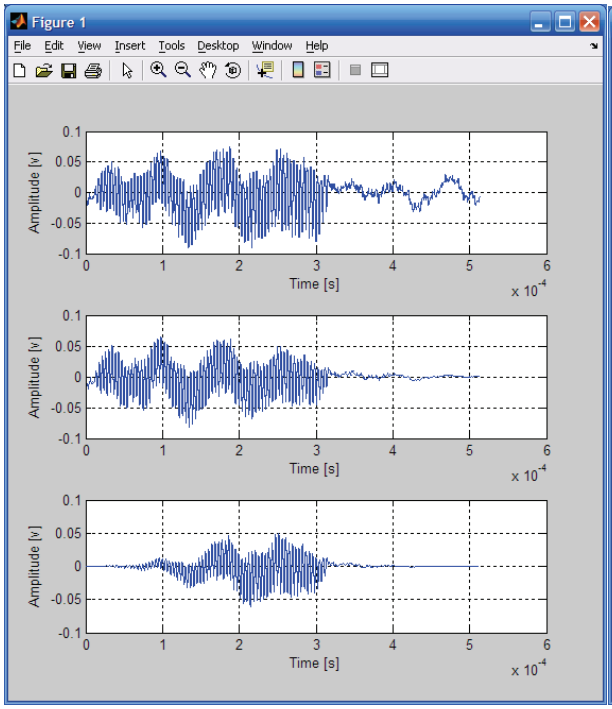


Figure 162. Original input signal (top), after ADC without window (middle) and after ADC with a Chebyshev window (bottom).

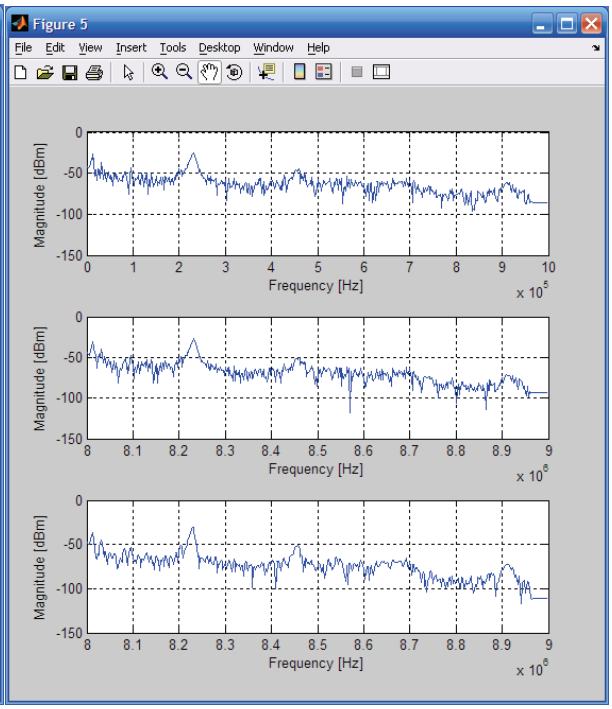


Figure 163. Spectra of original input signal (top), after ADC without window (middle) and after ADC with a Chebyshev window (bottom).

Of course two sinusoids still is not a real-life input signal most of the times. Therefore 4096 samples of an audio fragment are taken and used as having a 2MHz sample rate. The signal is attenuated so the maximum is less than -10dBm. Next the signal is up-converted with an 8.00005MHz sinusoid and becomes continuous. See Figure 161. The LO is set to 8MHz and a 1024-point FFT is taken.

Figure 162 show the original signal, together with the digitalised signal at IF without a window and with a Chebyshev window. Figure 163 shows the spectrum of the original signal, together with the spectrum at the output of the front-end without a window and with a Chebyshev window. The spectra are largely the same.

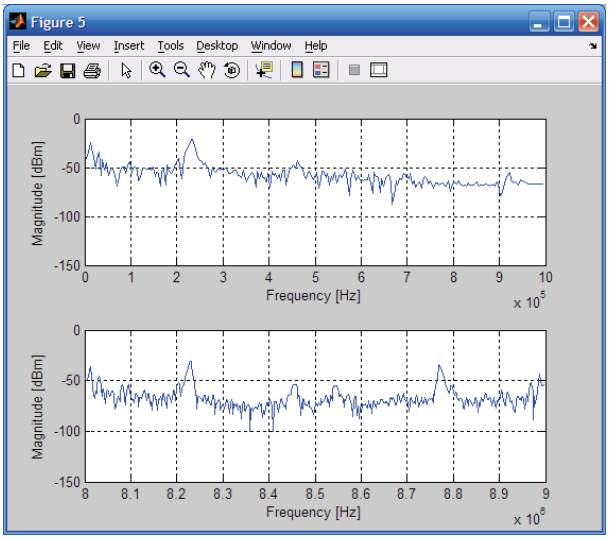


Figure 164. Spectrum of original input signal versus spectrum of the spectrum analyser with two LO steps.

However if the same range is analysed by using a stepping LO with two steps, the result becomes distorted again because of images as shown in Figure 164.

8.3 Montium

After the down converted input signal is digitised, it must be processed further. This can be done by different solutions. For example, the signal can taken of-chip for processing on a personal computer. Another option is implementing the processing in hardware.

If we assume we want a device that can be used as a spectrum analyser on its own, some form of hardware architecture is needed for processing. As the device must be low-cost an integrated solution is preferred. Integrated circuit hardware architectures can be divided into five groups with increasing energy efficiency, but decreasing flexibility [50].

The most flexible is a general purpose processor (GPP). A GPP is designed for general use and therefore offers the most flexibility, but has limited parallelism. Since operations are done sequentially, high clock speeds are needed to give good performance, resulting in low energy efficiency. GPPs are flexible in that they allow much control over what needs to be done. They implement one or more arithmetic and logic units (ALUs) that perform arithmetic or logic functions sequentially. Unfortunately this also causes a lot of control overhead, especially for computationally intensive (as in for the ALU) but repetitive tasks.

This applies to digital signal processing for example. A digital signal processor (DSP) limits its use to digital signal processing and can therefore provide higher performance and energy efficiency. It still provides much of the flexibility of a GPP, but also support for more specific signal processing tasks, such as a multiply-accumulate operation or a FFT butterfly operation (see below).

Fine-grained reconfigurable hardware is based on the principle that every function in the digital world can be implemented using a lookup table. Thus by providing a large number of small lookup tables and being able to connect them in a configurable way, one can implement any functionality by providing the contents of the lookup tables and the connections. Since the design of a fine-grained reconfigurable hardware device is very regular, it can be highly optimized for performance and energy efficiency. However, functionality is limited by the number of tables and connections and flexibility is limited because reconfiguration takes a relatively long time. Of course the overhead of connections and configuration increases its energy consumption. Because of the use of lookup tables, it is very efficient at bit-level (fine-grained) operations.

Course-grained reconfigurable hardware is designed for word level (course-grained) algorithms. These are the same algorithms as the DSP is intended for, but instead of running a program, the hardware is configured to perform a certain task. This implies signal paths are relatively stable helping energy efficiency. Its flexibility comes from the ability to reconfigure the hardware (change the interconnects) for a different algorithm in the hardware's application domain. Because the functional blocks are larger compared to fine-grained reconfigurable hardware, the overhead is less and this increases its power efficiency. At the same time the larger functional blocks lower its flexibility.

If designed properly, an application specific integrated circuit (ASIC) is the most energy efficient. Its energy efficiency is obtained because the hardware is designed specifically for certain functionality and is not changeable. Therefore its flexibility is limited to the design. Note that all the above architectures are implementations of an ASIC with different degrees of flexibility, which defines the degree of overhead and signal routing, which in turn defines its energy efficiency.

As the time-to-market for new products continues to lower and the product life-time shortens, more and more demand is generated for flexible architectures to adapt to fast changing requirements and technologies. A system-on-a-chip (SoC) [50] combines different hardware architecture on one chip. They are connected by a network-on-chip (NoC) interconnect. The different hardware architecture positions are called tiles. Each tile can do what it's good in. For example, a GPP tile can control everything and a coarse-grained reconfigurable hardware tile can perform a signal processing task such as audio decoding or error correction.

For the digital spectrum analyser, the digital RBW filter and the FFT would be suitable signal processing tasks for a coarse grained reconfigurable device.

8.3.1 Montium Tile Processor

The Montium Tile Processor (TP) is a coarse-grained reconfigurable device (Figure 165). Processing is performed using of five processing parts. The sequencer controls a sequence of instructions, i.e. a program. The instruction decoding block decodes instruction into control signals for the processing parts. Each processing part contains an ALU, a register bank, four deep for each of the four inputs, and two local memories. All signals are of 16-bit precision. Each ALU consists of two levels, which are shown in more detail in Figure 166. The first level contains four function units capable of logical functions and basic arithmetic. There are two function units on top, which are connected to the register bank. The two topmost function units are connected to four register banks providing input. The lower two function units are connected to the output of the units above. The second level of the processing part contains a multiplier, followed by an adder/subtracter giving a multiply-accumulate unit (MAC). The MAC is followed by a butterfly structure, extending the MAC to be able to perform a butterfly operation (see 8.3.2 below). The 5 processing parts and 10 memories are connected to each other and to the outside world by 10 global busses. Each ALU level, memory or entire processing part can be turned off when not used, saving energy [50].

The Montium's connection to the outside world, besides the 10 global busses, is the communications and configuration unit.

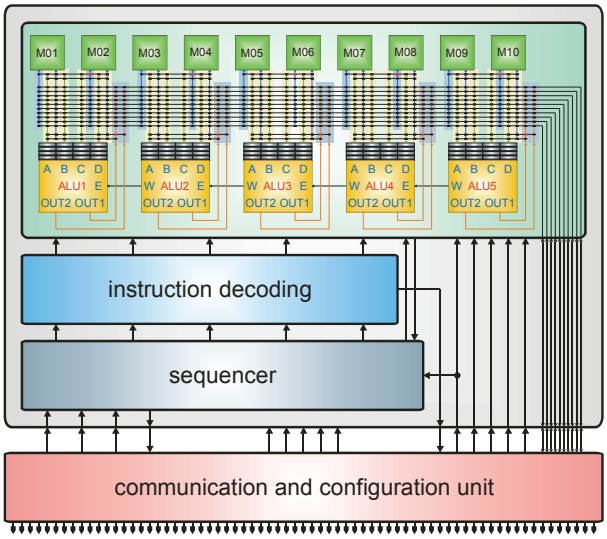


Figure 165. Montium Tile Processor.

