Implementation of an OFDM-based ISAC System on RF SoC using PYNQ Platform

Arif Arifianto

Department of Electrical and Information Technology
Lund University

Supervisor: Liang Liu (LTH), Anteneh A. Gebremariam (Ericsson AB), and Ariel Pola (Ericsson AB)

Examiner: Ove Edfors

August 7, 2024

Printed in Sweden
Tryckeriet i E-huset, Lund

Abstract

An integrated sensing and communications (ISAC) system using 5G NR waveform is implemented on a ZCU208 radio frequency system on chip (RFSoC) platform. Visualization of the data is performed using the PYNQ platform which facilitates data exchange between the processing system (PS) and the programmable logic (PL). The radar algorithm uses channel information in the frequency domain to detect a target's range and speed. Both the communication and sensing aspects of the design is verified to work in field testings. A simple post-processing method using background subtraction is implemented to isolate the target information from the effect of background stationary objects.

Popular Science Summary

Can you imagine if your Wi-Fi router can detect your current position? Or maybe you imagine a future where cars can drive itself through crowded traffic? The enabling technology for these features is sensing. For the longest time we have sensors to help machines understand its surrounding. However, sensors have a cost whether it be hardware, space, or computational complexity.

This is where the lucrative idea of ISAC comes in. ISAC can be considered as piggybacking a sensing system on top of a preexisting communication system. By smartly choosing algorithms and implementation method, information such as a target's range and speed can be acquired for very small cost. Moreover, this concept could be implemented in a way that only minimal changes is required on existing hardware.

This work focuses on an field programmable gate array (FPGA) implementation of an ISAC system. Using an FPGA greatly reduces the latency in performing the sensing computations. This system can be used as a testbed for further testing and development of more advanced ISAC systems which will accelerate the development of the concept. The implemented design has been field tested and verified to work in basic scenarios.

Acknowledgements

I would like to give my thanks to my supervisors: Liang Liu, Anteneh A. Gebremariam, and Ariel Pola. Without them I would not even know where to begin with this project. Their guidance have helped me hone an engineering and research mindset when tackling a difficult task. In addition I would like to thank Sijia Cheng and Mikael Henriksson who helped me with the basic concepts of the hardware.

I also extend my thanks and gratitude to Yizheng Wu, my thesis partner who have had the patience to deal with me as we worked on this project. I congratulate you and wish you good luck on your future as a doctoral student.

Last but not least I'd like to give appreciation to Bapak, Ibu, Dek Ica, and Rara who have all supported me on this monumental journey.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Next generation cellular technologies are motivating research into new applications which may not have been possible using legacy networks. Use cases such as vehicle-to-everything communications and sensor networks are currently being developed. The capabilities of existing communications infrastructure and standards must be able to follow the requirements needed for these use cases to be rolled out into the public. Among other things, the ability of a UE to sense its surroundings is an integral capability to be developed. Integrated sensing and communication (ISAC) is a concept where signals that are used for communications are also exploited to extract information about a transmitter's surroundings.

The HEXA-x project is a European initiative to shape the future of 6G. In [3] the gap in capability between curreng generation sensing solutions and the requirement for next generation applications is outlined. It highlights the need for robust and integrated sensing solutions some of which are implemented and documented in [4] and [5].

Some research about the conceptualization of ISAC systems within an automotive environment can be found in [6] and [7]. Currently the research of ISAC systems are mostly implemented using arbitrary waveform generators and signal analyzers where the data is then offloaded and processed asynchronously later such as in [8]. Other implementations use software defined radios such as the Universal Software Radio Peripheral (USRP) platform used within [9].

To move towards a system that is implementable in the real world, a test bed capable of real time processing with dedicated computational unit must be developed. Research of the usage of field programmable gate arrays (FPGA) to perform communication and sensing is therefore a logical next step in furthering the knowledge of ISAC systems. For example, a linear frequency modulated continuous wave (LFMCW) based radar implementations on an FPGA can be found in [10]. To allow for communication on the same waveform it is therefore desirable to implement a similar radar system but using orthogonal frequency division multiplexing (OFDM) waveforms.

1.2 Goal

This thesis aims to design and implement an OFDM radar and communications system on an FPGA board to balance the programming flexibility for development and speed needed to do real time processing. The research questions to be answered in this thesis is as follows:

- Which radar processing algorithms are suitable for target detection using the preexisting 3GPP 5G NR waveform?
- How well does the chosen algorithm perform when applied on the hardware? What optimizations can be done to achieve higher efficiency in silicon area/performance?
- Is it possible to share hardware to do both communication and sensing?

1.3 Research Method

The thesis project will be divided into stages with the final goal of having a working system that can both detect a target and communicate. The requirements for the sensing aspect is the ability to detect both range and relative velocity of the target. The different stages are defined as follows:

1. Literature review: Research the current state of the art that is related to the system and build up a solid theoretical foundation about the subject matter.
2. Simulations: Create a virtual environment where the different potential radar processing algorithms can be simulated in.
3. Hardware implementation: Implement the chosen algorithm into the FPGA and perform baseline measurements to verify that the system is working and if any calibration is needed.
4. Data collection: Conduct tests consisting of various scenarios to evaluate the system's performance.
5. Finalization: Analysis of results and write up of final report.

1.4 Report Organization

The report begins with an overview of basic theoretical concepts of OFDM communications and sensing in Chapter 2. The next chapter will go into detail on the different design aspects that is implemented during the course of this research. Chapter 3 will present the results gathered and how they can be used to evaluate the performance of the implemented design. Finally a conclusion will be given to summarize all of the findings within this thesis work.

Chapter 2

Theoretical Overview of OFDM Radar for Joint Communications and Sensing

The usage of the same waveform to both transmit data and perform sensing is an advantageous concept saving both bandwidth and power. OFDM has become the predominant modulation method in current and next generation communication systems. As such, this chapter will elaborate on viable processing techniques for OFDM radar.

First, a general description of OFDM modulation will be elaborated. This includes the appropriate mathematical conventions and assumptions with respect to this thesis work. The next portion of this chapter will compare the different processing techniques that is applicable for ISAC applications. A special focus will be given to the match filter and symbol domain estimation methods.

2.1 OFDM Modulation Technique

OFDM is a modulation technique which exploits multiple carriers to transmit data in parallel. The difference between OFDM and the more general FDM method is that in OFDM the sub-carriers are chosen in a way that ensures orthogonality between them [11]. This means that the sub-carriers which each contain conventional symbol based modulation schemes can be packed more densely together, allowing higher spectral efficiency.

The generation of an OFDM symbol can analogly be done using multiple mixers each with their own carrier frequencies representing each sub-carriers. This has now been replaced by a digital domain implementation through the usage of the DFT which are implemented using FFT/IFFT algorithms to reduce computing power. ISI is taken care of by prepending a portion of the OFDM symbol called the cyclic prefix (CP). A more detailed description of the OFDM modulation process will be described below.

One of the main advantage of OFDM is the simplification of channel equalization to combat fading. The usage of equivalent narrowband transmission with no cross-talk leads to the reduction of channel equalization to a series of multiplications in the frequency domain [12]. Consequently, no complex time domain adaptive filtering is needed to do equalization. The CP further simplifies the equalization by transforming the effects of the channel into a circular convolution. As a result,

each sub-carrier experiences flat fading which can be identified as just one complex coefficient.

2.1.1 Physical Layer Description of OFDM Waveforms

General Description

An OFDM symbol can be seen as the summation of its orthogonal sub-carriers in the time domain [13]. As such, let K be the total number of active sub-carriers for the OFDM symbol. The frequency of an individual sub-carrier at the k-th index f_k can then be described using the sub-carrier spacing (SCS) Δf as

$$f_k = f_0 + k\Delta f, \qquad k = 0, 1, ..., K-1. \tag{2.1}$$

As a mathematical aid, an alternate index p_k is defined as such that the 0-th index is located at the center frequency $f_c = f_{\frac{K-2}{2}} = f_{p_0}$. The new index is defined by

$$p_k = k - \frac{K-2}{2} \qquad k = 0, 1, ..., K-1. \tag{2.2}$$

Each of the k-th sub-carrier within the l-th OFDM symbol carries a quadrature amplitude modulation (QAM) symbol $c_{k,l}$. Hence the baseband time domain content of one sub-carrier is given as

$$s_{k,l}(t) = g_{rect}(t) c_{k,l} e^{j2\pi f_k t}. \tag{2.3}$$

Here the rectangular pulse g_{rect} is defined as having a constant pulse within time period $0 \leq t \leq T$. The complex amplitude of $c_{k,l}$ is chosen from a known constellation map of modulation order M. The modulation order indicates that b bits are coded within one symbol with the relation of $M = 2^b$.

The time domain representation of the entire l-th OFDM symbol is a simple summation of all of the sub-carriers

$$s_l(t) = \sum_{k=0}^{K-1} g_{rect}(t) c_{k,l} e^{j2\pi f_k t}. \tag{2.4}$$

The orthogonality of the sub-carriers can then be evaluated based on the description of the content within each sub-carrier. In mathematical notation, two signals s_1 and s_2 are orthogonal to each other within the time period t_1 and t_2 if and only if

$$\int_{t_1}^{t_2} s_1(t) \cdot s_2^*(t) dt = 0. \tag{2.5}$$

As OFDM is a frequency based modulation it is much more intuitive to evaluate the condition above in the frequency domain instead. First, by using the Fourier transform the frequency domain representation of the signal is given by

$$s_1(t) \xrightarrow{\mathcal{F}} S_1(f) \text{ and } s_2(t) \xrightarrow{\mathcal{F}} S_2(f). \tag{2.6}$$

The Parseval's theorem then states that the energy or power of a signal is equivalent in the time domain and in the frequency domain. Thereby, the orthogonality condition in the frequency domain can be written as

$$\int_{f_1}^{f_2} S_1(f) \cdot S_2^*(f)\, df = 0. \tag{2.7}$$

Applying the Fourier transform to the rectangular pulse yields the spectrum

$$g_{rect}(t) \xrightarrow{\mathcal{F}} G_{rect}(\omega) = T\frac{sin(\pi fT)}{\pi fT}. \tag{2.8}$$

The so called *sinc* function has zero crossings at integer multiples of $1/T$. This means that if the SCS is chosen correctly then the zero crossings can be made to line up with the peaks of the other sub-carriers. The value of Δf as defined in (2.1) needed to maintain orthogonality within the sub-carriers is then

$$\Delta f = f_{k+1} - f_k = \frac{1}{T}. \tag{2.9}$$

The bandwidth, BW, can then be found to be

$$BW = K \cdot \Delta f. \tag{2.10}$$

Generating OFDM Symbols

The generation of multi-carrier modulation waveforms were classically a tedious operation requiring multiple mixers for each of the sub-carriers with its equivalent in the receiver side. However, in modern day systems the OFDM symbols could be generated using the discrete inverse Fourier transform (IDFT) and then demodulated using the discrete Fourier transform (DFT). This allows for a digital approach which is much more efficient.

Consider the time domain OFDM symbol sampled to generate N samples. The sampling time T_s needs to be set to T/N to generate these samples. Conversely the sampling frequency f_s is then defined as

$$f_s = \frac{N}{T} = N \cdot \Delta f. \tag{2.11}$$

Referring back to (2.4) with the index p_k and the sampling frequency defined above, the sampled version of $s_l(t)$ is

$$\begin{aligned} s_l[n] = s_l(nT_s) &= \sum_{k=0}^{K-1} g_{rect}(nT_s) c_{k,l} e^{j2\pi f_k n T_s} \\ &= \sum_{k=0}^{K-1} c_{k,l} e^{j2\pi p_k \Delta f n T_s} \\ &= \sum_{k=0}^{K-1} c_{k,l} e^{j2\pi p_k \frac{n}{N}}. \end{aligned} \tag{2.12}$$

Note the similarities between the equation above and the definition of an IDFT. One modification that is needed to achieve the N-point IDFT that is desired is to pad zeros to the K sub-carriers to achieve an N point summation. The zeros are

added to the lowest and highest indexes of $c_{k,l}$. The number N is most commonly chosen to be a power of two so the Inverse Fast Fourier Transform (IFFT) and the Fast Fourier Transform (FFT) can be used to efficiently compute the IDFT and FFT.

