Master's Thesis

Instantaneous Channel Access for 3G-ALE Systems

Andreé Bengtsson

Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, August 2014.

Instantaneous Channel Access for 3G-ALE Systems

Andreé Bengtsson
Master of Science in Engineering
Engineering Physics
atf09abe@student.lu.se

Combitech
Ljungadalsgatan 2B
351 80 Växjö
Sweden

Advisor: Håkan Bergzén

August 31, 2014

Printed in Sweden
E-huset, Lund, 2014

Abstract

The Swedish armed forces HF radio systems use the 3G-ALE standard for automatic link establishment. Even though the standard defines a system that is faster and yielding better performance than its predecessor, the 2G-ALE standard, there are still situations in which an even shorter link establishment time is required, especially considering time critical messages. The standard uses a scanning mechanism that causes a latency in the system. By removing the need for the scanning mechanism and still be able to handle multiple channels the substantially latency can be decreased. This can be accomplished by using a parallel receiver that receive on all the channels in the channel group.

This thesis is written to examine the possibilities for a future implementation. A first conceptual prototype is also developed to show the advantages of this technique.

Acknowledgments

First I would like to express my very great appreciation to my advisor, Håkan Bergzén. His guidance and technical support has been highly valuable for this thesis. I would like to thank Rickard Berg for his constructive comments on my work. I wish to thank professor Ove Edfors for taking the time to be the examiner of this thesis. Many thanks to the staff at Combitech for assistance and for making me feel welcome to the company. Many thanks to friends, family and everyone that encourage me along the way and that made this thesis possible. I also wish to thank Malin Henriksson for her support and encouragement through my studies and particular this project.

Table of Contents

1 Background **3**
- 1.1 HF radio in the Swedish armed forces . . . 3
- 1.2 HF propagation modes . . . 5
- 1.3 Link Establishment . . . 7
- 1.4 Sideband modulation . . . 8

2 Aim **11**

3 Design/architecture **13**
- 3.1 Basic structure . . . 13
- 3.2 Extended structure . . . 13

4 Realization **17**
- 4.1 RF front-end . . . 17
- 4.2 Detector . . . 20
- 4.3 Decision . . . 21
- 4.4 Software Defined Radio . . . 21

5 Implementation of the detector **25**
- 5.1 Requirements . . . 25
- 5.2 MATLAB implementation . . . 27
- 5.3 Simulink implementation . . . 30

6 Result **37**
- 6.1 MATLAB implementation . . . 37
- 6.2 Simulink implementation . . . 40

7 System description **57**
- 7.1 Protocol . . . 57

8 Conclusion/Future work **61**

References **63**

List of Figures

1.1 Scanning mechanism for 1 detector 4
1.2 Burst waveform 5 timing 5
1.3 HF propagation modes 6
1.4 Carrier sense multiple access-collision avoidance 7
1.5 Constellation diagram 8-PSK 8
1.6 Double/single sideband-suppressed carrier modulation 9

2.1 Communication network 12
2.2 Scanning mechanism for multiple detectors 12

3.1 Basic structure of the receiver 13
3.2 Extended structure of the receiver 14
3.3 Structure of the parallel receiver 14
3.4 Structure of the complete receiver 14

4.1 Illustration of a basic Tayloe detector 18
4.2 Generalized functional architecture of a SDR 23

5.1 Illustration of a circular buffer 27
5.2 Carrier modulation 28
5.3 Square root raised cosine filter 29
5.4 Experimental setup to record input signals 30
5.5 Frequency components in transmitted signal 31
5.6 Illustration of the cross-correlation 35
5.7 Experimental setup to receive calls 35
5.8 Experimental setup for the parallel receiver 36

6.1 Reference signal before and after sampling 37
6.2 Input signal & the cross-correlation detection 38
6.3 Reference and sampled signal 40
6.4 Sample signal with varying offset 41
6.5 Frequency response for PSDR Tayloe detector 43
6.6 Frequency offset error 44
6.7 Frequency offset compensation 45

6.8 BAF convergence . . . 46
6.9 Simulink model . . . 46
6.10 Squelch detector block . . . 47
6.11 Signal processing block . . . 48
6.12 Channel combination block . . . 48
6.13 Frequency offset estimation block . . . 49
6.14 Cross-correlation block . . . 49
6.15 Frequency compensation block . . . 49
6.16 Prevent division by zero block . . . 50
6.17 Customized C-code blocks . . . 50
6.18 Starting the executable program . . . 51
6.19 Receive a call . . . 51
6.20 Threshold value . . . 53
6.21 Parallel receiver model . . . 54
6.22 RF front-end . . . 55
6.23 Receive a call on channel 2, 3, and 1 . . . 55

7.1 Description of the entire communication system . . . 57
7.2 Customized protocol, receiver-NCT . . . 58
7.3 Customized protocol, message . . . 58
7.4 Customized protocol, NCT-receiver . . . 59
7.5 Customized protocol 2, NCT-receiver . . . 60

List of Tables

4.1 Specifications of PSDR module 18
4.2 Specifications of PGEN45 and PGEN170 19
4.3 Specifications of R820T 19
4.4 Specifications of FunCube Pro+ 20

5.1 Linking probability requirements 26
5.2 Channel conditions 26
5.3 Samples used to represent the TLC, preamble and data sequences in BW5 32
5.4 BAF algorithm 33

6.1 Performance measurements MATLAB model 39
6.2 Cross-correlation values for false calls 39
6.3 Verification of PGEN45 specifications 42
6.4 Bandwidth PSDR 42
6.5 Content of "Model Header" and "System Outputs" 51
6.6 Performance measurements Simulink model 52

Preface

This thesis is inspired from a paper written by Rickard Berg and Håkan Bergzén named "Instantaneous Channel Access for 3G-ALE Systems"[1]. Their paper describes a problem present in the third generation HF radio establishment standard. In the paper an idea is given on how to solve this problem together with a basic design concept.

Chapter 1

Background

The purpose of this chapter is to give the reader a background to the thesis. How does the current system works and what technique is used? The answers to these questions are meant to give the reader a basic knowledge for the next chapters.

1.1 HF radio in the Swedish armed forces

1.1.1 HF 2000 and Radio 1512

In 2007 the Swedish armed forces deployed a HF radio system named HF 2000 [1]. The communication on the HF band can be used for both short and long propagation ranges. HF systems enable global communication with international acting units, moving units and units that cannot be reached with other communication means (e.g. VHF/UHF or SATCOM). The HF 2000 system also enables communication between ground-, sea-, and air units. HF 2000 follow international standard for HF communications which allow interoperability with units from other nations, see [2] for further details.

Another 3G-ALE based system in use by the Swedish armed forces is Radio 1512. This is a manpack radio mainly used by the army, either carried and operated by soldiers on the field or installed in vehicles. It is mainly used for tactical communications on distances not covered by the combat net modes operating in the VHF/UHF bands, see [3] for more information.

1.1.2 The ALE standard

ALE stands for automatic link establishment. The first generation of ALE systems are the various property systems developed by HF radio manufacturers in the early 1980's when affordable and small microcontrollers became available. However, none of these systems were able to communicate a system from another manufacturer. The US Department of Defence identified this problem and started a study which resulted in the first ALE standard in 1986, MIL-STD-188-141. This is commonly referred to as the 2G-ALE standard. The 2G-ALE standard defines a simple and robust system that became a huge success. Today there exist several thousands of installations used by almost all western armed forces, see [4] for further details.

Both HF 2000 and Radio 1512 use STANAG 4538 Fast Link set-Up for link establishment and STANAG 5066 or STANAG 4538 xDL for traffic forwarding. In the STANAG 4538 standard (also called the 3G-ALE standard) all radio stations scan the frequencies in phase, see [1] for further information.

The advantages of the 3G-ALE mode is the short link establishment time, compared to the 2G-ALE standard. Also the mode is more robust and provides more channels to choose from.

The 3G-ALE mode in the HF 2000 system uses several frequencies (channels) where it can transmit data. The choice of channel depends on the type of communication, if it is short/long range etc. The transmitter and receiver scan these channels synchronously. The transmitter chooses the best suitable channel and check if it is occupied, see Section 1.3. If the channel is not in use the transmitter wait for that frequency to appear in the scanning cycle and make the call in the next possible time slot. The 3G-ALE mode in the HF 2000 system will change the used frequencies in the channel group during the day as the channel condition changes. This approach makes the system more dynamic compared to a fixed frequency or a traditional frequency hopping system. As the fixed frequency system only uses one stationary frequency it is vulnerable to changes in the channel conditions. Also a traditional frequency hopping system is, compared to the 3G-ALE mode, relatively vulnerable as the channel group always consists of the same frequencies. Generally frequency hopping systems are not as effective on HF (compared to VHF/UHF) because of modem equalization forces the hop-rate to be slow, usually below 10 channels/second and the hop bandwidth must be sufficiently small to ensure similar propagation conditions.

In a HF 2000 radio network employing the 3G-ALE mode all the nodes will update the channel group synchronously. As the network scan each channel for 1.35 seconds (dwell period) the link establishment time has potential to be considerably long even though the 3G-ALE mode is relatively fast.

The 3G-ALE standard utilize a frequency changing method that scans through a given channel group. The channel group in the 3G-ALE standard can consist of up to 64 channels. This group of channels are frequently updated in HF 2000 as the conditions changes, see Section 1.2. Today the system uses one mixer to down-convert from the RF frequency; and consequently it only looks into one channel at each time slot. When the transmitter shall send a call it has to wait until the scanning process reaches a suitable channel. The scanning mechanism can be seen in Figure 1.1, further details can be found in [2].

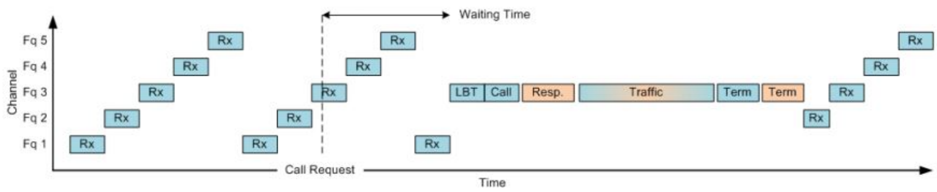

Figure 1.1: The system scanning through 5 channels and receive a call on the third channel[1].

To establish the link to another station the system uses the Burst Waveform 5 (BW5) in the STANAG 4538 standard. The BW5 timing can be seen in Figure 1.2. The Transmit Level Control (TLC) sequence gives the transmitter and the receiver opportunity to find a steady state before the preamble sequence is received. This includes for instance frequency compensation and gain control. After the TLC sequence follows the preamble sequence which gives the receiver opportunity to detect the presence of the waveform. The preamble used in the BW5 is unique for this waveform. Thereafter follows the data sequence, a 50-bit data package that contains source and destination addresses. It also includes a couple of other parameters essential for the link setup, see [5] for further details.

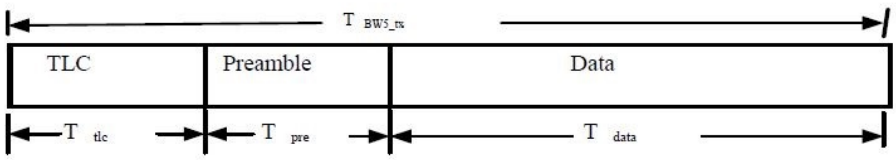

Figure 1.2: BW5 timing where $T_{tlc} = 106.667$ ms, $T_{pre} = 240$ ms and $T_{data} = 666.667$ ms which make the total time $T_{BW5\_tx} = 1013.333$ ms [5].

The burst waveform is modulated using an 8-ary PSK serial tone modulation with a carrier of 1800 Hz and a symbol rate of 2400 symbols/second. The modulated TLC, preamble and data sequence each consists of 256, 576 and 1600 PSK symbols. This will give the duration of each part shown in Figure 1.2. The total transmission time of 1.013 seconds is sufficiently small to fit into the channel dwell time of 1.35 seconds.

1.2 HF propagation modes

One unique property of wave propagation on the HF-band is the probability of using three different propagation modes. These are ground wave, ionospheric wave and Near Vertical Incidence Skywave (NVIS). The use and combination of these different modes enables global communication which is a mayor advantage of the HF-band.

1.2.1 Ground wave

This propagation method transmits waves on the ground level. The attenuation increases with frequency. Also the ground will affect the wave in different ways depending on the structure of the landscape and the material present in the ground. These implications do not make the ground wave suitable for long distance communication on the HF-band. Nevertheless the mode can be used for short distance communication.

1.2.2 Ionospheric wave (skywave)

The mode uses the property of refraction in the ionosphere and is the most common mode. If the conditions are right this enables the wave to bounce back to the earth, called a single hop. Extremely long propagation distances can be reached as the signal bounce between the earth and the ionosphere several times (multihop). The presence of this wave is an important contribution to transcontinental communication. One difficulty with this mode is the time dependency. The properties of the skywave vary over time depending on several things. As the ionization is caused by solar radiation the properties changes as the distance between the earth and sun varies during the year. The daily changes are also significant as the earth rotate, the sun will not radiate on the whole surface of the earth at the same time. The properties also changes when the number of sunspots varies. This phenomenon has a periodicity of 11 years.

1.2.3 Near Vertical Incidence Skywave

In some cases there exist areas that are not reached using the two modes described in Subsection 1.2.1 and 1.2.2. This area is called the skip-zone and is illustrated in Figure 1.3. Additional antennas are used to radiate in a near vertical direction. This way skywaves can be accomplished on shorter distances that covers the skip-zone. Further details can be found in [2].

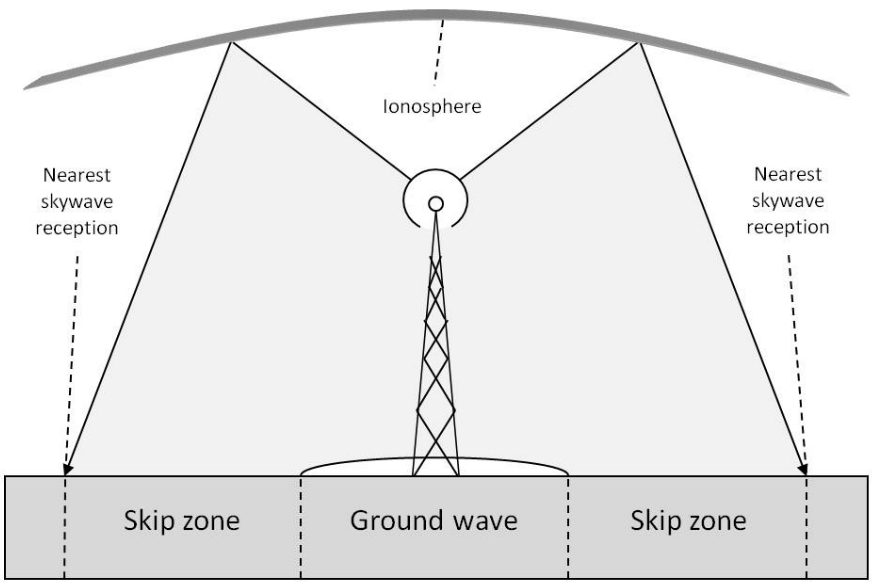

Figure 1.3: Description of the skip zone.

1.3 Link Establishment

The 3G-ALE mode uses Carrier Sense Multiple Access (CSMA) to search for suitable channels. In the basic CSMA protocol feedback from the receiver is used to determine if the channel is occupied or not. When the node request to make a call it will first listen to the channel and try to determine whether it is occupied or not. This procedure is called "Listen-Before-Transmit" (LBT). If the channel seems to be free transmission can start. This procedure is called non-confirmed as the calling station does not know if the to-be called station is in use or not.

The 3G-ALE standard provide the possibility to also use a add-on to the CSMA called Collision Avoidance (CA). This procedure make a two-way handshake between the stations. A "Link Set Up Request" is made to the to-be called station which response with a "Link Set Up Confirm" if the request is received. Thereafter the transmission can start with lower possibility to interfere with another call. This procedure is called confirmed as the handshake has been made. For further information, see [2]. Please refer to Figure 1.4 for a illustration of the terms carrier sense and collision avoidance.

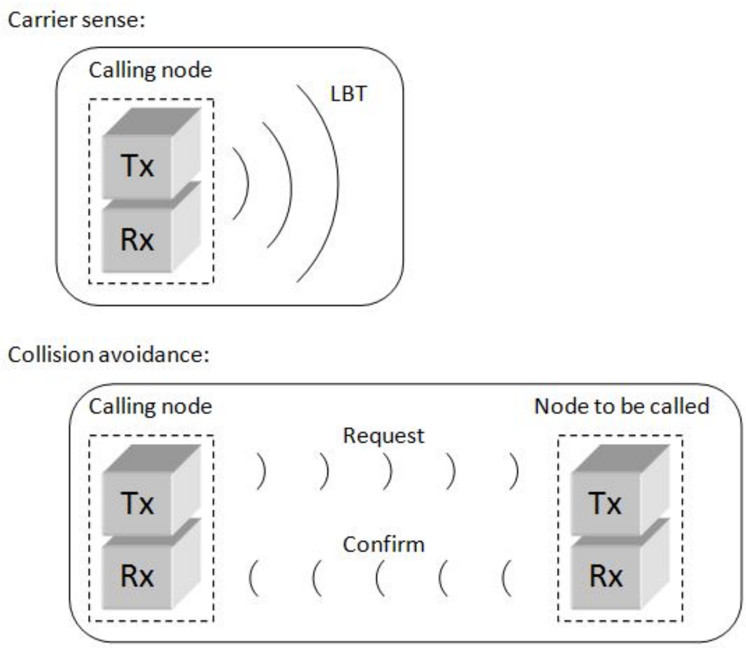

Figure 1.4: Description of the terms carrier sense and collision avoidance.