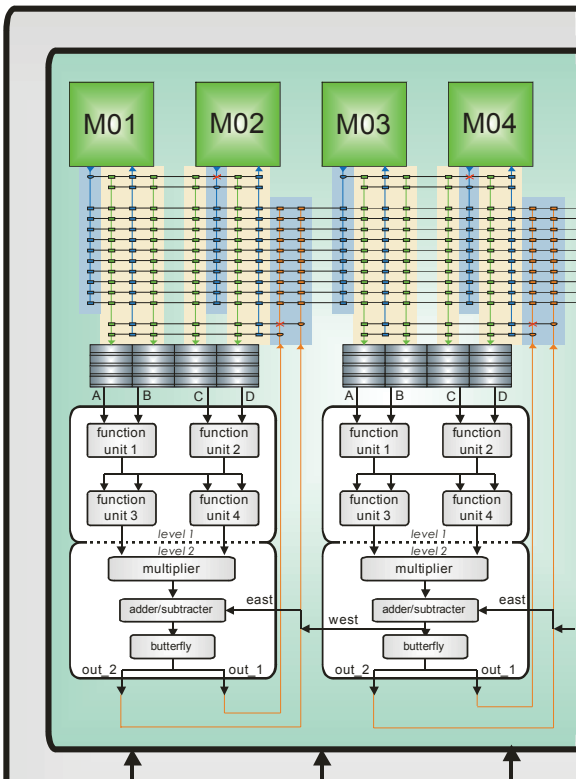


Figure 166. Montium processing parts.

The Montium can be used as DSP processor implementing a FIR filter, a window, an FFT and possibly other processing. In order words, everything after the ADC could be performed by one or more Montiums. However, the Montium is designed to be part of a system-on-chip, in which a Montium performs a specific task under control of a general purpose processor. The Montium is specifically suited to perform DSP tasks such as a FIR filter or an FFT. Implementations of both are available. We will focus on the implications of performing an FFT on the Montium.

8.3.2 FFT

An implementation for a "power-of-two point" FFT is available. The source can be compiled into a configuration binary for 2 to 11 stages for a 4 to 2048 point FFT. So for an N-point FFT, we need $log_2(N)$ stages. A new configuration must be compiled when a different number of stages is wanted. This can possibly be changed into a more flexible implementation, because normally only a one-size FFT is needed instead of a varying size FFT, so this probably has not been taken under consideration for the source.

The FFT implements the complex discrete Fourier transform. An elementary operation of the FFT consists of a MAC combination, which is structured with a winged like appearance. Therefore this is called a butterfly operation [50]. The butterfly operation corresponds to computing two complex equations. The Montium uses 16-bit real numbers. The butterfly operation is therefore broken up into real and imaginary parts, giving four equations. This can be implemented on the Montium using the MAC and butterfly structures of level 2 of the ALUs. One butterfly operation therefore needs four ALUs and with five ALUs on the Montium, one butterfly operation can be computed each clock cycle. Each butterfly uses two input samples to compute an output sample every stage with two cycles overhead. The total number of clock cycles for an N-point FFT is therefore $(N/2+2)\cdot log_2(N)$. This is 5140 clock cycles for an 1024-point FFT or 51.4µs at 100MHz. [50]

The maximum number of stages is limited by the Montium's local memories. There are ten local memories. Besides inputs and outputs, two local memories are used by the butterfly computation. Half the memories are inputs and the other half outputs. This leaves four memories for input samples. Complex numbers use two memory locations and thus we have two times the capacity of one local memory as number of input samples. One local memory has 1024 locations, so the largest FFT we can compute is a 2048-point FFT.

8.3.2.1 Fixed-point arithmetic

The Montium operates on 16-bit words. The number representation is indicated as <*a,b*>, with *a* the number of bits before the radix point and *b* the number after. The ALUs support 16-bit two's complement integer <*16,0*> arithmetic and <*1,15*> fixed-point arithmetic. The <*16,0*> supports integers from -32768 to 32767. In two's complement the first bit also indicates the sign of the number. The <*1,15*> representation has a range of -1 to $1-2^{-15}$. Again the first bit indicates the sign of the number, directly followed by the point. The remaining 15 bits, divides the -1 to 0 and 0 to $1-2^{-15}$ (from here on rounded as 1) ranges into $1/2^{15}\approx 3.05\cdot 10^{-5}$ sized steps. Note that a floating point representation allows for a much larger range in values (although also with varying step sizes over the range). However, the hardware implementation of floating-point arithmetic is much more difficult and is therefore not available on the Montium.

The first ALU level optionally and the second obligatory uses saturated arithmetic. This means that in case of an overflow, the representation remains saturated at the highest or lowest number. This minimizes the error of the computations. [50]

The FFT only uses second level saturated fixed-point arithmetic. To maximise the dynamic range and minimize the error, the largest possible input signal level of -10dBm must correspond to 1. The smallest (peak to peak) signal level is $3.05\cdot 10^{-5}$, which is -96dB. This corresponds to -106dBm with a maximum input signal of -10dBm.

8.3.2.2 Scaling

The DFT and thus the FFT includes a scaling factor. A signal with an amplitude of 1 in the time domain has an amplitude of N in the frequency domain. The butterfly operation consists of multiplications followed by additions and subtractions. Because the signals are represented as <1,15> fixed-point, the peak amplitude can never be larger as 1 or smaller as -1. The multiplication result therefore also can not be larger as 1 or smaller as -1. The following saturated addition can overflow however, as one and one is two. Each input signal is only used once each stage and can only become twice at large at maximum[14]. Because for an N-point FFT there are $log_2(N)$ stages, the amplitude of 1 can become N at most if it falls exactly in a frequency bin.

[14] One part of the butterfly contains two additions in a chain. However, one addition combines the result of a multiplication of two complex values, broken up into real and imaginary parts. This result can never be larger as one if the amplitudes of the complex values were equal to or smaller than one. For example a complex value with amplitude one has a real and imaginary part of $1/\sqrt{2}$ each or a real part of 1 and an imaginary of 0 or another combination. If multiplied with another complex value (by two multiplications for the four parts) with the same restriction the result becomes ½ in each path or 1 in one path and 0 in the other or something in between. Together this is 1 at maximum.

Unfortunately, an overflow severely distorts the results. Therefore, in order to avoid an overflow the input signal must have a maximum amplitude of 1/N, becoming 1 at the output. This on the other hand limits the dynamic range, especially for large N (i.e. with 10 bits for a 1024-point FFT).

Another option which is available on the Montium is scaling the result by a factor 1/2 each stage, instead of 1/N at the beginning. If all results are divided by 2 after a stage, you lose one bit of precision. Thus when the signal is known to be smaller then one before hand, scaling can be applied only in a few of the stages, indicated by a provided mask. One can imagine that it is optimal, as in the highest precision is achieved, if the signal is scaled at the latest possible. After the scaling of one stage, this applies again. The best result would thus be achieved by an adaptive scaling[15]. At the moment this is not available. For an optimum dynamic range to begin with, the maximum signal is best amplified to correspond with ±1 on the Montium. Scaling is then at least needed in the first stage. However, an amplifier may not always be possible (for example for cost or complexity reasons).

As the input signal of a spectrum analyser is unknown, it may contain only small signals or many large signals or anything in between. Assuming the largest possible signal (-10dBm) is scaled to correspond to ±1 on the Montium, scaling is always needed in every stage. Scaling the maximum signal (amplitude) to ±1 and scaling in every stage ensures the best possible precision without the possibility of an overflow.

8.3.2.3 Complex input

Note that a signal in the time domain only has real values; the imaginary part is zero. This leads to even symmetry for the frequency spectrum and odd symmetry for the phase spectrum. The imaginary part in the time domain can also be used by the FFT to provide twice as many results. If a different time domain signal is used as the imaginary part, their spectra can be separated in the frequency domain by using even/odd decomposition [49]:

(8-1) $$x_E[n] = \frac{x[n] + x[N-n]}{2},\ x_O[n] = \frac{x[n] - x[N-n]}{2}$$

The even frequency spectrum and odd phase spectrum belongs to the real time domain part. The odd frequency spectrum and even phase spectrum belongs to the imaginary time domain part.

However, in order to avoid an overflow as explained above, the real and imaginary time domain parts as a complex signal may not become larger than 1. Because:

(8-2) $$|x| = \sqrt{\Re(x)^2 + \Im(x)^2}$$

This condition is satisfied if both parts have an amplitude of less then $1/\sqrt{2} \approx 0.707$.

8.3.3 Results

In this section we will compare the results of the FFT in MATLAB/Simulink and on the Montium. Both use the same input samples, which are windowed and zero padded to the next power of two (Figure 167). A -10dBm input signal is used sampled and quantised by the 2MHz 14-bit ADC. Only single FFT is performed without LO stepping. This ensures no distortion is created by image frequencies. Four filter periods is waited for the filter response to settle.

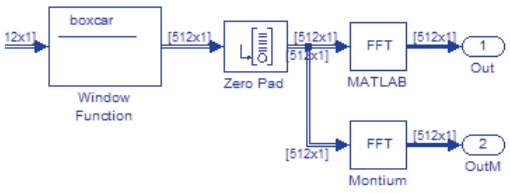


Figure 167. FFT in MATLAB and on the Montium.

The FFT in MATLAB operates with floating-point numbers, which therefore have a far higher dynamic range than the 16-bit fixed-point numbers of the Montium. As seen with the ideal simulation, the floating-point precision limits the minimum signal level at about -300dBm. For 16-bit numbers the dynamic range is about 96dB limiting the minimum signal level at about -106dBm.

[15] An overflow could directly trigger a recomputation with scaling within the same clock cycle. A flag could indicate that this was done. A different option would be a software implementation, but this would possibly add extra clock cycles.

For the Montium the following compilation and simulation software is used:

- Montium LLL compiler v01.09.10
- MontiumSIM simulator
- Power-of-two FFT source code

First the effect of scaling on the magnitude and phase errors is analysed. Next, the Chebyshev window with different attenuation depth is looked into.

