Linus Ekvall
linekv-2@student.ltu.se
Carl-Henrik Hanquist
hancar-0@student.ltu.se
Kristoffer Hägglund
krihgg-3@student.ltu.se
Last revision: January 26, 2026

Project in Engineering Physics and Electrical Engineering - Engineering Physics – F7042T

A Software Radio for Wireless Communication in Solids and Liquids

---

Abstract

Damaged materials that are somewhat hidden can be hard to investigate. For example a rock bolt that is wedged into a mountain. A possible way to detect the damage may be to investigate if the digital communication channel of a damaged rock bolt is different from an intact one. But in order to test this method one must have a reliable digital communication system that provides this information. In this report the foundation for such a system is presented. The result is a working digital communication system with limited propagation lengths for different materials. The system is able to transmit and receive information through various mediums such as water, Plexiglas and aluminium with small propagation lengths and various signal to noise ratios. The system should be able to handle propagation through longer materials but for this, a signal amplifier is needed.

Contents

1 Introduction

1.1 Background

This project is based on the Orthogonal Frequency Division Multiplexing method (OFDM) which is a method used to transfer data in digital communication systems. The method divides the available bandwidth into a set of orthogonal narrow banded subcarriers.
Future and ongoing research at Luleå Technical University would gain from having an easy to use and flexible software that can transmit, receive, decode and show channel properties from this type of communication system. There are similar software's already on the market such as the open source program GNU Radio[1], but it sometimes lack in documentation, mathematical explanations, simple diagnostics and easy modification possibilities.
Therefore this project was initiated to investigate if it was possible to create a flexible and simple MATLAB based software that can be used as an alternative for research and testing in the future.

Previous in related areas such as [2] where a DSP based OFDM system was done over an acoustic channel. The previous project report was a theoretical basis for the system in this repor. However since the field has developed a lot since the last project it cold be simplified and directly interfaced to MATLAB. The setup of the system was very similar to [3] in which the software GNU Radio was used. The authors of this paper has been serving as supervisors for this project.

Today OFDM, or some modification of it, is one of the most widely used data modulation technique in use. It has been used in wireless networks as well as in audio and video broadcasts [4] to mention a few examples. Today it is used in 4G telecommunication and will also be used in the up and coming new telecommunication standard 5G where tests are well under way for commercial applications [5],[6].

1.2 Goal

The objective of this project was to create an easy to use MATLAB based software radio. It should be created with the ability to be adapted to both radio and ultrasonic communication systems. An investigation into what extent the USRP N210 could be integrated with MATLAB for this specific system and what possible limitations it would bring would also be performed. The long term goal was to produce a software that could be used or modified for ongoing and future research within the field of digital communication at Luleå University of Technology.

1.3 Scope

The software should be able to initially send and receive bit signals via ultrasound through different short mediums suitable for ultrasonic transmission. If this was achieved, experiments on a 1m long rock bolt would be performed. After this experiment and if there was time left for the project, experiments with Radio Frequency communication would be attempted.

The system would be limited to sending and receiving bit streams as the break down and reconstruction of a source to and from bits could be added later on. This being said a text-to-bit and bit-to-text function would be implemented as this simplified troubleshooting to some extent.

The system should be constructed to be as modular as possible so that parts of the software radio could be expanded upon or replaced with ease. To be easy to use it should be executed in easy to understand scripts and a basic user guide should be written.

2 Theory

2.1 Digital modulation techniques

There are three major classes of digital modulation techniques. Those are Phase-shift keying (PSK), Amplitude-shift keying (ASK) and Frequency-shift keying (FSK). In this section the focus will be on some of the different types of PSK and ASK constellations. For a more in depth discussion see [7].

One of the PSK methods is the Quadrature Phase Shift Keying (QPSK), which is a two dimensional linear modulation that maps binary data in the set of two into complex numbers. Where a sequence of bits are mapped into different positions in the complex plane as seen in Figure(1).

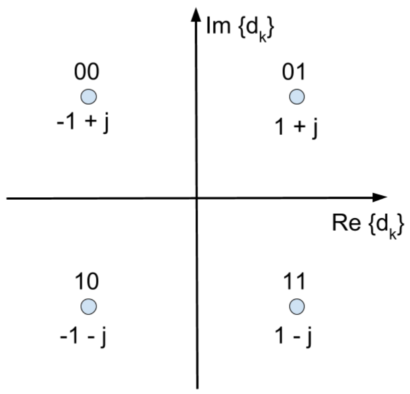


Figure 1: QPSK mapping.

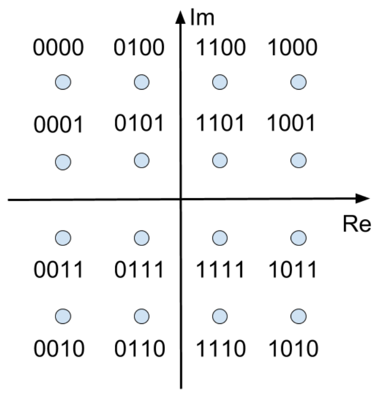


Figure 2: QAM mapping.

For example, the binary sequence {00011011} would become {-1+j,1+j,-1-j,1-j} as shown in Figure(1). With this approach, only the phase is shifted and the amplitude or the distance from the origin is equivalent for all four points. It is however possible to take advantage of the amplitude in the complex plane, which leads to the ASK approach.

The Quadrature Amplitude Modulation (QAM) can utilise both phase and amplitude to map the bit sequences. As seen in Figure(2), 16-QAM uses 16 modulation points and thus each point represents a sequence of 4 bits. The main difference between these two digital modulation techniques is that (given the same bandwidth noise and interference levels) the 16-QAM will support higher data rate while the QPSK is more robust in the sense that it will function better in low Signal to noise ratio (SNR) channels.

