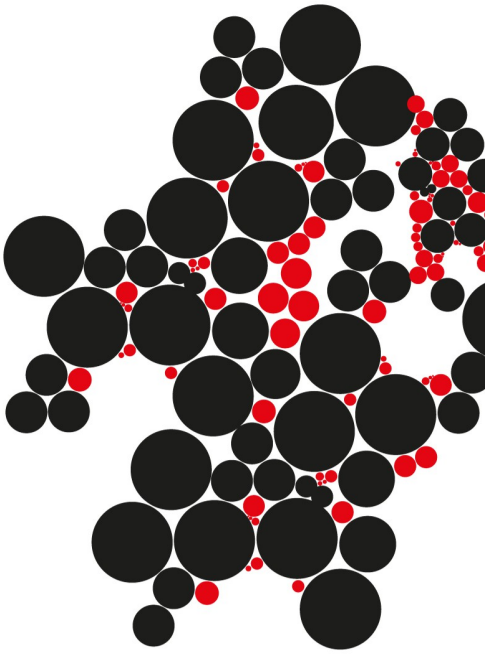
MASTER THESIS



A DFT BASED SYNCHRONIZATION SCHEME FOR CHIRPED COMMUNICATION

René Moll

FACULTY OF ELECTRICAL ENGINEERING, MATHEMATICS AND COMPUTER SCIENCE
COMPUTER ARCHITECTURE FOR EMBEDDED SYSTEMS

EXAMINATION COMMITTEE
Prof.dr.ir. G.J.M. Smit (CAES)
Dr. ir A.B.J. Kokkeler (CAES)
Dr.ir. R.A.R. van der Zee (ICD)
R. Dutta, M.Sc (ICD/CAES/SRR)

UNIVERSITY OF TWENTE.

04-02-2013

A DFT based synchronization scheme for chirped communication

Master's Thesis
by

René Moll
S0214671

Committee:

Prof.dr.ir. G.J.M. Smit (CAES)
Dr. ir A.B.J. Kokkeler (CAES)
Dr.ir. R.A.R. van der Zee (ICD)
R. Dutta, M.Sc (ICD/CAES/SRR)

University of Twente, Enschede, The Netherlands
February 4, 2013

Abstract

The goal of this thesis was to derive a synchronization algorithm for chirped wireless communication systems and provide a high level implementation of this algorithm. The main requirements were speed, accuracy and low power consumption. This work has been done to extend a chirped binary frequency shift-keying (BFSK) receiver developed at the Integrated Circuit Design, Computer Architectures and Embedded Systems and Short Range Radio chairs at the University of Twente.

Based on chirp signal properties, an algorithm has been derived, utilizing the link between time and frequency due to the chirp signal. Mixing two chirp signals leads to two frequency components, their exact frequencies depending on the time offset between the two signals. To determine this time offset, two detection methods are investigated. The first method, positive negative detection (PND), searches for both frequency components, while the second method, maximum power detection (MPD), searches for the strongest frequency component.
Next, mapping the algorithm onto hardware was simulated. Relevant parameters and their performance impact were investigated using an design space exploration, to achieve a high performance with minimal hardware requirements.

Results show that in ninety percent of the simulations the synchronization error is within the system's resolution, which is limited by the fast Fourier transform (FFT) resolution. While transmission power was kept at such a level that, to the receiver, the communication signal was buried under the noise floor. The final system consists of a 1 bit analog to digital converter (ADC), a 256 point FFT and uses the MPD method. The PND method showed to be unsuitable, with a correct synchronization performance of only fifty-five percent. Impact on the communication performance, measured via bit error rate (BER) curves, shows a maximum loss of 1 dB in signal to noise ratio (SNR) performance with a data rate of 4 kHz, but can be reduced by increasing the bandwidth of the modulation scheme.

The proposed algorithm and hardware meet the requirements for speed, accuracy and power requirements. Firstly, as the algorithm requires only one execution, for the duration of half a chirp period. And secondly, the accuracy of the system can be chosen such, that it does not impact the communication performance significantly. Furthermore, recommendations are presented on the type of chirp signals to use. Either one up or down chirp should be used for synchronization. While for communication purposes, a combination of both types should be used to avoid discontinuities in the carrier's frequency, as these discontinuities, in combination with any synchronization errors, lead to loss of communication.

Acknowledgements

Finally. At last I can finish my report and complete my master thesis, ending my time at the Univeristy of Twente. I have had a great time during my study, both with studying as well with activities outside the classroom. For some of these distractions I have to thank Bert Molenkamp, Jan Broenink and Sabih Gerez, it was a learning experience and fun to work for and with you.

Before this period of my life ends, I want to use this place to thank the people who helped me during my work on this project. First of all I want to thank my committee: André, Gerard, Ramen and Ronan. Thank you for your feedback, especially André for going through all the paperwork I produced.

I would also like to thank my friends and family, for their support and motivational questioning, yes this really is the final version :). Thanks Jos and Ed for reading my report, your suggestions helped me a lot.

And finally I want to thank the people at CAES, for creating this pleasant environment to work and be in.

René Moll
Enschede, Februari 2013

Acronyms

ADC analog to digital converter.

ASIC application-specific integrated circuit.

ASK amplitude shift-keying.

AWGN additive white gaussian noise.

BER bit error rate.

BFSK binary frequency shift-keying.

BOK binary orthogonal keying.

CMOS complementary metal−oxide−semiconductor.

CORDIC coordinate rotation digital computer.

CPM continuous phase modulation.

CW continuous wave.

DFT discrete Fourier transform.

DM direct modulation.

DQPSK differential quadrature phase shift-keying.

DSP digital signal processor.

FFT fast Fourier transform.

FoM figure of merit.

FRFT fractional Fourier transform.

FSK frequency shift-keying.

FWL fractional word length.

HBT heterojunction bipolar transistor.

IWL integer word length.

LFS linear frequency sweep.

LO local oscillator.

LSB least significant bit.

MIPS million instructions per second.

MPD maximum power detection.

NBI narrowband interference.

OFDM orthogonal frequency division multiplexing.

OSR oversampling ratio.

PE processing element.

PND positive negative detection.

PSD power spectral density.

PSK phase shift-keying.

RAM random access memory.

RISC reduced instruction set computing.

SAW surface acoustic wave.

SDC serial delay commutator.

SFDR spurious free dynamic range.

SINR signal to interference plus noise ratio.

SIR signal to interference ratio.

SNR signal to noise ratio.

SQNR signal to quantization noise ratio.

UWB ultra wideband.

VCO voltage controlled oscillator.

List of Symbols

β Chirp rate $^{1}/_{\text{Hz}}$.

$\mathcal{E}_b$ Energy of a signal representing a bit J.

F_s Sample frequency Hz.

I Discrete information stream in bits to be transmitted.

N_0 Power spectral density of white noise.

N_{fft} FFT order or number of bins.

N_Q Power spectral density of quantization noise.

P_i Interference power W.

T_b Bit period s.

T_c Chirp period s.

t_m Measurement window s.

T_s Sample period s.

W_c Chrip bandwidth Hz.

W_{fft} FFT bandwidth Hz.

W_{res} FFT resolution in $^{\text{Hz}}/_{\text{bin}}$.

Contents

CHAPTER 1

Introduction

This first chapter provides the reader with a brief overview of the aspects involved with digital communication devices, which serves to guide the reader to the context in which this work has been performed. This is followed by the assignment itself, how the thesis was performed and which requirements were taken into account.

1.1 Digital communication

The main goal of a digital communication device is to provide the means to exchange information. Current examples which are widely used include WiFi, Bluetooth and LTE/UMTS/GSM. These examples are actually a collection of different techniques performing different sets of tasks. This can be visualized by, for example, the OSI model, illustrated in Figure 1.1. Note that there are many different models, each with a different separation of tasks and layers. However, one may see some resemblance on a more abstract level. What is important to see here is that communication between devices consists of a number of tasks hidden behind these layers.

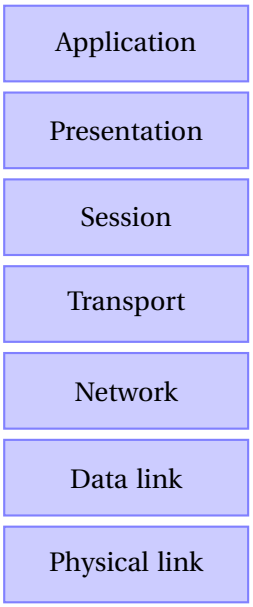


Figure 1.1: Basic OSI model

The techniques that are presented in this thesis belong to the lowest layer depicted, the physical layer, also known as the binary channel. It is, however, quite common in specifications, for example from IEEE, to specify both a physical layer and data link layer together, thus some overlap may be present. Figure 1.2 shows a block diagram of the (sub)tasks involved and their relations. The main task is to ensure that any data sequence presented at the input will be correctly reconstructed at the output.

Beginning at the input, a source encoder uses knowledge of the incoming data to reduce the amount of data to transmit. This may be done by using coding techniques based on probability functions. Such techniques and underlying theories are covered in the information theory sciences and are not of interest in this thesis. It's output is a sequence of bits, representing symbols. These bits are then passed to the channel encoder, which adds information to the stream such that the receiving end may perform error checking or correction. Finally the modulator maps the incoming bits onto physical waveforms. These waveforms may be lightwaves, electrical currents/voltages or, as is used in the thesis, radio waves. In all cases, the waveforms pass though some channel and are sensed by the receiver. At this point the inverse of the previous steps is executed in reverse order. Thus a demodulator transforms the waveforms to bits, the channel decoder checks for errors and may recover data where able and the source decoder provides decompression functionality. All to reconstruct the original transmitted data.

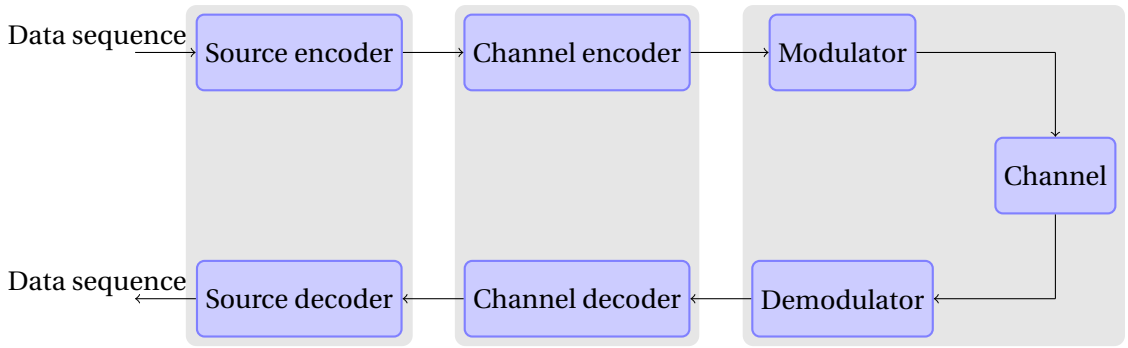


Figure 1.2: Basic layering of a communication system

Note that the demodulator only detects waveforms and maps these to bits with a certain probability. Where the channel decoder tries to verify the incoming stream of bits. One of the tasks of the data link layer is to perform a low level error correction. This is possible if, for example, the demodulator not only reports the bits received, but also their certainty level. This is called soft decision and is an example of how these layers may work together.

Since the main focus of this thesis lies on the physical layer, a more detailed diagram of the modulator and demodulator is drawn in Figure 1.3. Based on the incoming bits, appropriate message signals are selected. Concatenation of these signals form waveforms, which in turn may be up-converted for passband transmission. This is another term for adding a carrier signal. Thus a baseband signal is the information bearing signal without the carrier. Note that in these diagrams synchronization is not specified, but implicitly assumed.

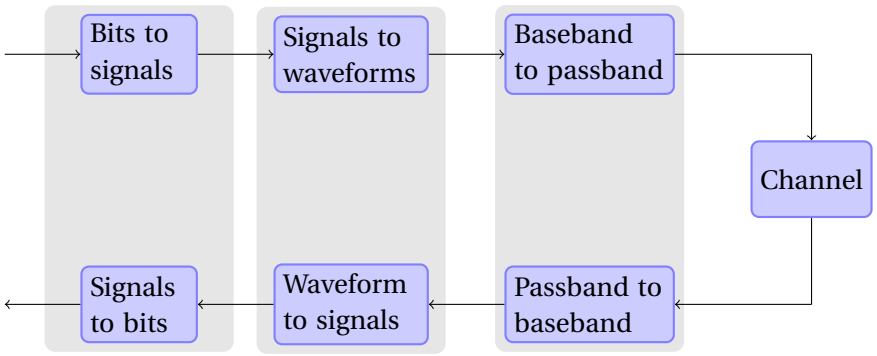


Figure 1.3: Functionality in the (de)modulator

1.2 Assignment

This assignment was performed as part of a low power radio transceiver which is to be robust against narrowband interference. One technique which is used to meet the low power and robustness requirements is so called chirped communications. This term implies that a chirp signal, a signal of which the frequency is a continuous function of time, is utilized as carrier signal, effectively sweeping through a specific frequency band.

The goal of this thesis is to develop a synchronization scheme for a chirped binary frequency shift-keying (BFSK) receiver. The receiver itself has already been designed to verify it's performance. This means that there is already an architecture available, which should be extended such that synchronization of the carrier is possible.

To achieve this goal the following subgoals are defined:

- *Formulate an algorithm to perform synchronization based on spectral properties*
  Initial analysis has already been performed on this subject, showing that mixing two chirp signals, for example one received and one from a local oscillator, results in a number of spectral components.

  Thus the first step is to formulate a mathematical description to link these components to the time difference between transmitter and receiver.

- *Perform a high level design of this algorithm*
  Following the mathematical description, a hardware implementation has to be designed. Thus functionality, components and relevant parameters need to be analyzed with the requirements in mind.

- *Determine the impact of this synchronization method on the communication's performance*
  One of the requirements, which is listed in the next section, describes the scenario in which the synchronization module is executed only once, to conserve power. This leads to the situation where synchronization errors will be present during the transmission and affect it. Thus, it is investigated what measure of synchronization error yields an acceptable loss of performance, indicated by bit error rate (BER) curves.

1.3 Requirements

During the thesis the following requirements have to be taken into account:

- *Synchronization should be performed within one chirp period*
  Synchronization is performed non-data aided, meaning a separate preamble is transmitted first such that synchronization may be achieved, followed by the actual data. This requirement limits the available time to synchronize and therefore also restricts some of the overhead of the transmission.

  In this context the chirp period is defined as the time required to transmit one up and one down chirp. Formal definitions are presented in section 2.3.

- *Low power consumption*
  As the receiver is required to operate with low power consumption, this also applies to the synchronization module. Linked with the previous requirement, the following scenario is used to assure this. By performing synchronization only once, when the transmission starts, and turning the module off when it's done, power requirements are reduced.

  When selecting components, this requirement has also been taken into account to reduce the required power further. A drawback is that this scenario does assume that any drift between the transmitter's and receiver's oscillators is negligible.

- *High accuracy*
  As the synchronization operation is executed only once, the first iteration must lead to a correct synchronization every time. This is commonly used in coarse-grain synchronization algorithms, which are followed by fine-grain synchronization. As fine-grain synchronization is skipped in this case, the requirement is loosened up to the point that it must be correct most of the time and otherwise the error must be within an acceptable region. This requirement is linked to the results of the 3rd subgoal.

1.4 Structure

The next chapter, Chapter 2, provides the background for chirped communication as is used during this thesis. The aim is to familiarize the reader with digital communication aspects, the chirp signal and how this chirp signal is used as a carrier signal. Some performance measurements are illustrated when the transmission is subjected to narrowband interference (NBI), to reproduce the results from (Dutta et al. [9]).

Chapter 3 treats the formulation of the synchronization scheme. Starting with a brief overview of the available literature, the spectral components due to mixing are described and mathematic expressions to link these to a time offset are derived. As there are multiple ways to transform this information into a time offset, a section is dedicated to formulating two detection methods, which are explored in the following chapter. This chapter ends with a review of using alternative compositions of chirp signals.

In the fourth chapter the algorithm is transformed into a high level design based on the required operations. To determine the best parameters, such as resolution, operating frequency and measurement window, the operations are replaced one by one from their ideal descriptions to a more realistic model. Continuous performance measurements are provided to support design choices.

Chapter 5 investigates the impact of a synchronization error on the communication performance. First, the effect of both frequency and time offset on BFSK is shown. Then the same is done for the chirp carrier, after which the last sections shows, via simulation, how this offset degrades the performance.

Finally Chapter 6 presents the conclusions and recommendations.

CHAPTER 2

Chirped communication

This chapter is an introduction to chirped communication, starting with the used modulation scheme, binary frequency shift-keying (BFSK), and analyzing the effect of a single interferer on it's performance. This is done with a mathematical model and verified via simulation. Before extending the model to chirped communication, the chirp signal is introduced together with a short background on it's usage. This chapter ends with an analysis of interference on the chirped communication model.

2.1 Introduction to binary frequency shift-keying

In this section the definition and properties of BFSK are briefly discussed to familiarize the reader with this modulation scheme. BFSK is the binary variant of frequency shift-keying (FSK), which is one of the three basic digital modulation schemes. It operates by manipulating the frequency of a carrier to represent different symbols. The other basic techniques are phase shift-keying (PSK) and amplitude shift-keying (ASK), which represent symbols by changing the phase or amplitude of a signal.

2.1.1 Definition

With FSK based modulation schemes each symbol is represented by a different frequency, while each frequency is orthogonal to the other possible frequencies. This is partly reflected in the definition of a message signal:

$$s_m(t) = \sqrt{\frac{2\mathscr{E}_b}{T_b}} \cos\left(2\pi f_m t\right) \quad 0 \leq t \leq T_b \tag{2.1}$$

Where $\mathscr{E}_b$ is the energy of the signal representing one bit, T_b the bit period and f_m the message's frequency, which may be written as:

$$f_m = f_o + m\Delta f \quad m \in \mathbb{N}, 0 \leq m < M \tag{2.2}$$

Where:

- f_o is an arbitrary offset frequency;
- Δf the frequency spacing;
- M the number of unique symbols also called alphabet size.

As FSK requires all message signals to be orthogonal, an expression for Δf is derived based on this requirement. Testing if two signals are orthogonal is performed by calculating their inner product (Equation 2.3) and must result in zero if the signals are not equal. If the signals to be tested are equal, the result will yield the energy of this signal. The expected result for FSK is given in Equation 2.4.

$$\langle s_m(t), s_n(t)\rangle = \int_{-\infty}^{\infty} s_m(t)s_n(t)dt \tag{2.3}$$

$$\langle s_m(t), s_n(t)\rangle_{FSK} = \begin{cases} 2\mathscr{E}_b & m = n \\ 0 & m \neq n \end{cases} \tag{2.4}$$

Substituting Equation 2.1 in Equation 2.3 for $s_m(t)$ and $s_n(t)$ the following is derived in Proakis and Salehi [30]:

$$\begin{aligned} \langle s_m(t), s_n(t)\rangle &= \frac{2\mathscr{E}_b}{T_b}\int_0^{T_b} \cos\left(2\pi f_m t\right)\cos\left(2\pi f_n t\right) dt \\ &= \frac{2\mathscr{E}_b \sin(2\pi T_b(m-n)\Delta f)}{2\pi T_b(m-n)\Delta f} \\ &= 2\mathscr{E}_b \text{sinc}(2T_b(m-n)\Delta f) \end{aligned} \tag{2.5}$$

For Equation 2.5 to lead to the required result (Equation 2.4), the sinc function must return a one when $m = n$ and a zero otherwise, which occurs when it's argument is zero or an integer. Thus the frequency separation should be a multiple of $\Delta f = \frac{1}{2T_b}$.

Sunde's FSK

A special case of FSK is Sunde's FSK (Haykin [16]), which is a form of continuous phase modulation (CPM) with the exception that no form of memory is used. A modulation scheme is a CPM when it's phase is continuous over time. In general this requires the knowledge of previous symbols to apply a differential increase or decrease of the transmitted signal's phase.

With Sunde's FSK, the frequencies are selected by using the following formula:

$$f_m = \frac{n_c + m}{T_b} \quad n_c \in \mathbb{N} \tag{2.6}$$

n_c being an integer number of offsets. This way of frequency selection guarantees a continuous phase by making sure that each message's frequency has a natural number of periods within the bit interval. Additionally, this selection criterium is valid for both coherent and non-coherent detection (Proakis and Salehi [30]).

2.1.2 Properties

Spectrum

To determine the spectrum of BFSK the definition of the message signal (Equation 2.1) is rewritten:

$$s_m(t) = \sqrt{\frac{2\mathscr{E}_b}{T_b}}\cos\left(\frac{\pi t}{T_b}\right)\cos\left(2\pi f_c t\right) \pm \sqrt{\frac{2\mathscr{E}_b}{T_b}}\sin\left(\frac{\pi t}{T_b}\right)\sin\left(2\pi f_c t\right) \tag{2.7}$$

Here f_c represents the carrier or center frequency, the mean of both message frequencies. The $\pm$ sign depends on the transmitted symbol being either zero or one. By splitting the original message signal it becomes clear that it consists of an in-phase component, independent of the transmitted symbols, and a quadrature component whose sign does depend on the transmission.

Transforming both components separately as in (Simon [34]) leads to the spectral representation:

$$S_{BFSK}(f) = \frac{\mathscr{E}_b}{2T_b}\left[\delta(f-\frac{1}{2T_b})+\delta(f+\frac{1}{2T_b})\right]+\frac{8\mathscr{E}_b\cos^2(\pi T_b f)}{\pi^2(4T_b^2f^2-1)^2} \tag{2.8}$$

From which the scaled Dirac pulses represent the in-phase part and the scaled cosine pulse the quadrature part. This is illustrated in Figure 2.1, for $\mathscr{E}_b = 1$ and $T_b = 0.5$, which leads to the message signals being placed at 2 Hz and 4 Hz, or an f_c of 3 Hz.

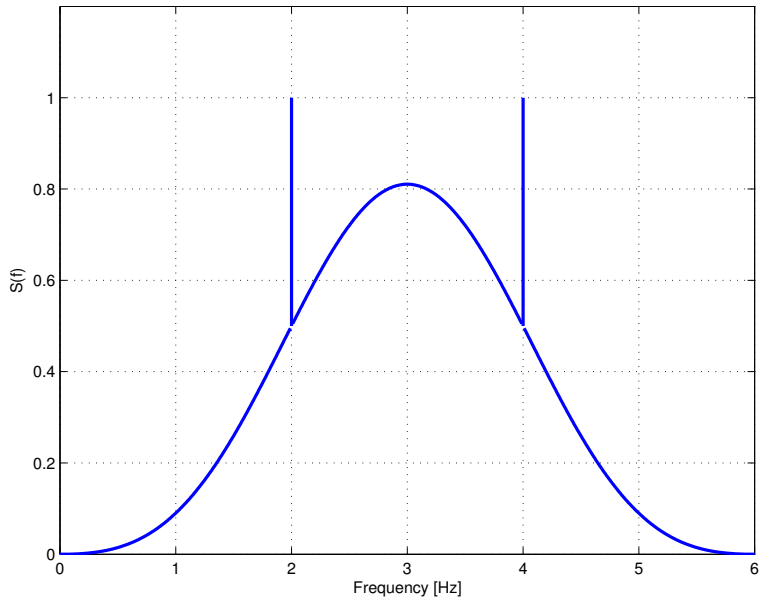


Figure 2.1: Power spectrum for BFSK

Bit error rate

For the theoretical bit error rate (BER) approximation, two expressions are relevant when using an additive white gaussian noise (AWGN) channel, depending on the receiver's architecture (Proakis and Salehi [30]). Namely if the receiver is coherent, phase synchronization is performed before demodulation of the received signal. An example of this type is the correlator receiver. The BER expression is:

$$P_{e,coherent} = Q\left(\sqrt{\frac{\mathscr{E}_b}{N_0}}\right) \tag{2.9}$$

Here $Q(\cdot)$ is the q function for Gaussian random variables and N_0 the white noise power due to the channel.

Alternatively, a matched filter implementation may be used, which does not require synchronization and thus is non-coherent.

$$P_{e,non-coherent} = \frac{1}{2}e^{-\frac{\mathscr{E}_b}{2N_0}} \tag{2.10}$$

Both expressions are plotted in Figure 2.2.

