3D Robotic Ultrasound System

Technische Universiteit Delft

Joshua Teye

TU Delft

3D Robotic Ultrasound System

A 3D Volumetric Ultrasound Imaging System using a CMUT Phased Array and a Robotic Arm

by

Joshua Teye

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on August 27 2020 at 1:30 PM.

Student number:	4779932	
Project duration:	February 1, 2020 – July 31, 2020	
Thesis committee:	Prof. dr. ir. R. Dekker,	TU Delft, supervisor
	dr. C. van Heesch,	Philips Research, supervisor
	Prof. dr. B. Hendriks,	TU Delft
	Prof. dr. A. Theuwissen,	TU Delft

Preface

As I draw the curtains on what would forever remain a seminal epoch in my life, it is expedient that I take a moment to reflect and appreciate individuals without whose help I wouldn't be in a position to indite this thesis. This thesis forms the final component towards my graduation from the Delft University of Technology. It has been two years of a thorough learning experience and a highlight of my strengths and weaknesses. A challenge to move outside my comfort zone and develop new skills to ensure the successful completion of the project. It was important and pleasing that the project sought to address a key medical challenge faced by people globally. It is my hope that this work serves as a springboard for future designs.

I would like to express my warmest appreciation to Ronald Dekker for the opportunity he afforded me to intern with Philips Research within the department of Microsystems and Devices. Thank for your guidance and motivation and most importantly for giving me a masterclass on how to make bitterballen. My next thanks goes to Chris van Heesch who supervised my thesis project. Thank you for always being available and creating enough room for me to explore my creativity and ideas. Also, I would like to thank Frank van Heesch for his support and guidance during the project. I enjoyed our discussion a lot. I would also like to thank the TU Delft Global Initiative for the Sub-Saharan Excellence Scholarship which gave me the opportunity to study at a world-class university. Finally to all researchers within the Department of Microsystems and Devices, friends, colleagues and family, I say a very big thank you for your unwavering support in diverse ways.

Joshua Teye
Eindhoven, July 2020

Contents

List of Figures

Abstract

Diagnostic imaging is a fundamental requirement for the effective treatment of about 25 % of patients globally. Ultrasound imaging is considered the safest, least expensive and most convenient diagnostic imaging modality. Its widespread adoption has been in the area of 2D imaging. However, the accuracy of diagnosis with 2D ultrasound imaging is highly dependent on the experience and expertise of the sonographer because the sonographer has to mentally create a 3D impression from multiple 2D images of the region of interest and this could lead to erroneous diagnosis. 3D ultrasound imaging addresses this concern, however, it drives up the cost of ultrasound imaging. Also, disadvantages such as difficulty in taking quality images, prolonged time in learning how to effectively take these images and human distress during the acquisition of these images have been reported. By leveraging the high level of precision, accuracy and maneuverability provided by robotic systems in conjunction with low-end ultrasound imaging platforms, 3D ultrasound images can be captured, the problem of human distress alleviated and subsequently reducing the cost of 3D ultrasound imaging. This project sought to explore the feasibility of designing a low-cost 3D ultrasound robotic system. The design followed a distributed system approach using ROS (Robotic Operating System) where the data acquisition decoupled from the processing and visualization unit. The performance of the design was tested by constructing 3D ultrasound images of custom made phantoms. These results were compared with 3D ultrasound images of the phantoms acquired using the Philips EPIQ 7 - a high-tier ultrasound machine. The results obtained with this design had a low resolution in comparison with those obtained with the Philips EPIQ 7 however, a 3D point cloud representation of the object embedded in the phantom can be seen. Also, a full 3D image of the phantom could not be acquired due the transducer movement limitations of the design. This design successfully demonstrates the feasibility of integrating low-end electronics with robotic systems to acquire 3D ultrasound images.

1

Introduction

1.1. Ultrasound Imaging

Ultrasound (US) Imaging is an integral part of clinical imaging. Other imaging modalities such as X-ray, magnetic resonance imaging (MRI) and computed tomography (CT) serve similar purposes in clinical imaging, but compared to US imaging, it is safer, less expensive and more convenient. Additionally, US obviates the exposure to radiations. These advantages have contributed to the increased attention US has received in recent years [1]. US are sound waves (vibrations) with frequencies above the range of human hearing ($> 20\ kHz$). These waves have two main clinical applications: imaging and therapy. The application of US in clinical imaging can be further classified into two categories: diagnosis and guidance. US imaging provides valuable information to gain insight into the anatomy of the human body to aid in the diagnosis of disease conditions. It has however, disadvantages, which include the difficulty in taking quality images and interpreting these images and also the prolonged time it takes to effectively learn how to take good US images. Intraopearative US allows surgeons to have a real-time visualization of soft tissue anatomy during surgical procedures [2]. US-guided robotic systems have been reported to minimize error bias and also minimize the degree of invasiveness in comparison to conventional needle insertion techniques [3, 4]. Therapeutic US entails the deposition of energy in a tissue to incite biological effects [5]. Therapeutic US has been employed to effectively treat Meniere's disease by destroying the vestibular nerve, and the treatment of Parkinson's disease by directing US beams to a specific tissue in the brain [5]. Other applications of therapeutic US include high-intensity focused ultrasound (HIFU), lithotripsy, drug delivery, hemostasis and thrombolysis. As promising as this application may sound, majority of the world's population mostly in rural and remote areas lack access to US imaging mainly due to cost and the lack of skilled sonographers [6]. This is a major problem because diagnostic imaging is a fundamental requirement for the effective treatment of 25 % of patients globally [7]. 2D US imaging has been widely used for diagnostic purposes because it can generate 2D images of the region of interest (ROI) in real-time [8, 9]. However, the accuracy of diagnosis for example, is highly dependent on the experience and expertise of the sonographer or the clinician [10]. This is because the human anatomy has a 3D representation and the clinician has to mentally create a 3D impression of the 2D US image [10]. This strategy of mentally transforming 2D images into 3D can be time-consuming, less efficient and decisions made on this approach have the tendency to be subjected to the expertise and experience of the sonographer which can lead to erroneous diagnosis and misleading information for surgical planning and the delivery of therapy [10]. Additionally, most diagnostic decisions are based on the assessment of an organ or tumor volume. 2D US imaging techniques calculate the volume by measuring the height, width and length in only two orthogonal views. This calculation assumes the organ or tumor under consideration as an ideal shape and this is likely to lead to inaccurate conclusions [10]. Also, 2D US imaging is less favourable for monitoring therapeutic procedures or performing progressive examination of a patient's response to treatment. The reason being the difficulty in determining the exact positioning of the transducer probe to obtain a similar 2D plane of the anatomical site in comparison to previous examinations [10]. These challenges have stimulated the development of 3D US imaging. 3D US imaging uses computational tools to integrate 2D images to generate a 3D image of the ROI. 3D US imaging has the potential to address the above mentioned challenges by providing clinicians the opportunity to gain a better understanding of the spatial anatomic relationship [1]. Additionally, real-time 3D US imaging improves estimation accuracy by providing a volumetric representation of the ROI, and it alleviates the dependency on the clinician's expertise in interpreting 2D US images as 3D, thereby decreasing diagnosis variability. 3D US imaging enables clinicians make diagnosis quickly, therefore eliminating the time spent in mentally creating a 3D representation from a 2D image and it also fosters easy coordination with the clinician during imaging [1]. Inasmuch as these advantages are desirable, 3D US is expensive and it further increases the cost of imaging which is already not accessible to people in rural and remote areas. There is therefore a need to explore and design alternative 3D ultrasound imaging systems to reduce the cost of US imaging while not compromising the quality and safety of these systems.

1.2. Ultrasound Robotic Systems

The exploration of robotics in ultrasound applications begun two decades ago and it was fostered by the advantages of robotic manipulators [11]. By using robotic manipulators in the acquisition of US images, a level of accuracy, precision, maneuverability and dexterity that cannot be realised with conventional ultrasound imaging techniques can be achieved [11]. Additionally, robots can effectively handle US probes and trace the probes with a high level of precision that can lead to the generation of uniformly spaced slides. This enables the generation of three-dimensional (3D) US imaging that is superior to 2D conventional techniques because 3D US volume construction involves the integration of 2D images taken from different planes of the region of interest (ROI) with precisely known location and orientation [11, 12]. Robotic systems have the ability to precisely track these locations and orientations. Robotic systems unlike human beings, do not suffer human distress such as hand tremor which can result in the fluctuation of forces applied during US imaging and laxity in concentration [11]. These advantages have the potential of improving imaging acquisition and to spur new applications of both real-time and non real-time ultrasound imaging. US robotic systems can also be guided by freehand and still achieve an appreciable level of precision. Another application of US robotic systems has to do with telesonography. With telesonography, a remote telesonographer performs the examination by using robotic systems to manipulate the US probe [11]. To effectively perform sonography, a specialized set of skills such as probe positioning, orientation, and image interpretation is required. Also, sonographers visualize 3D anatomy by mentally integrating 2D images. These skill-sets are not in abundant supply, especially the number of trained sonographers are in short supply in remote locations [13]. This gives importance to the area of telesonography as diagnostic imaging is a fundamental requirement for the effective treatment of about 25 % of patients globally [7]. These systems have a master-slave configuration, the slave-side imitates the movement of the master-side. The telesonographer controls the master-side and these actions are translated to the slave-side. These telesonograhic robotic systems have been primarily designed for diagnostic purposes, however, as promising as their prospects are, the amount of research on their effectiveness and long-term significance is limited [7].

1.3. Focus of the Project

This work focuses on the design of a low-end 3D ultrasound imaging system using Capacitive Micromachined Ultrasound Transducers (CMUTs), a robotic arm and an electronic platform designed by Philips Research for developing CMUT applications. Furthermore, packages which enable the real-time visualization of 3D ultrasound volumes using the Robotic Operating System (ROS) is proposed. The main contributions of this work are stated below

- ROS packages (a collection of scripts) that support the distributed acquisition and processing of ultrasound signals.
- A prototype that acquires and displays real-time ultrasound point cloud volumes using a CMUT phased array.

1.4. Thesis Organization

- **Chapter 2** provides background information on the use of robotic arms in ultrasound imaging. Relevant studies reported in literature on the three main application areas - guidance, diagnosis and therapy are discussed.
- **Chapter 3** describes the design concept. Various components that make up the entire system are highlighted and their respective functions are defined. Finally, The mode of operation of the entire system is described.
- **Chapter 4** begins by showing and explaining how the prototype works. The results from signal acquisition to the creation of a point cloud volume are presented.
- **Chapter 5** discusses the results, challenges and the limitations of the prototype.
- **Chapter 6** concludes the thesis with reflections on the relevance of the project and recommendations for future designs.

2

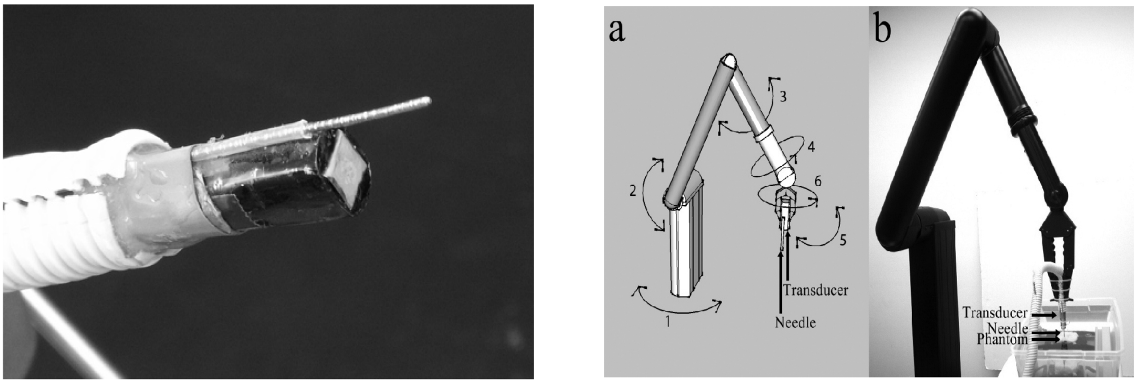
Background Information

A literature review with a focus on robotic systems in ultrasound imaging has been conducted to highlight their clinical relevance and impact on healthcare delivery. The results of the review have been segregated into the three widespread application areas of robotic systems in ultrasound: guidance, diagnosis and therapy.

2.1. Guidance

Liang *et al.* [14] developed an autonomous multiple core biopsy system guided by real-time 3D US and operated by a robotic arm with 6 + 1 DOF (Degree of Freedom). The system incorporates the iARM assistive robotic manipulator (Exact Dynamics B.V, Arnhem, The Netherlands) with a biopsy needle and a US probe. The tip of the biopsy needle is attached to the needle on the probe as shown in figure 2.1a [14]. The robot has three Cartesian coordinates and the three coordinates determine the orientation (yaw, pitch and roll) of the robot's gripper as shown in figure 2.1b [14]. The robot's gripper holds the transducer probe and is firmly secured by cable connections to ensure that the needle does not diverge during biopsy. The US probe and the needle function like a clinical end firing US probe as mostly observed in trans-rectal US-guided (TRUS) biopsy. Turkey breast models were used as tissue models to test the system. The system was able to locate the phantom autonomously and insert the needle rods for biopsy without human intervention. The results showed a success rate of about 93 % which translates into 3 misses out of 40 attempts. The robotic system and the experimental setup are shown in figure 2.1 [14].

Sen *et al.* [15] designed a cooperatively controlled robotic system for US-guided radiotherapy. Radiotherapy utilizes ionizing radiation and this makes it unsafe for a clinician to hold the US probe during the treatment session. The robotic system was designed to enhance patient setup and monitor the intended target during radiotherapy. It has a force sensor that monitors and controls the applied force between the US probe and the patient.

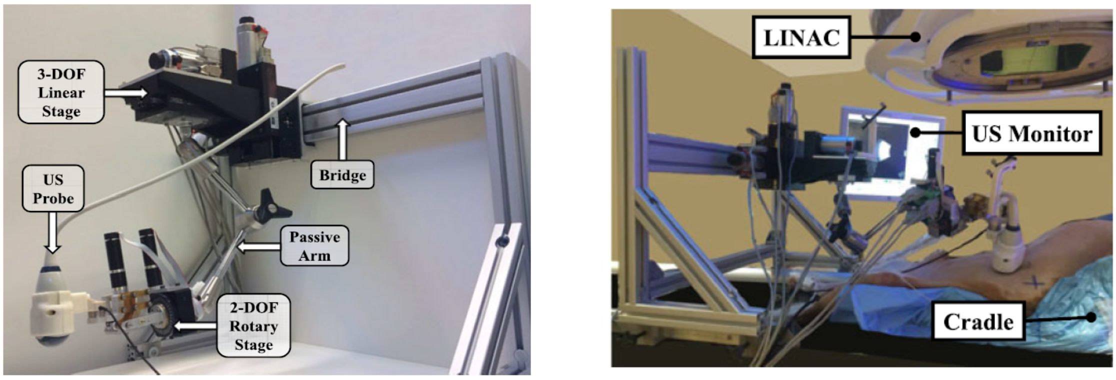


(a) US probe and biopsy needle

(b) Experimental Setup showing DOF

Figure 2.1: Autonomous Robotic Multiple-Core System

This recorded force enables reproducibility of the US probe placement. The robotic system has a 3-DOF linear stage, a 6-DOF passive arm and a 2-DOF tip for manipulating the US probe. The robotic system uses an optical tracking system to determine the position and orientation of the passive arm as shown in figure 2.2a [15]. Animal studies were performed with the designed system using a canine. The results of the study highlighted that the robotic system caused tissue deformation. However, it has the potential to improve the reproducibility of the patient setup and soft tissue deformation during fractionated radiotherapy. Figure 2.2b [15] shows the experimental setup with the canine. The same research group Sen *et al.* [16] proposed a novel approach for US image feedback for use in image-guided radiation therapy of soft-tissue targets. This system provides real-time image guidance to locate the soft tissue during treatment procedures thereby decreasing the need for extensive US imaging expertise. The system was designed to ensure accuracy and repeatability of patient setup during fractionated radiotherapy. Experiments with plastic abdomens were performed to validate the performance of the robotic system. The results highlighted that the system recorded an US image-based patient setup error less than 2.2 mm.



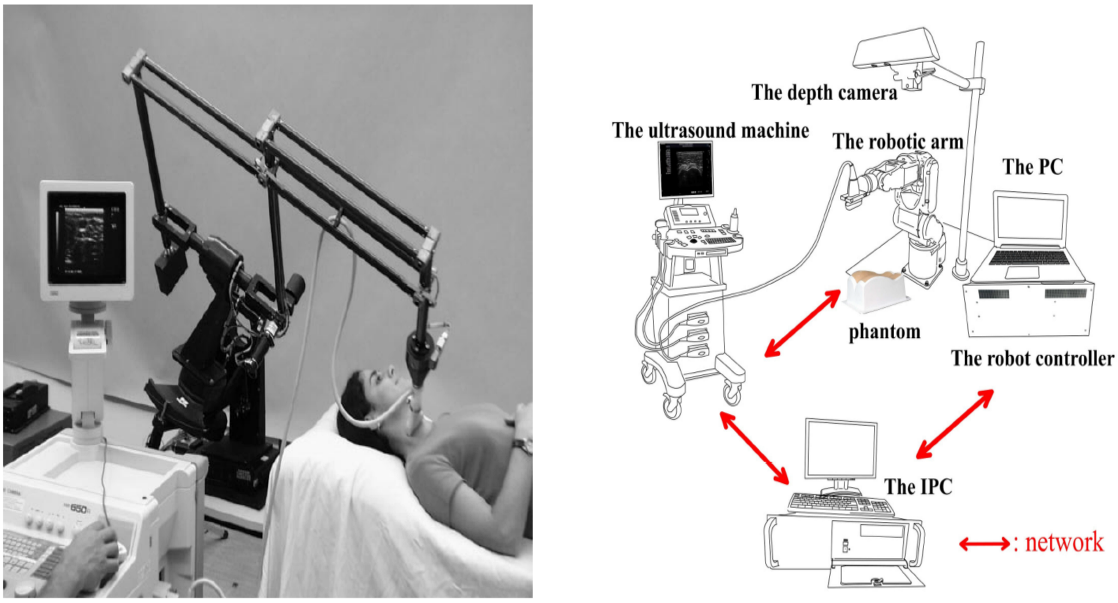
(a) Robotic System

(b) Experimental Setup

Figure 2.2: Cooperatively controlled robotic system for US-guided radiotherapy

2.2. Diagnosis

Abolmaesumi *et al.* [17] designed a robotic system for diagnosis of occlusions in the carotid artery. The entire system is made up of a user interface, a control system, slave manipulator and an US probe. Through tracking algorithms, occlusions in the carotid artery can be detected. The control of the robotic arm is achieved by using ultrasound visual servoing which guides the motion of the probe during examination. The setup is shown in figure 2.3a [17]. Huang *et al* [18] developed a robotic US system that incorporates a 6-DOF Epson robotic arm, an US probe, a conventional US machine, a depth camera and a control system as shown in figure 2.3b [18]. The depth camera was used to capture the contour of the tissue under consideration. Two force sensors were attached to the US probe to provide contact feedback for accurate tuning of the position of the robotic arm. A normal vector based approach was used to determine the position of the robotic arm and control the center of the US probe to reach the intended target. The system can fully manipulate the US probe to capture B-scan images with their corresponding spatial information, which can then be used for the volumetric reconstruction of the ROI. Experiments were conducted with phantoms and they yielded promising results. The same research group investigated the use of linear tracking for 3D medical ultrasound imaging [19]. The system was designed to manipulate the US probe at a uniform speed with little rotation by using a 1-DOF linear mechanical scanning device and a 6-DOF spatial sensor. The mechanical system was realized by implementing a linear sliding track with sensors to capture the positional information. The results of the phantom studies suggest that the system is well suited for generating 3D US images with a reasonable computational speed.