1.4 Sideband modulation

1.4.1 Sub-carrier

As described in Subsection 1.1.2 the BW5 is modulated with a 8-ary PSK. When the symbols are modulated using an 8-PSK symbol mapper the result is a complex representation of the signal corresponding to points in the constellation diagram. The symbol mapper places the symbols in a uniformly spaced manner on the unit circle, see Figure 1.5. As stated in the figure the PSK modulation does not has a offset, i.e. the first symbol is located at 0 radians. Also the symbols are binary coded as also can be seen in the figure.

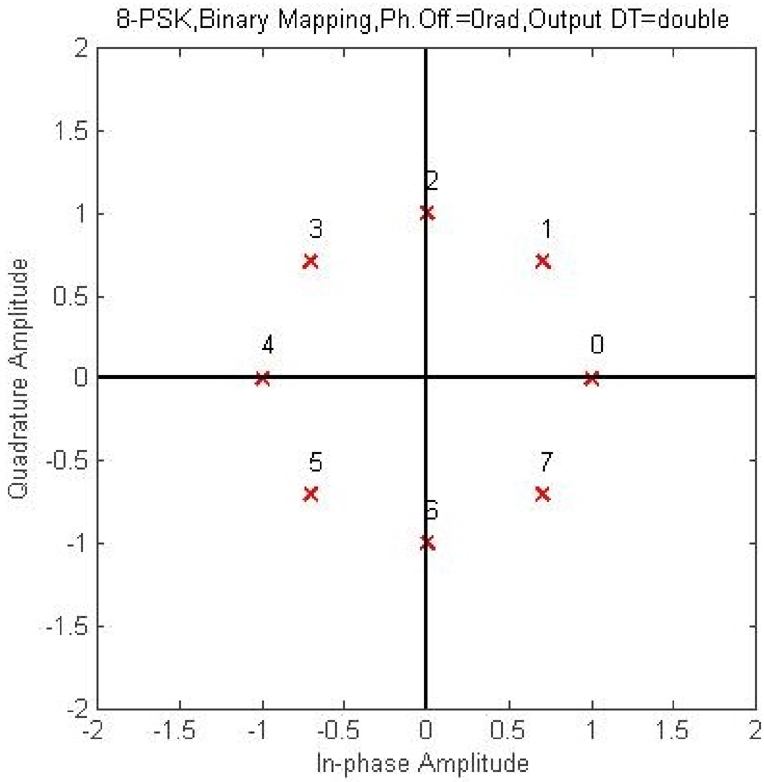

Figure 1.5: Constellation diagram of 8-PSK modulated baseband signal.

The in-phase and quadrature component is divided up and individually multiplied by cosine and sine to form $I(t)cos(\omega_c t)$ and $Q(t)sin(\omega_c t)$ where $\omega_c = 1800$ Hz as described in Subsection 1.1.2. These quantities are then combined to form the double-sideband modulated signal, see Equation (1.1) and Figure 1.6a. This means that the information in the signal is transmitted in both the upper and the lower sideband. The signal also have its carrier suppressed and finally forms a double-sideband-suppressed carrier (DSB-SC) modulation.

$$s(t) = I(t)cos(\omega_c t) - Q(t)sin(\omega_c t) \tag{1.1}$$

1.4.2 HF carrier

A HF frequency is added to the DSB-SC modulated sub-carrier signal. This is done using a single-sideband-suppressed carrier modulation, see Equation (1.2) and Figure 1.6b,c. $\hat{s}(t)$ is in this case the Hilbert transform of the signal in Equation (1.1). Equation (1.2) will form the SSB-SC upper sideband (USB) modulation, see [6] for further details. Note that the sign of the Hilbert transform may vary between different definitions.

Compared to the double sideband modulation the SSB-SC modulation has outstanding power and bandwidth efficiency. Refer to [7] for more information.

$$s_{HF}(t) = s(t)cos(\omega_{HF} t) - \hat{s}(t)sin(\omega_{HF} t) \tag{1.2}$$

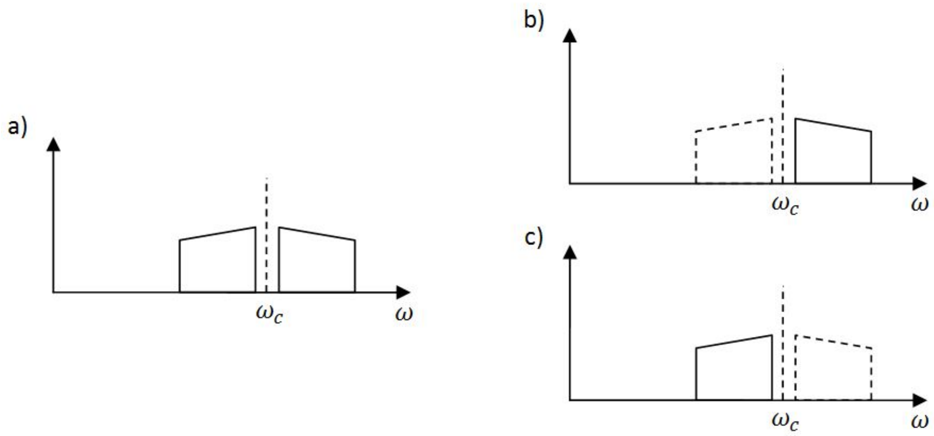

Figure 1.6: a) Double sideband-suppressed carrier (DSB-SC) modulation and single sideband-suppressed carrier modulation (SSB-SC) for b) upper sideband (USB) and c) lower sideband (LSB).

Chapter 2

Aim

This chapter shall give the reader an idea of the problem and how to solve it. The proposed idea is also briefly described.

The 3G-ALE standard is a powerful technique developed to perform fast automatic link establishment in a scanning system. Even though the standard gives short link establishment times it is still desirable to make it even faster. The scanning mechanism make the system slower compared to a fixed frequency system. For a call to be made the transmitter need to wait until the desired channel is being scanned as shown in Figure 1.1. This introduce a waiting time in the system that in the worst case is proportional to the number of channels in the channel group. As described in Subsection 1.1.2 there can exist up to 64 channels in a channel group. Using the dwell period of 1.35 seconds will, in such network, result in a worst case waiting time of 86.4 seconds. Even though this is the absolute worst case it is still a problem in some networks. For time-critical commands and control messages this latency can be a major problem.

This work is based on an idea on how to improve the 3G-ALE standard to achieve close to instantaneous link establishment. Note that the idea is still interoperable with existing 3G-ALE systems [1].

Figure 2.1 shows a network consisting of three transmitters and one parallel receiver. The idea is to use several detectors that listen for calls. If a call is made the detectors will inform the receiver and hand over the data traffic to this device. Another possibility is to integrate the parallel detectors and the receiver.

The purpose of using several detectors on the receiver side is to increase the number of simultaneously covered channels in the scanning procedure. Using for instance three channels and three detectors will result in full coverage and thus eliminate the need to scan through the channels. Each detector will listen to one channel and this channel will remain the same as long as the channel group is unchanged. This approach will give the best possible performance in the sense of speed. For such a system the transmitter can make a instant call on a empty channel. In the future it is desirable for such a receiver to be able to handle up to 64 channels, which is supported by HF 2000 as stated in Subsection 1.1.2, simultaneously.

If it is too demanding to implement 64 parallel receivers a compromised solution is proposed. Consider a frequency hopping system with five channels and three detectors might look like in Figure 2.2. The three detectors scan the chan-

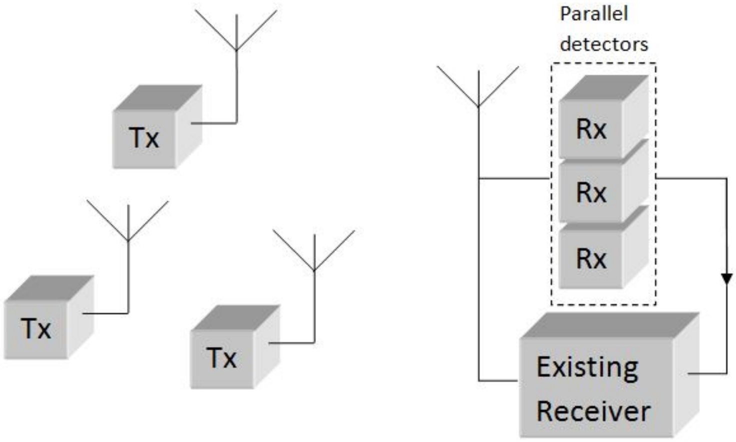

Figure 2.1: Example of communication system including several nodes.

nels in a synchronous manner and scan three channels at each time slot. If the transmitter want to make a call it will have to wait until one of the three detectors scan the desired channel. This will in the mean reduce the waiting time needed to make the call.

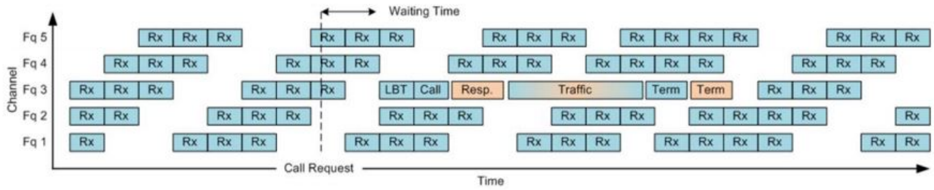

Figure 2.2: Scanning behaviour for a parallel receiver using 3 detectors and 5 channels [1].

Chapter 3

Design/architecture

The purpose of this chapter is to give the reader a basic understanding of the structure of the developed system. This chapter does not present any details about the implementation rather than the fundamental ideas in the system.

3.1 Basic structure

A system shall be developed that can detect incoming calls. Let us consider a fundamental structure of such a system shown in Figure 3.1. This structure consists of blocks that represents basic operations in the system. The first block is the RF front-end that is used to down-convert the RF frequency to a baseband frequency. This is followed by a detector that will search for the desired signal. A decision block is then used to interpret the outcome from the detector and decide whether a call has been made or not.

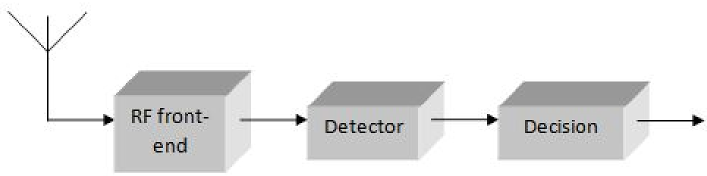

Figure 3.1: Basic structure of the receiver.

3.2 Extended structure

During the thesis some modification needed to be made to handle one phenomenon. The output from the RF front-end consist of a minor frequency error that needs to be taken care of. This can be handled using a frequency compensation block which corrects the carrier frequency of the incoming signal or adapt the carrier frequency of the reference signal. The signal is then handled as described in the basic structure. A schematic of such a system can be studied in Figure 3.2.

While this structure works it needs to be further modified to handle several detectors as a parallel receiver. This is basically done using several systems in

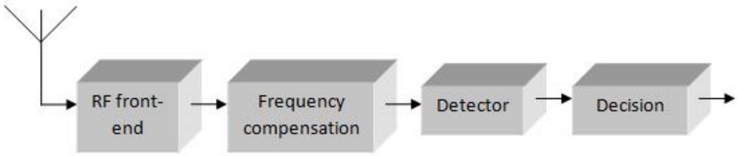

Figure 3.2: Extended structure of the receiver.

parallel, i.e. two detectors requires two set of each block as shown in Figure 3.3. For more detectors, just add more rows of blocks in parallel to the others. This way the structure of a parallel receiver is created. The RF front-end of each receiver will have the same HF input signal. Each of the receivers will extract a channel from the RF input signal to scan this one for calls.

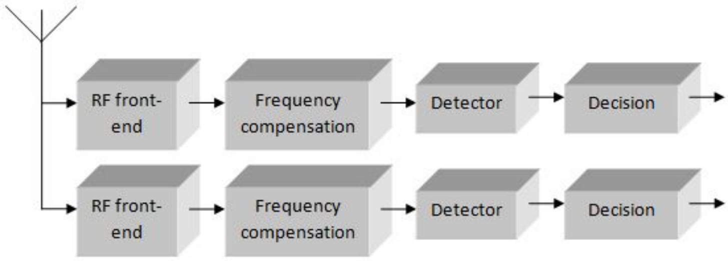

Figure 3.3: Structure of the parallel receiver.

To be able to interact with the regular 3G-ALE system a communication block is added at the end, see Figure 3.4.

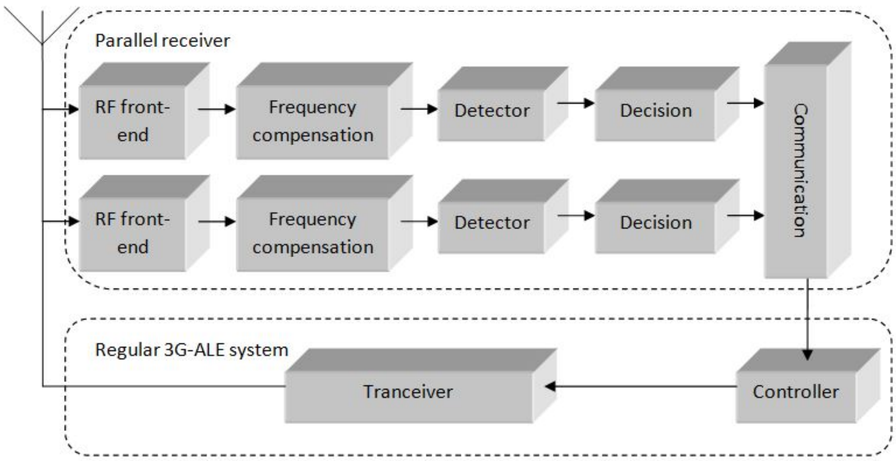

Figure 3.4: Structure of the complete receiver.

At start-up the 3G-ALE controller will inform the parallel receiver of the frequencies in the scan group. When the parallel receiver detects an incoming call it will inform the regular 3G-ALE system which will take over and handle the call.

Chapter 4
Realization

This chapter will continue the discussion of the blocks described in Chapter 3, but now more specific to this thesis, covering various options and the reasons for the selections that were made.

4.1 RF front-end

The RF front-end will handle the incoming signal and down-convert it from its HF carrier. This requires accurate and stable devices as the receiver shall be able to detect small changes in the signal condition. Depending on the requirements there exist some possible choices to consider. The requirements in this case should be a low cost RF front-end that can give an accurate and stable output. Also the bandwidth of the device should include the frequency range used by 3G-ALE systems, which is 2-30 MHz [2]. Another requirement is the physical size of the down-converter. It is preferable to have a small device as it shall be able to fit into confined spaces.

4.1.1 Tayloe detector

One possible implementation of the RF front-end is the Tayloe detector. This device uses a simple circuit to effectively extract the I and Q components of the incoming HF signal. By using a switch that operates at a frequency 4 times larger than the desired detection frequency (up to 120 MHz) the device can extract the baseband signal through direct down-conversion. Such a switch needs to be driven by a signal generator or similar feeding the Tayloe detector with a stable signal. The I and Q components are derived from the baseband signal using two operational amplifiers, see Figure 4.1. The bandwidth of the Tayloe detector is set by the parameters R and C. For further details, refer to [8].

PSDR

One implementation of the Tayloe detector is the PSDR module that can be found on the amateur radio market. The frequency range of the input signal to the PSDR module is 50 kHz - 24 MHz which almost includes the desired range. Although

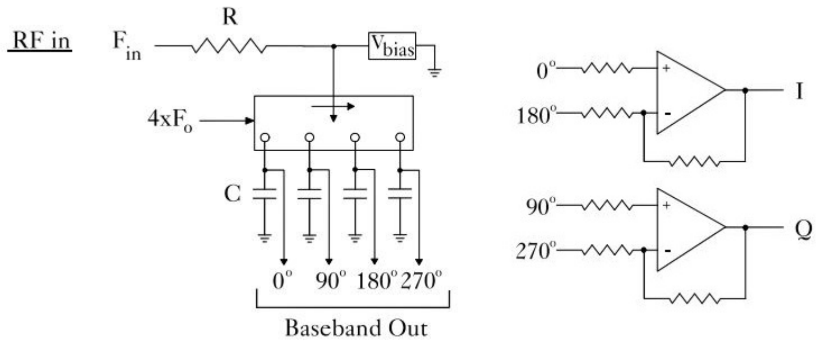

Figure 4.1: Illustration of a basic Tayloe detector [8].

Table 4.1: Specifications of PSDR module [9].

Feature	PSDR
Connection	RF Input: BNC Output: 3.5mm stereo
Price	26.69€

the whole range is not supported this module might be used in a proof-of-concept implementation. Further specifications of the module can be seen in Table 4.1.

