

RAM

● ROBOTICS
AND
MECHATRONICS

DESIGN OF A CONTROL STRATEGY FOR A ROBOTICALLY ASSISTED ULTRASOUND GUIDED BIOPSY

D. (Dimitrios) Pantelis

MSC ASSIGNMENT

Committee:

dr. ir. J.F. Broenink
M.K. Welleweerd, MSc
dr. F.J. Siepel, MSc
J.K. van Zandwijk, MSc

December, 2019

057RaM2019
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

Breast cancer is one of the most common forms of cancer today that is negatively affecting the lives of many people. One very important step in the diagnostics step is performing a biopsy procedure. During a biopsy, the radiologist inserts a needle inside the breast tissue and under ultrasound guidance he moves it towards the target lesion, to extract a sample. The extracted cells are then further examined for a diagnosis to be reached. However, the biopsy can be time-consuming and uncomfortable for patients. Additionally, the accuracy of the result is heavily dependent on the experience of the practitioner who is performing the process. These same practitioners also tend to suffer from fatigue and work-related musculoskeletal pain.

Many of these problems can be alleviated by using a robotic system that would assist the radiologists. Robotics is more capable at performing tasks that require repeatability and precision, compared to humans. This could result in improving the patient's experience, reduce the time of the entire diagnosis process, have a better accuracy rate, and benefit radiologists' working conditions. The MURAB project is one such attempt to create this kind of robotic system. An end-effector for this specific application has already been designed. It consists of an ultrasound probe holder and a needle guiding mechanism. This end-effector is mounted on the 7 DOF KUKA LBR Med articulated robotic arm.

In this thesis, a control strategy is designed that moves the end-effector to the appropriate position, as to allow the ultrasound probe to have a visual of the target and so that the needle orientation mechanic can guide the biopsy needle into the correct direction. For this to happen, what is referred to as the "Initialization Phase" was first created. Furthermore, to ensure that the target would be hit, the tissue located at the target position was tracked through the ultrasound images. This was achieved by designing a tracking algorithm, with optical flow as its core. Additionally, a controller that would guide the needle, using this tracking algorithm, was designed and implemented. This controller relied on controlling the actuators of the needle orientation mechanism. However, the controller cannot compensate inaccuracies presented by needle bending. For that reason, the possibility of directly tracking the needle using the Hough transformation was examined. Lastly, it could be argued that an issue that the above controller has is that it does not give the radiologist any control since the needle orientation mechanism is manipulated with position control. For that reason, an impedance controller was designed that would allow the radiologist to controller the degree in which the robotic system and the user has control over the direction of the needle. This design was then simulated in 20-sim, as to determine its correct behavior.

It was shown that the presented tracking algorithm was able to achieve sub-millimeter accuracy. Combing it with the needle orientation controller, also presented near sub-millimeter accuracy. The needle detection algorithm was able to compensate for big offsets of the needle orientation mechanism but was not able to reduce the error to the degree that the needle orientation controller was able. Furthermore, simulations of the impedance controller, show it can be used to allow the radiologist to manually adjust the needle's orientation if that is desired.

The results of the experiments are promising. They show that the proposed control strategy has the potential of being the bases of a robotic system that can improve the quality of the biopsy process.

Contents

1	Introduction	1
1.1	Context	1
1.2	MURAB project	2
1.3	Research Objective	2
1.4	Report Outline	4
2	Design Analysis	5
2.1	Related work	5
2.2	Design of the initialization phase	7
2.3	Design of the tissue deformation tracking algorithm	10
2.4	Design of the NOM controller.	10
2.5	Impedance Controller.	11
3	Setup	13
3.1	Hardware	13
3.2	MURAB End-Effector	15
3.3	Software Architecture	18
3.4	Ultrasound Phantom	19
4	Experiment Design	22
4.1	Experiment 1 - Tracking tissue deformation due to US probe contact	23
4.2	Experiment 2 - Initialization Phase	25
4.3	Experiment 3 - Deformation due to Needle insertion	26
4.4	Experiment 4 - Needle Orientation Controller	27
4.5	Experiment 5 - Complete Workflow	28
4.6	Experiment 6 - Needle Impedance Controller Simulation	29
5	Results	30
5.1	Experiment 1 - Tracking tissue deformation due to US probe contact	30
5.2	Experiment 2 - Initialization Phase	33
5.3	Experiment 3 - Deformation duo to Needle insertion	35
5.4	Experiment 4 - Needle Orientation Controller	36
5.5	Experiment 5 - Complete Workflow	38
5.6	Experiment 6 - Needle Impedance Controller Simulation	40
6	Discussion	42
7	Conclusion	45

