CHALMERS
UNIVERSITY OF TECHNOLOGY

# Software-Defined Radio Testbed for 6G Research

## Implementation of Synchronized Transceivers to Enable Experiments in D-MIMO Communications

Bachelor thesis in Electrical Engineering

Vilgot Bengtsson, Melker Wibeck, Joel Wolf-Watz

**DEPARTMENT OF ELECTRICAL ENGINEERING**
CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2025
www.chalmers.se

Bachelor's thesis

# Software-Defined Radio Testbed for 6G Research

Implementation of Synchronized Transceivers to Enable Experiments in D-MIMO Communications

Vilgot Bengtsson
Melker Wibeck
Joel Wolf-Watz


CHALMERS

Department of Electrical Engineering
*Division of Communications, Antennas, and Optical Networks*
Chalmers University of Technology
Gothenburg, Sweden 2025

Software-Defined Radio Testbed for 6G Research

Implementation of Synchronized Transceivers to Enable Experiments in D-MIMO Communications



**Supervisor:**

Kaan Okumus
Department of Electrical Engineering
Division of Communications, Antennas, and Optical Networks
Chalmers University of Technology

**Examiner:**

Erik Ström
Department of Electrical Engineering
Division of Communications, Antennas, and Optical Networks
Chalmers University of Technology

Bachelor's Thesis 2025
Department of Electrical Engineering
Division of Communications, Antennas, and Optical Networks
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: The scatterplot of a demodulated QAM signal where the receiver does not account for, among other things, symbol timing- or frequency offsets.

Typeset in LaTeX
Gothenburg, Sweden 2025

# Abstract

As the demand on speed, coverage and bandwidth continues to increase for wireless communications, the sixth generation of mobile networks builds on and extends existing technologies. There is a testbed at Chalmers University of Technology focusing on the promising technology of distributed multiple input multiple output (D-MIMO). This project aims at assisting the D-MIMO testbed by constructing a flexible user equipment based on a software defined radio. Specifically, this is done by configuring the USRP B205mini-i, a universal software radio peripheral (USRP) from Ettus Research, in MATLAB to fully utilize its capabilities in signal processing and wireless communication. Trough simulations in MATLAB and physical testing of both wired and wireless transmissions, several key aspects are evaluated: the maximum bandwidth and its limitations on data rate, the relationship between signal duration and the on-board memory of the USRP, and the possibility of synchronizing multiple USRPs. Results show a maximum bandwidth of 41.24 MHz and a symbol rate of 10.31 Msym/s. Moreover, the result does not show a strict limitation on the signal duration as the USRP seems to stream data directly to and from the host computer. Synchronization is possible between two USRPs with a precision of a few microseconds at best. The USRP B205mini-i proves to be a flexible and cost-effective candidate for UE in the D-MIMO testbed. However, performance enhancements, such as lower-level configuration via FPGA programming, may be necessary depending on future system requirements.



Keywords: distributed MIMO, software-defined radio, USRP, 6G, wireless communication, signal processing, synchronization, testbed.

# Sammanfattning

I takt med att kraven på hastighet, täckning och bandbredd fortsätter att öka för trådlös kommunikation, bygger den sjätte generationens mobilnät vidare på, och utökar befintliga teknologier. Vid Chalmers tekniska högskola finns en testuppställning som fokuserar på den lovande tekniken distribuerad multiple input multiple output (D-MIMO). Detta projekt syftar till att stödja D-MIMO-testuppställningen genom att konstruera en flexibel användarutrustning baserad på en mjukvarudefinierad radio. Specifikt görs detta genom att konfigurera en USRP B205mini-i, en universell mjukvarudefinierad radiomodul från Ettus Research, i MATLAB för att fullt ut utnyttja dess kapacitet inom signalbehandling och trådlös kommunikation. Genom simuleringar i MATLAB och fysiska tester av både trådbundna och trådlösa överföringar utvärderas flera viktiga aspekter: maximal bandbredd och dess begränsningar för datahastighet, sambandet mellan signallängd och USRP:ns interna minne, samt möjligheten att synkronisera flera USRP-enheter. Resultaten visar en maximal bandbredd på 41,24 MHz och en symbolhastighet på 10,31 Msym/s. Vidare visar resultaten ingen tydlig begränsning för signallängden eftersom USRP:n verkar strömma data direkt till och från värddatorn. Synkronisering är möjlig mellan två USRP-enheter med en noggrannhet på några få mikrosekunder som bäst. USRP B205mini-i visar sig vara en flexibel och kostnadseffektiv kandidat som användarutrustning i D-MIMO-testuppställningen. Dock kan prestandaförbättringar, såsom konfiguration på lägre nivå via FPGA-programmering, bli nödvändig beroende på framtida systemkrav.

# Acknowledgements

Before we begin, we would like a moment to express our gratitude towards the people who, through their technical expertise and continuous support, made this project possible.

Firstly, to our supervisor Kaan Okumus, who provided tremendous help in writing the thesis, as well as any other documents and presentations created as part of the study. Without his help, the document currently in your hands would not exist.

Secondly, to our technical advisor Lise Aabel, who on multiple occasions provided us with technical expertise required to advance the project. Without her, we would have never been able to implement the transceivers, much less synchronize them.

Finally, to our examiner Erik Ström, who besides administering the project also provided us with opportunities to share our findings with interested people. These experiences were some of the most memorable and encouraging parts of the project.

Vilgot Bengtsson, Melker Wibeck and Joel Wolf-Watz

Gothenburg, May 2025

# List of acronyms

Below is a list of acronyms used throughout the thesis listed in alphabetical order:

| | |
|---|---|
| 5G | Fifth generation of cellular network technology |
| 6G | Sixth generation of cellular network technology |
| D-MIMO | Distributed multiple input multiple output |
| EVM | Error vector magnitude |
| FIR | Finite impulse response |
| FPGA | Field programmable gate array |
| ISI | Intersymbol interference |
| MCR | Master clock rate |
| MIMO | Multiple input multiple output |
| OBW | Occupied bandwidth |
| PPS | Pulse per second |
| QAM | Quadrature amplitude modulation |
| RF | Radio frequency |
| RRC | Root-raised-cosine |
| SDR | Software defined radio |
| SPS | Samples per symbol |
| STO | Symbol timing offset |
| UE | User equipment |
| USRP | Universal software radio peripheral |

# List of figures

# List of tables

# Contents

# 1
# Introduction

Today's society depends heavily on various communication systems, such as mobile networks, enabling devices, and in extension people, to communicate with each other regardless of their physical distance. As the fifth generation of mobile wireless technology (5G) is becoming the industry standard, predicted to carry 80% of the data traffic by 2030 [1], sights are already set on the sixth generation (6G). With every new generation, the number of connected devices increases, as do the demands on speed, coverage, and bandwidth.

To increase the network capacity, one approach in 5G technology was the expansion of small-cell technology in contrast to the traditional macro-cell approach where a single base station has a large area of service. This meant each base station had a smaller area of service, decreasing the required transmit power and number of connected user equipments (UE) per station [2]. Moreover, the capability for more advanced multiple input multiple output (MIMO) technologies, more specifically massive MIMO, was introduced [3]. MIMO works by transmitting multiple messages over the same channel which can be decoded separately by multiple antennas in the receiver [4]. This is in contrast to just serving one user at a time, and thereby increasing the link capacity per base station. Massive MIMO increased the number of radios transmitting and receiving simultaneously per base station [3].

Further development of MIMO will play a crucial role in meeting higher demand for network capacity, speed, and energy efficiency, hence being one of the major areas of research for 6G. Distributed MIMO (D-MIMO) is one promising development potentially having the capability of meeting these requirements [5]. D-MIMO is an extension of MIMO technology that utilizes multiple distributed antennas instead of having them grouped at a single location, enabling the removal of cell boundaries and a move into a cell-free space. This means UEs will communicate with several access points simultaneously, in contrast to the conventional scheme of one dedicated base station per UE [6][7].

To implement solutions such as D-MIMO in 6G and future wireless generations, it is necessary to investigate the physical limits, feasibility, and algorithm performance. There is a testbed at Chalmers University of Technology [8] which is used to test D-MIMO in practice, as the theoretical principles for an optimal system are well established. One approach to facilitate practical investigations in this research, is the Universal Software Radio Peripheral (USRP), a series of software-defined radios (SDR). According to [9], the low costs of SDRs have made them attractive to many

proof of concepts, with USRPs in particular being mentioned as a promising solution. Hence, the possibility of integrating USRPs with the D-MIMO testbed at Chalmers should be investigated further, as they could prove a user-friendly and cost-effective addition to the current setup.

## 1.1 Purpose

The purpose of this project was to design a flexible UE using a USRP B205mini-i from Ettus Research that could be used for wireless communication experiments with the D-MIMO testbed at Chalmers. This would be accomplished by developing a MATLAB program for the USRP, featuring signal processing as well as methods to transmit and receive data wirelessly. To evaluate the performance of the system, the following research questions were to be addressed:

- What is the maximum signal bandwidth, and by extension the highest feasible symbol rate of the system?
- What is the maximum signal duration, and how does it relate to the USRPs on-board memory?
- Is it possible to synchronize two USRP modules to transmit simultaneously?

## 1.2 Scope and limitations

This study was conducted with access to two USRP modules, subsequently limiting all experiments to include a maximum of two UEs. In the context of this project, a UE refers to any radio frequency (RF) transceiver with signal processing capabilities. Interfacing with the USRPs was carried out exclusively using high-level programming languages and did not involve any direct adjustments to the device's field-programmable gate array (FPGA). Moreover, this project worked exclusively with data in the physical layer, neglecting any functionality present in the other layers of the Open Systems Interconnection (OSI) reference model [10]. Additionally, all experiments assumed that the transmitted signal was known at the receiving side, eliminating the need for a pilot sequence or preambles. This decision was made to focus solely on signal transmission and reception using the USRP, without addressing network protocol functionality.