The PSDR module has the possibility to control the switch using an external voltage controlled oscillator (VCO) [9]. This also set requirements on the oscillator and its performance. A low cost alternative from the same manufacturer is the PGEN45 or the PGEN170, see Table 4.2. The PGEN45 outputs a frequency covering the lower part of the desired frequency range while the PGEN170 covers the upper part of the range. In general the PGEN170 has better performance when looking at the frequency stability, temperature stability and duty cycle variation. As described in Subsection 4.1.1 the VCO needs to operate at a frequency 4 times the RF frequency. Unfortunately none of these VCO's covers the whole desired frequency range.

4.1.2 Other techniques

Besides the simple Tayloe detector there exists a couple of other alternatives. An attractive technique is to use a direct conversion receiver, implemented on a single integrated circuit, as the front-end. There exists such receivers on the market in the size of a USB stick with USB connection. Two types of USB dongles are described below and evaluated according to requirements of this thesis.

RTL-SDR

The RTL-SDR is a low cost SDR using a DVB-T TV tuner dongle. According to [10] the generally best performing RTL-SDR, and the cheapest one is the Rafael

Table 4.2: Specifications of two VCO's [11][12].

Feature	PGEN45	PGEN170
Operational frequency range	8.3 kHz - 45 MHz	17-170 MHz
Desired frequency range	8-120 MHz	8-120 MHz
Frequency stability	$< \pm 1.5\%$ (5 kHz - 20 MHz)	$< \pm 0.5\%$ (17-170 MHz)
Duty cycle Variations	50% ±1% (1 kHz - 2 MHz) ±5% (2-20 MHz)	50% ±2.5%
Temperature stability	40 ppm/°C	20 ppm/°C
Connection	BNC	BNC
Price	23.77€	25.23€

Table 4.3: Specification of the R820T [10][13][14][15][16].

Feature	Nobu Saito R820T RTL-SDR	KN0CK Direct Sampling Receiver
Operational frequency range	24-1766 MHz	0.5-54 MHz
Desired frequency range	2-30 MHz	2-30 MHz
Frequency stability	±2 ppm	-
Bandwidth	2.4 MHz*	-
Connection	Input RF: MCX Output: USB	Input RF: PAL Output: USB
Price	67.77€	60€

*Maximum sample rate without dropped samples, otherwise 3.2 MHz.

Micro R820T. Unfortunately it does not support our desired frequency range nor has a good enough frequency stability. The specifications of two modified devices can be read in Table 4.3. This shows specifications on the RTL-SDR modules developed by KN0CK and Nobu Saito. See further details in [14] and [15].

The RTL-SDR unit modified by KN0CK runs in a direct sampling mode that allow the dongle to tune to HF frequencies. Refer to [17] which describe the modification needed to receive a direct sampling receiver. A low-pass filter is also used in KN0CK's implementation that limit the receiving range to 0.5-54 MHz (includes the desired range 2-30 MHz), more details can be found in [14]. Although the desired frequency range is supported another problem still seems to be present. As far as the author knows the frequency stability is still not good enough for our purpose.

A solution to the frequency stability problem might be a modified device developed by Nobu Saito. This device use a Temperature Controlled Crystal Oscillator

Table 4.4: Specification of the FunCube Dongle Pro+ [18].

Feature	FunCube Dongle Pro+
Operating frequency range	0.15-240 MHz 0.42-1.9 GHz
Frequency stability	±1.5 ppm
Bandwidth	192 kHz
Connection	RF Input: SMA Output: USB
Price	£125

(TCXO) resulting in a frequency stability as low as ±2 ppm [15]. Nevertheless nothing is done regarding the frequency range and we are stuck with that problem considering this device.

Both the above devices using a USB interface intending to be used with a PC which will perform most of the signal processing. Thus, these devices are not intended to be used in the front end of a Field-Programmable Gate Array (FPGA) based SDR.

FunCube Dongle Pro+

The FunCube Dongle Pro+ is a receiver intending to be used as the ground receiver for the FunCube Satellite project. Maybe it can be used as the down-converter in this project too. Some specifications for the receiver can be seen in Table 4.4. The frequency stability of ±1.5 ppm is the same as 15 Hz at an operating frequency of 10 MHz. This is a good performance, especially for a receiver this wide-banded. Due to the good performance the price is also higher for this device.

Also the FunCube Dongle uses an USB interface suitable and intended for use with a PC. A possibility would be to use a couple of dongles together with a PC as a receiving radio system intended for multi channel reception. For this to work each dongle will need to identify itself as a unique device to the PC.

4.2 Detector

The purpose of the detector is to find the BW5 waveform described in Subsection 1.1.2 and is done by some signal processing algorithms. The main part of the detector will be a cross-correlation algorithm intended to search for a specific part of the incoming signal.

4.2.1 Cross-correlation

Cross-correlation is a method to measure the similarity between two signals. It returns a large value if the signals are correlated (similar) and a low value if the signals are uncorrelated (not similar). The cross-correlation between the two discrete time signals $f[n]$ and $g[n]$ is defined as Equation (4.1).

$$C[n] \equiv \sum_{\forall m} f^*[m]g[m+n] \tag{4.1}$$

Equation (4.1) will return a large value for two correlated signals. How large the value is also depends on the energy in the signals. For convenience lets limit the cross-correlation to the interval [-1,1]. This can be done by dividing Equation (4.1) with the L^2-norm of the two signals, see Equation (4.2).

$$\overline{C}[n] = \frac{\sum_{\forall m} f^*[m]g[m+n]}{\|f\|_2\|g\|_2} \tag{4.2}$$

If the signals are identical in shape the cross-correlation will return "1" at lag zero. If instead the signals are identical in shape but with opposite sign the cross-correlation will return "-1" at lag zero. Both these cases are signs of autocorrelation and indicate that the signals are identical. To make the measurement even more easy lets take the absolute value of the cross-correlation in Equation (4.2). This will result in Equation (4.3) which will return "0" for uncorrelated signals and "1" for completely correlated signals at lag zero.

$$\overline{C}[n] = \left| \frac{\sum_{\forall m} f^*[m]g[m+n]}{\|f\|_2\|g\|_2} \right| \tag{4.3}$$

4.3 Decision

The output from the detector will be a value between "0" and "1" as described in Subsection 4.2.1. For a disturbed call the value might not be that close to "1" as for a call over a perfect channel. This will set requirements on the decision threshold. It needs to be low enough such that disturbed signals still has a chance to be detected. At the same time the threshold cannot be too low as this might cause false detection. One way to test this is to send a lot of "false" calls to the detector and get a perception of the cross-correlation values for these hopefully uncorrelated signals. The threshold is then set such that it does not fall below these correlation values.

4.4 Software Defined Radio

To implement the signal processing and to efficiently process the signal a Software Defined Radio (SDR) can be used.

In May 1985 E-Systems (now Raytheon) released their company newsletter describing a new type of radio signal processing technique. It was a digital computer based receiver, also called the software radio. Experts saw great potential in the technique and thought it might revolutionize the signal processing of complex radio signals. Refer to [19] for further information.

The description of a software radio above is not a complete description, at least not according to Joseph Mitola III. He meant that a radio should be able to both receive and transmit and in 1992 he reinvented the term software radio [20].

The term software defined radio was later proposed by Stephen Blust in 1995 and many give him the credit of bringing commercial credibility to the concept. Until this time the technique was seen as a military technology and was not meant for the commercial market [21]. Thereafter the use of SDR has spread to many more areas and nowadays the technology can be found all around us.

The idea of a SDR is that traditionally hardware implemented components can be replaced by software implementation. SDR Forum (also Wireless Innovation Forum) give a definition of SDR presented in [22]:

"Radio in which some or all of the physical layer functions are software defined"

One problem with traditional radios are the impact of the hardware implementation which does not give much freedom and any modification needs to be done by physical intervention. This gives the traditional radio low flexibility in supporting multiple waveforms. Also the production cost of a high-performance radio is relatively high. With a SDR these drawbacks can be eliminated. The implementation realizes a more efficient and relatively inexpensive solution that will have no problem to support multiple waveforms with high performance. In fact one SDR platform can be used in many different implementations and though one software implementation can be reused and implemented in multiple devices. This also simplifies updating the radio with bug-fixes, new features etc.

Figure 4.2 shows a simplified base station system model for a SDR receiver (similar but inverted diagram for a transmitting device). The different functions are described as follows:

Antenna: This element is the first element in the chain and the one that provides the next elements with the incoming RF signal.

RF/IF: All the necessary analog signal processing is done here. The processing may include a preselector used to limit the range of frequencies that can be applied to the next step. The processing may also suppress known local interference signals. During this step a local oscillator may be used to down-convert the signal to an intermediate or baseband frequency.

Channel selector: The ADC will take place before this element and henceforth all the signal processing will be done within the software. In this element the processing will extract the desired signal from the proper channel. This might also be done directly in the previous step.

Baseband processing etc.: The purpose of this element is to continuously improve the link based on information from the previous step. Signal processing within this element may include power control, frequency hopping and antenna control. Also this element will be responsible for the decryption of the information if encryption is used in the link.

Call/MSG PROCESS and I/O: This element is responsible for delivering traffic to the rest of the system. The traffic might be speech, multimedia, messages etc. Further details can be found in [22].

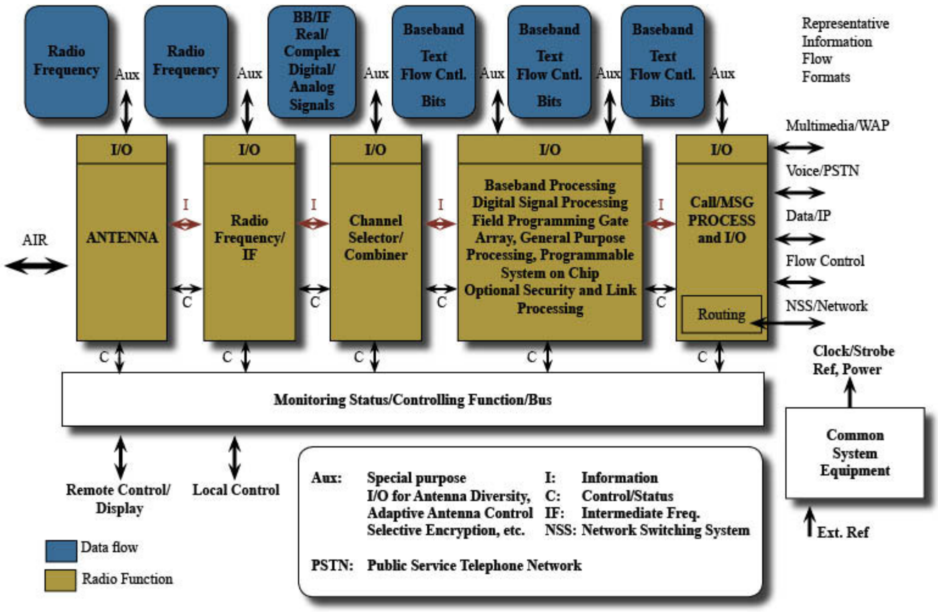

Figure 4.2: Generalized functional architecture of a SDR [22].

Chapter 5

Implementation of the detector

In this chapter the detector implementation is presented. First, the requirements on the detector are stated. This is followed by a detailed description of the implementation and the methods used in the design work.

5.1 Requirements

Initially there exist requirements on the returned value from the cross-correlation detector. The value is required to be large enough such that the cross-correlation should work even though the signal is influenced by noise and other disturbances.

5.1.1 HF channel model

The International Telecommunication Union (ITU) provides technical recommendations for radio communication. The ITU-R Recommendations are a set of international technical standards formed from results of studies made by ITU. ITU-R Recommendation F520-2 defines an HF channel model. Although this recommendation set requirements on different parameters that shall be used in the channel model, it does not set any deeper requirements on the actual model to be used. For more information, see [23].

The main characteristic of a HF channel is the multi-path time-varying environment. This results in time and frequency dispersion on the transmitted signal. According to Section 1.2 the most common mode is the ionospheric wave. The multi-path propagation is a result of reflection in several layers of the ionosphere. The time dispersion is a direct consequence of that different paths are received with different time delays. Each path also shows different fading characteristics due to the nature of the ionosphere which results in the frequency dispersion. Refer to [24] for more details.

Modulating the HF channels should be done using a high frequency ionospheric channel model employing the C.C. Watterson Model [25]. The Watterson model is a stationary representation of the channel valid for a short period (≈ 10 minutes) and band-limited channels (≈ 10 kHz). The model is realized using Equation (5.1) where i is the time index, y_i is the complex output, h_j represents the taps in the filter, x_i is the complex input and n_i is the AWGN in the channel.

Table 5.1: LSU Linking probability requirements for different SNR values presented in dB [5].

Probability Link Success	Gaussian	ITU-R F.520-2 Good	ITU-R F.520-2 Poor
25%	-10	-8	-6
50%	-9	-6	-3
85%	-8	-3	0
95%	-7	1	3

Table 5.2: Channel conditions for the good and poor channel [23].

Channel	Fading paths	Differential time delay	Frequency spread
ITU-R F.520-2 Good	2	0.5 ms	0.1 Hz
ITU-R F.520-2 Poor	2	2 ms	1 Hz

$$y_i = \sum_{j=0}^{L-1} h_j x_{i-j} + n_i \tag{5.1}$$

It is essential to set the signal power to 1 before sending it through the channel. The AWGN channel need to know this to determine the right SNR value as this is dependent on the signal. The power of the signal can be normalized to 1 using Equation (5.2) where x_i represent each sample of the signal and n represent the length of the signal.

$$\vec{x}_{norm} = \frac{\vec{x}}{\sqrt{\frac{1}{n}\sum x_i^2}} \tag{5.2}$$

The 3G-ALE standard (see Subsection 1.1.2) defines performance requirements on the linking shown in Table 5.1 where the first column show the required probability of link success for different channels. The other three columns represent different channels. The first channel just adds white Gaussian noise to the signal and the other two channels also add HF channel distortion to the signal, see [5] for further details.

The good and poor channel presented in Table 5.1 each consists of two independent fading paths described in Table 5.2. The two paths in the channels have equal frequency spread, equal mean attenuation and no-frequency shifts [23].

5.1.2 Parallel receivers

Using a PC as the receiver will require several input signals through an audio interface. Assume that the PSDR described in Subsection 4.1.1 is used as the down-converter together with signal generators. If so, the computer needs to receive several audio signals simultaneous each corresponding to one channel. This can be done using several external USB sound cards such as the Terratec Aureon Dual USB [26]. A couple of these was bought and tested to verify that they worked together. The result was positive. The only down side with these devices was that the input was mono causing the PC to just receive half of the signal energy. This is a problem in the sense of SNR but does not affect the result for perfect noise free signals.

5.1.3 Circular buffer

As the receiver is working it need to store the incoming samples in a vector to do the signal processing. This is done with some kind of buffer that stores a specific number of samples and for each new sample it drops the oldest one. Using a common vector will cause problem due to the fact that all the values in the vector have to be shifted for each new sample. By using a circular vector this shift can be eliminated and the complexity of the buffer can be decreased. Instead of insert every new sample in the beginning of the vector lets use a pointer to represent our location in the vector. Then the sample can be added relatively to the pointer and the user is still able to know where the buffer begins and end, see illustration in Figure 5.1.

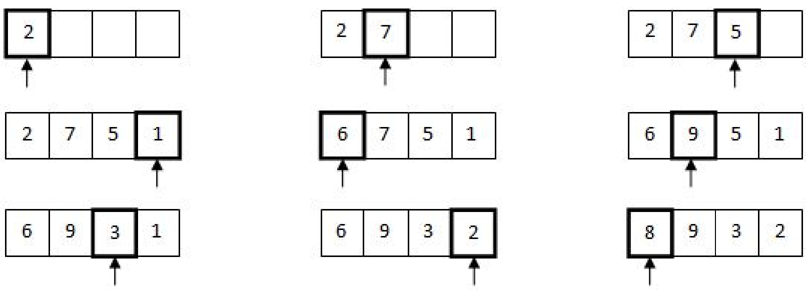

Figure 5.1: Illustration of a circular buffer using a buffer size of four samples. When the pointer reaches the end of the vector it starts over at the beginning and keep going round and round, adding the upcoming samples.

5.2 MATLAB implementation

In this section a MATLAB implementation of the single channel receiver is implemented. The implementation focus on the functionality of the detector and does not handle the real time situation. Signals processed by this implementation are pre-recorded and then imported into the program.

5.2.1 Reference signal

The first step in the process is to create the reference signal that shall be used in the detection. This shall be the preamble in the BW5 waveform described in Subsection 1.1.2. As described in that section the preamble consists of 576 symbols modulated with 8-ary PSK with a symbol rate of 2400 symbols/second. The PSK modulated signal is divided up into in-phase (I) and quadrature (Q) components. These are then sent through a pulse shaping filter, a square root raised cosine FIR filter, see further details later in this subsection. The I and Q components are frequency converted from baseband to the sub-carrier frequency of 1800 Hz and added together to the baseband signal s(t), see Figure 5.2.