7.1 Future work	46
A Tissue Deformation Tracking	47
A.1 Optical Flow	47
A.2 Lucas-Kanade method	48
A.3 Solving the small motion assumption	49
A.4 Implementation of Tissue Deformation Tracking in the NOM controller	49
B Confidence Map	51
C Needle Detection	52
C.1 Canny Edge Detection	52
C.2 Hough Transform	53
C.3 Incorporate Needle Detection in Control Loop	55
D Mathematics of Needle Orientation Mechanisms	56
D.1 Kinematics of Serial Manipulators	56
D.2 Dynamics of Robot Manipulator	59
E Interaction Control of Needle Orientation Mechanisms	62
E.1 Dynamic Interaction	62
E.2 Impedance Control	62
F Detailed Measurement Results	65
F.1 Experiment 1	65
F.2 Experiment 2	67
F.3 Experiment 3	69
F.4 Experiment 4	69
F.5 Experiment 5	71
Bibliography	72

1 Introduction

1.1 Context

Breast cancer is one of the most common types of cancer that is diagnosed. Particularly for women, this is one of the leading types of cancer, according to (International Agency for Research on Cancer (IARC), 2018). In the United States of America (USA), 1 in 8 women will develop invasive breast cancer throughout their lifetime ((U.S. Breast Cancer Statistics, 2019)). There are multiple methods used today to detect these dangerous cells.

X-ray mammography (MG) is a specific type of breast imaging tool that uses low-dose x-rays to detect cancer early before women experience symptoms, when it is most treatable. This is the most common method used for breast tissue evaluation. That being said, its sensitivity is significantly reduced when evaluating dense breasts ((Chou et al., 2007)). At least 50% of cancers are missed by MG when used on dense breast tissue ((Lander and Tabár, 2011)), which is a significant number if one takes into account that in 71% of the cases, breast cancer occurs in dense tissue breasts ((Arora et al., 2010)).

Another method used is the Magnetic Resonance Imaging (MRI) system. This method allows the detection of significantly more and smaller cancers than MG ((Kelly and Richwald, 2011)). MRI is the most sensitive technique that is used for cancer detection, with a lesion detection rate of 90% up to 99% ((An et al., 2013)). However, MRI has some significant drawbacks. It requires that the patient is injected with contrast medium, it is time-consuming, has a high cost of performance, and has a relatively low specificity for benign and malignant tumors.

The last method that will be mentioned in this chapter is performing a Hand-Held Ultrasonography (HHUS). According to (Wang et al., 2012), HHUS has a lesion accuracy rate of 85.3 %, a sensitivity of 90.6%, and can correctly classify benign and malignant tumors with a rate of 82.5%. Additionally, this method allows for real-time imaging, does not use ionizing radiation, and has a relatively low cost. However, one major draw-back of HHUS is that it is dependent on the operator's ability, hence it is difficult to reproduce. Additionally, according to (Berg et al., 2008), HHUS screening has a relatively high number of false positives and needs considerable effort to perform, in terms of physician time for exam, execution, and interpretation.

If a suspicious lesion is detected after a breast screening is performed, using MG or MRI, it is suggested that a sample of that tissue to be extracted and further examined. This process is referred to as a biopsy. This practice has increased during the past few years, mainly because of wider access to the above-mentioned screening processes. During the extraction phase, an intraoperative imaging technique is used to help guide the needle that is inserted inside the breast. Usually, the free-hand technique is used for this purpose. This technique requires highly skilled radiologists. During this process, the radiologist uses one hand to control the needle that does the cell extraction, and the other hand to hold the Ultrasound (US) device to keep track of the target lesion and the needle simultaneously. Because of the US devices real-time capabilities, HHUS is well suited for intraoperative imaging, however, it lacks spatial resolution. Additionally, properly positioning the transducer and the needle is not a minor task, and is strongly dependent on the experience of the radiologist. An alternative approach is to use a preoperative imaging system, like MRI. It provides more anatomical details but produces static images. While the needle is being inserted, the breast tissue moves around or deforms. These changes cannot be reflected in the static images. Another option would be to use an Automated Breast Ultrasound (AUBS) imaging system which can combine the benefits of both of the pre-mentioned methods.