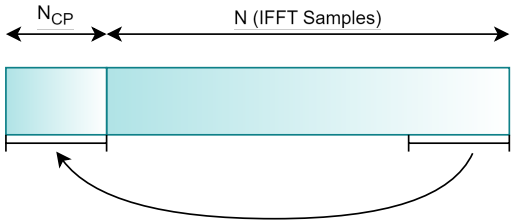

Figure 2.1: Diagram of cyclic prefix insertion for one OFDM symbol.

The final step of the generation of the OFDM symbol is the addition of the cyclic prefix (CP). This is done by copying the last N_{CP} samples of the time domain symbol and placing it in front of the symbol. These samples takes T_{CP} amount of time and makes the total observation time $T_O = T + T_{CP}$.

The function of the CP is twofold, first it allows a guard period between two OFDM symbols thus negating the effects of inter-symbol interference (ISI) from multipath propagation. The effect of the channel can be represented by a convolution of the time domain symbol and the channel impulse response $h(t)$ or if sampled then $h[n]$. This convolution means that the samples from the previous symbol have the potential of spreading into the current observed symbol. With this guard interval the residuals from the previous symbol's multipath will land within the CP which will be discarded before the demodulation.

The second reason is to simplify the equalization process. The existence of CP converts the linear convolution with the channel into a circular convolution within the effective period of one OFDM symbol. It can be seen that the first samples within the useful part of the OFDM symbol is affected by the multipath of the CP which comes from the last samples of the symbol thereby proving the circularity. After the DFT in the demodulation process, the circular convolution becomes a multiplication between each sub-carrier and its complex channel coefficient. This process can be illustrated mathematically as follows

$$y[n] = h[n] \otimes s[n] + w(n) \xrightarrow{\mathcal{F}} Y[k] = H[k] \cdot S[k] + W(k). \tag{2.13}$$

As such, the equalization can be done by simply dividing the received symbols with the estimated channel coefficients in the frequency domain.

2.1.2 Demodulating OFDM Symbols

Synchronization

Though this work will focus more on the sensing side of the ISAC system, a simple receiver to demodulate the communication information will be described. The demodulation of the symbols starts with the synchronization process which aims

to identify the starting point of the signal θ, and also detect and carrier frequency offset (CFO) ε. An offset in the time domain sampling point translates to a constant phase offset in the frequency domain, in other words a rotation of the constellation by a constant value (within the coherent time). On the other hand the CFO will cause a phase offset that continuously changes at a constant rate.

Schmidl and Cox [14] outlined a synchronization method using a training symbol which is made up of two identical halves of length L. A correlation of two sets of L samples with N offset between them is then taken to find the start of the packet.

The CP can be used as this repeated element in the method above as such that $L = N_{CP}$. In this implementation the correlations and energy term is calculated using [15]

$$\gamma(n) \triangleq \sum_{k=n}^{n+L-1} y(k)y^*(k+N), \tag{2.14}$$

and

$$\xi(n) \triangleq \sum_{k=n}^{n+L-1} |y(k)|^2 + |y(k+N)|^2. \tag{2.15}$$

The magnitude of the correlation coefficient between the two correlation windows is defined as

$$\rho \triangleq \frac{\sigma_s^2}{\sigma_s^2 + \sigma_n^2}. \tag{2.16}$$

where σ_s^2 is the signal power and σ_n^2 is the noise power. This metric is also known as the signal-to-interference-plus-noise ratio (SINR).

The maximum likelihood (ML) estimate of the start position θ is then

$$\hat{\theta} = \arg\max_{\theta}(2\ |\gamma(\theta)| - \rho\ \xi(\theta)). \tag{2.17}$$

Afterwards, the estimated delay is used to estimate the CFO ε using

$$\hat{\varepsilon} = -\frac{1}{2\pi}\angle\gamma(\hat{\theta}). \tag{2.18}$$

The correction is then simply done by choosing $y[\hat{\theta}]$ as the first sample and compensating the CFO through multiplication of the received sample by $e^{-j2\pi\hat{\varepsilon}\frac{n}{N}}$.

Equalization

The received signal will be impaired by the effects of the channel. The signal will be altered by propagation mechanisms such as delay, path loss, reflection, and diffraction. In this work the channel effects are generally modelled as the instantaneous channel response. A thorough description of the assumed channel model will be given in 2.1.3. A more statistical model of the fading profile will also be used to perform link budget analysis.

As previously mentioned, the addition of the CP allows for a simple equalization scheme in the frequency domain. Correcting the effects of the channel as outlined in (2.13) is simply a matter of estimating the channel complex channel coefficients in the time domain, then dividing the received symbols with the estimated value. Known pilot symbols can be transmitted in selected sub-carriers which can then be extrapolated to find the estimated channel coefficient of the remaining sub-carriers.

Consider the estimated frequency domain channel coefficient of $\hat{H}_{k,l}$ indexed to the sub-carriers without the addition of padding. The estimated IQ symbols $\hat{c}_{k,l}$ can then be found to be

$$\hat{c}_{k,l} = \frac{\tilde{c}_{k,l} + w_{k,l}}{\hat{H}_{k,l}} \tag{2.19}$$

where $\tilde{c}_{k,l}$ is the sent symbol altered by the channel and $w_{k,l}$ is the sampled noise component for the associated sub-carrier and symbol index.

Decoding

The remaining step is to decode the received IQ symbols. Typical detector structures can be used for this stage. In this work the minimum euclidean distance detector is employed to decode the received symbols. Each of the received noisy and altered symbols in the receiver are compared against a known map of symbols. The decision variable is to find the closest match to the symbol given the map. Mathematically the detector is described as [16]

$$\begin{aligned} \text{Decision } \hat{m} = m_l &\Leftrightarrow \min_i D_{r,i}^2 = D_{r,l}^2 \\ &\Leftrightarrow \max_i \int_0^{T_s} r(t) z_i(t) dt - E_i/2 \end{aligned} \tag{2.20}$$

2.1.3 Radar Perspective of Channel Effects on OFDM Waveforms

Implementing a sensing algorithm using OFDM requires knowledge of how the channel will transform the transmitted signal. In this work sensing will be performed using back-scattered communication signals, akin to conventional monostatic pulsed radar systems. This section will discuss the channel effects that will have a bearing on the design of the sensing algorithms.

Path Loss

When an electromagnetic signal is transmitted from a source in space it will experience attenuation before being received in the receiver. This attenuation can be caused by a combination of, among other things, free-space loss, reflection, absorption, or diffraction which can be added up together to form the total path loss. This means that the signal, as observed by the receiver, will be scaled by the attenuation.

One of the most important source of attenuation is the free-space loss. A radio signal's wave front expands in the shape of a sphere, meaning the power density

decreases the more distance a signal has to travel. Friis equation describes this phenomenon mathematically as

$$\frac{P_r}{P_t} = G_t G_r \left(\frac{\lambda}{4\pi d}\right)^2 \tag{2.21}$$

where P_t and P_r are the transmitted and received power respectively, G_t and G_r is the transmit and receive gain, λ is the signal's wavelength, and d is the distance between the receiver and transmitter. (2.21) is only valid for transmission within the Fraunhofer distance, or the so called far field, which is a region where $d > 2D^2/\lambda$, D being the largest dimension of the antenna.

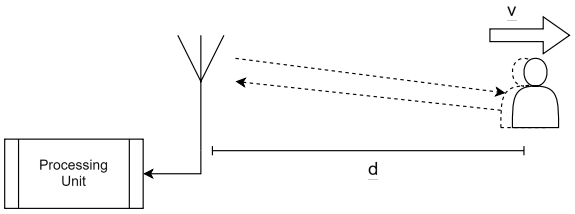

Figure 2.2: A general monostatic scenario with one target.

In the context of a radar system, consider a simple monostatic set up where the transmitter and receive uses the same antenna as in Figure 2.2. The transmitted signal is reflected back by the scattering object. Multiple targets numbering P at a distance d_p and with a radar cross section (RCS) of σ_p will reflect the transmitted signal with a different attenuation. As such the attenuation in voltage as observed by the receiver from the p-th object is [17]

$$b_p = \sqrt{\frac{\lambda^2 \sigma_p}{(4\pi)^3 d_p^4}}. \tag{2.22}$$

Propagation Delay

In the monostatic model the transmitted signal must travel from the transceiver to the target and back to the radar platform. The total distance travelled by the signal reflected by a particular target is $2d_p$. This is translatable to a delay in the time domain of

$$\tau_p = \frac{2d_p}{c}. \tag{2.23}$$

The delay can also be viewed as a phase shift in the frequency domain by $e^{-j2\pi\tau_p}$.

Doppler Shift

The movement of a reflecting object relative to the radar platform will cause a shift in frequency of the transmitted signal. This is due to the fact that the propagation distance increases (or decreases) in relation to time. In the monostatic model, the

signal experiences this effect for the first time when the signal reaches the target, then again when the signal is received again by the radar platform. As such, the doppler shift v caused by a relative velocity of v is formulated as [18]

$$\upsilon_p = f_c \frac{2\mathrm{v}_p}{c}. \tag{2.24}$$

Note that the doppler frequency is positive when the target is moving closer, and negative when the target is moving away.

Overall Channel Model

By combining the channel effects above an overall model of the channel can be formulated. Firstly, let's consider that the path loss and propagation delay of a reflection can change during the observation period of the symbol due to movement. This means that the channel becomes time-varying. The baseband equivalent signal as seen by the receiver, $r(t)$ can be considered to be a convolution, as briefly mentioned in (2.13), of the transmitted signal with the time-varying channel impulse response as follows [19]

$$r(t) = \int_{-\infty}^{+\infty} h(t,\tau)s(t-\tau)d\tau + w(t), \tag{2.25}$$

with the time-varying channel impulse $h(t,\tau)$ being

$$h(t,\tau) = \sum_{p=0}^{P-1} b_p(t) e^{j2\pi f_c \tau_p(t)} e^{j2\pi \upsilon_p t} \delta(t-\tau_p(t)). \tag{2.26}$$

A note can be made on some simplifications that can be done one the given channel model. First, the change in distance within one observation period is relatively small to the point where the position could be considered as constant. As such both b_p and τ_p can be made to be constants which simplifies the channel model as described by Braun [20] to be

$$r(t) = \sum_{p=0}^{P-1} b_p s(t-\tau_p) e^{j(2\pi \upsilon_p t + \phi_p)} + w(t). \tag{2.27}$$

The reflection will induce a random phase of ϕ_p and is affected by AWGN $w(t)$.

Though (2.25) and (2.26) gives a more accurate description of the channel, the specificity is not required to design the corresponding radar processing algorithm. In the context of simulation purposes it can be said that the difference between the two models is marginal, as explored in later portions of this work.

2.2 OFDM Radar Processing Algorithms

The aim of the radar processing algorithm is to estimate the distance, relative velocity, and RCS of the target. Throughout the years many researchers have discussed the usage of OFDM signal for sensing. Levanon [21] proposes the usage of the auto-correlation function (ACF) to extract the range of a reflector. His

method utilizes a specific cyclically shifted phase coded sequences to populate the sub-carriers, and as such is unviable for the ISAC concept. A proposal to use the same waveform for both communication and sensing is elaborated in [22] with the conclusion that the ACF of a ultra wideband (UWB) OFDM symbol with data in its sub-carriers is sufficient to perform ranging and communication.

In this section, two dedicated processing algorithm for OFDM radar that can coexist with communications will be discussed. The first method uses a match filter to extract the range-doppler information of the reflected signal. The second method operates on the demodulated frequency domain symbols.

2.2.1 Match Filter Based Estimation Algorithm

A common approach to target detection adapted from radars using other waveform is the usage of multiple correlator templates corresponding to a particular delay and doppler value. In the digital domain the correlator could be substituted with a match filter. The FFT/IFFT can then be used to implement the match filter in the frequency domain. This approach is first proposed by Tigrek [23] who outlines the following algorithm to calculate the range-doppler profile:

1. Demodulate OFDM symbols using FFT.
2. Perform cyclic shifts on demodulated symbols according to a given doppler profile.
3. Compensate for starting phase.
4. IFFT across k sub-carriers for l symbols.
5. Repeat **step 2** for different cyclic shift value, accumulating the range profiles.
6. FFT across l symbols to generate the range-doppler profile.