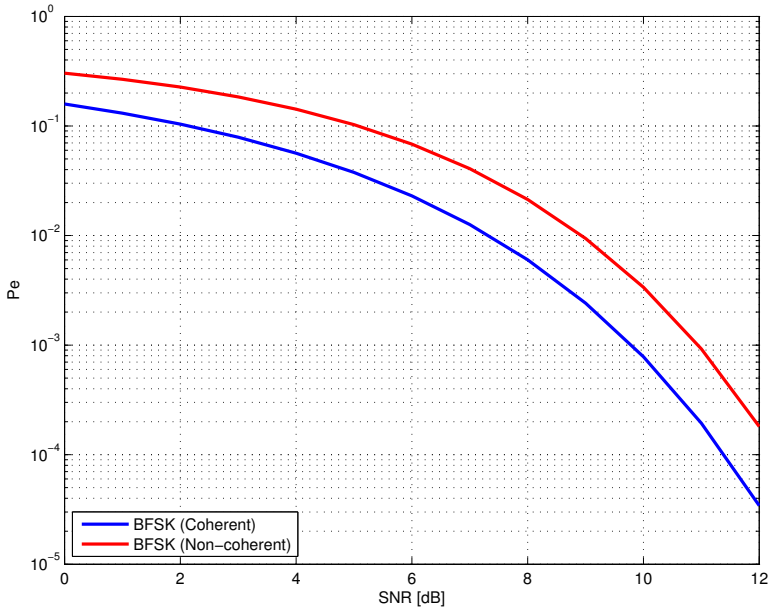


Figure 2.2: BER for coherent and non-coherent BFSK

2.2 Communication model

Now that the modulation technique has been introduced, it is of interest to investigate how these techniques perform when interference is received. To answer this question, a mathematical model is derived such that BER curves for various signal to interference ratio (SIR) can be approximated. To verify this model a simulation model is made in MATLAB.

First the model is introduced and it's components are described in a general form, followed by the specific implementation for BFSK. Next the mathematical model and simulation results are verified.

2.2.1 Components

Figure 2.3 shows the model which is used, representing a baseband transceiver. It consists of a transmitter, a AWGN channel with an interferer and a receiver. It should be noted that synchronization between the transmitter and receiver is assumed.

Transmitter

The input of the model is a stream I, which is assumed to be a discrete sequence containing uniformly distributed symbols of the value zero or one. Now assume that $\mathbf{I}[n]$ represents the n-th symbol and $s_m(t)$ the message signal corresponding to that symbol. Then $s_{I[n]}(t)$ is the message signal selected by the current symbol. As these message signals are only defined

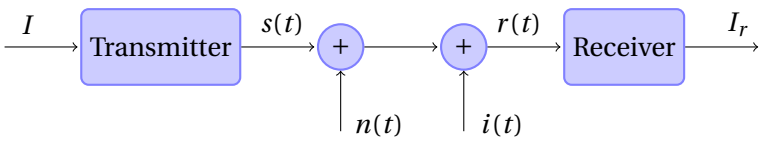


Figure 2.3: Overview of the baseband simulation model

during the interval $0 \leq t \leq T_b$, such as in Equation 2.1, the transmitted signal is defined as a concatenation of shifted message signals:

$$s(t) = \sum_n s_{I[n]}(t - nT_b) \tag{2.11}$$

Channel

The signal $s(t)$ is then passed through an AWGN channel, represented by the addition of $n(t)$ and has the following properties:

$$E[n(t)] = 0 \tag{2.12}$$

$$VAR[n(t)] = \frac{N_0}{2} \tag{2.13}$$

Optionally a continuous wave interference signal may be added to the channel:

$$i(t) = A_i \cos(2\pi f_i t + \phi_i) \tag{2.14}$$

with a predefined frequency f_i and phase ϕ_i as a random variable, uniformly distributed between $(0, 2\pi]$. The signal has an constant amplitude A_i, defined to be $\sqrt{2P_i}$ and P_i being the total interference power. With this definition properties such as the expected value and variance (van Etten [39]) are calculated:

$$\begin{aligned} E[i(t)] &= \int_0^{2\pi} \phi I(t,\phi) d\phi \\ &= 0 \end{aligned} \tag{2.15}$$

$$\begin{aligned} VAR[i(t)] &= E[i^2(t)] - E^2[i(t)] \\ &= \frac{1}{2} A_i^2 - 0 = P_i \end{aligned} \tag{2.16}$$

Receiver

After passing the transmitted signal through the channel, the receiver's input equals:

$$r(t) = s(t) + i(t) + n(t) \tag{2.17}$$

This signal is passed to the demodulator to generate a stream of demodulated symbols (I_r), which are compared with the input to determine the BER.

Demodulation

Demodulation is performed using a correlation receiver architecture, see Figure 2.4, implementing the optimal detection rule (Equation 2.18 and it's analog counterpart Equation 2.19) as stated in (Proakis and Salehi [30]). This rule is valid since it is assumed that each symbol is equiprobable and has an equal energy compared to other symbols.

$$\hat{m} = \arg \max_{1 \leq m \leq M} \mathbf{r} \cdot \mathbf{s}_m \tag{2.18}$$

$$\hat{m} = \arg \max_{1 \leq m \leq M} \int_{T_b} r(t) s_m(t) dt \tag{2.19}$$

Thus each branch correlates the received signal with a message signal over one bit period, which is expressed as:

$$\mathbf{r}_n = \int_0^{T_b} r(t) s_n(t) dt \quad n = 0, 1, ...M \tag{2.20}$$

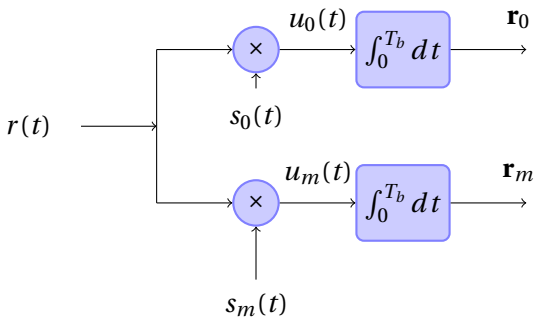


Figure 2.4: Schematic overview of a correlator demodulator

Here it is implicitly assumed that the integrators act like zero-order hold blocks, producing the discrete stream $\mathbf{r}_0$ and $\mathbf{r}_1$. Both streams are fed to a decision block which determines the received symbol.

Using Equation 2.17, the previous expression may be rewritten to:

$$\begin{aligned}\mathbf{r}_n &= \int_0^{T_b} s(t)s_n(t)dt + \int_0^{T_b} i(t)s_n(t)dt + \int_0^{T_b} n(t)s_n(t)dt \\ &= \mathbf{v}_n^1 + \mathbf{v}_n^2 + \mathbf{v}_n^3\end{aligned} \tag{2.21}$$

The subexpressions $\mathbf{v}_n^1$, $\mathbf{v}_n^2$ and $\mathbf{v}_n^3$ are again discrete streams, each representing the influence of a different part of the received signal on the receiver's output.

2.2.2 Modeling the BFSK receiver

Continuing with the subexpressions defined in Equation 2.21, the next paragraphs treat the solving of them for a BFSK receiver.

Message signal

If the receiver is configured to detect BFSK signals and receives such, the first term of the integral in Equation 2.21 becomes:

$$\mathbf{v}_n^1 = \int_0^{T_b} A_m \cos(2\pi f_m t) A_n \cos(2\pi f_n t) dt$$

Now if the message signals are equal, the receiver branch will output:

$$\mathbf{v}_n^1 = \mathscr{E}_b \quad m = n \tag{2.22}$$

For the case that $m \neq n$:

$$\begin{aligned}\mathbf{v}_{n,1}^1 &= \frac{A_m A_n}{2} \int_0^{T_b} \cos(2\pi(f_m - f_n)t) + \cos(2\pi(f_m + f_n)t) dt \\ &= \frac{A_m A_n}{4\pi} \left[\frac{\sin(2\pi(f_m - f_n)T_b)}{(f_m - f_n)} \right] \\ &= 0\end{aligned} \tag{2.23}$$

Keeping in mind that both f_m and f_n both represent a different message and are an integer multiple of $\frac{1}{T_b}$ (Equation 2.6), it can be seen that the above expression will be zero, independent of their actual frequencies.

Interferer

To determine the effect of the interferer on the receiver, we first look at the signal after mixing with a message signal. This corresponds to the point u_m for $m = 0, 1, ...M$ in Figure 2.4, and is defined as:

$$u_m(t) = i(t)s_m(t) \tag{2.24}$$

For further calculations, power contributions are analyzed in the frequency-domain with power spectral density (PSD) functions. This is accomplished by using a method described in (Haykin [16]) and (van Etten [39]) to calculate the effect of mixing a deterministic signal, the message signal, with a sinusoidal process,the interferer. The first step is to calculate the PSD of the mixer result as a function of the PSD of the deterministic signal.

Based on the properties of the interferer (Equation 2.15 and Equation 2.16) the process is called wide-sense stationary, allowing the use of the Wiener-Khinchin relation (Equation 2.25) to determine the PSD of the mixer output.

$$S_{XX}(f) = \int_{-\infty}^{\infty} R_{XX}(\tau)e^{-j2\pi f\tau}d\tau \tag{2.25}$$

To do so, the autocorrelation of the output is required. The definition of the autocorrelation is given in Equation 2.26, however as the interference is wide-sense stationary, the autocorrelation is independent of absolute time, thus Equation 2.27 is used.

$$R_{XX}(t, t+\tau) = E[X(t)X(t+\tau)] \tag{2.26}$$
$$R_{XX}(\tau) = E[X(t)X(t+\tau)] \tag{2.27}$$

As stated in (Haykin [16]) and (van Etten [39]), mixing a sinusoidal process with a deterministic signal leads to the following autocorrelation function:

$$R_{UU}(\tau) = R_{II}(\tau)R_{SS}(\tau) \tag{2.28}$$

in which R_{II} and R_{SS} are the autocorrelation functions of the interference and the current message signal. Calculating R_{II} leads to the following intermediate result:

$$R_{II}(\tau) = P_i\cos(2\pi f_i\tau) \tag{2.29}$$
$$R_{UU}(\tau) = P_i\cos(2\pi f_i\tau)R_{SS}(\tau)$$

which can already be transformed into the frequency-domain (Equation 2.30), independent of the message signal's properties. This completes the first step.

$$S_{UU}(f) = \frac{P_i}{2}\left[S_{SS}(f-f_i) + S_{SS}(f+f_i)\right] \tag{2.30}$$

The second step is calculating the PSD of the message signal and substituting the result in Equation 2.30.

$$s_m(t) = \sqrt{2P_b}\cos(2\pi f_m t)$$
$$S_m(f) = \sqrt{\frac{P_b}{2}}\left[\delta(f-f_m) + \delta(f+f_m)\right]$$
$$S_{SS}(f) = \frac{P_b}{2}\left[\delta(f-f_m) + \delta(f+f_m)\right] \tag{2.31}$$

Leading to the final result:

$$S_{UU}(f) = \frac{P_i P_b}{4}\left[\delta(f - f_i - f_m) + \delta(f - f_i + f_m) + \delta(f + f_i - f_m) + \delta(f + f_i + f_m)\right] \tag{2.32}$$

The above expression therefore represents the signal after mixing and before integration. As integration in the time domain serves as a low pass filter in the frequency-domain, see Equation 2.33, the first and last delta pulse will not have a significant contribution to the total power when the interference is close to the message signals. In such cases the power contribution of the interference can be approximated by Equation 2.34.

$$\left(\mathscr{F}\int_{-\infty}^{t} f(\tau)d\tau\right)(f) = \frac{-jF(f)}{2\pi f} \tag{2.33}$$

$$\mathbf{v}_n^2 = \begin{cases} \frac{P_i P_b}{2} & f_i \approx f_m \\ 0 & \text{othwerwise} \end{cases} \tag{2.34}$$

Additive white gaussian noise

Finally, the term $\mathbf{v}_n^3$ represents the white noise component. The result is again a Gaussian process with the properties:

$$E[\mathbf{v}_n^3] = 0 \tag{2.35}$$

$$\sigma^2_{\mathbf{v}_n^3} = \frac{N_0 P_b}{2} \tag{2.36}$$

2.2.3 Effect of interference on the bit error rate

Now that the power contributions of the three input components are known, we look at how these influence the BER of the system. If only AWGN is present, the BER curve for coherent BFSK is expressed as in Equation 2.9, reprinted here.

$$P_{e,coherent} = Q\left(\sqrt{\frac{\mathscr{E}_b}{N_0}}\right) \tag{2.9}$$

This expression is based on the signal to noise ratio (SNR), $\frac{\mathscr{E}_b}{N_0}$, to represent the channel conditions. When introducing interference, it is common to extend this definition to the signal to interference plus noise ratio (SINR) (Win et al. [41]):

$$\text{SINR} = \frac{P_b}{N_0 + \sum_K P_{i,k}} \tag{2.37}$$

where $P_{i,k}$ represents the power of the k-th interferer as sensed by the receiver. Substituting the result of Equation 2.34 for $P_{i,k}$, the expected BER for our communication model becomes:

$$\begin{aligned} P_e &= Q\left(\sqrt{\text{SINR}}\right) \\ &= Q\left(\sqrt{\frac{P_b}{N_0 + \sum_K \frac{P_{i,k}}{2}}}\right) \end{aligned} \tag{2.38}$$

Another common definition is the signal to interference ratio:

$$\text{SIR} = \frac{P_b}{\sum_K \frac{P_{i,k}}{2}} \tag{2.39}$$

2.2.4 Simulation results

To verify the previously stated expressions, a simulation model based on Figure 2.3 is made in MATLAB, implementing a BFSK transceiver. The message signals are selected using Sunde's FSK formula (Equation 2.6). Then a single tone interferer is placed on one of the message signals, effectively jamming the transmission. Note that it does not matter which of the two message signals is jammed, as both are equiprobable.

In Figure 2.5 the power spectrum of the interference signal is plotted as it passes through the receiver. It shows the spectra of the interference and the message signals, all with equal power, thus a SIR of one. Also plotted is the signal after mixing (corresponding to Equation 2.32), followed by the signal after integration (Equation 2.34).

The interference tone is modeled to have the same period as the symbol rate, during which the phase offset of the tone does not change. By simulating multiple symbols and varying the phase offset of the interference for each symbol duration, the simulation averages the effect of each individual offset. This also leads to the spectrum in Figure 2.5, which shows resemblance to a PSK spectral image. When simulating over one symbol period the resemblance with the mathematical expressions becomes clear.

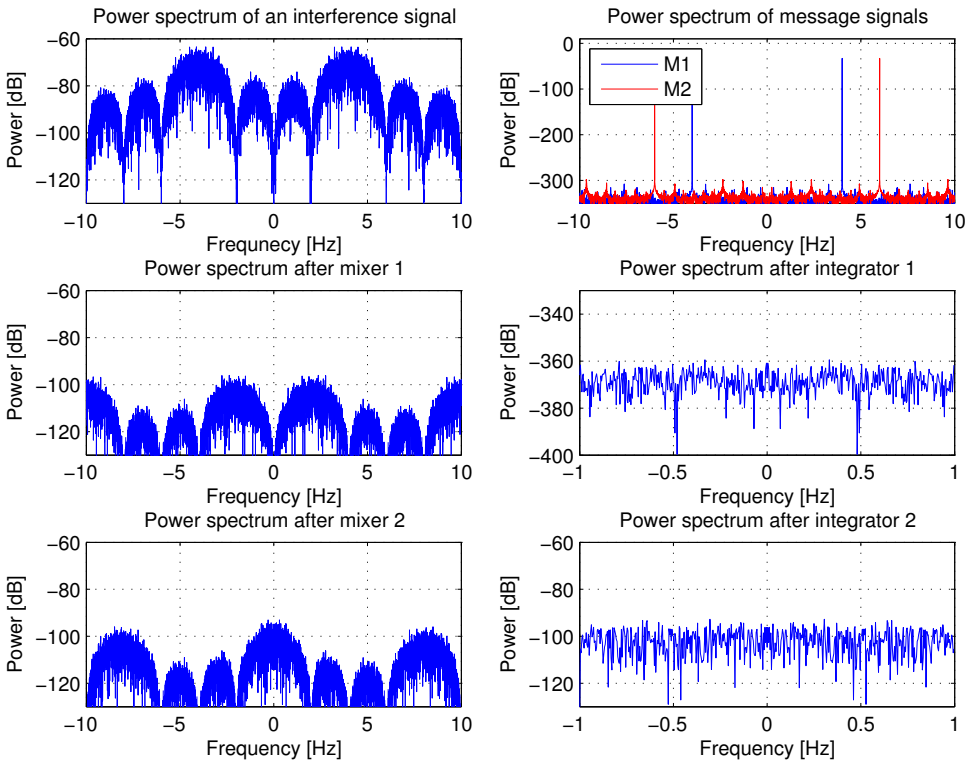


Figure 2.5: Power spectra of a interference signal, with multiple phase offsets, at various stages of a BFSK receiver.

Figure 2.6 depicts the BER curves for a number of SIRs, together with the approximation in Equation 2.38 and the theoretical curve as reference. As can be seen, there is an error with this approximation, which becomes clearly noticeable when the SNR becomes higher then eight. This is due to the fact that the approximation is based on the assumption that the interference is wideband, comparable to the white noise component. This, however, is not the case when a continuous wave (CW) interfering signal is received. Since an exact evaluation is out of the scope of this work, Equation 2.38 will be used as an approximation for a limited range of SNRs.

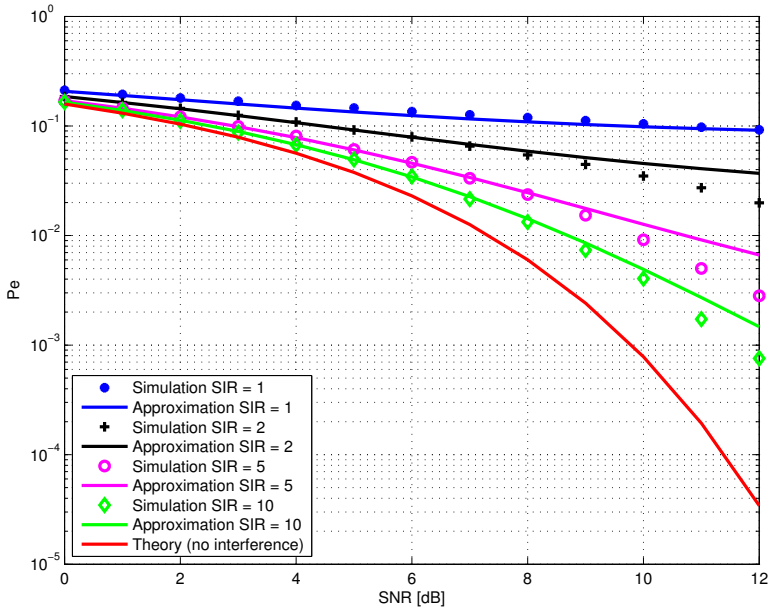


Figure 2.6: BER for BFSK with interference on one message signal for with various SIR

2.3 Introducing the chirp signal

This section serves as an introduction to the chirp signal by presenting it's definition, basic properties and a short summary of the usage of this type of signal. The aim is to familiarize the reader with this type of signal which is used in the remainder of this thesis.

2.3.1 Definition

A chirp signal is generally defined as a signal of which the frequency changes over time in a continuous manner. Given the following expression:

$$c(t) = \sqrt{2}\cos(2\pi f_c t + \phi(t)) \tag{2.40}$$

the signal becomes a chirp signal when $\phi(t)$ is a continuous non-linear function. To describe the frequency change it is common to look at the instantaneous frequency:

$$f_i = f_0 + \frac{d\phi(t)}{dt} \tag{2.41}$$

If f_i is linear and $\phi(t)$ therefore quadratic, the chirp is called a linear frequency sweep (LFS) signal. Unless otherwise noted, the term chirp implies a LFS signal for the remainder of this thesis. Then Equation 2.40 and Equation 2.41 become:

$$c(t) = \sqrt{2P_c}\cos(2\pi(f_c + \frac{\beta t}{2})t) \quad -\frac{T_c}{2} \leq t \leq \frac{T_c}{2} \tag{2.42}$$

$$f_i(t) = f_c + \beta t \tag{2.43}$$

Here

- P_c is the power of the signal in W;
- f_c is the center frequency in Hz;
- β the chirp rate in $^{\text{Hz}}/_{\text{sec}}$;
- and T_c the chirp period in sec.

Additionally, the sign of the chirp rate determines the direction of the sweep. If $\beta < 0$, the frequency decreases over time and the signal is classified as a down chirp. Similarly if $\beta > 0$, the frequency increases over time and the signal is called an up chirp. An example of an up chirp signal is depicted in Figure 2.7.

The frequency-domain representation is expressed as in (Pinkney [28]):

$$S_{chirp}(f) = \sqrt{\frac{1}{8\beta}} e^{-j\frac{\pi}{\beta}(f-f_c)^2} (C(x_1) + C(x_2) + jS(x_1) + jS(x_2)) \tag{2.44}$$

Where $C(\cdot)$ and $S(\cdot)$ are Fresnel integrals with arguments:

$$x_1 = \sqrt{\frac{2}{\beta}} \left(\frac{\beta}{2} + (f_c - f) \right) \tag{2.45}$$

$$x_2 = \sqrt{\frac{2}{\beta}} \left(\frac{\beta}{2} - (f_c - f) \right) \tag{2.46}$$

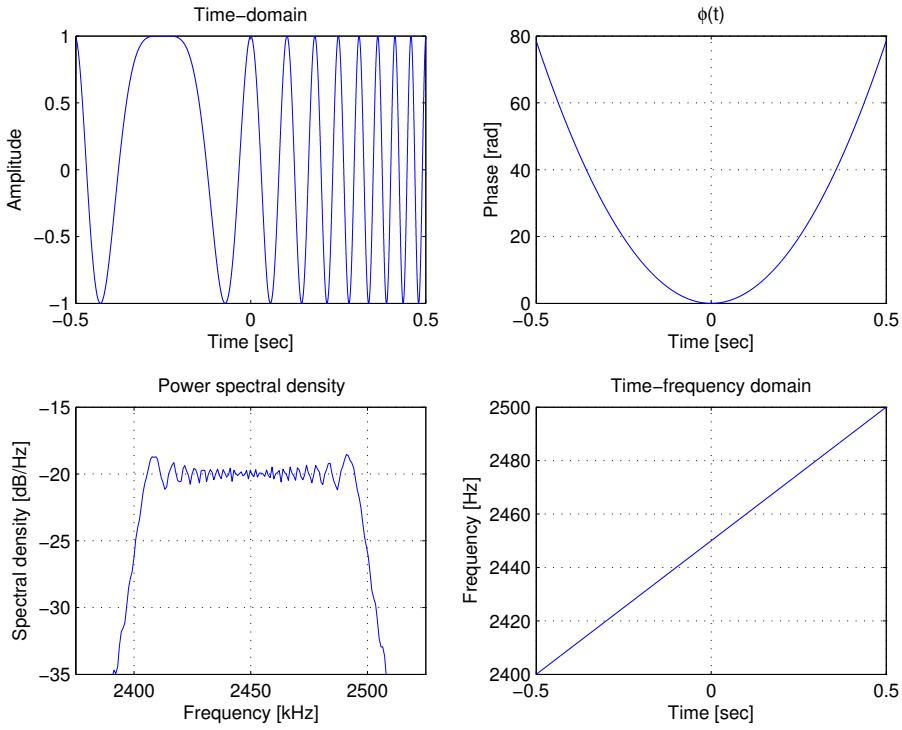


Figure 2.7: A chirp signal in various domains

2.3.2 Properties

The following paragraph briefly states a number of properties relevant to this thesis.

Energy

The energy in one chirp period is (Berni and Gregg [3]):

$$\begin{aligned} \mathscr{E}_c &= \int_{-\frac{T_c}{2}}^{\frac{T_c}{2}} |c(t)|^2 dt \\ &= P_c T_c \end{aligned} \tag{2.47}$$

Bandwidth

The bandwidth is equal to:

$$W_c = \beta T_c \tag{2.48}$$

Time-bandwidth product

Another commonly used property to compare techniques is the time-bandwidth product. This is approximated with (Berni and Gregg [3]):

$$BT = W_c \times T_c \tag{2.49}$$

$$\approx \beta T_c^2 \tag{2.50}$$

This time-bandwidth product is actually the processing gain for chirped communication systems (Pinkney [28]). It shows the interesting property that it does not depend on the symbol rate, such as with other spread spectrum techniques. Additionally, this property is a good measure for the resistance to jamming and (narrowband) noise in the receiver (Roberton and Brown [33]).