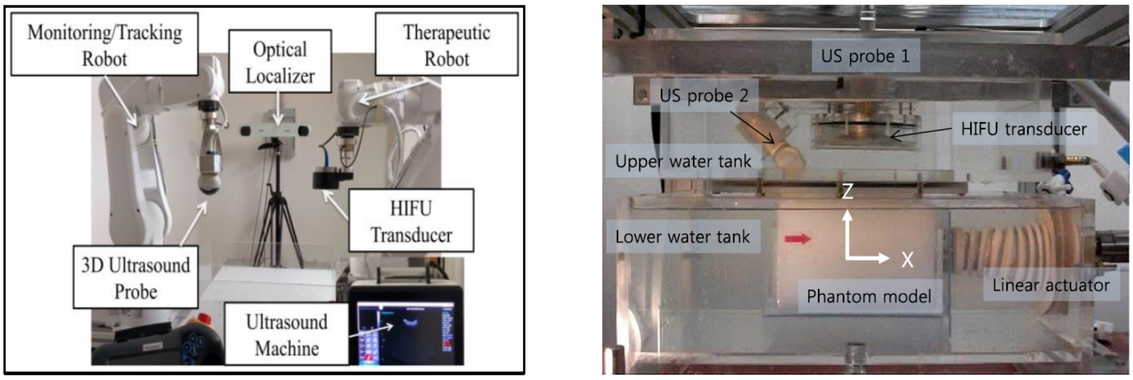
(a) Experimental setup for the diagnosis of Carotid artery occlusions

(b) Robotic US with a depth camera

Figure 2.3: Robotic arm US for 3D imaging

2.3. Therapy

The application of US for therapy has mainly been in the area of High Intensity Focused Ultrasound (HIFU). This is a non-invasive technique for thermal ablation [20]. Cafarelli *et al.* [21] assessed the accuracy of treatment using the FUTURA (Focused Ultrasound Therapy Using Robotic Approaches), a robotic assisted HIFU platform. The platform has a robotic module which is made up of two 6-DOF robotic arms, a therapeutic module which consists of HIFU transducer for thermal ablation, a monitoring module made up of US probes for monitoring the intended target for the therapy and a sensing module for calibrating the platform as shown in figure 2.4a [21]. The system was used to locate the target and the spatial information of the center of the target was used to position the HIFU transducer for the therapy. The recorded average error in the radial direction for the HIFU transducer was less than 1 mm and 3.7 mm in the axial direction due to mechanical effects of cavitation. Seo *et al.* [22] developed a HIFU system for the ablation of a moving target. Heart beat and respiration can cause the movement of abdominal organs and displace the organ of interest during therapy. By using US image visual servoing, the distance between the organ of interest and the current location of the HIFU transducer can be estimated, and this is used to adjust the robotic arm holding the HIFU transducer. An experimental setup shown in figure 2.4b [22] to mimic respiratory induced motion was simulated in a water tank with a kidney phantom and a linear actuator. The results showed a motion compensation accuracy of about 1.7 mm (RMS). Magnetic Resonance Image (MRI) guided HIFU application in the treatment of prostate cancer transrectally was explored by Yiallouras *et al.* [23]. An MRI compatible robotic system was used to navigate the HIFU transducer. The system was tested in a 1.5 T MRI machine using gel phantoms immersed in a water tank to evaluate the entire functionality of the robotic system and the repeatability of the process. The accurate navigation of the robotic systems ensured the creation of lesions in the gel phantoms under MRI guidance.

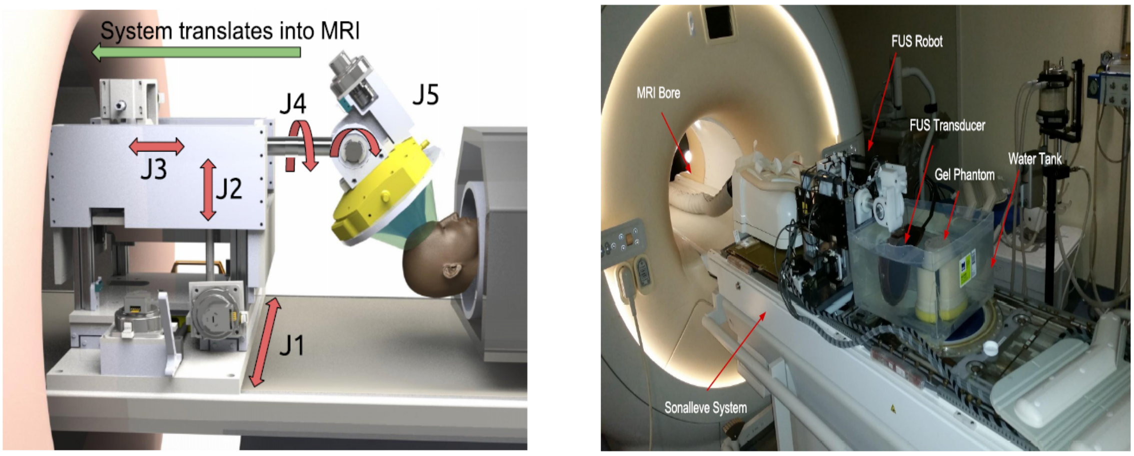


(a) Robotic assisted HIFU platform

(b) HIFU system for the ablation of a moving target

Figure 2.4: Robotic systems for HIFU therapy

Similar work done by Chopra *et al.* [24] sought to investigate the treatment of prostate cancer with HIFU under MRI guidance. This system is made up of a single planar US transducer which generates a collimated US beam for heating the prostate gland and a rotary motor which rotates $(360°)$ the US transducer to ensure a full coverage of the target. Experiments with phantoms and canines demonstrated that the system is able to achieve spatial heating patterns and therefore can be effective for the treatment of localized prostate cancer. Price *et al.* [25] developed an MRI guided robotic system for neonatal HIFU surgery. Focused Ultrasound Surgery (FUS) applied to the brain of neonates under MRI guidance has the potential to treat intraventricular hemorrhage (IVH). An MRI compatible robotic system was used to accurately position the HIFU transducer above the head of the neonate for therapy. A 5-DOF robotic manipulator which was integrated with the Philips Sonalleve MRgFUS (Magnetic Resonance guided Focused Ultrasound Surgery) ensured that the HIFU transducer is well steered around the skull of the neonate for transcranial FUS therapy. The results of experiments with tissue-mimicking phantoms showed that the system possesses the needed dexterity for FUS therapy. The experimental setup is shown in figure 2.5 [25].



(a) MRI compatible robotic system

(b) Experimental Setup

Figure 2.5: MRI guided robotic system for neonatal HIFU surgery

2.4. Discussion

US robotic systems designed for guidance mainly assist or guide clinicians by providing information on the patient's anatomy and the exact location of surgical instruments. As promising as some of the test results of the systems for these applications might be, they are mainly in their developmental stages with majority of testing being done with phantoms [3, 14, 16, 26, 27]. In the evaluation of the systems designed for biopsy, errors were recorded in centering the US transducer and the needle [14]. These were caused by the limited spatial resolution of the scanner, misalignment between the frame of reference of the robot and the US transducer, friction in the joints of the robot and miscalibration of the US velocity between the phantom and the medium [14]. Real-time target tracking using 2D US was a challenge in testing the feasibility of using US robotic systems for tumor manipulation during surgical procedures [26]. This was because 2D US imaging does not provide information on the volume of the target under consideration, and this makes it difficult to successfully manipulate the target.

Issues of safety are a major concern when interfacing robotic systems with patients. These systems are subjected to the highest forms of regulatory inspection before approval for clinical studies. The slightest error can have detrimental consequences. Safety concerns were reported by Merouche *et al.* [28] because the robotic arm acted haphazardly and as a result, safety protocols were incorporated into the software to shutdown the system during unwanted behaviour. This highlights how safety concerns retard the development of US robotic systems. Also, components that could possibly enhance the efficacy of most robotic systems are spurned due to safety concerns [29, 30].

All the systems covered in this review made use of conventional US machines and piezoelectric probes. Medical robotic systems have been reported to be expensive even though they have desirable advantages [31]. It is prudent that in the development of these US robotic systems, less expensive technology should be explored. A good example is the Capacitive Micromachined Ultrasound Transducers (CMUTs). These are Micro-electromechanical systems (MEMS) transducers devices and by relying on the well developed silicon manufacturing technology, the cost of these transducers can be made cheaper relative to PZT (Piezoelectric Transducers). Additionally, conventional US systems are expensive and other less expensive alternative systems should be designed for US robotic systems thereby minimizing cost.

3

Design Concept

3.1. CMUT

Ceramic piezoelectric transducers are the main transducer technology used in the ultrasonic industry today. Capacitive Micro-machined Ultrasound Transducers (CMUTs) on the other hand, recently surfaced . Piezoelectric transducers generate and receive sound waves using the piezoelectric principle. When short burst of electrical current is applied to the piezoelectric crystals, they undergo a deformation which generates acoustic waves. Conversely, when these mechanical waves are received by these crystals, they generate electric current in response to the applied mechanical stress. CMUTs generate ultrasounds based on flexural vibration of electrostatically-actuated plates [32].

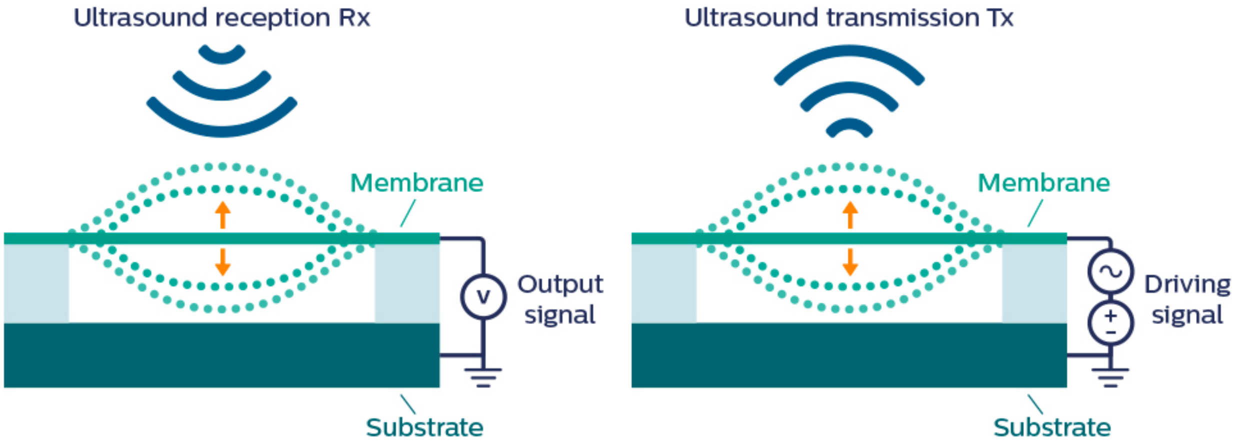


Figure 3.1: The mode of operation of a CMUT

They basically function like a capacitor with a thin movable plate over a vacuum [33]. A metal layer on the thin plate acts as the top electrode of the capacitor and the underlying substrate acts as the bottom electrode [33]. When a DC voltage is applied between the electrodes, an electrostatic force causes the top plate to be attracted towards the substrate. Applying an alternating current to the capacitor generates ultrasound. Conversely, if the plate is subjected to ultrasound, the pressure wave causes a change in the capacitance which generates electric current [33]. The mode of operation of the CMUTs is shown in figure 3.1 [34]. The major advantages of CMUTs have to do with microfabrication and integration with circuits for a myriad of applications. This integration enables many applications with a reduction in cost. CMUTs can be manufactured in large volumes while manual processing which scale linearly makes it difficult and expensive to manufacture Lead Zirconate Titanate (PZT) in large volumes. This makes CMUTs desirable for applications that require large volume production of ultrasound transducers. Philips Research has successfully designed CMUTs that have a large bandwidth, ability to be integrated with driver circuitry and can be easily fabricated [34]. One of these CMUTs was used for this work.

3.2. WUP

The Wearable Ultrasound Platform (WUP) is an electronic system designed by Philips Research to explore the applications of CMUTs. It consists of a Red Pitaya, an Arduino Teensy and a Philips Printed Circuit Board (PCB) (see figure 3.2).

3.2.1. Red Pitaya

The Red Pitaya is an open source measurement and signal generation device which runs a Linux operating system. It consists of a central processing unit (CPU) - dual core ARM Cortex A9 and Field Programmable Gate Array (FPGA), 2 RF inputs and 2 RF outputs, 50 MHz analog bandwidth, 14 bit analog-to-digital converters and 14 bit digital-to-analog converters. It has a buffer size of 16k and can store a maximum of 16384 samples at a sampling rate of 125 MS/s [35]. As such, a sampling rate of 125 MS/s takes $131.072\ \mu s$ to fill the buffer. Considering the speed of sound in body tissues which is approximately $1540\ m/s$, the maximum depth in which an echo can be recorded with the Red Pitaya is calculated in equation 3.1.

$$\begin{aligned}
velocity &= \frac{2 \times distance}{time} \\
distance &= \frac{velocity \times time}{2} \\
distance &= \frac{1540 \times 131.072 \times 10^{-6}}{2} \\
distance &\approx 10\ cm
\end{aligned} \tag{3.1}$$

3.2.2. Arduino Teensy

The Arduino Teensy is a microcontroller development board which runs on a 32-bit ARM cortex M4 processor [36]. It is programmable in the C++ programming language as well as with the Arduino

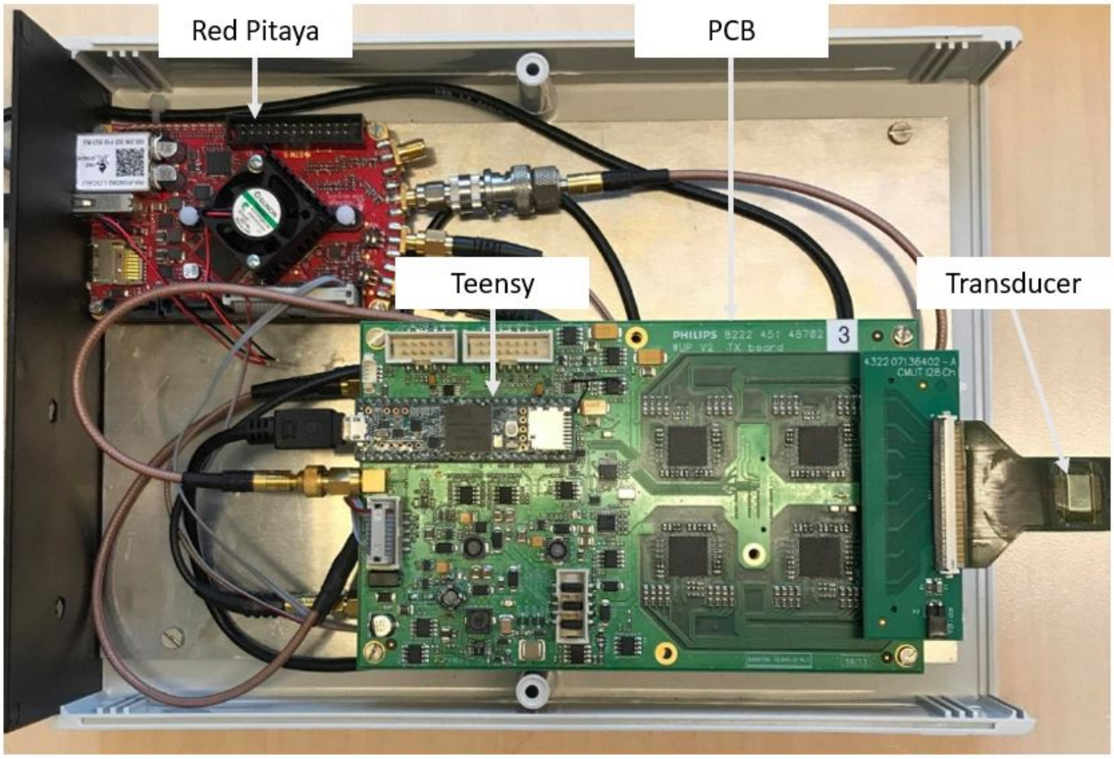
Integrated Development Environment (IDE) [36]. The Arduino Teensy is mounted on the Philips PCB and controls the pulsing elements (elements responsible for sending pulses to the CMUTs) on the PCB. Additionally, it controls delays, amplification factor, frequency, pulse shape, repetition frequency and the receive channels of the CMUT array. It basically functions as the microcontroller that controls the operation of the PCB.

3.2.3. Philips Printed Circuit Board (PCB)

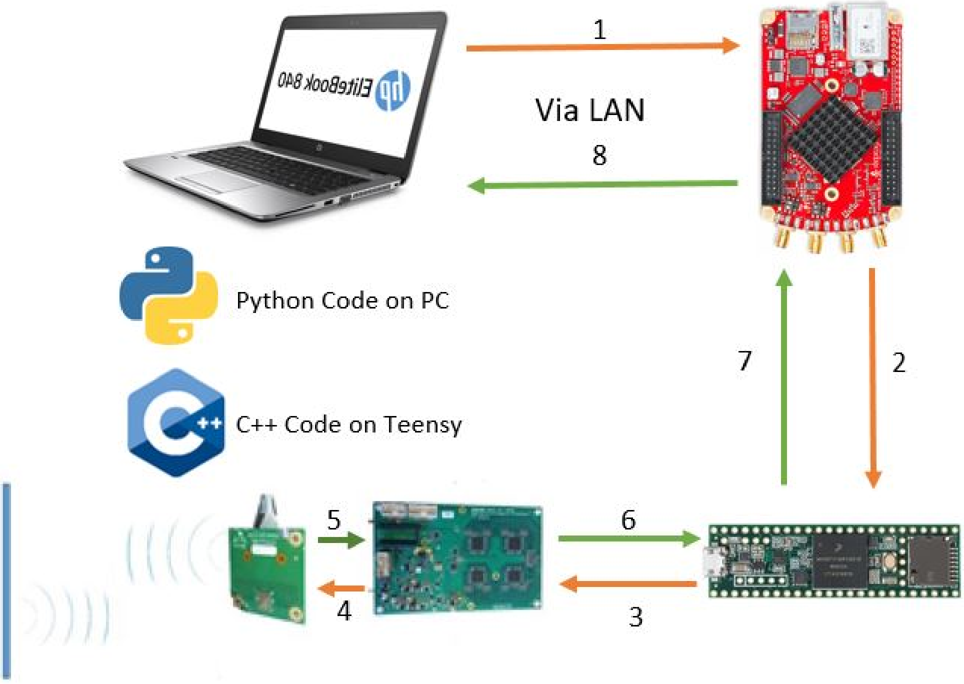
The Philips PCB was designed specifically to work in conjunction with the Red Pitaya and Teensy. The PCB controls the pulsing of the transducers and also the receiving of ultrasound. Received signals are filtered and amplified. The Teensy microcontroller controls the functionality of the PCB.