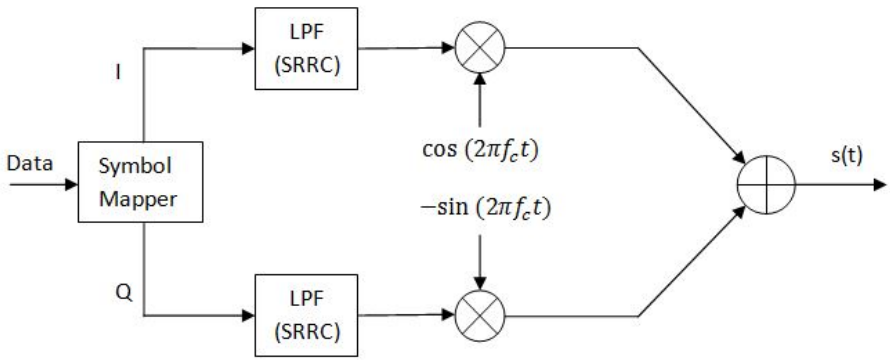

Figure 5.2: Carrier modulation.

Pulse shaping

The pulse shaping is done using a square root raised cosine filter which impulse response can be seen in Figure 5.3. The low-pass filter is used to minimize the impact of intersymbol interference (ISI). To do this the filter is non-zero at one symbol and zero at all nT where $n \neq 0$ as also can be seen in the figure. A roll-off factor of 0.25 is used and the filter order is set to 40. Many different values of the roll-off factor where evaluated to find the most suitable one. This was done using the cross-correlation and a real call signal generated by an actual HF 2000 system. When iteratively trying different values the best one was found. The filter order where also tried out and was found to not improve the receiver any more when exceeds 40.

The square root raised cosine filter is divided to be used in both the transmitter and the receiver. The frequency response at the transmitter and the receiver is each the square root of the total frequency response. Together they form a raised cosine filter with the frequency response shown in Equation (5.3). The $H_{srrc}(f)$ represent the square root raised cosine and the $H_{rc}(f)$ represent the raised cosine frequency response.

$$H_{rc}(f) = H_{srrc}(f) \cdot H_{srrc}(f) \tag{5.3}$$

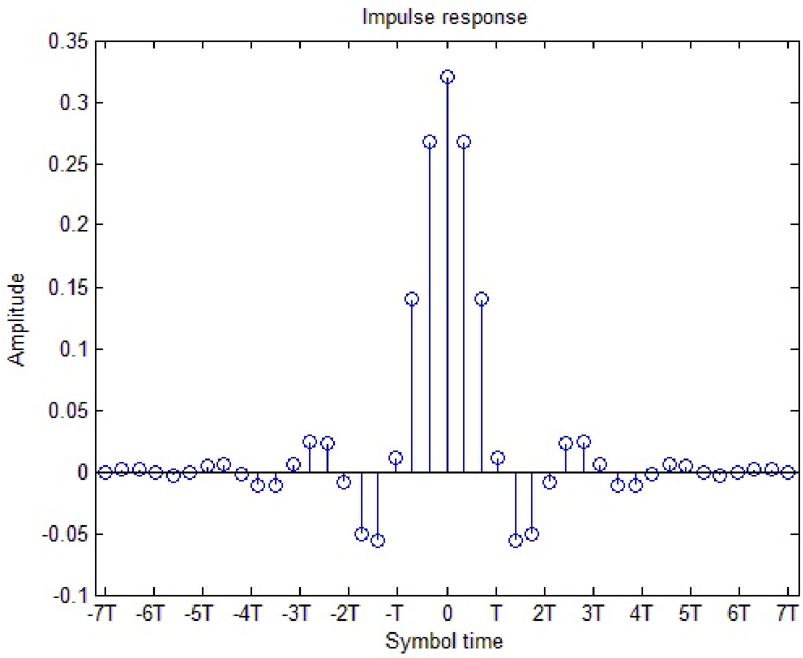

Figure 5.3: Impulse response of the square root raised cosine filter using a symbol rate of 2400 symbols/second and a sample rate of 8 samples/second.

The reason to divide the filter between the transmitter and the receiver is to minimize the impact of white noise and to maximize the SNR. This approach is called matched filtering and is widely used in digital communication [27]. In this particular implementation only the filter in the transmitter is being taken care of. The reference signal is processed with a square root raised cosine filter just as the transmitted signal. The received signal on the other hand is not processed with a square root raised cosine filter.

5.2.2 Cross-correlation detector

Next the cross-correlation detector is created. It is assumed that the signal is already down-converted from its RF frequency. The detector saves the incoming signal in a circular vector and then continuously compares this signal to the reference signal using cross-correlation. When the cross-correlation value exceeds a certain threshold value detection has been made.

The cross-correlation can be done using different parts of the signal, for instance the I/Q streams or the baseband signal. In the MATLAB implementation the real baseband signal was used directly without first extracting the I and Q components. Initially one HF 2000 transceiver was used to generate the input signals that was recorded and then evaluated in MATLAB.

5.2.3 Performance measurements

The measurement is done using a HF channel model described in Subsection 5.1.1 to simulate the impact of the HF channel. There exists several implementations of the model which might lead to different results. A MATLAB implementation of the model was chosen for convenience. The model follows the ITU-R F.520-2 standard and performs the channel simulation directly in the code with a already sampled signal.

For good performance the receiver shall be able to handle the channel conditions in Table 5.1. At the same time the chance of false detection shall be as small as possible.

5.2.4 Experimental setup

The setup used to capture the 3G-ALE signal can be seen in Figure 5.4. The HF 2000 node consist of one Node Control Terminal (NCT) and one Radio Station Control Unit (RSCU). The former is used for configuration and monitoring of the node and the latter is used for real-time control of one radio station. The NCT is connected to the RSCU via a LAN. The 3G-ALE call signal generated by the RSCU is intended to be further processed by a radio transmitter to represent the HF radio signal. In this setup it is desirable to take the baseband signal directly from the RSCU and send it directly to the computer using the audio input port. The signal is recorded and later processed by the MATLAB implemented cross-correlation detector.

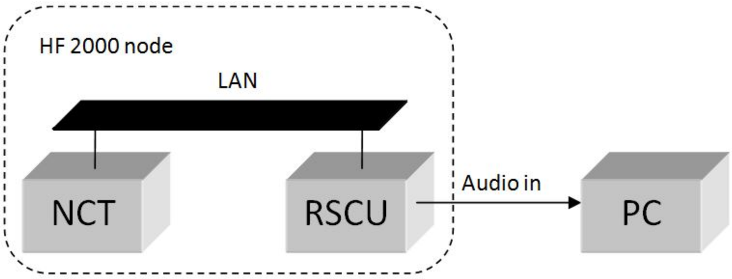

Figure 5.4: Basic description of the experimental setup used to capture the 3G-ALE signal produced by the RSCU.

5.3 Simulink implementation

The MATLAB implementation of the cross-correlation detector does not support real-time input signal processing and can only process data that has already been pre-recorded. In practice this is not a useful solution, a real-time implementation is needed! Using the block structure provided in the Simulink environment a new implementation was done using the same method as before, but in real-time.

5.3.1 Sampled signal

The signal stated in Equation (1.1) is the waveform used as the sub-carrier signal.

The upper sideband HF representation of this signal is obtained using the technique stated in Equation (1.2). A Hilbert transform is used in the derivation of the USB HF signal, see Equation (5.4) The USB HF signal is then derived using $s(t)$ and $\hat{s}(t)$ and can be seen in Equation (5.5). Figure 5.5 shows the frequency components in this signal.

$$\hat{s}(t) = H(s)(t) = I(t)sin(\omega_c t) + Q(t)sin(\omega_c t) \tag{5.4}$$

$$\begin{aligned} s_{HF}(t) = {} & [I(t)cos(\omega_c t + \varphi) - Q(t)sin(\omega_c t + \varphi)]\, cos(\omega_{HF}) \\ & - [I(t)sin(\omega_c t + \varphi) + Q(t)cos(\omega_c t + \varphi)]\, sin(\omega_{HF}) \end{aligned} \tag{5.5}$$

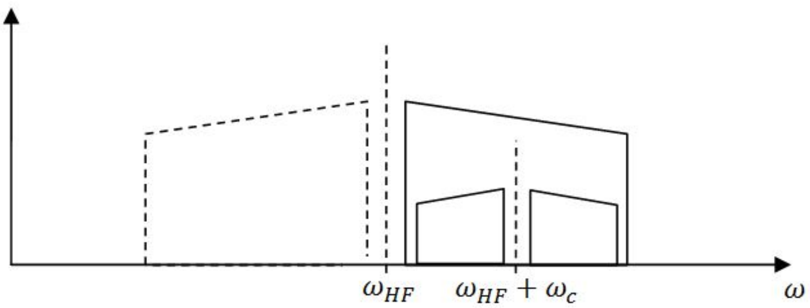

Figure 5.5: Frequency components in transmitted signal.

Using a Tayloe detector set to ω_{HF} as the front-end will result in two output signals shown in Equation (5.6).

$$\begin{cases} \hat{I}(t) = [I(t)cos(\omega_c t + \varphi) - Q(t)sin(\omega_c t + \varphi)] \\ \hat{Q}(t) = [I(t)sin(\omega_c t + \varphi) + Q(t)cos(\omega_c t + \varphi)] \end{cases} \tag{5.6}$$

The two input signals shown in Equation (5.6) will be sampled at a sample rate of 8 kSamples/second and stored in a 2764 samples long buffer. For each new sample the buffer will output the 2764 samples as a vector to be further processed in the receiver. The number of samples used in the buffer is the sum of the samples that shall be used in the frequency estimation and the samples that shall be used in the cross-correlation, i.e. 844+1920 samples. Table 5.3 shows the length of the different sequences in the BW5 waveform using our sample rate. The frequency estimation is performed using the TLC sequence where "only" the 844 last samples are used in the estimation. The TLC sequence follows by the preamble sequence that is used in the cross-correlation. Therefore the buffer length will be the sum of these samples as both sequences are used simultaneously. See Subsection 5.3.4 and 5.3.5 for further information about the buffer length.

Table 5.3: Samples used to represent the TLC, preamble and data sequences in BW5.

TLC	Preamble	Data
853.3	1920	5333.3

5.3.2 Squelch detector

As the computer is not the fastest alternative to be used in DSP, it is important to make the program efficient. One way to do this is to use a squelch detector. This is a circuit that only passes the signal through if it is strong enough or have the right shape. This way the receiver does not need to do all the heavy signal processing unless a signal is present. Together with a small buffer in the recording block this reduces the risk of dropped samples, see [28] for further details. The squelch detector designed in this thesis is on for 50 samples and then turns off for the next 5.000 to ensure that the receiver just look into the training- and preamble sequence of the signal. The drawback with using a squelch detector is that weak signals may not get detected. Also, in real skywave HF conditions the signal amplitude may vary considerably thus making it difficult to find a suitable squelch threshold value.

5.3.3 Hilbert transform

To ensure maximum SNR of the processed signal both $\hat{I}(t)$ and $\hat{Q}(t)$ are used in Equation (5.6). $\hat{I}(t)$ and $\hat{Q}(t)$ are the same signal with a phase shift of $\pi/2$. A Hilbert transform on $\hat{Q}(t)$ will produce $-\hat{I}(t)$. Taking the sum of $\hat{I}(t)$ and $-H(\hat{Q})(t)$ will result in $2\hat{I}(t)$ which in fact is the transmitted signal with an introduced phase shift, see Equation (5.7). Also this procedure will keep the energy in the signal in contrast to the traditional heterodyne receiver.

$$s(t) = 2\hat{I}(t) = 2\left[I(t)cos(\omega_c t + \varphi) - Q(t)sin(\omega_c t + \varphi)\right] \tag{5.7}$$

5.3.4 Frequency offset estimation and compensation

The frequency used in the up/down conversion of the signal need to be precise to achieve proper detection. Due to uncertainty in the carrier frequency of the received signal a frequency compensating technique is used to estimate and correct the potential offset. The method used in this implementation is called Best Angular Fit (BAF) [29]. It is a data-aided algorithm which means that it uses a known part of the received signal and analyzes it to obtain the frequency offset. The 3G-ALE standard using a burst waveform starting with a fixed training sequence (the initial TLC part) which in this thesis is used to achieve data-aided frequency estimation. The BAF is an iterative method derived from the least square criterion resulting in the update equations shown in Table 5.4.

The equations shown in Table 5.4 forms a low-complexity algorithm which converges in few iterations, ie. 3-5 iterations. Furthermore the complexity of the algorithm is $O(N)$ and that is really impressive. The downside with the algorithm

Table 5.4: Computation using the BAF algorithm [29].

BAF Algorithm
$\Delta\hat{f}^0 = 0$
for each n
$\{$
for each k
$\{$
$z_k^{(n)} = z_k e^{-j2\pi\Delta\hat{f}^{(n)}T_s k}$
$\hat{\varepsilon}_{\Delta f}^{(n)} T_s = \frac{3}{\pi(N-1)} \cdot arg\left\{\left(\sum_{k=1}^{N} k \cdot z_k^{(n)}\right) \cdot \left(\sum_{k=1}^{N} z_k^{(n)}\right)^*\right\}$
$\Delta\hat{f}^{(n)} = \hat{\varepsilon}_{\Delta f}^{(n)} + \Delta\hat{f}^{(n-1)}$
$\}$
$\}$
n: Iterations
k: Number of pilot samples

is the estimation range which is relatively small. More specific $|\Delta f T_s| < \frac{1}{N}$ where Δf is the detectable frequency offset, T_s is the sampling frequency and N is the number of samples used in the training sequence (which correspond 106.67 ms). Using a sample frequency of 8 kHz results in a training sequence of 853.3 samples, refer to Figure 1.2. Lets use the last 844 samples in the training sequence to maintain a margin in the beginning of the waveform. This will result in a frequency offset span shown in Equation (5.8).

$$|\Delta f| < \frac{1}{NT_s} \approx 9.5\,\text{Hz} \tag{5.8}$$

The input signal to the BAF algorithm is $z(t) = r(t) \cdot a^*(t)$ where $r(t)$ is the received signal and $a(t)$ is the transmitted signal, i.e. $z(t)$ is a description of the channel. This description prefer to have the transmitted and received signal as analytical signals, i.e. as complex exponential functions. The received signal can be written as Equation (5.9). This is the same as Equation (5.7) but with a introduced frequency error.

$$\begin{aligned} r(t) &= 2\left[I(t)cos(2\pi(f_c+\Delta f)t+\varphi) - Q(t)sin(2\pi(f_c+\Delta f)t+\varphi)\right] \\ &= \left[I(t)+jQ(t)\right]e^{j(2\pi(f_c+\Delta f)t+\varphi)} + \left[I(t)-jQ(t)\right]e^{-j(2\pi(f_c+\Delta f)t+\varphi)} \end{aligned} \tag{5.9}$$

In the BAF algorithm a baseband signal is used in the frequency estimation. The implementation in this thesis using the sub-carrier baseband signal. This signal has two exponential functions shown in Equation (5.9) instead of one as stated in the baseband signal in BAF [29]. The "transmitted" signal is in this algorithm assumed to be as described in Equation (5.10). This is not the actual transmitted signal but it is used to fit the BAF algorithm.

$$a(t) = [I(t) + jQ(t)]e^{j2\pi f_c t} \tag{5.10}$$

Using these equations will result in a channel model as described in Equation (5.11).

$$z(t) = \left[I^2(t) + Q^2(t)\right] e^{j(2\pi \Delta f t + \varphi)} + [I(t) - jQ(t)]^2 \, e^{-j(2\pi(2f_c + \Delta f)t + \varphi)} \tag{5.11}$$

$I(t)$ and $Q(t)$ are retrieved from the 8-ary PSK modulation and the sum of the two is a point on the unit circle. Hence the term $I^2(t) + Q^2(t)$ in Equation (5.11) has unit energy, i.e. $I^2(t) + Q^2(t) = 1$. To get the desired structure of $z(t)$ it is possible to remove the higher frequency component using a low pass filter. Even without the low pass filter the BAF algorithm works well. Therefore the low pass filter is not that important and may be neglected to get a simpler implementation.

The algorithm in Table 5.4 will estimate the compensation needed in the frequency domain. What the algorithm basically do is that it estimates the frequency error for each iteration and tries to compensate for it. Set $t = T_s k$ will return in a term $z_k^{(n)}$ on the following form shown in Equation (5.12).

$$z_k^{(n)}(t) = e^{j\left(2\pi\left(\Delta f - \Delta \hat{f}^{(n)}\right)T_s k + \varphi\right)} \tag{5.12}$$

For each iteration the remained error $\Delta f - \Delta \hat{f}^{(n)}$ will be estimated in the parameter $\hat{\varepsilon}_{\Delta f}^{(n)}$ and the estimated offset $\Delta \hat{f}^{(n)}$ is then updated using the error estimation.

As the algorithm estimates the frequency offset on the training sequence it also compensate for the change. The compensation is done on the preamble reference signal which is sequentially updated for each new sample. It can then be used as a modified reference in the cross-correlation. This way it is the reference that give itself a frequency offset error to match the input signal.

5.3.5 Cross-correlation

Cross-correlation is done on the preample part of the call signal, which is the last 1920 samples of the buffered signal. It is always done between the reference signal and a windowed part of the input signal. The process is illustrated in Figure 5.6 where a new sample is introduced in the input signal for each time step. For each new sample the input vector is shifted to the right and drops its last sample. This can be done quite efficient using a circular buffer described in Subsection 5.1.3. When the cross-correlation returns a value larger than a certain threshold value detection has been made!