In matrix notation the processing can be described as [24]

$$\mathbf{y} = \mathbf{F^{-1}PC^{-1}Fr} \tag{2.28}$$

with $\mathbf{r}$ being the $N \times 1$ column vector representation of the sampled back-scattered signal as described in (2.27). The next matrices are all of size $N \times N$. $\mathbf{F}$ is the FFT matrix for demodulation, $\mathbf{C^{-1}}$ is the cyclic shift operation which is a cyclically shifted identity matrix, $\mathbf{P}$ makes up the initial phase compensation, and the final IFFT represented by $\mathbf{F^{-1}}$.

Some weakness of the algorithms is the high computational complexity due to the requirement of performing match filtering for all doppler and delay templates. In addition, the processing produces doppler ambiguity due to the presence of CP and guard bands. Some methods are proposed to circumvent this issue by sending reference symbols, but this would mean that the system can't be used for communication in this period.

Around the same time, another method proposed in [25] performs the match filtering by multiplying the received frequency domain symbols with the conjugate of the transmitted symbols of $c^*_{k,l}$. The limiting factor in this case is that correlations may exist within the communication data [26]. This effect is further amplified when transmitting realistic information which are rarely gaussian in nature.

2.2.2 Symbol Domain Based Estimation Algorithm

Based on the discussions about the match filter based algorithm, it is desirable that the estimation of the range-doppler profile be independent of the data being sent within the communication waveform. An information-agnostic range-doppler processing method is first described in [27] by dividing the received frequency domain phase modulated symbols with the symbols initially sent.

To study the effects of the channel on the transmitted symbols, it is useful to evaluate them as matrices. Consider the matrix $\mathbf{F}_{TX}$ containing the complex symbols of $c_{k,l}$ as

$$\mathbf{F}_{TX} = \begin{bmatrix} c_{0,0} & \cdots & c_{0,M-1} \\ \vdots & \ddots & \vdots \\ c_{N-1,0} & \cdots & c_{N-1,M-1} \end{bmatrix}. \tag{2.29}$$

In the notation above, each column of the matrix contains complex symbols of the N sub-carriers of the l-th OFDM symbol, and conversely each row contains the symbols located in the k-th sub-carrier of the M symbols.

The channel effects on $\mathbf{F}_{TX}$ can be derived by evaluating the sampled version of (2.27). In the sampled frequency domain, the delay is translated to a frequency dependant phase offset of $e^{-j2\pi\tau_p(f_0+k\Delta f)}$. The doppler from (2.24) is also sampled at every OFDM symbol period to generate the complex sinusoid of $e^{j2\pi l T_O \upsilon_p}$. Thereby the system can be modelled as [20]

$$(\mathbf{F})_{k,l} = \sum_{p=0}^{P-1} b_p e^{-j2\pi\tau_p k\Delta f} e^{j2\pi l T_O \upsilon_p} e^{\varphi_p} + (\mathbf{Z})_{k,l}. \tag{2.30}$$

$(\mathbf{F})_{k,l}$ contains the element-wise division of $(\mathbf{F}_{RX})_{k,l}/(\mathbf{F}_{TX})_{k,l}$. The constant frequency term of f_0 is combined with the random frequency offset within $\varphi_p = \phi_p - 2\pi\theta_p f_0$. The noise is also defined as $(\mathbf{Z})_{k,l} = (\mathbf{W})_{k,l}/(\mathbf{F}_{TX})_{k,l}$ but keeps its white and gaussian distribution.

Estimating the values of τ_p and φ_p can be done through spectral analysis over k and l respectively [28]. The range-doppler profile, or so called "periodogram", can be defined as

$$Per_{\mathbf{F}}(n,m) = \frac{1}{NM}\left| \sum_{k=0}^{N_{per}-1} \left(\sum_{l=0}^{M_{per}-1} (\mathbf{F})_{k,l} e^{-j2\pi\frac{lm}{M_{Per}}} \right) e^{j2\pi\frac{kn}{N_{Per}}} \right|^2. \tag{2.31}$$

The innermost summation is the M_{per}-point FFT over the N rows of $\mathbf{F}$ which cancels out the exponential term, leaving peaks at the doppler index. In the same way, the next summation shows the N_{per}-point IFFT over the M symbols to extract the range index.

The indices of $\hat{n}$ and $\hat{m}$ can be converted to the actual distance and relative velocity using

$$\hat{d} = \frac{\hat{n}c}{2\Delta f N_{Per}} \text{ and } \hat{\mathrm{v}} = \frac{\hat{m}c}{2T_O M_{Per}}. \tag{2.32}$$

Chapter 3

System Design and Implementation

In this chapter a ISAC system based on the symbol-domain based processing is proposed. First, a general high level overview of the system is given in Section 3.1. Next is an explanation about the hardware upon which the system is developed with. The details about each processing blocks and its algorithms will be elaborated within Section 3.3.

3.1 System Model

This section aims to lay out a system level view of the design. Design considerations are examined from a theoretical standpoint and supported with the results of simulations. This section also contains thorough description of the requirements and constraints for both communications and sensing.

3.1.1 Processing Algorithm Selection

Detection Performance

The design process starts with an evaluation of the match filter and symbol domain based algorithms as described in Sections 2.2.1 and 2.2.2. A simulation environment is used to test the performance of the algorithms and also to provide base for the eventual hardware implementation.

It is mentioned in [26] that the peak-to-sidelobe ratio (PSR) of the match filter method is heavily affected by correlations within the transmitted information data. The Figures 3.1 and 3.2 shows the simulation result for the high and low SNR regime respectively. The simulation results shows that the PSR of the modulation symbol processing is much higher when compared to the match filter. However, in the low SNR regime, the gap in PSR is considerably lower.

The internal correlations within the data is the limiting factor for the match filter algorithm. As such, even in high SNR scenarios the performance does not continue to improve. The requirements of the ISAC concept dictates that the system must be able to perform sensing unhindered by the transmission of data. As such, the symbol based algorithm is better suited for this purposes.

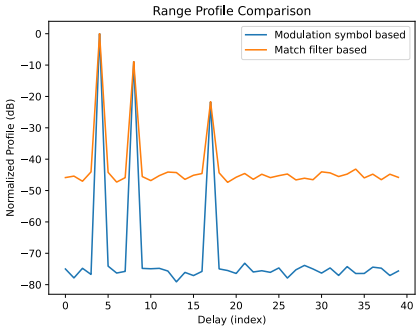

Figure 3.1: 50 dB SNR simulation of processing algorithm.

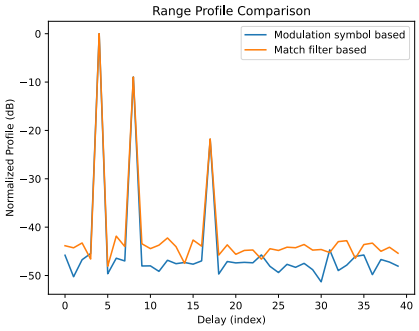

Figure 3.2: 10 dB SNR simulation of processing algorithm.

Algorithmic Complexity

Another consideration that must be made is the algorithmic complexity. Both methods use the IFFT and FFT to extract the range and velocity of the targets. In this sense the complexity of the two is similar. Where they differ is the pre-processing that is done beforehand.

The match filter method requires a multiplication with the complex conjugate of the transmitted symbol. Consider $\hat{c}_{k,l} = \mathrm{a} + \mathrm{b}i$ and $c^*_{k,l} = \mathrm{c} + \mathrm{d}i$, the multiplication operation can be broken down into

$$\begin{aligned} \hat{c}_{k,l} \cdot c^*_{k,l} &= (\mathrm{a} + \mathrm{b}i) \cdot (\mathrm{c} + \mathrm{d}i) \\ &= (ac - bd) + (ad + bc)i. \end{aligned} \tag{3.1}$$

A naive implementation of the operation requires four multiplications, and two additions/subtractions. There exist algorithms that only require three real value multiplications [29]. The division in the symbol based processing algorithm can also be analyzed in a similar way as follows

$$\begin{aligned} \frac{\hat{c}_{k,l}}{c^*_{k,l}} &= \frac{(\mathrm{a} + \mathrm{b}i)}{(\mathrm{c} + \mathrm{d}i)} \\ &= \frac{(\mathrm{a} + \mathrm{b}i)(\mathrm{c} - \mathrm{d}i)}{(\mathrm{c} + \mathrm{d}i)(\mathrm{c} - \mathrm{d}i)} \\ &= \frac{\mathrm{ac} + \mathrm{bd}}{\mathrm{c}^2 + \mathrm{d}^2} + \frac{\mathrm{bc} - \mathrm{ad}}{\mathrm{c}^2 + \mathrm{d}^2}i. \end{aligned} \tag{3.2}$$

In this case it takes six different multiplications, three additions/subtractions, and two divisions.

The conclusion is that the division requires more computing power to perform. The total number of times the complex multiplication or division that must be performed in both algorithm is identical. There is a consideration on the type of operations required. Addition and subtraction is easily implementable in hardware. Multiplication and especially division requires a lot of "area" to implement unless

specialized digital signal processing (DSP) blocks are used. The required data precision will also affect the area that is needed to implement the algorithm.

An aspect that must be considered for both algorithm is the memory requirement. Different operation might have different accumulation requirements. This must be balanced with the inherent need for a memory to store the incoming symbols before the doppler processing can be performed.

Selected Algorithm

In this work the symbol based algorithm is chosen to be implemented. The chosen hardware has many dedicated DSP blocks and as such possesses the required computing power. For other cases where hardware limitation is a point of consideration, it might be advisable to choose the match filter processing with sacrifice to the PSR.

3.1.2 High Level Design

The design of the system is mostly uses the same functional blocks as a typical OFDM transmitter/receiver. This means the algorithm could be easily deployed for existing hardware architectures. Some additional processes are added to the receiver to add the radar functionalities. A block diagram of the system can be found in Figure 3.3.

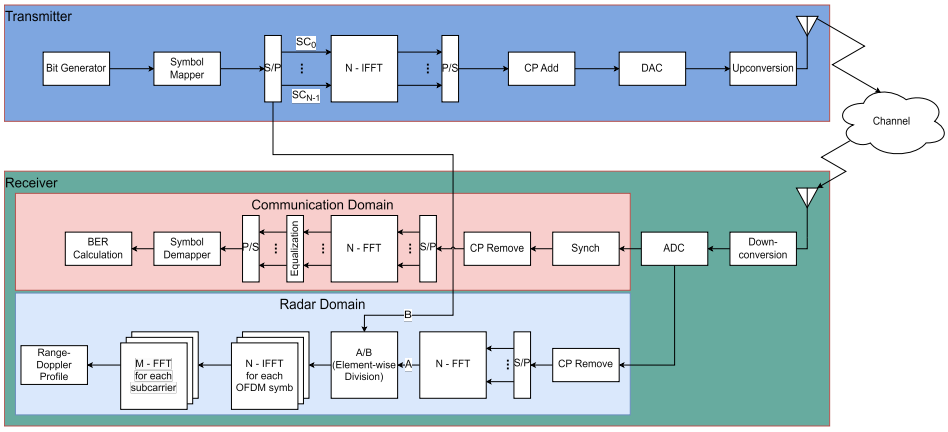

Figure 3.3: High level block diagram of the system.

In the proposed design a time division duplex (TDD) mechanism is assumed. This means that communication is performed in time-slots where only uplink or downlink can be performed. This means that when the transceiver unit is transmitting information, the receiver is turned on to receive the back-scattered signal and perform radar functionalities. When the unit is within the communication receive window, the received signal are processed only by the communication domain.

Transmitter

The transmitter implements the processes as outlined within Section 2.1.1. A bit generator feeds random bits or bits representing information into an arbitrary symbol mapper. Here the mapping that can be used is constrained into M-QAM modulation. The symbols are then allocated into the appropriate sub-carriers taking into account the addition of pilot symbols and guard bands. IFFT can then be performed to generate the time domain OFDM symbol. The final step in the digital domain is the addition of the CP.