3.2.4. Mode of Operation

A python script that runs on the computer (laptop) controls the operation of the Red Pitaya. It causes the Red Pitaya to generate a square wave and this wave is transmitted to the Teensy unit on the PCB. The Teensy upon acquisition of this square wave triggers the pulse generating elements on the PCB which causes the generation of ultrasound waves. A C++ Application Programming Interface (API) enables the control of the functionality of the Teensy by defining calls and requests, and it also provides a computing interface to interact of the WUP. The API accepts calls and requests from the user which drives the desired operation of the PCB and the CMUT array. Commands such as gain, transmit and receive channels, amongst others can be specified using the Teensy's user interface. The Teensy is configured such that, it triggers the pulse generating elements on the positive edge of the square wave. The PCB filters and amplifies the received ultrasound signals and the data is recorded by the Red Pitaya. The Red Pitaya has an ADC functionality which converts the analog signals to digital signals. The Teensy microcontroller controls the transmitting and receiving functionality of the PCB.



(a) Wearable Ultrasound Platform



(b) Mode of Operation

Figure 3.2: Overview of the Wearable Ultrasound Platform showing how the individual components are configured

3.3. Ultrasound Imaging

US imaging for diagnostic purposes often employ the pulse-echo principle [37]. The pulse-echo principle involves the intermittent generation of short bursts (pulses) into a region of interest and receiving the reflected pulses known as the echo.

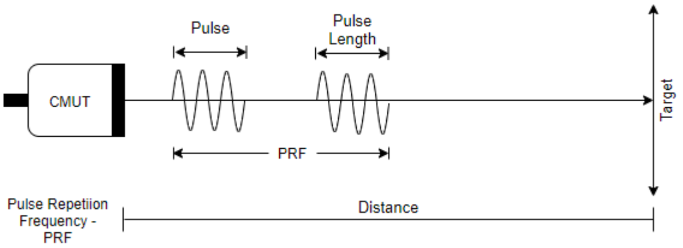


Figure 3.3: Pulse propagation from a CMUT. The Pulse length is the number of cycles in a pulse and the PRF is the number of pulses per second

These pulses are reflected as they propagate through body tissues which have different acoustic impedances. The echo signals that are received by the transducer causes the transducer to generate an electric signal, which is then analyzed and processed to form an image. This image is mostly displayed in brightness-mode (B-mode) as a 2D image. To reconstruct a 3D US volume from 2D US images, one approach is using a 2D US probe and keeping track of the relative position and orientation of the B-scans [1]. This technique was employed in this work by using a robotic arm to track the relative position and orientation of the B-scans.

3.4. Robotic Arm

Several approaches for keeping track of 2D image spatial information have been reported in literature. Some of these approaches include the use of mechanical 3D probes, mechanical localizers, freehand scanners, amongst others [1]. Under the freehand scanners approach, techniques such as acoustic positioner, optical positioner, magnetic field sensors and articulated arm positioner have been explored. The use of a robotic arm falls under the articulated arm positioner technique and this informed the decision to use a robotic arm for the purposes of this project. A 6 degree of freedom (DoF) robotic arm designed by Philips Research (figure 3.4) was used in this project. The arm has six joints and in each of these joints is an RM08 rotary magnetic encoder with a 12-bit resolution. It has an ESP32 low-power microcontroller which reads information from the arm's encoders. The arm has an average error of $0.264\ mm$ and repeatability of $0.951\ mm$.

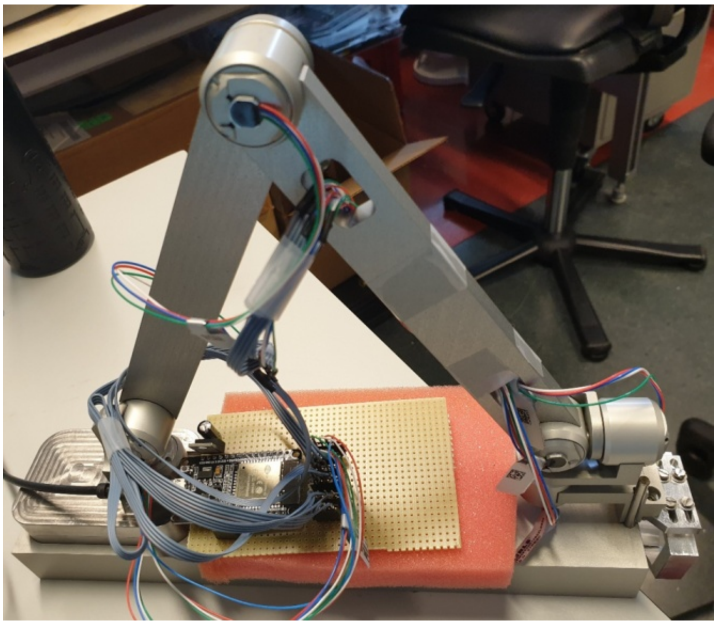


Figure 3.4: Robotic arm

3.5. Robotic Operating System (ROS)

ROS is an open source robotic software development framework, which functions as a middleware on the Linux operating system and facilitates operations on a distributed network [38]. It provides hardware level abstraction, the transfer of data (messages) from processes running on a hardware system to its package management system. The ROS architecture is organized into packages. A package is a collection of nodes, configuration files, ROS-independent library and other modules for a specific application. Nodes are executables that can communicate with other nodes in different packages or within the same package by publishing or subscribing to a topic. Topics are busses over which nodes can exchange messages. These nodes are programming language agnostic, which means their communication is independent of the programming language they have been written in. This creates an ecosystem which integrates and controls various functionalities on a distributed network. A master node provides unique naming conventions and services for all the nodes in the ROS network. It allows nodes on the distributed network to locate each other. Nodes establish a peer-to-peer connection once they locate each other. This provides a singular platform to integrate the WUP system, CMUT, the robotic arm and a computer. The ROS master architecture is shown in figure 3.5. A node notifies (advertise) the master that it wants to publish a message to a topic. The node begins to publish the message to the topic but no peer-to-peer connection is established. A peer-to-peer connection between nodes is only established when a node notifies the master that it wants to subscribe to that topic. Once subscribed, a node can retrieve messages from that topic via a callback function. This sequence of events enables nodes on the distributed network to communicate with each other.

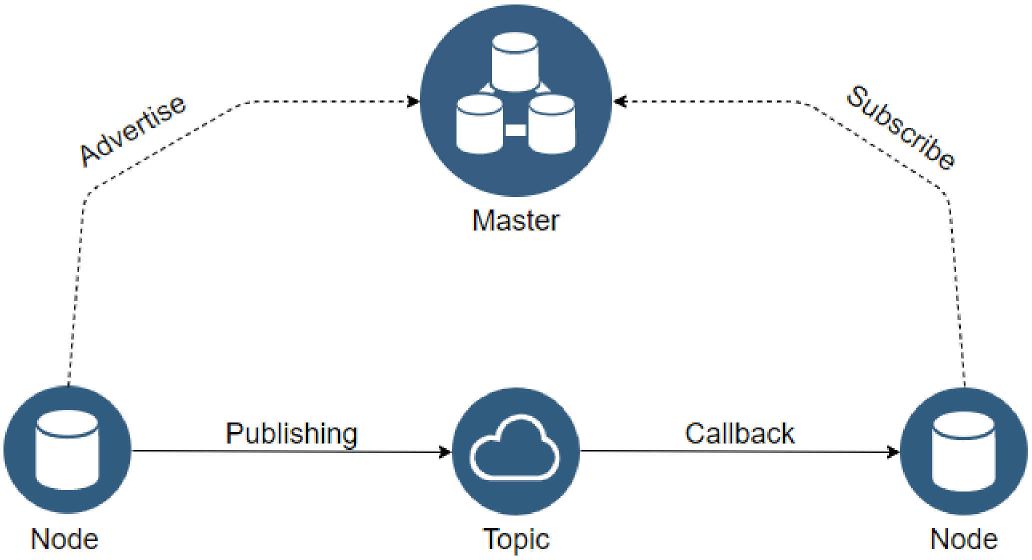


Figure 3.5: ROS communication architecture. Nodes publishes data onto a topic and other nodes gain access to the data by subscribing to the topic via a callback function

ROS was installed on the Red Pitaya and this provides access to control the functionality of the WUP system. ROS has a transform library that keeps track of multiple coordinate frames of the arm over time. At any given point in time, a transformation between any two coordinate frames can be requested. This transformation will provide the spatial location and orientation of the transducer and consequently the spatial location and orientation of the 2D US image plane. To make the robotic arm accessible within ROS, the arm has to be described in the Unified Robot Description Format (URDF). This is an Extensible Markup Language (XML) that describes a visual model of the arm. The XML contains links between different components of the arm and how they relate to each other (translation and rotation). A section of the graphical representation of how joints and links connect to each other is shown in figure 3.6. XYZ values denote the translation vectors while RPY denote the roll, pitch and yaw values in radians. The Standard Triangle Language (STL) file of each component is made accessible to the XML file by defining the path to the location of the STL file. Adding the STL files for each component produces the visual model of the robotic arm shown in figure 3.7. The ROS joint state package publishes the joint state values of a specified URDF to a transform topic. A node can subscribe to the transform topic and retrieve these joint state values. ROS has a 3D visualization tool (Rviz) that provides real time visualization of the movement of the arm and sensor information, amongst others. By constantly publishing the joint state values of the arm, the real-time movement of the arm can be visualized in the Rviz environment.

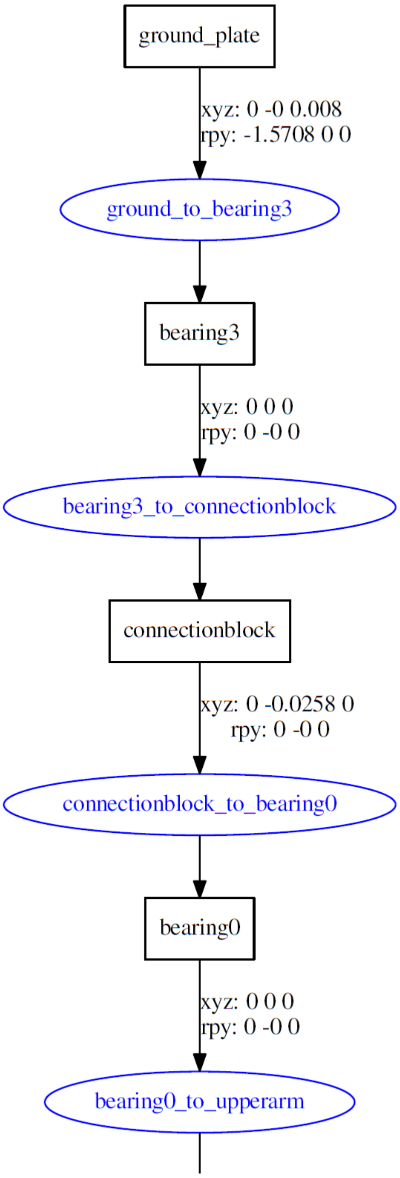


Figure 3.6: A portion of the URDF graphic representation of the robotic arm. XYZ values denote the translation vectors while RPY denotes the roll, pitch and yaw values in radians.

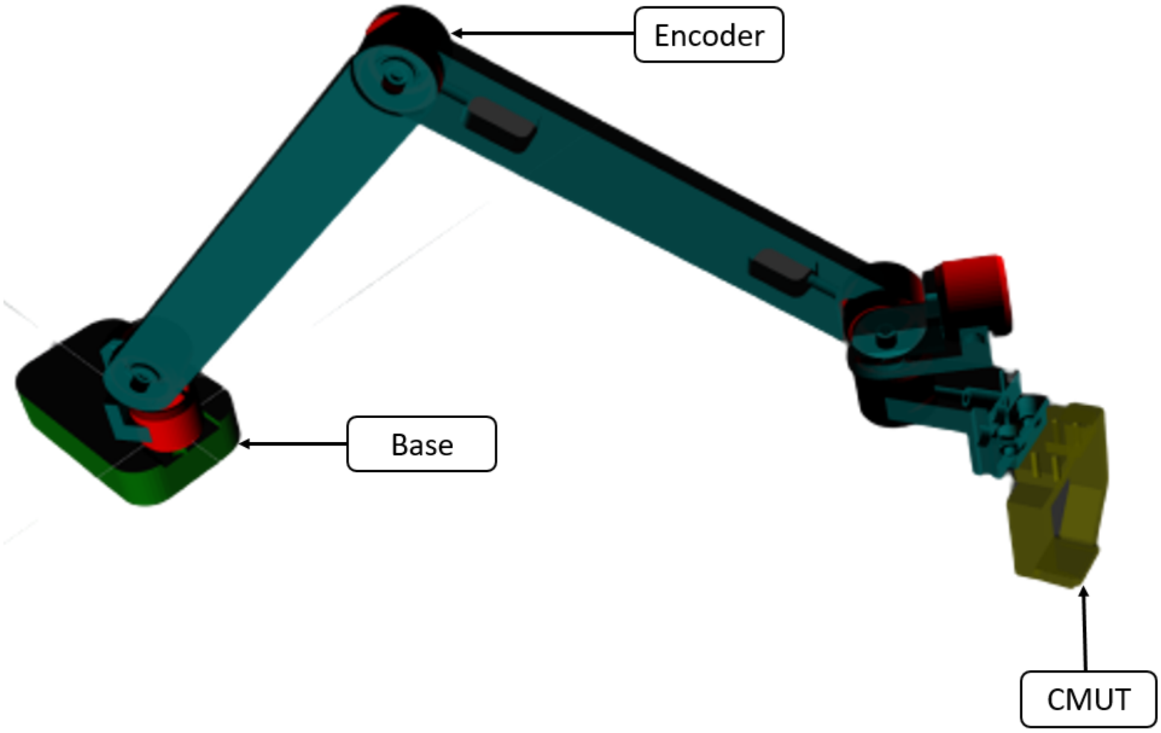


Figure 3.7: URDF description of the robotic arm visualized in Rviz

3.6. System Integration

This section describes how the various systems within the ROS network coordinate processes with each other. Figure 3.8 provides a visual representation of the distributed systems within the ROS network.

3.6.1. WUP Node

A node that runs on the Red Pitaya (a component of the WUP) controls the entire functionality of the WUP. The US data acquired by the Red Pitaya is digitized at a sampling rate of 125 MS/s. The raw RF data, which is an array (a python data structure) of float values is published to the US topic. This makes the US data accessible to any node within the ROS network.

3.6.2. Signal Processing (SP) Node

Filtering

The SP node, which runs on the computer (laptop) subscribes to the US topic and receives the RF data. The Savitzky-Golay (S-G) filter from the python Scipy library was implemented to smoothened the signal. The S-G filter is a low-pass filter that is based on the local least-squares approximation of polynomials [39]. This filter reduces the noise in the signal while maintaining the shape of the signal. A 5th order Butterworth bandpass filter implemented in python with the Scipy library was used to select

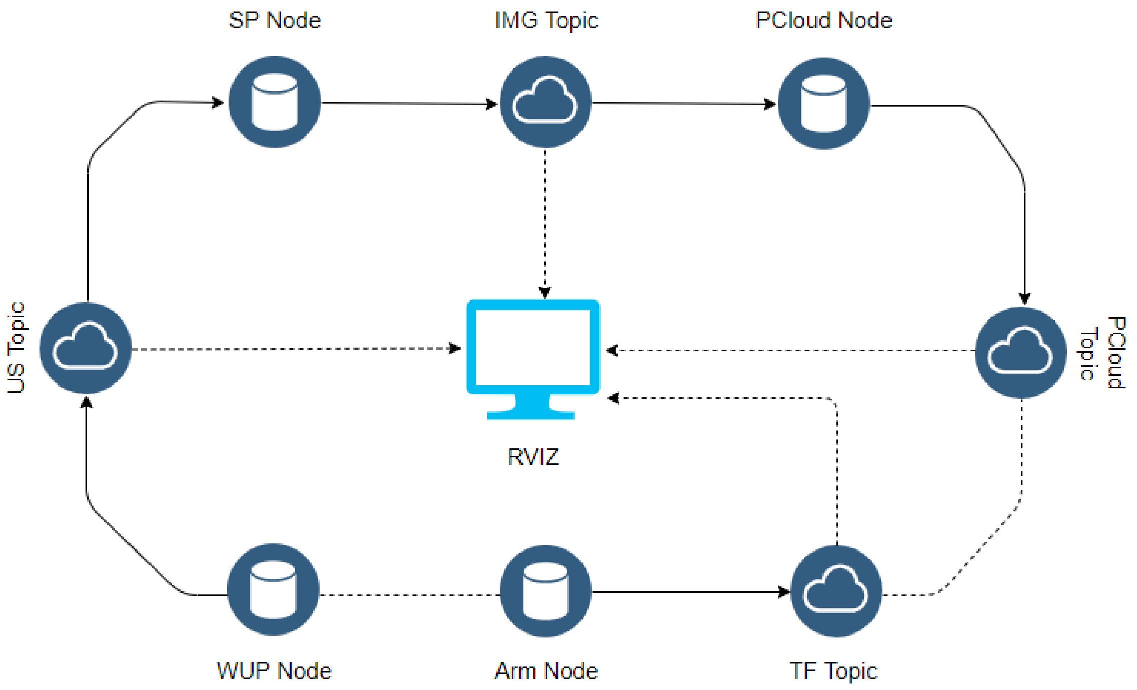


Figure 3.8: Graphic representation of nodes within the ROS distributed network and their connections to their respective ROS topics and the RVIZ. Dashed arrows indicate that all ROS topics are accessible from RVIZ

the frequency band of interest and remove any high frequency noise from the signal. The Scipy library is an open-source python library for scientific computing. It contains modules that facilitate the easy and quick implementation of processes such as filtering, optimization, interpolation, amongst others. Butterworth filters provide the best compromise between attenuation and phase response [40], and this motivated the choice to use a Butterworth filter.