5.3.6 Experimental setup

In the real-time implementation the receiver should be able to handle HF input signals in real time. Similar to the MATLAB implementation the modulated signal is created using the HF 2000 RSCU. The modulated signal is further processed by a radio transmitter which put a HF carrier onto the signal forming a real HF 2000 node. To represent the path loss on a typical link an attenuator is used

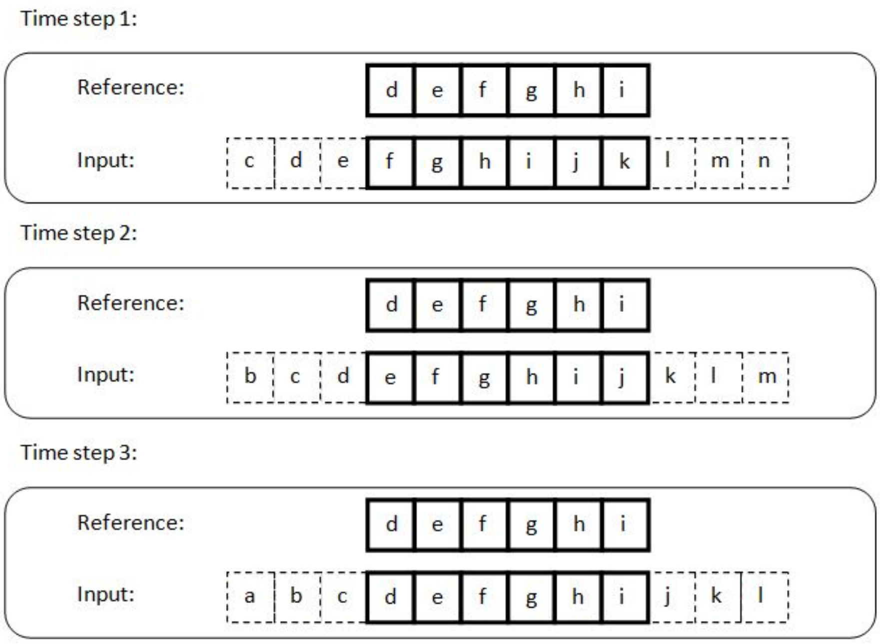

Figure 5.6: The reference signal and the input signal used during three time steps to calculate the cross-correlation value. At the last time step the cross-correlation will reach its maximum value and detection has been made.

between the transmitter and the receiver. The down conversion to the baseband frequency is done using a Tayloe detector together with a stable signal generator. This produces a stable signal that can be directed to the audio input port of the computer. An illustration of the setup is shown in Figure 5.7.

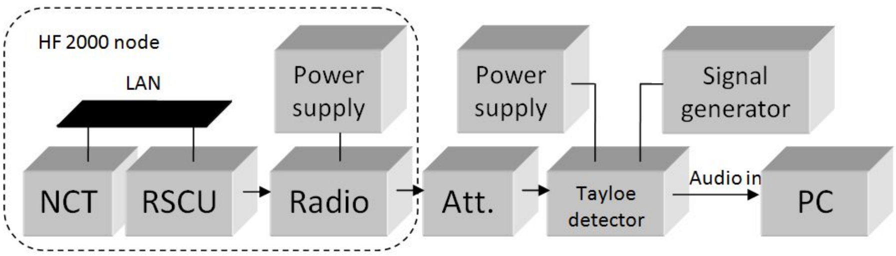

Figure 5.7: Basic description of the experimental setup used to up- and down convert the signal and receive the signal into the computer program.

Parallel receivers

Normally a computer only has one sound card meaning that it can only receive one channel at each time. In this thesis it is necessary to be able to monitoring and input several channels to the computer simultaneous. One possibility is to use external USB sound cards to enable several input channels. The sound cards can then be connected directly to the USB ports of the computer or use a USB hub to make the system more convenient to work with. Figure 5.8 shows a illustration of such a system.

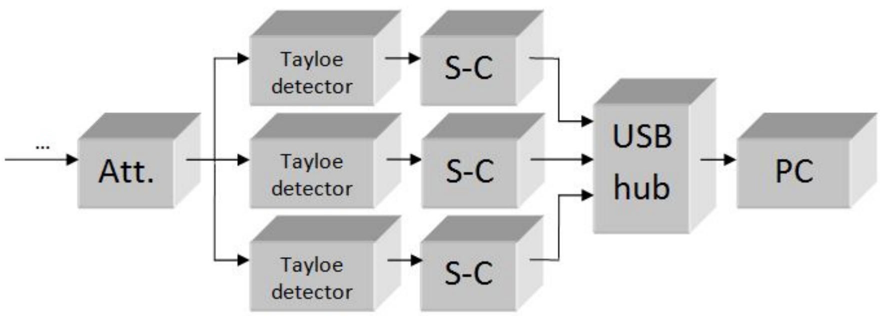

Figure 5.8: Description of the system that allows several input channels to a PC via a USB soundcard interface.

Chapter 6

Result

6.1 MATLAB implementation

A short part of the reference signal is shown in Figure 6.1, both as continuous signal and as sampled signal without using raised cosine filter. The continuous signal clearly show the behaviour of a phase modulated signal and through closer investigation the symbol rate is confirmed to be 2400 symbols/s.

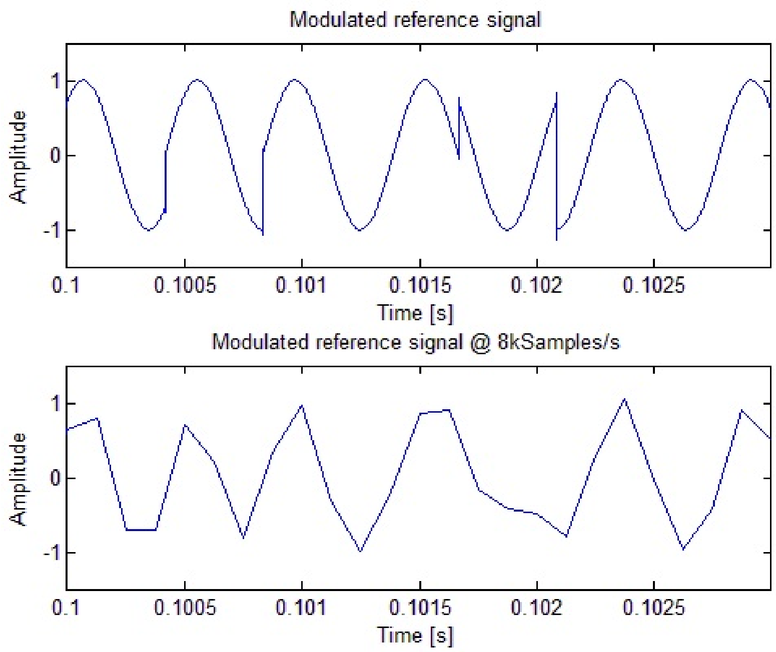

Figure 6.1: Reference signal before and after sampling.

Using the sampled signal in Figure 6.1 an input audio signal can be studied with the cross-correlation described in Section 4.2.1. The input signal used in the detection does not contain any noise as it was recorded almost directly from the source. See Figure 6.2 for the resulting detection on this input signal.

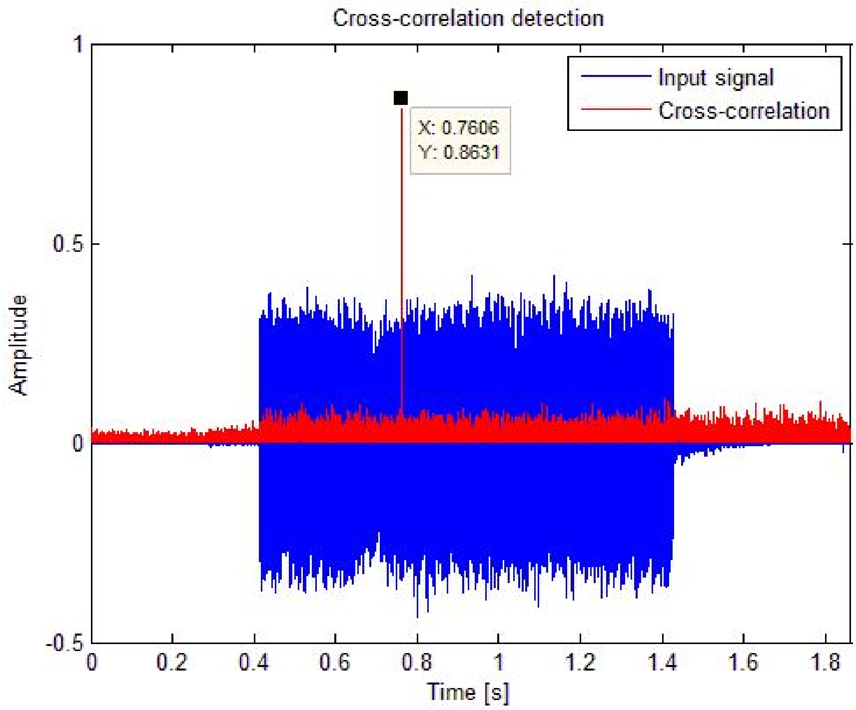

Figure 6.2: Input signal & the cross-correlation detection.

6.1.1 Performance measurement

Figure 6.2 shows a typical cross-correlation value of a detector with no significant impact of the channel. In real life it is not this simple; the channel will cause major distortion to the signal. Using the HF channel model discussed in Section 5.1.1 with a detection threshold of 0.27 will give the results shown in Table 6.1. The measurement is done 1000 times for each channel to give good statistics. A threshold of 0.28 was also evaluated but it did not fulfill the requirements. Consequently a threshold of 0.27 and below shall be used in the decision code.

False calls

It is undesired to detect calls that uses other standards [30]. Therefore some sequences are tested to ensure that the cross-correlation does not return a large value that indicates detection. Table 6.2 shows the largest values received during the cross-correlation. The results indicate that a threshold value larger than 0.15 shall be used in the decision mechanism in order to reject calls that uses other standards.

Table 6.1: Performance measurements using the HF channel discussed in Table 5.1 with a cross-correlation threshold value of 0.27.

Channel	SNR[dB]	Linking probability requirement	Using MATLAB HF simulator	Margins
Gaussian	-10	25%	38.1%	+13.1%
	-9	50%	67.7%	+17.7%
	-8	85%	85.9%	+0.9%
	-7	95%	97.2%	+2.2%
ITU-R F.520-2 Good	-8	25%	41.0%	+16.0%
	-6	50%	62.3%	+12.3%
	-3	85%	85.3%	+0.3%
	1	95%	96.1%	+1.1%
ITU-R F.520-2 Poor	-6	25%	60.3%	+35.3%
	-3	50%	84.1%	+34.1%
	0	85%	93.7%	+8.7%
	3	95%	98.3%	+3.3%

Table 6.2: Cross-correlation values for calls using other standards [30].

Standard	Uncoded	Coded, Zero Interleaving	Coded, Short Interleaving	Coded, Long Interleaving
MIL-STD-188-110A	0.1117	0.1362	0.1087	0.1268
MIL-STD-188-110A Narrow-Band	0.1167	0.1261	0.1321	0.1432
STANAG 4285	0.1058	-	0.1083	0.1235
STANAG 4529	0.1228	-	0.1367	0.1342

Sampled signal

One thing that will affect the quality of the cross-correlation value is the fact that the signal is sampled. The signal may be sampled "in phase" with the sampled reference signal. This will return the largest value from the cross-correlation. If the samples are "out of phase" the cross-correlation value will decrease. The samples are taken from different parts of the signal but are still compared directly to each other, see Figure 6.3. When sweeping the sample location over time the cross-correlation value will behave as in Figure 6.4. The periodicity corresponds directly to the sample rate (i.e. 0.125 ms). The behaviour will affect the performance negatively. The sample frequency can be increased to decrease the influence of this behaviour, but it will not be done in this implementation.

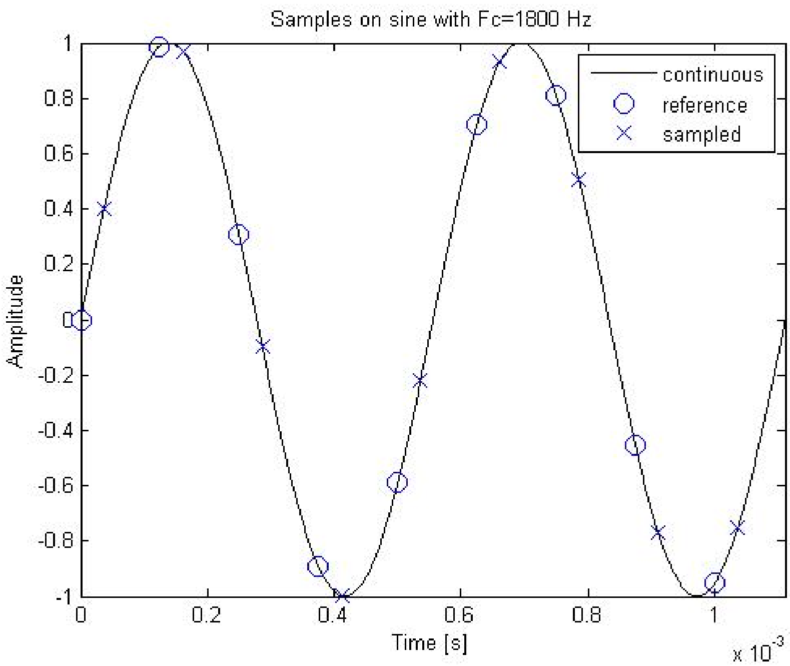

Figure 6.3: Samples of reference signal and the input signal. The sample rate is right, but the location of the samples does not match.

6.2 Simulink implementation

6.2.1 Verification of PSDR and VCO

For the real-time model to work it is essential that the RF front-end works correctly.

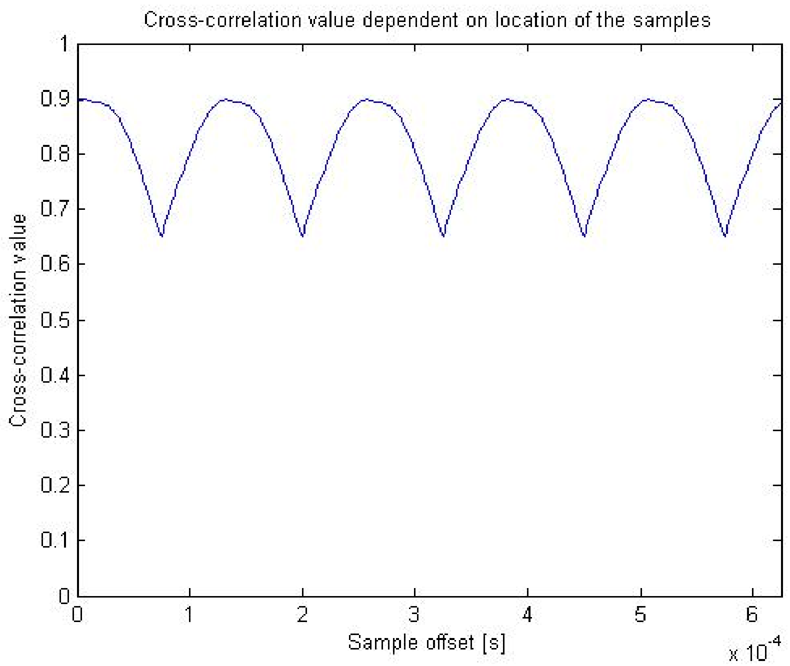

Figure 6.4: Sampled signal with varying offset compared to the reference signal.

VCO

To verify the specification a performance test was done on the PGEN45 described in Subsection 4.1.1. The result shown in Table 6.3 indicates better stability than stated in the specification. The only feature that did not fulfill the specification was the lower part of the frequency range (<1 MHz), but this does not matter for our desired frequency range (8-120MHz). Even though the specification for the frequency stability seems to be quite good it still vary several hundreds of hertz from its mean value. As our receiver shall be able to detect a sub-carrier at 1800 Hz, this deviation is too large. A comparison is done using a calibrated signal generator receiving a frequency stability 10.000 times more accurate than for the PGEN45. The difference is large but so is the physical size and cost of the signal generator. It is not reasonable to use a signal generator as the frequency source in a real system. Luckily that does not mean that it cannot be used in a proof-of-concept implementation.

PSDR

Except for the stability of the switch frequency, also the bandwidth of the Tayloe detector is important. The down-converter need to have a bandwidth of at least 3 kHz as this is the largest significant frequency of the baseband signal. Two signal generators are used to make the measurement, one as an input signal and one

Table 6.3: Verification of the PGEN45 specification.

Features	PGEN45	Signal generator
Operational frequency range	1-100 MHz*	DC-25 MHz
Desired frequency range	8-120 MHz	8-120 MHz
Frequency stability	±100 ppm (@10 MHz) ±50 ppm (@40 MHz)	±0.01 ppm (@10 MHz) ±0.01 ppm (@25 MHz)
Duty cycle	$50\% \pm 0.02\%$ (@10 MHz) $46\% \pm 0.09\%$ (@40 MHz)	$50\% \pm 0.02\%$ (@10 MHz) $50\% \pm 0.06\%$ (@25 MHz)

*The upper limit is officially not supported and has a poor frequency stability for our purpose ($\approx \pm 1$ MHz).

Table 6.4: Bandwidth for three different operation frequencies.