2.2 Orthogonal frequency division multiplexing (OFDM)

OFDM was first described in the 1960's and has since then been used in a variety of fields such as ADSL, LTE, Wifi systems, etc. It is a technique used to divide the available bandwidth into a set of orthogonal narrow banded information symbols called subcarriers. Since these subcarriers are orthogonal to each other there will ideally be no Inter Carrier Interference (ICI), which means that the different subcarriers will not interfere with each other even though they partly overlap in the frequency domain. By exploiting this method it is possible to send several information carrying symbols within one OFDM symbol as shown in Figure(3).

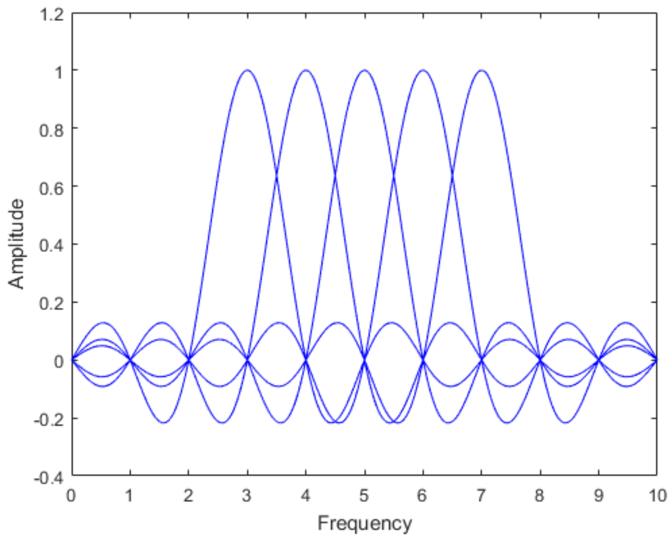


Figure 3: Theoretical example of orthogonal subcarriers overlapping in the frequency domain.

2.3 OFDM modulation

As seen in Figure(4), the OFDM modulation technique consists of a series of operations. The available bandwidth is divided into N subcarriers, each containing a complex symbol derived from the digital modulation technique. The system starts with performing an Inverse Fast Fourier Transform (IFFT) on the digital modulated data and adding a cyclic prefix (explained further below) before transmission through the channel, that is the propagation medium. After propagation, the cyclic prefix is removed and a Fast Fourier Transfer (FFT) is performed. Here the different ioeratuibs will be explained in a more detailed manner.

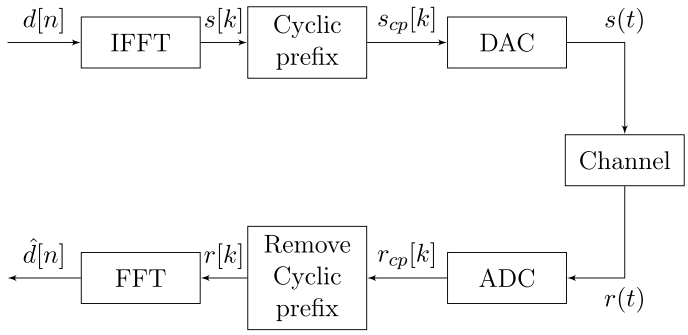


Figure 4: The OFDM modulation system.

Once the digital modulation technique has been established the data $d[n]$ is modulated from time- to frequency domain through an inverse discrete Fourier transform according to

$$s[k] = \sum_{n=0}^{N-1} d[n] e^{j\frac{2\pi nk}{N}} \tag{1}$$

A cyclic prefix is added to the symbol. This signal is now converted by a Digital to Analog Converter (DAC) and then the data is transmitted through the channel according to

$$r(t) = s(t) * h(t) + n(t) \tag{2}$$

where $h(t)$ is the impulse response of the channel and $n(t)$ is additive noise. The cyclic prefix is removed at this point since it is no longer needed. The signal is modulated back from time to frequency domain using FFT corresponding to

$$\hat{d}[n] = \sum_{k=0}^{N-1} r[k] e^{-j\frac{2\pi nk}{N}} \tag{3}$$

and the data should be ready for the next step that is decoding.

2.3.1 Pilot configuration

Pilot signals are signals usually transmitted over a single frequency over a communications system. This is done in order to supervise, control, synchronise and equalise the system to name a few of the applications. Pilot symbols are known both to the transmitter and receiver and they are often generated as pseudo-random sequences. By observing how the pilots are affected by the channel, it is possible to undo the effects and compensate for eventual perturbations.

There are several approaches on how the pilots can be implemented with the data. These are known as different pilot configurations where a specific type of configuration is preferred in a certain situation and another in a different situation. Two common arrangements are block-type and comb-type configurations [8], [9].

The block-type pilot configuration is useful in situations where the channel is not changing rapidly. The pilots are inserted periodically at different times, occupying a a complete OFDM-symbol. All subcarriers are used as pilots for a specific period. If the channel is constant, there will be no channel estimation error. The estimation can be performed using either LS or MMSE.

The comb-type pilot configuration was developed to deal with more time-varying channels. The pilots are uniformly inserted between data-carrying subcarriers. They are present in every symbol, basically some subcarriers are dedicated to carrying pilot values which are then used to estimate the channel at those specific points. In order to get a complete estimate for the channel with this configuration, interpolation using the channel information received from the pilots is necessary.

The two different pilot arrangements are visualised in time and frequency domain in Figure(5) and Figure(6). The red dots represents the pilots and the black dots are the data points. Each row on the axis labelled 'frequency' represents a subcarrier and each column is a symbol for an instance of time.