As this project tests the performance of a communication system under certain conditions, a threshold must be defined for what constitutes a satisfactory or successful transmission. The transmitted signals were modulated with quadrature amplitude modulation (QAM) at modulation order $M = 16$ and a threshold was set in accordance with the new radio (NR) standards [11], which state that the error vector magnitude (EVM) for a communication system performing 16-QAM must be below 12.5 %.

# 2

# Theoretical background

This chapter covers some fundamental concepts on the subject of communication systems engineering required for this project. The main focus is on the modulation and demodulation of continuous wave signals, as well as the non-ideal effects that must be taken into account when constructing a physical receiver.

In a communication system, information travels through a channel that connects transmitters and receivers. A channel may be either a guided medium, such as a coaxial cable [12], or a non-guided medium, such as the free space between two antennas [13]. An information-bearing signal must conform to the limitations of the medium through which it propagates. According to [14], the continuous nature of all physical media is incompatible with discrete bit streams of data that are typically to be transmitted. Thus, the bit stream must be modulated onto a waveform before being transmitted. This process is illustrated in Fig. 1.

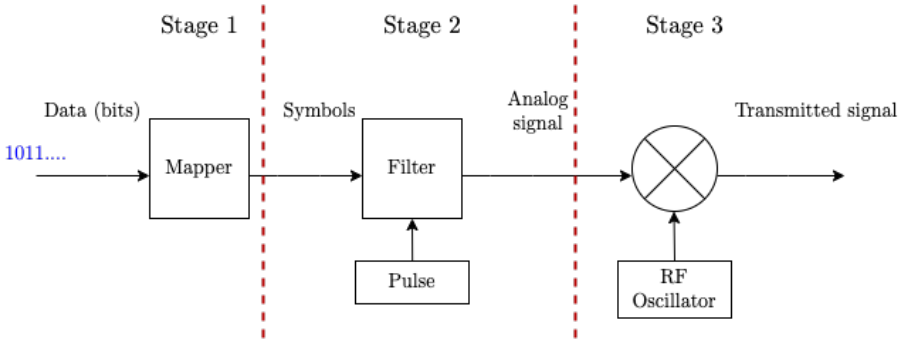



**Figure 1:** Block diagram of a transmitter.

The stages shown in Fig. 1 will be explained in detail in the following sections. The focus will be on stage 1, bit to symbol mapping, and stage 2, symbols to analog signal, since these form the foundation of the work of this project. The mixing with a high frequency carrier wave, stage 3, is required for transmission in the correct frequency band.

## 2.1 Symbol mapping and signal constellations

A sinusoid has three tunable properties: amplitude, phase, and frequency. By alternating these in a systematic way, it is possible to modulate data onto an analog wave, and in doing so creating a means of transporting digital information [15].

Taking amplitude shift keying (ASK) as an example, in its most basic form, there are two states, or amplitude levels that are transmitted, which are interpreted in the receiver as either a binary 1 or 0. The amplitude levels are denoted as symbols $a_k$ which are restricted to an alphabet $\mathcal{A}$ so that $a_k \in \mathcal{A}$. The size of $|\mathcal{A}| = M$ determines the number of bits represented by a symbol by the relation $\log_2 M$; for example 1 bit per symbol in binary modulation. More amplitude levels (i.e., symbols) can be introduced to increase the number of bits per symbol, resulting in a more effective bit transfer and consequently increased bit rate [14]. This process is depicted on stage 1 in Fig. 1. The symbols are often mapped to bits using Gray code, so that neighboring symbols differ by as few bit flips as possible, which has proven to be most efficient in minimizing bit errors [16].

To further increase the information density on a wave, ASK and phase shift keying (PSK) can be combined to QAM, which is one of the most commonly used modulation schemes in modern communication systems [15]. QAM uses two carrier signals, one in-phase and one quadrature, represented mathematically by cosine and sine, respectively. These two waves are orthogonal to minimize disturbance, and are added into a superposed transmission wave. To modulate this wave, complex symbols are introduced as $a_k = a_k^I + ja_K^Q$ where the real part is multiplied with the cosine carrier wave and the imaginary part multiplied with the sine carrier wave [14]. Samples of such a modulated wave are often referred to as IQ-samples. This results in amplitude modulation in two dimensions and therefore improves information density. The receiver can then extract the in-phase and quadrature waves separately and decode the information [17].

When introducing a second amplitude dimension, the need for a visualization of the complex alphabet is rising. This is typically done by plotting the complex alphabet as a set of points in a complex plane, known as a constellation diagram [14] depicted in Fig. 2. The x-axis represents the amplitude of the in-phase component and the y-axis represents the amplitude of the quadrature component. The projection of a point on the axes resembles the peak amplitude in the respective carrier [15]. The bit pattern corresponding with the symbol is also displayed.

## 2.2 Pulse shaping and filtering

When a bit pattern has been translated into a symbol set, the next step in transmission is to filter the symbols with a pulse as shown in stage 2 in Fig. 1. This is applied to limit the bandwidth of the transmitted signal as the discrete samples are transformed into a smooth signal [14]. The modulated signal $s(t)$ can be written as:

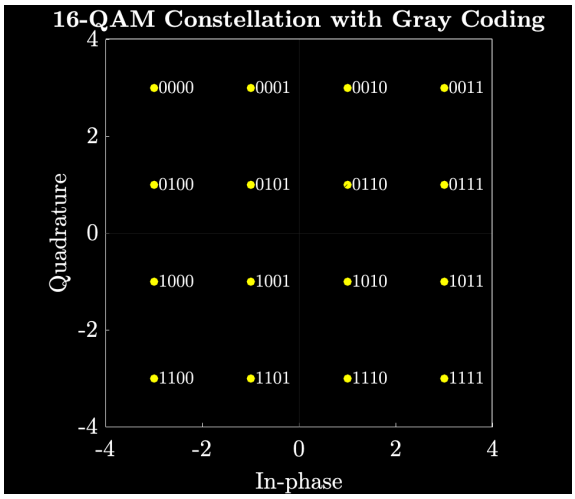



**Figure 2:** 16-QAM complex symbols plotted in the complex plane with corresponding gray coded bit patterns.

$$s(t) = \sum_{k=-\infty}^{\infty} a_k g(t - kT) \tag{1}$$

where $a_k$ is the complex symbol in the QAM symbol set, T is the symbol duration, and $g(t - kT)$ is the $k$:th time shifted pulse. Observe that (1) can be seen as the convolution between the symbols and the pulse train.

Moving on, it is important to avoid intersymbol interference (ISI) at the receiver to make an accurate demodulation of the signal. ISI is the interference caused by neighboring symbols and could thus be a source of error during demodulation [14]. Hence, the characteristics of the received signal should ideally be ISI-free. In the simplified case where the channel is not restricted to a limited bandwidth and is noise-free, the received signal can be seen as the convolution between the transmitted signal and a finite impulse response (FIR) filter:

$$r_x(t) = g(t) * f(t) \tag{2}$$

where $f(t)$ is the receive filter [18]. If the received signal $r_x(t)$ satisfies being ISI-free, it is called a Nyquist pulse [19]. When designing the transmitted pulse g(t), it should be designed such that the convolution in (2) is ISI-free. According to [14], it is common to use a matched filter in (2) which is defined by:

$$f(t) = t_{MF}(t) = g^*(-t) \tag{3}$$

meaning that the receive filter should be the time reversed complex conjugate to the transmitted pulse $g(t)$.

Distortion giving rise to ISI is unavoidable since channels always have limited bandwidth and a signal limited in frequency must have infinite duration in time [20]. However, there are ways of minimizing the distortion by selecting an appropriate

pulse. As stated in [21], it is necessary to balance the trade-off between signal-to-noise ratio and bandwidth efficiency when choosing an optimal pulse. Rectangular pulses are known for their simplicity but their broad frequency distribution results in low bandwidth efficiency. It was concluded that the optimal pulse satisfying the requirements of bandwidth efficiency and minimal distortion is the raised cosine pulse. Another appropriate pulse shape that can be used is the root raised cosine (RRC), which is the square root of the raised cosine pulse. The roll-off factor $\beta$ varies between 0 and 1 where a high value imply a more gradual transition in frequency domain, improving the spectral containment. The bandwidth of a RRC pulse meeting the Nyquist criteria of being ISI-free can be expressed as:

$$W = (1 + \beta)R_{\mathrm{s}} \tag{4}$$

where $W$ is the half-power bandwidth centered at 0 and $R_{\mathrm{s}}$ is the symbol rate [14][22].

## 2.3 Signal demodulation

To receive a passband signal and convert it back to data bits, essentially the same process described by the transmitter block diagram is conducted in reverse, but with some additional steps, as shown in Fig. 3.

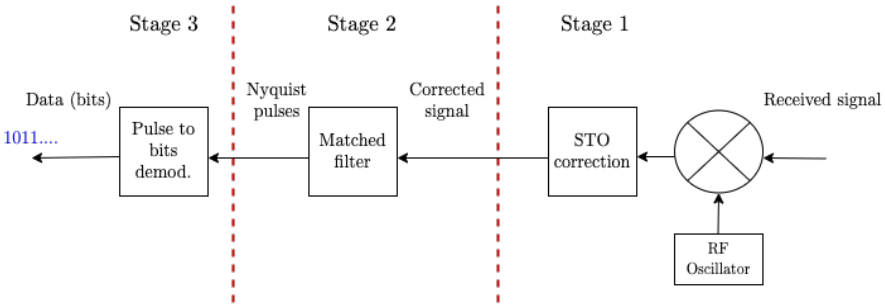



**Figure 3:** Block diagram of a receiver

First, it is important that the transmitter and receiver has the exact same perception of frequency, otherwise a symbol timing offset (STO) will occur. In the presence of a frequency offset, the downsampling of the received wave will continuously drift from the ideal peaks, resulting in a rotating constellation. Moreover, if there is a constant phase error between the transmitter and receiver, the constellation will be tilted [14]. These two factors needs to be accounted for to correctly demodulate a signal, shown as stage 1 in Fig. 3, after the passband signal has been downconverted to a baseband signal. One solution to mitigate the frequency offset is to connect the communicating devices to a common frequency reference. There are also more advanced software-based solutions, but they are beyond the scope of this report. A constant phase shift can be mitigated by having the transmitter and receiver agree upon a pre-programmed common signal and then calculate the phase difference.