The conversion to the analog domain is performed by the digital-to-analog converter (DAC). The sampling frequency must be chosen critically in accordance to (2.11) and (2.10) to generate the required bandwidth. A more detailed parameterization will be discussed in the next section. The analog baseband signal can then be upconverted into the desired passband with carrier frequency f_c and transmitted over the air. The analog domain in the receiver then reverses these processes by downconverting into baseband and then sampling it into the digital domain.

Communication Processing

The receiver can then be split into the communication and radar part. A TDD system is assumed. As such, the radar part is active during the transmission window (exploiting the back-scattered transmitted signal), and the communication part is active during the receiving window. The processing method also works for frequency division duplex (FDD), but in that case a dedicated radio frequency (RF) chain tuned to the transmit frequency is needed to receive and process the back-scattered signal.

The communication processing starts with the time and frequency synchronization according to (2.17) and (2.18). After the starting point of the signal is recognized, the CP can be removed. The received signal is parallelized and demodulated with the N-point FFT. The next step is to equalize the signal in the frequency domain through simple division with the estimated channel coefficients. The remaining steps is de-mapping the soft-output symbols into hard-output bits. A calculation of the bit error rate (BER) can then be done by comparing the known transmitted bits and the received bits.

Radar Processing

In the case of the radar processing, it requires the back-scattered signal without any pre-processing such as synchronization or equalization. Within the radar processing system the received signal is recorded from the moment the transmitted signal leaves the transmit module after which the CP is removed. As such all reflections must be received before T_{CP} to ensure that the algorithm works properly.

The demodulation FFT is done directly on the output of the analog-to-digital converter (ADC). The element-wise division is then performed on the frequency domain symbols to get the $\mathbf{F}$ matrix as defined in (2.30). The next steps extracts the range-doppler profile through the N-IFFT of each incoming OFDM symbol,

followed by the M-FFT of each active sub-carriers. Further processing of the output periodogram can be done to perform more reliable detection.

3.1.3 System Parameterization

The implemented design will heavily depend on the required specifications of the system. Physical constraints can also pose restrictions on the capabilities of the designed system. As the main focus of this work is to "piggyback" a sensing system on top of an existing communications system, the design will follow suite where the radar system will follow the constraints set by the communications part.

Communications Standards and Constraints

This work will consider the general physical layer description of the 5G NR standard. As the sensing part is data agnostic, it is not constrained to being able to work only for specific signals and channels. As such, implementation of detailed protocols and channels within the 5G NR standard is considered to be out of the scope of this work.

In this work the operating carrier frequency is set to work in the millimeter-wavelength (mmWave). This band is defined by 3GPP in [30] as FR2 which is further classified into FR2-1 consisting of 24.25–52.6 GHz and FR2-2 for the higher frequencies of 52.6–71 GHz. The specific frequency that will be used is 28 GHz which lies within the operating band n257.

Varying transmission bandwidth can be achieved by manipulating the number of active resource blocks N_{RB}. Each resource block contains 12 sub-carriers. An exhaustive table showing the relation between the SCS, N_{RB}, and the corresponding bandwidths can be found in [30, Table 5.3.2-1].

μ	SCS (kHz)	T (μs)	T_{CP} (μs)	T_O (μs)	$N_{slot}^{subframe}$
0	15	66.67	4.69	71.35	1
1	30	33.33	2.34	35.68	2
2	60	16.67	1.17	17.84	4
3	120	8.33	0.57	8.92	8
4	240	4.17	0.29	4.46	16
5	480	2.08	0.14	2.22	32
6	960	1.04	0.07	1.11	64

Table 3.1: Physical layer parameters of different numerologies.

Another important parameter within the 5G NR standard is the numerology μ. The numerology corresponds to the SCS according to $\Delta f = (15 \text{ kHz}) \cdot 2^{\mu}$ [31]. As the SCS increases, the length of one OFDM symbol decreases linearly. The first two numerologies (0 and 1) are not defined for the FR2 band.

The largest time unit is the frame which is a 10 ms time period and is made up of 1 ms subframes. Each subframe is made up of $N_{slot}^{subframe}$ slots which are a group of 14 OFDM symbols. A detailed breakdown of the interplay between the chosen numerology with the different parameters can be observed in Table 3.1.

Another point of importance is the length of the CP for a given numerology. The length of CP for the l-th OFDM symbol within a subframe in seconds as provisioned in the 3GPP standard [31] is

$$T_{CP,l} = \begin{cases} T_{ref} \cdot 144 \cdot 2^{-\mu}, \ l = 0 \vee l = 7 \cdot 2^{\mu} \\ T_{ref} \cdot (16 + 144 \cdot 2^{-\mu}), \ else \end{cases} \tag{3.3}$$

with T_{ref} being the reference LTE sampling period defined as $T_{ref} = 1/(2048{\cdot}15{\cdot}10^3)$ corresponding to the 2048 sub-carriers and 15 kHz SCS in that standard. An exception is made for the first and middle (from the perspective of one subframe) which contains $T_{ref} \cdot 16$ more CP duration. In this work, this differing length of CP is ignored and it is assumed that all of the CP follows (3.3), a detailed study about the effects of differing CP lengths on the sensing performance can be found in [32]. To get the length of CP in numbers of samples all that is needed is to divide T_{CP} with the system sampling period of T_s.

Radar Parameterization

The performance characteristics of the symbol domain processing is outlined in [33]. The following conditions must be fulfilled for the processing algorithm to work:

- The length of CP T_{CP} is longer than the maximum excess delay from the furthest reflection τ_{max}.
- The SCS Δf is one order of magnitude larger than the maximum excess doppler υ_{max}.
- The OFDM frame length and bandwidth lies within the coherence time and bandwidth of the channel.

For typical indoor and short-medium range outdoor cases the three conditions above are easily fulfilled. For example, with numerology 3 a τ_{max} of 0.57 μs corresponds to a reflection at 85.5 meters. In the velocity detection, consider a carrier frequency of 28 GHz and the same numerology as before. The υ_{max} becomes 12 kHz which translates to a movement with relative velocity of 64.28 m/s, a speed that won't be encountered in terrestrial vehicles.

Another fundamental limit of the algorithm is the maximum unambiguous range and velocity. Ambiguity arises because of the discrete sampling of the **F** matrix, causing higher frequency oscillations to be aliased. The unambiguous range is [20]

$$d_{unamb} = \frac{c_0 \tau_{unamb}}{2} = \frac{c_0}{2\Delta f}. \tag{3.4}$$

The unambiguous velocity is also defined as

$$\mathrm{v}_{unamb} = \frac{c_0 \upsilon_{unamb}}{2f_c} = \frac{c_0}{2f_c T_O}. \tag{3.5}$$

The resolution that is achievable for the radar depends on the parameters of the OFDM waveform. In general a larger bandwidth gives a more precise

range resolution, and a longer frame yields a better velocity resolution. It can be formulated mathematically as

$$\Delta d = \frac{c_0}{2K\Delta f} \text{ and } \Delta \text{v} = \frac{c_0}{2f_c T_O M}. \tag{3.6}$$

Example Configurations

The communications and radar constraints are combined together to evaluate a number of possible configurations. These parameters are tested within a simulation environment to better understand its theoretical performance and its feasibility in a real system. A comparison of different bandwidths, SCS, and number of OFDM symbols observed will be studied.

No.	Distance (m)	Velocity (m/s)	RCS (m^2)
1	6	-10	10
2	10	3	10
3	21	0	10

Table 3.2: Targets for parameter evaluation simulation.

The targets used in all simulations are listed in Table 3.2. Each of the periodogram are cropped to just show the region of interest. In this way, differences in resolution can be clearly visualized. All simulations use a carrier frequency of 28 GHz. A detailed summary of the achieved performance for all sets of configurations can be found in Table A.1.

The first set of simulation uses different transmission bandwidth for $\mu = 3$ by increasing the number of active sub-carriers used in the OFDM symbol. The simulation result can be observed in Figures 3.4–3.7. The visible increase in range resolution (specificity between range bins) follows (3.6).

The next set of simulation compares the effects of maintaining the same bandwidth for two different SCS values, in this case 60 kHz and 120 kHz. It is clear in Figure 3.8 that the velocity resolution is more precise in comparison to Figure 3.9. The better velocity resolution is achieved by lengthening the OFDM symbol period of T_O through decreasing f_s. The range resolution does not change since the bandwidth for both systems are fixed to 100 MHz by activating more sub-carriers in the 60 kHz SCS case.

The range resolution can also be increased or decreased by manipulating the number of observed OFDM symbols M. Figures 3.10–3.12 shows the progression in velocity resolution with the increase of M.

In conclusion, a larger bandwidth combined with many observed OFDM symbols will result in better resolution. Figure 3.13 shows an extreme case that maximizes the two parameters. Though achievable in simulation, this extreme parameterization is not practical in a real world scenario. From a channel study perspective, a very long observation period might mean that it exceeds the coherent time of the channel. The bandwidth might also extend beyond the range of the coherent bandwidth. In the hardware design perspective this also poses a problem since more resources

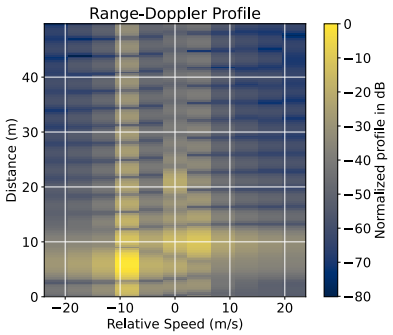

Figure 3.4: 50 MHz bandwidth simulation.

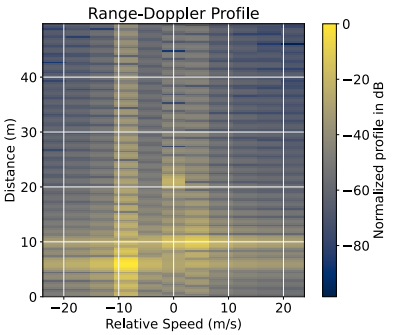

Figure 3.5: 100 MHz bandwidth simulation.

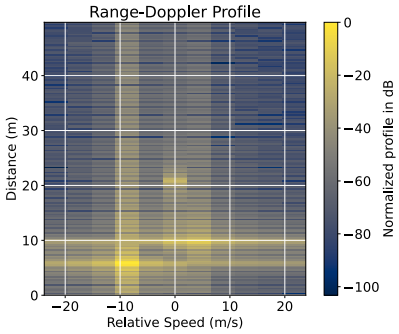

Figure 3.6: 200 MHz bandwidth simulation.

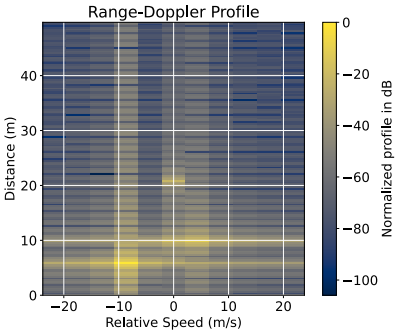

Figure 3.7: 400 MHz bandwidth simulation.

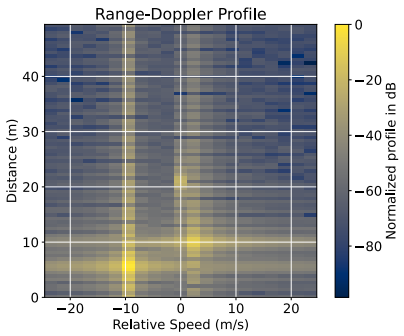

Figure 3.8: 60 kHZ SCS with fixed bandwidth simulation.

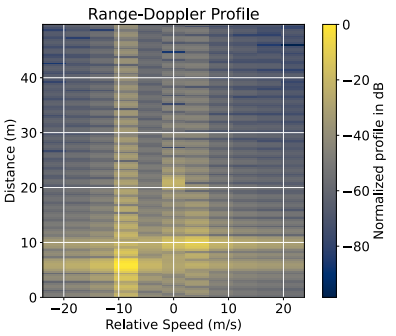

Figure 3.9: 120 kHZ SCS with fixed bandwidth simulation.

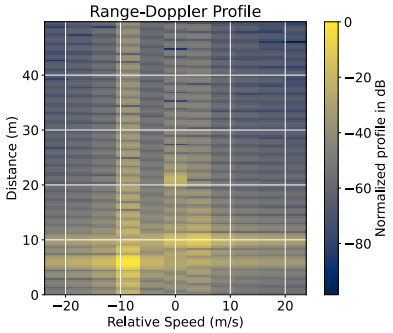

Figure 3.10: 128 observed symbols simulation.