Envelope Detection

In B-mode (Brightness mode) US imaging, only the amplitude of the signal is relevant to the construction of the image. The amplitude of the signal was calculated via a process known as envelope detection. Envelope detection discards the phase information and uses the Hilbert transform to determine the envelope of the signal. The Hilbert transform calculates the square root of the sum of the squares of the real components and the quadrature components. It filters out all the negative frequencies components in the signal [41]. The Hilbert transform for a given signal $x(t)$ is given by:

$$\begin{aligned} &x_a(t) = x(t) + iH\{x(t)\} \\ &\text{where} \\ &H\{x(t)\} = \frac{1}{\pi t} * x(t) \\ &\text{hence} \\ &x_a(t) = x(t) + i\left(\frac{1}{\pi t} * x(t)\right) \\ &\text{* is a convolution operator} \end{aligned} \tag{3.2}$$

The Hilbert transform converts the given signal $x(t)$ into an analytical signal $x_a(t)$.

Log Compression

The signal obtained after applying the Hilbert transform has a high dynamic range. If the amplitude values are mapped directly to gray-scale pixel intensity values, important details contained in the image will be missed [42] due to the high dynamic range, as the presence of high amplitude values will overshadow other pixel intensity values. Log compression is typically used to reduce the dynamic range of the signal by mapping the amplitude values nonlinearly to the gray-scale pixel intensity values with the log function shown in equation 3.3 [43]. The diagrammatic representation of all the signal processing steps is shown in figure 3.9 [44].

$$I(t) = 20\log_{10}(1 + env(t)) \tag{3.3}$$

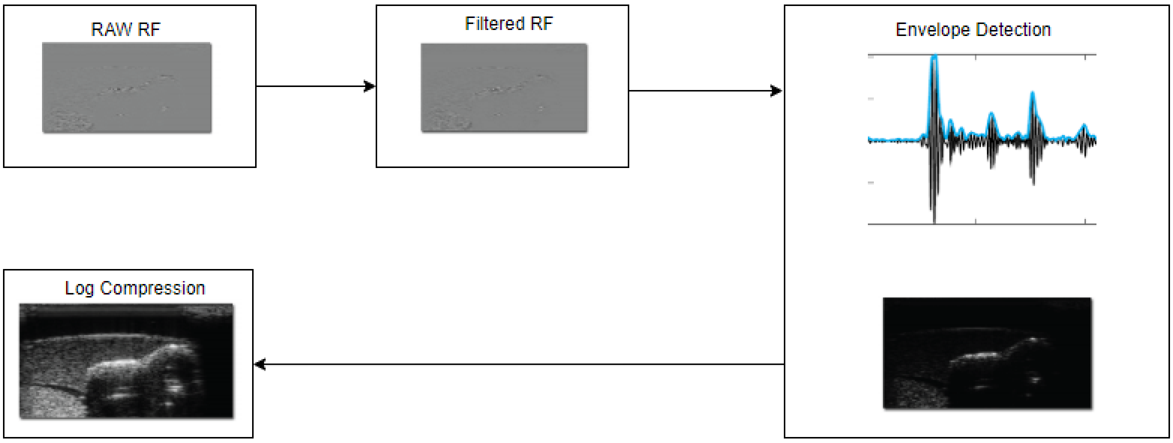


Figure 3.9: Stepwise representation of signal processing techniques depicting how raw RF signals are converted to an ultrasound image

3.6.3. Format Conversion

The resulting array after log compression is mapped to a 0 - 255 pixel intensity value range. The echo intensities and time delay between echo provides a visual representation of the object being imaged. The array of pixel intensity values is published to the IMG topic.

3.6.4. Point Cloud (PCloud) Node

The ROS visualization environment (Rviz) provides a Point Cloud Library which enables the real-time visualization of 2D or 3D US data. A point cloud is a collection of points that represent a shape in space. Each point in the point cloud has x, y, z coordinate values. The PCloud node subscribes to the IMG topic and receives the image data. The image data is a 1D array of gray-scale pixel intensity values. To reduce the number of intensity values while retaining relevant information inherent in the image, and to decrease the computational load involved in displaying these images, spline interpolation was used to downsample the 1D array. Spline interpolation provides the best trade-off between accuracy and the cost of computation [45]. The resulting array after interpolation is converted into a point cloud. The point cloud data is published to the PCloud topic and it is accessible within Rviz. All topics within the ROS network are accessible within Rviz and this is shown by dashed arrows connecting them to Rviz in figure 3.8.

3.6.5. Arm Node

The arm node is a python script that reads the spatial information from the robotic arm and publishes the states of all the joints to the TF topic (JointState). These values are provided by the six encoders of the robotic arm. This information enables the ROS transform (tf) package to keep track of the 3D coordinate frames of the robotic arm. The tf package uses a tree structure to maintain the relationship between multiple coordinate frames. These coordinate frames are buffered and at any point in time the transformation of one coordinate frame with respect to another can be obtained [46].

4 Results

4.1. System Setup

The system setup is shown in figure 4.1. The CMUT has 64 transducer elements but only 32 elements are supported by the WUP. It can therefore transmit ultrasound signals from only the 32 supported channels. It can also receive echoes from the 32 channels but does not have the functionality to process the signal received by each element. It has only two channels to process signals - it averages signals received on the odd number elements (1 -31) and processes them on channel one (CH 1) and averages signals received on even number elements (2-32) and processes them on channel two (CH 2). The number of elements used to transmit and receive ultrasound signals can be specified through the WUP node. Irrespective of the number of elements used for transmitting and receiving, the WUP can only process two signals by averaging the signal from the old and even number transducer elements. The end effector of the robotic arm holds the CMUT and this configuration provides the spatial orientation (translation and rotation) of the CMUT by resolving the forward kinematics. Key Teensy settings that enable the desired operation of the CMUT are shown in table 4.1.

Table 4.1: Key Teensy Settings that drive the CMUT's Operation

Frequency	$2\ MHz$
Number of Cycles	1
Number of Transmitting Channels	32
Number of Receiving Channels	32 (CH1: 1-31, CH2: 2-32)
Pulse Repetition Frequency	$3\ kHz$

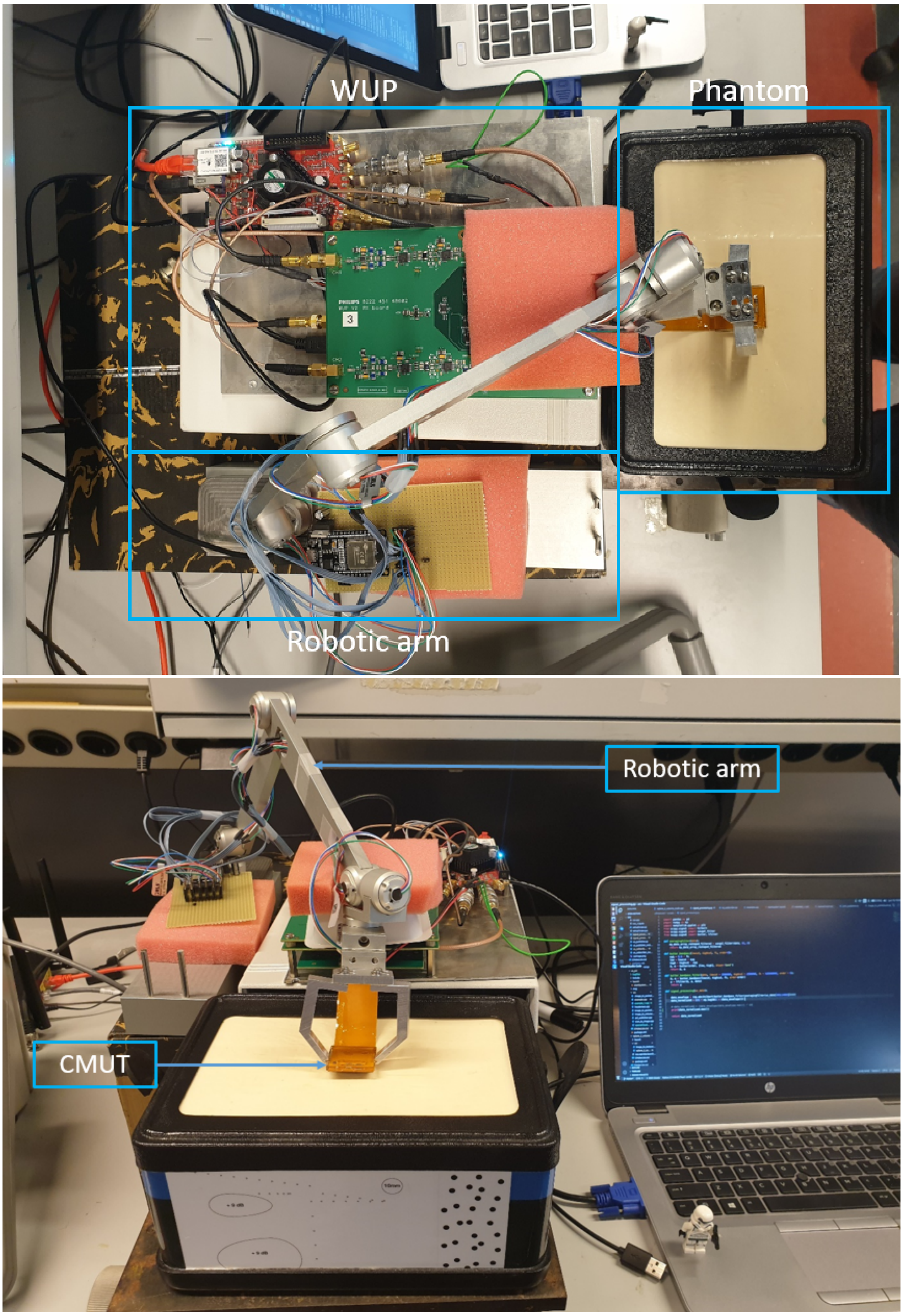


Figure 4.1: System setup showing the Wearable Ultrasound Platform (WUP), robotic arm, phantom, and CMUT

4.2. Pulse Echo - RF

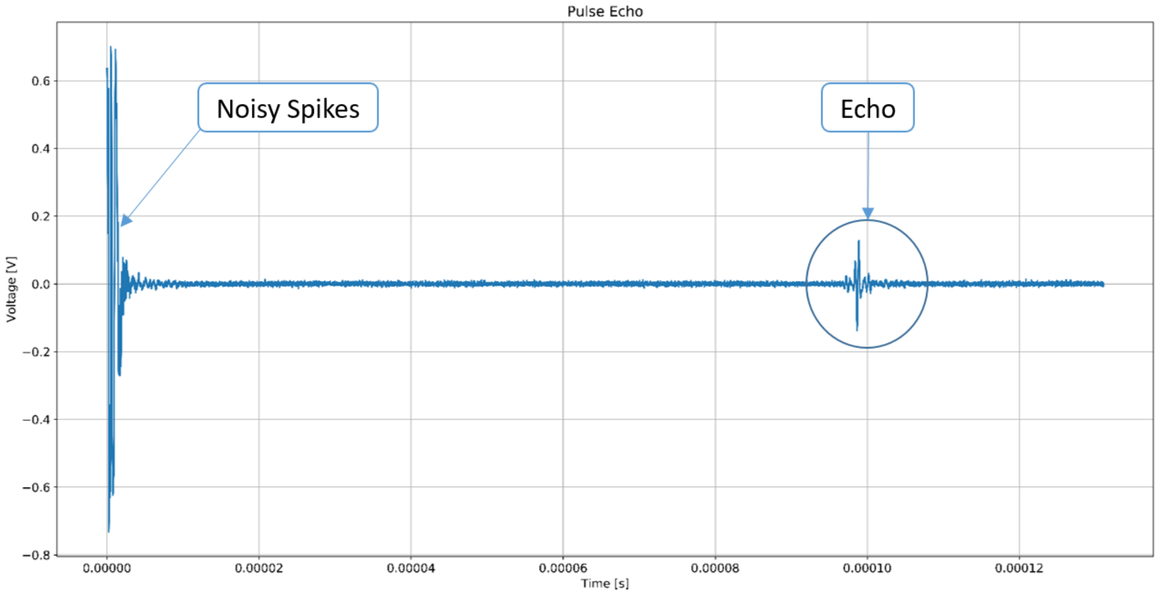


Figure 4.2: Raw RF data received for a single pulse-echo cycle.

Figure 4.2 shows the result of a single pulse echo cycle. The Red Pitaya has fast analog inputs which acquire analog signals and converts them to digital signals. It has a 16k buffer and a decimation factor determines the duration of signal acquisition. A decimation factor of 1 was used which translates to a time scale of $131.072\ \mu s$ and this completely fills the 16k buffer. Signal acquisition begins when the triggering criteria is met. The noisy spikes shown in figure 4.2 are the result of desaturation of diodes on the PCB. High voltage pulses are blocked by the low voltage receivers. A trigger level of $100\ mV$ was used to configure the trigger settings of the Red Pitaya. The echo received is shown in figure 4.2. This is the raw radio-frequency (RF) data that needs to be processed to produce an ultrasound image. The results of the various processing steps are discussed below.

4.3. Filtered Signal

The received RF data was smoothened with the Savitzky-Golay (S-G) filter and the noisy spikes showed in figure 4.2 were removed. The result of this process is showed in the figure 4.3. A 5th order Butterworth bandpass filter implemented in python was used to filter the signal. This reduces noise present in the signal outside the frequency of interest [47] and the result is showed in figure 4.4. Applying the bandpass filter to the signal causes an attenuation in the amplitude of the signal. This is expected as filter implementations do not have an instantaneous transition from pass band to the stop band. This attenuation is widely referred to as the stopband attenuation [48]. It is interesting to note that band pass filtering can also be used to obtain harmonic or sub-harmonic signals which are relevant in harmonic imaging [49]. Key bandpass filter variables are shown in table 4.2. The complete python implementation is shown in appendix B.2.2.

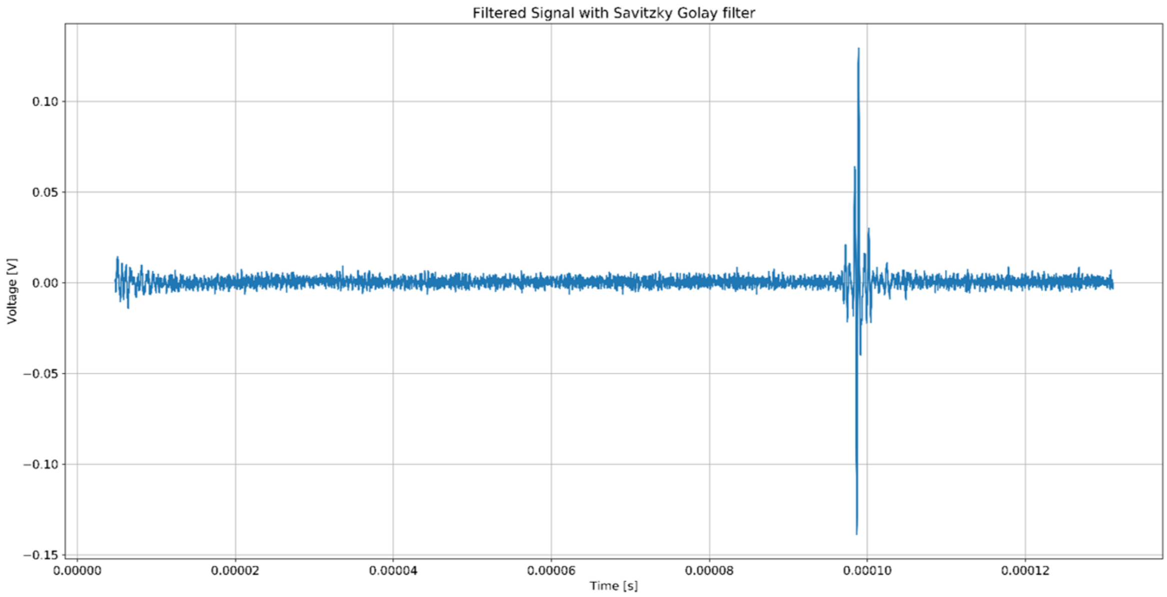


Figure 4.3: RF data after applying the Savitzky-Golay filter

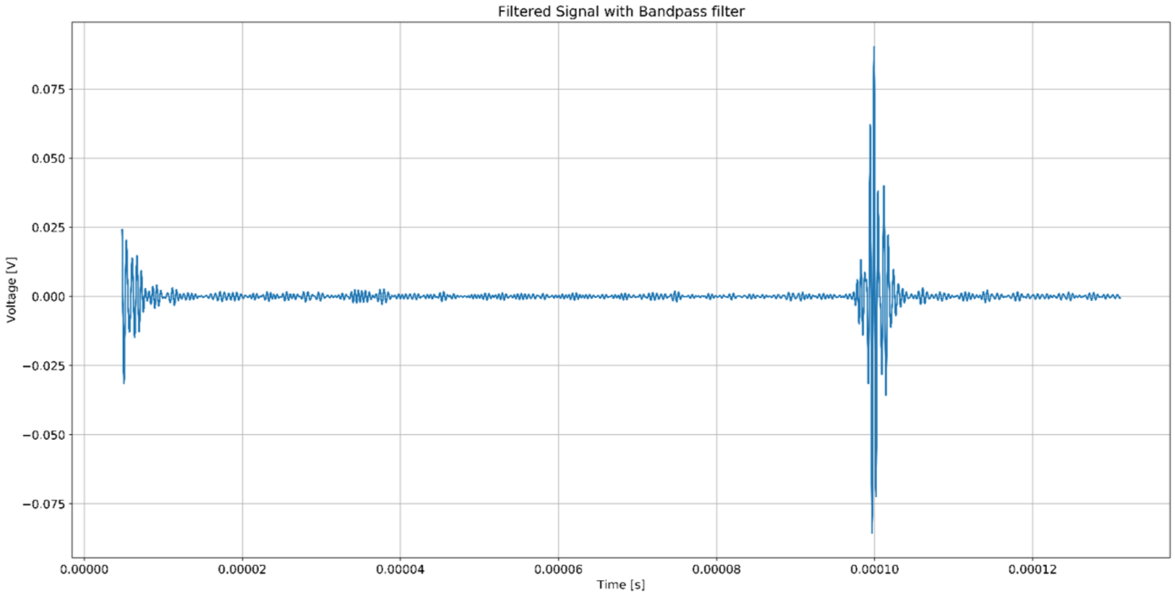


Figure 4.4: RF data after applying a 5th order butterworth bandpass filter

Table 4.2: Bandpass Filter Settings

Low cutoff frequency	1.5 MHz
High cutoff frequency	2.5 MHz
Order	5
Sample Frequency	125 MHz

4.4. Envelope Detection

Applying the Hilbert transform to the filtered signal enables the detection of the envelope of the signal. Figure 4.5 shows a plot of the filtered signal with the detected envelope of the signal and figure 4.6 zeros in on the echo.