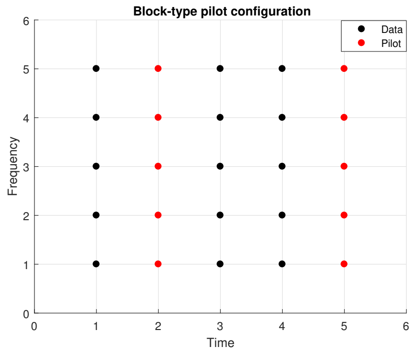


Figure 5: Block-type pilot configuration

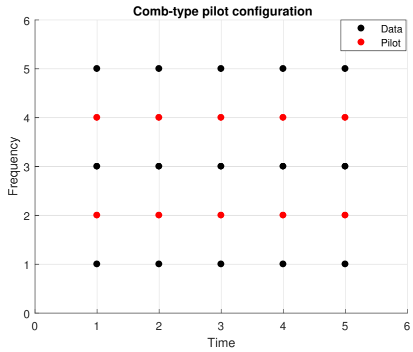


Figure 6: Comb-type pilot configuration

2.3.2 Cyclic prefix

Most of the time the data being transmitted requires several OFDM-symbols. These symbols are transmitted in succession. This may subject the data to interference such as inter-symbol interference (ISI). This is a form of distortion of a signal where one symbol interferes with subsequent symbols. The symbol is delayed and spread out of its allotted time interval, causing it to intrude on subsequent symbols time interval There are several methods to combat the effects of ISI. Here, this problem is approached with the usage of a cyclic prefix. The cyclic prefix introduces a guard interval to the symbol by copying a part of the end of the symbol as placing it first, acting as a buffer, as shown in figure(7).

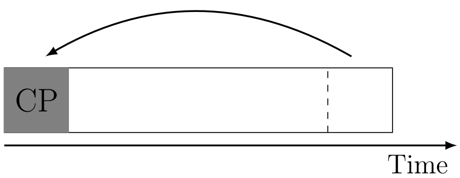


Figure 7: Cyclic prefix of an OFDM-symbol

By copying a part of the symbol instead of adding a random sequence for example allows the receiver to identify the end points of each symbol and correctly correlate the information, thereby eliminating the interference problem. In addition, the cyclic prefix also helps to make an initial estimation of time and frequency synchronisation, using the same reasoning correlation of known information that arrives over time The cost of using this technique is that it is applied in the time domain of the OFDM-symbol, making it slightly longer, reducing data capacity.

2.3.3 Time Delay and Frequency Correction

When transferred over a channel a time delay and frequency offset gets introduced into the received signal. How bad the signal is affected depends on the quality of the channel as well as oscillator mismatch between the transmitter and receiver. There exists several different methods with different level of complexity to perform this step. In more sophisticated systems with complex channels a coarse synchronisation is performed, followed by a fine synchronisation. Due to the limited scope of this project and the stable channel only a fine synchronisation is used. The method chosen was a maximum likelihood estimation of the time and frequency offset using a log likelihood function [10]. The method relies upon the cyclic prefix being correlated with the end of the OFDM symbol and uncorrelated with every other part. Thus it uses a observation interval which is big enough to view a single symbol and its cyclic prefix, see figure(8).

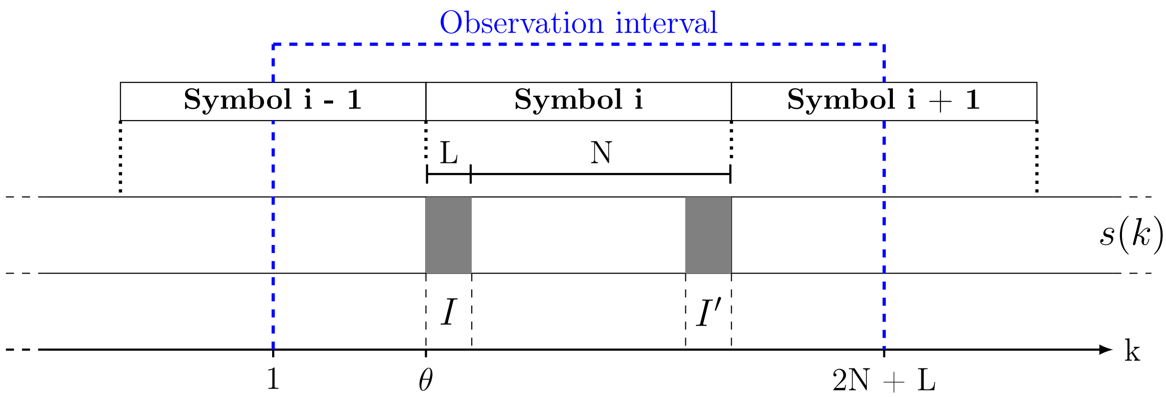


Figure 8: Observation interval for detecting time delay in a sequence of OFDM symbols. The I part of the symbol is the cyclic prefix, which is a copy of the end of the symbol denoted I'

The data from the observation interval $\bar{r}$ is then used to calculate the log-likelihood function. In this project the compressed version of the log-likelihood function was used. The compressed version assumes that a rough synchronisation has already been performed, but with a stable channel this was estimated to be unnecessary. The downside of this is that only a frequency offset of $|\epsilon| < \frac{1}{2}$ is detectable. It also is assumed that the SNR of the signal is high enough to be approximated to $+\infty$, which is not true for a real system. With these assumptions the time delay θ and the frequency offset ϵ can be estimated as

$$\hat{\theta}_{ML} = \arg\max_{\theta}\{|\gamma(\theta)| - \Phi(\theta)\} \tag{4}$$

$$\hat{\epsilon}_{ML} = -\frac{1}{2\pi}\angle\gamma(\hat{\theta}_{ML}) \tag{5}$$

where γ and Φ are calculated as

$$\gamma(m) = \sum_{k=m}^{m+L-1} r(k)r^*(k+N) \tag{6}$$

$$\Phi(m) = \frac{1}{2}\sum_{k=m}^{m+L-1} |r(k)|^2 + |r(k+N)|^2 \tag{7}$$

where $r(\cdot)$ are elements from the observation interval $\bar{r}$.