8.3.3.1 Scaling

As a start the effect of an overflow is analysed by not scaling the input signal at all. A 512-point FFT is used. The result for a -10dBm 8.3MHz sinusoid without a window is shown in Figure 168. The blue result is from MATLAB, the red result is from the Montium. It is clear that the peak from the input signal at 0.3MHz is clipping. Figure 169 shows the magnitude and phase error. The difference between the Montium and MATLAB is shown in red and the absolute difference in green. Indeed the largest error is at 0.3MHz at the IF. When the result becomes too large, it is saturated. This saturated result is also used for the calculation of other frequencies, which also become distorted as shown.

An overflow is thus clearly undesirable as a lot of other frequencies are also distorted. One way of preventing an overflow is by scaling the input signal so the largest possible amplitude corresponds to 1 for the Montium values. We assume an input signal with a maximum amplitude of 0.1V (see section 8.2.1.2), thus the input signal is multiplied with 10. The result is shown in Figure 170. Indeed no large distortions are visible. The

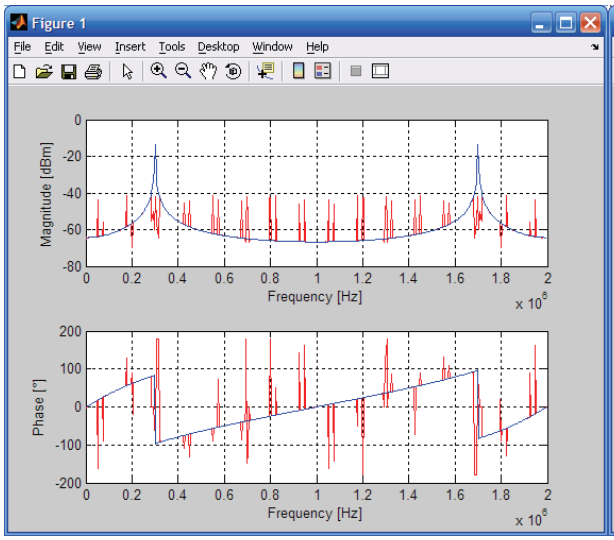


Figure 171. Unscaled input results in clipping for the Montium.

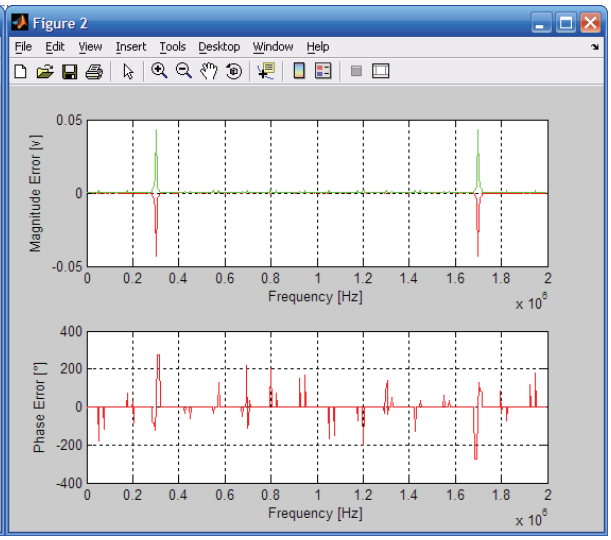


Figure 168. Error of the Montium with an overflow.

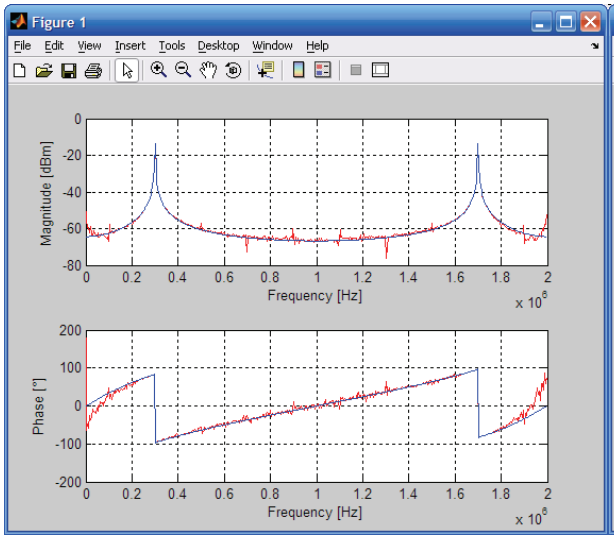


Figure 169. Input scaled by N before FFT.

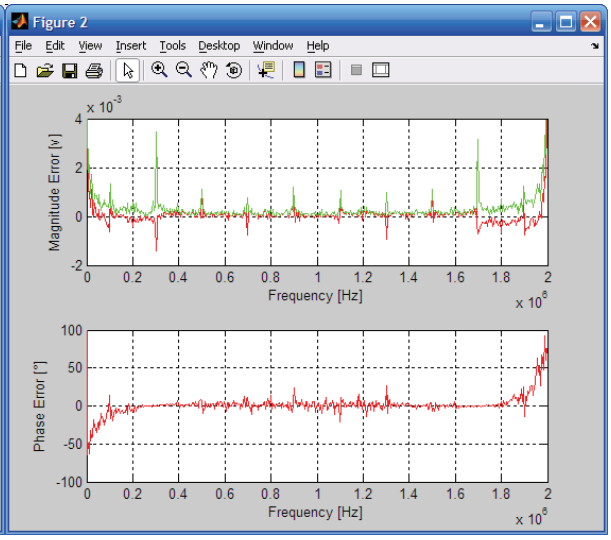


Figure 170. Error with input scaled by N before FFT.

magnitude error, shown in Figure 171, has a peak of $8 \cdot 10^{-3}$, which is -42dB. The average error is $3.3 \cdot 10^{-4}$, which is -70dB. Note that the error becomes larger for low frequencies, especially the phase error. The reason for this is not further analysed.

A better option then scaling before hand, is scaling every stage of the FFT, as explained in section 8.3.2.2. This is enabled in the source and the result is shown in Figure 172. The error in magnitude and phase is hardly visible in this figure. Figure 173 shows the error. It is about 100 times smaller, with the largest error at about $8 \cdot 10^{-5}$, which is -82dB and the average error at $2.5 \cdot 10^{-5}$, which is -92dB. The phase error is relative large, up to 3°. The average error is 0.55°.

The results are also checked for two -16dBm (0.05V amplitude) sinusoids and for the sound wave from section 8.2.2.3. This is shown in Figure 174 and Figure 175. For both, the average magnitude error is also $2.5 \cdot 10^{-5}$. The angle shows some wrapping (jumps from -180° to 180° or the other way around), which does not occur for the MATLAB result. When unwrapped, this gives only a small error.

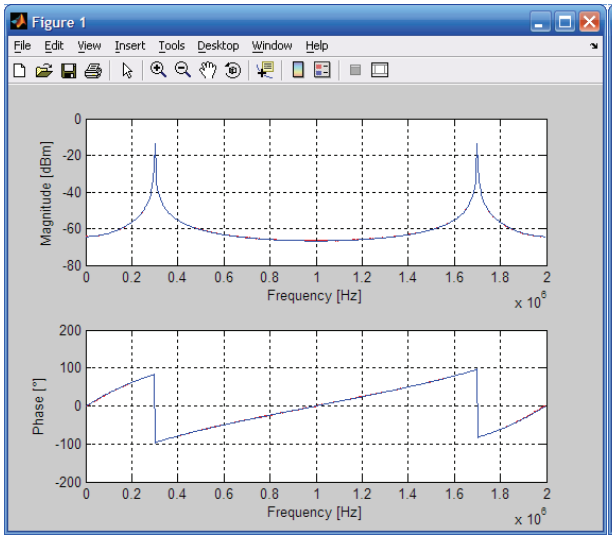


Figure 172. Signal scaled each stage.

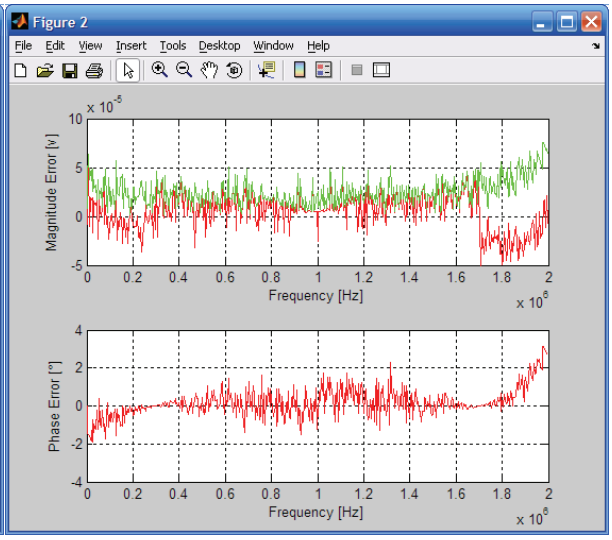


Figure 173. Error with signal scaled each stage.

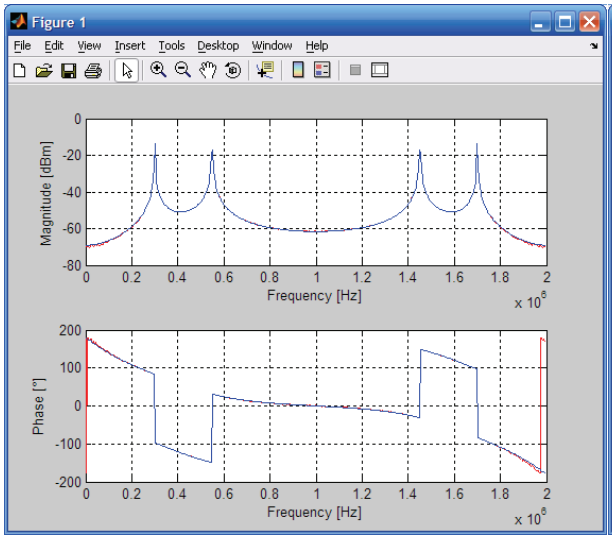


Figure 174. Two input sinusoids, scaled each stage.