1.2 MURAB project

This thesis is written as part of the MURAB project, which is an acronym for "MRI and Ultrasound Robotic Assisted Biopsy". This project aims to drastically improve the precision and effectiveness of the biopsy process for cancer diagnostic operations. It is a joint effort from KUKA, SIEMENS, ZGT, Radboud umc, the Robotics and Mechatronics (RAM) group of the University of Twente, University of Verona and The Medical University of Vienna. A biopsy robot will be designed that combines the high precision of MRI images with the flexibility of real-time ultrasound-guided biopsy. The usage of expensive MRI is reduced to a minimum in the work-flow of MURAB because only one MRI image is made.

The original MURAB work-flow is as follows. First, an MRI scan would be performed. Then guided by an MRI-Ultrasound (US) registration, a robotically steered US transducer equipped with an acoustically transparent force sensing will autonomously scan the target area and optimally acquire volumetric and elastographic data. Once that is done, the radiologist can select the target on the mixed image and the robot will steer the instrument to the exact desired pose by adapting the Needle Orientation Mechanisms (NOM) based on real-time US measurements. Tissue deformations will be predicted based on the acquired elastographic measurements. In this thesis, an alternative work-flow is examined. No MRI-US registration is performed, nor any volumetric or elastographic data are calculated. Instead, using the US images, a model-free approach is implemented to allow detection of tissue deformation. Knowing the motion of the tissue at the location of the target, the needle is then guided accordingly. This will allow tracking any tissue deformations in real-time. The radiologist will then manually insert the needle inside the breast, while the needle orientation mechanisms make any needed adjustments, based on visual feed-back given by the US probe.

In its core, the MURAB project attempts to combine the advantages of MRI scans and US probe in one robotic system. MRI provides high-quality images of the anatomy of the breast, which makes it perfect for localization of the target, but is static, representing the status of the breast during only one-time instance. The US probe allows to scan the tissue in real-time but lacks in sensitivity. Robotic manipulators present high precision and repeatability but are complex to control.

The MURAB project can be broken down in to three main phases:

1. MRI imaging
2. Autonomous ultrasound imaging
3. Needle insertion

1.3 Research Objective

This thesis will focus on developing a control strategy for the Needle Insertion phase. When tissue samples are to be extracted, a radiologist will use an HHUS device to guide a needle to the appropriate locations. Given that the accuracy of a biopsy process is affected by the experience of the radiologist, properly positioning a US probe and orientating the needle is a task much more suitable for a robotic system. An end-effector has been designed for the robotically assisted US guided biopsy procedure. This end-effector contains a US probe and the Needle Orientation Mechanism (NOM), which will determine the direction of the needle that is inserted into the breast. This concept is illustrated in Figure 1.1. Given that the lesion location is known on the MRI image of the undeformed breast, the NOM should be able to perform the US guided biopsy. It is assumed that the target will not be directly distinguishable in the US images.

Several challenges need to be addressed for the needle to successfully hit the target of interest. Firstly, for the target to be tracked, the US probe needs to be in contact with the breast as to