2.3.3 Usage of chirp signals

Chirp signals are used in a variety of applications, including RADAR, SONAR, both wireless and underwater communication, channel sounding etc. One aspect which makes chirp signals interesting for communication purposes is it's property of not being affected by multi path interference (He et al. [17], Tsai and Chang [38]).

In (Pinkney [28]) one can find an interesting history about the use of chirp signals in communication systems from the sixties to the late nineties. Therefore I would recommend reading that work if one is interested. This section notes some of the mentioned research and extends to more current work.

Chirped communication was originally proposed in the sixties and was based on a binary orthogonal keying (BOK) scheme, using up and down chirps to transmit zeros and ones. Others continued from this work to create complete transmission systems, such as (Gott and Newsome [13]). Also notable is the work in (Berni and Gregg [3]) where FSK, PSK and chirped BOK are compared from a mathematical point of view. It is concluded that while chirped BOK has superior characteristics in partially coherent reception and fading channels, it is not seen as a practical solution due to the need of precise phase synchronization.

This need for synchronization can be avoided by using surface acoustic wave (SAW) filters to generate and filter out chirp carrier signals. Combining a modulation technique and utilizing the chirp signal as carrier are called direct modulation (DM) systems. Examples are (Gugler

et al. [14], Springer et al. [35], Pinkney et al. [29]), which combine the use of SAW filters with differential quadrature phase shift-keying (DQPSK) for non-coherent wireless transmission.

With IEEE's amendment to the IEEE Std 802.15.4 (iee [1]), chirp spread spectrum is officially proposed as a possible physical layer for low-rate wireless personal area networks. Interestingly, this has lead to renewed interest in chirped BOK in a multi-user environment (Wang et al. [40]) or using fractional Fourier transform (FRFT) to detect the distinct chirps (Huang et al. [20]).

2.4 Chirped communication model

In the previous sections the chirp signal and a simulation model illustrating the effect of interference on BFSK have been introduced. This section treats the combination of the two by using a chirp signal as a carrier signal for the modulated signal, thereby spreading the information signal over a larger bandwidth before it passes through the channel. This with the aim to reduce the influence of any interference present in the channel on the receiver.

2.4.1 Components

Figure 2.8 shows the model as used in section 2.2, extended with a chirp signal generator. Additionally, the interference is changed from a single tone to a multi-tone signal such that it also is spread out over a specific frequency band.

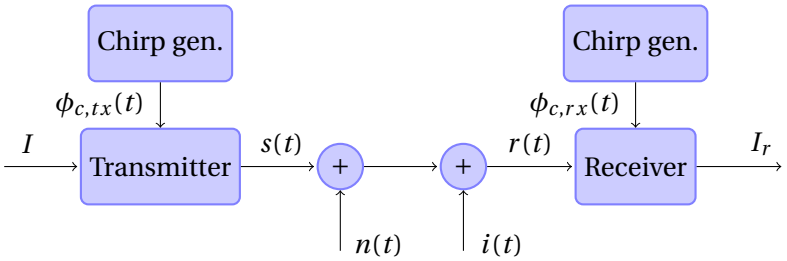


Figure 2.8: Overview of the chirped communication model

Transmitter

The transmitter, Figure 2.9, is modified to mix the modulated signal with the chirp signal via a voltage controlled oscillator (VCO). Unlike the transmitter as has been used previously, this one uses the phase information which represent the message signals (ϕ_m). Thus for each symbol:

$$s(t) = A_c \cos(\phi_c(t) + \phi_m(t)) \tag{2.51}$$

$$= A_c \cos(2\pi f_c t + \pi\beta t^2 + \phi_m(t)) \tag{2.52}$$

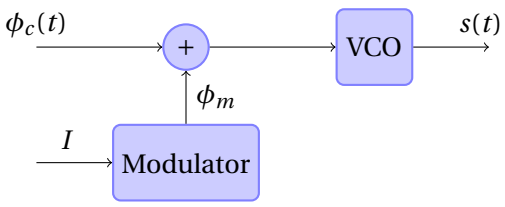


Figure 2.9: Schematic overview of the transmitter

The chirp signal will ideally sweep through a specified frequency band. Thus over a single chirp period the PSD will intuitively look like a rectangular block. In reality this does not hold exactly, as can be seen in Figure 2.7.

However, the time-bandwidth product is related to the amount of energy present inside the defined frequency band. For time-bandwidth products from 100 and above, approximately 98% to 99% of the signal's energy lies within the desired bandwidth (Richards et al. [32]). Therefore a time-bandwidth product of at least 100 will be used such that the approximation in Equation 2.53 can be used, under the assumption that $A_c = \sqrt{2P_c}$. Figure 2.10 provides a visual comparison of the approximation and the actual chirp spectrum.

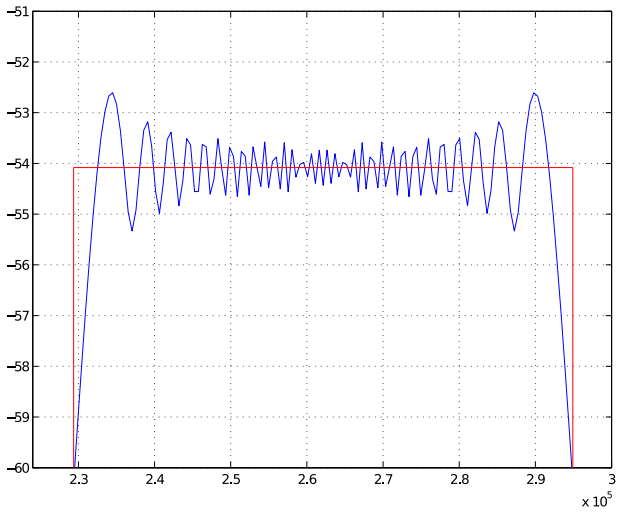


Figure 2.10: Amplitude spectrum of a chirp signal (blue), $f_c = 262\,\text{kHz}$, BW = 64 kHz, BT = 128, and its approximation (red).

$$S_c(f) = \begin{cases} \frac{P_c}{W_c} & f_c - \frac{W_c}{2} \leq f \leq f_c + \frac{W_c}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.53}$$

Channel

The channel consists of AWGN and an interference source. This interference source however is now modeled as a signal which occupies a specific frequency band:

$$S_i(f) = \begin{cases} \frac{P_i}{2W_i} & f_i - \frac{W_i}{2} \leq |f| \leq f_i + \frac{W_i}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.54}$$

For details on how this is modeled in MATLAB see Appendix B.

Receiver

Equivalent to the transmitter, the receiver has been extended with a mixer to remove the chirp carrier from the received signal, see Figure 2.11. However, another change has been made, the mixers for both the chirp signal and the correlation of message signals are extended to complex mixers. This is done to match the model with the BFSK receiver for which the synchronization algorithm is intended.

The input signal, $r(t)$, expressed in it's spectral components has the following form:

$$S_r(f) = S_s(f) + S_i(f) + \frac{N_0}{2} \tag{2.55}$$

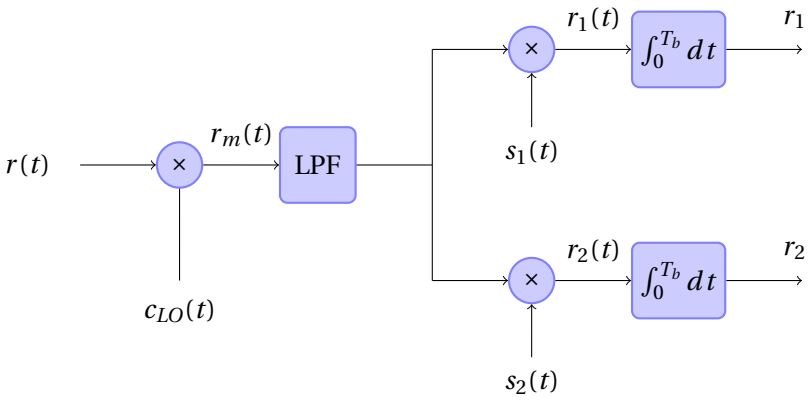


Figure 2.11: Schematic overview of the receiver

Down-mixing

Assuming perfect synchronization, down-mixing with the local chirp signal, Equation 2.56, will remove the carrier from the message signal and shift any other signal components.

$$c_{LO}(t) = A_{LO} e^{j(2\pi f_c t + \pi\beta t^2)} \tag{2.56}$$

$$\begin{aligned} r_m(t) &= r(t)c_{LO}(t) \\ &= s(t)c_{LO}(t) + i(t)c_{LO}(t) + n(t)c_{LO}(t) \\ &= r_m^1(t) + r_m^2(t) + r_m^3(t) \end{aligned} \tag{2.57}$$

For the message signals:

$$\begin{aligned} r_m^1(t) &= s(t)c_{LO}(t) \\ &= A_c A_{LO} \cos(2\pi f_c t + \pi\beta t^2 + \phi_m(t)) e^{j(2\pi f_c t + \pi\beta t^2)} \\ &= \frac{A_c A_{LO}}{2} \left[e^{j(2\pi f_c t + \pi\beta t^2 + \phi_m(t))} + e^{-j(2\pi f_c t + \pi\beta t^2 + \phi_m(t))} \right] e^{j(2\pi f_c t + \pi\beta t^2)} \\ &= \frac{A_c A_{LO}}{2} e^{j(4\pi f_c t + 2\pi\beta t^2 + \phi_m(t))} + \frac{A_c A_{LO}}{2} e^{-j(\phi_m(t))} \end{aligned} \tag{2.58}$$

Showing that the mixing results in a upscaled version of the received signal and the modulation component. Then this result is passed through a lowpass filter:

$$r_{lpf}^1(t) = \frac{A_c A_{LO}}{2} e^{-j(\phi_m(t))} \tag{2.59}$$

Showing that the only residual effect of the chirp carrier is an amplitude scaling of $A_{LO}/2$.

Down-mixing the interference component, $r_m^2(t)$, is analyzed with the spectral estimations as specified in Equation 2.53 and Equation 2.54. This leads to the following convolution integral to evaluate:

$$S_{ci}(f) = \int_{-\infty}^{\infty} S_c(g) S_i(f-g) dg \tag{2.60}$$

$$= S_{ci,p}(f) + S_{ci,n}(f) \tag{2.61}$$

$S_{ci,p}(f)$ and $S_{ci,n}(f)$ represent the contribution of the positive and negative frequency side of the interference spectrum. Equation 2.62 shows one of the two functions. It's negative counterpart has an equal structure, with opposite signs for the interference related terms.

$$S_{ci,p}(f) = \begin{cases} \frac{P_c P_i}{2W_c} & f_c - \frac{W_c}{2} + f_i - \frac{W_i}{2} > f > f_c - \frac{W_c}{2} + f_i + \frac{W_i}{2} \\ \frac{P_c P_i}{2W_c W_i}(W_c - W_i) & f_c - \frac{W_c}{2} + f_i + \frac{W_i}{2} > f > f_c + \frac{W_c}{2} + f_i - \frac{W_i}{2} \\ \frac{P_c P_i}{2W_c} & f_c + \frac{W_c}{2} + f_i - \frac{W_i}{2} > f > f_c + \frac{W_c}{2} + f_i + \frac{W_i}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.62}$$

The expression is split into two parts, both indicating two distinct spectral components present in the resulting signal. Both form a rectangle at the positive and negative frequency domain. Any overlap between the spectrum of the chirp and interference signal will lead to an overlap between these shapes, which is illustrated in Figure 2.12. Due to the position of this shape, it overlaps with the bandwidth of the message signals, thus contributing to the noise sensed by the correlator receiver.

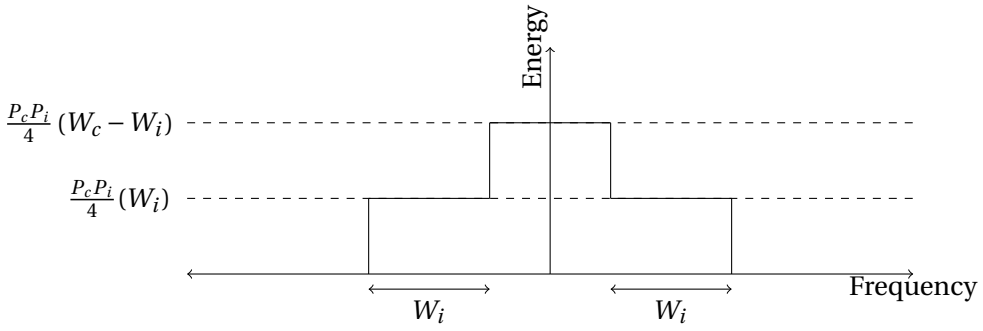


Figure 2.12: Down mixed interference signal when it overlaps with the chirp carrier.

2.4.2 BFSK implementation

After down-mixing, demodulation is performed and again the power contributions are split into three components. For reference Equation 2.21 is reprinted:

$$\begin{aligned} \mathbf{r}_n &= \int_0^{T_b} r_m^1(t) s_n(t)\,dt + \int_0^{T_b} r_m^2(t) s_n(t)\,dt + \int_0^{T_b} r_m^3(t) s_n(t)\,dt \\ &= \mathbf{v}_n^1 + \mathbf{v}_n^2 + \mathbf{v}_n^3 \end{aligned}$$

Message signal

The message signal after down mixing and filtering, as given in Equation 2.59 is:

$$r_{m,1}(t) = \frac{A_m A_c^2}{2}\cos(2\pi f_m t)$$

Following the same method as used in subsection 2.2.2, the following result is derived:

$$\mathbf{v}_n^1 = \begin{cases} \mathscr{E}_b A_c^2 & m = n \\ 0 & m \neq n \end{cases} \tag{2.63}$$

As could be expected based on the previous results, the result is equal to standard BFSK with the addition of a power scaling due to the chirp signal.

Interference

The power contribution of the interference depends on the positioning of the chirp carrier and the interference in the frequency domain. When the carrier is not in the region of the interference, the power contribution will be zero. Once the signals overlap, the power contribution will increase as is illustrated in Figure 2.12.

Therefore two scenarios can occur. The first is when there is no overlap and the power contribution is zero. The second scenario is when the chirp passes through the interference during which an additional power contribution of $\frac{P_i P_c}{2}$ is present. To compensate for any transition effects, the average power is determined and considered to be the input power over the complete region.

Additive white gaussian noise

The final power contribution is of the white noise due to the channel. As this noise is uncorrelated to the chirp signal, it's contribution is equal to the previous result (Equation 2.36):

$$E[\mathbf{v}_n^3] = 0 \tag{2.64}$$

$$\sigma^2_{\mathbf{v}_n^3} = \frac{N_0 P_b}{2} \tag{2.65}$$

2.5 Simulation results

Figure 2.13 shows the result of mixing a chirp signal with an interference band. Additionally, the approximation curve is plotted over the result, showing that the average power in this band is approximated quite well. This supports the use of the approximations in favor of the original signal descriptions to approximate their power contributions

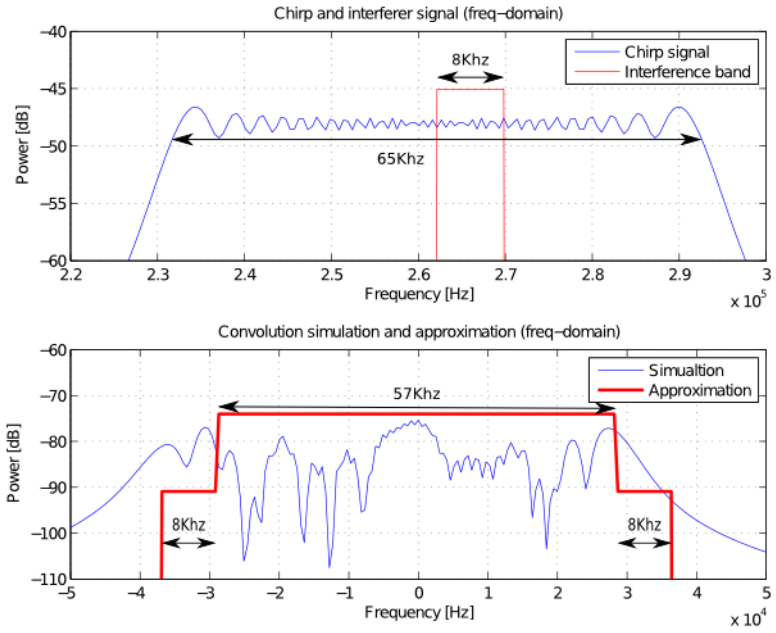


Figure 2.13: Convolution simulated (blue) and approximation (red) with some overlap present

The BER approximation for this model, see Equation 2.66, is split into two sections, for the same reason that the interference influence was split. One part for when there is no interference present in the frequency band and one for when there is. In the first case the standard BER

expression is used and in the second case the SINR is used, since the interference signal now occupies a relatively large band compared to the receivers sensitivity. Thus the approximation of a wide band noise signal holds under the condition that the receiver's chirp is overlaps with the interference signal.

$$P_e = \left(1 - \frac{W_i}{W_c}\right) Q\left(\sqrt{\frac{\mathscr{E}_s}{N_0}}\right) + \frac{W_i}{W_c} Q\left(\sqrt{\frac{\mathscr{E}_s}{N_0 + \mathscr{E}_i}}\right) \tag{2.66}$$

Here $\mathscr{E}_s$ represents the bit energy ($\mathscr{E}_b A_c^2$) and $\mathscr{E}_i$ represents the interference energy ($\frac{P_i P_c}{2}$). Figure 2.14 shows the simulation results combined with this approximation. For a SIR of two or four, the BER approximation holds reasonably well. However, when the SIR becomes one the error of the approximation is clearly visible. This is likely due to the error of the convolution approximation, which has a mismatch at lower frequencies, while the receiver branches are sensitive for frequencies in this region.

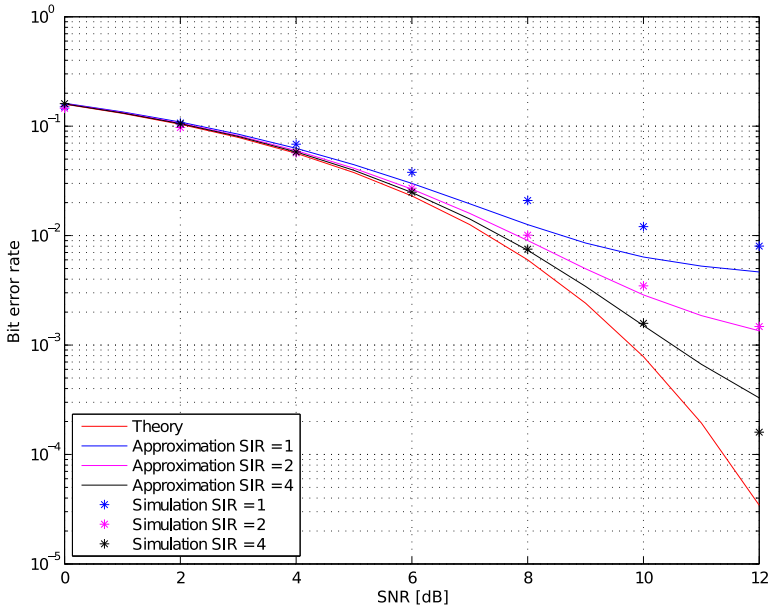


Figure 2.14: BER for chirped BFSK modulation with an interference band occupying 10% of the chirp bandwidth.

CHAPTER 3

The proposed synchronization scheme

In this chapter the proposed synchronization scheme is introduced. However, first a short overview is presented regarding synchronization methods, as described in literature. Based on the proposed synchronization scheme, two detection methods are defined and reviewed. This leads to a final algorithm of which the implementation is explored in the next chapter.

3.1 Chirp based synchronization in literature

As has been indicated in subsection 2.3.3, in previous decades surface acoustic wave (SAW) filters were commonly used to generate and receive chirp signals. Additionally, when using SAW filters non-coherent modulation schemes were used, omitting the use of a synchronization scheme.

However, in recent years synchronization based on chirp signals has received some new attention, both in the field of chirped communication as well as in other communication fields where a chirp preamble is introduced. The amount of research in this area is, however, still quite limited. The following works represent the recent work as was known at the time of writing this thesis

In the work of (Sun and Liu [37]), a matched filter is used to compress the received chirp signal to a pulse. These peaks are then used to determine the timing offset for the demodulator. A drawback of this method is that it is very sensitive to white noise. Therefore, a mechanism is used to improve the signal to noise ratio (SNR), based on (Dowla et al. [8]), involving an analog feedback loop.

Alternatively, (Boumard and Mammela [5]) propose the use of chirp signals as a preamble for orthogonal frequency division multiplexing (OFDM) communication. Synchronization is performed in two steps, coarse and fine-grain, via the combination of autocorrelation, reversed autocorrelation and matched filtering. Results show few to no errors for a SNR of 2 dB or better.

3.2 The algorithm

This section introduces the fast Fourier transform (FFT) based synchronization algorithm. Figure 3.1 illustrates the frequency of two up-chirp signals which are "out of sync" with each other over time. For now the focus will be on the specified up-chirp signals, while the same approach can be used for down-chirp signals. The combination of up- and down-chirps is treated in section 3.4.

In Figure 3.1 the signal starting at $t = 0$ represents the receiver's local chirp or local oscillator (LO), while the other signal is the received chirp to synchronize with. Note that it is the relative offset we are interested in. The time offset is indicated by Δt_1 & Δt_2 and the frequency offset by Δf_1 & Δf_2.

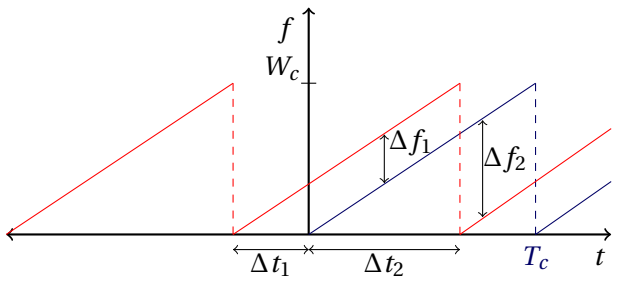


Figure 3.1: Illustration of two out of sync up-chirp signals

From the illustration the following can be observed:

- The sum of both time and frequency offsets are equal to the chirp's period and bandwidth.
- The previously introduced chirp rate (section 2.3) also relates the time and frequency offsets.

Based on the last observation the time-offset can be determined by detecting the frequency-offsets, which become available after mixing both signals. The mixing will be performed by a complex mixer to be able to distinguish between positive and negative frequencies. Thus the received signal will be multiplied with the LO:

$$s_{mix}(t) = r(t)s_{LO}(t) \tag{3.1}$$

For simplicity, assume that the transmitted signal is not modulated and that no noise is added to the transmission. The received signal will then only consist of a chirp signal:

$$\begin{aligned} r(t) &= \sqrt{\frac{2\mathscr{E}_c}{T_c}}\cos(2\pi f_c t + \pi\beta t^2) \\ &= \frac{A_c}{2}\left(e^{j(2\pi f_c t+\pi\beta t^2)} + e^{-j(2\pi f_c t+\pi\beta t^2)}\right) \end{aligned} \tag{3.2}$$

While the LO will generate a complex chirp signal:

$$s_{LO}(t) = A_{LO}e^{j(2\pi f_c t+\pi\beta t^2)} \tag{3.3}$$

Next, two time intervals are considered, the first being $0 \le t < \Delta t_2$ and the second $\Delta t_2 \le t < T_c$, corresponding to Figure 3.1. For the first part the received signal is:

$$\begin{aligned} r_1(t) &= c(t+\Delta t_1) \\ &= \frac{A_c}{2}\left(e^{j2\pi f_c(t+\Delta t_1)+\pi\beta(t+\Delta t_1)^2} + e^{-j2\pi f_c(t+\Delta t_1)+\pi\beta(t+\Delta t_1)^2}\right) \end{aligned} \tag{3.4}$$

And mixing leads to the following expression:

$$\begin{aligned} s_{1,mix}(t) &= \frac{A_c A_{LO}}{2}\left(e^{j(2\pi f_c(t+\Delta t_1)+\pi\beta(t+\Delta t_1)^2)} + e^{-j(2\pi f_c(t+\Delta t_1)+\pi\beta(t+\Delta t_1)^2)}\right)e^{j(2\pi f_c t+\pi\beta t^2)} \\ &= \frac{A_c A_{LO}}{2}\left(e^{j(2\pi f_c(t+\Delta t_1)+\pi\beta(t^2+\Delta t_1^2+t\Delta t_1))} + e^{-j(2\pi f_c(t+\Delta t_1)+\pi\beta(t^2+\Delta t_1^2+t\Delta t_1))}\right)e^{j(2\pi f_c t+\pi\beta t^2)} \\ &= \frac{A_c A_{LO}}{2}\left(e^{j(2\pi f_c(2t+\Delta t_1)+\pi\beta(2t^2+\Delta t_1^2+t\Delta t_1))} + e^{-j(2\pi f_c\Delta t_1+\pi\beta(\Delta t_1^2+t\Delta t_1))}\right) \end{aligned} \tag{3.5}$$

The result consists of two components. The first is a new chirp signal at twice the center frequency (f_c) and chirping at least twice as fast. The second component is the desired tone due to the timing offset with an additional phase offset. For completeness, once synchronization is reached Equation 3.5 will reduce to:

$$s_{1,mix}(t) = \frac{A_c A_{LO}}{2}\left(e^{j2(2\pi f_c t + \pi\beta t^2)} + 1\right) \tag{3.6}$$

Again the additional chirp signal is present. However, the second component reduces to a constant value.