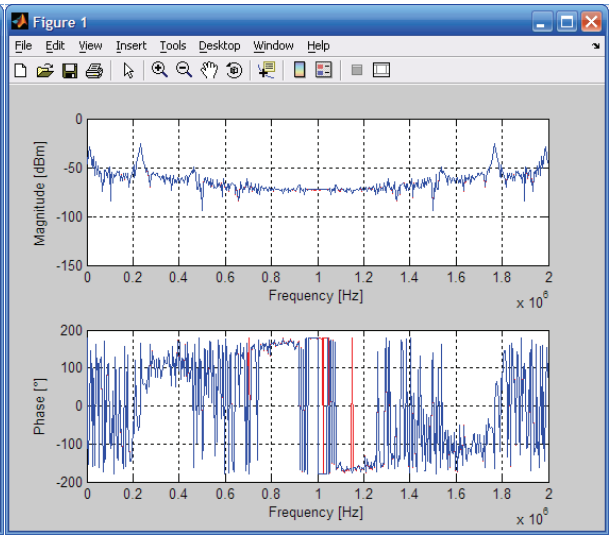


Figure 175. Sound wave input, scaled each stage.

8.3.3.2 Window

The side-lobes with a rectangular window (i.e. no window) go down to about -60dB and therefore do not go down to the noise floor. We would like the noise floor to be below -94dBm (section 7.4.2.2). With a maximum signal of -10dBm this is an 84dB range. Thus, we apply a Chebyshev window with an 84dB attenuation. The result is shown in Figure 176 and Figure 177. Again the result from MATLAB is shown in blue and the result from the Montium in red. The green result gives the absolute difference in magnitude.

The results show the Montium has a lower limit of about -96dBm, which is -86dB. The magnitude does not have a continuous response because some of the results are zero, which can not be shown on a logarithmic scale. This limit is not caused by the Chebyshev window as shown by the MATLAB result. This result was also checked with a much smaller quantisation step (i.e. a 44-bit ADC) as well as with signals of different amplitudes. The result is always limited at about -86dB and is thus caused by the calculation (round-off) error of running an FFT on the Montium.

The round-of error is not uniformly distributed because the MATLAB results are consistently much smaller than -86dB. The MATLAB results can therefore be seen at virtually zero compared to -86dB. If about half of the results are zero and half are -86dB, which is $5.0 \cdot 10^{-5}$, the average is $2.5 \cdot 10^{-5}$. The found limit is thus confirmed by the average magnitude error, which is also $2.5 \cdot 10^{-5}$. In the worst case, -86dB can be seen as the noise level. Note that 16-bit signals have a signal to noise ratio (S/N) of 96dB, thus running the FFT on the Montium adds about 10dB (or 2-bits) of noise in the worst case. As the proposed ADC for the spectrum analyser has 14-effective bits with an 84dB S/R, this is not a problem.

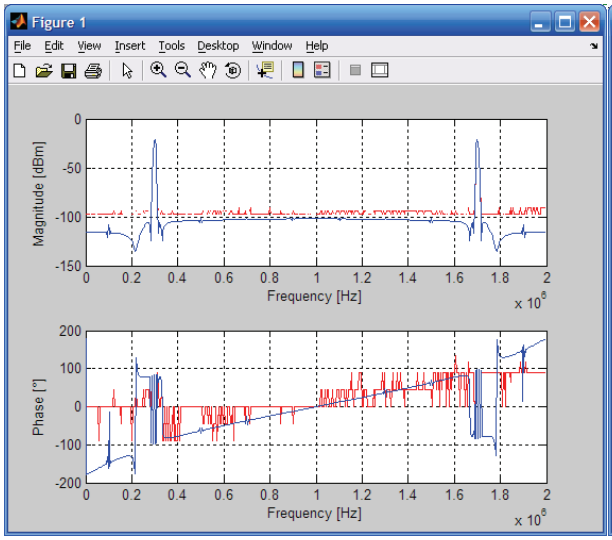


Figure 176. FFT results with 84dB Chebyshev window.

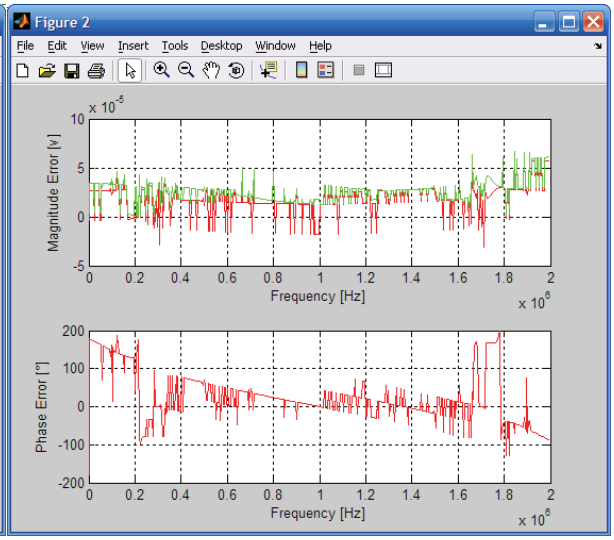


Figure 177. Error of the Montium with a 84dB Chebyshev window.

It was found above that the magnitude error was not different that the results of section 8.3.3.1. However, a consequence of the noise limit is that everything below -96dBm gives a meaningless phase result. The phase is therefore severely distorted. If the phase is important, the attenuation caused by the window must not be allowed to fall under the round-off noise level.

Figure 178 shows the result for a attenuation of 74dB. Because this attenuation is down from the peak of the signal and the peak is at -20dBm, the spectral leakage level is

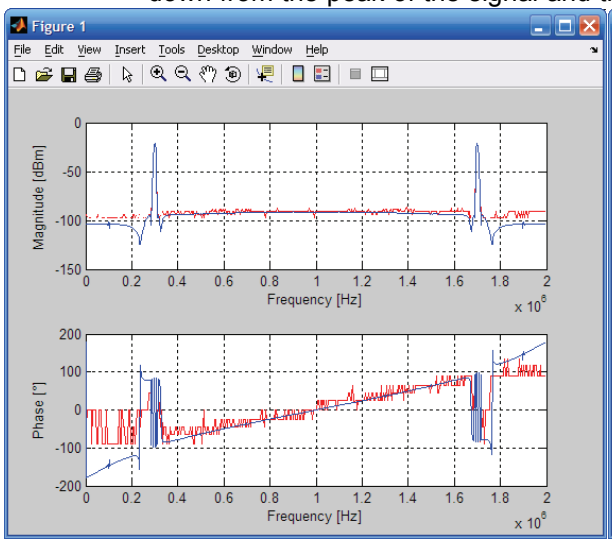


Figure 178. FFT results with 74dB Chebyshev window.

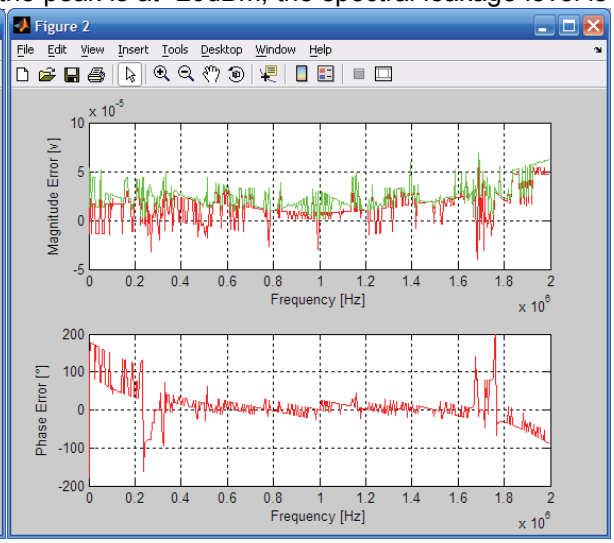


Figure 179. Error with a 74dB Chebyshev window.

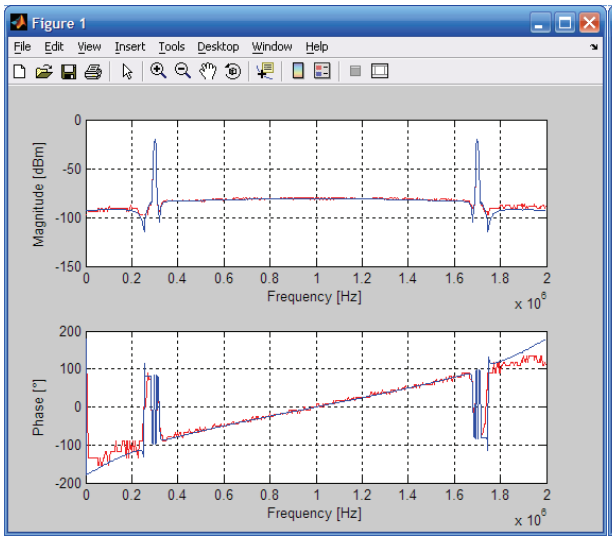


Figure 180. FFT results with 64dB Chebyshev window.

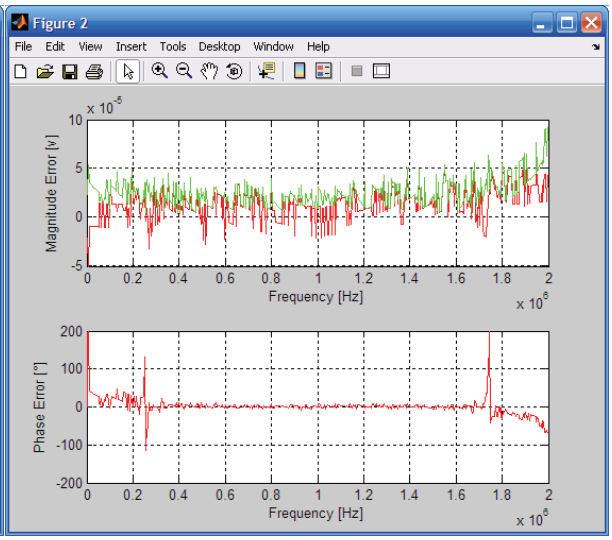


Figure 181. Error with a 64dB Chebyshev window.

at -94dBm, just above the round-off noise. Figure 179 shows the phase error is much smaller for higher frequencies, but is still large around the signal peak and at low frequencies.

Figure 180 and Figure 181 show the results for a 64dB Chebyshev window. Now the phase error is small from the signal at 0.3MHz and up.