Second, the receiver has to locate the beginning of a message to correctly decode it. In communication systems, framing bits, or pilots, that follow a deterministic pattern are often used for this purpose [14]. However, in a development stage of a communication system where the message is known in prior, a cross-correlation calculation can be done to find the best matched sequence of the signal.

Once the STO estimation is done and the signal is extracted, the baseband signal is convolved with a matched filter according to (2) converting the pulse train (the baseband signal) into Nyquist pulses, shown as stage 2 in Fig 3. Lastly, the signal is downsampled and normalized [14]. The normalization is necessary as the convolution multiplies the amplitude of the signal by its signal energy. The output from this step is symbols with added noise, $n$, from the channel and can be described by:

$$z_k = a_k + n \tag{5}$$

where $z_k$ is the $k$:th symbol in the received modulated signal. These symbols are then compared to the original symbol alphabet, and identified according to minimum-distance sequence detection [14]. This can be written as:

$$\{\hat{a}_k\} = \arg \min_{\{a_k\} \in \mathcal{A}^L} \sum_{k=0}^{\infty} \left| z_k - \sum_{l=0}^{L-1} a_l \right|^2 \tag{6}$$

where $\hat{a}_k$ is the decision based on $z_k$. Lastly, the mapper in Fig. 3 stage 3 converts the decisions into data bits.

## 2.4 Transmission performance

The precision of the decisions, i.e., how many times $\hat{a}_k$ matches with the true $a_k$-value, is important to analyze when looking at the performance of the communication system. EVM is a metric that is used to assess distortion into the received signal after being subjected to the limitations and noise of the channel [23]. According to [24], EVM is defined as the root-mean-squared (RMS) value between the received symbols $\hat{a}_k$ and ideal transmitted symbols $a_k$. The general expression for EVM can be written as:

$$\text{EVM} = \sqrt{\frac{\frac{1}{N}\sum_{k=1}^{N} |\hat{a}_k - a_k|^2}{P_0}} \tag{7}$$

where $N$ is the number of symbols and $P_0$ is the average power of the modulated symbols. The calculated EVM in (7) is often given as percentage by multiplying the value by 100 %. It can also be interesting to calculate the EVM for an individual symbol, instead of the RMS value of the whole signal in 7. Then, one can find the received symbol with the largest deviation from its ideal position in the constellation diagram, called the maximum EVM.

# 3
# Methodology

This chapter describes the methods used to communicate with the USRPs as well as the development of custom signal processing code that enabled the devices to transmit and receive user-defined signals. It also outlines the experimental procedures carried out to ensure that the theoretical results and simulations aligned with practical observations.

## 3.1 Software development for signal processing

To develop UEs suitable for use in a communications system, the USRPs had to be configured and programmed with software. MATLAB was chosen as it provides a high-level interface for the USRP through the Communications Toolbox and its Support Package for USRP Radio, along with an extensive library of signal processing functions. Moreover, this made it possible to exchange signal data with the USRP just through IQ-samples.

### 3.1.1 Configuring the USRP as a transmitter

In order to configure the USRP as a transmitter, a connection to a host computer was established using USB 3.0, after which the device was initialized through a `comm.SDRuTransmitter` system object in MATLAB. During initialization, this object provided the USRP with necessary settings, such as the frequency of the carrier wave, sampling rate of the digital-to-analog converter (DAC), whether to lock the device to an external frequency or timing reference and the gain for the RF front end. The code used for the setup is provided in Appendix A. With the USRP configured, transmitting data was simply a matter of calling the system object and providing a complex valued vector of IQ-samples as an argument.

The data used for transmitting modulated signals was generated randomly using the modulation order $M = 16$. This resulted in a vector containing $N$ random numbers $n \in [0, 15]$. Each element was then mapped to a symbol in a 16-QAM-constellation where the average power was normalized to one watt. Pulse shaping was then performed to merge the symbols into a single signal. The pulse used was a root-raised cosine FIR filter with roll-off factor $\beta$ and samples per symbol $n_{\text{SPS}}$, total number of spanning symbols $n_{\text{span}}$. These parameters were selected experimentally, by testing the system with different settings and choosing a combination that resulted in a fast but reliable transmission, while also satisfying the criterion for successful transmis-

sions in Section 1.2. The filter was applied using convolution after inserting $n_{\text{SPS}}$ zeros around each sample in order to match the filter settings. Finally, the signal was stripped of the tails produced by the convolution, ensuring the correct amount of samples when downsampling at the receiver.

### 3.1.2 Configuring the USRP as a receiver

Configuring the USRP as a receiver included mostly the same steps as for the transmitter, but using the `comm.SDRuReciever` system object instead. The code used for the setup is provided in Appendix A. IQ-samples of any received signal could then be retrieved simply by calling the system object. The `SDRuReciever` was set up to collect a large quantity of samples per function call, at least double that of the expected signal, to ensure that no data was missed. To extract a complete message from the received samples, an STO estimation was performed by cross correlating the received signal with the transmitted one and removing any excess samples. Moreover, phase shifts induced in the channel had to be accounted for by comparing the angle of the sent and received signals, then phase shifting all received IQ-samples by their difference.

The resulting vector was then convolved with a matched RRC filter, as per the transmitter, and downsampled by a factor of $n_{\text{SPS}}$ starting from sample $n_{\text{SPS}} + 1$. This produced a vector of QAM-symbols which was normalized to have an average power of one watt and visualized in a constellation diagram. This illustrated the received symbols and together with the EVM measurement, was used to assess the quality of the transmission. Finally, the symbols were demapped using the unit average power as a reference and compared to the original data to evaluate the success rate of the transmission.

## 3.2 Simulations

To test the functionality of the written scripts before utilizing them in conjunction with the USRPs, simulations were conducted to determine initial values for filter parameters and overall signal design. These simulations involved generating data, followed by modulating, upsampling and filtering it using the developed scripts. The resulting signal was then padded with a random amount of zeros in the beginning and end to emulate STO and passed through a receiver chain consisting of STO correction, filtering, downsampling and demapping. Once verified that the data was transmitted and received perfectly in the simulated environment, subsequent testing moved on to a physical setup.

## 3.3 Measurements

Initial measurements were conducted by transmitting signals from the USRP and examining them using an oscilloscope. The devices were connected via an SMA cable, as illustrated in Fig. 4, thus matching the impedance throughout the system.

The oscilloscope used was a two channel Rohde & Schwarz RTB2000 with a sampling rate of 1.25 Gsamples/s and a bandwidth of 70 MHz. The center frequency of the signals was set to 100 MHz and the USRP initially had a 30 dB attenuator attached to make sure the power sent to the oscilloscope was within limitations.

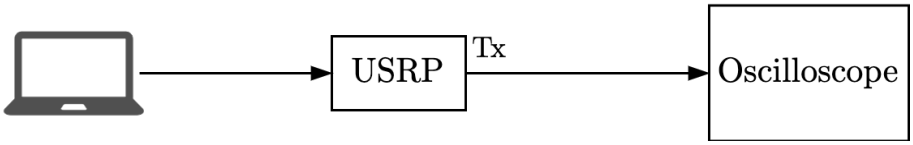



**Figure 4:** Setup for wired transmission from USRP to oscilloscope.

By looking at the signals in the time-domain, it was possible to observe how various system settings affected the passband signal, such as how changing the gain altered the transmitted power. The measurements were also able to provide some insights as to whether the USRP transmitted subsequent messages continuously or in partitions with pauses in between.

After ensuring that the transmitted waves aligned with expectations of how the data should be modulated, the testing proceeded to utilize the digital receiver constructed in section 3.1.2. For this, the USRPs were directly connected to each other via SMA cable, as illustrated in Fig. 5. Both devices were set up separately, one as transmitter and one as receiver, using two different instances of MATLAB running on the same host computer. The transmitter had a 10 dB attenuator attached to keep the signal power within limitations, and the receiver had a DC-blocker attached to ensure that no DC-offsets were present when demodulating the signals.

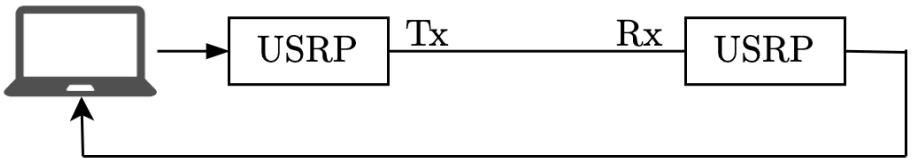



**Figure 5:** Setup for wired transmission from USRP to USRP.

To ensure the USRPs had the same frequency reference, they were both connected to the same external 10 MHz reference clock, with BNC-SMA cables, taken from an external function generator. With the connected and synchronized USRPs, transmission of signals with simple modulation schemes were conducted, beginning with 2-QAM. Each transmissions consisted of 1000 symbols, sent multiple times to ensure that the receiver captured all the data. A constellation diagram was plotted on the receiving side, displaying what data reached the receiver. Here, tuning the filter properties — including the $n_{\text{SPS}}$, $n_{\text{span}}$, and $\beta$ — as well as the receiver and transmitter gain, was necessary to achieve as clear and precise a constellation as possible, i.e., low EVM. Further transmissions involved increasing the modulation order to test the capabilities of the communication system in terms of the number of information bits per symbol, and by extension, the achievable bit rate.