2.3.4 Channel Estimation

A channel can be seen as a Linear system where the sent signal x is modified by the channel h and affected by additive noise n. The result of this linear system is the received signal y. In frequency domain this linear system is represented by a single equation

$$y(\omega) = h(\omega)x(\omega) + n(\omega) \tag{8}$$

where one is interested in recovering the original message x, which is easily done by dividing with the channel

$$x(\omega) = \frac{y(\omega)}{h(\omega)} + \frac{n(\omega)}{h(\omega)} \approx \frac{y(\omega)}{h(\omega)} \tag{9}$$

where the minus sign of the noise is omitted since it is assumed to be AWGN noise, so the sign does not matter. If the noise is small enough we can approximate the sent signal by simply dividing the received signal with the channel frequency response. To do that we recognise the fact that equations (8) and (9) applies for each subcarrier of the system, since each subcarrier is on a different frequency. This implies that in the frequency domain each subcarriers amplitude is affected by the channel. To be more precise the amplitude of the subcarriers real and imaginary part is affected, since the signal is complex valued. A noiseless example of this is shown in Figure(9).

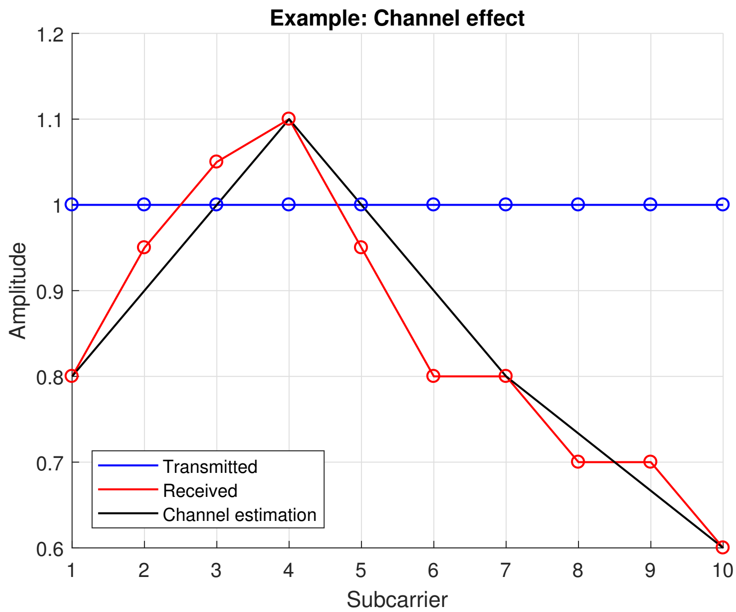


Figure 9: Example of a simple channels effect on the transmitted signal. The amplitude change illustrated affects the real and imaginary part of the signal separately.

The blue line represents the sent message $x(\omega)$ while the red line represents the received message $y(\omega)$. The black line is the estimation of the channel. The estimation is an interpolation between pilot carriers which carry values known in both the transmitter and receiver. In this example the subcarriers numbered 1,4,7 and 10 are pilot carriers. The channel can then be calculated for these carriers by rewriting equation (9) to

$$\hat{h}(\omega) \approx \frac{y(\omega)}{x(\omega)} \tag{10}$$

and simply interpolating between these values. As is visible in Figure(9) this is not always exact and the channel estimation will not always be identical to what is happening in reality. Though there are several sophisticated methods to perform a channel estimation they are beyond the scope of this project.

When the channel has been estimated an equalisation of the signal can be performed. Equalisation is basically undoing the effects of the channel by using equation (9). The equalisation is performed on both the real and imaginary part of the complex signal, which will reconstruct the sent signal if the channel estimation is good enough.

3 Method

3.1 Experimental Setup

The experimental setup can be seen in Figure(10). It consisted of two PCs with MATLAB installed, the PCs interface to a Universal Software Radio Peripheral (USRP) via Ethernet cable. The USRPs were of the type Ettus Research model N210. The transmitting USRP had a LFTX daughterboard and the receiving USRP had a LFRX daughterboard. Each daughterboard had a range of 0–30 MHz, a 12 bit ADC/DAC and a maximum output power of 100 mW. Each USRP were connected by coaxial cable to a Olympus Videoscan (382-SU) ultrasonic transducer with a center frequency of 3.5 MHz.

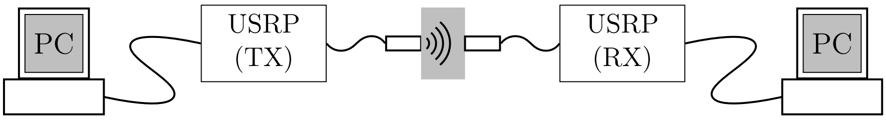


Figure 10: Experimental setup

The medium between the transducers was changed between plexiglass, aluminium, water, and a rockbolt, but could be switched out for any medium suitable for ultrasonic wave transmission.

To confirm that the transmitting USRP was sending data a real time spectrum analyser of model Tektronix RSA306B was used instead of the receiving USRP. After confirmation of a signal the receiving USRP could be connected and tests of the communication system were done.

3.2 Design Choices

3.2.1 Pilots

Several types of pilot configurations were considered but since none of the project members had previous experience with OFDM and since there was an interest in examining the channel, the Comb-type pilot configuration as in Figure(6) was chosen. It was by far the easiest to implement and was easily expanded into only sending pilots to examine the changes in the channel.