8.4 Summary and Conclusion

This chapter confirms the proper operation of the digital spectrum analyser front-end. The result in MATLAB is compared with the result of running the digital signal processing on a reconfigurable processor; the Montium.

8.4.1 Summary

A library of spectrum analyser is presented and used to implement the digital spectrum analyser front-end. The analogue part consists of the input signal, the LO, the mixer, the anti-alias filter and the ADC. The LNA and LPF are not implemented. For the input different implementation are used. The LO is an ideal sine wave. The mixer is implemented as a linear multiplier. The anti-alias approximates a Gaussian filter with an 8^{th} order Bessel filter. The transient response of the Bessel filter takes twice as long to settle and four times the period of the corner-frequency is used as offset instead of the two needed for a Gaussian filter. The ADC is implemented with a 2MHz sampling rate, 0.1V saturation and 14-bit quantisation.

At the digital side, an FFT is used preceded with an offset for removing the filter transient distortion, a window and zero-padding. The result after the zero-padding is used as input for the FFT for both MATLAB and the Montium.

The design is simulated with a single sine input; without spectral leakage, with quantisation noise with the same period as the input signal and with spectral leakage. The resulting side lobes give an attenuation of about 60dB. Next a Chebyshev window with 84dB of attenuation is applied. The result has a higher attenuation, but the peak is 2 to 3 times wider and the phase is partly distorted. The offset is confirmed to need at least four filter periods.

Next the LO frequency is stepped over a frequency range and the results are combined in a single figure. Aliased results of the single sinusoid are identified and the anti-alias filter shape is shown in the resulting spectrum. These responses are removed by over-sampling, but one extra peak remains. This is recognised as the image location of the input sinusoid, resulting from the use as a low-IF architecture. The results of using a 32-point FFT with 8 LO steps and a 1-point FFT with 128 LO steps are compared over the same frequency range. Both thus have 128 points of resolution in the resulting spectrum. The first is a low-IF architecture and has two resulting frequencies. The second is a zero-

IF architecture which shows one resulting frequency, but with larger side lobes from the anti-alias/RBW-filter.

Then the result of using two sinusoids is shown. First the sinusoids are located in two different frequency bands, giving four resulting frequencies. Next they are close together with not enough resolution to differentiate them. The resolution is increased giving four resulting frequencies again. If gain is applied, the ADC overflows and is saturated. The resulting spectrum is severely distorted. Finally, one sinusoid is located as the image of the other, resulting in a distorted wanted signal as expected.

Also a few samples of a sound wave are used as input. First the sample rate is changed to 2MHz and then it is modulated with a slightly different frequency then it is down-converted with. With the complete bandwidth of the sound wave as input to the FFT, the result is fairly accurate. When the bandwidth is analysed in two LO steps, the result is distorted, because one frequency band is the image of the other.

The Montium reconfigurable tile processor is introduced. The Montium is found to be particularly suited to perform dedicated DSP tasks, while also able to be quickly reconfigured. This makes the Montium a good choice for performing the RBW FIR filter and the FFT of a digital spectrum analyser. The FFT is further analysed.

The Montium uses 16-bit fixed point numbers. The signal-to-noise ratio is therefore limited to 96dB and the fixed-point representation support number from -1 to 1 in $3 \cdot 10^{-5}$ sized steps.

Because the FFT results add up each stage, a sinusoid of amplitude 1 results in a frequency component with magnitude N (with N the number of FFT points). This causes an overflow. Therefore the input signal must be scaled before the FFT or during. It is shown that scaling the maximum amplitude of the input signal to correspond to 1 on the Montium and scaling after every stage of the FFT, gives the highest precision.

Finally, we show that because the Montium (as does MATLAB) calculates the complex discrete Fourier transform, we can use a second time based signal as imaginary input, giving two times as many results for one FFT.

Next, the FFT is simulated on the Montium and compared with the result from MATLAB. First it is shown that an overflow distorts the results at multiple frequencies. Then the input signal is scaled before running the FFT. The average magnitude error is $3.3 \cdot 10^{-3}$. The best result is achieved when scaling after each stage of the FFT. The average magnitude error is now $2.5 \cdot 10^{-5}$.

The effect of a window is also evaluated. It is shown that the FFT on the Montium has a round-off error (noise) level of about -86dB, which agrees with the average magnitude error. If the spectrum fall below this level, the phase result becomes meaningless. Therefore, if the phase is important the attenuation must be reduced, as shown for a -74dB and -64dB attenuation.

8.4.2 Conclusion

The digital spectrum analyser architecture is simulated in the ideal case with a simple mixer and a stepping LO. An 8^{th} order Bessel filter is used as an approximation for the Gaussian anti-alias filter. The proper operation is confirmed for a single sinusoid as well as for a real-life input signal. The effects of spectral leakage and using a window are shown. Aliased frequencies are recognised and rejected by using over-sampling. The image for the low-IF architecture is identified and shown not to exist when used as a zero-IF architecture. The image is shown to be problematic when analysing a real-life input with multiple LO steps.

A 16-bit reconfigurable processor, the Montium is adequate for performing the digital signal processing. The maximum input signal must be scaled to correspond to 1, the maximum value of the fixed-point representation of the Montium. This makes scaling in every stage of the FFT necessary and gives the lowest distortion of the results and thus highest precision. The round-off error level is at about -86dB. Furthermore, the attenuation of the Chebyshev window must be lowered to about 64dB if the phase response is important.

Chapter
9

9 Results

In this chapter, an overview of the results is given. A summary of the main conclusions with respect to the goals set is provided in section 9.1. For the treatment of the conclusions in more detail, see the summary and conclusion section at the end of each chapter. A discussion of the results and future work in provided in section 9.2.

9.1 Conclusion

The main conclusions of this thesis are:

- The spurious free dynamic range is used as a performance indicator for spectrum analyser and used to derive the other specifications.
- Frequency conversion is needed to support a large (up to 6GHz) frequency range. The zero-IF architecture is recommended as low-cost integrated conversion architecture without an image problem.
- The time-frequency resolution duality limits the use of a high frequency resolution with a limited sweep time. A Gaussian filter is found to show the best speed versus selectivity trade-off, which follows from the fact that a Gaussian filter has the same shape in both domains.
- The synchronously tuned filter is the recommended Gaussian filter approximation, because of its simple design and less critical component values.
- A quadrature architecture is recommended for instant amplitude measurement and the determination of the phase.
- A digital spectrum analyser with the ADC placed directly after the frequency conversion provides the most flexible solution.
- Digital RBW filters provide a larger selection in frequency resolution and better selectivity. Very high selectivity at the cost of time is possible.
- The FFT is faster and can achieve a higher resolution. It also provides the phase spectrum.
- When an FFT is used, the front-end has low-IF characteristics and images must by rejected by using a quadrature architecture. The image rejection of a quadrature architecture is 40~60dB, reducing the dynamic range.
- The proper operation of the ideal front-end design is verified by simulation.
- A 16-bit fixed-point reconfigurable processor is adequate for the digital signal processing following the analogue front-end.
- The LO higher harmonics and the DC offsets are the most prominent remaining problems.

The aim of his project was to provide a well-balanced spectrum analyser front-end. The main objective of a spectrum analyser is providing the power spectrum of a selected frequency range accurately. Of secondary concern is determining the phase relation between signal, useful for code and modulation domain analysis.

The aims and goals as defined in chapter 1 have been met. A balanced set of specifications is provided and an analogue and digital spectrum analyser front-end design with clear documentation of its limitations and constraints is presented. The digital design is confirmed by simulation.

We distinguish two operating principles. The analogue spectrum analyser uses a resolution bandwidth (RBW) filter to conceptually select a “single” frequency after which the signal is measured. This is repeated for the frequency band of interest one resolution bandwidth after the other. Digital spectrum analysers can also use a fast Fourier transform (FFT) to select a number of frequency bands at once and measure the signal.

An input amplitude range of 70dB and an input frequency range from 0~3/6GHz must be supported. The frequency resolution to support is from 10kHz to 1MHz in 7 steps.

The spurious-free dynamic range (SFDR) specification combines a number of specifications to provide an important indicator of the spectrum analyser performance. Overall, there are three important factors limiting the dynamic range:

- the overall noise floor of the system limited by the temperature and the used bandwidth,
- the phase noise of the local oscillator determined by its Q factor,
- and the linearity distortions of the system.

For lower RBWs, the maximum dynamic range is limited by the phase noise and for higher RBWs by the noise figure, while improving the linearity distortions helps both.

The best compromise is a SFDR of 70dBc at an input level of -20dBm. The DANL is specified at -154dBm/Hz, which corresponds to a 20dB NF. Together, this gives linearity requirements of IP_2=54dBm and IP_3=17dBm. At a 1MHz RBW the input range is -90dBm ~ -20dBm. For a 10dB smaller RBW, the DANL and the smallest supported signal also fall 10dB. Note that these recommended requirements are still quite ambitious but feasible. The specification are at some points even competitive compared to the high-end devices available.

9.1.1 Analogue spectrum analyser front-end

A low-noise amplifier and low-pass filter condition the input signal of the spectrum analyser to a -90dBm ~ -20dBm input amplitude range with a frequency range of 0~3/6GHz. After this, gain of the signal increases the linearity requirements of the following blocks, while attenuation lowers the dynamic range. Therefore no gain is applied.

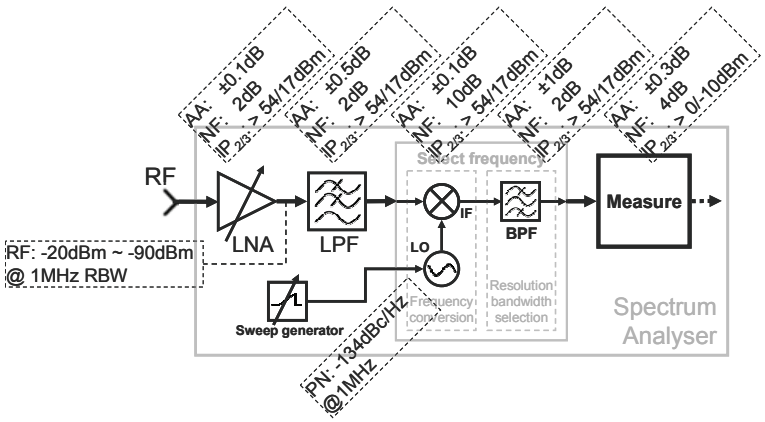


Figure 182. Division of the specifications over the spectrum analyser's sub-blocks.