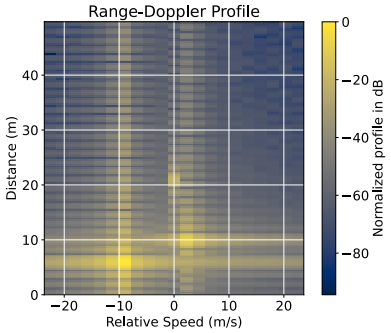

Figure 3.11: 256 observed symbols simulation.

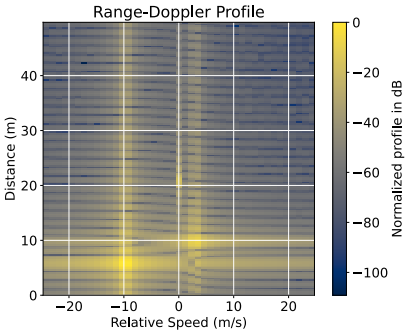

Figure 3.12: 512 observed symbols simulation.

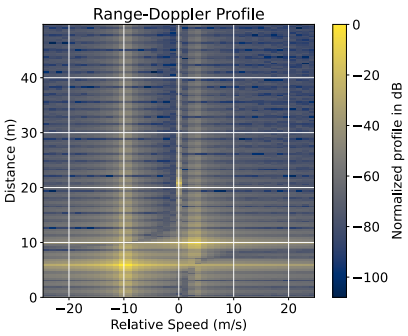

Figure 3.13: Simulation of 400 MHz bandwidth with 512 observed symbols

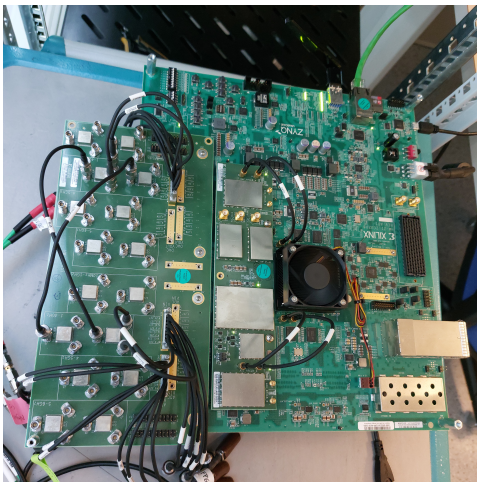

Figure 3.14: The ZCU208 board used in this thesis work.

must be allocated to store the current sent of symbols to be processed or perform higher order FFT/IFFT.

A reasonable balance between the desired resolution and resource usage must be considered. From the simulation data above it is decided that the system will use numerology 3 with 100 MHz bandwidth as a baseline. Changes to the bandwidth and observation length can be made to achieve less or more resolution from the baseline configuration.

3.2 Hardware Description

In this section, all of the relevant details of the hardware is elaborated. The first part will focus on the radio frequency system-on-a-chip (RFSoC), its structure, and some critical design configurations. The second half will explain the RF frontend that is used to upconvert the signal from the intermediate frequency (IF) into the radio frequency (RF).

3.2.1 ZCU208 RFSoC

The ZCU208 is an evaluation board from AMD-Xilinx equipped with the Zynq UltraScale+ RFSoC Gen 3 ZU48DR chip [34]. The evaluation kit provides easy to use interfaces which includes, among other things, USB serial access, ethernet access, external clock boards, and breakout boards. The ZU48DR is a highly capable SoC catered towards radio communications applications [35]. The chip is equipped with a 64-bit quad-core and dual-core Arm® processing system (PS) combined with an UltraScale based programmable logic (PL). The RF data converter unit contains 8 ADCs and 8 DACs each with a dedicated digital down converter (DDC), and digital up converter (DUC).

Figure 3.14 shows the board and the physical connections used in this work.

Key features of the chip, along with how it will be used in the design will be explained in the following sections.

PYNQ Platform and PS - PL Integration

The combination of PS and PL is a powerful tool provided by the Zynq FPGA platform. In general the PL is used to perform intensive computation requiring dedicated hardware. The PS can then be used to build applications where users can provide input to the FPGA or view the output data provided by the FPGA. These interactions between PS and PL peripherals are facilitated by AXI interconnects.

This thesis work uses the PYNQ platform which runs on the Arm® PS to control the PL, perform data transfer to and from the PL, provide data visualization, and perform post-processing. PYNQ allows developers to combine the performance of the PL with the ease of use of Python to build interactive and high performance applications. In comparison to hardware focused software design tools such as Vitis, PYNQ is much more accessible to developers who do not initially have a background in embedded electronics.

PYNQ communicates with the PL through mainly the memory-mapped input-output (MMIO). This allows direct read/write access to a system memory mapped which corresponds to a particular register and address space within the PL. As such, drivers could be written in PYNQ which manipulates the behaviours of processing blocks in the form of Xilinx intellectual properties (IP) or register-transfer level (RTL) blocks. Within this thesis the MMIO is used to perform on-the-fly configuration of IP and RTL blocks and also to send enable or reset signals.

A more performance oriented data transfer method is also supported by the direct memory access (DMA). A more detailed description of the DMA will be given in a later section.

RF Data Converter

The RF Data Converter (RFDC) IP from Xilinx provides many core functionalities that is used to build a communications system. The ADCs, DACs, decimators, interpolators, digital down converters (DDC), and digital up converters (DUC) are all controlled by this IP block. In general, the block functions to take digital AXI4-Stream data from the PL and convert them to analog with an optional upconversion to the IF or RF. It can also perform the reverse operation and provide receive side functionalities. A block diagram showing the different components of the RFDC can be found in Figure 3.15.

In this work the RFDC takes a high frequency clock input with a reference frequency of 491.52 MHz. A PLL is used to convert the reference frequency to the desired hardware sampling frequency. Decimation and interpolation can then be done to convert the configured sampling frequency $f_{s,\text{conf}}$ to the desired effective sampling frequency f_s with the relation

$$f_s = f_{s,\text{conf}} \cdot I \tag{3.7}$$

where I is the interpolation factor.

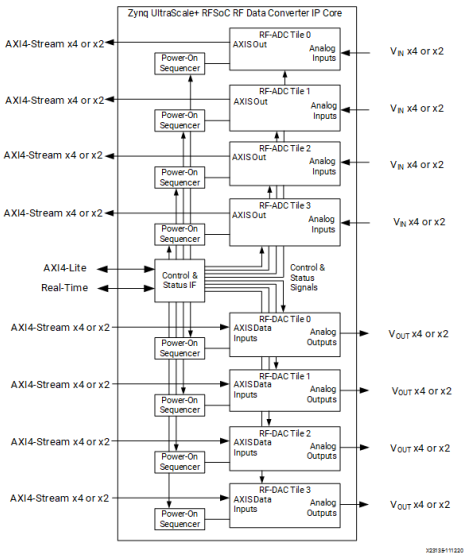

Figure 3.15: Architecture overview of the RFDC highlighting the different converters and accepted signals [1].

From (2.11) and the baseline parameterization from Section 3.1.3 the desired f_s can be found to be 122.88 MHz which means a valid configuration would be a configured sampling rate of 1966.08 Msps with an interpolation factor of 16. A detailed discussion on the effects of using a high order interpolation filters or having a too high sampling rate is outlined in [1], but is otherwise not discussed further in this thesis.

Direct Memory Access

As previously mentioned, the DMA is used when high performance or large data transfers is required between the PS and PL. This IP block facilitates direct access between the AXI4 memory mapped and AXI4-Stream IP interfaces [2]. Figure 3.16 shows a system level view of the DMA IP block.

To perform a transfer to and from the PS and the PL using PYNQ, a memory block must be allocated. In the PS side this takes the form of an array within Python stored in the distributed random access memory (DRAM) of the PS. Information about the memory's size and address are then sent through the DMA to the target IP block within the PS. A transfer from the PS to the PL means that data is read from the memory mapped from the AXI4 Memory Map Read port then sent to the IP block through the AXI4-Stream Memory Mapped to Stream, this process is facilitated by the read channel of the DMA. Conversely, an IP can send data to the PS by writing data to the AXI4-Stream to Memory Mapped which accesses the pre-allocated address spaces of the DRAM through the AXI4 Memory Map Write port.

In PYNQ, the allocated memory block can be manipulated like a regular

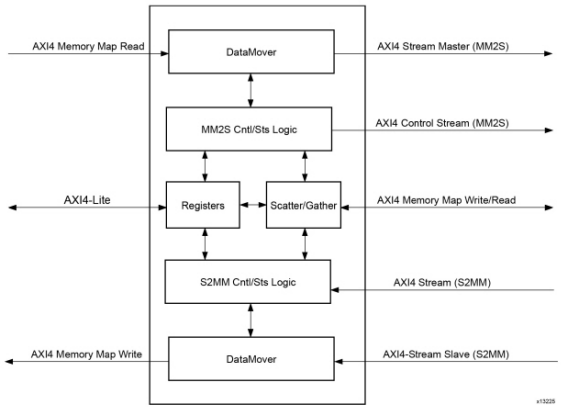

Figure 3.16: Architecture overview of the DMA highlighting the different input output relations [2].

Python array. Data transfer to and from the PS can be done by invoking the transfer command to the appropriate channel which moves data to or from the allocated buffer. This is especially useful when attempting to offload data from PS for visualization purposes.

Block Random Access Memory

Some design requires data to be temporarily stored within the PS, in this case a block random access memory (BRAM) can be used to store arbitrary data within the memory of the PS. The BRAM can be used to provide buffer between two processing blocks, provide clock domain crossing, and also to temporarily accumulate data to be processed later.

Storing data into a BRAM is done by specifying an address, the data, and some other configurations on how the data should be written. Reading from the BRAM also uses the same process of accessing a specific address of the BRAM primitive. This means the technicalities on the exact instance to start reading/writing or to which address data is read/written to is completely controllable down to the clock cycle through an RTL block.

PYNQ also provides functionalities to write and read to the BRAM from the PS using the MMIO. This method does not require an allocation of the PS DRAM since the data are being directly sent through the peripheral memory mapped to the corresponding BRAM address. For large amounts of data this method is not recommended since it will take a lot of resources on the shared AXI interface.

3.2.2 Sivers RF frontend

The Sivers RF frontend, specifically the EVK02004, is a highly capable module combining amplifiers, filters, and mixers into a compact form factor. This particular board supports 24.25–29.5 GHz, suitable for applications in the mmWave range. It can be operated in zero IF or low IF baseband. One of the defining feature is the

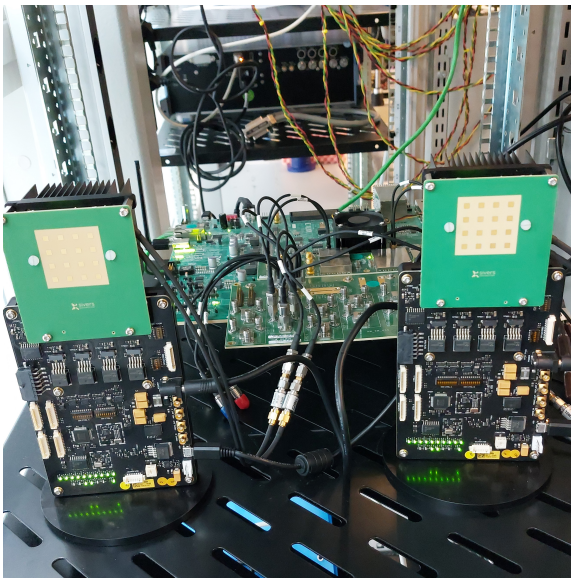

Figure 3.17: The RF frontend that are used in the project.

BFM02801 RF module which facilitates a highly configurable beamformer allowing the possibility to implement active electronically scanned array (AESA) radars.

Figure 3.17 shows the two frontends connected to the ZCU208 board. It is connected directly to the DAC and ADC which will be configured to output an IF signal.