When transmission over cable gave satisfactory results, the channel was changed to transmit over the air with antennas, as shown in Fig. 6. The wireless transmissions were brought out in the same manner as the cabled tests, starting with a low modulation order and gradually increasing it but at a center frequency of 2.35 GHz. The results from cable transmissions were used as a base for the filter design and gain settings but tuned to satisfy the faster amplitude decay and noise influence induced in wireless transmissions.

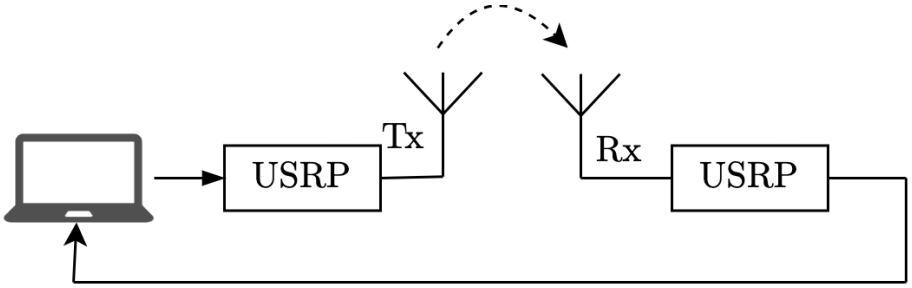



**Figure 6:** Setup for wireless transmission from USRP to USRP.

### 3.3.1 Signal bandwidth and symbol rate

To find an answer on what is the maximum signal bandwidth and symbol rate, two steps were brought out. The first step involved a theoretical assessment to identify which parameters to focus on. The second step consisted of practical measurements using the physical devices. The overall approach to find the maximum symbol rate was to investigate and measure the maximum signal bandwidth, and calculate the symbol rate from the measured signal bandwidth.

#### 3.3.1.1 Theoretical assessment

The theoretical assessment began with analyzing the datasheet from Ettus Research of the USRP B205mini-i [25]. The value of instantaneous bandwidth was noted. This was interpreted as the passband bandwidth, and used as a reference when performing measurements of the real bandwidth.

A relationship between the baseband signals bandwidth and symbol rate was investigated. It was found that the sample rate of the USRP, $R_\mathrm{s}$ is equal to the master clock rate, $R_\mathrm{c}$. To find the symbol rate, the sample rate is then divided by the number of samples per symbol. However, the interpolation factor, $f_i$, inserts samples between every data sample, lowering the resulting symbol rate. Thus, the relationship between the symbol rate and sample rate is:

$$R_{\mathrm{sym,ideal}} = \frac{R_C}{n_{\mathrm{SPS}} f_i} \tag{8}$$

This equation was used when calculating the ideal value of $R_{\text{sym}}$. Lastly, the Nyquist criteria for ISI-free transmission with an RRC pulse was used, according to (4), and the following equation obtained:

$$W = R_{\text{s}}(1 + \beta) \tag{9}$$

where $W$ is the baseband bandwith centerd at 0 and $\beta$ is the roll-off factor of the rrc pulse. The symbol rate is then calculated as:

$$R_{\text{sym,meas}} = \frac{W}{(1 + \beta)f_i} \tag{10}$$

and used to obtain a measured value of the symbol rate by measuring $W$.

#### 3.3.1.2 Measurements

To measure the signal bandwidth, thus the value of $R_{\text{sym,meas}}$, the USRPs were used in the wireless transmission configuration as in Fig. 6. An external pulse per second (PPS) source regarding the timing reference, and the internal 10 MHz clock regarding the frequency reference was used. The filter was tuned to utilize as large bandwidth as possible by changing the roll-off factor $\beta$ and samples per symbol $n_{\text{SPS}}$. Furthermore, the sampling rate was changed by adjusting the $R_{\text{c}}$. At the receiving side, the 99.9% of the occupied bandwidth (OBW) of the received signal was calculated in MATLAB. The value of $R_{\text{sym,meas}}$ was then obtained by inserting OBW and $\beta$ into (10).

To find out how different parameters affect signal bandwidth, a series of tests were conducted. First, the roll-off factor $\beta$, and $R_{\text{c}}$ were fixed, while the $n_{\text{SPS}}$ was varied incrementally from 2 to 10. The $n_{\text{SPS}}$ value that produced the highest bandwidth while still meeting the transmission criteria as defined in Section 2.2 was selected for the next test. Then, $\beta$ was varied from 0.2 to 1 in order to determine its effect on bandwidth, again ensuring that all results satisfied the same criteria. Finally, the $R_{\text{c}}$, i.e., the sampling rate was varied from its lowest supported value of 5 MHz up to 61.44 MHz to observe its impact on the resulting bandwidth.

### 3.3.2 Signal duration and message length

To find the maximum signal length and if the on board memory was a restricting factor, two tests were brought out. For these tests, a 1-PPS reference was used instead of a 10 MHz reference as previously. Firstly, two wirelesly communicating USRPs were used and the amount of symbols per transmission and was increased sequentially from 1000 to 9000 symbols. The receiving USRP recorded the success rate and EVM to give an insight on how a longer message impacted the performance of the system.

Secondly, two USRPs were connected to the same host computer, transmitting to an oscilloscope on two separate channels, as shown in Fig. 7. The transmitter objects were called directly after each other in MATLAB to see whether there was

an overlap in time between the transmitted signals at the oscilloscope. Such an overlap could indicate the presence of internal buffering in the USRPs and provide insight into potential memory limitations. Moreover, the message length was varied to investigate whether this overlap changed and if there was a case when both USRPs transmitted simultaneously. The master clock rate was also changed to see if it would have an impact on these factors.

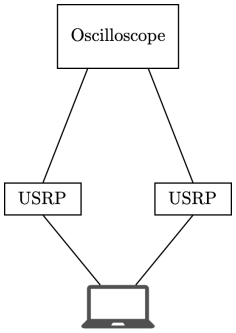



**Figure 7:** USRP setup for message overlap measurements.

### 3.3.3 Synchronization

In order to synchronize the start of transmissions from two different USRPs, they had to agree on what time to start transmitting. As such, the 10 MHz frequency reference was replaced by a 1 PPS timing reference, causing their internal clocks to move at the same speed. It was also crucial that both USRPs were able to operate simultaneously, i.e. in parallel. For this purpose, the Parallel Computing Toolbox in MATLAB was used, more specifically, the `parfeval` function which can assign entire function calls to separate MATLAB workers. As such, initialization, utilization and deactivation of a USRP was grouped into a single function call, `transmit`. Both USRPs were connected to the same computer and configured in the same way, but through separate function calls. Most notably, the setup of the `comm.SDRuTransmitter` object differed from previous configurations in that `'EnableTimeTrigger'` was set to `'True'`. After initialization, `TimeTrigger` was then set in order to determine at what time to trigger the transmission. The time specified was the current time of the internal USRP clock, obtained using the `getRadioTime` method, plus a few seconds delay.

In order to make sure that both processes called `getRadioTime` simultaneously, even though they operated on different workers and interfaced with different hardware, a `tic` timer object was passed as a argument to `transmit` and only when `toc(timer)` reached a certain delay was the time requested. The resulting signals were observed through an oscilloscope using the same measurement setup as in Fig. 7, in which each transmitter port was connected to a different channel. The metric recorded was the time between the start of different messages.

# 4
# Results

The purpose of this project was to design a flexible UE using a USRP that could support experiments using the D-MIMO testbed at Chalmers. The maximum bandwidth was measured to 41.24 MHz with a symbol rate of 10.31 Msym/s and the memory was found to be streamed and therefore not limiting the signal duration. Moreover the USRPs could be synchronized within a few microseconds at best. This section will essentially be divided into two subsections, one covering the results from the setup stage and the other covering more detailed answers to the research questions.

## 4.1 Parameter values

This section will cover the results from the setup stage, i.e., the simulation and physical testing. The results from these measurements, regarding setting parameter values, is presented are Tab. 1.

**Table 1:** Parameter values from initial testings

| **Parameter** | **Value** |
|---|---|
| Roll-off factor, $\beta$ | 0.4 |
| Samples per symbol, $n_{\text{SPS}}$ | 4 |
| Filter span, $n_{\text{span}}$ | 24 |
| Interpolation factor, $f_i$ | 2 |
| Master clock rate, $R_c$ | 32 MHz |
| **Wired Link** | |
| Transmitting gain | 44 dB |
| Receiving gain | 32 dB |
| Center frequency | 100 MHz |
| **Wireless Link** | |
| Transmitting gain | 44 dB |
| Receiving gain | 52 dB |
| Center frequency | 2.35 GHz |

Note that most the values in Tab. 1 are used in the coming sections. If a parameter value is changed during an experiment, this will be mentioned.

## 4.2 Measured signal bandwidth and symbol rate

The maximum signal bandwidth and symbol rate was measured to be 41.24 MHz and 10.31 Msym/s respectively when setting the sampling rate, $R_{\text{c}}$, to 61.44 MHz, $\beta$ to 1 and $n_{\text{SPS}}$ to 2 samples per symbol. Note that the values on the other parameters presented in Section 4.1 are kept constant. The bandwidth specified in the Ettus Research datasheet was found to be 56 MHz [25].

The different measurements to obtain values for the parameters that maximizes the bandwidth, and the corresponding symbol rate, are presented below. All of the presented values are the mean value of 3 independent measurements, where the transmitted message consisted of 1000 symbols. The first presented parameter is $n_{\text{SPS}}$, where $\beta$ and $R_{\text{c}}$ are kept constant to 0.4 and 32 MHz.