Continuing with Equation 3.5, the signal is passed through an ideal lowpass filter to remove the undesired chirp signal. This leads to a signal containing only the desired tone:

$$\begin{aligned} s_{1,lpf}(t) &= \frac{A_c A_{LO}}{2} e^{-j(2\pi f_c \Delta t_1 + \pi\beta(\Delta t_1^2 + t\Delta t_1))} \\ &= \frac{A_c A_{LO}}{2} e^{-j(2\pi(f_c + \frac{\beta\Delta t_1}{2})\Delta t_1 + \pi\beta t\Delta t_1)} \end{aligned} \tag{3.7}$$

The result shows that the filtered signal consists of two components, a phase offset and the desired frequency offset. By taking the derivative of the phase in Equation 3.7, the instantaneous frequency is determined, resulting in a relation between the frequency-offset and time-offset.

$$\begin{aligned} \phi(t,\Delta t_1) &= -2\pi f_c \Delta t_1 - \pi\beta\Delta t_1^2 - \pi\beta t\Delta t_1 \\ \frac{\partial\phi(t,\Delta t_1)}{\partial t} &= -\pi\beta\Delta t_1 \\ &= 2\pi\left(-\frac{\beta\Delta t_1}{2}\right) \\ \Delta f_1 &= -\frac{\beta\Delta t_1}{2} \qquad (3.8) \\ \Delta t_1 &= -\frac{2\Delta f_1}{\beta} \qquad (3.9) \end{aligned}$$

This result is valid for the region $0 \leq t < \Delta t_2$ in Figure 3.1. For the second interval the input signal is shifted by Δt_2:

$$r_2(t) = c(t - \Delta t_2) \tag{3.10}$$

Using the same steps as before, this results in:

$$\Delta f_2 = \frac{\beta\Delta t_2}{2} \tag{3.11}$$

$$\Delta t_2 = \frac{2\Delta f_2}{\beta} \tag{3.12}$$

3.3 Frequency detection

In the previous section, the algorithm for the synchronization has been discussed and relevant relations have been presented. However, the topic of actually detecting the frequencies has been left open. This section continues to extend the previously presented chain of operations towards a high-level system.

3.3.1 Discrete Fourier Transform

Figure 3.2 illustrates the processing steps which are taken so far, which are already present at the receiver, and the operations which are still required. A discrete Fourier transform (DFT) operation will be used to determine the frequency offsets. To do so the signals coming from the lowpass filters must first pass through an analog to digital converter (ADC) and some control logic will have to calculate the time offset based on the DFT's output and adjust the LO to match the incoming signal.

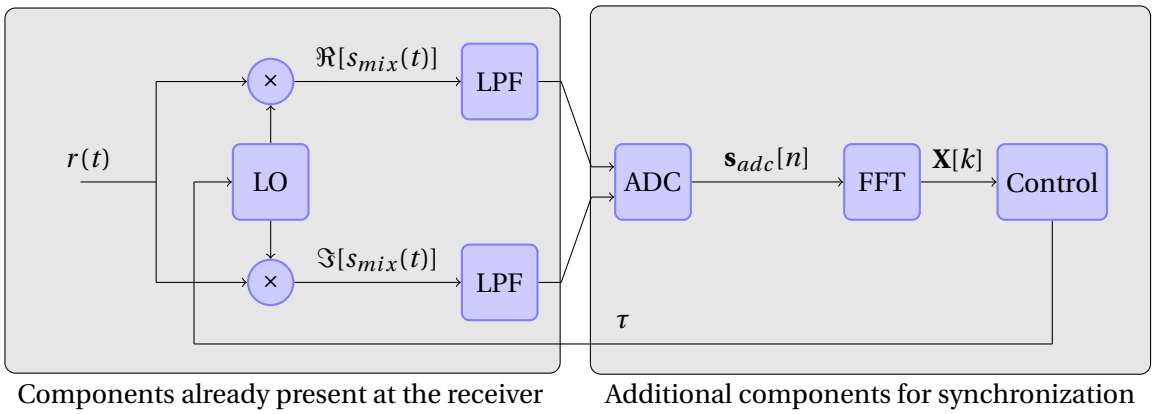


Figure 3.2: Diagram of the synchronization model

The first step is sampling, followed by applying some quantization function $q(x)$, stated in Equation 3.13 and Equation 3.14 and illustrated in Figure 3.3, both of which are assumed to be ideal for the time being. Later on noise contribution will be added such as quantization noise and aperture jitter. This type of jitter is included as it will be the dominant noise source at the desired samples rates (Le et al. [24]). The functionality of a anti-aliasing filter is incorporated in the lowpass filter.

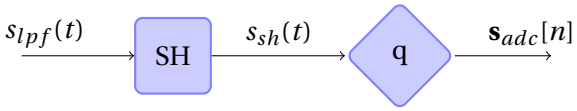


Figure 3.3: Functionality and internal signals of the ADC

$$s_{sh}(t) = \sum_{n=-\infty}^{\infty} s_{lpf}(t)\Pi(t - nT_s) \tag{3.13}$$

$$\mathbf{s}_{adc}[n] = q(s_{sh}(nT_s)) \tag{3.14}$$

The next step is applying the DFT, which is defined as:

$$\mathbf{X}[k] = \sum_{n=0}^{N-1} \mathbf{x}[n] e^{-j2\pi \frac{kn}{N}} \tag{3.15}$$

where N is the number of points or order. Since it is assumed that the quantization is ideal,

Equation 3.13 is substituted into Equation 3.15 for $\mathbf{x}[n]$ with $t = nT_s$, resulting in:

$$\begin{aligned}\mathbf{X}[k] &= \sum_{n=0}^{N-1} \frac{1}{2} e^{-j(2\pi(f_c+\frac{\beta\Delta t_1}{2})\Delta t_1+\pi\beta nT_s\Delta t_1)} e^{-j2\pi\frac{kn}{N}} \\ &= \frac{1}{2}\sum_{n=0}^{N-1} e^{-j(2\pi(f_c+\frac{\beta\Delta t_1}{2})\Delta t_1+\pi\beta nT_s\Delta t_1+2\pi\frac{kn}{N})} \\ &= \frac{1}{2} e^{-j(2\pi(f_c+\frac{\beta\Delta t_1}{2})\Delta t_1)} \sum_{n=0}^{N-1} e^{-j(\pi\beta nT_s\Delta t_1+2\pi\frac{kn}{N})}\end{aligned} \tag{3.16}$$

As was already noted in the previous section, the first component in the exponential's argument leads to a phase offset and thus does not influence the offset frequency. To determine this frequency the bin number, $\tilde{k}$, corresponding to the largest frequency component, is determined.

$$\begin{aligned}\tilde{k} &= \underset{0\le k<N}{\arg\max} |\mathbf{X}[k]| \\ &= \underset{0\le k<N}{\arg\max} |\sum_{n=0}^{N-1} e^{-j(\pi\beta nT_s\Delta t_1+2\pi\frac{kn}{N})}|\end{aligned} \tag{3.17}$$

The maximum value is reached when the two components in the argument of the exponential are equal to each other. This will lead to a sum of vectors, independent of n, which do not cancel each other out. However, the DFT has a limited accuracy, which leads to an approximation of the frequency Δf_1, indicated by $\Delta\hat{f}_1$.

$$\begin{aligned}\pi\beta nT_s\Delta t_1 &\approx 2\pi\frac{\tilde{k}n}{N} \\ \beta\Delta t_1 &\approx 2\frac{\tilde{k}}{NT_s}\end{aligned} \tag{3.18}$$

Substituting Equation 3.8 in Equation 3.18 and including the inherent rounding, indicated by the $\lfloor\cdot\rceil$ or round to nearest integer operation, leads to the approximation:

$$\Delta\hat{f}_1 = -\frac{\tilde{k}}{N}F_s = \left\lfloor\frac{\Delta f_1}{F_s}\right\rceil F_s \tag{3.19}$$

Thus the frequency offset is resolved towards the nearest DFT bin, leading to the approximation $\Delta\hat{f}_1$. For $\Delta\hat{f}_2$ a similar result may be achieved, only with the opposite sign. Furthermore it can be seen that the number of DFT points directly influences the accuracy of the system, as $\Delta\hat{f}_1$ can only be assigned values which are a integer multiple of $\frac{F_s}{N}$.

3.3.2 Detection methods

Based on the results from the previous section, it is clear that two relevant frequencies may be extracted from the DFT's result, $\Delta\hat{f}_1$ and $\Delta\hat{f}_2$. Therefore it would make sense to use both frequencies to determine the timing offset. However the fact that these two frequencies are present does not guarantee their detectability.

In an ideal, noise free environment, performing the DFT will result in correctly detecting the two expected tones. However this is not the case in a practical system. Due to noise, not only present in the received signal but also generated by the system itself, the tones may be buried in the noise floor.

This phenomenon can be made clear by calculating the energy of the tones. Assuming the detection is performed for one chirp period, this interval is again split in two parts Δt_1 and Δt_2. Now the energy in each signal, assuming no noise, will be relative to the duration of each part.

$$\mathscr{E}_i = \int_T |s_{i,lpf}(t)|^2 dt \tag{3.20}$$

$$\begin{aligned}\mathscr{E}_1 &= \int_0^{\Delta t_2} |\frac{A_c A_{LO}}{2} e^{-j(2\pi f_0 \Delta t_1 + \pi\beta(\Delta t_1^2 + t\Delta t_1))}|^2 dt \\ &= A_c^2 A_{LO}^2 \frac{\Delta t_2}{4}\end{aligned} \tag{3.21}$$

$$\mathscr{E}_2 = A_c^2 A_{LO}^2 \frac{\Delta t_1}{4} \tag{3.22}$$

Keeping in mind that $\Delta t_1 + \Delta t_2 = T_c$, the energy of each tone is scaled linearly depending on the timing offset it represents. This is also directly related to the power, as illustrated in Figure 3.4, where the dashed line indicates the maximum power for each frequency. One can also see that when $\Delta t_1 = \Delta t_2$ then $P_1 = P_2$.

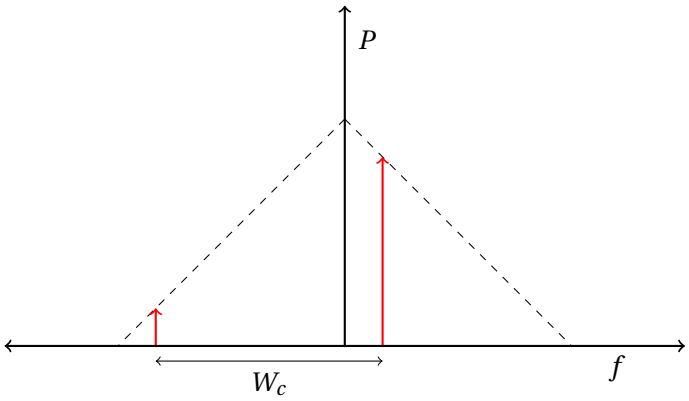


Figure 3.4: Illustration of the power

Based on this result, a detection scheme is formulated:

$$\tau = \sum_i \alpha_i \Delta t_i \quad i \in (1,2) \tag{3.23}$$

$$\sum_i \alpha_i = 1 \tag{3.24}$$

in which α_i represent the i-th weight factor, allowing the use of various detection schemes. However, as a single user situation is assumed two simple, yet strict, methods are considered:

- The positive negative detection (PND) method.
  This finds the strongest frequency component in both the positive and negative spectrum and averages the resulting time offset, thus $\alpha_i = \frac{1}{2}$.
- The maximum power detection (MPD) method.
  This finds the strongest frequency component in the complete spectrum.

$$\alpha_i = \begin{cases} 1 & \Delta f_i = \min(\Delta f_1, \Delta f_2) \\ 0 & \text{otherwise} \end{cases}$$

Alternative detection methods are, off course, possible however not considered in this work. Such alternatives include, for example, expressing the weight factor as a ratio of the detected frequencies and the chirp bandwidth, or detecting more then two tones to resolve false positives.

3.4 Using a up and down chirp combination

The described synchronization method explicitly uses either up or down chirps, such that two distinctive tones will indicate both the time and frequency offset between the signals. An alternative scheme would be to use the combination of an up and down chirp, as is used as carrier signal in (Dutta et al. [9]). This is illustrated in Figure 3.5. In this case the combination of both types form a signal chirp period.

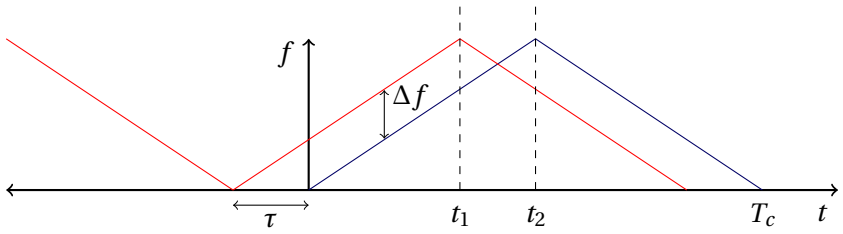


Figure 3.5: Illustration of two out of sync up-down-chirp signals

As can be seen in this illustration, only one time and frequency offset is indicated, as Δf, and it's negative counter part $-\Delta f$, are valid for the largest period of time ($0 \leq t < t_1$) and ($t_2 \leq t < T_c$). When looking at the spectrum, these tones will be present together with another chirp signal, as the frequency difference within $t_1 \leq t \leq t_2$ varies linearly over time, from Δf to $-\Delta f$.

In this specific situation the detection of the offset τ, with the previously described methods, will function. However, one can imagine that when the time offset increases, both the offset frequencies and the range of this additional chirp will increase, leading to an increase of noise around the origin of the frequency axis. This effect is shown in Figure 3.6, where an offset of 20% of the chirp period is applied.

As discussed in the previous section the power of the synchronization tones decreases when the time offset increases. This principle still applies to these synchronization tones. The additional chirp, however, has a static power level while it's total energy increases. As a consequence, a time offset around half the chirp period becomes undetectable. Other offset values are still be detectable.

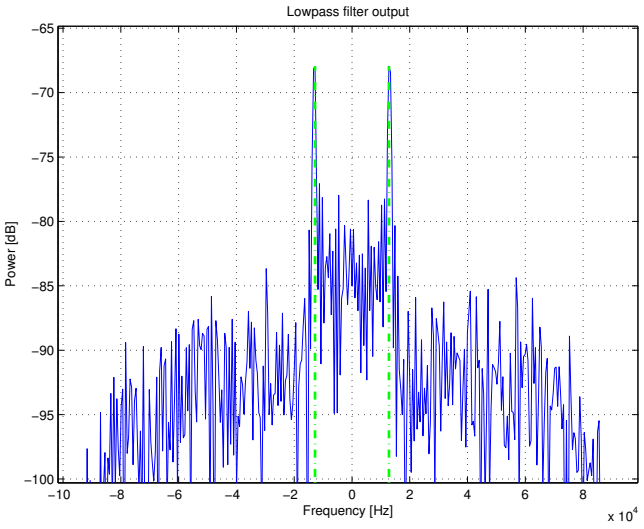


Figure 3.6: Resulting spectrum after mixing two up-down chirps with an offset of $0.2T_c$

3.5 Conclusions

In this chapter a synchronization method is proposed. Mixing two unsynchronized chirp signals will lead to two distinct frequency components, named synchronization tones, around the origin of the frequency spectrum. The frequencies of these tones are, via the definition of the chirp rate, directly related to the corresponding time offsets.

To perform the required set of operations, the receiver needs to be extended with an ADC, DFT and digital logic, implementing a detection method. This detection method performs the conversion from a frequency offset to a time offset. Two detection methods are proposed, which will be investigated further in the next chapter. First the positive negative detection (PND), which averages the time offset corresponding to the two synchronization tones. The second method, maximum power detection (MPD), utilizes the fact that the power of each tone is inversely linearly related to the time offset and focusses on detecting only the strongest tone.

The use of a combination of up and down chirps to synchronize is not recommended as the detectability of the synchronization tones is troublesome when the input and local chirp are completely "out of phase", corresponding to a time offset around $0.5T_c$.

CHAPTER 4

Design of the synchronization module

Following the introduction of the synchronization algorithm and a high-level design (Figure 3.2), this chapter treats the identification of the parameters which influence the synchronization performance and finds the optimum values considering the given requirements.

Based on the components introduced in the previous chapter, the following parameters are investigated:

- Analog to digital converter
  - Sample rate
  - Resolution
- Discrete Fourier transform
  - Order or number of bins
  - Measurement window
  - Window functions
  - Twiddle factor quantization

A number of these parameters are directly related to each other. These are:

$$N_{fft} = \frac{W_{fft}}{W_{res}} \tag{4.1}$$

$$t_m = \frac{N_{fft}}{F_s} \tag{4.2}$$

Where:

- N_{fft} is the number of fast Fourier transform (FFT) points;
- W_{fft} is the FFT's input signal bandwidth;
- W_{res} is the FFT's resolution (Hz/bin);
- F_s is the sample frequency of the analog to digital converter (ADC);
- t_m is the measurement window.

In the next section some initial values for these parameters are presented together with simulation results, to give a first impression of the algorithm's performance. The sections which follow discuss the parameters for the ADC and FFT and their impact on the overall performance. Additionally an indication for power requirements is presented. Finally, the last section is aimed at the design of the control logic implementing the detection method.

4.1 Baseline model

The previous chapter described the system from an operational point of view, without taking any implementation of these operations into account nor investigating the performance. This section describes the base model, a simulation model which gives a first indication of how the algorithm performs. Before doing so, two questions must be answered, how to choose the initial parameters and how to determine the performance?

For synchronization, the algorithm must find the current time/frequency offset between a received and the local chirp signal. Accuracy, however, is limited by the use of the discrete Fourier transform (DFT) as the number of bins limits the frequency representation. Therefore performance will be determined by the DFT finding the bin which is the closed to the actual frequency offset for a given FFT resolution.

The initial parameters are listed in Table 4.1, with corresponding expressions where available and other values which are based on the results of preliminary simulations. First, the signal to noise ratio (SNR) is chosen at zero, corresponding to a low power transmission. Next is the lowpass filter, implemented as a 6th order IIR Butterworth filter of which the characteristics are illustrated in Figure 4.1.

As shown in Figure 3.2, two implementations are present, one for the real component and one for the imaginary. To determine the bandwidth of the filters recall that the synchronization tones occur within the range $-W_c < \Delta f < W_c$. By filtering the real and imaginary components separately, and the fact that the bandwidth is centered around the origin of the frequency-axis, the passband can be set equal to half the required bandwidth or one chirp bandwidth (W_c).

Sampling and quantization of the complex signal is performed at a rate equal to it's bandwidth, which is also equal to the lowpass filter's stop band such that it acts as an anti-aliasing filter for the ADC. The initial resolution of the ADC is chosen to be 6 bits, minimizing the effects of quantization noise. Finally the number of FFT bins follow from the number of samples available when measuring over one chirp period (T_c).

Parameter	**Expression**	**Value**
SNR		0
Filter pass band	W_c	65536 Hz
Filter stop band	$2W_c$	131072 Hz
ADC sample rate	$2W_c$	131072 Hz
ADC resolution		6 bits
FFT bins	$2W_c/W_{res}$	256
Simulation iterations		10.000

Table 4.1: Parameter values for the base model

4.1.1 Simulation method

Given the parameters from Table 4.1 and a MATLAB model of Figure 3.2, Monte Carlo simulations are performed in which a uniformly distributed random time offset is generated between two chirp signals. These signals are fed to the model which will determine the synchronization tones based on the two detection methods, as explained in the previous section.

Figure 4.2 shows one representation of the frequency spectrum before and after the ADC. In both figures the synchronization tones are indicated by the green vertical lines. While the negative frequency component can clearly be distinguished, the other component is only a few decibels stronger then the noise.

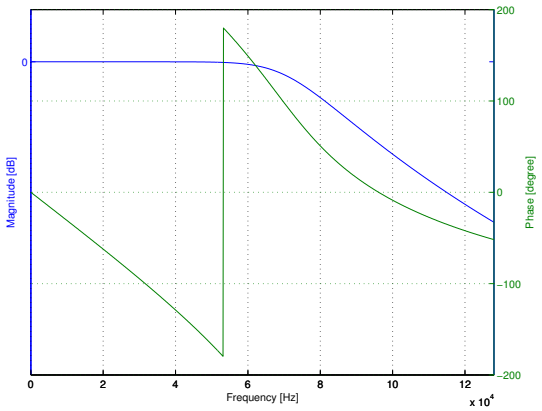
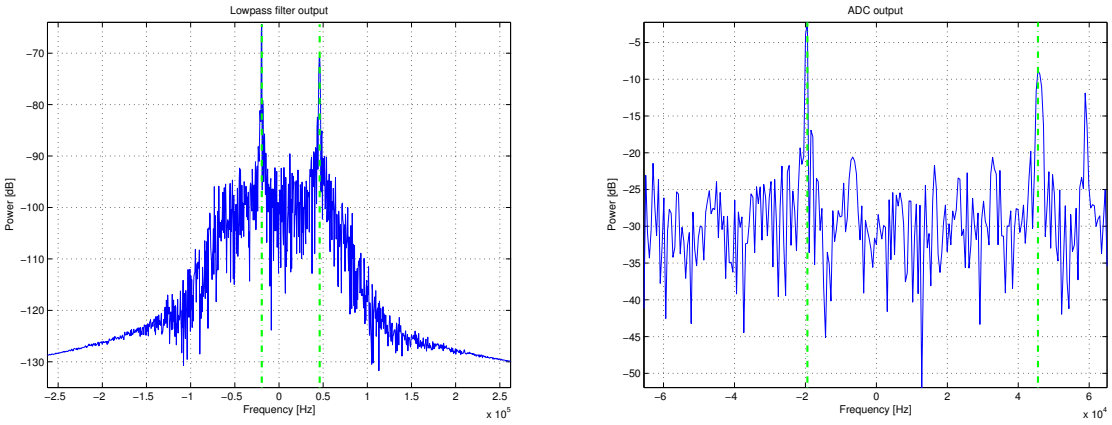


Figure 4.1: Ideal lowpass filter specification, a 3 stage 6th order Butterworth filter



(a) Lowpass filter output spectrum

(b) ADC output spectrum

Figure 4.2: Spectra before and after the ADC, showing the bandwidth reduction, quantization effects and the synchronization tones (two green vertical lines)

This is also shown in Figure 4.3, where the output of the DFT block is depicted. In this figure the additional dashed line indicates the maximum power level the synchronization tones can reach, for a given a SNR value. While one tone is clearly visible, it does not reach it's maximum value. However, a number of smaller components are present right next to the tone due to spectral leakage.

In Figure 4.4, the performance of this base model is shown by the percentage of correct synchronizations. Each bin in the figure corresponds to an FFT bin as this is a limiting factor of the algorithm's accuracy. This result shows that the positive negative detection (PND) method leads to a correct result in approximately 55% of the simulations. The maximum power detection (MPD) method leads to a correct result in approximately 92% of the total simulations. This difference is mainly due to the power ratio between the tones, as stated earlier. When one of the two tones gets close to or buried in the noise floor, results degrade rapidly for the first method.

Observing the error spreading, it was noticed that both are symmetrical around the zero bin, as could be expected given the uniformly distributed offset and symmetry present in the characteristics of the synchronization tones. With the MPD method, the errors made are all present in the two adjacent bins, likely due to spectral leakage in the DFT.