Operating frequency [MHz]	2	10	25
3dB bandwidth [kHz]	17.21	15.35	12.17

as an oscillator. The frequency response using a 10 MHz input signal gives the graph shown in Figure 6.5. The measurement was performed at three different frequencies, see Table 6.4. At the operation frequency 25 MHz the bandwidth was 12.17 kHz, well suitable for our desired 3 kHz bandwidth.

In conclusion, there is no problem with the Tayloe detector as such, the problem is the VCO. Using a more stable VCO should improve the performance significant. It is quite hard to find really stable VCO's that are also small and have a relatively low cost. This will probably not be a problem in a future real implementation as the VCO's can be implemented by a manufacturer. As described in Subsection 4.1.2 there already exist products that has the functionality to operate with a high frequency stability.

6.2.2 Frequency offset estimation and compensation

The received signal is not perfect and has an unwanted frequency offset shown in Figure 6.6. The offset in the figure can be estimated to 4 Hz and it is relatively constant. If one assumes that the frequency offset is always zero it will result in an undetectable signal by the cross-correlation circuitry even though the signal is there!

One way to find the frequency is to adjust the signal generator and find the optimum frequency for down-conversion. This will work, but it is not a long-term

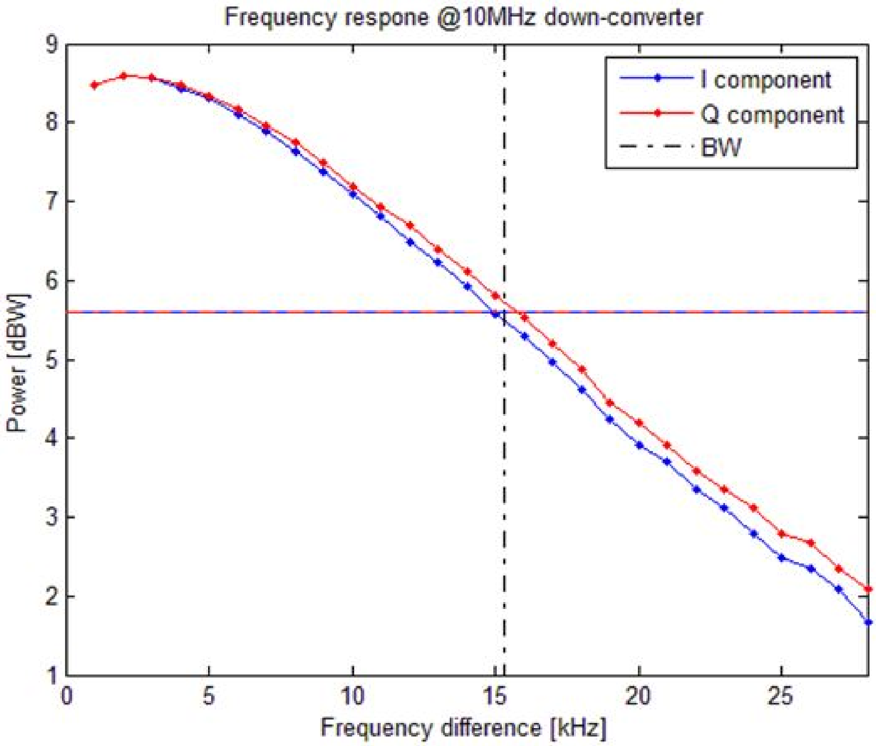

Figure 6.5: Frequency response for PSDR Tayloe detector. The horizontal line represent the power -3dB lower than the largest value while the vertical line represent the -3dB bandwidth.

solution. Using the technique described in Subsection 5.3.4 called BAF increases the possible frequency offset that is allowed for a possible detection. As shown in Figure 6.7 the maximum frequency offset without the compensation is about ±1 Hz for good detection. With the compensation this span is increased to ±10 Hz. This matches the theoretical result shown in Equation (5.8) that predict a possible frequency offset span of ±9.5 Hz. Figure 6.7 was received when manually trying different frequency offsets and measured the cross-correlation value for each offset.

The BAF algorithm also needs to work on a noisy signal. Thus 19 signals with different frequency offset was modulated through a noisy channel and then processed by the algorithm. The 19 different signals have frequency offsets between -9 Hz and 9 Hz with steps of 1 Hz. A SNR value of -10dB was used to illustrate a badly distorted channel. A typical result is shown in Figure 6.8. Often the algorithm significantly decrease the frequency error, but sometimes the error is getting worse. This can be seen for the frequency offset 9 Hz where the error is increasing above 12 Hz.

Simulations shows that this behavior is more common for larger frequency offsets. The noise might disturb the algorithm as much as the estimated frequency jump outside the estimation range of [-9.5,9.5] Hz. Larger offsets are closer to

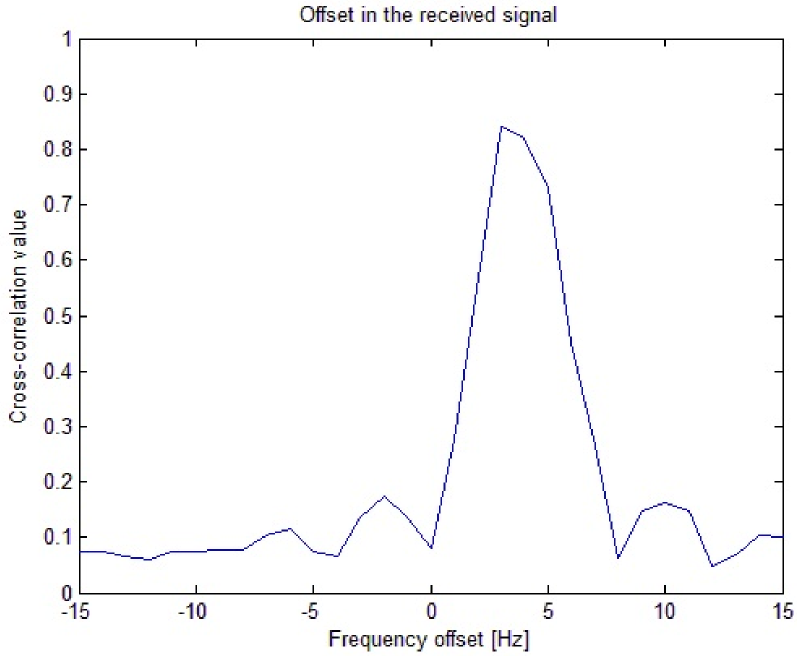

Figure 6.6: Frequency offset present in the received signal causing problem!

the limit of the estimation range causing them more sensitive to noise that might make the BAF algorithm jumping outside the range. Those offsets that the BAF algorithm succeeded to estimate converges in about 2-3 iterations and therefore 3 iterations is used in the implemented detector model.

The noise will also affect the final estimated frequency offset which can also be seen in Figure 6.8. Without noise all these frequency offsets will be estimated with a high accuracy. In presence of noise there will still remain a small error in the estimated value that will increase with the noise.

6.2.3 Simulink model (one detector)

The Simulink model developed to handle all the digital processing can be seen in Figure 6.9. Here the squelch detector can be seen which outputs a logical "1" if a signal is present and a logical "0" otherwise. The "1" will trigger two switches, the first one to start the signal detection search and the second one to ensure that the system is triggered for only a short time period since a call signal has a limited duration. When the squelch detector triggers the system it enables the "signal processing" block for a specific amount of samples. All the heavy processing is done in this block. The block contains the Hilbert transform that makes it possible to merge the two input channels. It also contains the frequency estimation algorithm and the cross-correlation. The output signals "Freq_offset"

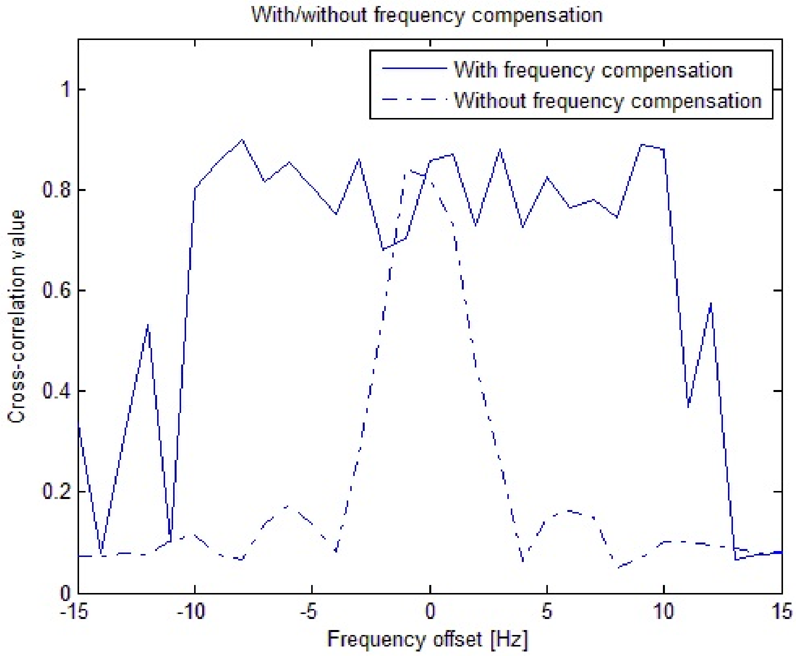

Figure 6.7: Compensation of the frequency offset will widen the range of acceptable offsets.

and "Cross_corr_val" are further handled by some customized C-code which is enabled as the model is built into an executable file. The "model header" block contains code to declare necessary variables. In the "system output" block C-code is generated to process the values contained by the two outputs. Also this code will output text in the command window if detection has been made.

From audio device

This block retrieve samples from a soundcard at a user defined sample rate, here 8 kSamples/s. The user can also specify which audio device to use and how many channels this device offers. In this implementation 2 channels are used to maximize the SNR as described in Subsection 5.3.3. It is also possible to set a buffer that prevents dropped samples, the buffer causes a small delay in the system and makes the program more dynamic if it should not work in real-time all the time.

Buffer

The buffer will store a certain amount of samples and output them as a vector. A buffer size of 2764 samples (see Subsection 5.3.1) is used in this implementation with an overlap of 2763 samples. The buffer will output all its content for every new sample. The buffer works as a circular buffer and this is an efficient way to handle the procedure. Frame based signals are contained within this block and

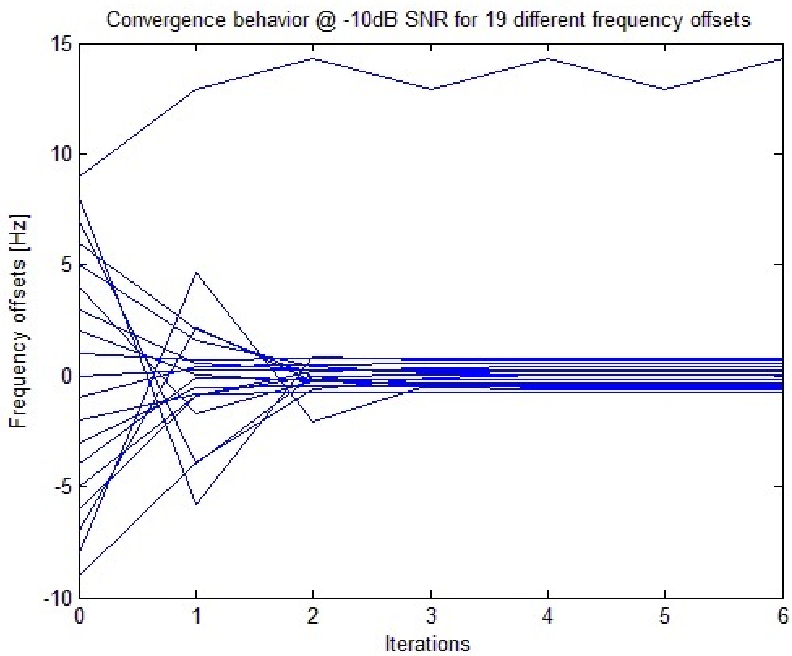

Figure 6.8: Convergence of the BAF algorithm at impact of noise.

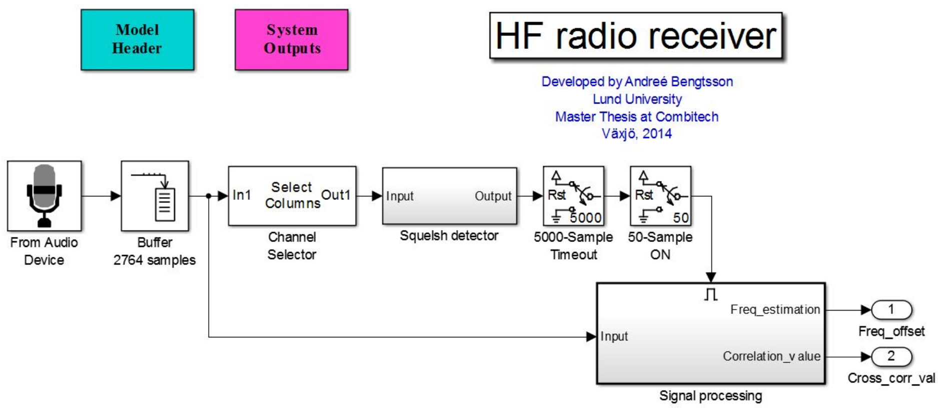

Figure 6.9: Simulink model.

these makes future computation more efficient. When using frame based signals multiple samples can be processed at once.

Channel selector

This block is used to select one of the two channels in the input signal as the upcoming squelch detector just need one channel to trigger on. What channel that are used in the squelch detector is not important for us, thus the first channel is used.

Squelch detector

The squelch detector contains a simple structure to trigger the "signal processing" block. The structure used takes the 10 first samples of the buffer, take the absolute value, sum the elements and compare the answer to a threshold. If the value exceeds the threshold level the output is "1", otherwise it is "0", see Figure 6.10. The threshold needs to be adjusted depending on the strength of the incoming signal. It will not work that good in a real receiver where the incoming signal have varying strength and contains much more than a clean signal.

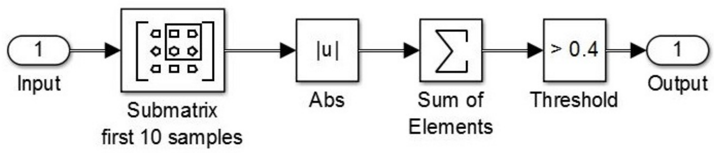

Figure 6.10: Squelch detector block.

Switch 1 & 2

The two switches can be seen in Figure 6.9 after the squelch detector. Switch 1 has a standard output of "1" while switch 2 has a standard output of "0". Switch 1 will trigger on a rising or falling edge, i.e. when the squelch detector output changes between "0" and "1". This switch will then output "0" for the next 5.000 samples. Switch 2 will trigger on a falling edge as when switch 1 turn to "0". Switch 2 will now output "1" for the next 50 samples and trigger the "signal processing" block during this time.

Signal processing

The "signal processing" block can be studied in Figure 6.11. It contains a "channel combination" block, a "frequency offset estimation" block and a "cross-correlation" block. The input is taken from the output of the buffer.

Channel combination The "channel combination" block consists of a "channel selectors" blocks that will select each of the two input channels. An "analytical" block together with the "imag" block will produce the Hilbert transform of the second channel. A filter order of 10 is used in the "analytic" block and this produce a delay of 5 samples in the system. The "delay" block is introduced on the first channel to put both channels in phase. The signals should now be the same so a "add" block is used to add them together and produce a stronger signal, see Figure 6.12.

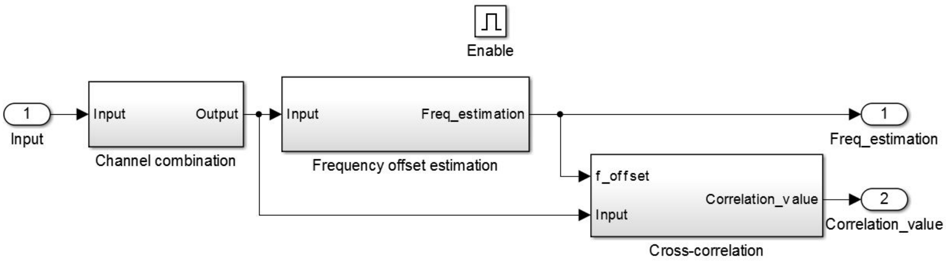

Figure 6.11: Signal processing block.

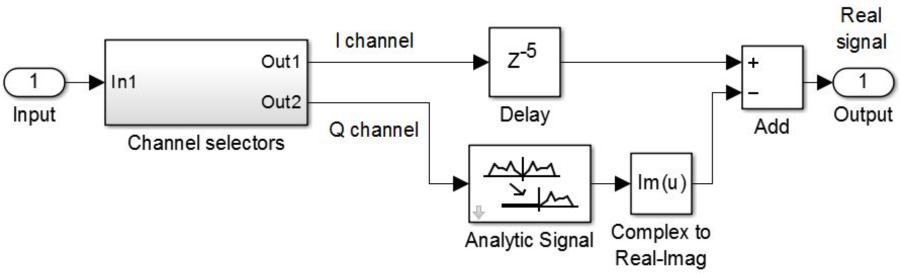

Figure 6.12: Channel combination block.

Frequency offset estimation The output from the "channel combination" block is the input to the "frequency offset estimation" block, see Figure 6.13. The conjugate of the first 844 samples (see Subsection 5.3.1) of the input will be used in the frequency estimation. A reference signal is used to estimate the frequency offset and this is imported from two mat-files consist of the real and imaginary part of the reference signal. These signals are added together to a complex representation and multiplied with the input signal. The BAF algorithm will then estimate the frequency offset using this new signal, please refer to Table 5.4 for further information on how the algorithm works.