**Table 2:** Impact of $n_{\text{SPS}}$ on bandwidth and symbol rate with $\beta = 0.4$ and $R_{\text{c}} = 32$ MHz.

| $n_{\text{SPS}}$ | OBW [MHz] | Success rate | EVM | maxEVM | $R_{sym,meas}$ [Msym/s] |
|---|---|---|---|---|---|
| 2 | 17.21 | 100 % | 5.19 % | 12.29 % | 6.15 |
| 4 | 8.61 | 100 % | 7.72 % | 19.48 % | 3.08 |
| 6 | 5.74 | 100 % | 8.55 % | 20.93 % | 2.05 |
| 10 | 3.45 | 100 % | 7.08 % | 18.63 % | 1.23 |

From Tab. 2 it can be seen that $n_{\text{SPS}} = 2$ yields the highest bandwidth at 17.21 MHz with an acceptable EVM. The maximum symbol rate is $R_{sym,meas} = 6.15$ Msym/s, also when $n_{\text{SPS}} = 2$. The corresponding ideal value given from (8) gives $R_{\text{sym,ideal}} = 8$ Msym/s.

The second parameter presented is $\beta$, where $n_{\text{SPS}}$ and $R_{\text{c}}$ are kept constant to 2 and 32 MHz respectively.

**Table 3:** Impact of $\beta$ on bandwidth and symbol rate with $n_{\text{SPS}} = 2$ and $R_{\text{c}} = 32$ MHz.

| $\beta$ | OBW [MHz] | Success rate | EVM | maxEVM | $R_{sym,meas}$ [Msym/s] |
|---|---|---|---|---|---|
| 0.2 | 16.46 | 89.83 % | 29.87 % | 127.14 % | 6.86 |
| 0.4 | 17.21 | 100 % | 5.19 % | 12.29 % | 6.15 |
| 0.6 | 18.50 | 100 % | 8.34 % | 19.65 % | 5.78 |
| 0.8 | 19.86 | 100 % | 5.50 % | 12.63 % | 5.52 |
| 1 | 21.51 | 99.8 % | 9.79 % | 176.77 % | 5.38 |

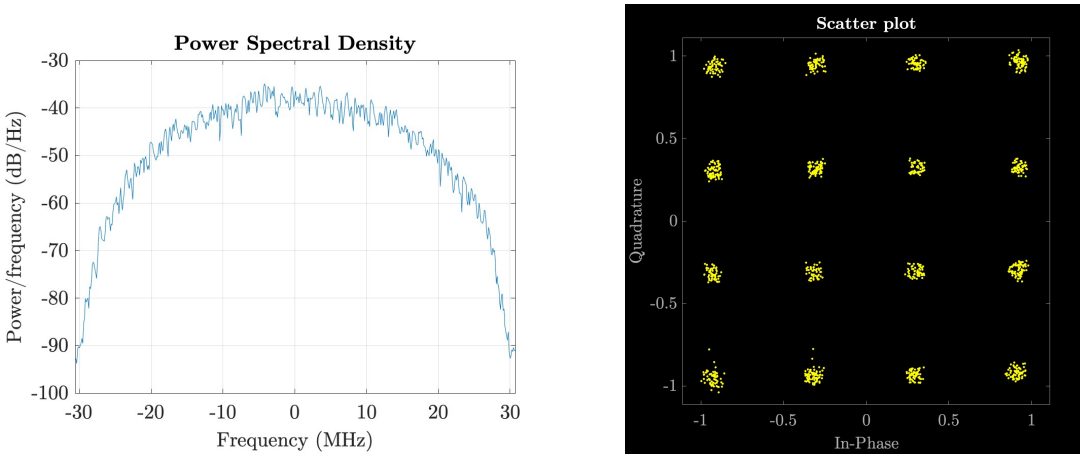
From Tab. 3, it can be seen that $\beta = 1$ yields the highest bandwidth at 21.51 MHz with an acceptable EVM. The maximum symbol rate, with an acceptable EVM, is obtained when $\beta = 0.4$ to $R_{sym,meas} = 6.15$ Msym/s. The corresponding ideal symbol rate is $R_{\text{sym,ideal}} = 8$ Msym/s.

The third and last presented parameter is $R_{\text{c}}$, where $n_{\text{SPS}}$ and $\beta$ are kept constant to 2 and 1 respectively.

**Table 4:** Impact of $R_c$ on bandwidth and symbol rate with $\beta = 1$ and $n_{SPS} = 2$.

| $R_c$ [MHz] | OBW [MHz] | Success rate | EVM | maxEVM | $R_{sym,meas}$ [Msym/s] |
|---|---|---|---|---|---|
| 5 | 3.36 | 99.80 % | 14.90 % | 179.55 % | 0.84 |
| 20 | 13.43 | 99.80 % | 9.22 % | 175.12 % | 3.36 |
| 32 | 21.51 | 99.80 % | 9.79 % | 176.77 % | 5.38 |
| 45 | 30.25 | 99.80 % | 8.96 % | 175.44 % | 7.56 |
| 61.44 | 41.24 | 99.80 % | 10.21 % | 177.19 % | 10.31 |

Tab. 4 reads that the highest $R_c = 61.44$ MHz yields the highest bandwidth of 41.24 MHz. The maximum symbol rate is $R_{sym,meas} = 10.31$ Msym/s, also when $R_c = 61.44$ MHz. The corresponding ideal symbol rate is $R_{sym,ideal} = 15.36$ Msym/s. The frequency spectrum and constellation diagram from the measurements giving the presented maximum bandwidth and symbol rate are shown Fig. 8a.




**(a)** Spectrum **(b)** Constellation diagram

**Figure 8:** Received signal with maximum signal bandwidth at $R_c = 61.44$ MHz, $\beta = 1$, $n_{SPS} = 2$.

## 4.3 Results on signal duration and message length

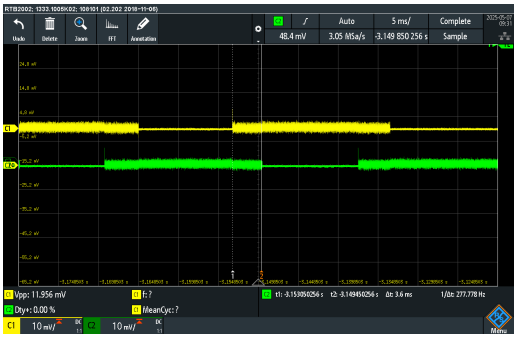
The results when investigating the maximum signal duration show that there is no definite limit. When reading the documentation of the used toolboxes, MATLAB configures the USRP to stream its memory directly to the computer and therefore the USRP is not limited by its on-board memory. This is also reflected in the results, that neither the precision nor fail-rate is drastically affected by an increase in the message length, see Tab. 5.

**Table 5:** Transmission success rate when increasing the number of symbols.

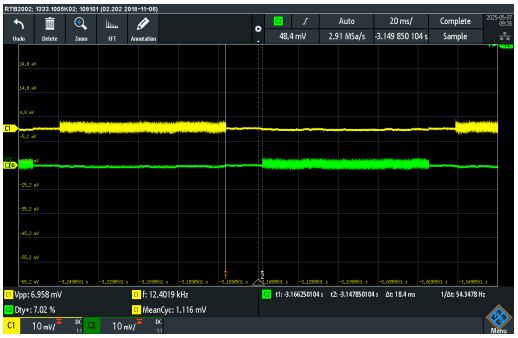
| Nr. symbols | Success rate | EVM | maxEVM |
|---|---|---|---|
| 1000 | 100 % | 8.01 % | 18.43 % |
| 2000 | 100 % | 9.44 % | 23.65 % |
| 3000 | 100 % | 10.35 % | 26.93 % |
| 4000 | 100 % | 7.80 % | 20.23 % |
| 5000 | 100 % | 8.12 % | 22.71 % |
| 6000 | 100 % | 6.08 % | 19.20 % |
| 7000 | 100 % | 10.11 % | 29.50 % |
| 8000 | 100 % | 10.19 % | 28.83 % |
| 9000 | 100 % | 12.52 % | 34.75 % |

As can be seen in Tab. 5, the EVM does increase as message length increases, but not in a manner that would be expected if the on-board memory is a limiting factor.

When connecting both the USRPs to different channels on the oscilloscope and sending data from the same host computer, some overlap of the signals could be seen, see Fig. 9. For shorter messages, the pause between messages from the same USRP also shortened, eventually resulting in both USRPs sending data continuously and simultaneously. The threshold for when the overlap could be seen changed with the master clock rate setting, as illustrated in Fig. 10 based on measurement data from Appendix B.



**(a)** Positive overlap



**(b)** Negative overlap

**Figure 9:** Message overlap from transmission with two USRPs to an oscilloscope.

Fig. 10 shows that for the highest master clock rate, the messages are never sent in a continuous stream, indicated by the time between messages never reaching 0 ms. Furthermore, a lower master clock rate enables a larger message to be sent in a continuous stream.