3.2.2 Bit Map Constellation

The map constellation chosen for the project was grey coded QPSK, the reason being the simplicity and clear decision regions. The code was however constructed to be expandable and supports M-QAM constellations by simply changing the configuration file.

3.2.3 Time Delay and Frequency Estimation

The time delay and frequency offset was assumed to be very stable in solid mediums such as aluminium. Therefore only a fine frequency offset correction algorithm was implemented as more advanced synchronisation methods were considered too complicated to implement with the project time limit.

3.3 Full System setup

A discrete overview of the communication system can be seen in Figure(11). The first part from the Bit Source to the signal being sent to the USRP is handled by the transmitter script, see

Figure(12a). After the signal has been received by the receiving USRP it is sent to the second PC which processes the received signal in the receiver script, see Figure(12b).

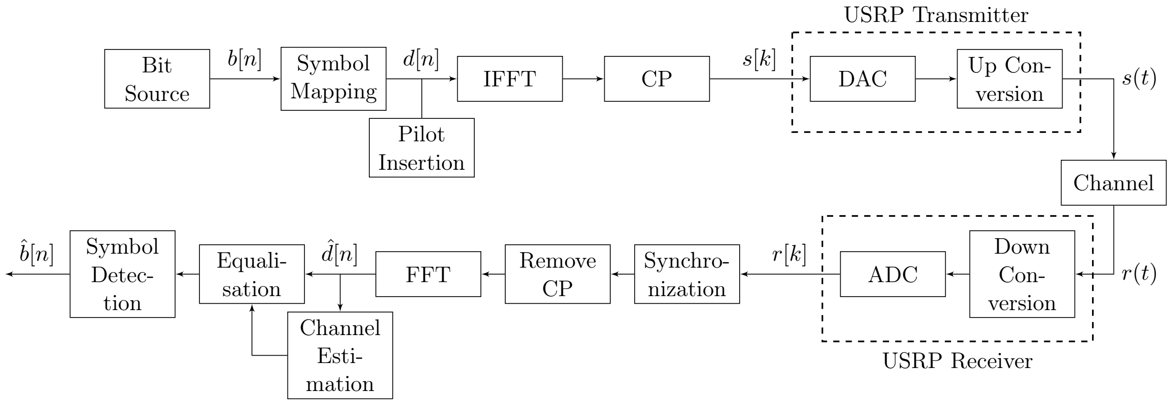
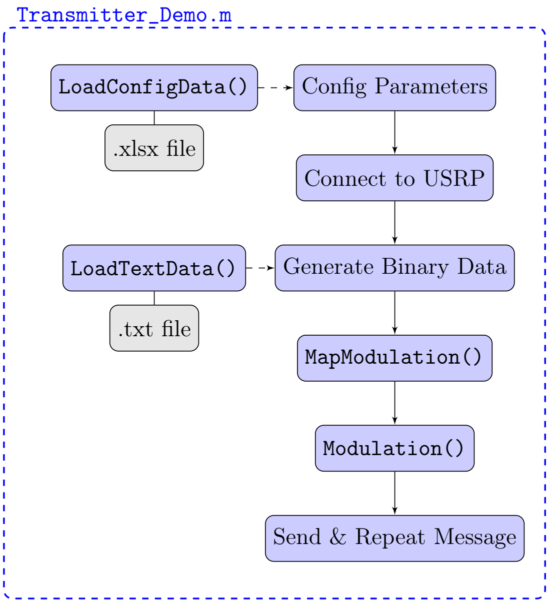


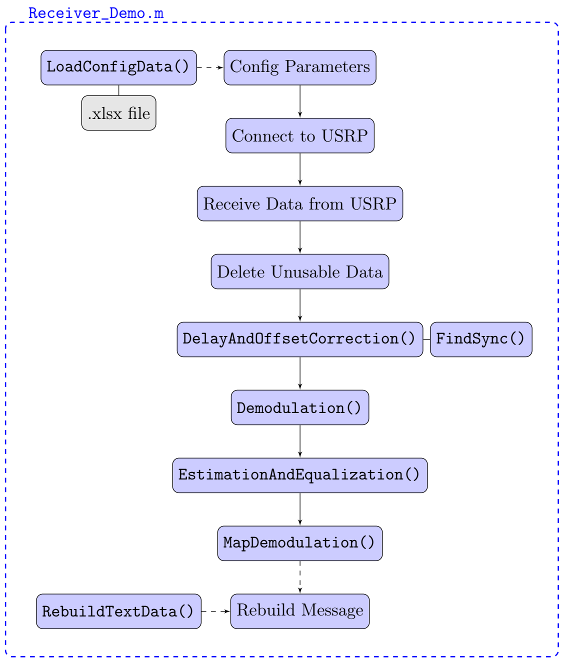
Figure 11: Simplified discrete version of the system

3.4 Code overview

The scripts were constructed to be modular and having separate functions being swapped out if need be. Figure(12a) and Figure(12b) show an overview of the code.



(a) Flowchart over transmitter script



(b) Flowchart over Receiver script

Figure 12: Flowcharts over transmitter and receiver scripts

The parts with a dashed line indicate that they are optional and not needed to run the scripts. The overview is rather rough and a more detailed example of how the code and each function works is detailed more with code comments and function descriptions. Additional documentation is found in the user manual accompanying the code.

3.5 Transmitter

The transmitter was constructed to initially set up all the configurations and start the connection to the USRP. Then it performs all the modulation of the data. When the data is prepared it enters a loop which continously sends the same repeated message to the USRP.