3.3 Processing Block Implementation

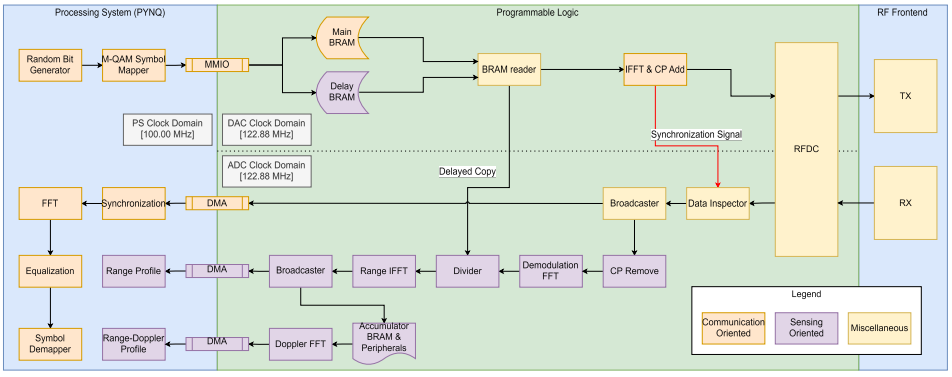

Figure 3.18: Block diagram of implemented components and its location within PS or PL.

The implemented design takes advantage of processing both in the PS and PL.

In this work it is decided to focus the hardware implementation for the sensing portion of the system since FPGA implementation of OFDM transceivers is already a well researched topic. Furthermore, the sensing operations shares many of the same blocks as the OFDM communication processing which leads to the possibility of hardware reuse.

The blocks as outlined in Section 3.1.2 is implemented as either Python code within PYNQ or IP/RTL blocks within the PL. Figure 3.18 shows the overall structure of the implemented design. As previously mentioned, most of the communication domain and visualization functionalities are handled within PYNQ. The processing blocks implemented within the PL are time sensitive and computation intensive in nature.

In the transmitter part, PYNQ generates the M-QAM symbols and transfers it into two BRAMs. The content of the main BRAM is read into the IFFT block which also adds the CP. The second BRAM will be used later to transfer the symbols into the receiver for division. The data is then sent to the RFDC to be converted to the analog domain, upconverted into the IF, and then sent to the RF frontend.

At the same time the receiver is put into a standby mode waiting for a synchronization signal to be sent from the transmitter structure. Upon receiving the synchronization signal, the data inspector passes data into a broadcaster which transmits two copies of the data. The first copy is directly offloaded using a DMA for communications processing according to methods described in Sec.2.1.2.

The second stream is used for radar processing starting with CP removal and the FFT to extract the received symbols that contains information about the channel. Next is the divider block which divides the received symbols with the transmitted symbol. The transmitted symbols are sent to the divider by reading the BRAM at a delay to compensate for the processing time of the transmitter IFFT, RFDC, and receiver FFT. Next an IFFT is taken to extract the range profile from a singular OFDM symbol which is, again, broadcasted into the DMA and the next set of processing blocks.

Each range profile output will be stored within the accumulator BRAM. In addition to storing M OFDM symbols, the accumulator also performs a transpose through manipulation of the order the addresses are read. The final operation within the PL is the FFT to extract the doppler information. The data is finally offloaded into the PS through a DMA.

A more detailed look at critical blocks will be given in the next sections.

3.3.1 Clock Domains

The implemented design can be divided into different clock domains. In Figure 3.18 the largest division of clock domains is between the PS where 100 MHz is used, and the PL where 122.88 MHz is used. There is a "soft" clock domain crossing between the transmitter structure and the receiver structure since their clocks are synchronized to the DAC and ADC respectively even though it is derived from the same frequency.

The source of these clocks can be seen in Figure 3.19. The CLK104 board generates a clock frequency of 245.76 MHz using the LMK04828 chip which also

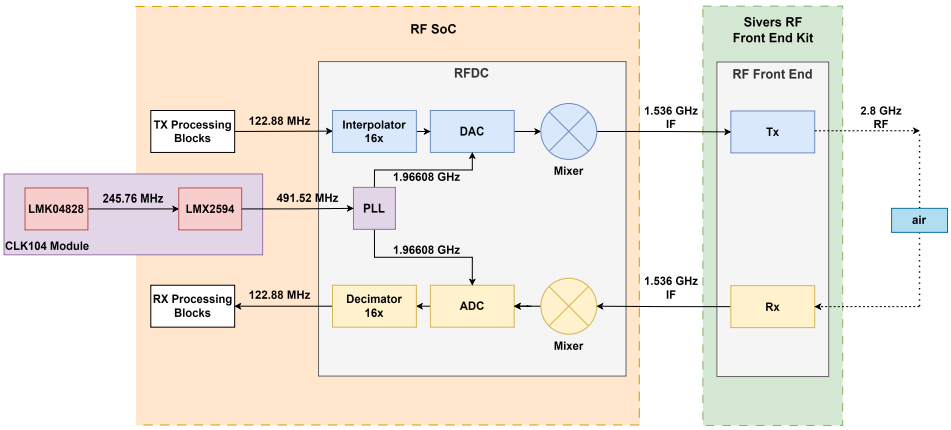

Figure 3.19: Clock domains and connections.

acts as a jitter cleaner. The 491.52 MHz frequency used for the input of the RFDC is generated by the LMX2594 chip using a PLL. Another PLL within the RFDC module creates the sampling clock of 1.96608 GHz and critically interpolating or decimating the samples by a factor of 16 generates the effective sampling rate of 122.88 according to (3.7).

3.3.2 Transmit BRAM Structure

The BRAM structure at the transmitter side is made up of two BRAM primitives, one controller for both, and one reader. The purpose of this processing block is to store data from the PS and control how it is being read to the IP/RTL blocks within the PL.

First the BRAM is connected to the PS through the AXI BRAM Controller Xilinx IP block which facilitates the data transfer between the PS to the BRAM through MMIO. The data here are branched into two different BRAMS, and as such the two BRAMs will contain identical data.

The second port of each BRAM is connected to an RTL block called "BRAM reader". The purpose of this block is to facilitate reading from the BRAM and outputting an AXI4-Stream to the next processing block. Reading from a BRAM is done through the following steps:

1. Send a high enable signal to the BRAM along with the address that wants to be read from.
2. In the next clock cycle receive the data from the BRAM and store into a temporary register.
3. Output the data into the next processing block.

Each step is repeated every clock cycle. In the primary reader output this process starts as soon as the RTL block receives an enable signal from PYNQ. When the reader runs out of addresses to read (it exceeds the numbers of OFDM symbols stored within the BRAM) then the address restarts from 0.

The read function for the secondary BRAM can be configured through PYNQ. Simply put the secondary output waits a certain number of clock cycles after receiving an enable signal before it starts reading. This is to accommodate the need for the secondary stream to output a delayed version of the symbols with gaps between each OFDM symbol corresponding to the length of the CP. Some other functionalities such as being able to read the addresses in reverse or at an offset is also added. A complete list of configurable variables can be found in Table 3.3

Variable	Bits	Description
Enable	1	Self explanatory
Delay	32	Delay, CP length, reverse, and offset
Num. Sample	17	Number of sample within OFDM symbol
Num. Symbol	13	Number of total symbols stored

Table 3.3: BRAM reader configurations accessible through PYNQ.

Through calibration testings it is found that it takes 4518 clock cycles delay so that the copy of transmitted symbols align with the received symbols in the divider block.

3.3.3 FFT/IFFT

In this design the FFT/IFFT Xilinx IP block is used to process data at four instances. This particular IP has the feature to add CP which is used in the transmitter IFFT. The IP block supports both fixed-point and floating-point input [36]. The block is configured in block floating-point mode when using fixed-point input to approximate the floating-point representation.

The IP block can be configured on-the-fly by sending configuration data through a DMA. The list of configurable variables are given in Table 3.4. In the table below, the number of actual useful bits are put in parentheses while the other number is the one with padding added to make a full bytes.

Variable	Bits	Description
Type	8(1)	Forward or inverse FFT
CP Length	16(8)	Self explanatory
Transform Length	8(4)	Number of FFT/IFFT points

Table 3.4: FFT/IFFT IP block configurations accessible through PYNQ.

This IP block is used once in the transmitter configured as an IFFT with maximum transform length of 1024 to correspond with the number of sub-carriers (active and guard bands) that is implemented within the system. The option to add CP is turned on for this instance and is set to add 72 samples in accordance to (3.3).

In the receiver the first instance of the block is used to demodulate the symbols. The configuration for this block is a forward FFT with the same length of 1024.

The CP is removed beforehand using an RTL block.

The next two instances appear after the divider block where the data type is a single-precision floating-point. As such the configured data type for the IP block is changed accordingly. The length of the range IFFT follows the number of sub-carriers. For the doppler FFT, the length is determined by how many OFDM symbols are observed. As mentioned in Section 3.1.3 more observed symbols will result in a better doppler resolution, however the latency will increase. Another hardware consideration is the required area for having a large order of FFT for the doppler.

3.3.4 Complex Divider

The complex divider is implemented based on (3.2). All of the required operations are handled using the Xilinx floating-point IP core. A detailed description of the IP block can be found in [37]. For numbers $a + bi$ and $c + di$ steps for the operation can be broken down as follows:

1. Convert a, b, c, and d into floating-point in parallel
2. Perform the multiplications of $a \cdot c$, $b \cdot d$, $b \cdot c$, $a \cdot d$, $c \cdot c$, and $d \cdot d$ in parallel.
3. Perform addition and subtraction to get $ac + bd$, $bc - ad$, and $c^2 + d^2$.
4. Replicate $c^2 + d^2$ to perform the final two division resulting in the real and imaginary part.
5. Replace NaN with 0.

Since there exist some guard-bands or sub-carriers that are intentionally set to 0 the divide operation will run into a division by 0. In this case the IP block will output an NaN. In [20] it is mentioned that the division with an inactive sub-carrier (a sub-carrier where no symbol is sent) can be considered to yield zeros instead of an invalid number. An RTL block is added to replace the NaN with 0.

3.3.5 Accumulator

The doppler detection is not possible without the use of the accumulator. The function of this block is to store the incoming range profiles into the appropriate addresses within the BRAM. Once enough symbols are stored, the block reads back the data in a transpose order to be fed into the doppler FFT. The accumulator uses a data trimmer, a BRAM writer, the actual BRAM, and a BRAM reader.

Saving all samples of the range profile is costly in the hardware. In addition, not all range bins will be useful for detection purposes. After a certain range the back-scattered signal will be practically undetectable due to losses, noise, and interference. A data trimmer is implemented to crop the range profile to the region of interest. Theoretically it can also be used to "scan" different regions of interest based at different observation periods.

The cropped data is then stored in the BRAM through the usage of an RTL BRAM writer. Data comes into the BRAM writer in increasing sub-carrier index before moving to the next OFDM symbol. This is different from the needs of the

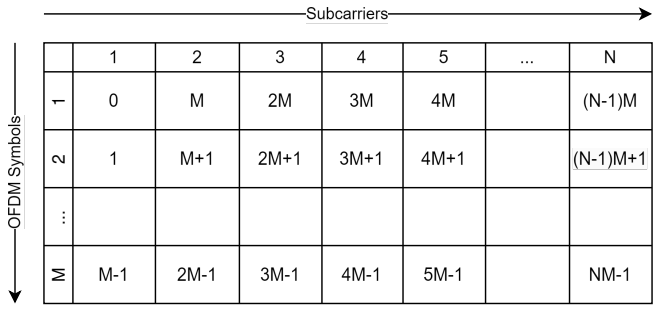

Figure 3.20: Addressing within the accumulator BRAM.

final doppler FFT where the same sub-carrier of all of the observed OFDM symbols is processed as a block.

The strategy is to write into the BRAM in a non-sequential manner as illustrated in Figure 3.20. Now if the BRAM is read sequentially, the data will be rearranged (transposed in comparison to the initial input) resulting in a block containing all of the sub-carriers of all OFDM symbols. The data can then be processed by the final FFT which gives the ultimate range-doppler profile.

Chapter 4
Testing and Verification

This chapter will present the measurement results to verify the functionality and performance of the designed system. In the second part of the chapter some post-processing methods are used to attempt to isolate the target and reduce the effects of the self-interference.