## 4.4 Synchronization results

When attempting to send messages simultaneously using the `TriggerTime` property and parallel processing as described in Section 3.3.3, the messages obtained commonly look as in Fig. 11. The parallel processes do indeed seem synchronized when

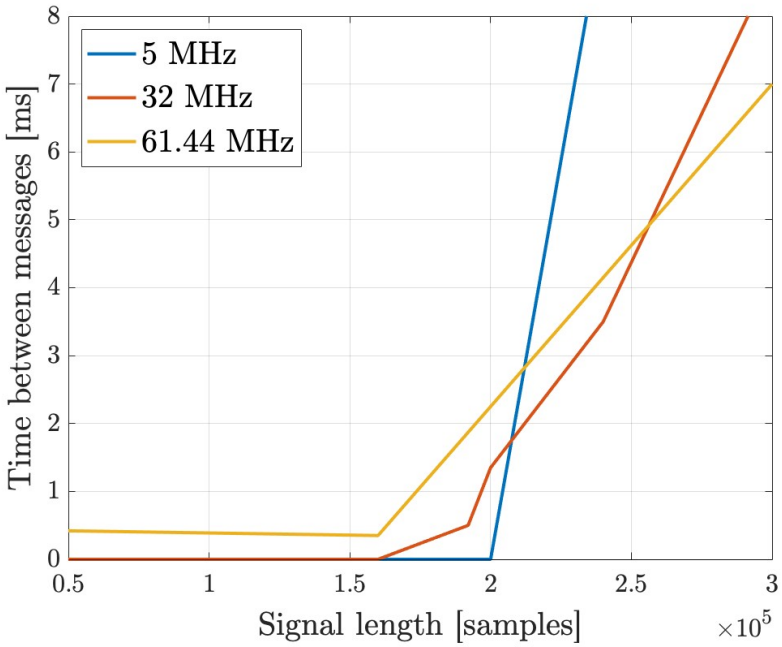
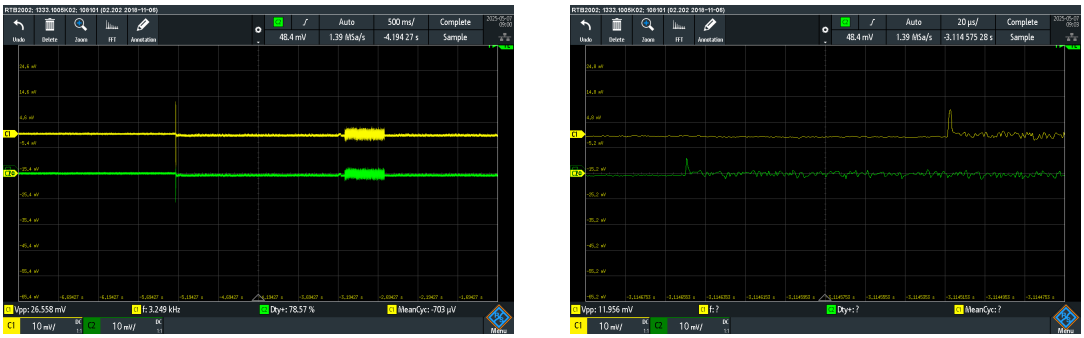



**Figure 10:** Pause time between transmissions for different sized messages and master clock rates. A time of 0 ms indicates a continuous stream of data.

observed over a period of hundreds of milliseconds, such as in Fig. 11a, but when zooming in, such as in Fig. 11b, a difference becomes apparent.



**(a)** 500 ms per division **(b)** 20 $\mu$s per division

**Figure 11:** Oscilloscope pictures of two identical signals transmitted from different USRPs set to operate simultaneously, shown at different time scales.

This delay, which was present throughout repeated experiments, varies a lot between attempts. Sometimes the messages arrive hundreds of milliseconds apart, other times within a few microseconds. A histogram containing the probabilities for delays of a certain magnitude occurring can be observed in Fig. 12.

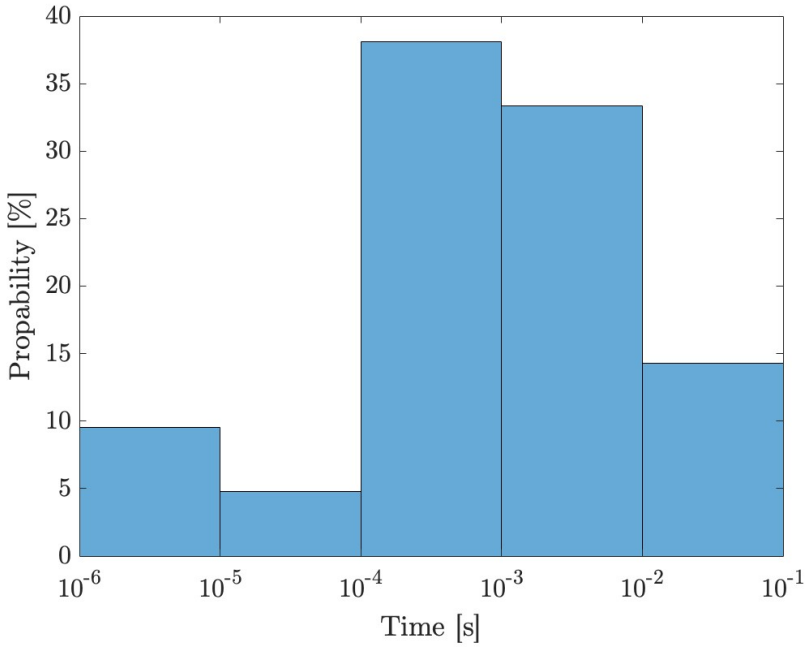



**Figure 12:** Histogram showing the likelihood of delays between synchronized messages falling within a certain magnitude.

# 5
# Discussion

This chapter will discuss the findings from analyzing the built communication system using two USRPs. Possible reasons for failed tests and suggestions on further experiments will also be presented.

## 5.1 Analysis of signal bandwidth and symbol rate

The result from Section 4.2 showed that the measured maximum symbol rate was 10.31 Msym/s when the signal bandwidth was measured to 41.24 MHz and $\beta = 1$. This part of the discussion will discuss and and clarify the approach of investigating the maximum signal bandwidth to find the maximum symbol rate. Furthermore, an analysis comparing the ideal and measured values on the symbol rate will be presented. Lastly, some interesting observations from the measurements, where the parameters $n_{\text{sps}}$ and $\beta$ were varied, will be presented.

Firstly, it should be clarified that the obtained maximum bandwidth of the signal is not the same as the symbol rate. The approach of finding the maximum symbol rate by looking for the maximum signal bandwidth could indeed be questioned. It was concluded that $\beta = 1$ gave the highest bandwidth, but this does not imply the highest symbol rate since $\beta$ is the excess of bandwidth in the signal. This can also be seen in Tab. 4 where the maximum signal bandwidth was obtained when $\beta = 1$. However, the maximum symbol rate using (10) is obtained when $\beta = 0.4$. Therefore, when investigating the last parameter in Tab. 4, $\beta$ should have been 0.4 and not 1. Thus the maximum value of the measured symbol rate was likely not reached in this project.

Moving on, comparing the measured maximum signal bandwidth, 41.24 MHz, to the bandwidth stated by Ettus Research in the datasheet to the USRP, 56 MHz, one can see that the measurements did not reach the maximum signal bandwidth. Although, as stated before, this would not directly imply to the highest symbol rate, it would be interesting to investigate if it is possible to reach the 56 MHz presented in the datasheet. Therefore, other measurements could have been conducted to verify the obtained value, for instance, parameters as $n_{\text{span}}$. More measurements could also have been done, testing the same parameter multiple times with different values. It would have been relevant to know if the USRP could transmit a signal with a bandwidth close to 56 MHz with an acceptable EVM, i.e. to verify that the maximum bandwidth given in the datasheet can be used with an acceptable performance.

Furthermore, it can be seen that the maximum measured symbol rate is smaller than the corresponding calculated ideal value presented in section 4.2, where $R_{sym,meas}$ = 10.31 Msym/s and $R_{sym,ideal}$ = 15.36 Msym/s. The fact that the symbol rate in reality is lower than the ideal case is not surprising. The limited bandwidth of the channel, as stated in the theory, gives rise to distortion in the signal. Depending on the magnitude of the distortion, this might have affected the bandwidth of the transmitted signal and thus the symbol rate. However, since the limitations of the channel was not investigated, this could be an interesting topic for further study. Moreover, the difference between the measured and ideal symbol rate in this case is to be considered quite big. One possible reason to this could be that the USRPs actual sample rate is lower than expected, i.e. the configured sample rate in MATLAB does not match with the speed of the DAC.

Lastly, some interesting observations on the parameter $\beta$ and $n_{\text{sps}}$ were made when analyzing the data in Tab. 3 and Tab. 2 respectively. It can be seen that the lowest value of $\beta = 0.2$ gave the worst success rate and EVM. One explanation to this could be that the Nyquist criteria fails at $\beta = 0.2$ and ISI occurs. In addition, it seems that $\beta = 1$, besides giving the highest bandwidth, results in a drastic increase of the maximum EVM and did always give a success rate of 99.80 %. This was not expected as the EVM shows that the performance is inside the acceptable range. Looking at the the constellation diagram in Fig. 8b one can also see that all points seem to be placed in the right decision area. Hence, the odd pattern that $\beta = 1$ always resulted in a success rate of 99.80 % might be explained by an undetected error in the code.

Regarding $n_{\text{sps}}$, it is not clear why the value of EVM decreases with an increasing $n_{\text{sps}}$. It was thought that a high $n_{\text{sps}}$ would provide more redundancy to each symbol in the signal and therefore increase the performance of EVM. This could, however, potentially be explained by the fact that a symbol containing many samples is more susceptible to channel-induced errors.

## 5.2 Signal duration and message length

Prior to testing, it was hypothesized that the USRP had an on board memory which it filled with samples and, upon command, either transmitted to the computer or forwarded to the RF front end, depending on the context. However, while reviewing the MATLAB documentation and as conducting experiments, this assumption was revised. It was concluded that samples are transferred as a continuous real-time stream between the USRP and the host computer. This implies that there is no limitation on message length, at least within the scope of these tests and as long as the host computer can sustain the required data rate. If the USRP's memory was a limiting factor, it would be expected that only part of a message would be transmitted or that the entire message would be corrupted, neither of which was observed during these experiments.

If there is a case where memory streaming is unwanted, Python is expected to support configuration of such behavior, allowing for fixed-size buffers on the USRP, which are filled and then transferred to the host. However, this functionality was not explored in this project and requires further investigation.