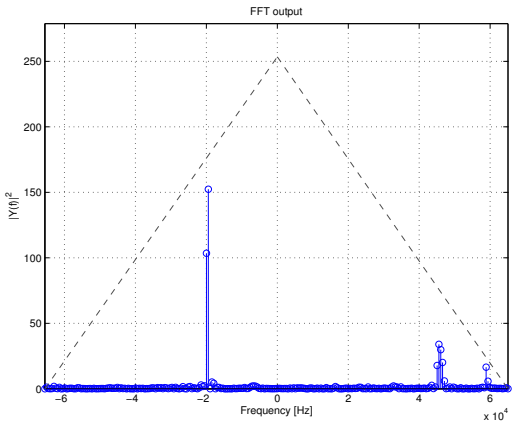


Figure 4.3: Output of the FFT component, the dashed line indicates the maximum power similar to Figure 3.4

For the PND method the error is spread out over the complete bandwidth of the DFT. As with the other method, a portion is due to spectral leakage. However, the largest part is due to the averaging with wrongly selected frequencies, occurring when one tone is close to the origin of the frequency-axis. Either leakage will contribute energy in bins on both the positive and negative side of the spectrum, or the second tone becomes indistinguishable from the noise contributions. Additionally, the lowpass filter does attenuate the highest frequencies slightly, smoothing any peaks present.

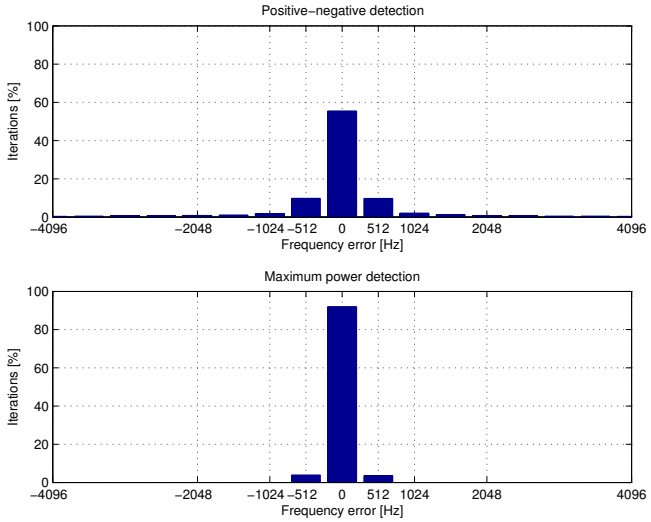


Figure 4.4: Synchronization errors of both detection methods when simulating the base model

4.2 Analog to digital converter

The purpose of an ADC is to convert a continuous-time signal with an infinite range of values to a discrete-time signal with a finite number of finite values. This conversion is performed in two interchangeable steps: sampling and quantization.

The next subsection gives a brief background on both operations, followed by two subsection in which the parameters of both operations are investigated.

4.2.1 Background

In this subsection definitions of both sampling and quantization are presented together with their dominant noise components, which are also included in the model.

Sampling

By sampling a signal, it is transformed from the continuous-time to the discrete-time domain. This is done by multiplying the input signal with a Dirac pulse train as expressed in Equation 4.3. The time interval between the pulses is the sample period (T_s). Note that the values of the resulting discrete-time signal are still continuos.

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(t)\delta(t - nT_s) \tag{4.3}$$

Performing the sampling operation limits the bandwidth of the output signal to half of the sampling frequency (F_s), which in turn constraints the bandwidth (W_x) of the input signal x_s. This is expressed by the Nyquist criterion (Reed [31]):

$$W_x \leq \frac{F_s}{2} \quad x_s \in \mathbb{R} \tag{4.4}$$

If this criterion is not satisfied, aliasing will occur causing artifacts in the output spectrum. Besides the addition of undesired frequency components, the ability to reconstruct the original signal is lost. Therefore it is common to limit the input signal's bandwidth by an anti-aliasing filter.

This criterion is aimed at real signals, when sampling complex signals, the bandwidth is effectively doubled as not one but two samples are taken at each sample interval, thus:

$$W_x \leq F_s \quad x_s \in \mathbb{C} \tag{4.5}$$

Since a complex signal consists out of two real signal, the first criteria can also be applied on both parts, real and imaginary, individually.

Aperture jitter

Aperture jitter is caused by variations of the sample period, generally due to variations in an external clock source. Thus instead of sampling at the exact sample period, compared to the previous sample, the current sample period is either shorter or longer. This is also known as sample-to-sample uncertainty, which leads to additional distortion in the ADC. This noise source even dominates quantization noise at sample rates above 10 MHz, as has been observed in (Le et al. [24]).

To ensure that the noise due to this jitter does not dominate the quantization noise, an expression may be derived to determine the maximum allowable jitter. Assuming that an ADC is presented with an input signal, utilizing the full swing range and with the maximum allowed frequency, any deviation in the sample time may only lead to less then half a quantization level change in the sampled values. This leads to the following constraint as has been derived in literature for Nyquist sampled ADCs (Reed [31], Le et al. [24]):

$$\tau_{a,max} \leq \frac{1}{2^B \pi f_{max}} \tag{4.6}$$

$$f_{max} \leq \frac{1}{2^B \pi \tau_{a,max}} \tag{4.7}$$

τ_a representing the aperture jitter and B the number of bits of the ADC. Additionally, the contribution to the SNR may also be expressed as a function of the aperture jitter:

$$\text{SNR} = 20\log_{10}\left(2\pi f_s \tau_a\right) \tag{4.8}$$

An example is illustrated in Figure 4.5. Here a tone of 16 Hz is sampled at a frequency of 128 Hz, in which the SNR degrades an additional 10 dB due to the jitter. The aperture jitter is modeled by generating a uniformly distributed value within the range $(-\tau_a, \tau_a)$ for each sample period and is added to this sample period. τ_a is defined as the allowed deviation, $\tau_a = \frac{R_a}{f_{max}} \leq \tau_{a,max}$ with R_a being the ratio between the maximum jitter and the period of the maximum frequency. The value of 0.1 is used for R_a as is suggested in (Reed [31]).

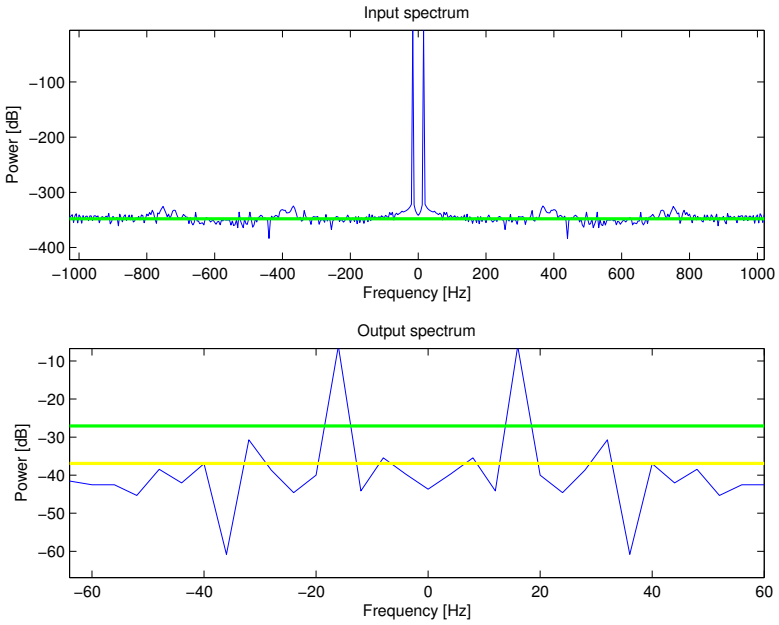


Figure 4.5: Illustration of the the effect of aperture jitter on the output spectrum. The green horizontal lines indicate the SNR, calculated with Equation 4.14 and excluding the effect of jitter. The yellow line illustrates the SNR with the jitter effect taken intro account.

Analytically modeling the effects of this type of noise is still the subject of discussion in literature. Commonly it is assumed that the jitter's spectrum is white or conditionally white (Da Dalt et al. [7]), resulting in a white error spectrum. A generalized model is presented in (Arkesteijn et al. [2]), indicating for example that Equation 4.6 is too pessimistic. However, the presented method requires performance measurements of the sample clock to determine the jitter spectrum.

Quantization

While the first operation of the ADC is limiting the number of values/samples that are presented to the component, it's second operation is to limit these values to predefined discrete levels. To do so a quantizer rounds the input value to the nearest digital code available. The smallest possible step between digital codes is called the least significant bit (LSB). Thus this LSB corresponds, for a uniform quantizer, to the minimal value the input has to increase/decrease for the the output to change. For a uniform quantizer we may therefore determine the LSB as follows:

$$\Delta = \frac{V_{FS}}{2^B} \text{ V/bit} \tag{4.9}$$

Where V_{FS} stands for the full scale voltage or the input range of the quantizer and B for the number of bits. From this the input range can be divided into a number of decision levels, each level corresponding to a unique code. The number of levels is equal to:

$$\text{Levels} = 2^B - 1 \tag{4.10}$$

Based on the previous statements, there are two ways to place the decision levels on the input range. These are the mid-riser and mid-tread. Figure 4.6 shows an illustration and the corresponding function for the two methods. This illustration shows that the difference lies in how to pass through the origin. Which type of quantizer to use generally depends on the required number of decision levels. When this number is odd a mid-tread is best suited and for an even number of levels a mid-riser.

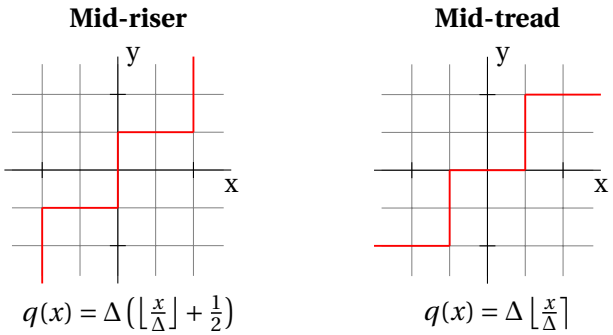


Figure 4.6: Visual and mathematical representation of the mid-riser and mid-tread quantizers

Decision levels

After deciding which quantization method to use, arises the question of how to determine one LSB and thereby the decision levels of the quantizer. In this thesis the ADC's decision levels are based on the input signal's statistical characteristics as described in (Kokkeler [23]). This method uses the standard deviation of the input signal to determine the full swing range of the ADC, after which the decision levels are placed within this range.

For a given input signal x, it's standard deviation σ_x is first normalized and then transformed to match the full range of the quantizer. Since this is too strict, too much clipping will occur. Therefore the factor H, or Headroom factor, is added to increase the input range. Note that this method should be applied to real and imaginary signals separately.

$$g = \frac{2^{B-1}}{H\sigma_x} \tag{4.11}$$

$$\mathbf{x} = \frac{1}{g} q(gx) \tag{4.12}$$

To determine how much headroom is required, a simple simulation is performed for a number of values, as there is no way to determine this value otherwise. For a sine wave which is passed through additive white gaussian noise (AWGN) with a SNR of 0 dB, a headroom factor of 2 yields the best signal to quantization noise ratio (SQNR). More information about this simulation is available in Appendix A.

Quantization noise

Quantizing a signal is a non-linear operation which leads to an error (e_q) in the output signal's representation when compared to the original.

$$e_q(x) = q(x) - x \tag{4.13}$$

This quantization error results in a noise component in the output, or the noise floor when looking at the output spectrum. The ratio between the input signal power (P_x) and the quantization noise power (N_Q) is called the signal to quantization noise ratio.

$$\text{SQNR} = 10\log_{10}\left(\frac{P_x}{N_Q}\right) \text{dB} \tag{4.14}$$

Which may be approximated by a function of the number of bits (Reed [31]):

$$\text{SQNR}_{avg} = 6.02B + 4.77 - 10\log_{10}(\eta) \text{ dB} \tag{4.15}$$

$$\text{SQNR}_{avg,sinusoid} = 6.02B + 1.76 \text{ dB} \tag{4.16}$$

η is defined as the average power of the input signal over it's peak power value. Additionally, the SQNR may be improved by oversampling the output signal (Reed [31]). Thus for a certain output bandwidth W_y, $F_s > 2W_x$.

$$\text{OSR} = \frac{F_s}{2W_y} \tag{4.17}$$

$$\text{SQNR}_{avg} = 6.02B + 4.77 - 10\log_{10}(\eta) + 10\log_{10}(\text{OSR}) \text{ dB} \tag{4.18}$$

With the previous expressions it is assumed that, firstly the quantization noise is white and secondly a single sinusoid is present at the input. This first assumption does not hold for resolutions lower then 7 bits (Pelgrom [27]). In such a case harmonics of the input signal become noticeable in the output spectrum, also called quantization spurs. Lowering the resolution leads to an increase in the power of these harmonics and effectively reduces the spurious free dynamic range (SFDR) of the ADC.

In (Blachman [4]), expressions are presented to determine the amplitude of these spurs in various scenarios. The most common scenarios include one or two sinusoidal inputs, resulting in Equation 4.19 and Equation 4.20. The first expression shows the odd harmonics due to the non-linear behavior of the quantifier, the second equation also describes the intermodulation products.

$$\begin{aligned} y &= \sum_{p=1}^{\infty} A_p \sin(2\pi p f t) \\ A_p &= \begin{cases} \partial_{p,1} A + \sum_{n=1}^{\infty} \frac{2}{n\pi} J_p(2n\pi A) & \text{for odd p} \\ 0 & \text{for even p} \end{cases} \end{aligned} \tag{4.19}$$

$$\begin{aligned} y &= \sum_{p=-\infty}^{\infty} \sum_{q=-\infty}^{\infty} A_{pq} \sin(2\pi(pf_1 + qf_2)t) \\ A_{pq} &= \partial_{p,1}\partial_{q,0} A_1 + \partial_{p,0}\partial_{q,1} A_2 + \sum_{n=1}^{\infty} \frac{2}{n\pi} J_p(2n\pi A_1) J_q(2n\pi A_2) \quad p+q = \text{odd}, p+q > 0 \end{aligned} \tag{4.20}$$

The symbol $\partial_{a,b}$ is equal to one when $a = b$ and zero otherwise. The term A_i represents the amplitude of the i-th sine wave at the input.

Dynamic range

The SFDR represents the smallest tone which can be sensed above the noise floor in the presence of another, stronger, tone. It is measured from the maximum input value to the maximum spur. For a sine wave input this is the third harmonic.

This value may be approximated by (Oude Alink et al. [26]):

$$\text{SFDR} = 8.07B + 3.29 \text{ dB} \tag{4.21}$$

Example

Figure 4.7 shows both input and output spectra of the ADC model when a 16$[Hz]$ signal is presented at the input. In the lower plot, three horizontal lines are added, indicating the following:

- the black lines indicate the SFDR, in this example with one bit resolution, theory yields a value of 11.3 dB and the model 10.4 dB;
- the red line indicates the noise floor located at -31.8 dB, which is not comparable to the SQNR of 16.8 dB (including the oversampling contribution).

The remaining difference is caused by the FFT operation, which acts as a narrowband spectrum analyzer. It's contribution, also known as the FFT's processing gain (Kester [22]), is calculated by:

$$G_{FFT} = 10\log_{10}\left(\frac{N_{fft}}{2}\right) \text{ dB} \tag{4.22}$$

In this example G_{FFT} has a value of 15.0 dB, which when added to the SQNR leads to the noise floor level.

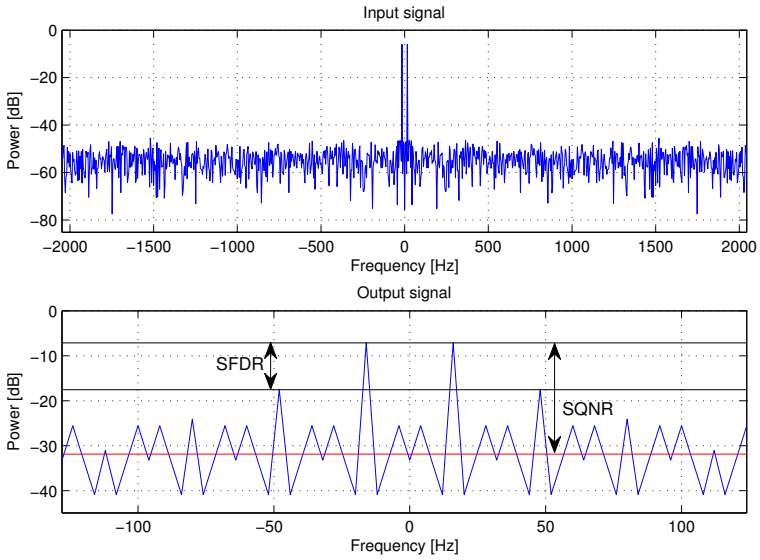
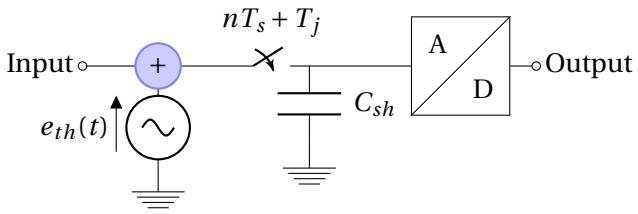


Figure 4.7: Input and output spectrum of the ADC model with one bit resolution.

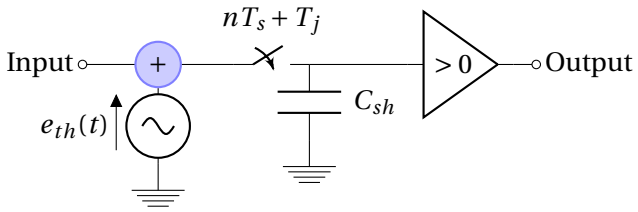
4.2.2 Resolution

The first parameter to explore is the resolution, which is performed via simulation. Each detection method has been subjected to simulations in which the number of bits has been varied from six down to one bit. All other parameters are kept as listed in Table 4.1.

During the simulation two models are used for the ADC, one for resolutions of two and more bits and one with a one bit resolution. For the first type a standard ADC model is used, see Figure 4.8a. The second version replaces the quantizer by a comparator which compares the input signal with ground, effectively extracting the sign of the input signal, see Figure 4.8b. Both models include aperture jitter (T_j) and thermal noise (e_{th}) as additional noise sources in addition to the inherent quantization noise.



(a) Multibit ADC model



(b) 1-bit ADC model

Figure 4.8: Schematic representation of both ADC models

Based on the background presented in the previous section, it is expected that lowering the number of bits will results in a degradation of the performance of both detection methods. Mainly because a lower resolution leads to less and stronger harmonic spurs, which may be falsely identified as synchronization tones. The PND will most likely suffer more, as this method requires the detection of both tones. As it's search is performed on both the positive and negative spectrum separately, the changes of detecting spurs instead of the synchronization tones increase, considering the effect on time offset on the power of each tone, see subsection 3.3.2.

By varying only the resolution, the results from Figure 4.9 are generated. As has been shown in earlier results, there is a significant difference in performance between the two detection methods. Secondly, it is interesting to see that even when the resolution is reduced to one bit, 90% of the iterations still result in correct synchronization with the MPD method. Increasing the number of bits gradually increases this percentage to nearly 93%.

Additionally, the standard deviation of the error for both methods is listed in Table 4.2. Based on previous results found in Figure 4.4, the order of magnitude is as expected. Around 160 Hz, or 0,2% of the chirps bandwidth for the MPD and 8 kHz / 12% for the PND method.

Detection method	**Resolution**					
	1	2	3	4	5	6
PND	9716	8466	7735	7691	7458	7573
MPD	164.0	159.2	157.1	155.9	157.7	157.5

Table 4.2: Standard deviation in Hz of the error spreading of both detection methods for various ADC resolutions

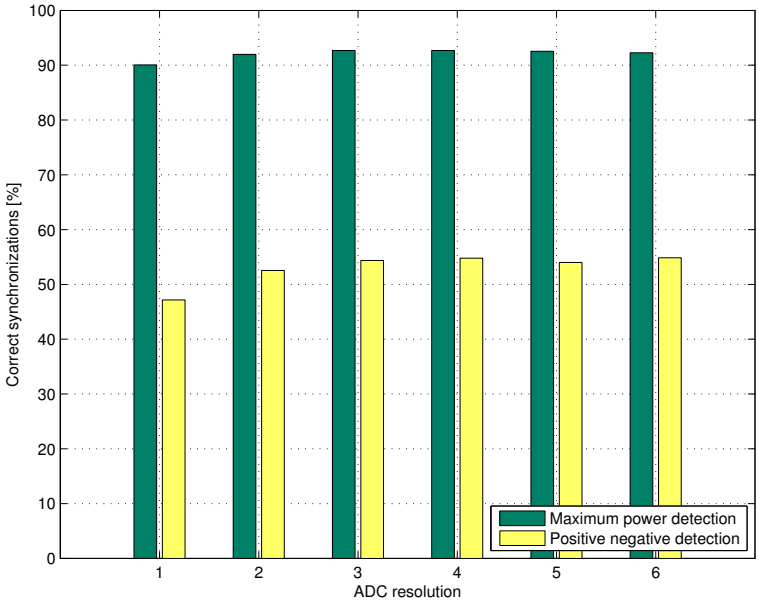


Figure 4.9: Influence of the ADC's resolution on the algorithm's performance.

4.2.3 Sampling

The second parameter is the sample rate which can be varied in two ways. The first way is oversampling, which will increase the dynamic range of the quantizer, as expressed in Equation 4.18. Therefore it is expected that the PND method will benefit from this change.

The second way is lowering the sample rate by reducing the required bandwidth, from the filters to the DFT. This option must be combined with the MPD method, as this detection scheme selects the powerful frequencies which are present in the region $-\frac{W_c}{2} \leq f \leq \frac{W_c}{2}$. This has been indicated in subsection 3.3.2, leading to a reduction of the bandwidth and therefore the sample rate by a factor of two.

For comparable results, the number of bins in the FFT are increased/decreased accordingly such that the resolution is kept equal for each variation. Figure 4.10 shows the performance of both detection methods with different resolutions and oversampling ratios. As expected, the PND method shows the highest performance increase, an additional 5% when oversampling twice. However, the gain is limited when increasing the oversampling ratio (OSR) even further. Also it should be noted that small differences are due to simulation variation, or else one might think oversampling actually reduces performance when using a 6 bit resolution and the MPD method.

When reducing the sample rate to half the chirp bandwidth and adjusting the lowpass filter accordingly, a performance decrease is visible in Figure 4.11. On average this difference is 10%. When analyzing the error spread it can be seen that the difference is added to the bins adjacent to the center bin. Further analysis of the simulation show that these errors occur when synchronization tones are present near the edge of their range ($-\frac{W_c}{2} \leq f \leq \frac{W_c}{2}$). By increasing only the bandwidth of the filter, the performance increases again, showing that the detection method is quite sensitive to the filter characteristics.

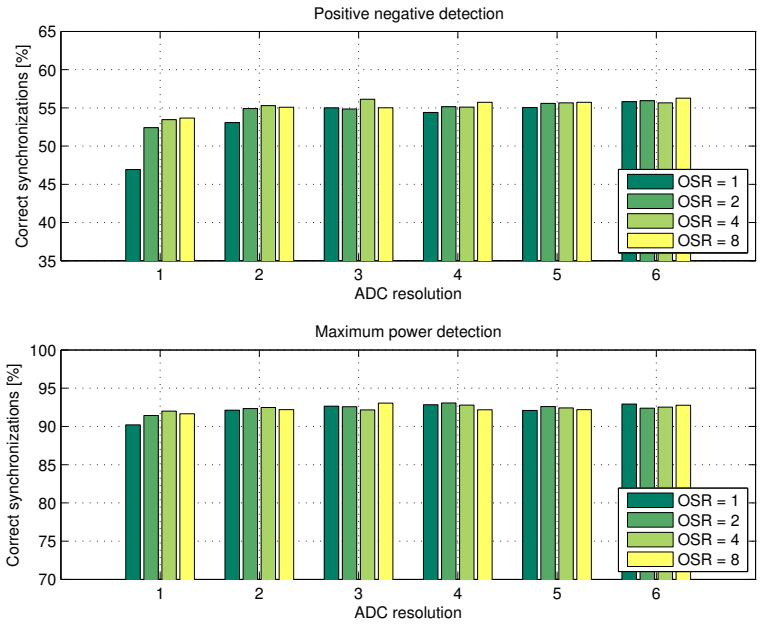


Figure 4.10: Performance under various oversampling ratios of the ADC

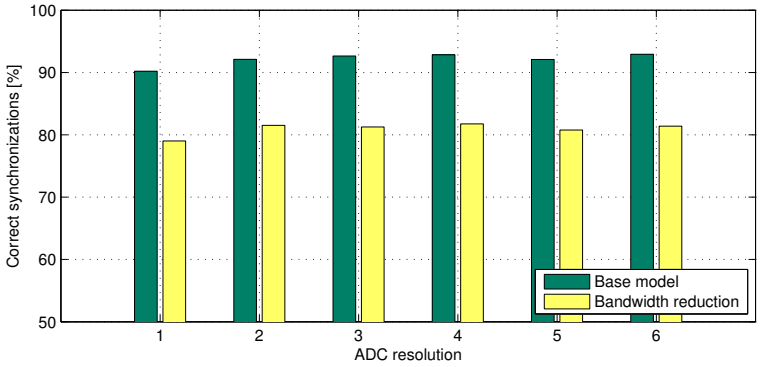


Figure 4.11: Influence of reducing the ADC output bandwidth on the overall performance

Implementation

The actual implementation of the ADC module is out of the scope of this work. However the required components are described here together with an indication of their characteristics based on recent work, see Table 4.3. The values listed in the column "Energy per operation" are, where required, derived from the specification via: $\frac{Power\,[\mathrm{W}]}{Clock\,[\mathrm{Hz}]}$. It is to note that the architecture in (Goll and Zimmermann [12]) consists of only a comparator while the other works include additional components such as preamplifiers.