The "select frequency" block is sub-dived into a frequency conversion and frequency selection block. The frequency conversion consists of a mixer and a local oscillator, which is multiplied with the input signal. A pure sinusoid LO is prefered with a linear mixer. As this is probably not achievable, higher harmonics, which also convert to the intermediate frequency, remain a problem. The frequency selection consists of the RBW filter.

The division of specifications over the block is indicated in Figure 182. The amplitude accuracy is indicated by AA and the phase noise by PN.

Frequency conversion is needed to support the complete input frequency range. Frequency conversion is always based on mixing in this design, but reveals different aspects to consider depending on the choice of IF and mixer.

If the LO frequency is higher than half the RF, the image is a distance of two times the LO frequency from the RF. If the LO frequency is lower, the image is a distance of two times the IF from the RF.

A quadrature mixer can be used to instantly determine the amplitude and phase of the input signal. Another important use is the ability to select the upper or lower sideband around the LO frequency, which is exploited to perform image rejection.

Image rejection can be achieved by filtering or by a quadrature architecture. They can also be used both. For an analogue front-end, the Hartley quadrature architecture is preferred because a Hilbert transform over 1MHz, is easier than the needed filter for the secondary image of the Weaver architecture. The digital architecture gives more control over the achieved image rejection.

The image rejection is typically 40dB because of mismatch and can be improved to typically 50~60dB by calibration or differential circuits. The image rejection of a quadrature architecture is not enough for the required 70dB SFDR.

For the choice of the IF location, the zero-IF architecture comes out preferably, because it has no image problem. For low-IF, a quadrature architecture is needed which does not provide enough dynamic range. For the mid-IF and high-IF solutions, the high frequencies and the need for multiple down-conversion stages is prohibitive.

The zero-IF is distorted by DC offsets caused by LO self-detection, RF self-detection, thermal noise, 1/f noise and second order distortion products. DC removal can be performed by a high-pass filter. The high-pass corner frequency is chosen at $1/10^{th}$ of the RBW bandwidth, because of the trade-off between DC removal and signal loss. Possible input signals that are filtered out are likely to be distributed into the pass-band by the LO phase noise. For a 10kHz RBW, the sweep time with DC removal is about 10s (only taking the RBW settling into consideration). If a 1/f noise corner frequency of 1kHz is achieved, the DANL will drop when going from 1MHz to a 10kHz RBW. Otherwise, the DANL will be at a fixed level for an RBW lower than 10 times the 1/f corner frequency.

The resolution bandwidth filter performs the frequency selection. We would like a good selectivity, preferably blocking the next RBW -74dB (the SFDR plus 4dB of headroom), as well as a fast response so the sweep time is short.

An ideal filter (with respect to selectivity) is not realisable because it is not causal. It also exhibits a lot of ringing in the time domain. Filters designs approximate the ideal response according to different criteria. During the evaluation, a selectivity versus speed trade-off is identified being a fundamental property of the time and frequency domain relationship (the scaling theorem of Fourier transforms). It is found that a filter with a (nearly) Gaussian frequency response provides the best trade-off.

The Fourier transform of a Gaussian is also a Gaussian, which explains the well-balanced frequency and time behaviour. The synchronously tuned filter approximates a Gaussian response because of the Central limit theorem. Because of its simple design and less critical component values, it is recommended.

For the analogue spectrum analyser, the measurement block consists of a linear amplifier, which scales the input range or a logarithmic amplifier, which compresses the dynamic range. The signal is measuremed by an envelope detector or 8-bit 1MHz ADC.

9.1.2 Digital spectrum analyser front-end

For a digital spectrum analyser, the ADC can be seen as the measurement part, which is placed after the frequency conversion in front of the frequency selection. At least a 16-bit (with 14 effective bits) 2MHz ADC is needed. An 1MHz anti-alias filter is needed which is also used as an analogue 2MHz RBW, when using the RBW principle. The same filter as for the analogue spectrum analyser front-end is used.

Because of the added flexibility, while still being able to perform the spectrum analyser functionality with the same specifications as the analogue architecture, the digital spectrum analyser implementation is recommended. This comes at the cost of an ADC and digital signal processor (DSP).

A digital spectrum analyser design is presented which can both perform frequency selection by stepping the input range through a digital RBW filter as by performing an FFT. Both operating principles can use the same front-end if a quadrature architecture is used.

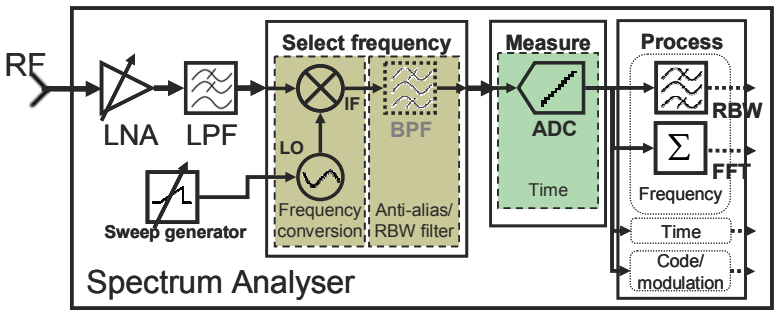


Figure 183. Digital spectrum analyser architecture.

The RBW operating principle provides a higher dynamic range of 70dB, because the image problem is alleviated by choosing the zero-IF architecture. This means the LO must still make RBW sized steps. The same smallest RBW size as the analogue spectrum analyser is recommended, which is 10kHz. Only the largest RBW filter is implemented as an anti-alias filter, the other RBW filters are performed digitally. A quadrature architecture

provides the advantage of instant amplitude and phase determination However, the RBW solution is slow, especially at small bandwidths. Therefore a Gaussian filter is normally used to limit the settling time. The selectivity of digital RBW filter can be much better. With enough time a nearly ideal rectangular frequency response can be implemented.

The FFT operating principle is faster and can achieve a higher resolution, only being limited by the amount of acquired samples and therefore the time to wait and the available processing power. The FFT additionally provides the phase. For the FFT, the front-end is used as a low-IF architecture. A quadrature architecture is necessary for image rejection. The image rejection of a quadrature architecture is limited by the matching and is typically 40~60dB. This is not enough for the specified spurious free dynamic range of 70dB.

For the digital spectrum analyser, the RBW solution can be used if a high dynamic range or very high selectivity is needed or the FFT if high speed and/or resolution is needed. The range of possible RBW filters is also much more flexible, both in terms of bandwidth as well as frequency response.

The digital spectrum analyser architecture is simulated in the ideal case with a simple mixer and a stepping LO. The proper operation is confirmed for a single sinusoid as well as for a real-life input signal. The image for the low-IF architecture is identified and shown not to exist when used as a zero-IF architecture.

A 16-bit reconfigurable processor, the Montium is adequate for performing the digital signal processing. The maximum input signal must be scaled to correspond to 1, the maximum value of the fixed-point representation of the Montium. This makes scaling in every stage of the FFT necessary and gives the lowest distortion of the results and thus highest precision. The round-off error level is at about -86dB. Furthermore, the attenuation of the Chebyshev window must be lowered to about 64dB if the phase response is important.

9.2 Discussion and future work

The presented spectrum analyser front-end design is based on ambitious specifications. However, they are balanced against typical specifications of the identified sub-blocks, every time asking to reach a little further, but by no means asking the impossible. This ensures a competitive product. Strengthening our findings is the trend of recent spectrum analysers such as the Agilent FSP or the Tektronix Real-Time spectrum analyser, which aim for a low-IF followed by digitisation. However these designs use many external components or a high-IF front for image rejection followed by down-conversion to a low-IF. [7][10]

The presented digital design uses a low-cost integrated approach, with a unique combination of a zero-IF based RBW operating principle and a low-IF based FFT operating principle. This provides the high dynamic range of the RBW based approach as well as the higher selectivity, speed, flexibility and code and modulation domain measurements.

As a top-down system and topology study was performed, specific implementation issues of the sub-blocks are not touched upon. The feasibility of the design is based on typical performance. Each sub-part could thus be studied further on its own. However, there are some prominent remaining problems.

The first choice for future work is the design of a local oscillator that can achieve the postulated specifications. The local oscillator needs to support a very wide tuning range from 0 to 3GHz with small 10kHz steps. Furthermore, the required phase noise of -114dBc at 10kHz is difficult. It is easier for lower LO frequencies, but with a zero-IF architecture, the LO frequencies are the lowest possible for the input frequency range. The generated frequency must thus be very stable. Possibly, the complete tuning range can not be achieved with a single LO. A combination of multiple LOs which are multiplied to achieve the final frequency can be thought of. In the worst case, multiple (super-heterodyne) conversion stages of the spectrum analyser are needed. In this case, the

advantages of the zero-IF architecture only apply to the last stage. Image rejection for earlier stages must be evaluated.

Another important issue of the LO in combination with the mixer, is the problem of higher harmonics which also down-convert input signals to zero. This has only been discussed briefly and only the high-IF architecture is identified as a solution. Possible solutions for further research are using a sine approximation with a linear mixer implementation, filtering the higher harmonic locations or using a harmonic rejection mixer. Another idea that surfaced is the use of multiple measurements. By assuming a low probability of higher harmonic distortions, we can perform multiple measurement with slightly different LO frequencies and take the results that match as the undistorted value and the others as distorted. A probability model could be designs which estimated the correctness of the presented result. This result can also be presented to the spectrum analyser user, who can take appropriate actions to remedy the distortions when needed and possible.

A good second would be the further analysis of DC removal. An indication is given that the self-detection problem can be solved by proper design and layout. This could be confirmed with a design and/or simulation. However, this still leaves us with the 1/f noise around DC. Possible solution can be analysed and presented.

If the DC offset can not be adequately removed or is too difficult or too slow, there are some promising candidates for the quadrature architecture. Both the use of a double-quadrature architecture in combination with poly-phase filters, as the use of phase calibration, show an increase in image rejection to 60dB. The calibration approach even predicts that 70dB is possible with gain calibration. This would allow a low-IF design with enough dynamic range and no DC offset problem.