3.6 Receiver

The receiver starts of by setting up the configurations and opening a connection to the USRP. It then starts saving the data from the receiver, however it starts doing this before the connection is fully established with the USRP. So before processing any of the data the first 1000 values are removed, as around the first 700-1000 USRP frames will only be zeros and not hold any real data, or even noise. Then the synchronisation is performed and the CP removed. After this the data can be demodulated from OFDM and the channel estimation is performed and the received data is equalised, then the pilots are removed. Next the data is demodulated back to bits and the original message can be rebuilt.

4 Results

In this section some results are presented where the propagation mediums were water, plexiglass and aluminium. No results for the rock bolt are shown since no real communication could be performed through it due to low SNR.

4.1 Propagation through water

Figure(13) shows the received data after transmission through approximately 5cm of water before equalisation, while Figure(14) shows the same data after equalisation had been performed. The red circle in Figure(14), Figure(16) and Figure(17) signify a hard coded energy threshold (ϵ) that deletes any "junk data" used to fill out half filled OFDM symbols.

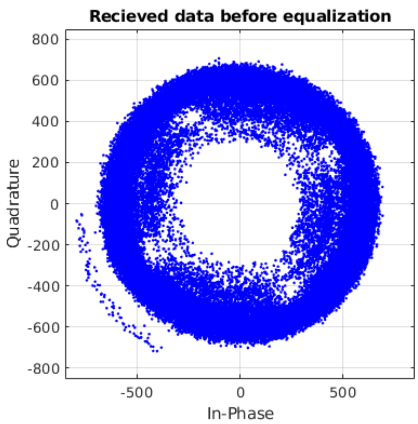


Figure 13: Received data before equalization. (Water ≈ 5cm).

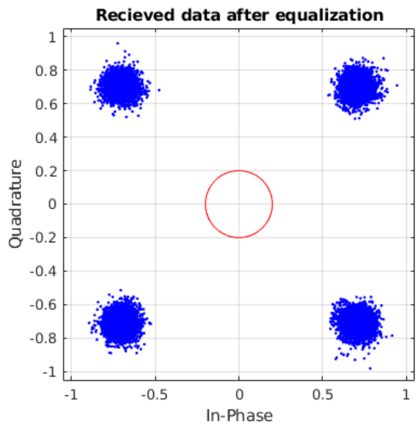


Figure 14: Received data after equalization. (Water ≈ 5cm).

The absolute value of the waters channel can be seen in Figure(15). The plot was calculated by only sending pilots through the system and estimating the channel. 512 subcarriers were used and each colored line is the result of the channel estimation of a single received OFDM-symbol.

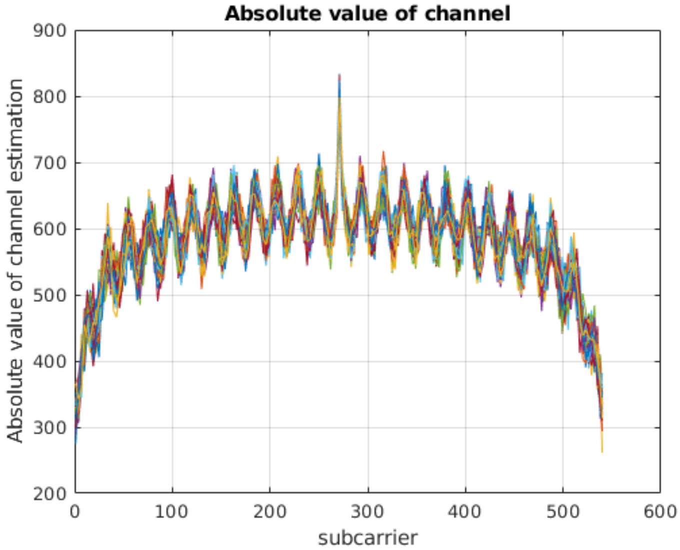


Figure 15: Channel estimation after propagation through ≈ 5 cm of water. 512 subcarriers were used per symbol.

4.2 Plexiglass and Aluminium

Received data after equalisation for plexiglass and aluminium can be seen in Figure(16) and Figure(17) respectively. The thickness of the material was approximately 5 cm in both experiments.

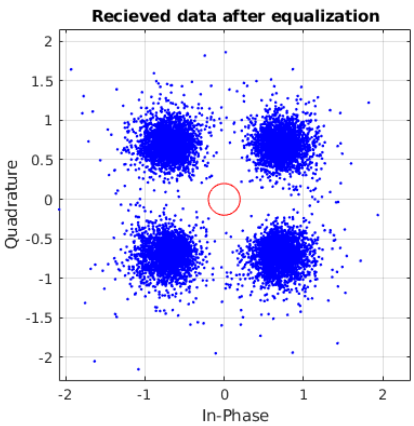


Figure 16: QPSK demapping using plexiglass as propagation medium.

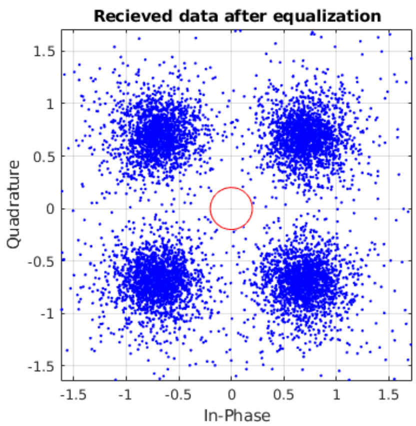


Figure 17: QPSK demapping using aluminium as propagation medium.

5 Discussion

5.1 Experimental Results

The results of the equalisation had a great impact on the system, this is illustrated in Figure(13) and Figure(14). The received signal mapping before equalisation is both amplified and phase shifted significantly compared to the signal mapping after equalisation. The remnants of the phase shift that the fine synchronisation could not take care of is taken care of by the channel estimation. If this was not the case the data would be completely corrupted and a more sophisticated synchronisation would have to be implemented.