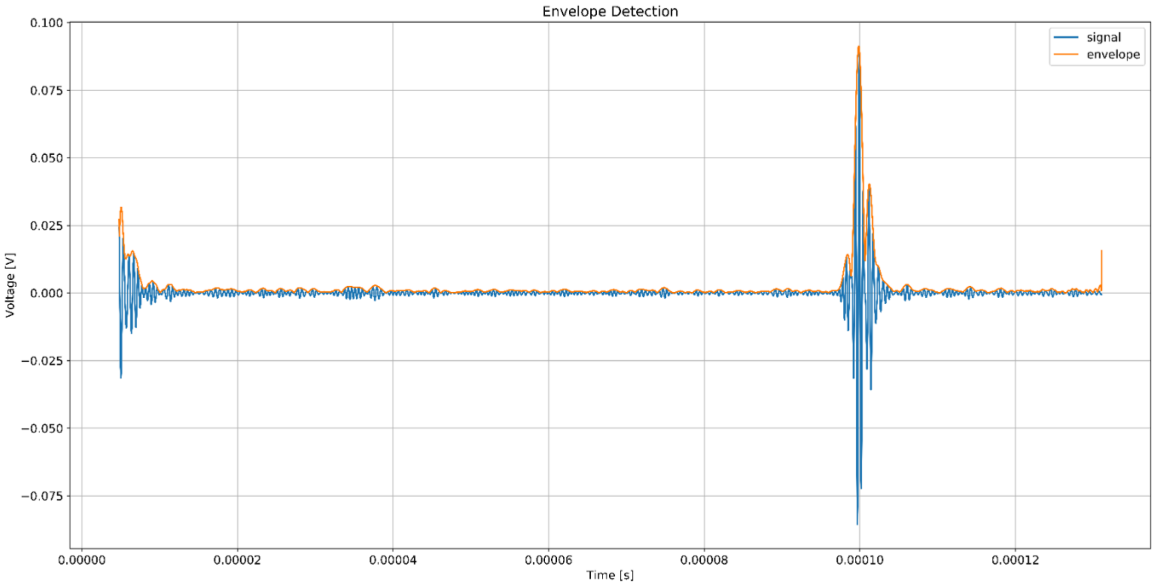


Figure 4.5: Envelope Detection of filtered RF signal

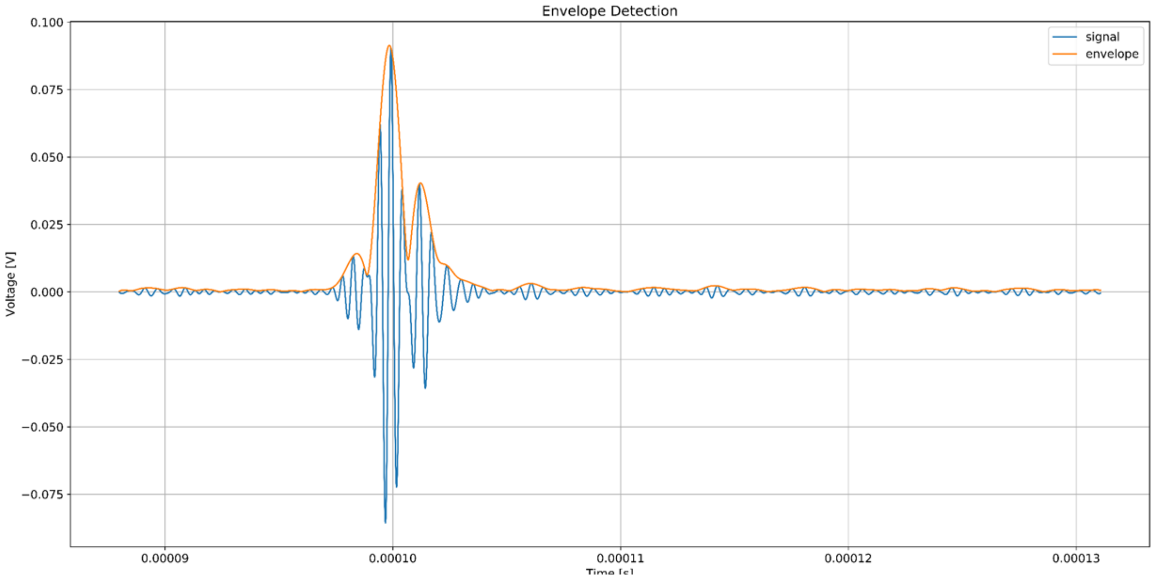


Figure 4.6: Envelope Detection of filtered RF signal - echo zoomed in

4.5. Point Cloud

The detection of the envelope of the signal produces an array of amplitude values that are converted to a point cloud. This point cloud represents a single scan line or image slice which is visualized in real-time in Rviz. Integrating scan lines from multiple regions of the area of interest produces a 3D image. TF (transformation) which is a ROS library saves information of the location and orientation of the robotic arm and their timestamps in a buffer. This functionality allows the synchronization of data on the position of the arm and ultrasound scan at any point in time. This approached made it possible to track the location and orientation of the arm and visualization its coordinate frames in real-time during scanning. Scanning a volume of interest was achieved by making an ultrasound sweep with the end effector of the robotic arm. Figure 4.7 provides a visual representation from a single scan line to a volume by scanning across a small region. Beam steering was not implemented.

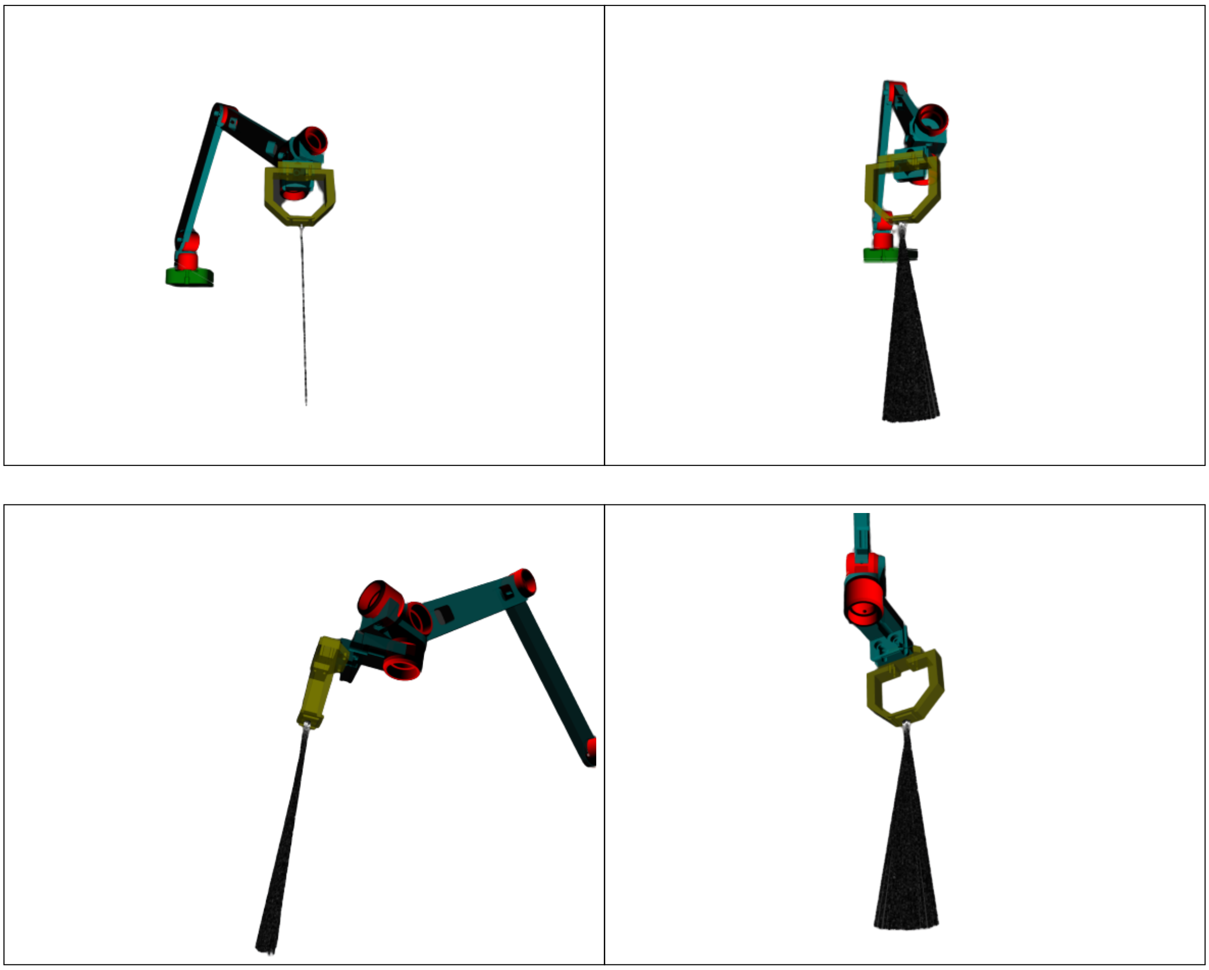
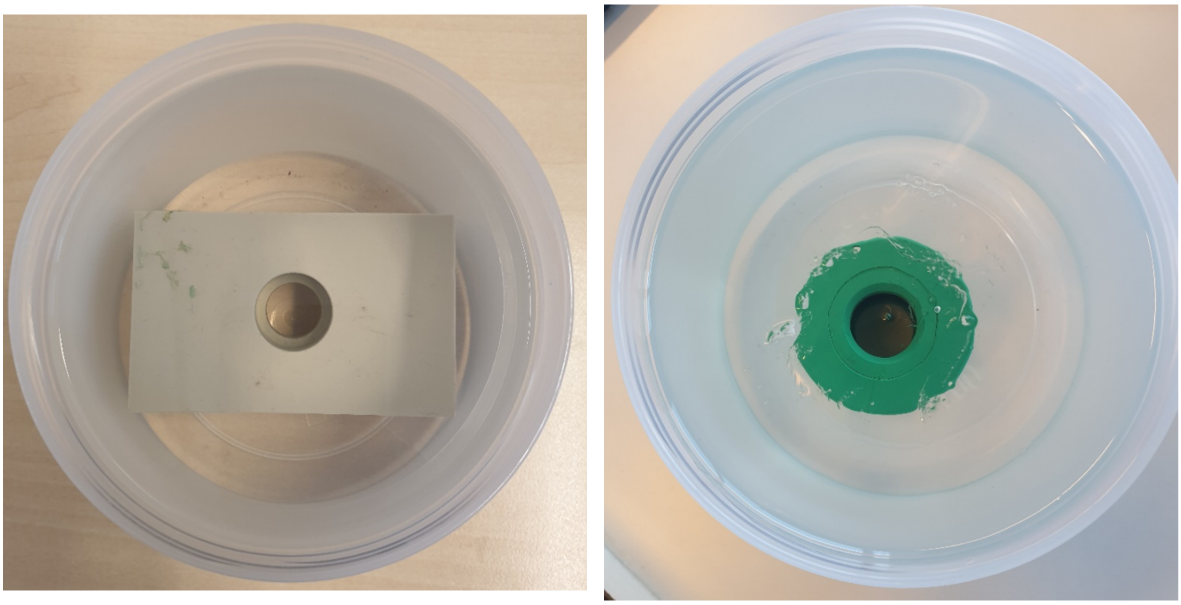


Figure 4.7: Rviz Visualization environment

4.6. 3D Scans of Phantoms

Gel phantoms with plastic shapes embedded in them were prepared to test the performance of the design. The results of 3D image acquisition of two phantoms with the final design and the Philips EPIQ 7 ultrasound machine are presented in this section.



(a) Phantom 1

(b) Phantom 2

Figure 4.8: Phantoms used for 3D imaging

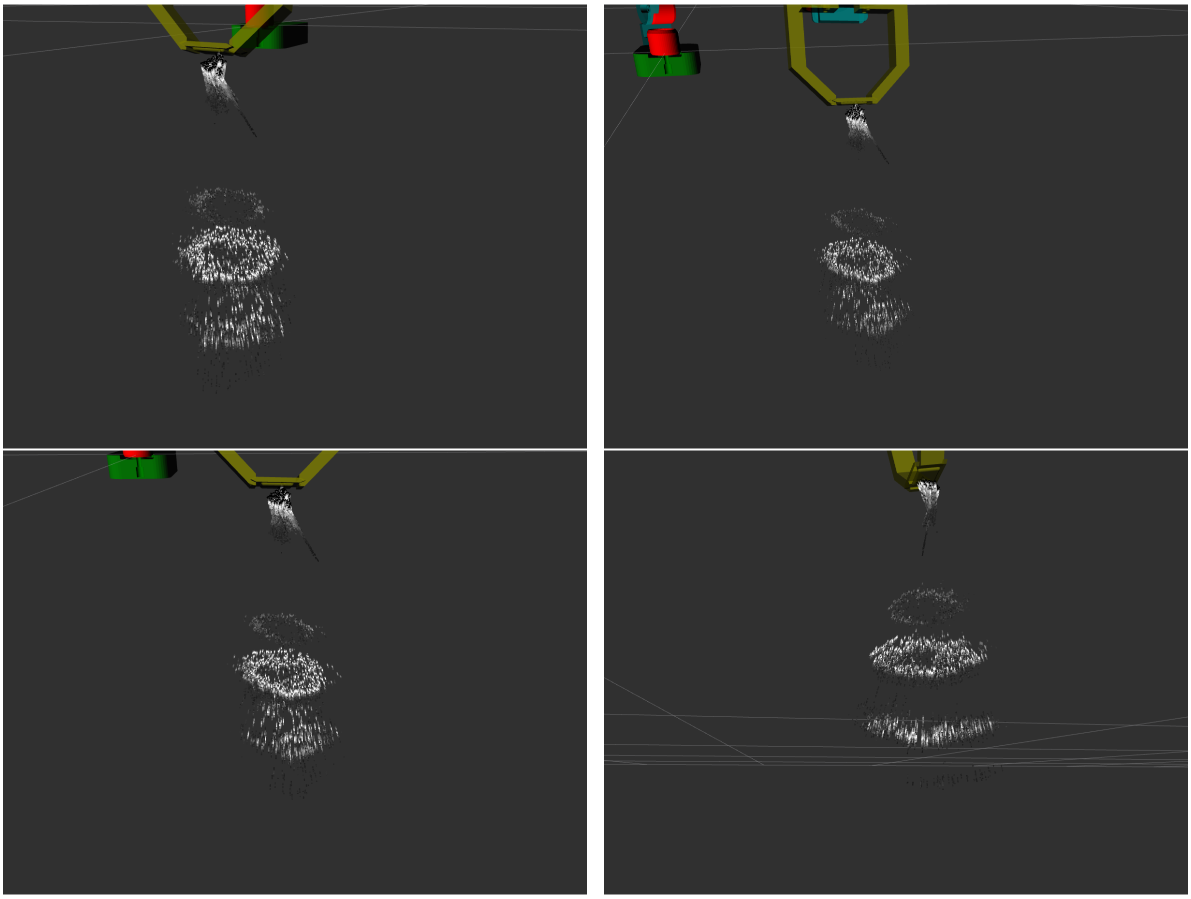


Figure 4.9: 3D point cloud of ultrasound sweep with the robotic arm - Phantom 1

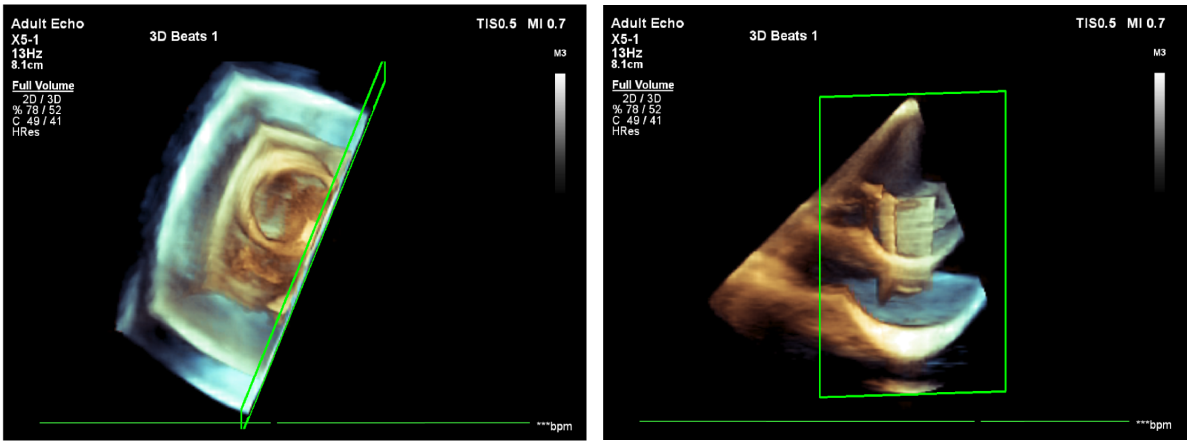


Figure 4.10: 3D ultrasound scan acquired with the Philips EPIQ 7 ultrasound machine - Phantom 1

Phantom 1 (see figure 4.8) was used for the scans shown in figures 4.9 and 4.10. It has a plastic object with a cylindrical shaped void embedded in it. The scans from the Philips EPIQ 7 ultrasound machine (figure 4.10) clearly captures this cylindrical shape and its vertical cross section. Figure 4.9 shows the 3D scans obtained using the system designed in this project. It captures sections of the cylindrical shaped void. The top and mid section circular shapes can be seen in all the images shown in figure 4.9. The CMUT was positioned above the cylindrical shaped void of the plastic object in phantom 1

(A.4) and ultrasound sweeps within the range of motion afforded by the length of CMUT's connecting cable were made around that region. These ultrasound sweeps were able to captured the edges of the cylindrical void and also show the void (figure 4.9). The 3D Scans from the EPIQ 7 have a high resolution and captures a lot of details. These scans are subjected to post processing techniques such as interpolation and rendering. Such post processing techniques were not implemented in this design and it explains why there exist spaces or gaps between adjacent points in the point clouds. Also, only a small section of the phantom can be scanned during the construction of the 3D image due to the movement limitations of the CMUT. It has a short connecting cable to the analog front-end and this does not leave sufficient room for movement across a wide region of the phantom.