In case of a digital spectrum analyser, the FFT approach now provides all its advantages at the same dynamic range. The RBW approach still can provide a higher selectivity however, so both remain useful. Another advantage of the low-IF architecture with enough dynamic range is that the LO needs steps the size of the ADC bandwidth. Further selection can be performed in the digital domain (for example by down-conversion or filter banks).

Other possible future work consists of a wideband LNA design with an IP_3 of 17dBm or the analysis of the trade-off between over-sampling and taking a more selective filter. Also more extensive simulations could be performed, such as a quadrature architecture and linearity distortions and noise. Finally, the use of analogue active filters could be analysed.

References

[1] Norman Balabanian, *Electric circuits*, U.S.A, McGraw-Hill, 1994

[2] Samir S. Soliman & Mandyam D. Scrinath, *Continuous and Discrete Signals and Systems*, Second edition, New Jersey, Prentice-Hall International, 1998

[3] Behzad Razavi, *Design of Analog CMOS Integrated Circuits – Preview Edition*, New York, McGraw-Hill, 1999

[4] Behzad Razavi, *RF Microelectronics*, New Jersey, Prentice-Hall, 1998

[5] Simon Haykin, *An introduction to analog and digital communications*, U.S.A., John Wiley & Sons, Inc., 1989

[6] A.V. Oppenheim, R.W. Schafer, *Discrete-Time Signal Processing*, Upper Sadle River, NJ, U.S.A., Prentice-Hall, Inc., 1999

[7] Agilent, *Spectrum Analysis Basics*, Application Note 150, U.S.A, Agilent Technologies, Inc, 2004

[8] Agilent, *Spectrum Analysis Measurements and Noise*, Application Note 1303, U.S.A., Agilent Technologies, Inc., 2003

[9] Agilent, *Optimizing Spectrum Analyzer Amplitude Accuracy*, Application Note 1316, U.S.A., Agilent Technologies, Inc., 2000

[10] Tektronix, *Fundamentals of Real-Time Spectrum Analysis*, U.S.A., Tektronix, Inc., 2005

[11] Derek K. Schaeffer, *The Design and Implementation of Low-Power CMOS Radio Receivers*, Stanford University, 1998

[12] Sri Welaratna, *Thirty Years of FFT Analyzers*, U.S.A., Sound and Vibration Magazine, 01/1997

[13] *Wikipedia – The Free Encyclopedia* (2005), http://www.wikipedia.org

[14] *Bruco B.V.* (2005), http://www.bruco.nl

[15] *Agilent Technologies* (2005), http://www.agilent.com

[16] *Rohde & Schwarz* (2005), http://www.rohde-schwarz.com

[17] *Tektronix* (2005), http://www.tektronix.com

[18] Hameg Instruments, http://www.hameg.com

[19] Poor Man's Spectrum Analyzer, http://www.science-workshop.com

[20] The simple man's Spectrum Analyser, http://www.hanssummers.com/electronics/equipment/spectrumanalyser/

[21] Pico Technology, http://www.picotech.com/

[22] HAMEG, *1GHz Spectrum Analyser 5012-2*, Mainhausen, Germany, HAMEG Instruments GmbH, 2005

[23] Tektronix, *Real-time Spectrum Analysers RSA2200A Series*, U.S.A., Tektronix, Inc., 2004

[24] Rohde&Schwarz, *Spectrum Analyser R&S FSP*, Specifications, Version 05.00, Germany, Rohde&Schwarz GmbH&Co, 2004

[25] Rohde&Schwarz, *Signal Analyser R&S FSIQ*, Germany, Rohde&Schwarz GmbH&Co, 2005

[26] Agilent, *PSA Series Spectrum Analyzers*, Data Sheet, U.S.A., Agilent Technologies, Inc., 2004

[27] Metric Test, *Used test equipment rentals, electronic equipment from Agilent, Fluke, Tektronix* (01/2006), http://metrictest.com

[28] *DDS Scope Trading & Measurement. Verkoop en Verhuur van Hameg, Tektronix, Dynatek, Tiepie, Sein* (01/2006), http://www.oscilloscoop.nl

[29] Thomas H. Lee, Ali Hajimiri, *Oscillator Phase Noise: A Tutorial*, IEEE Journal of Solid-State Circuits. Vol 35, No 3, pp. 326-336, 2000

[30] F. Bruccoleri, E.A.M. Klumperink, B.Nauta, *Generating All 2-Transistor Circuits Leads to New Wide-Band CMOS LNAs*, IEEE Journal of Solid-State Circuits, Vol. 36, No. 7, pp. 1032-1040, 07/2001.

[31] Behzad Razavi, *Design Considerations for Direct Conversion Receivers*, IEEE Transactions on Circuits and Systems-II: Analog and Digital Signal Processing, Vol. 44, No. 6, pp. 428-435, 06/1997

[32] Jussi Ryynänen, Kalle Kivekäs, Jarkko Jussila, Aarno Pärssinen, Kari Halonen, *RF Gain Control in Direct Conversion Receivers*, IEEE Proceedings of International Symposium of Circuits and Systems (ISCAS), Vol. IV, pp. 117-120, 05/2002
[33] Tajinder Manku, Christopher Snyder, Michele Ting, Yang Ling, Javad Khajehpour, Bill Kung, Lawrence Wong, *Dual Mixer Downconversion Architecture Using Complex Mixing Signals: Enabling Solutions for Software Defined Radios*, IEEE Custom Integrated Circuits Conference, pp. 227-234, 2002
[34] Raymond Montemayor, Behzad Razavi, *A Self-Calibrating 900-MHz CMOS Image-Reject Receiver*, ESSCIRC '00, pp. 320-323, 09/2000
[35] Jan Crols, Michel S.J. Steyaert, *A Single-Chip 900MHz CMOS Receiver Front-End with a High-Performance Low-IF Topology*, IEEE Journal of Solid-State Circuits, Vol. 30, No. 12, pp. 1483-1492, 12/1995
[36] Farbod Behbahani, Yoji Kishigami, John Leete, Asad A. Abidi, *CMOS Mixers and Polyphase Filters for Large Image Rejection*, IEEE Journal of Solid-State Circuits, Vol. 36, No. 6, pp. 873-886,06/2001
[37] A.P. van der Wel, E.A.M. Klumperink, S.L.J. Gierkink, R.F. Wassenaar, H. Wallinga, *MOSFET 1/f Noise Measurement Under Switched Bias Conditions*, IEEE Electron Device Letters, Vol. 21, No. 1, 01/2000
[38] Eric A.M. Klumperink, Sander L.J. Gierkink, Arnoud P. van der Wel, Bram Nauta, *Reducing MOSFET 1/f Noise and Power Consumption by Switched Biasing*, IEEE Journal of Solid-State Circuits. Vol. 35, No. 7, 07/2000
[39] R.V. Snyder, *A wide-band tuneable filter technique based on double-diplexing and low-Q tuning elements*, Microwave Symposium Digest., 2000 IEEE MTT-S International, Vol 3, pp. 1759-1762, 2000
[40] J.O Smith, *Introduction to Digital Filters*, Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, 09/2005
[41] J.O Smith, *Mathematics of the Discrete Fourier Transform*, Center for Computer Research in Music and Acoustics (CCRMA), Stanford University, 2003
[42] Anatol I. Zverev, *Handbook of Filter Synthesis*, Westinghouse Electric Corporation, U.S.A, John Wiley & Sons, Inc., 1967
[43] H.A. Alzher, H.O. Elwan, I. Ismail, *A CMOS Highly Linear Channel-Select Filter for 3G Multistandard Integrated Wireless Receivers*, IEEE Journal of Solid-state Circuits, Vol. 37, No 1, pp. 27-37, 01/2002
[44] James L. Tonne, *Elsie Student Edition, the electrical filter design and analysis program v2.04* (2005), http://www.tonnesoftware.com
[45] Maplesoft, *Maple v9.5* (2005), http://www.maplesoft.com
[46] T.M.J. Meijer, *kansrekening en statistiek voor EL/INF (probability calculus and statistics)*, nr. 519, University of Twente Press, Enschede, The Netherlands, 1998
[47] T. Deliyannis, Yichuang Sun, J.K. Fidler, *Continuous-Time Active Filter Design*, U.S.A., CRC Press LLC, 1999
[48] Robert H. Walden, *Analog-to-Digital Converter Survey and Analysis*, IEEE Journal on selected areas in communications, Vol 17, No. 4, April 1999
[49] Steven W. Smith, *The Scientist and Engineer's Guide to Digital Signal Processing*, San Diego, CA, U.S.A., California Technical Publishing, 1997
[50] P.M. Heysters, *Coarse-Grained Reconfigurable Processors – Flexibility meets Effieciency*, Enschede, the Netherlands, University of Twente, 2004
[51] The MathWorks, *MATLAB & Simulink Release 14* (2005), http://www.matworks.com

Appendix

A

A Master Thesis Assignment

This is the original master thesis assignment as provided by Bruco [14].

A.1 Front-end investigations for a low-cost spectrum analyzer

Aim

Low-cost spectrum-analyzer receiver system investigations and topology-study.

Description

In order to acquire knowledge of radio systems and to explore new fields of radio-applications for Bruco in general, a low-cost spectrum-analyzer project was set-up. This spectrum-analyzer system has to be designed with an as high as possible degree of integration.

Because the aim of the project is to design a low-cost system, it is accepted that the requirements will be (most likely) rather modest compared to state-of-the-art analyzers from the well known companies like Rohde and Schwarz and Agilent.

The goal of this “D-opdracht” is twofold

a. Perform system investigations.
b. Comparison of (two) different methods of conversion with image-rejection.

Ad a)

The first part of this “D-opdracht” is to derive the relations between thermal-noise, phase-noise, linearity and bandwidth (R.B.W.-bandwidth) of the system up to the output of the R.B.W.-filter (R.B.W. = Resolution BandWidth). Subsequently, the trade-offs must be given between the different quantities in relation to the receiver topology.

The goal is to be able to derive a balanced set of requirements for the spectrum-analyzer in relation to the receiver-topology and chip-technology.

Ad b)

The next step is to compare (two) different downconversion topologies with respect to the above mentioned quantities.

The two most promising topologies are the single (non-zero) IF topology with an integrated image-rejection (polyphase) IF-filter and a double-conversion topology where image-rejection is achieved by two conversions with quadrature oscillators. In this topology a cheap external R.B.W.-filter can be used.