While the comparator will form the core of the 1-bit ADC, additional components will be required to connect it with the lowpass filter's output on one side and the DFT module on the other side. Depending on the maximum filter output levels, some voltage scaling might be required. The work in (Goll and Zimmermann [12]) will need additional functionality as this comparator performs best with a relatively high common-mode input voltage as compared to the supply voltage, effectively reducing the input's voltage range.

Work	Clock GHz	Sensitivity $^{mV}/_{GHz}$	Operating voltage V	Power	Energy per operation	Technology
[42]	20	$^{<10}/_{10}$	3.3	31 mW	1.55 pJ	0.35 μ m BiCMOS
[12]	0.6	$^{12.1}/_{0.6}$	0.65	128 μ W	213 fJ	65 nm CMOS
[25]	20	$^{8.9}/_{3}$	1.2	82 mW	4.1 pJ	0.25 μ m HBT

Table 4.3: Characteristics and performance of a number of comparators

Conclusion

Based on the results presented in this section it can be concluded that a one bit ADC is a feasible option. Additionally, the PND method is clearly not a suitable detection method and thus will be disregarded in the remainder of this work. As for the sampling rate, Nyquist sampling leads to the best results, considering that oversampling increases the operational frequency and theirby the power requirements with an equal factor while the performance gain is minimal.

It was unexpected that the reduction of the ADC's output signal bandwidth, by lowering the sample rate while keeping the other system parameters such as DFT resolution and measurement time constant, degrades the overall performance, as the strongest synchronization tone is always present in this region. It is likely that the characteristics of the lowpass filter are the influence of this degradation, however this is not investigated further.

4.3 Discrete Fourier transform

For sensing the frequencies after the ADC, a DFT operation is performed, transforming the time-domain signal into the frequency-domain.

The next subsection provides a brief background in the parameters and characteristics of this operation. Thereafter these parameters are explored during the performance simulations.

4.3.1 Background

The discrete Fourier transform, Equation 4.23, is the discrete form of the Fourier Transformation, evaluated on a set of N discrete frequencies. One can see this as quantizing the spectrum of a given signal. This is also visible in the definition of the DFT's twiddle factors, which divide the unit circle in N sections, see Equation 4.24.

$$\mathbf{X}[k] = \sum_{n=0}^{N-1} \mathbf{x}[n] W_N^{kn} \tag{4.23}$$

$$W_N^{kn} = e^{-j2\pi \frac{kn}{N}} \tag{4.24}$$

Fast Fourier transform

Direct implementation of the DFT is not very efficient as it would require N^2 multiplications ($\mathcal{O}\left(N^2\right)$). Under the name of "fast Fourier transform", more computationally efficient implementations have been derived, reducing the number of operations to $\mathcal{O}\left(N\log_2(N)\right)$ for a basic implementation. The basic building block of any FFT is the butterfly, for example a radix-2 butterfly corresponds to a 2-point FFT. Two options are available to derive the expression of any radix based FFT, either using decimation in time or decimation in frequency. The first method combines even and odd input terms while the second combines the even and odd

output terms. For both methods the radix number corresponds to the number of inputs or outputs which are grouped. For example a 4-point radix-2 FFT via decimation in frequency is derived as follows:

$$\mathbf{X}[k] = \sum_{n=0}^{3} \mathbf{x}[n] W_4^{nk} \qquad k = 0 \ldots 3 \tag{4.25}$$

$$\mathbf{X}[2k] = \sum_{n=0}^{1} \mathbf{x}[n] W_4^{2kn} + \sum_{n=2}^{3} \mathbf{x}[n] W_4^{2kn} \qquad k = 0 \ldots 1 \tag{4.26}$$

$$= \sum_{n=0}^{1} (\mathbf{x}[n] + \mathbf{x}[n+2]) W_2^{kn} \tag{4.27}$$

$$\mathbf{X}[2k+1] = \sum_{n=0}^{1} (\mathbf{x}[n] - \mathbf{x}[n+2]) W_4^{n} W_2^{kn} \qquad k = 0 \ldots 1 \tag{4.28}$$

Notice that both expressions, $\mathbf{X}[2k]$ and $\mathbf{X}[2k+1]$, contain the twiddle factor W_2^{kn}, which correspond to the factors ± 1, allowing a simple replacement for the multiplication. Furthermore, the order of the twiddle factors has been reduced by a factor equal to the radix number.

The development of an efficient FFT has been and remains an actual topic in research as the transformation is used in a growing number of applications, for example in communication like orthogonal frequency division multiplexing (OFDM) and ultra wideband (UWB) techniques, (music) compression and image processing. This contributes to a large collection of implementations and techniques to improve performance. Therefore, for the implementation of this DFT operation, a selection is made out of available designs instead of creating a custom design. These implementations are reviewed in the following sections.

Window functions

As has been noted in the previous chapter, DFTs may suffer from spectral leakage, this is a consequence of the finite input sequence which is fed to the DFT. Because it is assumed that such a sequence is periodic, allowing the use of the Fourier Transform. However, it is uncommon that all the frequency components of the sampled signal fit an integer number of times into this sequence. This leads to discontinuities in the time-domain and to the spreading of energy over multiple frequencies in the frequency-domain. In the worst case scenario, spectral leakage causes frequencies which are present in the input signal to be undetectable after the transformation.

One option to reduce this spectral leakage is to apply a window function on the input sequence before the transformation. Such a window can force the sequence to be zero at both the start and end, smoothing any discontinuities. Figure 4.12 illustrates the time and frequency domain representations of a number of common window functions. An interesting note is that as the time function become smoother, the bandwidth around $f = 0$ becomes larger.

4.3.2 Resolution and measurement time

In this subsection both resolution and measurement time are treated, as both are linked. This becomes visible when substituting Equation 4.1 in Equation 4.2:

$$F_s = \frac{W_{fft}}{t_m W_{res}} \tag{4.29}$$

Since the bandwidth of the DFT is already limited by the sample frequency F_s and this frequency is set to be equal to twice the chirp bandwidth, it becomes clear that in this case t_m and W_{res} are inversely linear related. Therefore the measurement window will be varied between a

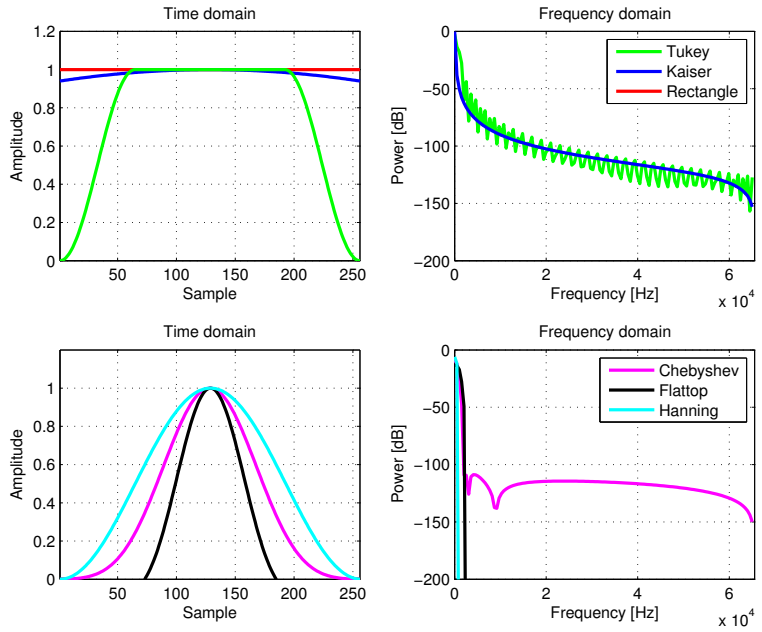


Figure 4.12: Characteristics of a number of window functions

half, one and two chirp periods to investigate this effect, meanwhile the DFT resolution is modified such that the product of the measurement time and DFT resolution remains constant.

In Figure 4.13, the performance results are displayed for these different measurement times combined with an ADC resolution from one to three bits. As is expected from the previous results, the number of bits does not influence the performance in a significant matter. However, when a measurement window of two chirp periods is selected, the results show a significant difference. Therefore a detailed explanation for this set of parameters is illustrated in Figure 4.14.

Keeping in mind that these figures illustrate the performance as a function of the accuracy of the DFT, it can be expected that increasing the resolution (in Hz/bin), resulting in fewer bins, leads to combining small errors and indicating these as perfect synchronizations. Thus by increasing the measurement window, the DFT resolution decreases, leading to a more accurate spectrum. This is supported by the standard deviations as listed in Table 4.4.

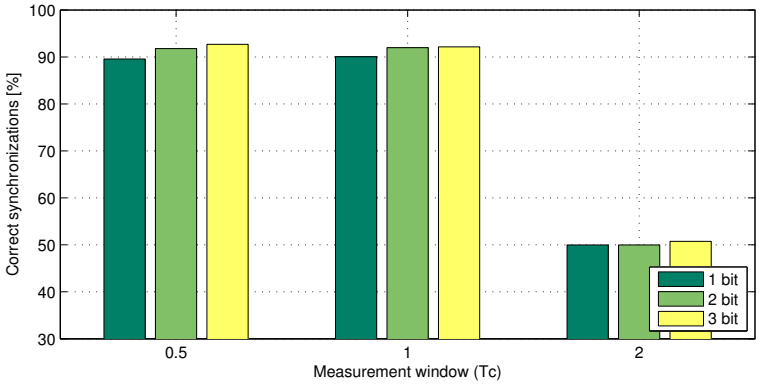


Figure 4.13: Performance with various measurement windows

Thus, when lowering the resolution, by increasing N_{fft}, the accuracy of the DFT is increased. However, from Table 4.4 it is also clear that there is no linear relation between the two. This

is indicated by the slight difference between the results corresponding with a measurement window of one and two chirp periods.

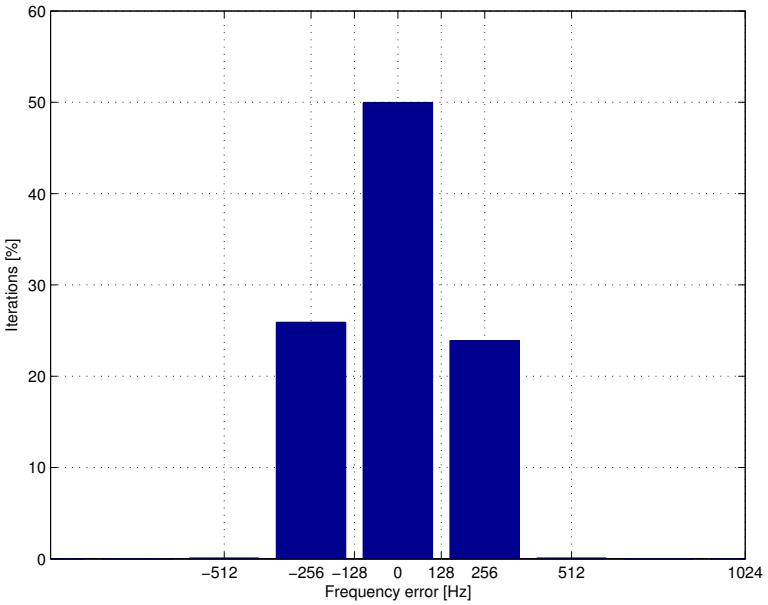


Figure 4.14: Synchronization error with the settings $t_m = 2T_c$, $N_{fft} = 512$ and $W_{fft} = W_c$

t_m	W_{res}	N_{fft}	Mean Hz	Standard deviation Hz
$0.5T_c$	512	128	-24	1194
$1.0T_c$	256	256	0.3	164.0
$2.0T_c$	128	512	2.4	156.2

Table 4.4: Statistics from the frequency offset error (Using a one bit ADC)

Window function

In subsection 4.1.1, during the discussion of the initial performance of the algorithm, it was noted that the synchronization tones do not seem to reach their expected power values due to spectral leakage. To compensate for this phenomena two options are available. The first being an increase in the number of bins to increase the accuracy of the DFT, as has been explored in the previous section. The second is to apply a window function before the DFT operation.

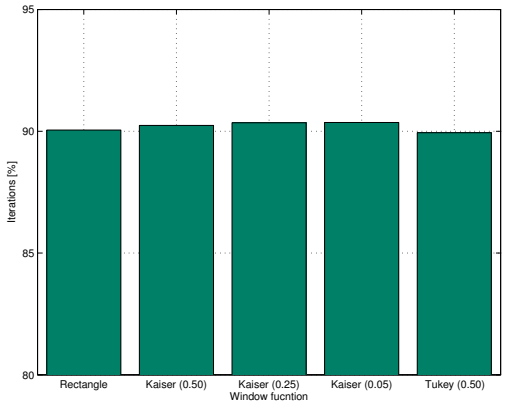
The following window functions, plotted in Figure 4.12, are considered:

- Rectangle
- Chebyshev
- Flat Top
- Hanning
- Kaiser
- Tukey

From the above list of window functions, Figure 4.15a and Figure 4.15b present the results from the windows which do not degrade the overall performance. From both the figure and table it

is concluded that applying a window function does not improve the detection of the synchronization tones. It should be noted however that applying certain window functions, such as the Flat Top window, does result in a spectrum which is a clearer representation of the input signal.

An example of this, is how the synchronization tones at the edge of their range become visible in the spectrum. This is illustrated in Figure 4.16, showing a peak at both expected position of the tones, indicated by the vertical lines. However as also can be seen, there is a smaller dynamic range due to this window function, making it harder for the proposed detection methods to detect the correct peak each time.



(a) Correct synchronizations based on the DFT resolution

Window function	Standard deviation Hz
Rectangle	164.06
Kaiser (0.50)	164.58
Kaiser (0.25)	203.67
Kaiser (0.05)	164.12
Tukey (0.50)	166.30

(b) Standard deviation of the frequency error

Figure 4.15: Performance comparison of various windows function

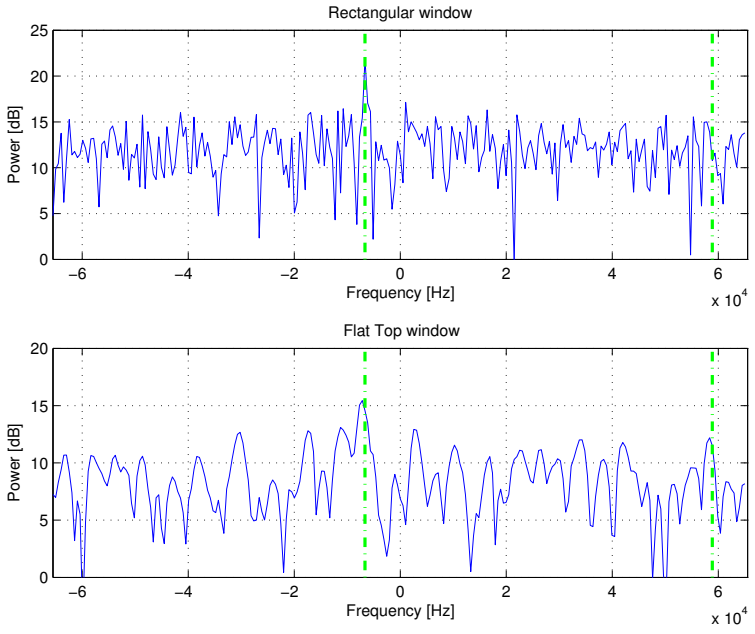


Figure 4.16: Output spectra after applying a rectangular and a flat top window, green lines indicate the expected position of the synchronization tones

4.3.3 Twiddle factors

A fixed point analysis is performed by using the same simulation model as was previously outlined, based on the samples going to and from the DFT. This ensures the dataset contains values captured from various time offsets, covering the important scenarios. The analysis is performed with the fixed point toolbox included with MATLAB. For completeness Figure 4.17 illustrates the definitions for a fixed point word. The word is split into two parts, the integer word representing the number before the decimal point and the fractional word representing the number after the point. The number of bits used to describe both words are referred to as: integer word length (IWL) and fractional word length (FWL).

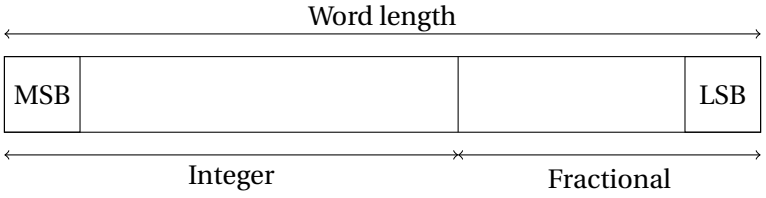


Figure 4.17: Fixed point representation of a word, including definitions

Table 4.5 shows the world length and FWL for the input, twiddle factors and output of the DFT module. Intermediate signals are not considered in this example as these depend on the implementation. In the error column, FWL corresponds to the error made in the fractional range. This is likely due to rounding errors in MATLAB, as the numbers which cannot be represented are in the order of 2^{-50}.

	Word length	**Fractional length**	**Error**
Input	2	0	0
Twiddle factors	6	5	2.1% FWL
Output	10	0	0

Table 4.5: Minimal fixed point analysis for the DFT (256 points)

Overall performance is comparable to previous results, as the correct iterations remains at 90%. The difference lays in the standard deviation, which increased from 164 Hz to 517 Hz. This is due to reduction of the dynamic range of the DFT, a consequence of reducing the number of bits.

4.3.4 Implementation

As the implementation of the DFT is generally performed with FFT architectures to reduce the number of computations, this subsection will review a number of techniques and architectures which have been introduced in recent years. The considered work consists of a number of implementations, demonstrating different techniques to implement FFTs, including different architectures and physical implementations. Therefore none will match the requirements, as listed in Table 4.6, exactly. However, the presented results will serve as an impression and the presented works are scalable to the desired parameters.

The first question is how to implement the FFT, as there are several platforms available. In (Boyapati and Raja Kumar [6]) the FFT is implemented in an application-specific integrated circuit (ASIC), a dedicated and a reduced instruction set computing (RISC) digital signal processor (DSP) and in a generic DSP. With these implementations a comparison is made to determine which is the most energy efficient. The ASIC implementation is both the fastest and

Parameter	Value
Minimal operating frequency	128 [Mhz]
Number of points	256
Minimal data-path (IWL, FWL)	15 (10, 5)

Table 4.6: Requirement for the FFT module

the most energy efficient implementation, followed by the RISC DSP with only half the amount of MIPS per Watt. The least efficient is the generic DSP implementation.

Table 4.7 presents a selection of FFT implementations. (Suleiman et al. [36]) proposes a scalable architecture in which a number of parallel processing elements are placed in a feedback loop with a memory component. This allows dynamically altering the FFT order depending on the required detail. An interesting detail is that both output and input are "in-place", meaning that there is no need for any additional bit/byte reversal scheme.

(Henry and Nazhandali [18]) use a similar configuration as (Suleiman et al. [36]), but focus on lowering the power consumption. This is firstly done by supplying the transistors with sub-threshold voltages, changing the well known behavior from a digital switch to a modulator of leakage currents. Secondly the idea of parallel pipelined architectures is used as a number of butterflies are placed and executed in parallel. This is not done with the aim to increase throughput but to compensate for the loss of throughput from lowering the supply voltage. While this leads to a significant power reduction, it also leads to a large increase in area. Additionally, scalability is added by simply turning off a number of the parallel butterflies and showing that this technique outperforms dynamic clock and voltage scaling.

Another loop based architecture is presented in (Ghissoni et al. [10]), which is set apart from the previous two by using a constant multiplier and decomposing the twiddle factors into smaller factors which, by addition/subtraction, form the required factor, instead of having multiple separate multipliers.

(Huang and Chen [21]) present an FFT architecture aimed specifically at the IEEE 802.15.3c UWB communication system. Which requires a high throughput, reflected by the few cycles which are required to perform an FFT operation. This architecture is included to show the difference between utilizing low-power requirements by either fast computation with shortly high power requirements or slow computation and continuous a low power requirement.

Finally the architecture proposed in (Han et al. [15]), consists of a Radix-4 serial delay commutator (SDC) with a number of interesting optimizations. These include a multipliersless architecture and exploiting the equality of the twiddle factors. The number of unique values is significantly lowered by viewing the complex twiddle factors as a combination of two factors out of one set and also independent of their sign. This however, comes at the expense of additional control logic for routing the correct input to the multipliers and for sign correction. The throughput of the design is increased by using parallel-pipelined architectures. This is at

Work	FFT bins	Data-path	Clock Mhz	Power mW	Technology
[36]	1024	32	654	183	45 nm Bulk CMOS
[18]	1024	32	222	9.00	UMC 90 nm
[10]	32	16	69.4	52.6	UMC 130 nm
[21]	512	12	320	103	UMC 90 nm
[15]	64	32	62.5	40.5	UMC 0.18 μ CMOS

Table 4.7: Performance information of various FFT implementations

the expense of the required area, due to additional processing elements and extended data shufflers.

While the requirements for fixed point size and operating frequency are easily compared in Table 4.7, their energy requirements are not. Therefore a figure of merit (FoM) is needed such that a comparison can be made independent of the used technology. First a number of normalization factors are introduced followed by the used FoM.

Inspired by (Huang and Chen [21]), the following two factors are used to normalize technology usage:

$$\frac{45\,\text{nm}}{L_{tech}}\left(\frac{0.9}{V_{DD}}\right)^2 \tag{4.30}$$

Both technology L_{tech} and supply voltage V_{DD} are scaled to the most recent technology in the set. The scaling itself is based on the assumption that the dynamic power, see Equation 4.31, will roughly scale linearly for capacitance, as dimensions scale linearly, and exponential for the voltage supply.

$$P_d = \alpha f C V_{DD}^2 \tag{4.31}$$

Where:

- α represents the switching activity;
- f the clock frequency;
- C the capacitance load;
- V_{DD} the supply voltage.

Additionally, as the various implementations have a different number of bins (N_{fft}), a third scaling term is introduced to normalize the computational complexity:

$$\frac{1}{N_{fft}\log_2(N_{fft})} \tag{4.32}$$

Based on the previous normalization factors, the following two FoMs are proposed and used to compare the different implementation's power requirements. The results are listed in Table 4.8.

- Energy per FFT operation normalized by it's order:

$$E_{FFT} = PT_{clk}N_{cycles}\frac{1}{N_{fft}\log_2(N_{fft})}\frac{45\,\text{nm}}{L_{tech}}\left(\frac{0.9}{V_{DD}}\right)^2\ [J] \tag{4.33}$$

  In which $PT_{clk}N_{cycles}$ represent the power requirement, operating frequency and the number of cycles required to perform one DFT, resulting in the energy required for one DFT operation.

- Energy per processing element (PE) normalized to a Radix-2 butterfly:

$$P_{PE} = \frac{P}{N_{PE}}\frac{2}{N_{radix}}\frac{45\,\text{nm}}{L_{tech}}\left(\frac{0.9}{V_{DD}}\right)^2\ [W] \tag{4.34}$$

  Where N_{PE} is the number of PEs and N_{radix} the radix of these elements. A PE can include a number of butterflies and multipliers but no memory.