The solid materials showed to have lower SNR. As the SNR gets lower the the demapped data passes the border to another quadrant in the QPSK case. This implies data corruption. For plexiglas there was some corruption as can be seen in Figure(16).

Figure(17) showed that aluminium has even worse SNR with increased data corruption as a result. As the SNR gets lower the four clouds of the QPSK demapping gets a larger variance which is dependent on signal magnitude, propagation medium and the propagation length. Experiments not shown here were done on 10 cm of aluminium which further decreased the SNR. The experiments on the rock bolt were not able to get any visible data through and the OFDM spectrum was only hardly visible with the spectrum analysator. Thus those results were left out of this report.

The increased spread of the decoded constellation aside from crossing the decision regions also enter the junk data threshold limit ϵ, the red circle in Figure(14), Figure(16) and Figure(17). Which led to some data corruption as well. The threshold could be lowered, but then junk data might not be deleted, which again introduces corruption.

The channel of water proved to be the most stable, compared to the other mediums. Aside from a small ripple the channel had a smooth Gaussian-like shape as can be seen in Figure(15). While the plot itself is of little importance it exemplifies the stability of the channel as each new line corresponds to an individual OFDM symbol sent at different time instances. Since there hardly is any variation it proves that the channel is rather stable as was assumed.

The small variations across the frequency in Figure(15) were present in all experiments and all mediums. While it was not able to be identified it was theorised to be caused by either the electronics, ultrasound transducer resonance modes or ultrasonic reflections from previous symbols in the medium. By varying the length of a transmission medium the cause could probably be identified or some of the possible causes eliminated.

5.2 Performance

The performance of the system fell off target for several different reasons. One being that the SNR in solids was lower than expected and that the transmitting USRP had a fixed gain on the output. From experimenting with different mediums the overall performance was as shown in table 1.

Medium	SNR	Data Corruption
Rock bolt	Terrible	N/A
Aluminium	Poor	High
Plexiglas	OK	Low
Water	Excellent	None

Table 1: Overview of the communication performance for different mediums

In general the solid mediums had a low SNR, which got worse with the length of the medium. This was not unexpected, but meant that no data could be sent and received through the rock bolt. The low SNR is in part caused by the coupling between the transceivers and the solid, where the direction of the tranceiver and the use coupling liquid is of great importance.

When messaging in water however, a complete recovery of the sent message was achieved. The SNR in water was good enough to use 16-QAM and still have full recovery of the message, the SNR in the solids was not good enough for this.

Regarding bit rates in the communication no real diagnostic, plot or measure of this has been implemented in the code. The plan was to implement this, but with the time of the project running out, fixing critical problems discovered late in the testing was prioritised. Some calculations of the bit rate was performed and when using QPSK the bit rate was above 1 Mbit/s in some experiments. The bit rate could easily be increased much higher by using other constellations or adjusting bandwidth and the number of subcarriers.

5.3 Known Problems

The constructed transmitter and receiver had several issues that have mostly been worked around but could be eliminated completely in future versions.

5.3.1 Message Detection

When sending data across the communication system the message will consist of any arbitrary number of ones and zeros. But for the code to work each OFDM symbol needs to be completely filled. This was solved by adding junk data to any partially filled OFDM symbol, specifically setting the remaining subcarriers to a value of zero. This zero value would however be affected by the noise in the transmission channel and electronics and no longer be zero. To filter out this junk data a parameter named ϵ was introduced which removes any data below a certain energy limit in the receiver. ϵ was manually tuned in the config struct until a good setting was found. The drawback was that if the channel has too much noise it ran the risk of deleting actual data as the spread of the received signals was increased and overlaps with the junk data. So if the system was run in a channel with good SNR like water, ϵ was sufficient to assure complete reconstruction of the data. As can be seen by the clear cut constellation in Figure(14). While plexiglas and aluminium had a worse SNR and there was some data corruption. The simple solution to this problem would be to construct a start/end of message sequence and include this in the sent data. Then junk data can still be used to fill out the partially filled OFDM symbols but ϵ would no longer be needed.

5.3.2 MATLAB - USRP Connection Problems

When connecting to the USRP there were several issues which we were not able to solve.

RX Busy: If the transmitter script is cancelled and run again it sometimes signal that the USRP is busy, as if it is stuck in a loop of still sending data. If one encounters this message the only solution we found was to restart MATLAB. This error can be avoided for the most part by executing `clc,clear` in the command window between cancelling the transmitter script and running the script again. The same error occurs if one aborts the receiver script before it completes its loops.

Under-/Over running: Sometimes the received data in a transmission is exactly zero, this can occur in only a few OFDM symbols or part of a received USRP frame. It is clearly visible when plotting the absolute value of the received complex signal where gaps in the signal are seen as the value is completely zero. It is also seen in the channel estimation as it is completely off from all other estimations as it tries to estimate the received data, the result of that estimation is that some data points get set to completely absurd values in the received scatterplot after equalisation. The exact zero tells us that this is not real data as even just receiving data from the USRP without anything connected to it will introduce some noise from the electronics. So this data was most likely due to some form of Under-/Over running in the system or a communication problem between the PC and the USRP in general. Sadly there was not enough time to look into this problem in the project, but from experience we noticed that this can sometimes be solved by increasing the number of subcarriers for a given bandwidth.

`FastSendDatagramThreshold` **errors:** This error occurs often and does not seem to affect the communication for the most part. It is described further and solutions to it are presented in the common problems and fixes part of the MATLAB toolbox documentation [11]. We were unsure how this affects the communication as we couldn't see any errors caused by it, but we followed the suggested solutions and the error disappeared, no changes were observed and everything was still the same. Two weeks later it came back again but we did not bother to remove it again.