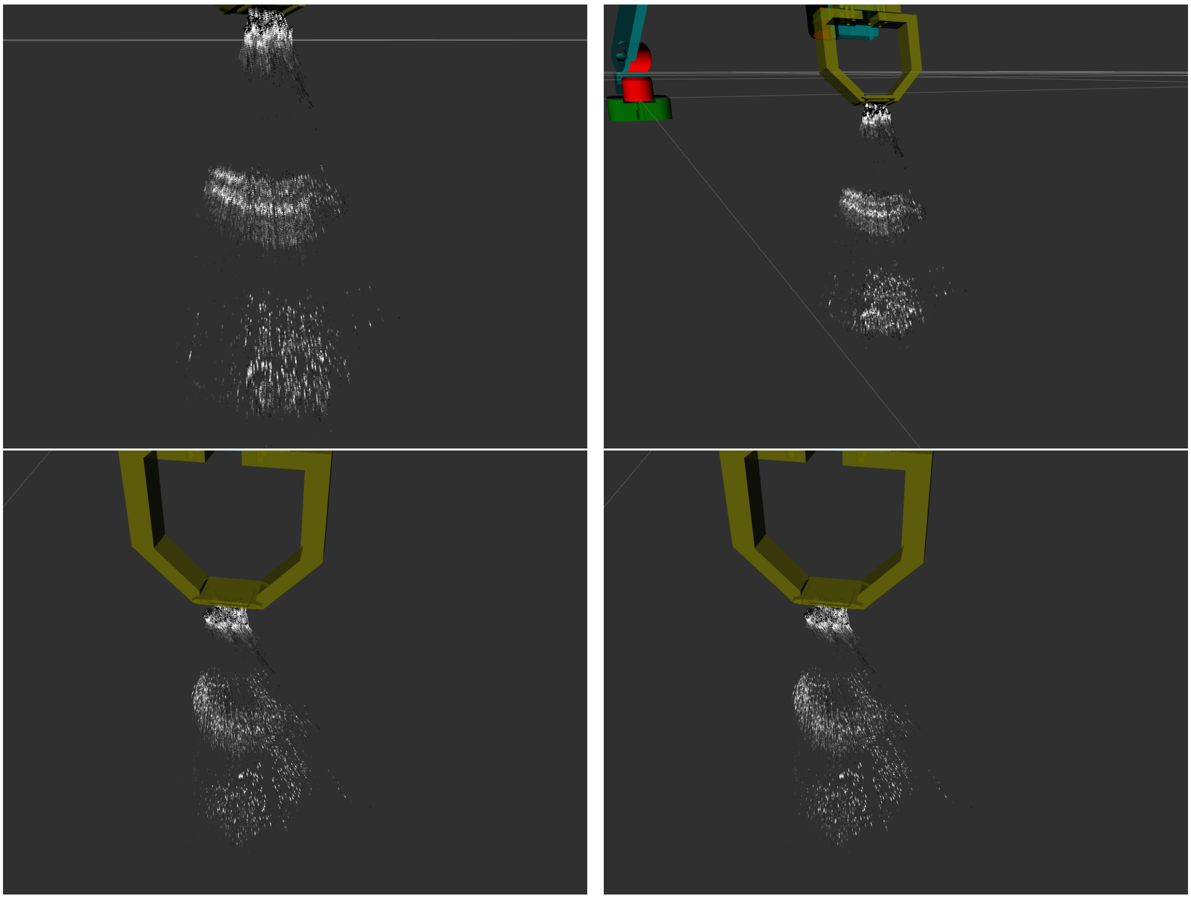


Figure 4.11: 3D point cloud of ultrasound sweep with the robotic arm - Phantom 2

Figure 4.11 and 4.12 are the 3D ultrasound images obtained using Phantom 2 (see figure 4.8)). The scans from the EPIQ 7 provides a more detailed representation of the object inside the phantom. There is cylindrical void within the plastic in the phantom. This was clearly capture by the 3D image from the EPIQ 7 as shown in figure 4.12. Also, the EPIQ scans show the presence of artifacts at the bottom of the 3D image. These artifacts maybe due reverberations - the acoustic beam continuously bouncing off the surface of the phantom container. Comparing the EPIQ 7 scans to that obtained by the prototype, a view from the top shows a circular top surface with a void. It does not capture much detailed information of the morphology of the plastic object embedded in the phantom.

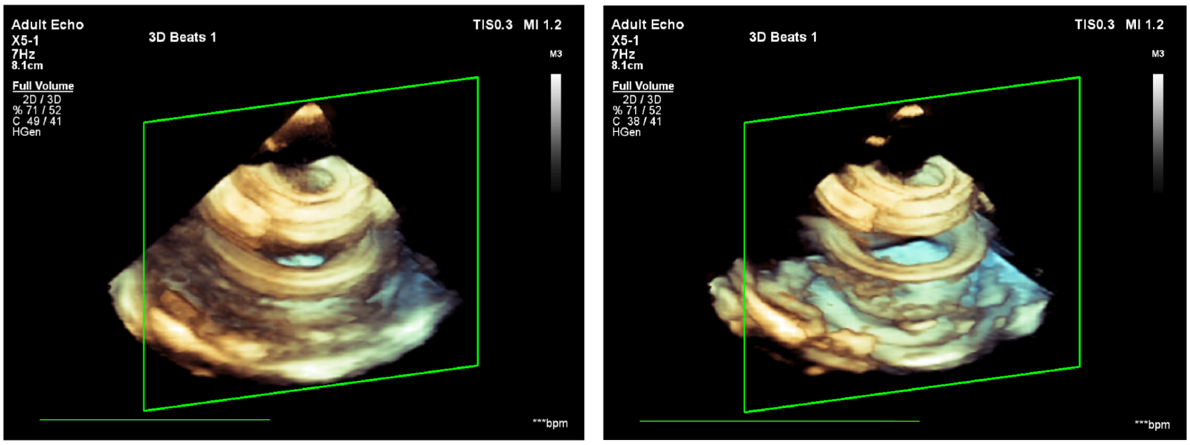


Figure 4.12: 3D ultrasound scan acquired with the Philips EPIQ 7 ultrasound machine - Phantom 2

The results presented demonstrate that interfacing CMUTs with low-end analog front-end and robotic systems can be used to acquire 3D ultrasound images. The comparison of the results between these two devices shows that the prototype presented in this project was able to capture similar 3D representation of the objects in the phantom. This comparison was done by observing the shapes of the 3D images. It however, does not provide enough information to fully describe the shape of the object as compared to scans obtained with the EPIQ 7 ultrasound machine. To provide enough information to represent the object, more points are required in the point cloud. Increasing the number of points also increases the amount of computation required to display the point cloud. There is therefore a trade-off between image quality and computational power. Only pixel intensity values above 2 (grayscale pixel intensity value) were displayed in Rviz. Those below 2 do not carry any information as they represent sections within the phantom where no reflections occurred. This reduced the number of points that have to be displayed and thereby reducing the computational load on the GPU (Graphics Processing Unit). When all the points are shown in Rviz (i.e figure 4.7), with time, the computer (laptop) begins to lag in performance and also affects the response time of the robotic arm. Decreasing the number of points displayed by Rviz solves this problem. The latency between manipulations made by the robotic arm and the visual representation shown in Rviz is negligible hence, manipulations made during 3D image acquisition were visible in real-time in the Rviz. The EPIQ machine creates the images shown in figures 4.10 and 4.12 almost instantaneously while it takes about 20 - 60 seconds to create the images shown in the figures 4.9 and 4.11. This is due to the size of the transducer used and the limited capacity of the analog front-end to process information received by the 32 elements. Hence, it takes the aforementioned time to complete an ultrasound sweep to produce an appreciable 3D image. The node that runs on the WUP (WUP Node) which is responsible for ultrasound data acquisition takes an average of $40.56\ \mu s$ to execute a single pulse-echo cycle. The signal processing node (SP Node) takes an average of $10\ \mu s$ to processes data from one pulse-echo cycle and the node responsible for the conversion of the data to point cloud (PCloud Node) takes an average of $71.5\ \mu s$ to run. The PCloud Node takes the maximum amount of time to execute because apart from point cloud conversion, it also resolves the location and orientation (translation and rotation) of the end effector of the robotic arm and synchronizes it with the scan line obtained from a single pulse-echo cycle.

5

Discussion

The aim of this project was to demonstrate the feasibility of using a 32-channel CMUT in combination with a low-end analog front-end and a robotic arm to acquire 3D ultrasound volumes. The implementation of the design concept followed a distributed system approach where the CMUT and analog front-end are decoupled from the processing and visualization unit. This reduces the amount of computation required by the analog front-end and allows information to be sent over a network to the processing unit. Packages were developed in ROS that enables the real-time acquisition of ultrasound data and the transfer of this data to the visualization environment. The RF ultrasound data received was processed for image reconstruction and the results were displayed in the ROS visualization environment.

Typical linear ultrasound scanners have 128 - 512 piezoelectric elements [50] with each element having a connection to the processing unit. This makes every echo received by each element available for processing. In this project, a 32-channel CMUT which averages the information captured by all the even number elements as one output channel and all the odd number elements as another output channel was used. This is due to the fact that, the analog front-end was not designed to process all the information captured by the 32 channels. CMUT leverages on the well defined semiconductor-industry manufacturing techniques which include large scale wafer-level production which drives down cost, maximizes control of device properties and integration with electronics [51]. One highlight of this project was demonstrating the feasibility of integrating CMUTs with low-end analog electronics. Even though CMUTs have rapidly experienced development in the last decade, only few have had market success. A typical example is the 9000 element 2D CMUT array probe developed by Butterfly Network [52].

Apart from CMUTs application in imaging, it can also be used in monitoring applications. An example is the Doptone, which is a hand-held portable device for fetal monitoring. It uses the Doppler effect to detect fetal heartbeat [53]. This design implementation was able to trigger the pulsing elements, emit ultrasound, receive the echoes and transfer this information to the processing unit via the ROS network. The angle of the ultrasound beam generated by the CMUT was not altered with respect to the transducer elements. Basic filter settings in the analog and digital domain were used to process the signal and this does not provide optimal image quality. Typically, Application-Specific Integrated Circuits (ASIC) are used to handle the high data rate and computational requirements in most ultrasound systems [54].

The location and orientation of scan lines was tracked using a robotic arm. The packages were built in ROS hence the implementation made use of some standard ROS libraries like the joint state publisher. The CMUT used in this project has a very short connecting cable to the analog front-end and this reduces the amount of movement that can be made during scanning thereby limiting the range of coverage. The test results of two custom-made phantoms reveal sections of the objects embedded in the phantoms. The Philips EPIQ 7 ultrasound machine with the Philips X5-1 probe (3040 elements) was used to acquire 3D scans of the phantoms. Comparing 3D scans obtained by both devices, it was obvious that the Philips EPIQ 7 ultrasound machine produced very superior 3D ultrasound images. This was expected as the EPIQ 7 is a high end ultrasound machine. Our design was able to produce 3D images that provide representation of the objects embedded in the phantom however, with less clarity.

To create the 3D US image, spatial compounding, which is a technique that combines co-planar, tomographic ultrasound images captured from different orientations into a single image [55] was used. Spatial compounding involves two distinct approaches - forward-warping and backward-warping. Forward-warping was used in the volumetric reconstruction because it is computationally efficient. It involves the direct projection of acquired ultrasound scan lines (slices) into the volume without any form of interpolation. This however, creates gaps between the slices in the volume and this can be seen in the 3D scans showed under results (figures 4.9 and 4.11). Backward-warping on the other hand is a superior reconstruction technique which calculates the voxel of the volume by considering adjacent slices [56]. It is however, a computationally expensive approach which normally requires graphics hardware to reduce the processing time [56].

The analog front-end used in this project was designed to explore CMUT applications. Ideally, analog front-ends for ultrasound imaging are custom made for such purposes. A maximum of four scan lines per second could be achieved without overloading the ADC (Red Pitaya). Any increase in the aforementioned rate would prevent the ADC buffer from becoming fully filled. This is because, a pulse-echo cycle will not be fully recorded by the buffer before another cycle begins. This has an impact on the time it takes to complete a scan of a region of interest. It takes an average of 20 - 60 seconds to create a 3D image by performing a freehand ultrasound sweep. The construction of the 3D image is dependent on the number point cloud data points available within Rviz. Thus, the faster Rviz can be populated with data points, the faster the 3D image can be constructed. The speed at which the 3D image is constructed can be increased by either increasing the number of scan lines per second, or by upgrading the analog front-end to be able to processing US data received from multiple transducer elements. This will increase the number of data processed by the analog front-end and subsequently increasing the number of data points transferred to the Rviz environment. The improvement of the speed of imaging of this design should be an area of interest for future work as fast ultrasonic imaging is highly desirable in 3D/4D imaging. An example of a domain application is the evaluation of myocardial function by enabling the real-time accurate imaging of the heart movements [57].

Studies using robotic arms to construct 3D images have been reported in literature. A common denominator among all these studies is the use of conventional ultrasound machines and probes. A thorough search of relevant literature did not yield a study where low-end electronics for the purposes of ultrasound imaging was interfaced with a robotic arm to capture 3D images. What has been mostly reported is the use of conventional 2D ultrasound machines and systems that track the probe positioning. The Stanford Ultrasound Imaging and Instrumentation Lab is currently working on a similar project where low cost systems are designed to track the probe positioning [58]. It is apparent that designing low cost 3D ultrasound systems presents constraints and challenges that need to be addressed to demonstrate their feasibility for commercialization. Interfacing robotic arms with conventional ultrasound machines has a unique application also in telesonography which seeks to address the lack of skilled sonographers in many locations globally. This is highly appreciable however, it can be argued that there is a need for the development of low cost, point-of-care and probably disposable ultrasound image technologies. This need is pressing because most people in developing and under developed nations lack access to ultrasound imaging. It is now more than ever the time to redouble efforts and channel resources to developing these systems that aid in healthcare delivery as the World Health Organization (WHO) estimates that by the year 2050, 80 % of all older people will be living in low and middle-income countries [59].

6

Conclusion

We have demonstrated the feasibility of integrating CMUTs, low-end electronics and a robotic arm for 3D ultrasound imaging using ROS. ROS provides an extensive middleware on which packages that contain nodes (executable scripts) were developed. These packages manage the entire ultrasound 3D image construction process. This begins with the emission of ultrasound into a region of interest to the final stage of image visualization. Volumetric reconstruction from freehand sweeps was realized using forward-warping. 3D phantoms made from gel were used to test the performance of the design. Recorded results were then compared to 3D ultrasound images obtained from a high-tier commercial ultrasound machine. While the results from the high-tier commercial ultrasound machine were superior, some sections of the shapes captured by the commercial machine can be seen in the corresponding results obtained by this design. Additionally, it takes about 20 - 60 seconds to form an appreciable 3D US image with our design while the high-tier commercial ultrasound machine creates this 3D images instantaneously. The speed of image acquisition of our design is not on par with that of the commercial system which is to a large extent expected. We have, however, succeeded in building a system that provides 3D ultrasound imaging functionalities. Future work can focus of enhancing the speed of acquisition and improving visualization. This can improve the image quality and subsequently provide a better representation of structures hidden in phantoms. Robotic systems are gradually gaining traction in medical imaging. Success have already been witnessed in their application in surgical procedures. Reports of hand tremor due to prolonged time in taking ultrasound images and constantly maneuvering ultrasound probes, lack of skilled sonographers coupled with the precision and accuracy afforded by robotics systems has caused their integration with ultrasound imaging systems. Robotic systems in ultrasound imaging may need extensive studies before they can make it to the clinical setting but in the opinion of the author, it is highly probable that the future of ultrasound imaging would be characterised by high levels of automation.

A

Supplementary Info

A.1. CMUT

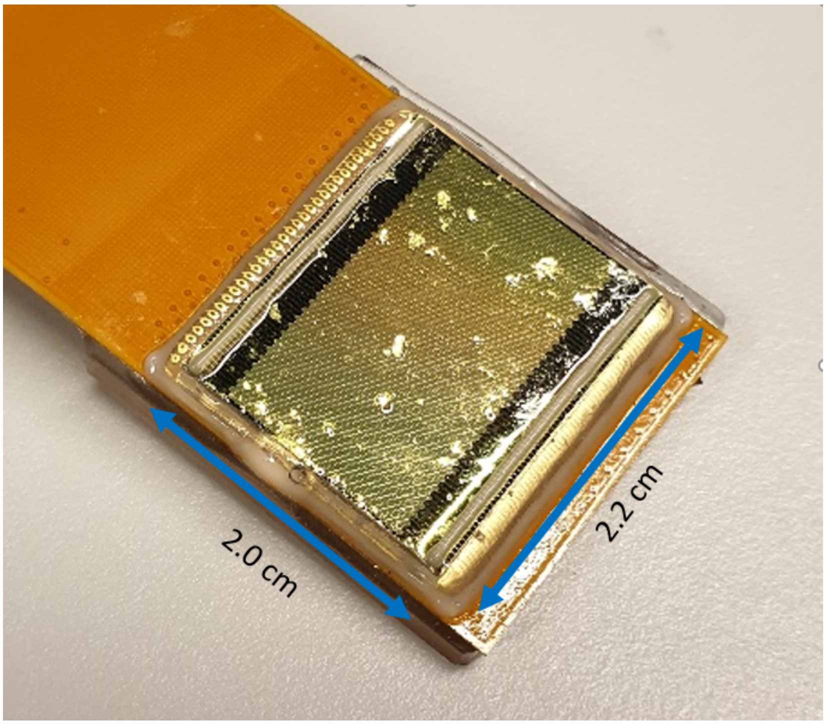


Figure A.1: CMUT array with 32 active transducer elements

A.2. Red Pitaya

The Red Pitaya was configured via a python script to initialize the oscilloscope parameters and setup the burst parameters. The burst signal generated by the Red Pitaya causes the microcontroller to gen-

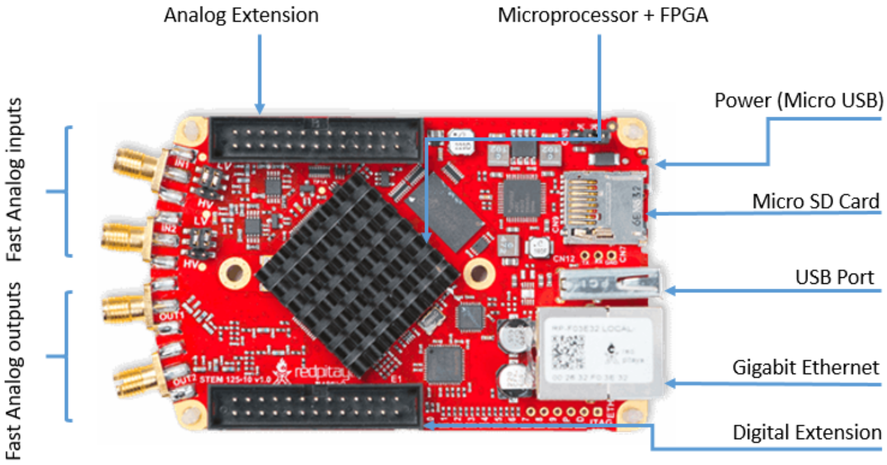


Figure A.2: The Red Pitaya is an open source oscilloscope and signal generator Zynq® field programmable gate array (FPGA). It is powered by a Linus operating system and can be controlled via SSH protocol, usb-serial console and a web browser using its server functionality.