It should become clear what the maximum dynamic range of the integrated filter topology can be as a function of chip-area (total capacitance), supply-voltage and power-consumption and what image-rejection properties can be achieved in practice for all the topologies. Note that calibration is allowed for this spectum-analyzer system.

Deliverables

Report (in English) with all relevant issues included (as conducted during the research project) and with clear conclusions and suggestions for further study.

Appendix

B

B Spectrum analyser history

As technology evolves the requirements to measure meaningful signals have increased. Signals can become very small, very close together and with advanced methods of encoding information, such as for CDMA and TDMA [13]. Thus, the requirements of spectrum analysers have become larger and larger, but on the other hand, new products can only be effectively designed if they can be checked for proper operation, so in effect the development of spectrum analysers and new RF products are probably co-dependant.

Jean Baptiste Joseph Fourier publishes a work in 1822 in which he claims that a function of a variable can be expanded into a series of sines of multiples of the variable. This with certain restrictions forms the basis of spectrum analysis.

The theoretical basis of the propagation of electromagnetic waves was first described in 1873 by James Clerk Maxwell. Depending on that work, Heinrich Rudolf Hertz demonstrated in 1888 the existence of electromagnetic radiation by building a device to produce radio waves. Reception was at first by a number of fixed band-pass filters followed by a detector. A slightly tuneable solution was the crystal receiver by Greenleaf Whittier Pickard from circa 1900, based on a crystalline mineral. The heterodyne receiver was patented in 1902 by Reginald Fessenden, which used a mixer to amplify and convert the input signal to an audible frequency. However, apparently the general utility of frequency conversion was unrecognised until later [11]. The tuned radio frequency (TRF) receiver was used in the early 20th century and patented in 1916 by Ernst Alexanderson and is used to select or tune to a certain frequency. Note that if you select frequencies and detect the amplitude you have a spectrum analyser. This receiver was quickly replaced by the super-heterodyne receiver invented and patented by Edwin Armstrong in 1918, which uses a mixer and a fixed intermediate frequency (IF). This enables better amplification and filtering. Almost all radio transmitter or receiver designs including spectrum analysers up to today use this principle. [13]

The invention of the transistor in 1947 and the advancements in semiconductor device fabrication in the mid 20th century made integrated circuit possible. This resulted in the digital revolution, which consistently moves the digitalisation of the input signal more to the front. This enables more and more advanced signal processing and introduces the concept of software defined radio, which in the ideal case would directly digitise the input signal and process it for all radio standards implemented in the software. The analogue to digital converter (ADC) is not possible at the high frequencies (speed) and resolution needed. Thus, today's software defined radio uses a down-conversion step.

A Fourier transform can be performed in software by the fast Fourier transform (FFT) algorithm, first discussed by Cooley and Tukey in 1965, although Gauss had actually described the critical factorization step as early as 1805. The first purpose built hardware to perform a FFT was the Time/Data 100 of Ed Sloan and Bruce McKeever in 1967. A device that does directly digitise the input signal and perform a FFT for use as a spectrum analyser is the Fourier analyser. The first device was the HP5450 with Ron Potter regarded as the father of Fourier analysers. [12]

Combining the super-heterodyne front-end with the Fourier analyser back end, (vector) signal analysers (VSA) measure amplitude and phase of signals over a wide input range and enable modulation domain analysis.

Recently advanced frequency domain triggering has been added to give real-time spectrum analysers. [10]

The different receivers are further discussed in section 2.2.

Appendix

C

C Fourier transform of a Gaussian shape

The derivation of the inverse Fourier transform is performed.

First we define the frequency response of the Gaussian shape and simplify the representation:

(C-1) $$H(j\omega)=\sqrt{2\pi\sigma^2}\,\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\left(\frac{\omega'-\mu}{\sqrt{2}\sigma}\right)^2}=e^{-a(\omega'-\mu)^2}=e^{-a(\omega)^2}$$

Then we take the inverse Fourier transform:

(C-2) $$h(t)=\mathcal{F}^{-1}\left[H(j\omega)\right]=\frac{1}{2\pi}\int_{-\infty}^{\infty}H(j\omega)\cdot e^{j\omega t}d\omega=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{-a(\omega)^2}\cdot e^{j\omega t}d\omega$$

$$h(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{-a\left(\omega^2-j\frac{t}{a}\omega\right)}d\omega=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{-a\left(\omega^2-jt'\omega\right)}d\omega=\frac{1}{2\pi}\int_{-\infty}^{\infty}e^{-a\left(\omega-j\frac{t'}{2}\right)^2}\cdot e^{-\frac{1}{4}at'^2}d\omega$$

(C-3) $$h(t)=\frac{e^{-\frac{1}{4}at'^2}}{2\pi}\int_{-\infty}^{\infty}e^{-a\left(\omega-j\frac{t'}{2}\right)^2}d\omega$$

Which we can simplify as:

(C-4) $$b\int_{-\infty}^{\infty}e^{-au^2}du\quad with\ b=\frac{1}{2\pi}e^{-\frac{1}{4}at'^2},u=t'+j\frac{\omega'}{2},du=dt$$

Unfortunately this integral is not directly solvable. However, notice that:

(C-5) $$\frac{d}{dx}\left(\frac{-1}{2a}e^{-ax^2}\right)=xe^{-ax^2}\Rightarrow\int xe^{-ax^2}=\frac{-1}{2a}e^{-ax^2}$$

Thus we know the solution to the integral of an exponential resembling the one we want to resolve. We can transform our exponential to the above form by performing a coordinate transformation with a "trick". First we square the integral:

(C-6) $$\left[\int_{-\infty}^{\infty}e^{-au^2}du\right]^2=\left[\int_{-\infty}^{\infty}e^{-ax^2}du\right]\cdot\left[\int_{-\infty}^{\infty}e^{-ay^2}du\right]=\iint_{\Re^2}e^{-a\left(x^2+y^2\right)}dxdy$$ with x=y=u

Note that $\sqrt{(x^2+y^2)}$ corresponds to the radius of a Cartesian to "cylindrical coordinate transformation:

(C-7) $$x=r\cdot\cos\theta,y=r\cdot\sin\theta,r^2=x^2+y^2,dx\cdot dy=\left|\frac{\partial(x,y)}{\partial(r,\theta)}\right|dr\cdot d\theta=|r|drd\theta=rdrd\theta$$

$$\iint_{\Re^2}e^{-a\left(x^2+y^2\right)}dxdy=\int_0^{2\pi}\int_0^{\infty}e^{-ar^2}rdrd\theta=\int_0^{2\pi}\left[-\frac{1}{2a}e^{-ar^2}\right]_0^{\infty}d\theta=\int_0^{2\pi}\frac{1}{2a}d\theta=\frac{\pi}{a}$$

Thus

(C-8) $$\int_{-\infty}^{\infty}e^{-au^2}du=\sqrt{\frac{\pi}{a}}$$

(C-9) $$h(t)=b\int_{-\infty}^{\infty}e^{-au^2}du=b\sqrt{\left[\int_{-\infty}^{\infty}e^{-au^2}du\right]^2}=b\sqrt{\frac{\pi}{a}}=\frac{1}{2\pi}\sqrt{\frac{\pi}{a}}e^{-\frac{1}{4}at'^2}\ \text{with}\ t'=\frac{t}{a}$$

(C-10) $$h(t)=\frac{1}{2\pi}\sqrt{\frac{\pi}{a}}e^{-\frac{1}{4}a\left(\frac{t}{a}\right)^2}=\frac{1}{2\pi}\sqrt{\frac{\pi}{a}}e^{-\frac{t^2}{4a}},a=\frac{1}{2\sigma^2},\omega=\omega'-\mu$$

(C-11)

$$H(j\omega) \leftrightarrow h(t) \Rightarrow e^{-a(\omega)^2} \leftrightarrow \frac{1}{2\pi}\sqrt{\frac{\pi}{a}}e^{-\frac{t^2}{4a}} \Rightarrow e^{-\frac{1}{2}\frac{(\omega)^2}{\sigma^2}} \leftrightarrow \frac{1}{2\pi}\sqrt{2\pi\sigma^2}e^{-\frac{1}{2}\sigma^2 t^2}$$

$$H(j\omega) \leftrightarrow h(t) \Rightarrow e^{-\frac{1}{2}\frac{(\omega'-\mu)^2}{\sigma^2}} \leftrightarrow \frac{1}{2\pi}\sqrt{2\pi\sigma^2}e^{-\frac{1}{2}\sigma^2 t^2} = \frac{1}{\sqrt{2\pi\sigma'^2}}e^{-\frac{1}{2}\frac{t^2}{\sigma'^2}}$$

with $\sigma' = \frac{1}{\sigma}$

The result is that the (inverse) Fourier transform of a Gaussian frequency response is a Gaussian transient response. Of course this also applies the other way around, as they form a transform pair.

Glossary

AA	amplitude accuracy
AC	alternating current
ADC	analogue-to-digital converter
AM	amplitude modulation
BPF	band-pass filter
CDMA	code division multiple access
CMOS	complementary metal oxide semiconductor
DAC	digital-to-analogue converter
DANL	displayed average noise level
DC	direct current
DFT	discrete Fourier transform
DR	dynamic range
F	noise factor
FDMA	frequency division multiple access
FFT	fast Fourier transform
FIR	finite impulse response
FM	frequency modulation
GSM	global system for mobile communications
HPF	high-pass filter
I	in-phase
IEEE	institute of electrical and electronics engineers
IF	intermediate frequency
IIR	infinite impulse response
IM	image
IM_x	intermodulation product
IP_2	second-order intercept point
IP_3	third-order intercept point
LAN	local area network
LNA	low noise amplifier/attenuator
LO	local oscillator
LPF	low-pass filter
MEMS	micro-electro-mechanical systems
N	noise power
NF	noise figure
P	power
PM	phase modulation
PN	phase noise
PSD	power spectral density
Q	quadrature
RBW	resolution bandwidth
RF	radio frequency
RMS	root-mean-square
S	signal power
S/N	signal-to-noise ratio
SFDR	spurious-free dynamic range
TDMA	time division multiple access
TRF	tuned radio frequency
UMTS	universal mobile telephone system
V	voltage
WLAN	wireless local area network