Another interesting factor tested was the overlap or pause time between transmissions. Message overlap was initially observed during a separate experiment, which led to the hypothesis that such overlap could reflect a fixed buffer size within the USRP. This could have been interesting if the overlap showed to hold a constant number of samples, which is not the case. The overlap changes with both message length and master clock rate which indicates on a restriction elsewhere. However, one finding was that for a shorter message, both the transmission streams seemed contiguous without a pause time in between messages. This can be interesting to maximize the rate of information transfer over time, that several shorter messages are transmitted faster than one long. How this overlapping is connected to on board memory is nevertheless still unproven.

## 5.3 Discussion on synchronous transmissions

The method of synchronization utilized in this project provided a workaround for several issues related to the USRP hardware as well as the software interface, but ended up relying heavily on digital timers and non sequential processing. The result was a system capable of transmitting messages simultaneously, but unable to achieve synchronization with sub-microsecond precision. Nevertheless, it appears plausible that nanosecond-level synchronization could be achieved with further modifications to the system.

Firstly, one might consider keeping the current MATLAB interface but replacing the hardware. The B205mini-i lacks several substantial features, most notably a second reference port. This makes true synchronization across multiple devices, i.e. a common perception of both time and frequency, impossible. Utilizing dual reference ports would allow for synchronization of entire messages, not just matching the moment of initialization. Switching out the hardware would also allow for better utilization of the software, as many functions in the USRP Hardware Driver (UHD), upon which the MATLAB interface builds, are not supported by the B205mini-i. This includes features from the `MultiUSRP` class, such as configuring multiple USRPs simultaneously or resetting the time across all devices. These features would eliminate the need for a MATLAB timer and reduce the amount of times needed to call `getRadioTime`. Thus, replacing the USRPs with a model featuring better hardware and more support within the software could provide an easy, though costly, solution to overcoming the millisecond-level synchronization barrier.

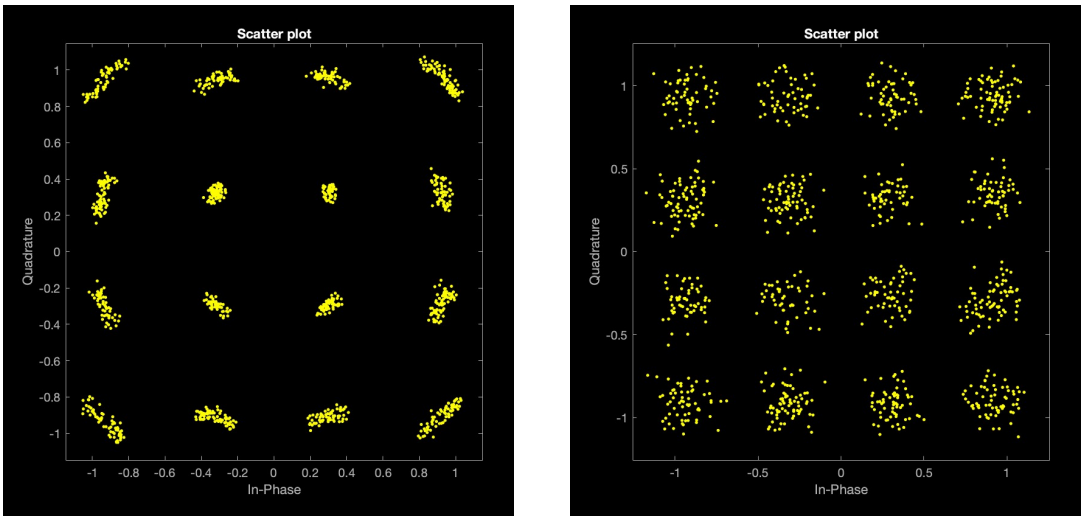
Secondly, working with the `comm.SDRuTransmitter` as well as `parfeval` in MATLAB does come with some caveats. If a user wants to transmit multiple signals by repeatedly setting a new `TriggerTime`, the resources of the entire `SDRuTransmitter`

must be released and then reconfigured, which slows the system down significantly. The `SDRuTransmitter` must also be configured within `parfeval` due to the function serializing all of its inputs, making external system objects unusable. And so even if the hardware supported synchronized configuration through a single object, as is the case with the `MultiUSRP` class, it could not be utilized using the current method of using two parallel processes. Switching from MATLAB to the UHD application programming interface and utilizing a programming language with more versatile multiprocessing capabilities, such as Python or C++, could prove faster and provide workaround for the separate thread configurations of devices.

Finally, using a high-level software interface might just simply be too slow for this application. The user is provided with very little control over the inner workings of the USRP and processing delays seem unpredictable. To gain better control, one might consider reprogramming the USRPs FPGA, forcing internal processing and transmission to occur using something like a general-purpose input trigger. An approach like this could potentially create compatibility issues with the UHD and fell outside the scope of this project, but would potentially achieve the most precise results.

## 5.4 Reliability issues

During experiments with the developed communication system, some reliability issues were noticed. More specifically, while transmitting a fixed sequence of symbols consecutively and repeatedly triggering the receive script, the results varied sometimes significantly. In most cases, a clear constellation diagram with low EVM was obtained, but occasionally the diagrams were skewed in a rotational- or spread out fashion, as depicted in Fig. 13.




**(a)** Circular tendencies **(b)** Spread out symbols

**Figure 13:** Received constellations from continuous transmission attempts.

The exact cause of this issue could not be isolated, however several possible explanations are proposed here. One reason could be a power related issue in the USRP. As they were both connected to one computer, they shared the same USB-C port with an USB-C to USB-A hub. This seemingly worked to power both the devices. However there could have been an occasional shortage of power when both USRP were used for transmitting and receiving respectively, resulting in a loss of samples and therefore misalignment when downsampling. This issue could be eliminated by using two different USB controllers to power the devices, such as two host computers if those resources are available. Unfortunately, external power supplies are not available for the USRP B205mini-i [26].

Another reason for the occasional skewed constellations could be related to the memory streaming. If the host computer becomes busy with another prioritized task, samples could be lost. There are flags for giving indications on these issues in MATLAB called under-run and over-run. This could have been used to further consolidate a skewed transmission along with the constellation diagram. However, as these are just flags, a retransmission is still necessary to obtain a more concise constellation. This could be used in a deployment scenario to make the transmitter resend the data, but as this was outside the scope of this project, these flags were not used.

# 6
# Conclusion

In this project, a flexible UE using a USRP B205mini-i was developed for wireless communications experiments at a D-MIMO testbed at Chalmers. The system was successfully able to transmit and receive data with 16-QAM and well under the EVM threshold of 12.5 %. The results showed a maximum signal bandwidth of 41.24 MHz with a symbol rate of 10.31 Msym/s and that the on-board memory did not restrict the signal duration as the memory was streamed to the host computer. Moreover, it was possible to transmit messages from two different USRPs within microseconds of each other, with higher precision seemingly within reach given some modifications to the system. In conclusion it seems feasible to integrate the USRPs with the D-MIMO testbed, but depending on the requirements put on synchronization and data rate, some modifications and further developments might be in order.

# References


[1] Ericsson, "Ericsson Mobility Report November 2024," 2025. [Online]. Available: https://www.ericsson.com/en/reports-and-papers/mobility-report/reports/november-2024, Accessed on: 2025-02-03.

[2] D. Lopez-Perez et al., "Enhanced intercell interference coordination challenges in heterogeneous networks," *IEEE Wireless Communications*, vol. 18, no. 3, pp. 22–30, Jun. 2011, doi: 10.1109/MWC.2011.5876497.

[3] D. Astely, P. Von Butovitsch, S. Faxér, and E. Larsson, "Meeting 5G network requirements with Massive MIMO," *Ericsson Technology Review*, vol. 2022, no. 1, pp. 2–11, Feb. 2022, doi: 10.23919/ETR.2022.9881229.

[4] R. A. Valenzuela, "Mobile communications,"*Access Science*, 2023, doi: 10.1036/1097-8542.428800.

[5] A. Nimr et al.,"Early results of 6G Radio Key Enablers," Hexa-X-II, Deliverable D4.3, Version 1.0, 2024. [online]. Available https://hexa-x-ii.eu/wp-content/uploads/2024/04/Hexa-X-II_D4_3_v1.0_final.pdf, Accessed on: 2025-02-23.

[6] S. Zhou, M. Zhao, X. Xu, J. Wang, and Y. Yao, "Distributed wireless communication system: a new architecture for future public wireless access," *IEEE Communications Magazine*, vol. 41, no. 3, pp. 108–113, Mar. 2003, doi: 10.1109/MCOM.2003.1186553.

[7] H. Q. Ngo, A. Ashikhmin, H. Yang, E. G. Larsson, and T. L. Marzetta, "Cell-Free Massive MIMO Versus Small Cells," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1834–1850, Mar. 2017, doi: 10.1109/TWC.2017.2655515.

[8] L. Aabel et al., "A TDD Distributed MIMO Testbed Using a 1-bit Radio-Over-Fiber Fronthaul Architecture," *IEEE Transactions on Microwave Theory and Techniques*, vol. 72, no. 10, pp. 6140–6152, Oct. 2024, doi: 10.1109/TMTT.2024.3389151.

[9] R. Zitouni and L. George, "Output power analysis of a software defined radio device," *2016 IEEE Radio and Antenna Days of the Indian Ocean (RADIO)*, Oct. 2016, pp. 1–2. doi: 10.1109/RADIO.2016.7771996. Accessed on: 2025-05-08.

[10] *Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model*, ISO/IEC 7498-1:1994, ISO. Available: https://www.iso.org/standard/20269.html.

[11] NR; Base Station (BS) radio transmission and reception, 3GPP TS 38.104 V19.0.0, 3GPP, Sophia Antipolis, France, 2025

[12] A. B. Carlson, P. B. Crilly, and J. C. Rutledge, *Communication Systems*, 4th ed. New York, NY, USA: McGraw-Hill, 2002.