4.1 Measurements

4.1.1 Processing Time

The goal of implementing processing on an FPGA is to achieve higher computing performance through the usage of dedicated hardware. To measure the gain from using dedicated hardware a measurement is done on the processing time of two systems. In the first system all of the processing blocks are implemented within PYNQ, including the sensing ones. The second system is the system as designed in Section 3.3.

To perform a fair comparison, the starting time for both system is defined to be the point the data comes into the IFFT block in the transmitter. The timing ends when the range-doppler plot is generated. The parameters of both systems are ensured to be the same.

PYNQ	PL	Acceleration Factor
27.56 ms (0.40)	3.81 ms (0.05)	7.23

Table 4.1: Average PYNQ and PL execution time with their Acceleration Factors.

From Figure 4.1 it can be observed that the PL implementation takes significantly less time compared to the fully PYNQ implementation. The statistics of the averaged measured execution times can be found in Table 4.1, the number in parentheses represent the standard deviation of each implementation. The actual runtime within the PL is in actually lower than the measured time as these measurements include the time needed to transfer data between the PS and PL. In that sense the actual hardware acceleration factor is actually higher if only the actual processing is considered.

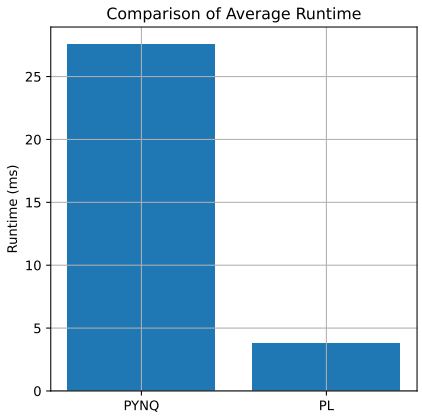

Figure 4.1: A comparison of the average execution time of the fully PYNQ implementation and the PL implementation.

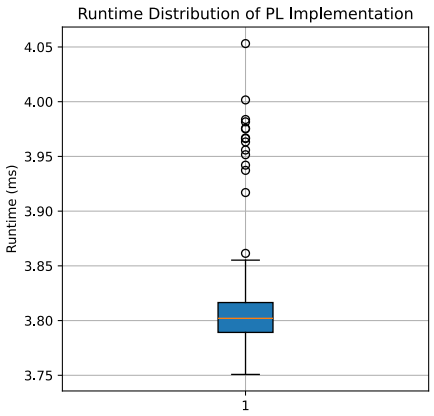

Figure 4.2: Distribution of execution times for the PL implementation.

Parameter	Value
Carrier frequency	28 GHz
ADC/DAC sampling rate	1966.08 Msps
Interpolation/Decimation	16x
Effective sampling rate	122.88 Msps
SCS	120 kHz
BW	100 MHz
N-FFT	1024
Active sub-carriers	792
Range bins	1024 (cropped to 32)
Doppler bins	64

Table 4.2: Parameter values for field testing

Figure 4.2 gives insight to how the execution times are distributed for the PL implementation of the ISAC system. Overall it can be noted that most of the outliers are clustered together, around 0.1 ms above the mean value. The cluster of outliers indicate a processing delay/bottleneck that is only triggered on some of the runs. It is possible that the bottleneck causing the outliers is the relative inconsistency of the DMA transfer between the PL to the PS.

4.1.2 Sensing Performance

The performance of the sensing is evaluated by field testing the implemented design. The system parameters used for the purpose of this test are listed within Table 4.2. In summary, a range resolution of around 1.2 m and a doppler resolution of 4.86 m/s is realized.

First, a measurement of the environment is taken to establish a baseline snapshot. The testing environment can be seen in Figure 4.3. Three main features are the cluster of chairs, the glass wall with metal framing, and the wooden wall in the background. The objects are considered to be static.

Figure 4.4 shows the range-doppler profile of the testing environment. The three landmarks are clearly visible as clearly marked by the circles with colors corresponding to the real world object. One extra target is visible at 30 meters indicating the back wall of the room behind the wooden wall. This shows that the radar can be used even for zero line-of-sight detection of targets that are fully obscured by solid objects.

Note the large peaks around the 0 doppler and 0 range area. This is caused by the self interference that happens between the transmit and receive antennas. To minimize its effects the transmit and receive antennas are separated by a distance of 30 cm. A hardware focused solution is discussed in [38] where an RF filter is used to cancel out the effects of the interference from the transmit antenna. An algorithmic approach is explored in a later section of this work.

In the second test a large aluminum target is placed 4.8 m away from the antenna module as shown in Figure 4.5. This object is detectable as can be seen on Figure 4.6 with a large peak at around 5 m.

4.1.3 Communication Performance

For simplicity, the communication portion is verified by decoding the back-scattered signal received at the transceiver module. In an ideal scenario it would be better to test the communication domain by placing the transmit and receive module at a given distance from each other. For the purposes of verifying that communication can be done use the same waveform this test is deemed to be satisfactory.

Figures 4.7 and 4.8 show the results of tests using 4-QAM and 16-QAM respectively. The constellation shows that there are no visible large scale phase shifts from failed equalization or uncorrected time/frequency offsets. It can be seen that the SNR is low indicated by the spreading of the constellation diagram. The signal power in these tests is low and only the back-scattered signal is used. The reflector is an aluminum sheet with thickness of 3 mm placed at 5 m, the surface is

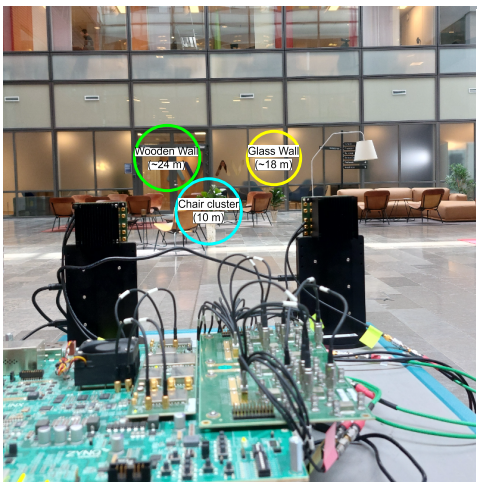

Figure 4.3: Baseline condition of the test environment without additional targets.

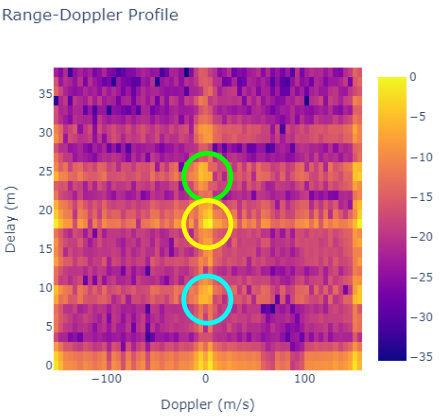

Figure 4.4: Range-doppler profile for the background environment.

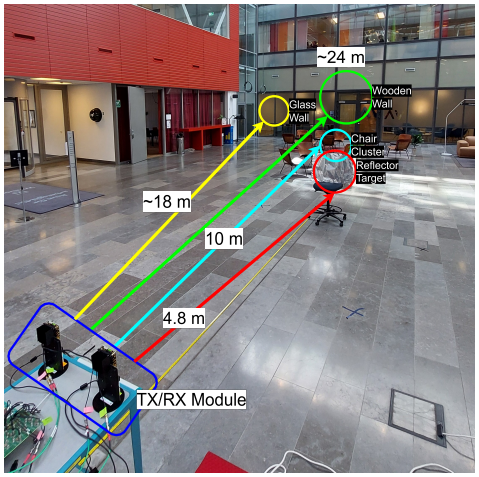

Figure 4.5: Test environment with a reflector as a target at 4.8 m.

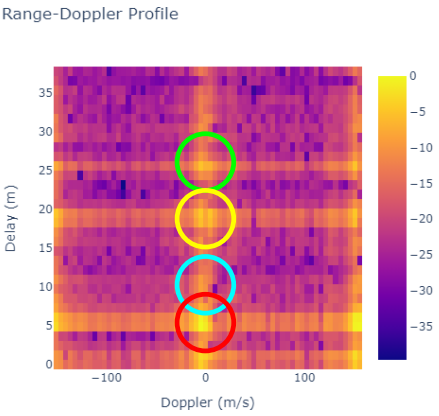

Figure 4.6: Range-doppler profile for the reflector target.

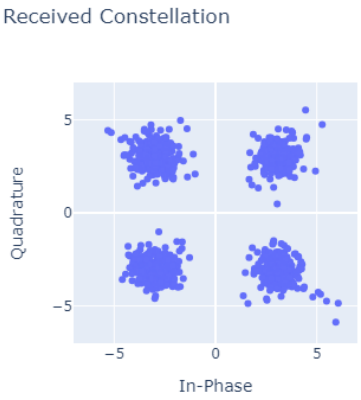

Figure 4.7: Communication test using 4-QAM.

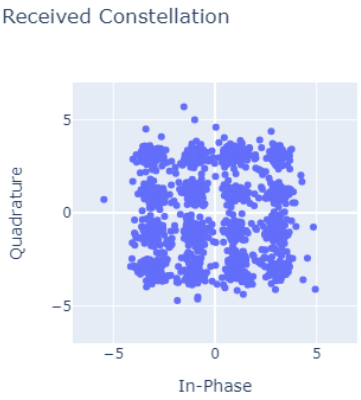

Figure 4.8: Communication test using 16-QAM.

polished but not flat. In spite of that, an error-less communication is consistently achieved using 4-QAM while the 16-QAM averages at a BER of 0.014.

It must be pointed out that the system is uncoded. Using channel code such as LDPC or exploiting MIMO diversity can greatly increase the reliability of the communication portion.

4.2 Analysis and Post-processing

For cases where the sensing unit is stationary it is useful to be able to disregard the stationary environment's effects on the range-doppler profile. A simple post-processing technique that can be used is cancellation of the range-doppler profile using a calibration reading of only the environment without any objects. For instance if sensing is performed by an indoor base station with the aim of detecting people then measurements can be taken when the room is empty.

The simplest way of performing the cancellation is through background subtraction [39]. In the dB domain this translates to subtracting the range-doppler profile with the calibration measurements results. One of the main drawbacks of using this method on the data is the noise enhancement problem. background subtraction minimizes the interference (in this case the artefacts from the background environment), however this means that noise will be enhanced in areas where no object is present.

Figure 4.10 shows how background subtraction can be applied on the range-profile data. In this case the measurement data containing the target shown in Figure 4.9 is subtracted by the data collected of only the background objects. The result shows a much more noticeable peak around the object of interest.

More advanced processing algorithms such as Kalman filtering could be applied on the output range-doppler profile to perform tracking [40]. The implemented

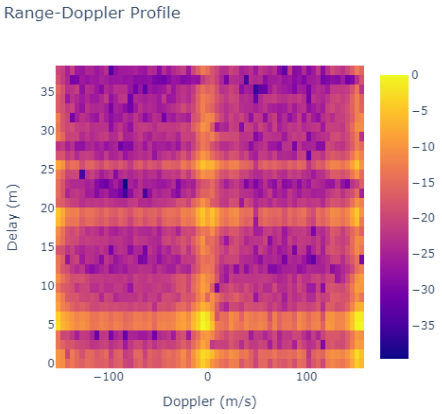

Figure 4.9: Raw measurement result.

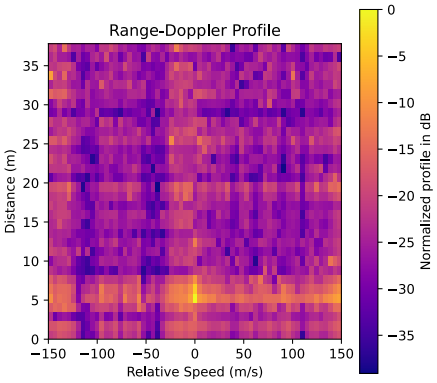

Figure 4.10: Result of background subtraction processing

design could also be used to harvest data to train AI models for object recognition.

Chapter 5
Conclusion and Future Work

5.1 Conclusion

This thesis work evaluates the usability of 5G NR waveforms for sensing purposes. Literature review and simulations shows that many processing algorithms are suitable to perform sensing using OFDM symbols. The 5G NR standards provides degrees of freedom which are not previously available in the LTE standard to design OFDM radars of varying capabilities. The FR2 mmWave band provides opportunities to exploit the larger value of doppler shift to achieve a more specific velocity resolution. Larger bandwidths also result in greater range resolution which is desired for a sensing system.