Cross-correlation The "cross-correlation" block contains a submatrix that subtracts the preamble from the input signal. At the same time the "frequency compensation" block take the estimated frequency offset and use it to customize the reference signal to have the same offset as the input signal. Then the two signals are multiplied together and have the elements summed together. The result will be divided with the L^2-norm of the two original signals and the absolute value will be used on that result, just as the normalized cross-correlation described in Equation (4.3). As the input might be zero a block is added to the L^2-norm to prevent that this turns zero and that the normalized cross-correlation get divided by zero. See the whole subsystem in Figure 6.14.

Frequency compensation The frequency compensation is done on the reference signal; this block build up the reference signal using the frequency offset

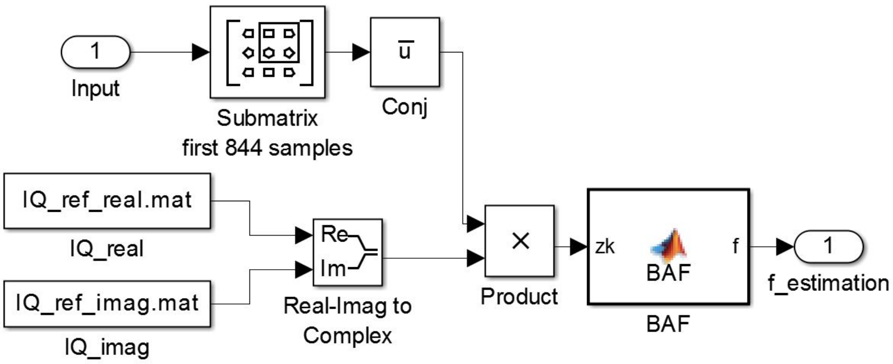

Figure 6.13: Frequency offset estimation block

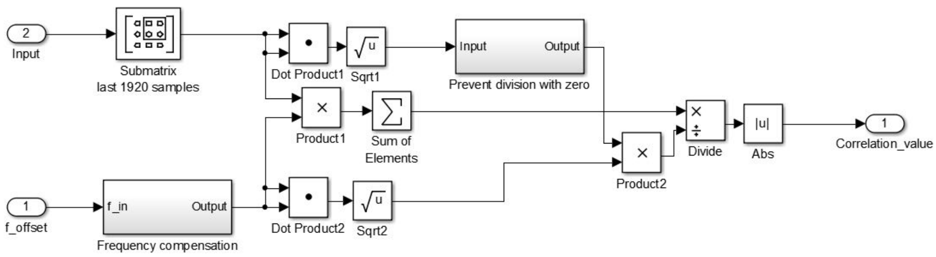

Figure 6.14: Cross-correlation block.

estimation retrieved from Figure 6.13. The estimated frequency is then fed to the "frequency compensation" block, see Figure 6.15.

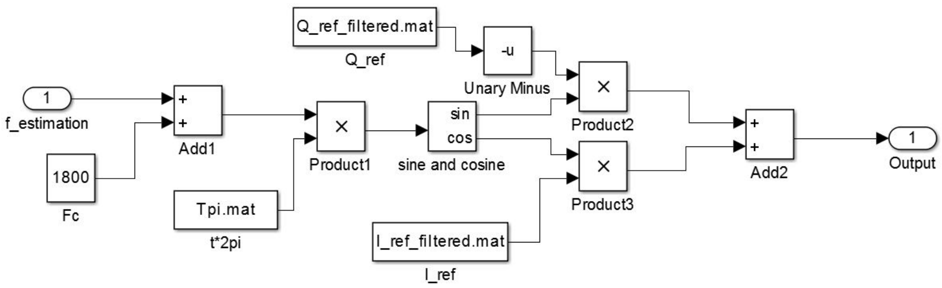

Figure 6.15: Frequency compensation block.

Prevent division by zero This subsystem examines if the signal is zero and if it shall not be used in the normalization of the cross-correlation. If the signal is zero it will be replaced by a "1" but if the signal is non-zero the subsystem will pass-through the signal. See schematics in Figure 6.16.

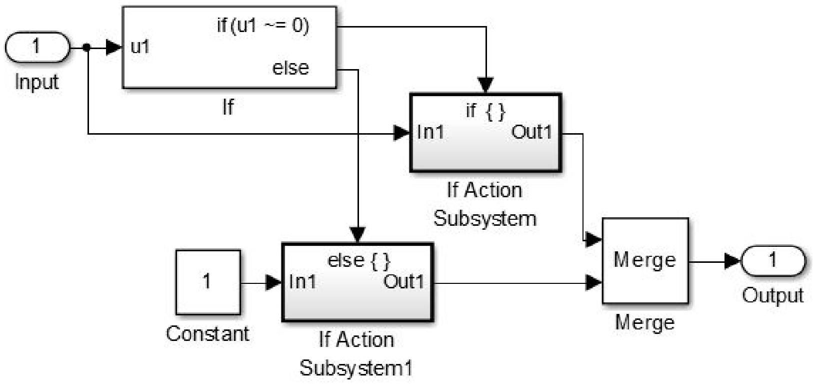

Figure 6.16: Prevent division by zero block.

Customized C-code

The two blocks shown in Figure 6.17 contains customized code that is used in the code generation described in Section 6.2.4. The threshold of the decision stage is put into this code and also the code enables output data to the command prompt as a fully working program. Table 6.5 shows the most important parameters and functions in the two blocks.

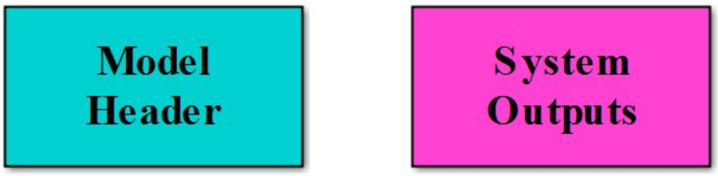

Figure 6.17: Customized C-code blocks.

6.2.4 Code generation (one detector)

The Simulink environment offers the opportunity to generate C-code from the entire model and also generate an executable version of the program. This will make it possible to run the program on a computer that does not have access to MATLAB and Simulink. Starting the program will look like in Figure 6.18.

When a call is detected the program will notify the user as shown in Figure 6.19. The program will also print the actual cross-correlation value and the estimated frequency offset on the incoming signal. For each new call the program will keep writing the next detection after the previous.

Table 6.5: Content of "Model Header" and "System Outputs"

Model header	
int k	This variable will introduce a small delay in the system that makes it possible to retrieve the largest cross-correlation value when a call is made.
float A_max	The largest cross-correlation value for one call is saved in this variable.
float f_off	This variable will keep track of the estimated frequency offset.
System Outputs	
printf	This function will output messages to the command prompt window. The messages can be seen in Figure 6.19.

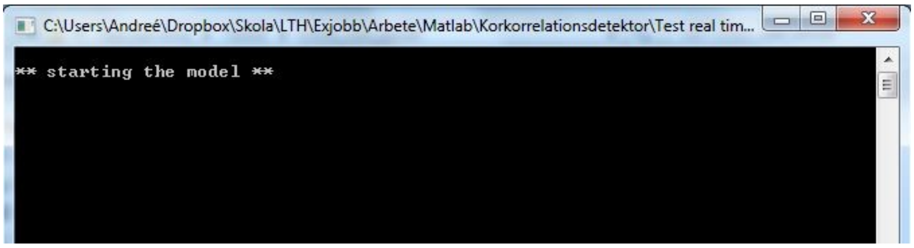

Figure 6.18: Starting the executable program.

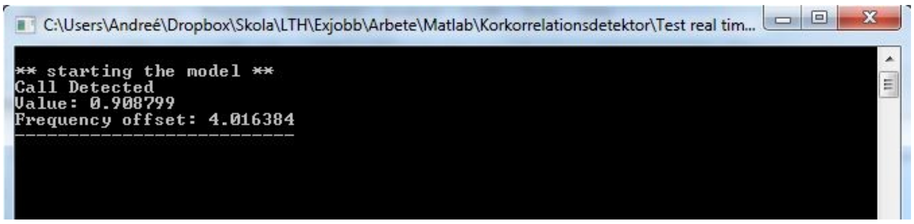

Figure 6.19: Receive a call when running the executable program.

6.2.5 Performance measurement

The main difference between the MATLAB and Simulink model is the presence of a frequency compensation algorithm in the Simulink model. Table 6.6 shows how the model perform in a distorted channel following the channels described in Table 5.1. It can be seen that the performance is similar for lower SNR. For higher SNR the performance is decreasing compared to the MATLAB implementation, especially for the good and poor channel. This may be explained by the tough requirements for higher SNR values. This make the performance more sensitive to changes, for instance due to the frequency compensation algorithm.

Table 6.6: Performance measurements using the HF channel discussed in Table 5.1.

Channel	SNR[dB]	Linking probability requirements	Using MATLAB HF simulator	Margins
Gaussian	-10	25%	35.6%	+10.6%
	-9	50%	73.1%	+23.1%
	-8	85%	90.4%	+5.4%
	-7	95%	98.9%	+3.9%
ITU-R F.520-2 Good	-8	25%	44.4%	+19.4%
	-6	50%	59.5%	+9.5%
	-3	85%	76.4% (87.8*)	-8.6%
	1	95%	88.4% (95.4*)	-6.6%
ITU-R F.520-2 Poor	-6	25%	52.2%	+27.2%
	-3	50%	63.8%	+13.8%
	0	85%	79.3% (89.6*)	-5.7%
	3	95%	89.9% (95.6*)	-5.1%

*Using a threshold of 0.18 instead of 0.27. This results in a system that fulfill the requirements set by Table 5.1.

To examine the chosen threshold level several call sequences was processed using the Simulink model. This was done on the four channels that needed a threshold value of 0.18 to meet the linking requirements. The result for the "ITU-R F.520-2 Good" channel with a SNR of -3dB can be seen in Figure 6.20. This was the case which gave the highest cross-correlation value (=0.1118) when removing the values corresponding to calls. Note that only a 10 call long sequence was examine in this example. Thus the result is not definite, but it gives a hint in what range the threshold value may be. The threshold level need to be large enough to prevent false detection, i.e. larger than 0.1118. To increase the requirements even more the results in Table 6.2 are used. This increases the lower bound of the threshold to roughly 0.15, which is seen as a lower bound in this implementation. In conclusion, our threshold value of 0.18 is above the accepted bound.

6.2.6 Simulink model (three detectors)

The detector described in Subsection 6.2.3 can be used in a parallel receiver as shown in Figure 6.21, here three detectors are put in parallel. Each detector block consists of the scheme described in Subsection 6.2.3, but the "from audio devices" block is set to listen to different sound cards. The digital clock is introduced to show the elapsed time in the application. Also the "model header" and the "system outputs" are modified to handle three detectors.

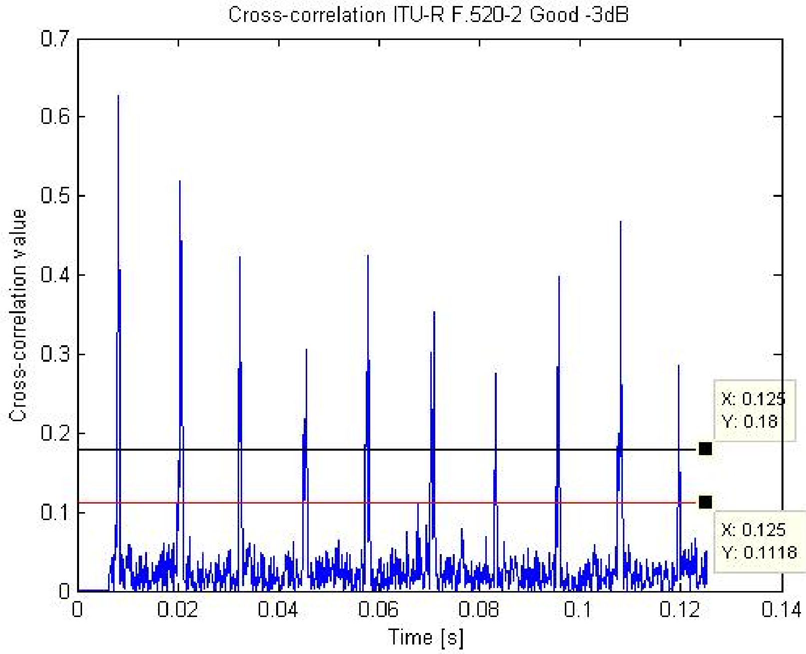

Figure 6.20: Examination of the threshold value.

RF front-end

The RF front end built in this thesis can be seen in Figure 6.22. The figure show the final prototype with all its components (except for the external radio) discussed in this subsection. Imagine that the external radio is replaced by another Tayloe detector. This will result in a three detector system that can be placed in a confined space such as the box used in Figure 6.22.

The received RF signal is connected to the "RF input" port where it is guided into a splitter that divide the signal into three equal signals, resulting in a 4.77dB signal loss (assuming no further loss in the split). However the SNR stays the same since the input noise value is much higher than the noise factor of the RF front-end module. One of the signals are guided out from the box and is down-converted to baseband using a commercial radio. The output from the radio is fed back into the box through the "Audio in" port. The two other signals are each guided to a Tayloe detector that down-convert the signal to baseband. This is done together with two signal generators which are fed through the "VCO 1" and "VCO 2" port. Also a 12V power supply is needed to drive the two Tayloe detectors and this is done through the "Power" port. The output from the Tayloe detectors are, together with the output from the commercial radio, fed to three different USB soundcards. These soundcards are connected to a USB HUB that is connected to the "Audio out" port which are connected to the PC.

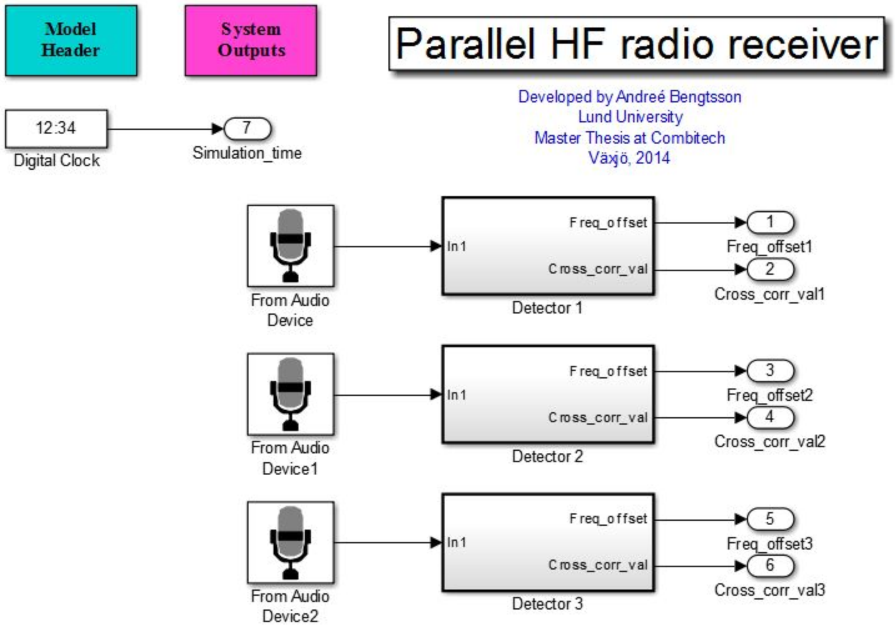

Figure 6.21: Parallel receiver model.

Code generation (three detectors)

When executing the program the output result is shown in Figure 6.23. Here three calls are received on three different channels, which are shown as call 1, 2, and 3. This program is convenient to use when the user just wants to know whether a call has been made or not. Note that the program will not say anything about the call (called station, calling station, message etc.)

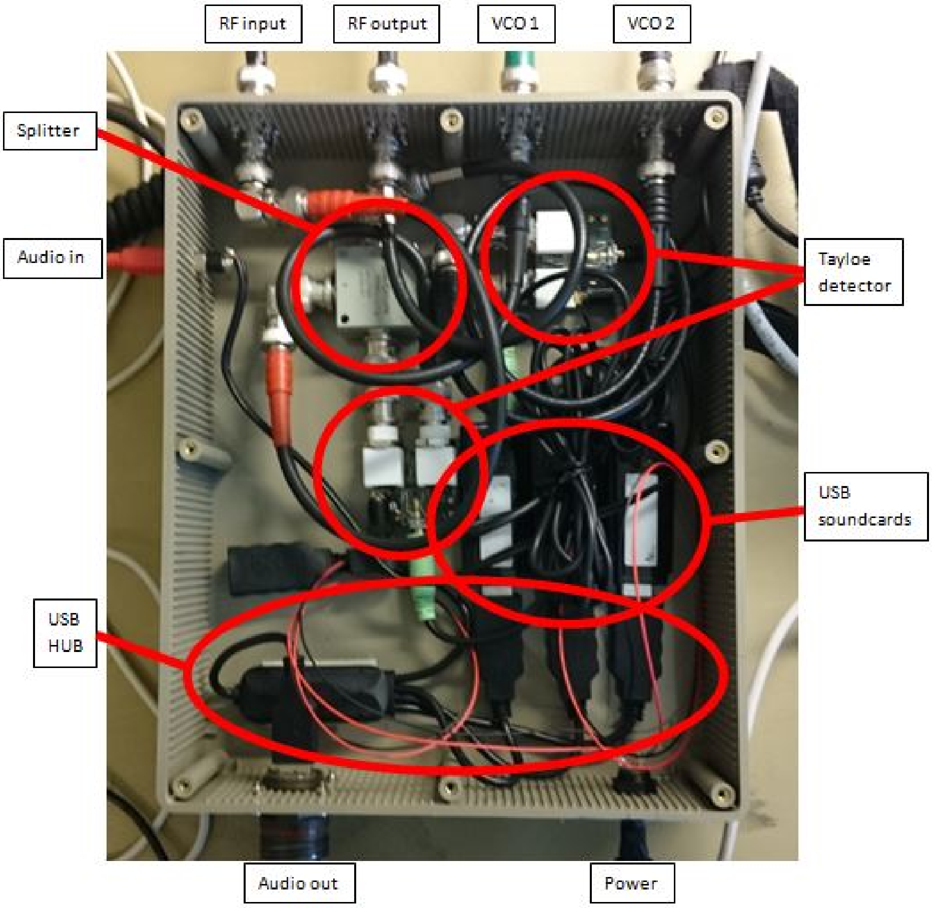

Figure 6.22: RF front-end.