[13] M. S. Alencar and V. C. da Rocha Jr., *Communication Systems*. 2nd ed., Switzerland: Springer Cham, 2019. [Online] Available: https://doi.org/

10.1007/978-3-030-25462-9, Accessed on: 2025-04-10.

[14] J. R. Barry, E. A. Lee, and D. G. Messerschmitt, *Digital Communication*, 3rd ed., New York, NY, USA: Springer, 2004. [Online]. Available: https://doi.org/10.1007/978-1-4615-0227-2, Accessed on: 2025-04-09.

[15] Behrouz A. Forouzan, *Data Communications and Networking*, 4th ed., New York, NY, USA: McGraw-Hill, 2007.

[16] E. Agrell, J. Lassing, E. G. Strom, and T. Ottosson, "On the optimality of the binary reflected Gray code," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3170–3182, Dec. 2004, doi: 10.1109/TIT.2004.838367.

[17] S. A. White, "Amplitude modulator," *AccessScience*, Oct. 2019, doi: 10.1036/1097-8542.030830.

[18] Q. Zhang, P. Qu, S. Sun, E. Yu, and W. Yan, "Design of Full Digital Signal Generator and FIR Digital Filter System Based on FPGA," in *2022 41st Chinese Control Conference (CCC)*, Hefei, China, 2022, pp. 4501–4506. [Online]. doi: 10.23919/CCC55666.2022.9901637, Accessed on: 2025-04-09.

[19] K. Takahashi, K. Watanabe, T. Inoue, and T. Konishi, "Multifunctional Wavelength-Selective Switch for Nyquist Pulse Generation and Multiplexing," *IEEE Photonics Technology Letters*, vol. 30, no. 18, pp. 1641–1644, Sep. 2018, doi: 10.1109/LPT.2018.2866098.

[20] H. J. Helgert, "Pulse modulation," *AccessScience*, Jan. 2020, doi: 10.1036/1097-8542.556900.

[21] Q. Q. S. Ghusaini, T. Moazzeni, "Pulse Shaping Filters: Finding the Sweet Spot for SNR and Compactness" in *2023 24th International Arab Conference on Information Technology (ACIT)*, Ajman, United Arab Emirates, 2023 . [Online]. doi: 10.1109/ACIT58888.2023.10453791, Accessed on: 2025-04-07.

[22] B. P. Lathi and M. Wright, "Bandwidth requirements (communications)," *AccessScience,* Nov. 2019, doi: 10.1036/1097-8542.071900.

[23] *IEEE Recommended Practice for Estimating the Uncertainty in Error Vector Magnitude of Measured Digitally Modulated Signals for Wireless Communications,* IEEE Std 1765-2022, IEEE, 2022. doi: 10.1109/IEEESTD.2022.9942923.

[24] M. R. Kuppusamy, S. M. Bokhari, and B. M. Anjaneyalu, "Error Vector Magnitude (EVM)-Based Constellation Combiner for Cooperative Relay Network," *IEEE Communications Letters*, vol. 20, no. 2, pp. 304–307, Feb. 2016, doi: 10.1109/LCOMM.2015.2504374.

[25] *USRP B205mini-i Datasheet*, San Jose, CA, USA: Ettus Research, 2019 [Online]. Available: https://www.ettus.com/wp-content/uploads/2019/01/USRP_B200mini_Data_Sheet-1.pdf, Accessed on: 2025-05-12.

[26] Ettus Research, "B200/B210/B200mini/B205mini - FAQ," 2024. [Online]. Available: https://kb.ettus.com/B200/B210/B200mini/B205mini#FAQ, Accessed on: 2025-05-07.

# Appendix A

## MATLAB code

**Listing Appendix A.1:** getUSRP.m

```
% Attempts to find USRPs and configures them according to parameters
function [UsrpTx, UsrpRx, NumRadios] = getUSRP( CenterFreq, ...
                                                ClockSource, ...
                                                TimeSource, ...
                                                Gain, ...
                                                MasterClockRate, ...
                                                OversamplingFactor, ...
                                                ConnectionIndex )

% Function inputs:
%   <CenterFreq> ~ Center frequency of the desired passband
%   <ClockSource> ~ Source of the ocillator's frequency reference
%       'Internal' - Operates using internal clock
%       'External' - Operates using an external 10 MHz signal
%   <TimeSource> ~ Source of the ocillator's timing reference
%       'Internal' - Operates using internal time
%       'External' - Operates using an external 1 pps signal
%   <Gain> ~ Overall signal gain in the USRP data path
%   <MasterClockRate> ~ ADC and DAC clock rate
%   <OversamplingFactor> ~ Interpolation (Tx) and Decimation (Rx) factor
%   <ConnectionIndex> ~ Setting to determine which USRPs to connect to
%       0 ~ Configures both USRP1 & USRP2 for the script
%       1 ~ Configures USRP1 for the script
%       2 ~ Configures USRP2 for the script
%
% Function outputs:
%   <UsrpTx> ~ Struct containing all configured comm.SDRuTransmitters
%   <UsrpRx> ~ Struct containing all configured comm.SDRuRecievers
%   <NumRadios> ~ The number of configured USRPs


% Attempts to find USRPs
Radios = findsdru;

% In the case of wanting to connect to every USRP
if ConnectionIndex == 0

    % Defines the amount of radios as every avalible one
    NumRadios =  length(Radios);

    % Redefines the connection index to a vector containing all indicies
    ConnectionIndex = 1:NumRadios;

% In the case of connecting to just one USRP
else

    % Defines the amount of radios as just one
    NumRadios = 1;

end

% Creates empty cell arrays to fill with comm.SDRu objects
UsrpTx = cell(1,NumRadios);
```

```
UsrpRx = cell(1,NumRadios);

% Iterates through every radio connection
for i = ConnectionIndex

    % If the connection was unsucessful
    if ~contains('Success', Radios(i).Status)
        error(['Error connecting to the USRP: ' Radios(i).Status])

    % If the connection was sucessful
    else

        % Stores hardware information
        Platform = Radios(i).Platform;
        SerialNum = Radios(i).SerialNum;

        % Determines where to store the SRDu object
        if isscalar(ConnectionIndex); CellIndex = 1;

        else; CellIndex = i;

        end

        % Configures USRP as a transmitter
        txObj = comm.SDRuTransmitter(...
              'Platform',Platform, ...
              'SerialNum',SerialNum, ...
              'ChannelMapping',1, ...
              'CenterFrequency',CenterFreq, ...
              'ClockSource',ClockSource, ...
              'Gain',Gain, ...
              'MasterClockRate',MasterClockRate, ...
              'InterpolationFactor',OversamplingFactor, ...
              'PPSSource',TimeSource, ...
              'EnableTimeTrigger',0,...
              'TransportDataType','int8');

        % Confirmation message
        disp(['USRP', num2str(i), ' configured for TX']);

        UsrpTx{CellIndex} = txObj;

        % Configures USRP as a reciever
        rxObj = comm.SDRuReceiver(...
              'Platform',Platform, ...
              'SerialNum',SerialNum, ...
              'ChannelMapping',1, ...
              'CenterFrequency',CenterFreq, ...
              'ClockSource',ClockSource, ...
              'Gain',Gain, ...
              'MasterClockRate',MasterClockRate, ...
              'DecimationFactor',OversamplingFactor, ...
              'PPSSource',TimeSource, ...
              'TransportDataType','int8', ...
              'SamplesPerFrame',50000);

        % Confirmation message
        disp(['USRP', num2str(i), ' configured for RX']);

        UsrpRx{CellIndex} = rxObj;

    end

end

disp(['Successfully configured ', num2str(NumRadios),' USRPs'])
```

# Appendix B

## Overlap measurements

Data from message overlap measurements are presented in the tables below.

**Table 6:** Message overlap with MCR = 5, 32 and 61.44 MHz.

| MCR [MHz] | Samples | Overlap [ms] | Message time [ms] | Pause time [ms] |
|---|---|---|---|---|
| 5 | 40004 | - | - | 0 |
| 5 | 160016 | - | - | 0 |
| 5 | 200020 | - | - | 0 |
| 5 | 240024 | 42.4 | 94.6 | 9.4 |
| 5 | 320032 | 41.4 | 126 | 44 |
| 5 | 640064 | 41 | 257 | 175 |
| 5 | 1280128 | 36 | 510 | 442 |
| 5 | 2560256 | 26 | 1003 | 997 |
| 32 | 40004 | - | - | 0 |
| 32 | 80008 | - | - | 0 |
| 32 | 160016 | - | - | 0 |
| 32 | 192024 | 5.5 | 11.8 | 0.5 |
| 32 | 200020 | 5.5 | 12.2 | 1.35 |
| 32 | 240024 | 5.5 | 14.70 | 3.50 |
| 32 | 320032 | 4.5 | 20 | 10.5 |
| 32 | 640064 | 2.7 | 40 | 34.3 |
| 32 | 1280128 | -1.4 | 80 | 81.8 |
| 32 | 2560256 | -10.2 | 160 | 180.5 |
| 61.44 | 40004 | 0.348 | 1.164 | 0.426 |
| 61.44 | 80008 | 0.83 | 2.455 | 1.15 |
| 61.44 | 160016 | 1.38 | 5.04 | 0.35 |
| 61.44 | 320032 | 1.22 | 10.25 | 7.95 |
| 61.44 | 640064 | -3 | 20.4 | 26.4 |
| 61.44 | 1280128 | -4.6 | 41.8 | 51.5 |
| 61.44 | 2560256 | -18.5 | 83 | 115 |

**DEPARTMENT OF ELECTRICAL ENGINEERING**
**CHALMERS UNIVERSITY OF TECHNOLOGY**
Gothenburg, Sweden 2025
www.chalmers.se


CHALMERS
UNIVERSITY OF TECHNOLOGY