Table A.1: Red Pitaya Specifications

Storage	MicroSD
Fast ADC	Dual channel, 14-bit, 125MS/s
Fast DAC	Dual channel, 14-bit, 125MS/s
Power	5V x 2A maximum, 0.9A
Connectivity	Wifi dongle, Ethernet, USB Console
RAM	512 MB
Processor	Xilinx Zynq-7010 SoC ARM dual core CPU and Artix 7 FPGA

erate ultrasound signals. The python script for this configuration is shown below.

```python
import redpitaya_scpi as scpi
from time import sleep

def setupScope(rps_1):

    #SETUP SIGNAL GENERATION PARAMETERS

    rps_1.tx_txt('GEN:RST')
    rps_1.tx_txt('SOUR1:FUNC SQUARE')
    rps_1.tx_txt('SOUR1:FREQ:FIX 500')
    rps_1.tx_txt('SOUR1:VOLT 1')

    #SETUP BURST PARAMETERS

    rps_1.tx_txt('SOUR1:BURS:STAT ON')
    rps_1.tx_txt('SOUR1:BURS:NCYC 1')
    rps_1.tx_txt('SOUR1:BURS:NOR 1')
    rps_1.tx_txt('SOUR1:BURS:INT:PER 5000')

    #RESET OSCILLOSCOPE - ACQUISITION

    rps_1.tx_txt('ACQ:RST')

    #SETUP ACQUISITION PARAMETERS

    rps_1.tx_txt('ACQ:DEC 1')
    rps_1.tx_txt('ACQ:TRIG:LEV 100 mV')
    rps_1.tx_txt('ACQ:TRIG:DLY 8192')
    rps_1.tx_txt('ACQ:SOUR1:GAIN LV')
    rps_1.tx_txt('ACQ:AVG OFF')

    #START ACQUISITION

    rps_1.tx_txt('ACQ:START')
    rps_1.tx_txt('ACQ:TRIG CH1_PE')
    rps_1.tx_txt('OUTPUT1:STATE ON')

    return rps_1
```

A.3. PCB

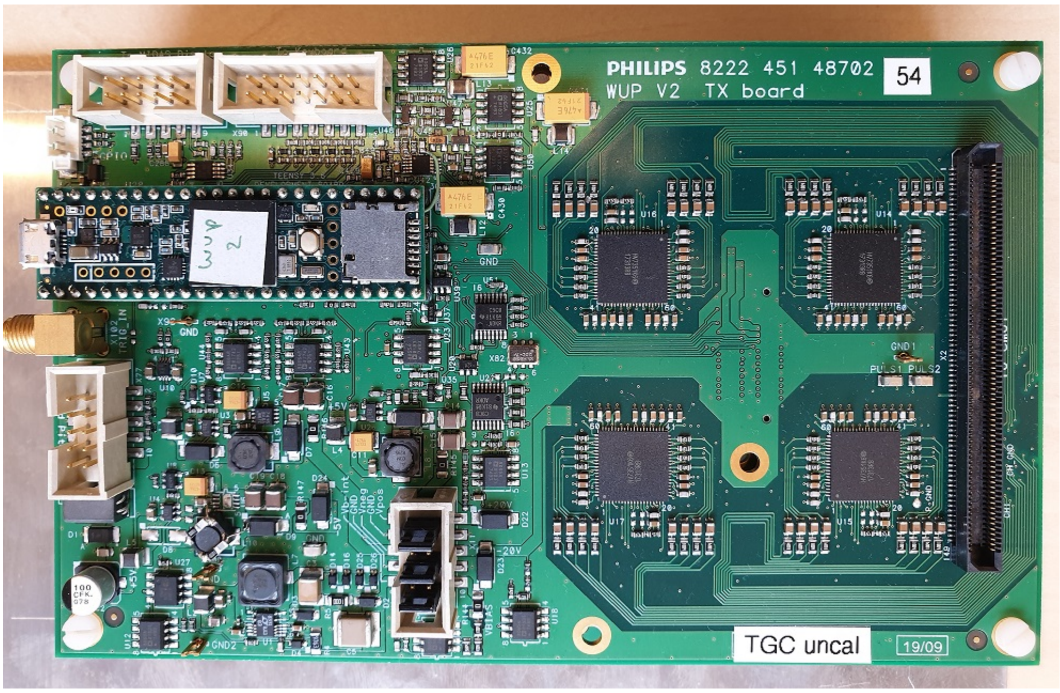
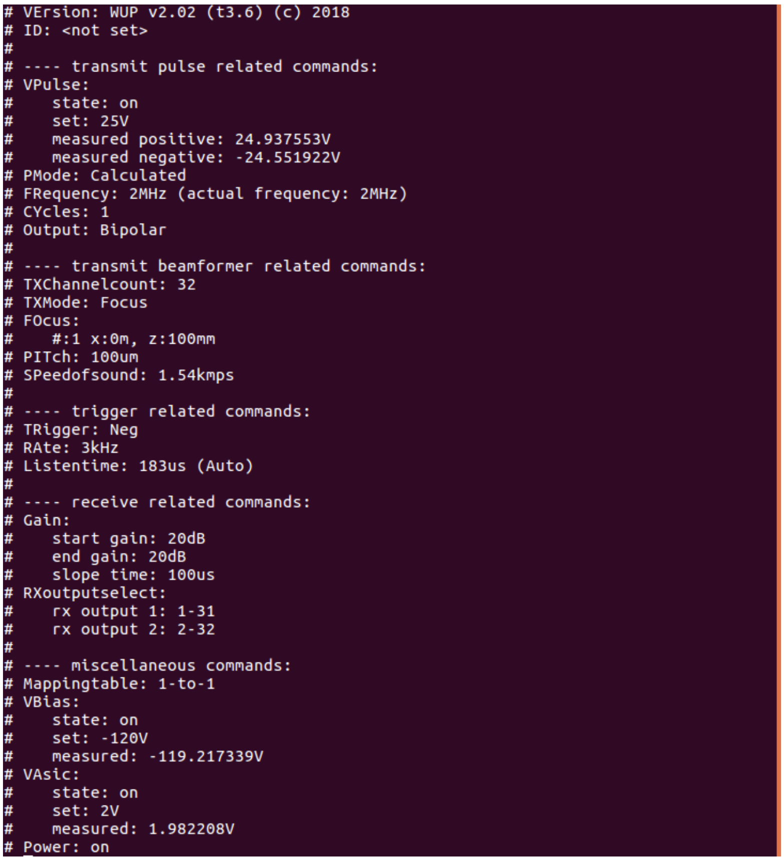


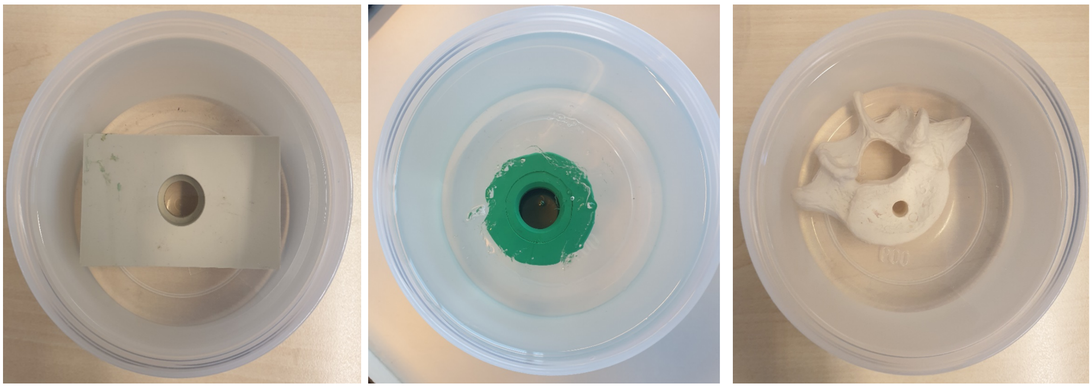
Figure A.3: Philips Printed Circuit Board (PCB)



```
# VErsion: WUP v2.02 (t3.6) (c) 2018
# ID: <not set>
#
# ---- transmit pulse related commands:
# VPulse:
#    state: on
#    set: 25V
#    measured positive: 24.937553V
#    measured negative: -24.551922V
# PMode: Calculated
# FRequency: 2MHz (actual frequency: 2MHz)
# CYcles: 1
# Output: Bipolar
#
# ---- transmit beamformer related commands:
# TXChannelcount: 32
# TXMode: Focus
# FOcus:
#    #:1 x:0m, z:100mm
# PITch: 100um
# SPeedofsound: 1.54kmps
#
# ---- trigger related commands:
# TRigger: Neg
# RAte: 3kHz
# Listentime: 183us (Auto)
#
# ---- receive related commands:
# Gain:
#    start gain: 20dB
#    end gain: 20dB
#    slope time: 100us
# RXoutputselect:
#    rx output 1: 1-31
#    rx output 2: 2-32
#
# ---- miscellaneous commands:
# Mappingtable: 1-to-1
# VBias:
#    state: on
#    set: -120V
#    measured: -119.217339V
# VAsic:
#    state: on
#    set: 2V
#    measured: 1.982208V
# Power: on
```

Figure A.4: Snapshot of PCB Application Program Interface (API) commands

A.4. Phantoms



(a) Phantom 1

(b) Phantom 2

(c) Phantom 3

Figure A.5: Gel Phantoms made from candle wax. Embedded in the phantoms are plastics objects.

B

Robotic Operating System (ROS)

B.1. Unified Robot Description Format (URDF)

URDF provides a description of the robotic arm. It organizes its components in a tree structure and provides a relation between various components in the tree structure. Figure B.1 [60] shows how a sample URDF description represents the robot. Links are what defines the relationship between one joint and another. A joint can have multiple links. A portion of the URDF of the robotic arm used in this project is shown below. Describing the robot in XML consists of many lines of code and displaying all the lines of code may be overwhelming.

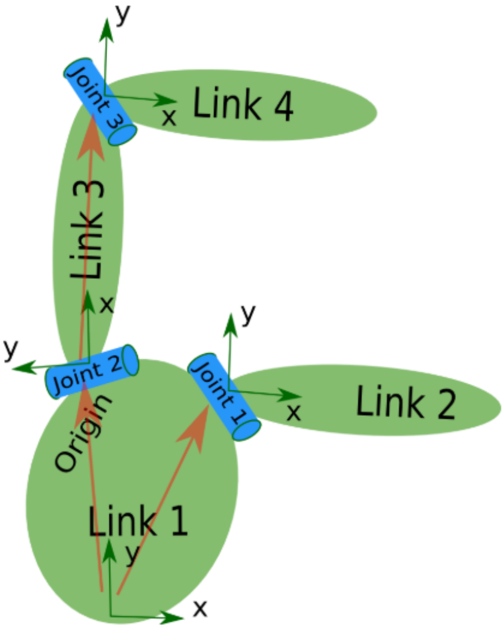


Figure B.1: A URDF description of a simple robot showing how links and joints are connected

```
<robot name= ”robotic_arm”>

<material name= ”green”>
<color rgba= ”0 0.5 0 0.9”/>
</material>

<material name= ”teal”>
<color rgba= ”0 0.5 0.5 0.9”/>
</material>

<material name= ”olive”>
<color rgba= ”0.5 0.5 0 0.8”/>
</material>

<link name=”ground_plate”>
    <visual>
      <origin rpy=”0.0 0 0” xyz=”0 0 0”/>
      <geometry>
        <mesh filename=”package://robotic_arm/meshes/mm_groundplate.stl”/>
      </geometry>
      <material name=”green” />
    </visual>
  </link>

  <joint name=”ground_to_bearing3” type=”fixed”>
    <parent link=”ground_plate”/>
    <child link=”bearing3”/>
    <origin rpy=”-1.5707963268 0 0” xyz=”0 -0.0 0.008”/>
  </joint>

  <link name=”bearing3”>
    <visual>
      <origin rpy=”0.0 0 0” xyz=”0 0 0”/>
      <geometry>
        <mesh filename=”package://robotic_arm/meshes/mm_bearing.stl”/>
      </geometry>
    </visual>
  </link>

  <joint name=”bearing3_to_connectionblock” type=”continuous”>
    <parent link=”bearing3”/>
    <child link=”connectionblock”/>
    <origin xyz=”0 0.000 0.0000”/>
    <axis xyz=”0 1 0”/>
  </joint>

  <link name=”connectionblock”>
    <visual>
      <origin rpy=”1.5707963268 0 0” xyz=”-0.01 0 0”/>
      <geometry>
```

```
        <mesh filename=”package://robotic_arm/meshes/mm_connectionblock.stl”/>
      </geometry>
      <material name=”teal” />
    </visual>
  </link>


</robot>
```

B.2. ROS Nodes

B.2.1. WUP Node

```
 1  #!/usr/bin/env python
 2
 3  import rospy
 4  import redpitaya_scpi as scpi
 5
 6  from std_msgs.msg import String
 7  from ultrasound.msg import Floats
 8  from run import run
 9
10  rps_1 = scpi.scpi(”192.168.1.61”)
11
12  def us_publisher(rps_1):
13          pub = rospy.Publisher('ultrasound_signal', Floats, queue_size=1)
14          rospy.init_node('us_publisher', anonymous = True)
15          rate = rospy.Rate(3)
16          us_data = Floats()
17
18          while not rospy.is_shutdown():
19                  data = run(rps_1)
20                  us_data.data = data
21                  rospy.loginfo(”Data Published”)
22                  pub.publish(us_data)
23                  rate.sleep()
24
25
26  if __name__ == '__main__':
27
28          try:
29              us_publisher(rps_1)
30          except rospy.ROSInterruptException:
31              pass
```

The WUP node imports a run function which is responsible for pulse-echo. The pulse-echo data received (line 19) is converted into a ROS message and published to the US Topic. This data then becomes accessible to all the nodes within the ROS network.

B.2.2. SP Node

```python
#!/usr/bin/env python

import rospy
import numpy as np
from skimage.transform import resize

from std_msgs.msg import String
from ultrasound.msg import Floats
from signal_processing import signal_processing

pub = rospy.Publisher('image_data_publisher', Floats, queue_size=1)

def receive_us_data(us_data):

    us_data_topublish = Floats()
    image_data = signal_processing(np.asarray(us_data))
    image_data = resize(image_data, (250, 1))
    us_data_topublish.data = image_data
    rospy.loginfo(”Data Published”)
    pub.publish(us_data_topublish)


def callback(data):
    receive_us_data(data.data)


def us_subscriber():
    rospy.init_node('us_subscriber', anonymous=True)
    rospy.Subscriber('ultrasound_signal', Floats, callback)
    rospy.spin()


if __name__ == '__main__':
    us_subscriber()
```

```python
import numpy as np

from scipy.signal import hilbert
from scipy.signal import savgol_filter
from scipy.signal import butter, lfilter

def averagingFilter(data):
    my_data_array_reshaped_filtered = savgol_filter(data, 31, 8)
    return my_data_array_reshaped_filtered

def butter_bandpass(lowcut, highcut, fs, order=5):
    nyq = 0.5 * fs
    low = lowcut / nyq
    high = highcut / nyq
    b, a = butter(order, [low, high], btype='band')
    return b, a

def butter_bandpass_filter(data, lowcut = 1500000, highcut = 2500000, fs = 125000000, order = 5):
    b, a = butter_bandpass(lowcut, highcut, fs, order=order)
    y = lfilter(b, a, data)
    return y

def signal_processing(us_data):

    data_envelope = (np.abs(hilbert(butter_bandpass_filter(averagingFilter(us_data[300:])))))
    data_normalized = (20 * np.log10(1 + (data_envelope)))
    return data_normalized
```

The SP node performs averaging, bandpass filtering and uses Spline interpolation to downsample the signal. The processed signal is published to the IMG Topic.