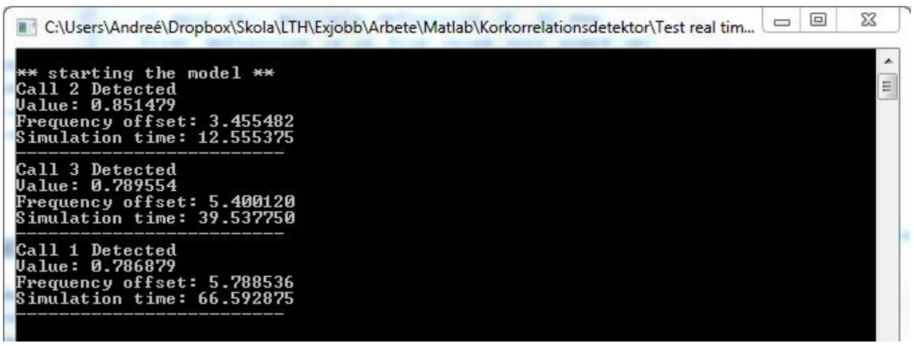

Figure 6.23: Receive a call on channel 2, 3, and 1 when running the executable program.

Chapter 7

System description

The entire communication system might look something like Figure 7.1. The regular HF 2000 station consists of NCT, RSCU and transceiver as described in Subsection 5.3.6. When a call is detected it is handed over to the regular HF 2000 station that decide whether the call is meant for this station or not. If it is, the station will keep receiving the rest of the call, otherwise the call is ignored. To get a complete system it is necessary that the parallel receiver can communicate with the NCT on the same node. This can be done using protocols that contains information about occupied channels, detection etc.

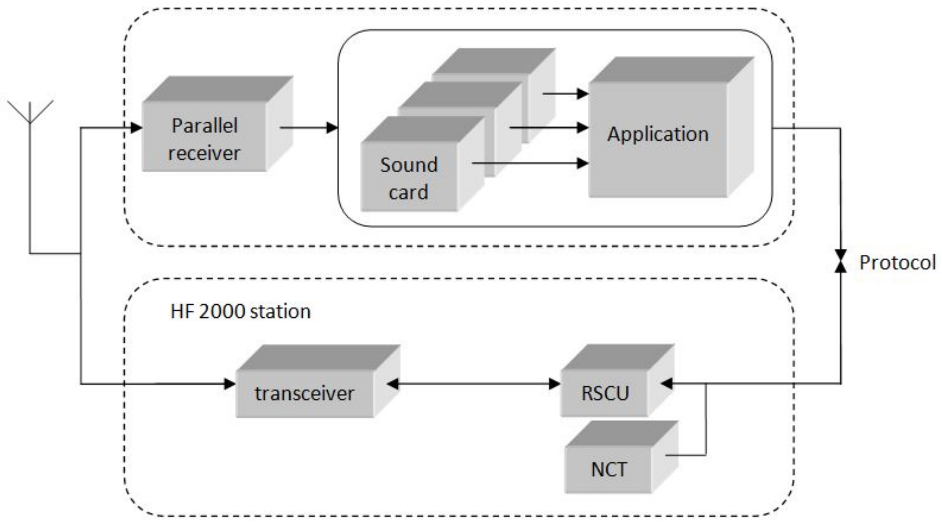

Figure 7.1: Description of the entire communication system.

7.1 Protocol

This part is not fully implemented in this thesis; the protocol communication is only performed from the receiver to the NCT. The messages are not completely developed, the purpose is to give a indication on how the whole system might look. For the NCT to communicate with the parallel receiver it will be necessary to modify the HF 2000 system and it is beyond the scope for this thesis. To establish a link between the receiver and the NCT the application FTP is used. The protocol used is a customized one implemented for this purpose only. In this case three different messages have been defined.

7.1.1 Receiver-NCT (implemented)

The NCT side of a node wants to know if a call has been made to the receiver. In this implementation the structure in Figure 7.2 is used to enable this message.

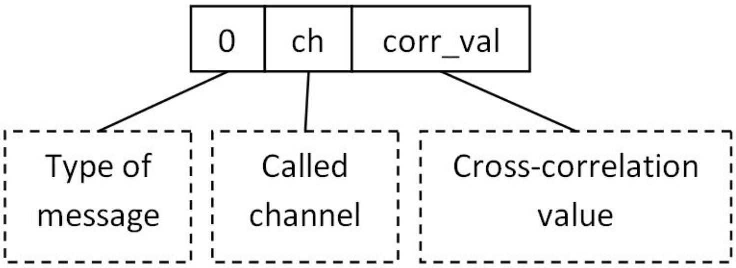

Figure 7.2: Customized protocol to be sent from the receiver to the NCT when a call is obtained.

The first value in the message in Figure 7.2 describes which type of message that is sent. Using this first value the NCT know how to read the rest of the message. The second value represents which channel the parallel receiver detected a call on and the third value represent the cross-correlation value for the detection. Figure 7.3 shows the messages being received to the NCT from the receiver during three calls on three different channels.

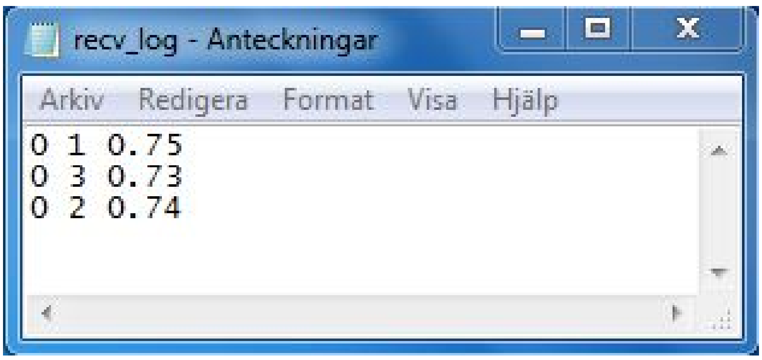

Figure 7.3: Messages received by the NCT when the receiver detects three calls.

7.1.2 NCT-Receiver (not implemented)

In a complete system the NCT also need to send information to the receiver. These messages might include information about the channel being scanned in the synchronous system or if a channel is currently being occupied.

Figure 7.4 shows an example of a protocol message that can be used to inform the receiver if a channel is currently being used by the transmitter at the same node. The first value show what type of message it is, just as described in Subsection 7.1.1. The second value represents the channel and the third value show whether it is in use or not. If the channel is in use it is not necessary for the receiver to monitor that channel. A channel may only be used by one station at the same time. Otherwise there are risks for interference between different transmissions. When the channel is no longer occupied by the station the message is sent once more. This value tells the receiver that the channel is no longer in use by the station and that the receiver can start monitor it again.

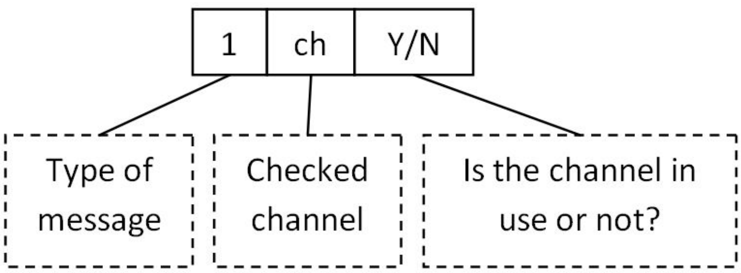

Figure 7.4: Customized protocol to be sent from the NCT to the receiver when a channel is in use or not.

Another message that will be of interest is the one that tell the receiver what channels to monitor in the channel group, see Figure 7.5. The message will be repeated for each channel in the channel group. To end the series of messages a final one will be sent consist of, for instance, a negative channel. This way the receiver knows that all the messages have been sent and that it is now up to date. The messages will be sent in the start-up of the system as well as when the channel group changes.

In a complete implementation a more extensive protocol is probably needed.

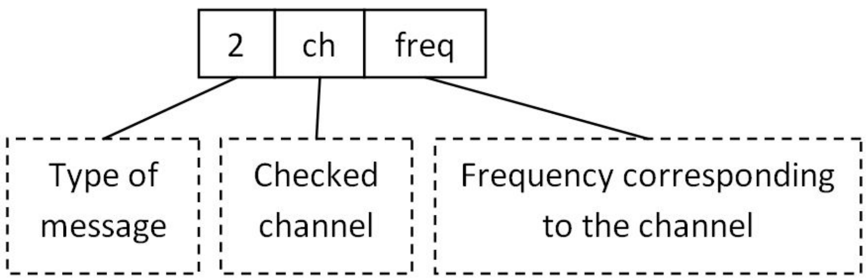

Figure 7.5: Customized protocol to be sent from the NCT to the receiver to inform what channels (frequencies) to monitor.

Chapter 8

Conclusion/Future work

A proof-of-concept demonstrator of a real-time parallel radio receiver has been developed and verified in this thesis. The purpose was to build up a prototype that has the ability to listen to several channels simultaneously. The task was successfully accomplished and a complete system description has been developed. This proof-of-concept demonstrator system can listen to three channels and decide whether a call is made or not. The receiver then reports the call to the NCT as a future possible commercial implementation might do.

For the proposed improvement to be realized in the present system some details needs to be taken care of. The dimensions of the RF front-end can be reduced substantially by being implemented on a circuit board and thus eliminating BNC connections, cables, and the dead space that exist between components. Also the signal generators need to be minimized. As discussed in the thesis there exist USB dongles that uses a crystal to achieve the clock frequency with sufficient accuracy. These devices are in the size of our USB soundcard and they have a stable reference frequency.

Another issue that needs to be considered is the signal processing. The use of a PC is not an optimal solution and even a powerful PC will have problem to process up to 64 channels simultaneously. In a real implementation a more specialized SDR is recommended. This can be a Field-Programmable Gate Array (FPGA) which is a integrated circuit intended to be configured by the user. Another alternative are the Digital Signal Processors (DSP) which are specialised microprocessors optimized for signal processing. Both these devices are commonly used in SDR applications.

For the system to be completely functional the 3G-ALE system needs to be able to read the protocols sent from the parallel receiver. The protocol need to be in a format that is understood by the 3G-ALE system.

If the system shall be compatible with present systems it is necessary for the network to know what other nodes are using the new technique. This information need to be part of the required modification of the 3G-ALE system. This information will tell the transmitting node whether it has to wait for the scanning mechanism or if it can make an instant call. The information can be included in the addresses of the different nodes in the network. This way not all nodes need to have the new technology to work. Some nodes may not need to be faster and will thus not require this ad-on.

The frequency compensation algorithm used in this thesis perform well for most channels. Still it struggles with the heavily distorted HF channel models. The frequency estimation algorithm works within a specific range. Probably a badly distorted HF channel increase the risk that the algorithm falls outside this range and the estimation becomes invalid.

There are also room for improvements in the generation of the reference signal. There was a discrepancy between the reference signal generated by the MATLAB implementation of a 3G-ALE station and samples recorded from a real 3G-ALE station. The reason for this is probably that I do not know exactly how the signal is created in present systems. For instance there might exist some filters that I am not aware of. Fully knowledge of how the signal is processed would improve the performance of the detectors even more.

References

[1] R. Berg, H, Bergzén, "Instantaneous Channel Access for 3G-ALE Systems", *HFIA*, pp. 7.2.1-7.2.8, Stockholm, Sweden, August 2013.

[2] FMV, *Teknisk systembeskrivning HF 2000*, 2005.

[3] HARRIS, *RF-5800H-MP ADVANCED TACTICAL HF RADIO - OPERATIONAL MANUAL*, USA, March 2009.

[4] International Telecommunication Union, *Handbook - Frequency-adaptive communication systems and networks in the MF/HF bands*, Switzerland, 2002.

[5] NATO, *ANNEX C TO STANAG 4538*, 2009.

[6] F.R. Kschischang, *The Hilbert Transform*, University of Toronto, October 22, 2006.

[7] P.H. Young, *Electronic Communication Techniques*, fifth edition, Upper Saddle River, New Jersey, Pearson Education, 2004.

[8] D. Tayloe, "Ultra Low Noise, High Performance, Zero IF Quadrature Product Detector and Preamplifier", *RF Design*, pp.58-69, March 2003.

[9] *PSDR.* (n.d.). Retrieved March 2, 2014, from: `www.pandatron.com/?partno=91001&tx=sdr_receiver_-_the_module_psdr`

[10] About RTL-SDR. (n.d.). Retrieved March 3, 2014, from: `www.rtl-sdr.com/about-rtl-sdr/`

[11] *PGEN45.* (n.d.). Retrieved March 2, 2014, from: `www.pandatron.com/?partno=91042&tx=pgen45_-_generator_8_khz_to_45_mhz`

[12] *PGEN170.* (n.d.). Retrieved March 2, 2014, from: `www.pandatron.com/?partno=91056&tx=pgen170_-_generator_5_mhz_to_170_mhz`

[13] Rafael Microelectronics *R820T High Performance Low Power Advanced Digital TV Silicon Tuner Datasheet*, Taiwan, 2011.

[14] KN0CK HF SDR Receiver. (n.d.). Retrieved June 24, 2014, from: `http://www.kn0ck.com/HF_SDR/`

[15] *MODIFIED RTL-SDR WITH TEMPERATURE CONTROLLED OSCILLATOR (TCXO)*, 2013, Retrieved March 3, 2014, from: `www.rtl-sdr.com/modified-rtl-sdr-temperature-controlled-oscillator/`

[16] 1090 MHZ-SHOP by jetvision.de. (n.d.). Retrieved June 24, 2014, from: `http://h2204566.stratoserver.net/SmartStore.NET/de/sdr-dongle-r820trtl2832u-with-tcxo-2ppm#.U6nQGJR_uNy`

[17] T.D. Thulesius, "Bygg en kortvågsmottagare med en DVB-T-pinne", *QTC*, no.6, pp.4-7, 2014.

[18] Specifications. (n.d.). Retrieved March 4, 2014, from `http://www.funcubedongle.com/?page_id=1201`

[19] E-Systems, *NEW RESEARCH LAB LEADS TO UNIQUE RADIO RECEIVER*, vol.5, no.4, pp.6-7, May 1985.

[20] J. Mitola III, Software Radios - Survey, Critical Evaluation and Future Directions, *National Telesystems Conference*, pp.13/15-13/23, Washington DC, USA, 1992.

[21] W.H. Tuttlebee, *Software Defined Radio: Enabling Technologies*, Chichester: John Wiley & Sons Ltd, 2002.

[22] Wireless Inovation Forum. (n.d.). Retrieved January 27, 2014, from: `www.wirelessinnovation.org/introduction_to_sdr`

[23] International Telecommunication Union, *F.520-2*, 1992, Retrieved February 25, 2014, from: `www.itu.int/dms_pubrec/itu-r/rec/f/R-REC-F.520-2-199203-W!!PDF-E.pdf`

[24] W.N. Furman, J.W. Nieto, *UNDERSTANDING HF CHANNEL SIMULATOR REQUIREMENTS IN ORDER TO REDUCE HF MODEM PERFORMANCE MEASUREMENT VARIABILITY*, Rochester, New York: Harris Corporation, RF Communications Division, 2001.

[25] C.C. Watterson, J.R. Juroshek, W.D. Bensema, Experimental Confirmation of an HF Channel Model, *IEEE*, vol.18, no.6, pp.792-803, 1970.

[26] Terratec, *AUREON DUAL USB*, 2013, Retrieved June 11, 2014, from: `www.terratec.net/media/filer_public/10542-en.pdf`

[27] G. Lindell, *Introduction to Digital Communications*, Lund: Lund University, 2006.

[28] J.J. Carr, *The Technician's Radio Receiver Handbook*, Elsevier Inc., 2000.

[29] W. Steinert, L. Friederich, S. Godtmann, A. Pollok, N. Hadaschik, G. Ascheid, et al. *A Least-Squares Based Data-Aided Algorithm for Carrier Frequency Estimation*, 2006.

[30] Sounds of the Modern Commercial and Military Digital Station. 1998. Retrieved June 27, 2014, from: `http://www.milspec.ca/modems/modems.html`

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2014-408

http://www.eit.lth.se