Speed related errors: The speed related issues does not seem to affect the communication with only a few subcarriers and a relatively narrow bandwidth. However as the number of subcarriers and bandwidth is increased it can be a problem and crash MATLAB. We were not able to solve this problem or even look into it due to the time limit. It could be that this was only seen as a problem as our computers used USB-to-Ethernet adapters to interface with the USRPs and that the error disappears if one only uses an Ethernet connection.

5.3.3 Unable to Perform RF Signaling

Initially we wanted to be able to send and receive over Radio frequencies around 2.5GHz since equipment for this was available. In the end this was not achieved as the fine synchronisation was not enough to compensate for the channel effects seen in a radio channel. Implementing a more sophisticated synchronisation method would solve this problem and make the software radio viable for use in RF. Additionally a more complicated pilot configuration might also be needed, since RF channels tend to be complicated and sensitive to disturbances.

5.4 Future Work

The main effort of future work is recommended to be focused on identifying and correcting the problems presented in the section above. When this is done other areas could be expanded upon in the programming, below are some ideas which were considered during the project. Most of them were a result of discussions with the supervisors of the project, while others were considered for implementation but rejected due to the time limit of the project.

5.4.1 Expanded diagnostics and performance measures

Since the project time was not enough to build functions that would automatically calculate and plot measurements of the performance this was left as a future work part. For example measures and plots of the Bitrate, subcarrier spacing, OFDM symbol length in time, current Bit error rate (BER) and a theoretical comparison to mention a few might be interesting to implement.

5.4.2 Amplifier Construction for the Transmitter

The reason that this might be necessary is to boost the transmitter signal high enough to be able to travel through a metal rock bolt. During the experiments the SNR was too low to perform any meaningful communication through it. The cause of this was that the gain of the transmitter daughter board could not be adjusted. An analog amplifier should be enough to boost the SNR enough to communicate through a longer metal medium.

5.4.3 Single USRP as Transmitter and Receiver

If one were able to use a single USRP as both transmitter and receiver one could perform pulse-echo measurements with ultrasonic transducers. Introducing OFDM into these measurements would give a channel estimation which then could be connected to physical changes in the medium.

5.4.4 Graphical User Interface

If used by other students without knowledge about the system it might be a good idea to implement a GUI for easier use of the programs. The program could then be used in labs to illustrate the OFDM technology and show how different parameters affect the signal. The software radio should however be fully constructed and implemented before attempting any GUI construction as it most likely will be difficult to adapt any GUI to changes in the code afterwards.

6 Conclusion

In this report a software radio in MATLAB was presented together with the basic theory and design choices made during the construction. Several issues and problems with the software and hardware are described and some ideas for future work was outlined. Overall the project succeeded in the construction of the software, which is capable of sending and receiving data through different mediums. It fell short however on delivering communication through longer metal mediums such as a rock bolt and communication over radio frequencies. The design of the software should however be modular enough to expand into radio frequencies and a simple amplifier might be enough to make it viable for longer metal mediums.

Acknowledgements

The authors would like to thank Professor Johan Carlson, Professor Jaap van de Beek and Ph.D. student Medhat Mohamad. All belonging to the Signals and Systems division, of the Department of Computer Science, Electrical and Space Engineering at Luleå Technical University. In addition to making this project possible, they have provided guidance and insightful discussion throughout the whole project.

References

[1] Inc. The GNU Radio Foundation. *GNU Radio*. 2018. URL: https://www.gnuradio.org/ (visited on 01/29/2018).

[2] Niklas Bellander et al. "DSP baserat OFDM system ùver akustisk kanal". In: (1997).

[3] Johan E Carlson, Jaap van de Beek, and Medhat Mohamad. "Mbit/second communication through a rock bolt using ultrasound". In: *Ultrasonics Symposium (IUS), 2017 IEEE International*. IEEE. 2017, pp. 1–4.

[4] Taewon Hwang et al. "OFDM and its wireless applications: A survey". In: *IEEE transactions on Vehicular Technology* 58.4 (2009), pp. 1673–1694.

[5] NyTeknik. *Då spurtar utvecklingen av 5g*. 2017. URL: https://www.nyteknik.se/digitalisering/ericsson-da-spurtar-utvecklingen-av-5g-6886152#conversion-122831618 (visited on 01/12/2018).

[6] NyTeknik. *Ericsson och Telia har smygstartat 5g - ombord på färja*. 2017. URL: https://www.nyteknik.se/innovation/ericsson-och-telia-har-smygstartat-5g-ombord-pa-farja-6874507#conversion-122831618 (visited on 01/12/2018).

[7] Upamanyu Madhow. "Fundamentals of Digital Communication". In: (2008).

[8] Youssefi My Abdelkader et al. "Optimal spacing design for pilots in OFDM systems over multipath fading channels". In: *Communications and Network* 2.4 (2010), pp. 221–229.

[9] Sinem Coleri et al. "Channel estimation techniques based on pilot arrangement in OFDM systems". In: *IEEE Transactions on broadcasting* 48.3 (2002), pp. 223–229.

[10] Jan-Jaap Van de Beek, Magnus Sandell, and Per Ola Borjesson. "ML estimation of time and frequency offset in OFDM systems". In: *IEEE transactions on signal processing* 45.7 (1997), pp. 1800–1805.

[11] MathWorks®. *Common Problems and Fixes*. 2017. URL: https://se.mathworks.com/help/supportpkg/usrpradio/ug/common-problems-and-fixes.html?s%5C_tid=gn%5C_loc%5C_drop (visited on 01/12/2018).