Work	N_{cycles}	V_{DD}	N_{PE}	**Radix**	E_{FFT} J	P_{PE} W
[36]	1279	0.9	4	2	32.3×10^{-12}	42.2×10^{-3}
[18]	427 [1]	0.34	16	8	5.78×10^{-12}	0.48×10^{-3}
[10]	80 [1]	1.2 [2]	5	2	73.8×10^{-15}	2.05×10^{-6}
[21]	71	1 [2]	48	2/16	2.02×10^{-12}	0.15×10^{-3}
[15]	12 [1]	1.8 [2]	12	4	6.34×10^{-12}	0.10×10^{-3}

Table 4.8: Approximations of the energy and power requirements for the FFT architectures

From the results presented in Table 4.8, two designs stand out. Firstly (Suleiman et al. [36]), this design requires significantly more energy per operation and power per butterfly. This is likely due to the combination of a high number of cycles in combination with few butterflies and the reordering structure. This last feature has the advantage of placing the FFT output in order, but to do so requires $2 \times 10!$ delay elements for a 1024 point FFT, as was implemented by the authors.
Secondly, the design of (Ghissoni et al. [10]) is interesting as it performs significantly better from a power requirements point of view. While the use of multiplierless butterflies is also used in other designs, using such a butterfly in a feedback loop and performing multiplications not on individual or vector but matrix level has a positive effect on the power requirements.

Besides power requirements, the number of cycles (latency), to calculate the DFT is of interest and presented in Table 4.9. This is because an indication of the time it takes to perform the synchronization can be presented. Of the selected work, some provide the details which are required to determine the latency. From others this is derived based on the architecture description, which does mean these numbers are only an indication.

Work	N_{cycles}	**Serial IO**	**In order**
[36]	255	No	Yes
[18]	85 [3]	No	No
[10]	1024 [3]	Yes	No
[21]	70	No	Yes
[15]	64 [3]	Yes	No

Table 4.9: (Estimated) Number of cycles required to perform an DFT operation for the selected work and IO properties

It must be noted that the numbers presented in Table 4.9 do not take the number of cycles required to fill the random access memory (RAM) in the case when serial input is not an option.

4.3.5 Conclusion

Based on the results presented in this section the following is concluded. First the measurement time was investigated and the results have shown that an interval of one chirp period gives the best results, when taking both performance and hardware requirements into account. Lowering the measurement interval to half a chirp period shows a degradation of performance through the standard deviation of the output by a factor close to 7, see Table 4.4.

Secondly, the effect of a window function has been simulated, showing little performance variation of the selected functions. While some window functions do result in a spectrum which

[1] Value not presented in the paper, approximated by the number of butterfly operations: $\frac{N \log_{\text{radix}}(N)}{\text{radix}}$
[2] Value not presented in the paper, based on generic process information
[3] Value not presented in the paper, based on architectural analysis

gives a cleaner representation of the input, the current detection schemes do not benefit from this aspect.

Finally a number of FFT designs are presented, each using a different approach towards lowering power consumption. An effort is made to compare these works on the properties of energy efficiency and latency. Based on these results it was found that when taking these aspects into account the work of (Han et al. [15]) presents the best results, with a reasonable energy efficiency and relative low latency. The results are stored out of order, requiring some additional control logic to compensate. The work of (Ghissoni et al. [10]), while extremely low on power requirements, is not selected as the number of cycles to complete is more then 10 times larger then most alternatives. Additionally, though not taken into account while select a design, the work of (Han et al. [15]) uses common known techniques allowing a quick implementation.

4.4 Control logic

After completion of the DFT, the detection method must determine which frequency is the most powerful, or which bin corresponds to the largest amplitude. As the DFT produces a set of complex values, a conversion is required to determine the length of each element, for which a coordinate rotation digital computer (CORDIC) module can be used. When used in vectoring mode, the module rotates the given vector as closely as possible upon the positive side of the x-axis. It's precision is determined by the number of iterations performed.

This operation and further processing is listed in algorithm 4.1. A number of functions are introduced in this algorithm. First the `CORDIC` function is listed, which matches the previously described behavior. The function `ToBin` converts a number representing the current iteration to the corresponding bin number. If an out of place FFT implementation is used, this function performs the bit/byte reversal. On the other hand if an in place FFT is used, the function is not needed and `n` will be assigned to `bin`.

The last function, `BinToTime(n)`, transforms the given bin number to the corresponding time offset using the following equation:

$$\hat{\tau} = -\frac{2n}{\beta N_{FFT}} F_s \tag{4.35}$$

This conversion may be simplified to a shift operation by choosing the parameters to be numbers that are a power of two. Furthermore as $\hat{\tau}$ is periodic, the minus sign can be avoided.

Algorithm 4.1: Control logic for the MPD method

```
1  bin ← 0
2  max ← 0
3  for n ← 0 to N_fft − 1 do
4  |   f ← X[n]
5  |   (radius,error) ← CORDIC (ℜ[f], ℑ[f])
6  |   if radius > max then
7  |   |   max ← radius
8  |   |   bin ← ToBin(n)
9  |   end
10 end
11 τ ← BinToTime(bin)
```

CHAPTER 5

Impact on the BER due to imperfect synchronization

In the previous chapters the synchronization algorithm has been mathematically described and an implementation has been simulated to produce performance figures. These performance measurements provide an indication of the timing and frequency error that can be made during operation. In this final chapter the effect of this error on the communication performance is analyzed via simulations. The goal is to verify if the current synchronization performance is acceptable.

The first section defines expressions which indicate the limit of the error due to the synchronization module. This is followed by two sections which provide a brief review of both time and frequency offset on the binary frequency shift-keying (BFSK) modulation and the chirp signal processing as is performed in the receiver. The last section presents simulation results indicating the performance change due to synchronization errors.

5.1 Time and frequency offset

Recall Equation 3.9 and Equation 4.35, reprinted below for convenience. These equations describe the actual and approximated time offset as a function of the detected frequency offset.

$$\Delta t_1 = -\frac{2\Delta f_1}{\beta} \tag{3.9}$$

$$\hat{\tau} = -\frac{2n}{\beta N_{FFT}} F_s \tag{4.35}$$

Assuming Δt_1 represents the time offset of the strongest synchronization tone, this factor represents the timing offset τ. Furthermore, n may be decomposed into $n = k + m$, k indicating the correct bin and m the error relative to k. The error which is made is then limited by:

$$\begin{aligned} \epsilon_\tau &= \tau - \hat{\tau} \\ &= -\frac{2\Delta f_1}{\beta} + \frac{2(k+m)}{\beta N_{FFT}} F_s \\ &= \frac{m}{\beta N_{FFT}} F_s \\ \frac{\min(m)}{\beta N_{FFT}} F_s \leq \epsilon_\tau &\leq \frac{\max(m)}{\beta N_{FFT}} F_s \end{aligned} \tag{5.1}$$

Simulation results from the previous chapter have shown that for some cases the error is limited to the two bins adjacent to the zero error bin. In that case the previous expression reduces

to:

$$-\frac{F_s}{\beta N_{FFT}} \leq \epsilon_\tau \leq \frac{F_s}{\beta N_{FFT}} \tag{5.2}$$

This shows, as has been stated before, that the accuracy of the discrete Fourier transform (DFT) is the limiting factor for this synchronization scheme. Once this error has been introduced by the synchronization module, it propagates to the local oscillator (LO) as the local chirp signal is to be corrected by the timing offset. This leads in a time mismatch between the received chirped signal and the local chirp signal.

Due to the nature of the chirp signal, this time mismatch also leads to a frequency mismatch. As time and frequency are linked though the chirp rate, any time offset also leads to a frequency offset within the limits:

$$-\frac{\min(m)}{N_{FFT}} F_s \leq \epsilon_f \leq \frac{\max(m)}{N_{FFT}} F_s \tag{5.3}$$

5.2 Effect of offsets on BFSK modulation

Assuming that a signal is received at a correlator receiver, which introduces an time offset τ, and that the received signal can be decomposed into a message signal and noise contributions, as has been done in section 2.2, then the following expressions are valid for each branch:

$$\mathbf{r}_n = \int_\tau^{T_b+\tau} r(t_{tx}) s_n(t_{rx})\, dt_{rx} \quad n = 0, 1, \ldots M \tag{5.4}$$

$$\begin{aligned}
t_{rx} &= t_{tx} + \tau \\
\mathbf{r}_n &= \int_0^{T_b} r(t-\tau) s_n(t)\, dt \quad n = 0, 1, \ldots M \\
r(t) &= s(t) + n(t) \\
\mathbf{r}_n &= \int_0^{T_b} s(t-\tau) s_n(t)\, dt + \int_0^{T_b} n(t-\tau) s_n(t)\, dt \\
&= \mathbf{v}_n^1 + \mathbf{v}_n^2
\end{aligned} \tag{5.5}$$

where n indicates the branch number corresponding to a symbol. Our interest is in the effect of time and frequency offset on the component representing the message signal, $\mathbf{v}_n^1$:

$$\mathbf{v}_n^1 = \int_0^{T_b} s(t-\tau) s_n(t)\, dt$$

Both expressions, $s(t)$ and $s_n(t)$ are message signals which are valid for one symbol period. For $s(t)$ this is the region $0 < t < T_b$ and for $s(t-\tau)$ it is $\tau < t < T_b + \tau$. Leading to the following expression:

$$\mathbf{v}_n^1 = \begin{cases} \int_0^\tau s_{i-1}(t-\tau) s_n(t)\, dt + \int_\tau^{T_b} s_i(t-\tau) s_n(t)\, dt & \tau > 0 \\ \int_0^{T_b-\tau} s_{i-1}(t-\tau) s_n(t)\, dt + \int_{T_b-\tau}^{T_b} s_i(t-\tau) s_n(t)\, dt & \tau < 0 \end{cases} \tag{5.6}$$

in which i indicates the current and $i-1$ the previous symbol. In the following text it is assumed that $\tau > 0$ or, since τ is periodic when the chirp signals are used, is made positive.

Using the following signal definitions:

$$\begin{aligned} s_i(t) &= A_i \cos(2\pi f_i t) \\ &= \frac{A_i}{2}\left(e^{j2\pi f_i t} + e^{-j2\pi f_i t}\right) \\ s_n(t) &= e^{j2\pi f_n t} \end{aligned}$$

and substitution of these expressions in Equation 5.6 yields:

$$\begin{aligned} \mathbf{v}_n^1 =& \int_0^\tau \frac{A_i}{2}\left(e^{j2\pi(f_{i-1}(t-\tau)+f_n t)} + e^{j2\pi(f_{i-1}(\tau-t)+f_n t)}\right) dt \\ &+ \int_\tau^{T_b} \frac{A_i}{2}\left(e^{j2\pi(f_i(t-\tau)+f_n t)} + e^{j2\pi(f_i(\tau-t)+f_n t)}\right) dt \\ =& \mathbf{v}_{n,i-1}^1 + \mathbf{v}_{n,i}^1 \\ \mathbf{v}_{n,i-1}^1 =& \frac{A_i}{2}\frac{e^{j2\pi f_n \tau} - e^{j2\pi f_{i-1}\tau}}{j2\pi(f_n - f_{i-1})} \\ \mathbf{v}_{n,i}^1 =& \frac{A_i}{2}\frac{e^{j2\pi((f_n-f_i)T_b+f_i\tau)} - e^{j2\pi f_n \tau}}{j2\pi(f_n - f_i)} \end{aligned}$$

Next we introduce the terms $f_n = \frac{m_n}{T_b}$ and $f_i = \frac{m_i}{T_b} + f_o$, where f_o represents a frequency offset and m_n the n-th symbol. After some rewriting, this leads to:

$$\mathbf{v}_{n,i-1}^1 = \frac{A_i}{2}\frac{e^{j2\pi m_n \frac{\tau}{T_b}} - e^{j2\pi(m_{i-1}\frac{\tau}{T_b}+f_o\tau)}}{j2\pi(f_n - f_{m,i-1} - f_o)} \tag{5.7}$$

$$\mathbf{v}_{n,i}^1 = \frac{A_i}{2}\frac{e^{j2\pi(f_o T_b + m_i\frac{\tau}{T_b}+f_o\tau)} - e^{j2\pi m_n \frac{\tau}{T_b}}}{j2\pi(f_n - f_{m,i} - f_o)} \tag{5.8}$$

Note that in the case that τ is equal to zero, the term $\mathbf{v}_{n,i-1}^1$ will not exist. Next the following terms are introduced:

$$\lambda = \frac{\tau}{T_b} \tag{5.9}$$

$$\rho = f_o T_b \tag{5.10}$$

Such that both time and frequency offset are expressed as factors of the message period T_b.

$$\mathbf{v}_{n,i-1}^1 = \frac{A_i T_b}{2}\frac{e^{j2\pi m_n \lambda} - e^{j2\pi(\lambda(m_{i-1}+\rho))}}{j2\pi((m_n - m_{i-1}) - \rho)} \tag{5.11}$$

$$\mathbf{v}_{n,i}^1 = \frac{A_i T_b}{2}\frac{e^{j2\pi(\rho+\lambda(m_i+\rho))} - e^{j2\pi m_n \lambda}}{j2\pi((m_n - m_i) - \rho)} \tag{5.12}$$

The above expressions show that in the case of time offset, an additional term occurs in each branch. This can be seen as leakage due to the fact that the local and received signal are not

orthogonal (also see Equation 2.5) anymore:

$$\begin{aligned}
\langle s_m(t), s_n(t)\rangle &= \int_{-\infty}^{\infty} s_m(t-\tau)\bar{s_n}(t)dt \\
&= \int_{\tau}^{T_b} e^{j2\pi f_m(t-\tau)} e^{-j2\pi f_n t} dt \\
&= \int_{\tau}^{T_b} e^{j2\pi((f_m-f_n)t-f_m\tau)} \\
&= \left[\frac{e^{j2\pi((f_m-f_n)t-f_m\tau)}}{j2\pi(f_m-f_n)}\right]_{\tau}^{T_b} \\
&= \frac{1}{j2\pi(m-n)\Delta_f}\left[e^{j2\pi((m-n)\Delta_f T_b - m\Delta_f\tau)} - e^{-j2\pi n\Delta_f\tau}\right]
\end{aligned}$$

The above expression cannot be reduced in such a way that the criterion of Equation 2.4 is satisfied. Consequences and effects on the performance of BFSK are well described in the work of (Hinedi et al. [19]) and will therefore not be covered here.

5.3 Effect of offsets on chirped communication

In chapter 3, timing offsets were introduced on the transmitter side, as the received signal was out of sync with the local oscillator. Now it is assumed that synchronization is performed, thus any error is now due to timing offset in receiver's oscillator:

$$s_{LO}(t+\tau) = e^{j(2\pi f_c(t+\tau)+\pi\beta(t+\tau)^2)} \tag{5.13}$$

Next, the steps from Equation 3.5 to Equation 3.7 are repeated with the modified local signal.

$$r_1(t) = \frac{1}{2}\left(e^{j2\pi(f_c+f_m)t+\pi\beta t^2} + e^{-j2\pi(f_c+f_m)t+\pi\beta t^2}\right) \tag{5.14}$$

$$\begin{aligned}
s_{1,mix}(t) &= \frac{1}{2}\left(e^{j(2\pi(f_c+f_m)t+\pi\beta t^2)} + e^{-j(2\pi(f_c+f_m)t+\pi\beta t^2)}\right) e^{j(2\pi f_c(t+\tau)+\pi\beta(t+\tau)^2)} \\
&= \frac{1}{2}\left(e^{j(2\pi(f_c+f_m)t+\pi\beta t^2)} + e^{-j(2\pi(f_c+f_m)t+\pi\beta t^2)}\right) e^{j(2\pi f_c(t+\tau)+\pi\beta(t^2+\tau^2+t\tau)} \\
&= \frac{1}{2}\left(e^{j(2\pi f_c(2t+\tau)+2\pi f_m t+\pi\beta(2t^2+\tau^2+t\tau))} + e^{j(2\pi f_c\tau-2\pi f_m t+\pi\beta(\tau^2+\tau t))}\right)
\end{aligned} \tag{5.15}$$

$$\begin{aligned}
s_{1,lpf}(t) &= \frac{1}{2} e^{j(2\pi f_c\tau-2\pi f_m t+\pi\beta(\tau^2+\tau t))} \\
&= \frac{1}{2} e^{j(2\pi(f_c+\frac{\beta\tau}{2})\tau+\pi\beta t\tau)} e^{-j2\pi f_m t}
\end{aligned} \tag{5.16}$$

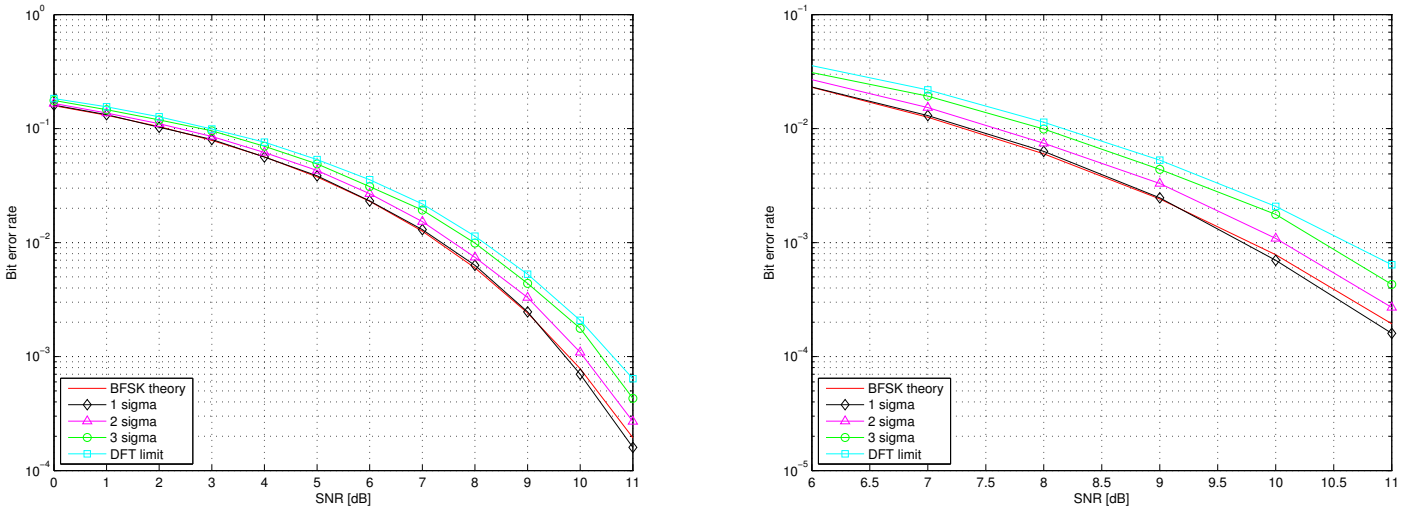
The resulting expression shows the residual frequency component, the first exponential, due to the timing error. The timing error is a result of the demodulator being synchronized with the local oscillator and thus operating periodically from $t+\tau$ to $t+T_b+\tau$, which is comparable to the results of the previous section.

5.4 Simulation results

Based on the results presented in the previous chapter and the description in the previous sections, the communication model from section 2.4 is modified such that a time offset can be introduced in the LO. This time offset is derived from the specified frequency deviation due to the DFT.

$$\tau = \frac{\epsilon_f}{\beta} \tag{5.17}$$

Next, a number of bit error rate (BER) simulations are performed. Firstly, the time offset is based on the standard deviation of the error spreading, as has been explored in the previous chapter, thus $\sigma = 160$ Hz. Secondly, based on Equation 5.3, a time offset based on a DFT resolution of 512 $^{\text{Hz}}/_{\text{bin}}$ is select as the DFT limit. This corresponds to a 256 point DFT, as is concluded in the previous chapter, see subsection 4.3.5. Thus, where σ is based on statistics from multiple simulations, the DFT limit is the maximum deviation for one execution. The results of these simulations are depicted in Figure 5.1.



(a) Global performance overview

(b) Detailed view

Figure 5.1: BER performance due to a time shift between transmitter and receiver.

As expected, the performance degrades when the offsets increases. However, for the selected settings the change is small, a maximum of 1 dB loss. This can be explained by the fact that the frequency offset is relatively small compared to the BFSK frequencies.

Equations 5.9 and 5.10 may be rewritten such that these may be substituted into the definition of the chirp rate. As the frequency offset due to the timing error between transmitter and receiver, f_o, is Δf.

$$\tau = \lambda T_b \tag{5.9}$$

$$f_o = \Delta f = \frac{\rho}{T_b} \tag{5.18}$$

$$\frac{\beta\tau}{2} = \Delta f \tag{3.11}$$

$$\frac{1}{2}\beta T_b^2 = \frac{\rho}{\lambda} \tag{5.19}$$

The left hand side term of the last expression, Equation 5.19, shows similarity to the time-bandwidth product of a chirp signal, see Equation 2.50. Therefore this factor can also be seen as a time-bandwidth product, but now of the chirped message signal. In Figure 5.2 the simulation results for three of these products are plotted. The data rates correspond to bandwidth time products of 4, 1 and 0.25 for the data rates of 2 kHz, 4 kHz and 8 kHz.

Lowering the data rates and increasing the bit periods has a negative effect on the performance. This seems counterintuitive. However, in this work the BFSK message frequencies are deter-

mined by Sunde's formula, Equation 2.6:

$$f_m = \frac{n_c + m}{T_b} \quad n_c \in \mathbb{N} \tag{2.6}$$

Thus for lower data rates, the bit period increases, placing the message signals closer to each other. Leaving less margin for the allowable absolute frequency error, as any overlap between the message signals will cause communication errors.

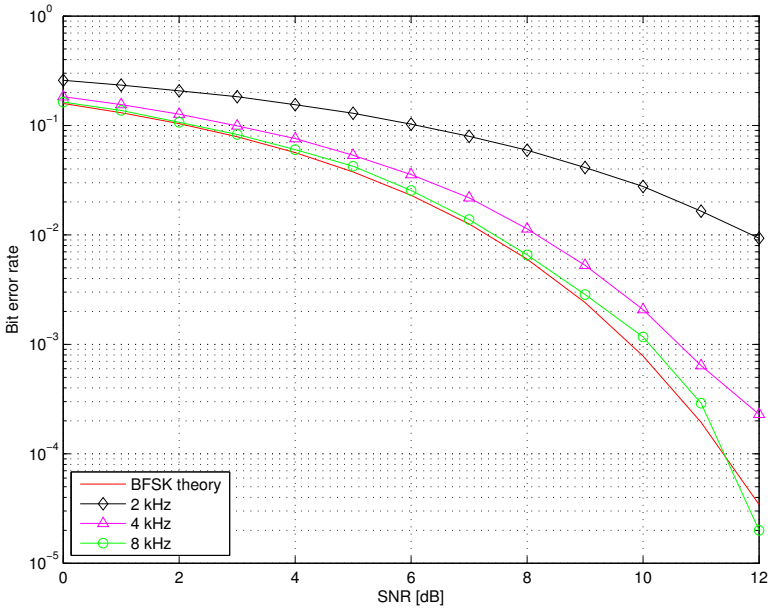


Figure 5.2: BER performance when varying the data rate.

CHAPTER 6

Conclusions & Recommendations

6.1 Conclusions

An algorithm has been derived and a high level system design has been proposed to extend the existing binary frequency shift-keying (BFSK) receiver, in order to perform coarse grain synchronization to a chirp carrier wave. Simulation results show that the combination of a 1 bit analog to digital converter (ADC) and a 256 point discrete Fourier transform (DFT) provide the best performance based on the design space exploration in chapter 4. It also shows that adding additional hardware does not increase performance in a significant manner.

Two detection schemes are proposed to translate the DFT output to a time offset. One scheme detects the strongest frequency tone in the negative and positive frequency spectrum separately, called positive negative detection (PND). The other scheme searches for the strongest tone in the complete spectrum, called maximum power detection (MPD). It is shown that the first method is not suitable for accurate synchronization. Alternatively, the MPD yields a performance of 90%. This means that in 90% of the iterations, any error falls within the systems resolution.

The final goal, to determine the impact of any synchronization error on the communication performance, is achieved by means of simulation. First is it shown that, for a data rate of 4 kHz, the signal to noise ratio (SNR) performance degrades with 1 dB in the bit error rate (BER) curve. Secondly, a link is made between the chirp carrier, the data rate and the synchronization error. Showing that the distance between the two message signal's frequencies is linked to the error made due to any frequency offsets. As these frequencies are inversely related to their bit periods, an increase of the data rate and thereby lowering the bit period, leads to higher frequencies and a larger bandwidth for the BFSK transmission while reducing the sensitivity to a frequency offset. This is observed by looking at the BER curves for different data rates when an equal synchronization error was introduced.