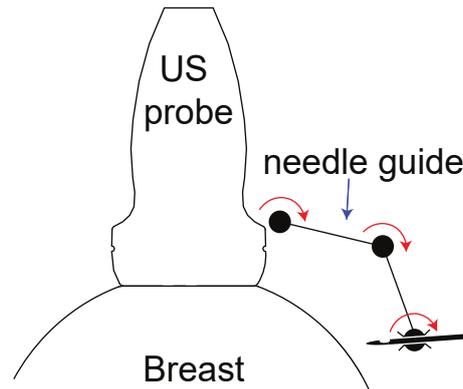


Figure 1.1: Photograph of End Effector

receive a visual of the tissue. However, when that contact happens, tissue deformation occurs. Additionally, when the needle is being inserted, the tissue in its neighborhood is also deformed. The proposed system should be able to address these forms of deformation. Furthermore, when the needle is inserted, depending on the tissue's stiffness, the needle may bend. The error introduced by such a phenomena should also be taken in to account. Lastly, while the needle is inserted, it may be that the radiologist would want to move the needle in a slightly different direction then what the NOM controller has determined. This can be because tissue tracking algorithms have not been yet perfected, US images are of low quality, or because errors can always appear in complicated systems like the one under investigation. So it makes sense to allow the radiologist to take control of the biopsy process if they so choose so. For that reason, the NOM should be compliant with any rotational motion, with respect to the point of insertion, done by the radiologist. With all this in mind, the research objective of the thesis is as follows:

The design and implementation of an initialization protocol for the robotic arm and a control strategy for tracking the desired target, while properly orientating the needle towards that direction.

The final goal is to deliver a proof of concept that given the coordinates of the lesion and an the surface of the breast, relative to said lesion coordinates, the robotic system will be able to autonomously: (1) come in contact with the breast, (2) start tracking tissue deformation around region of interest, (3) track needle for potential bending, and (4) allow the radiologist to manually move the needle to a different direction then that determined by the controller. The following objectives have been defined to achieve those goals:

- *Initialization Phase:* The design and implementation of the motion of the robotic arm in such a way as to allow the US probe to start properly tracking the given target.
- *Tissue Deformation Tracking Algorithm:* The design and implementation of a tissue tracking algorithm using a model-free approach.
- *NOM Controller:* The design and implementation of the controller for the needle orientation mechanism.
- *Impedance Controller* Design and implement compliant behaviour for the needle orientation mechanism.

Since the MURAB project is still in its research phase, having actual patients is not possible. For that reason, a phathom will be designed. All the above objectives will be then tested on that phantom.

1.4 Report Outline

This report is organized as follows:

1. **Chapter 2:** The proposed design is presented.
2. **Chapter 3:** The hardware and software components used for the implementation, experimentation, and simulation, are presented. Additionally, the design of the phantom used is also shown.
3. **Chapter 4:** Describes the design of each experiment.
4. **Chapter 5:** The results of the experiment are presented.
5. **Chapter 6:** discusses the results of the experiments.
6. **Chapter 7:** concludes this work and provides recommendations for ongoing work.

2 Design Analysis

In this chapter, the proposed design for each objective will be discussed, alongside how some design choices were derived from the literature review. In the beginning, the robotic arm will come in a small amount of contact with the phantom, with the target located in its viewpoint. Then the tissue tracking algorithm will start tracking the desired location, which is described in section 2.3. Following that, the US probe will come in full contact with the phantom. This is the initialization phase and it will be described in depth in section 2.2. The tissue tracking algorithm needs to start when there is small contact, as to not produce a lot of deformation, and have the target move to a different location. After the initialization phase is completed, the NOM controller is activated. Using the estimated coordinates that the tissue tracking algorithm gives, it will aim the needle to the target. The controller's design is presented in section 2.4. In the same section, the needle detection algorithm is presented, which assists the NOM controller by trying to compensate possible bending of the needle. This control loop, between the NOM controller, tissue deformation tracking algorithm, and needle detection algorithm, which is activated after the initialization phase is completed, is illustrated in Figure 2.1. Lastly, an impedance controller was designed that would allow the user to manually move the needle by hand if that is desired. This is presented in section 2.5.