B.2.3. PCloud Node

```cpp
#include <ros/ros.h>
// PCL specific includes
#include <visualization_msgs/Marker.h>

#include <sensor_msgs/PointCloud2.h>
#include <sensor_msgs/PointCloud.h>
#include <std_msgs/Float32MultiArray.h>
#include <pcl_conversions/pcl_conversions.h>
#include <pcl/point_cloud.h>
#include <pcl/point_types.h>
#include ”ultrasound/Floats.h”
#include ”std_msgs/Int32MultiArray.h”
#include ”std_msgs/Float32.h”

#include <tf2_ros/transform_listener.h>
#include <tf2_geometry_msgs/tf2_geometry_msgs.h>
#include <tf2_ros/static_transform_broadcaster.h>
#include <tf2_ros/transform_broadcaster.h>
#include <tf2/LinearMath/Quaternion.h>

//some working defaults in case nobody sets these values
#define WIDTH 0.100
#define HEIGHT 0.100
#define LENGTH 0.400
#define RESOLUTION 0.001

double g_length = 0.400;  // work area size
double g_height = 0.100;  // work area size
double g_width = 0.100;   // work area size
double g_resolution;
```

```

ros::Publisher pub;
ros::Publisher g_workarea_pub;
ros::Publisher pub_location_array;


void tocloud_cb (const ultrasound::Floats::ConstPtr& input)
{
  static tf2_ros::Buffer            g_tfBuffer;
  static tf2_ros::TransformListener g_tfListener(g_tfBuffer);

  static bool marker_initialized = false;
  const int volumeWidth  = (int)(g_width  / g_resolution);
  const int volumeLength = (int)(g_length / g_resolution);
  const int volumeHeight = (int)(g_height / g_resolution);

  geometry_msgs::TransformStamped transformStamped;

  if (!marker_initialized){

  fprintf(stderr, "Creating workarea... %dx%dx%d\n",volumeWidth,volumeLength,volumeHeight );
  visualization_msgs::Marker marker;
  // Set the frame ID and timestamp.
  marker.header.frame_id = "world";
  marker.header.stamp = ros::Time::now();
  marker.ns = "";
  marker.id = 0;
  marker.type = visualization_msgs::Marker::CUBE;
  marker.action = visualization_msgs::Marker::ADD;
  // Set the scale of the marker -- 1x1x1 here means 1m on a side
  marker.scale.x = g_length;
  marker.scale.y = g_width;
  marker.scale.z =  0.110;//g_height;
  // Set the color -- be sure to set alpha to something non-zero!
  marker.color.r = 0.7;
  marker.color.g = 0.7;
  marker.color.b = 1.0;
  marker.color.a = 0.7;

  // Set the pose of the marker.  This is a full 6DOF pose relative to the frame/time specified in the header
  marker.pose.position.x = g_length/2;
  marker.pose.position.y = g_width/2;
  marker.pose.position.z = g_height/2;
  marker.pose.orientation.x = 0.0;
  marker.pose.orientation.y = 0.0;
  marker.pose.orientation.z = 0.0;
  marker.pose.orientation.w = 1.0;
  marker.frame_locked = false;
  marker.lifetime = ros::Duration(0.0);

  g_workarea_pub.publish(marker);
  marker_initialized = true;
  fprintf(stderr, ” -- done \n”);
  }

  try

  {
    transformStamped = g_tfBuffer.lookupTransform(”ground_plate”, ”xplanes_origin”, ros::Time(0));
  }

  catch (tf2::TransformException ex)
  {
    fprintf(stderr, ”no transform yet\n”);
    return;
  }


  fprintf(stderr, ”got image data\n”);
  sensor_msgs::PointCloud cloud;

  cloud.header.frame_id = ”xplanes_origin”;
  fprintf(stderr, ”set header\n”);
  cloud.header.stamp = ros::Time::now();
  fprintf(stderr, ”set timestamp\n”);

  int height = 1;
  int width = 250;

  cloud.points.resize(height * width);

    cloud.channels.resize(1);
    cloud.channels[0].name = ”intensities”;
    cloud.channels[0].values.resize(height * width);

  int maxSize = (int)(input->data.size());
  fprintf(stderr, ”image_data_size: %d\n”, maxSize);

  int c = 0;

    for(int i =0; i < height; i++){

      for(int a = 0; a < width; a++){
          cloud.points[c].x = i*0.0005;
          cloud.points[c].y = a*0.0005;
          cloud.points[c].z = 0;

          cloud.channels[0].values[c] = input->data[c];
          c++;
      }

  }
```

```

  pub.publish (cloud);

}

int main (int argc, char** argv){
  // Initialize ROS
  ros::init (argc, argv, ”point_cloud_publisher”);
  ros::NodeHandle nh;

  // Create a ROS subscriber for the input point cloud
  ros::Subscriber sub = nh.subscribe(”image_data_publisher”, 1, tocloud_cb);

  // Create a ROS publisher for the output point cloud
   pub = nh.advertise<sensor_msgs::PointCloud> (”PointCloud_publisher”, 10);
   pub_location_array = nh.advertise<ultrasound::Floats>(”location_publisher”, 10);
   g_workarea_pub = nh.advertise<visualization_msgs::Marker>(”xplane_workarea”, 1);

  ros::spin ();
}
```

B.2.4. Arm Node

```python
#!/usr/bin/env python

import rospy
import math
import std_msgs
import serial

from std_msgs.msg import String
from sensor_msgs.msg import JointState


ser = serial.Serial('/dev/ttyUSB0', 115200)
joint_names = ['bearing3_to_connectionblock', 'bearing0_to_upperarm', 'bearing1_to_lowerarm',
                'bearing2_to_connectionperpendicular',
                'bearing4_to_connectionstrip', 'bearing5_to_connectionplate']


joint_offset = [-813, -223, -302, -613, -15, -273]

def joint_pub():

 while not rospy.is_shutdown():
   data = ser.readline()

   if data:
     sensorStringList = data.split(',')

     try:

      positionList = [((float(value)-float(offset))*(math.pi/1024.0))
                      for value, offset in zip(sensorStringList, joint_offset)]

     except ValueError:
      continue

     js = JointState()
     js.header =  std_msgs.msg.Header()
     js.header.stamp = rospy.Time.now()
     js.header.frame_id = ''
     js.name = joint_names
     js.position = positionList

   pub.publish(js)
   rospy.sleep(0.001)



if __name__ == '__main__':

  try:

    pub = rospy.Publisher('joint_states', JointState, queue_size=1)
    rospy.init_node('joint_publisher')
    joint_pub()

  except rospy.ROSInterruptException:
    pass
```

Bibliography

[1] Q. Huang and Z. Zeng, “A Review on Real-Time 3D Ultrasound Imaging Technology,” *BioMed Research International*, vol. 2017, 2017.

[2] V. S. Vasudeva, M. Abd-El-Barr, Y. A. Pompeu, A. Karhade, M. W. Groff, and Y. Lu, “Use of Intraoperative Ultrasound During Spinal Surgery,” *Global Spine Journal*, vol. 7, no. 7, pp. 648–656, 2017.

[3] E. M. Boctor, G. Fischer, M. A. Choti, G. Fichtinger, and R. H. Taylor, “A dual-armed robotic system for intraoperative ultrasound guided hepatic ablative therapy: A prospective study,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2004, no. 3, 2004, pp. 2517–2522.

[4] E. M. Boctor, M. A. Choti, E. C. Burdette, and R. J. Webster, “Three-dimensional ultrasound-guided robotic needle placement: An experimental evaluation,” *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 4, no. 2, pp. 180–191, jun 2008. [Online]. Available: http://doi.wiley.com/10.1002/rcs.184

[5] D. L. Miller, N. B. Smith, M. R. Bailey, G. J. Czarnota, K. Hynynen, and I. R. S. Makin, “Overview of therapeutic ultrasound applications and safety considerations,” *Journal of Ultrasound in Medicine*, vol. 31, no. 4, pp. 623–634, 2012.

[6] S. Sippel, K. Muruganandan, A. Levine, and S. Shah, “Review article: Use of ultrasound in the developing world,” *International Journal of Emergency Medicine*, vol. 4, no. 1, p. 72, dec 2011. [Online]. Available: https://intjem.biomedcentral.com/articles/10.1186/1865-1380-4-72

[7] J. E. Sutherland, D. Sutphin, K. Redican, and F. Rawlins, “Telesonography: Foundations and future directions,” *Journal of Ultrasound in Medicine*, vol. 30, no. 4, pp. 517–522, 2011.

[8] Q. Huang, F. Yang, L. Liu, and X. Li, “Automatic segmentation of breast lesions for interaction in ultrasonic computer-aided diagnosis,” *Information Sciences*, vol. 314, pp. 293–310, sep 2015.

[9] Q. Huang, Y. Luo, and Q. Zhang, “Breast ultrasound image segmentation: a survey,” pp. 493–507, mar 2017.

[10] A. Fenster, D. B. Downey, and H. N. Cardinal, “Three-dimensional ultrasound imaging,” may 2001.

[11] A. M. Priester, S. Natarajan, and M. Culjat, “Robotic ultrasound systems in medicine,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 60, no. 3, pp. 507–523, 2013.

[12] M. Z. Mahmoud, M. Aslam, M. Alsaadi, M. A. Fagiri, and B. Alonazi, “Evolution of Robot-assisted ultrasound-guided breast biopsy systems,” *Journal of Radiation Research and Applied Sciences*, vol. 11, no. 1, pp. 89–97, 2018. [Online]. Available: https://doi.org/10.1016/j.jrras.2017.11.005

[13] P. Arbeille, J. Ruiz, P. Herve, M. Chevillot, G. Poisson, and F. Perrotin, “Fetal tele-echography using a robotic arm and a satellite link,” *Ultrasound in Obstetrics and Gynecology*, vol. 26, no. 3, pp. 221–226, 2005.

[14] K. Liang, A. J. Rogers, E. D. Light, D. Von Allmen, and S. W. Smith, “Simulation of autonomous robotic multiple-core biopsy by 3D ultrasound guidance,” *Ultrasonic Imaging*, vol. 32, no. 2, pp. 118–127, apr 2010. [Online]. Available: http://journals.sagepub.com/doi/10.1177/016173461003200205

[15] H. T. Sen, M. A. Lediju Bell, Y. Zhang, K. Ding, E. Boctor, J. Wong, I. Iordachita, and P. Kazanzides, “System Integration and in Vivo Testing of a Robot for Ultrasound Guidance and Monitoring during Radiotherapy,” *IEEE Transactions on Biomedical Engineering*, vol. 64, no. 7, pp. 1608–1618, jul 2017.

[16] H. T. Şen, A. Cheng, K. Ding, E. Boctor, J. Wong, I. Iordachita, and P. Kazanzides, “Cooperative control with ultrasound guidance for radiation therapy,” *Frontiers Robotics AI*, vol. 3, no. AUG, pp. 1–12, 2016.

[17] P. Abolmaesumi, S. E. Salcudean, W. H. Zhu, M. R. Sirouspour, and S. P. DiMaio, “Image-guided control of a robot for medical ultrasound,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 11–23, 2002.

[18] Q. Huang, J. Lan, and X. Li, “Robotic Arm Based Automatic Ultrasound Scanning for Three-Dimensional Imaging,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 1173–1182, feb 2019.

[19] Q. H. Huang, Z. Yang, W. Hu, L. W. Jin, G. Wei, and X. Li, “Linear tracking for 3-D medical ultrasound imaging,” *IEEE Transactions on Cybernetics*, vol. 43, no. 6, pp. 1747–1754, dec 2013.

[20] I. A. Elhelf, H. Albahar, U. Shah, A. Oto, E. Cressman, and M. Almekkawy, “High intensity focused ultrasound: The fundamentals, clinical applications and research trends,” pp. 349–359, jun 2018.

[21] A. Cafarelli, M. Mura, A. Diodato, A. Schiappacasse, M. Santoro, G. Ciuti, and A. Menciassi, “A computer-assisted robotic platform for Focused Ultrasound Surgery: Assessment of high intensity focused ultrasound delivery,” in *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2015-Novem. Institute of Electrical and Electronics Engineers Inc., nov 2015, pp. 1311–1314.

[22] J. Seo, N. Koizumi, M. Mitsuishi, and N. Sugita, “Ultrasound image based visual servoing for moving target ablation by high intensity focused ultrasound,” *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 13, no. 4, pp. 1–7, 2017.

[23] C. Yiallouras, K. Ioannides, T. Dadakova, M. Pavlina, M. Bock, and C. Damianou, “Three-axis MR-conditional robot for high-intensity focused ultrasound for treating prostate diseases transrectally,” *Journal of Therapeutic Ultrasound*, vol. 3, no. 1, pp. 1–10, 2015.

[24] R. Chopra, N. Baker, V. Choy, A. Boyes, K. Tang, D. Bradwell, and M. J. Bronskill, “MRI-compatible transurethral ultrasound system for the treatment of localized prostate cancer using rotational control,” *Medical Physics*, vol. 35, no. 4, pp. 1346–1357, 2008.

[25] K. D. Price, V. W. Sin, C. Mougenot, S. Pichardo, T. Looi, A. C. Waspe, and J. M. Drake, “Design and validation of an MR-conditional robot for transcranial focused ultrasound surgery in infants,” *Medical Physics*, vol. 43, no. 9, pp. 4983–4995, aug 2016. [Online]. Available: http://doi.wiley.com/10.1118/1.4955174

[26] V. G. Mallapragada, N. Sarkar, and T. K. Podder, “Robot-assisted real-time tumor manipulation for breast biopsy,” *IEEE Transactions on Robotics*, vol. 25, no. 2, pp. 316–324, 2009.

[27] H. Bassan, T. Hayes, R. V. Patel, and M. Moallem, “A novel manipulator for 3D ultrasound guided percutaneous needle insertion,” *Proceedings - IEEE International Conference on Robotics and Automation*, no. April, pp. 617–622, 2007.

[28] S. Merouche, L. Allard, E. Montagnon, G. Soulez, P. Bigras, and G. Cloutier, “A robotic ultrasound scanner for automatic vessel tracking and three-dimensional reconstruction of b-mode images,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 63, no. 1, pp. 35–46, jan 2016.

[29] C. Freschi, E. Troia, V. Ferrari, G. Megali, A. Pietrabissa, and F. Mosca, “Ultrasound guided robotic biopsy using augmented reality and human-robot cooperative control,” in *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*, 2009, pp. 5110–5113.

[30] J. Bax, D. Smith, L. Bartha, J. Montreuil, S. Sherebrin, L. Gardi, C. Edirisinghe, and A. Fenster, “A compact mechatronic system for 3D ultrasound guided prostate interventions,” *Medical Physics*, vol. 38, no. 2, pp. 1055–1069, 2011.

[31] W. F. Abdelmoaty, C. M. Dunst, C. Neighorn, L. L. Swanstrom, and C. W. Hammill, “Robotic-assisted versus laparoscopic unilateral inguinal hernia repair: a comprehensive cost analysis,” *Surgical Endoscopy*, vol. 33, no. 10, pp. 3436–3443, oct 2019.

[32] A. S. Savoia and G. Caliano, “MEMS-based transducers (CMUT) for medical ultrasound imaging,” in *Frontiers of Medical Imaging*. World Scientific Publishing Co., jan 2014, pp. 445–464.

[33] B. T. Khuri-Yakub and Ö. Oralkan, “Capacitive micromachined ultrasonic transducers for medical imaging and therapy,” *Journal of Micromechanics and Microengineering*, vol. 21, no. 5, may 2011.

[34] “Capacitive micromachined ultrasonic transducers | Philips Innovation Services.” [Online]. Available: https://www.innovationservices.philips.com/looking-expertise/mems-micro-devices/mems-applications/capacitive-micromachined-ultrasonic-transducers-cmut/

[35] “3.1.1. STEMlab boards comparison — Red Pitaya STEMlab 0.97 documentation.” [Online]. Available: https://redpitaya.readthedocs.io/en/latest/developerGuide/125-10/vs.html

[36] “Teensy 3.6 Australia.” [Online]. Available: https://core-electronics.com.au/teensy-3-6.html

[37] C. A. Sennoga, “Ultrasound imaging,” in *Bioengineering Innovative Solutions for Cancer*. Elsevier, jan 2020, pp. 123–161. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/B9780128138861000073

[38] “ROS.org | Core Components.” [Online]. Available: https://www.ros.org/core-components/

[39] R. W. Schafer, “What is a savitzky-golay filter?” *IEEE Signal Processing Magazine*, vol. 28, no. 4, pp. 111–117, 2011.

[40] W. Jung, “Op Amp Applications Handbook (Analog Devices Series),” 2004. [Online]. Available: https://www.analog.com/en/education/education-library/op-amp-applications-handbook.html

[41] M. A. Hassan and Y. M. Kadah, “Digital Signal Processing Methodologies for Conventional Digital Medical Ultrasound Imaging System,” *American Journal of Biomedical Engineering*, vol. 2013, no. 1, pp. 14–30, 2013. [Online]. Available: http://journal.sapub.org/ajbe

[42] “Ultrasound Image Computation - WikiSonix.” [Online]. Available: http://www.ultrasonix.com/wikisonix/index.php/Ultrasound{_}Image{_}Computation

[43] C. Wachinger, T. Klein, and N. Navab, “The 2D analytic signal for envelope detection and feature extraction on ultrasound images,” *Medical Image Analysis*, vol. 16, no. 6, pp. 1073–1084, aug 2012.

[44] “3D_RFUltrasound_Reconstruction/RFtoBmode.png at master · TJKlein/3D_RFUltrasound_Reconstruction.” [Online]. Available: https://github.com/TJKlein/3D{_}RFUltrasound{_}Reconstruction/blob/master/RFtoBmode.png

[45] E. H. Meijering, “Spline interpolation in medical imaging: Comparison with other convolution-based approaches,” in *European Signal Processing Conference*, vol. 2015-March, no. March, 2000. [Online]. Available: https://ieeexplore-ieee-org.tudelft.idm.oclc.org/document/7075214

[46] T. Foote, “Tf: The transform library,” *IEEE Conference on Technologies for Practical Robot Applications, TePRA*, 2013.

[47] M. Ali, D. Magee, and U. Dasgupta, “Signal processing overview of ultrasound systems for medical imaging,” *SPRAB12, Texas ...*, no. November, pp. 1–27, 2008. [Online]. Available: http://cdn.medicexchange.com/images/whitepaper/signal-processing-overview-of-ultrasound-systems.pdf

[48] H. Uhrmann, R. Kolm, and H. Zimmermann, “Analog Filters,” *Springer Series in Advanced Microelectronics*, vol. 45, pp. 3–11, 2014.

[49] T. L. Szabo, *Diagnostic Ultrasound Imaging: Inside Out: Second Edition*. Elsevier Inc., 2004.

[50] W. Wersing and R. Lerch, “Ultrasonic Imaging,” in *Springer Series in Materials Science*. Springer Verlag, 2008, vol. 114, pp. 199–221. [Online]. Available: https://link-springer-com.tudelft.idm.oclc.org/chapter/10.1007/978-3-540-68683-5{_}8

[51] I. Wygant, “A comparison of CMUTs and piezoelectric transducer elements for 2D medical imaging based on conventional simulation models,” in *IEEE International Ultrasonics Symposium, IUS*, 2011, pp. 100–103.

[52] K. Brenner, A. S. Ergun, K. Firouzi, M. F. Rasmussen, Q. Stedman, and B. Khuri-Yakub, “Advances in capacitive micromachined ultrasonic transducers,” *Micromachines*, vol. 10, no. 2, pp. 1–27, 2019.

[53] S. A. Alnuaimi, S. Jimaa, and A. H. Khandoker, “Fetal cardiac doppler signal processing techniques: Challenges and future research directions,” dec 2017. [Online]. Available: /pmc/articles/PMC5743703/?report=abstracthttps://www-ncbi-nlm-nih-gov.tudelft.idm.oclc.org/pmc/articles/PMC5743703/

[54] F. K. Schneider, A. Agarwal, Y. M. Yoo, T. Fukuoka, and Y. Kim, “A fully programmable computing architecture for medical ultrasound machines,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 2, pp. 538–540, mar 2010.

[55] R. R. Entrekin, J. R. Jago, and S. C. Kofoed, “Real-Time Spatial Compound Imaging: Technical Performance in Vascular Applications,” in *Acoustical Imaging*. Kluwer Academic Publishers, may 2006, pp. 331–342. [Online]. Available: https://link-springer-com.tudelft.idm.oclc.org/chapter/10.1007/0-306-47107-8{_}46

[56] A. Karamalis, W. Wein, O. Kutter, and N. Navab, “Fast hybrid freehand ultrasound volume reconstruction,” in *Medical Imaging 2009: Visualization, Image-Guided Procedures, and Modeling*, vol. 7261. SPIE, feb 2009, p. 726114.

[57] J. F. Cruza, M. Perez, J. M. Moreno, and C. Fritsch, “Real Time Fast Ultrasound Imaging Technology and Possible Applications,” *Physics Procedia*, vol. 63, pp. 79–84, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.phpro.2015.03.013

[58] “Low-Cost 3D Ultrasound | Stanford Ultrasound Imaging & Instrumentation Lab | Stanford Medicine.” [Online]. Available: https://med.stanford.edu/ultrasound/research/sensor3d.html

[59] V. Lewin-Fetter, “Ageing and health,” p. 27, 2010. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health

[60] “urdf/Tutorials/Create your own urdf file - ROS Wiki.” [Online]. Available: http://wiki.ros.org/urdf/Tutorials/Createyourownurdffile