Finally, the main requirement stated that synchronization should be performed within one chirp period, where one chirp period consists of one up and one down period. The systems meets this requirement by sensing for half the period, therefore only the up chirp, such that the second half is available for calculation and correction. The suggested implementation allows a serial connection between the ADC and fast Fourier transform (FFT) such that the DFT computation may be performed while the signal is still sampled.

6.2 Recommendations

Based on the work done during this thesis the following recommendations have been compiled:

- Synchronization is performed on either an up or down chirp carrier and the option of using a combination of both is not recommended. However, the use of the combination of the two types is recommended for the communication part of the system. As the synchronization is not perfect, an offset in time will be present between the transmitter and receiver.

  When only up or down chirps are used, a transition period occurs when the chirp signal switches from it's high frequency boundary to it's lower boundary. During this transition and due to the time offset, the frequency difference will become equal to the difference of the chirp bandwidth and the frequency error which was previously present. In general this difference is too large for the modulation scheme to recover any data, thus communication during this transition period is not possible.

  One option is to increase the bit period such that it is larger than the transition period. The second option is to avoid this larger frequency difference by combining up and down chirps as the carrier signal. In such a case the frequency offset (ϵ_f) will actually vary between $\pm\epsilon_f$. This last option also preserves the separation between the carrier and the modulation scheme.

- The current design is based on the idea of one iteration to perform synchronization. Simulation results show that the errors made with the MPD method lie within two times the systems frequency resolution, raising the question if an additional iteration can resolve such errors.

  Preliminary simulation results have shown an increase of correct synchronizations from 90% to 96%, when one additional iteration is performed. A third iteration has no additional effect. The main question is therefore if this second iteration weighs up to the additional power requirements and time overhead against the (limited) increase in accuracy.

- The proposed detection schemes are based on the assumption that only one transmitter is active, thus detected peaks are always due to one transmitter. Depending on the usage of this module, multi-user access can be made possible by modifying the detection logic.

- A lowpass filter with a passband equal to the chirp bandwidth is required. In this work a fairly ideal filter is assumed, however in a realistic systems such a steep filter may not be desired to implement. However, changing the frequency response of the filter may affect the detectability of the synchronization tones. It is therefore worth investigating the effect of changes to the spectrum which is used to detect the synchronization tones.

- Implement the design to verify it's performance

APPENDIX A

Model information

A.1 ADC Headroom factor

The model of the multi-bit analog to digital converter (ADC) is an implementation of the model description in (Kokkeler [23]). This ADC uses the standard deviation of the input signal to place the decision levels of the quantizer. As this criteria is too strict, the headroom factor was introduced, such that the decision levels are placed at multiples of this standard deviation.

However, a method to determine this headroom factor was not presented. Therefore, in this work, a Monte Carlo simulation is used to determine this factor. This simulation performs the quantization of an input signal with additive white gaussian noise (AWGN) to determine the signal to quantization noise ratio (SQNR) of the quantizer for various headroom factors. The results are shown in Figure A.1.

In the synchronization simulations, a signal to noise ratio (SNR) of zero is used, which has a maximum SQNR when a headroom factor of 2.1 is used.

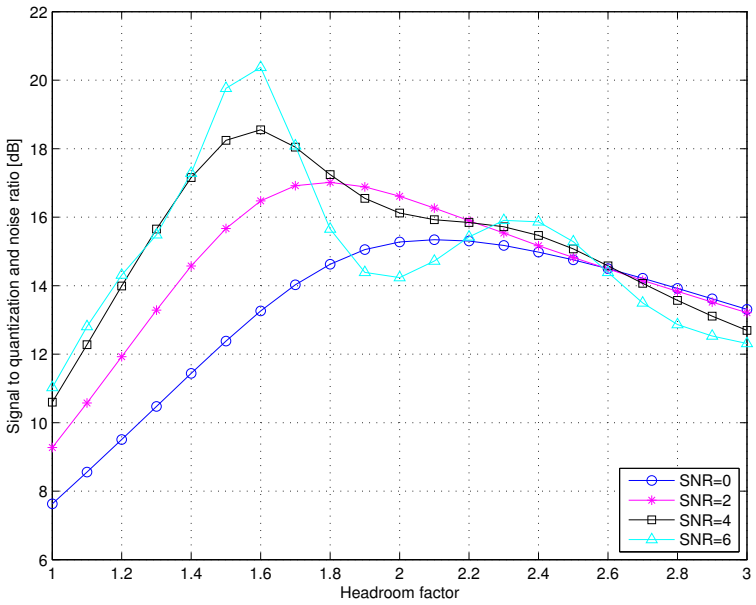


Figure A.1: Influence of the ADC's headroom factor on the quantizers SQNR for various SNR values

A.2 MATLAB

Various MATLAB models have been made for simulation purposes, this section presents a description on the organization and usage of these models.

A.2.1 Filenames

Roughly all files may be seen as either simulation files, results files and support files. Simulation files implement the Monte Carlo simulations and their filenames always starts with `sim_`, for simulator. The results of these simulation are automatically saved to MATLAB data files (`*.mat`). Other files are there for support, either used by the simulation files to perform some repeating functionality or the investigate phenomena, such as parameters influence.

A.2.2 Simulator structure

Each simulator is structured in the following manner:

- **Introduction**
  A comment block describing the purpose of this simulation, component models which have been used and the required support files.
- **Operation**
  Here parameters are defined which influence the simulation mode. For example, `nMode` lets the user run the simulation either once for functional analysis or in Monte Carlo mode for actual results.
- **Parameters**
  The parameters of the various components which are used are defined in these sections. Each component has a set of parameters, stored in a structure. Details are provided in the next subsection.
- **Validation**
  All parameters are validated against criteria, also listed in the next subsection, and may be altered to confirm with these criteria.
- **Helpers**
  Creation of variables which assist the simulator in it's operation, such as sample frequency, a time vector and other variables which do not change during simulation.
- **Init**
  Signal and component initialization.
- **Simulator**
  The actual simulator code, executing the model.
- **Results**
  Depending on the simulation mode, results are saved to a data file and plotted from here.

A.2.3 Simulator parameters

Name	Unit	Generated	Description	Criteria
`sType`		No	Channel type	AWGN
`nOS`		Yes	Additional power, due to oversampling	

Table A.1: Channel parameters

Name	Unit	Generated	Description	Criteria
nFstart	Hz	No	Lower frequency bound	$2^n n \in \mathbb{N}$
nBW	Hz	No	Bandwidth	$2^n n \in \mathbb{N}$
nBT		No	Bandwidth-time product	$2^n n \in \mathbb{N}$
ndBm	dBm	No	Signal strength	
sType		No	Chirp direction	Up, down or updown
nT	sec	Yes	Chirp period period	
dPc	W	Yes	Signal strength	
nBeta	$^{Hz}/_{sec}$	Yes	Chirp rate	

Table A.2: Chirp signal parameters

Name	Unit	Generated	Description	Criteria
nFstart	Hz	No	Lower frequency bound	$2^n n \in \mathbb{N}$
nBW	Hz	No	Bandwidth	$2^n n \in \mathbb{N}$
ndBm	dBm	No	Signal strength	
nTones		No	If defined, leads to the generation of a single tone	
nSamplesPerPhase		No	Number of samples before a phase change (per tone)	
dP	W	Yes	Signal power	

Table A.3: Interence signal parameters

Name	Unit	Generated	Description	Criteria
nFpass	Hz	No	End of the passband	
nFstop	Hz	No	Begin of the stop band	
nApass		No	Attenuation in the passband	
nAstop		No	Attenuation in the stop band	

Table A.4: Lowpass filter parameters

Name	Unit	Generated	Description	Criteria
sType		No	Modulation scheme	FSK
nM		No	Alphabet size	$2^n n > 1, n \in \mathbb{N}$
nFb	Hz	No	Symbol/Data rate	$2^n n \in \mathbb{N}$
ndBm	dBm	No	Signal strength	
nOffset		No	Frequency selection offset	$n \in \mathbb{N}$
dPb	W	Yes	Power per bit	
vFrequency	Hz	Yes	nM message frequencies	

Table A.5: Modulation parameters

Name	Unit	Generated	Description	Criteria
nSamplesSim		No/Yes	Number of samples to simulate in one iteration	$2^n n \in \mathbb{N}$
nSymbols		No/Yes	The minimal number of symbols to simulate	$n \in \mathbb{N}$
vSNR		No	Signal to noise ratios to simulate	
nTimeResolutionMin	sec	No	If provided, ensures a minimal time resolution	
nFs	Hz	Yes	Sample frequency	
nSampleSymbol		Yes	Number of samples per symbol	
nSamplesSim		Yes	Corrected number of samples per iteration	
nSymbols		Yes	Corrected number of symbols to simulate, original value is seen as the minimum	
nSymbolsSim		Yes	Number of symbols per iteration	
nIterations		Yes	Number of iterations	
time		Yes	Local time vector for one iteration	

Table A.6: Simulator parameters

Name	Unit	Generated	Description	Criteria
nBits		No	Quantizer resolution	$n \in \mathbb{N}$
nFs	Hz	No	Input signal's sample frequency	$2^n n \in \mathbb{N}$
nFs_adc	Hz	No	Ouptut signal's sample frequency	$2^n n \in \mathbb{N}$
bThermal		No	Enable thermal noise source	0 or 1
nTemperature	C	No	Environment temperature	$n \in \mathbb{R}$
bJitter		No	Enable jitter in the sample circuit	0 or 1
nJitterRatio		No	Time ratio between sample period and jitter period	$n \in \mathbb{R}$
nH		No	Headroom factor, multi-bit only	$n \in \mathbb{R}$

Table A.7: ADC parameters

Name	Unit	Generated	Description	Criteria
`bIdeal`		No	Switch between floating point (1) and fixed point operation (0)	0 or 1
`nOrder`		No	Number of points	$2^n n \in \mathbb{N}$
`nWLinput`		No	Input signal's word length	$n \in \mathbb{N}$
`nFLinput`		No	Input signal's fractional length	$n \in \mathbb{N}$
`nWLoutput`		No	Output signal's word length	$n \in \mathbb{N}$
`nFLoutput`		No	Output signal's fractional length	$n \in \mathbb{N}$
`nWLtwiddle`		No	Twiddle factor's word length	$n \in \mathbb{N}$
`nFLtwiddle`		No	Twiddle factor's fractional length	$n \in \mathbb{N}$

Table A.8: fast Fourier transform (FFT) parameters

APPENDIX B

Discrete multi-tone interferer

In this thesis a number of simulation were performed in which the effect of an interference band on the receiver was of interest. To generate this interference, a discrete multi-tone signal (Giorgetti et al. [11]) is used. As illustrated in Figure B.1, such a signal consists of multiple individual frequency components, tones, equally spaced within a specified frequency band.

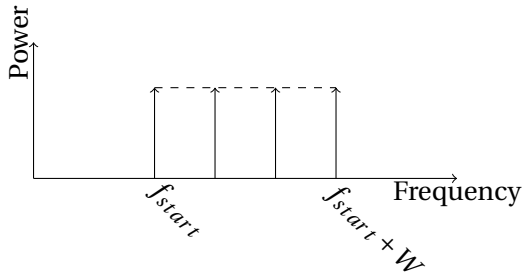


Figure B.1: Illustration of a discrete tone interferer

To generate such a signal, we start with a simple formulation:

$$i(t) = \sum_{M} Acos(2\pi f_m t + \phi_m) \tag{B.1}$$

reflecting that the interferer consists of a number, M, of individual tones, f_m, each with a uniformly distributed phase $[0, 2\pi)$. Each individual tone is defined as:

$$f_m = f_{start} + m\Delta f_i \tag{B.2}$$

As illustrated in Figure B.1, f_{start} is the frequency at which the interferer band starts and Δf_i is the frequency spacing between two tones. Since the generation is performed discrete, any tone must fit an integer times into the sample period to avoid undesired frequency components due to truncation. When N is the number of samples to generate, it must hold to the following constraint:

$$N = 2^k \quad k \in \mathbb{N} \tag{B.3}$$

combined with the sample time T_s, the frequency spacing is defined as:

$$T = NT_s \tag{B.4a}$$

$$= \frac{N}{F_s} \tag{B.4b}$$

$$\Delta f_i = \frac{F_s}{N} \tag{B.4c}$$

here T indicates the time represented by the total number of samples, and is inversely related to the smallest frequency which may be selected. Finally the number of tones is calculated via:

$$M = \frac{W}{\Delta f_i} \quad \in \mathbb{N} \tag{B.5a}$$

$$= \frac{NW}{F_s} \quad \in \mathbb{N} \tag{B.5b}$$

To insure that $M \in \mathbb{N}$, constraints must be placed on both W and F_s as follows:

$$W \in \mathbb{N} \tag{B.6}$$

$$F_s = 2^l W \quad l \in \mathbb{N} \tag{B.7}$$

$$k \geq l \tag{B.8}$$

the constraint in B.8 is explained by substituting both B.3 and B.7 in B.5.

To both normalize and set the signal power to P_i, the amplitude is defined as:

$$A = \sqrt{\frac{2P_{int}}{M}} \tag{B.9}$$

Resulting in the final formula:

$$i(t) = \sum_{m=0}^{M-1} \sqrt{\frac{2P_{int}}{M}} cos(2\pi(f_{start} + m\Delta f_i)t + \phi_m) \tag{B.10}$$

While it was not stated before, the lower frequency bound, f_{start} must also be a power of two integer.

Bibliography

[1] IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks - Specific Requirement Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). *IEEE Std 802.15.4a-2007 (Amendment to IEEE Std 802.15.4-2006)*, pages 1 –203, 2007. doi: 10.1109/IEEESTD.2007.4299496.

[2] V.J. Arkesteijn, E.A.M. Klumperink, and B. Nauta. Jitter requirements of the sampling clock in software radio receivers. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 53 (2):90 – 94, feb. 2006. ISSN 1549-7747. doi: 10.1109/TCSII.2005.856893.

[3] A. Berni and W. Gregg. On the utility of chirp modulation for digital signaling. *IEEE Transactions on Communications*, 21(6):748 – 751, jun 1973. ISSN 0090-6778. doi: 10.1109/TCOM.1973.1091721.

[4] N. Blachman. The intermodulation and distortion due to quantization of sinusoids. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 33(6):1417 – 1426, dec 1985. ISSN 0096-3518. doi: 10.1109/TASSP.1985.1164729.

[5] S. Boumard and A. Mammela. Robust and accurate frequency and timing synchronization using chirp signals. *IEEE Transactions on Broadcasting*, 55(1):115 –123, march 2009. ISSN 0018-9316. doi: 10.1109/TBC.2008.2008712.

[6] H.K. Boyapati and R.V. Raja Kumar. A comparison of dsp, asic, and risc dsp based implementations of multiple access in lte. In *4th International Symposium on Communications, Control and Signal Processing (ISCCSP), 2010*, pages 1 –5, march 2010. doi: 10.1109/ISCCSP.2010.5463464.

[7] N. Da Dalt, M. Harteneck, C. Sandner, and A. Wiesbauer. On the jitter requirements of the sampling clock for analog-to-digital converters. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 49(9):1354 – 1360, sep 2002. ISSN 1057-7122. doi: 10.1109/TCSI.2002.802353.

[8] F. Dowla, F. Nekoogar, and A. Spiridon. Interference mitigation in transmitted-reference ultra-wideband (uwb) receivers. In *IEEE Antennas and Propagation Society International Symposium, 2004.*, volume 2, pages 1307 – 1310 Vol.2, june 2004. doi: 10.1109/APS.2004.1330425.

[9] R. Dutta, A.B.J. Kokkeler, R.A.R. van der Zee, and M.J. Bentum. Performance of chirped-fsk and chirped-psk in the presence of partial-band interference. In *18th IEEE Symposium on Communications and Vehicular Technology in the Benelux (SCVT), 2011*, pages 1 –6, nov. 2011. doi: 10.1109/SCVT.2011.6101316.

[10] S. Ghissoni, E. Costa, J. Monteiro, and R. Reis. Combination of constant matrix multiplication and gate-level approaches for area and power efficient hybrid radix-2 dit fft realization. In *18th IEEE International Conference on Electronics, Circuits and Systems (ICECS), 2011*, pages 567 –570, dec. 2011. doi: 10.1109/ICECS.2011.6122338.

[11] Andrea Giorgetti, Marco Chiani, and Moe Z. Win. The effect of narrowband interference on wideband wireless communication systems. *IEEE Transactions on Communications*, 53(10):1773, oct. 2005. ISSN 0090-6778. doi: 10.1109/TCOMM.2005.857224.

[12] B. Goll and H. Zimmermann. A comparator with reduced delay time in 65-nm cmos for supply voltages down to 0.65 v. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 56(11):810 –814, nov. 2009. ISSN 1549-7747. doi: 10.1109/TCSII.2009.2030357.

[13] G.F. Gott and J.P. Newsome. H.f. data transmission using chirp signals. *Proceedings of the Institution of Electrical Engineers*, 118(9):1162 –1166, september 1971. ISSN 0020-3270. doi: 10.1049/piee.1971.0210.

[14] W. Gugler, A. Springer, and R. Weigel. A robust saw-based chirp- pi;/4 dqpsk system for indoor applications. In *IEEE International Conference on Communications, 2000.*, volume 2, pages 773 –777 vol.2, 2000. doi: 10.1109/ICC.2000.853604.

[15] W. Han, A.T. Erdogan, T. Arslan, and M. Hasan. High-performance low-power fft cores. *ETRI Journal*, 30(3):451–460, 2008.

[16] S.S. Haykin. *Communication systems*. Wiley, 1994. ISBN 9780471571766.

[17] Chengbing He, Qinwei Meng, Jianguo Huang, and Qunfei Zhang. A novel bandwidth efficiency chirp-based underwater acoustic communication. In *International Conference on Information Networking and Automation (ICINA), 2010*, volume 1, pages V1–531 –V1–534, oct. 2010. doi: 10.1109/ICINA.2010.5636762.

[18] M.B. Henry and L. Nazhandali. Hybrid super/subthreshold design of a low power scalable-throughput fft architecture. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5409 LNCS:278–292, 2009.

[19] Sami Hinedi, Marvin Simon, and Dan Raphaeli. Performance of noncoherent orthogonal m-fsk in the presence of timing and frequency errors. *IEEE Transactions on Communications*, 43(2 -4 pt 2):922–933, 1995.

[20] Kewu Huang, Ziyu Wang, and Ran Tao. Study of incoherent demodulation technique in chirp spread spectrum communication systems. In *9th International Conference on Signal Processing, 2008*, pages 1926 –1929, oct. 2008. doi: 10.1109/ICOSP.2008.4697519.

[21] Shen-Jui Huang and Sau-Gee Chen. A green fft processor with 2.5-gs/s for ieee 802.15.3c (wpans). In *International Conference on Green Circuits and Systems (ICGCS), 2010*, pages 9 –13, june 2010. doi: 10.1109/ICGCS.2010.5543105.

[22] Walt Kester. *Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor*. Analog Devices, Inc. URL `http://www.analog.com/static/imported-files/tutorials/MT-003.pdf`.

[23] A. B. J. Kokkeler. *Analog-Digital Codesign using Coarse Quantization*. PhD thesis, Faculty of Electrical Engineering, Mathematics and Computer Science, University of Twente, Enschede, The Netherlands, April 2005.

[24] Bin Le, T.W. Rondeau, J.H. Reed, and C.W. Bostian. Analog-to-digital converters. *IEEE Signal Processing Magazine*, 22(6):69 – 77, nov. 2005. ISSN 1053-5888. doi: 10.1109/MSP.2005.1550190.

[25] Xiangtao Li, Wei-Min Lance Kuo, Yuan Lu, R. Krithivasan, Tianbing Chen, J.D. Cressler, and A.J. Joseph. A 7-bit, 18 ghz sige hbt comparator for medium resolution a/d conversion. In *Proceedings of the Bipolar/BiCMOS Circuits and Technology Meeting, 2005*, pages 144 – 147, oct. 2005. doi: 10.1109/BIPOL.2005.1555220.

[26] M.S. Oude Alink, A.B.J. Kokkeler, E. Klumperink, K.C. Rovers, G. Smit, and B. Nauta. Spurious-free dynamic range of a uniform quantizer. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 56(6):434 –438, june 2009. ISSN 1549-7747. doi: 10.1109/TCSII.2009.2020929.

[27] M.J.M. Pelgrom. *Analog-to-Digital Conversion*. Springer, 2010. ISBN 9789048188871.

[28] John Pinkney. *Low Complexity Indoor Wireless Data Links using Chirp Spread Spectrum*. PhD thesis, The University Of Calgary, 2003.

[29] J.Q. Pinkney, A.B. Sesay, S. Nichols, and R. Behin. A robust high speed indoor wireless communications system using chirp spread spectrum. In *IEEE Canadian Conference on Electrical and Computer Engineering, 1999*, volume 1, pages 84 –89 vol.1, may 1999. doi: 10.1109/CCECE.1999.807175.

[30] J.G. Proakis and M. Salehi. *Digital communications*. McGraw-Hill, 2008. ISBN 9780072957167.

[31] J.H. Reed. *Software Radio: A Modern Approach to Radio Engineering*. Prentice Hall Communications Engineering and Emerging Technologies Series. Prentice Hall, 2002. ISBN 9780130811585.

[32] M.A. Richards, J.A. Scheer, and W.A. Holm. *Principles of Modern Radar: Basic Principles*. Principles of Modern Radar. SciTech Publishing, 2010. ISBN 9781891121524.

[33] M. Roberton and E.R. Brown. Integrated radar and communications based on chirped spread-spectrum techniques. In *IEEE MTT-S International Microwave Symposium Digest, 2003*, volume 1, pages 611 – 614 vol.1, june 2003. doi: 10.1109/MWSYM.2003.1211013.

[34] M.K. Simon. On the bit-error probability of differentially encoded qpsk and offset qpsk in the presence of carrier synchronization. *IEEE Transactions on Communications*, 54(5): 806 – 812, may 2006. ISSN 0090-6778. doi: 10.1109/TCOMM.2006.874002.

[35] A. Springer, W. Gugler, M. Huemer, L. Reindl, C.C.W. Ruppel, and R. Weigel. Spread spectrum communications using chirp signals. In *IEEE/AFCEA EUROCOMM 2000. Information Systems for Enhanced Public Safety and Security*, pages 166 –170, 2000. doi: 10.1109/EURCOM.2000.874794.

[36] A. Suleiman, A. Hussein, K. Bataineh, and D. Akopian. Scalable fft architecture vs. multiple pipeline fft architectures - hardware implementation and cost. In *IEEE International Conference on Systems, Man and Cybernetics, 2009.*, pages 3792 –3796, oct. 2009. doi: 10.1109/ICSMC.2009.5346639.

[37] Jia Sun and Hao Liu. An improved synchronization method of chirp uwb. In *ISECS International Colloquium on Computing, Communication, Control, and Management, 2008. CCCM '08.*, volume 1, pages 654 –658, aug. 2008. doi: 10.1109/CCCM.2008.230.

[38] Ywh-Ren Tsai and Jin-Fu Chang. The feasibility of combating multipath interference by chirp spread spectrum techniques over rayleigh and rician fading channels. In *IEEE Third International Symposium on Spread Spectrum Techniques and Applications, 1994*, pages 282 –286 vol.1, jul 1994. doi: 10.1109/ISSSTA.1994.379577.

[39] W. van Etten. *Introduction to random signals and noise*. Wiley, 2005. ISBN 9780470024119.

[40] Xiaowei Wang, Minrui Fei, and Xin Li. Performance of chirp spread spectrum in wireless communication systems. In *11th IEEE Singapore International Conference on Communication Systems, 2008. ICCS 2008.*, pages 466 –469, nov. 2008. doi: 10.1109/ICCS.2008.4737227.

[41] M.Z. Win, P.C. Pinto, and L.A. Shepp. A mathematical theory of network interference and its applications. *Proceedings of the IEEE*, 97(2):205 –230, feb. 2009. ISSN 0018-9219. doi: 10.1109/JPROC.2008.2008764.

[42] Kuai Yin, Qiao Meng, Kai Tang, and Yi Zhang. A 20gs/s low-power bicmos comparator using an active inductor load. In *4th International High Speed Intelligent Communication Forum (HSIC), 2012*, pages 1 –4, may 2012. doi: 10.1109/HSIC.2012.6212984.