The symbol domain processing method is compared with the match filter algorithm. Numerical and simulation results shows that the symbol domain processing is more suited for ISAC application. The match filter algorithm results in a comparatively low PSR due to internal correlations within the data. The symbol domain processing method does result in a more intensive computation requirement due to the presence of the complex number division operation. The implemented radar processing blocks use many similar blocks as communication which means theoretically a sharing mechanism can be implemented using existing functionalities within OFDM communication chip sets.

The system is implemented on an FPGA. The processing is split with the communication portion happening mainly within the PS and the radar functionalities performed within the PL. Testing shows that performing the range-doppler calculation within the PL is more than 7 times faster than performing it entirely on the PS. Field testing of the implemented design verifies that the system functions as expected. A simple post-processing method is implemented by performing background subtraction on the measured data using a calibration measurement. The method is proven to work though it faces the noise enhancement problem.

5.2 Future Work

One of the avenue to progress the research in this topic is to perform optimizations on the hardware implementation side. This work uses naive implementation methods which can be sub-optimal both in the processing time, required area, or data accuracy. Optimization of the hardware will lead to more efficient systems

which will be more applicable in real world scenarios. An interesting topic to pursue is how to perform hardware sharing between communication and sensing domain. A comparison could be made on the energy consumption of a separated communication and sensing system with an ISAC system as designed in this work. For instance, reusing the FFT and IFFT block by performing scheduling on the incoming data can lead to very significant reduction of needed resources in both chip area and energy consumption.

Another aspect of sensing not touched upon within this work is angle of arrival detection. The usage of two dimensional arrays can be used to detect the incoming angles for complete spatial awareness. Algorithms such as MUSIC or SAMV can be implemented on the FPGA to truly integrate the complete sensing system.

Performing post-processing on the acquired sensing data will also lead to a better sensing performance. A more intensive measurement campaign can lead to extensive datasets which will be useful to train AI models. Interesting concepts such as applying hardware accelerated machine learning models on the FPGA would greatly enhance the value of the research in this topic.

References

[1] *Zynq UltraScale+ RFSoC RF Data Converter v2.6 Gen 1/2/3/DFE LogiCORE IP Product Guide*, [PDF document], AMD-Xilinx, 2024, accessed May 4th, 2024. [Online]. Available: https://docs.amd.com/r/en-US/pg269-rf-data-converter/Overview?tocId=r_g45~I2cOX2wmwgcHJSnA

[2] *AXI DMA LogiCORE IP Product Guide*, [PDF document], AMD-Xilinx, 2022, accessed May 10th, 2024. [Online]. Available: https://docs.amd.com/r/en-US/pg021_axi_dma/Features

[3] H.-X. Consortium, "Hexa-X Deliverable D3.1: Localisation and Sensing Use Cases and Gap Analysis," Hexa-X, Tech. Rep., June 2021. [Online]. Available: https://hexa-x.eu/deliverables/

[4] ——, "Hexa-X Deliverable D3.2: Initial Models and Measurements for Localisation and Sensing," Hexa-X, Tech. Rep., December 2022. [Online]. Available: https://hexa-x.eu/deliverables/

[5] ——, "Hexa-X Deliverable D3.3: Final Models and Measurements for Localisation and Sensing," Hexa-X, Tech. Rep., June 2023. [Online]. Available: https://hexa-x.eu/deliverables/

[6] C. Aydogdu, H. Wymeersch, and M. Rydström, "Can automotive radars form vehicular networks?" in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–6.

[7] S. N. Sur, P. Sharma, H. Saikia, S. Banerjee, and A. K. Singh, "Ofdm based radar-communication system development," *Procedia Computer Science*, vol. 171, pp. 2252–2260, 2020, third International Conference on Computing and Network Communications (CoCoNet'19). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050920312357

[8] J. B. Sanson, D. Castanheira, A. Gameiro, and P. P. Monteiro, "Fusion of radar and communication information for tracking in ofdm automotive radar at 24 ghz," in *2020 IEEE/MTT-S International Microwave Symposium (IMS)*, 2020, pp. 1153–1156.

[9] Y. Liu, G. Y. Liang, D. Garmatyuk, and Y. T. J. Morton, "Usrp based ofdm radar systems for doorway detection," in *2014 IEEE Radar Conference*, 2014, pp. 0875–0880.

[10] M. Tang, D. Xu, W. Xu, H. Wang, X. Li, and L. Qi, "Design and implementation of lfmcw radar target detection system based on zynq," in *2022 IEEE 10th Asia-Pacific Conference on Antennas and Propagation (APCAP)*, 2022, pp. 1–2.

[11] S. B. Weinstein, "The history of orthogonal frequency-division multiplexing," *IEEE Communications Magazine*, vol. 47, no. 11, pp. 26–35, 2009.

[12] H. Sari, G. Karam, and I. Jeanclaude, "An analysis of orthogonal frequency-division multiplexing for mobile radio applications," in *Proceedings of IEEE Vehicular Technology Conference (VTC)*, 1994, pp. 1635–1639 vol.3.

[13] G. Lindell, "An introduction to ofdm," [PDF document], 2016. [Online]. Available: https://www.eit.lth.se/fileadmin/eit/courses/ettn01/HT2-2017_Rusek/OFDM_lecture_notes_161115.pdf

[14] T. Schmidl and D. Cox, "Robust frequency and timing synchronization for ofdm," *IEEE Transactions on Communications*, vol. 45, no. 12, pp. 1613–1621, 1997.

[15] S. Huang, Y. Su, Y. He, and S. Tang, "Joint time and frequency offset estimation in lte downlink," in *7th International Conference on Communications and Networking in China*, 2012, pp. 394–398.

[16] G. Lindell, *Introduction to digital communications : (including basics in analog communications)*. Lund: Electrical and Information Technology, Lund University, 2006.

[17] M. Sadiku, *Elements of electromagnetics*. New York: Oxford University Press, 2015.

[18] A. F. Molisch, *Wireless Communications*, 2nd ed. Wiley Publishing, 2011.

[19] R. Fens, M. Ruggiano, and G. Leus, "Channel characterization using radar for transmission of communication signals," in *2008 European Conference on Wireless Technology*, 2008, pp. 127–130.

[20] K. M. Braun, "Ofdm radar algorithms in mobile communication networks," Ph.D. dissertation, Karlsruhe Institute of Technology, 2014.

[21] N. Levanon, "Multifrequency radar signals," in *Record of the IEEE 2000 International Radar Conference [Cat. No. 00CH37037]*, 2000, pp. 683–688.

[22] D. Garmatyuk, J. Schuerger, Y. T. Morton, K. Binns, M. Durbin, and J. Kimani, "Feasibility study of a multi-carrier dual-use imaging radar and communication system," in *2007 European Radar Conference*, 2007, pp. 194–197.

[23] R. F. Tigrek, W. J. A. de Heij, and P. van Genderen, "Multi-carrier radar waveform schemes for range and doppler processing," in *2009 IEEE Radar Conference*, 2009, pp. 1–5.

[24] ——, "Solving doppler ambiguity by doppler sensitive pulse compression using multi-carrier waveform," in *2008 European Radar Conference*, 2008, pp. 72–75.

[25] C. R. Berger, S. Zhou, P. Willett, B. Demissie, and J. Heckenbach, "Compressed sensing for ofdm/mimo radar," in *2008 42nd Asilomar Conference on Signals, Systems and Computers*, 2008, pp. 213–217.

[26] C. Sturm and W. Wiesbeck, "Waveform design and signal processing aspects for fusion of wireless communications and radar sensing," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1236–1259, 2011.

[27] C. Sturm, T. Zwick, and W. Wiesbeck, "An ofdm system concept for joint radar and communications operations," in *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*, 2009, pp. 1–5.

[28] C. A. Sturm, "Gemeinsame realisierung von radar-sensorik und funkkommunikation mit ofdm-signalen," Ph.D. dissertation, Karlsruhe Institute of Technology, 2012.

[29] K. D. Rao, C. Gangadhar, and P. K. Korrai, "Fpga implementation of complex multiplier using minimum delay vedic real multiplier architecture," in *2016 IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics Engineering (UPCON)*, 2016, pp. 580–584.

[30] 3rd Generation Partnership Project (3GPP), *3GPP TS 38.101-2 V18.4.0 (2023-12) – NR; User Equipment (UE) radio transmission and reception; Part 2: Range 2 Standalone*, 3GPP Std., December 2023, version 18.4.0.

[31] ——, *3GPP TS 38.211 V18.1.0 (2023-12) – NR; Physical channels and modulation*, 3GPP Std., December 2023, version 18.1.0.

[32] C. Munnell, R. Mattingly, S. Flandermeyer, and J. G. Metcalf, "On the practical use and experimentation of lte signals for radar-communications," in *2022 IEEE Radar Conference (RadarConf22)*, 2022, pp. 1–6.

[33] M. Braun, C. Sturm, A. Niethammer, and F. K. Jondral, "Parametrization of joint ofdm-based radar and communication systems for vehicular applications," in *2009 IEEE 20th International Symposium on Personal, Indoor and Mobile Radio Communications*, 2009, pp. 3020–3024.

[34] AMD-Xilinx, "Product brief: Zynq ultrascale+ rfsoc zcu208 evaluation kit," [PDF document], 2021, accessed June 2nd, 2024. [Online]. Available: https://www.xilinx.com/content/dam/xilinx/publications/product-briefs/xilinx-zcu208-product-brief.pdf

[35] *Zynq UltraScale+ RFSoC Data Sheet: Overview*, [PDF document], AMD-Xilinx, 2023, accessed June 4th, 2024. [Online]. Available: https://www.xilinx.com/content/dam/xilinx/publications/product-briefs/xilinx-zcu208-product-brief.pdf

[36] *Fast Fourier Transform v9.1 Product Guide*, [PDF document], AMD-Xilinx, 2022, accessed May 27th, 2024. [Online]. Available: https://www.xilinx.com/support/documents/ip_documentation/xfft/v9_1/pg109-xfft.pdf

[37] *Floating-Point Operator v7.1 Product Guide*, [PDF document], AMD-Xilinx, 2020, accessed June 1st, 2024. [Online]. Available: https://docs.amd.com/v/u/en-US/pg060-floating-point

[38] C. Baquero Barneto, T. Riihonen, M. Turunen, L. Anttila, M. Fleischer, K. Stadius, J. Ryynänen, and M. Valkama, "Full-duplex ofdm radar with lte and 5g nr waveforms: Challenges, solutions, and measurements," *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 10, pp. 4042–4054, 2019.

[39] R. Rollakanti, T. Arulananth, S. Prasad, J. Anand, B. Ravi, and M. Polisetti, "Moving object detection using background subtraction," in *2022 International Conference on Advancements in Smart, Secure and Intelligent Computing (ASSIC)*, 2022, pp. 1–5.

[40] A. Macaveiu and A. Câmpeanu, "Automotive radar target tracking by kalman filtering," in *2013 11th International Conference on Telecommunications in Modern Satellite, Cable and Broadcasting Services (TELSIKS)*, vol. 02, 2013, pp. 553–556.

Appendix A

Simulation Configuration and Parameterization

No.	μ	BW (MHz)	M	$d_{\max}$ (m)	$v_{\max}$ (m/s)	d_{unamb} (m)
1	2	100	128	175.6	32.1	2498.3
2	3	50	128	87.8	64.2	1249.1
3	3	100	128	87.8	64.2	1249.1
4	3	200	128	87.8	64.2	1249.1
5	3	400	128	87.8	64.2	1249.1
6	3	100	256	87.8	64.2	1249.1
7	3	100	512	87.8	64.2	1249.1
8	3	400	512	87.8	64.2	1249.1

No.	v_{unamb} (m/s)	Δd (m)	Δv (m/s)
1	150.0	1.58	2.36
2	300.1	3.15	4.73
3	300.1	1.58	4.73
4	300.1	0.79	4.73
5	300.1	0.39	4.73
6	300.1	1.58	2.35
7	300.1	1.58	1.17
8	300.1	0.39	1.17

Table A.1: Relation between parameter configuration and detection performance.