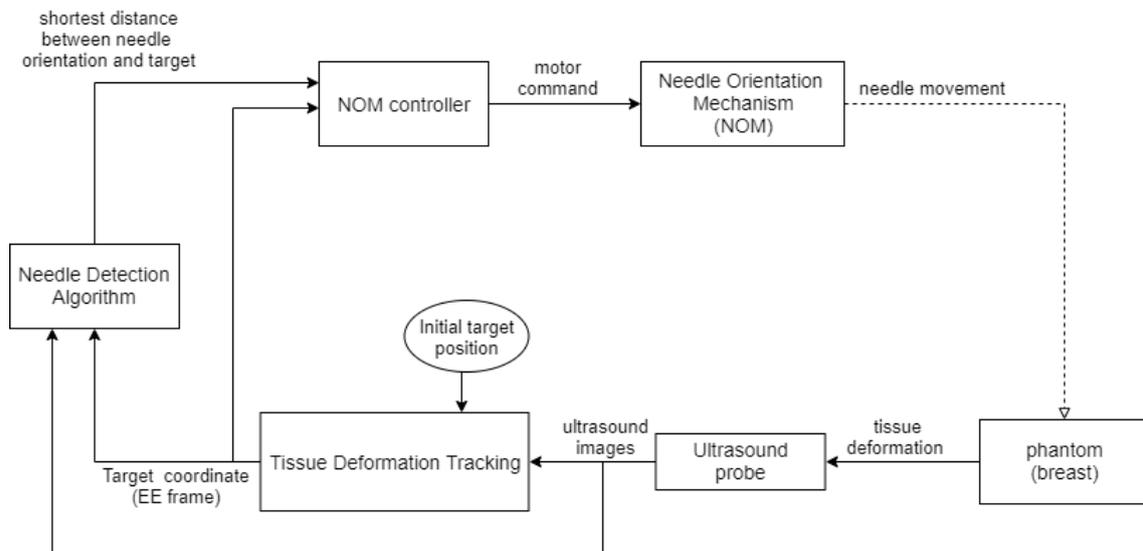


Figure 2.1: Overview of proposed control system

2.1 Related work

In this section we will review relevant research done for tracking tissue motion in US images and detection of needles in US images. This information will be used later on in designing the tissue deformation tracking algorithm and the needle detection algorithm.

2.1.1 Tissue Deformations Tracking

In (Hong et al., 2004) a tracking system is proposed where a gallbladder target position is tracked using a motion-optimized active contour model. It allows the target to be tracked even when it is deformed, alongside its surrounding tissue. The active contour model is used in combination with US images. However, this requires the target to be visible and easily observed in the US images.

An alternative possibility is the use of feature tracking algorithms. In (Abolmaesumi et al., 2002) a comparison between 5 different feature tracking methods are presented when applied to US images and compared to each other.

Another very popular method is Speckle Tracking. This allows for detecting in-plane tissue motions. It is possible to also track out-of-plane tissue motion using speckle decorrelation, as presented in (Krupa et al., 2009) and in (Ren et al., 2010). Work done on out-of-plane motion will be ignored since in our case every motion is done in-plane. Speckle tracking in 2-D is basically what is referred to as block matching in video applications (Noguchit et al., 1999). In (Neubach and Shoham, 2010) a system is designed where a flexible needle is guided inside tissue by using US images. Specifically, it uses a virtual spring model to calculate the required needle base movement, based on the desired direction of the needle. The virtual spring model requires to know the tissue stiffness at different points. In this paper, they use a modified version of the speckle tracking algorithm proposed by (Basarab et al., 2008) for that. The speckle tracking algorithm by (Basarab et al., 2008) uses a normalized correlation for its implementation. However, in (Friemel et al., 1995) a comparison between normalized correlation, non-normalized correlation and Sum of Absolute Difference (SAD) for speckle tracking is done. They found that all algorithms perform similarly, while normalized correlation is significantly more power consuming. In (Bohs et al., 2000) speckle tracking based on SAD is done for determining the 2D velocity of flows in US images. Many of the propositions presented in this paper are due to the fact that the speckle tracking needs to happen fast, as to calculate the flow velocity as fast as possible.

The last method examined for tracking tissue motion is done by using optical flow. In (Chunke et al., 1996) the potential of using optical flow for echo-cardiographs is analyzed. In particular, the Lucas and Kanade (Lucas and Kanade, 1981) approach is used. They found that the bigger the window size was, the more intolerant the results were to noise but with the drawback of increasing motion blur. For that reason, they propose a hierarchical improvement to this conventional method. When receiving the two frames for which the optical flow would be calculated, they would down-sample them. Then, the optical flow would be calculated for both pairs of frames, and the resulting two velocities would be used in a linear equation that would give the final optical flow velocity. Using this approach, the effect of noise was reduced without increasing too much the motion blur. (Sühling et al., 2005) goes a step further by modifying Lucas and Kanade approach to estimate heart motion from a two-dimensional echo-cardiographic sequence. However, most of these modifications are tailored to the needs of analyzing the shape, size and dynamic of the heart. An example of this is the fact that in the paper they track multiple points which have different velocities in different directions. An idea that can be taken from (Sühling et al., 2005) is the use of multiple frames for performing optical flow. In (Pellet-Barakat et al., 2004) US elastography is performed by using optical flow. This is due to the ability of optical flow to model very well local deformations. Although most of the material in this study was, again, tailored to the specific needs of elastography, it does illustrate that optical flow is a good option when it comes to tracking local deformation in US images. Lastly, a comparison between different optical flow solutions is presented in (Baraldi et al., 1996) and (Karami et al., 2017). In both studies, the Lucas and Kanade solutions performed the best. To the author's knowledge, the behavior of Optical Flow for breast biopsy has not been yet researched.

2.1.2 Needle detection in US Images

Many studies were conducted on biopsy robots and needle localization while utilizing US images. In (Neubach and Shoham, 2010) a flexible needle steering setup is presented, where the needle location was detected via the US images. As the needle tip would advance, the main difference between consecutive frames would be that needle tip. By using a simple image sub-

traction of these consecutive frames, the needle movement can be detected. There is a risk of noise or tissue deformation to also be detected as a difference between the frames, so only the immediate area around the needle tip would be analyzed. Initially, given that the area of entry of the needle can be estimated, that area is the region where the needle tip is first expected to be seen. After that, the new searching area is determined by the localization of the needle tip in the previous iteration.

In (Kaya et al., 2015) the needle tip is estimated with the Gabor filter based image processing algorithm. The whole process is broken up into two stages. In the first stage, a Gabor filter kernel is created based on a rough insertion reference angle. After the input image is convolved with the kernel filter, a median filter is applied in an attempt to smooth it. The smoothed image is then binarized using a multiplication of the entropy based tuning parameters and the Otsu's thresholding. Lastly, a morphological erosion is applied, followed by the RANSAC line fitting algorithm, which gives the region of interest (ROI). In stage two a similar process is followed with the difference that the angle used in the Gabor filter is the one calculated by the RANSAC algorithm and that a probability map is calculated using the image binarization (of stage two) as input in combination with the coordinates of the needle ROI. Based on the probability map, the needle tip is estimated. In case there is too much noise, it is possible that a few steps described above can be unsuccessful. For that reason, the paper also used a Kalman Filter for noise estimation and, eventually, better image smoothness.

In (Hong et al., 2004) a needle is robotically guided via the US images. For the detection of the needle, the Hough transform ((Duda and Hart, 1972)) is used. One advantage of this method is that even if some points of the line are not visible (a none continues line), the transform can still detect it. (Wijata et al., 2018) used a similar approach. However, it is stated there that even though the Hough transform is a relatively simple and effective technique, it requires a good image. If the needle it not well visible, the position cannot be determined. To overcome that problem, a combination of the Shock filter and the Gabor filter is used.

2.2 Design of the initialization phase

As already stated, the only information given is the location of the lesion and the location of the surface of the breast, relative to the robot frame. Using this information, the point on which the US probe needs to come in to contact with can be determined. However, the US probe cannot simply be placed on that point and then start tracking the tissue deformation. The reason is that, once the US probe comes in contact with the phantom, some deformation occurs. Hence, the target has moved to a different location than that in which it was in the MRI scan. On the other hand, it is not possible to start tracking the tissue deformation (and the target as a result) without having any US images. Obviously, to receive these images, the US probe needs to come in to contact with the phantom. As a result, the challenge here is how will the tracking algorithm start without moving the target to a different location.

Before describing the proposed Initialization Phase, a question that comes to mind is how can we determine when the US probe is in contact with the phantom. One way of doing that is by using confidence maps (Karamalis et al., 2012). Each pixel of the confidence map determines how certain it is that the specific pixel indeed represents a correct US signal that when through the phantom. This would ensure that what is seen in the US images is not just noise but indeed a picture of the phantom. More details about confidence maps can be found in the appendix chapter B.

The proposed solution is as follows. Firstly, the End-Effector frame (Ψ_{EE}), which is the center of the US probe sensor, will be aligned with the Interaction Point Frame (Ψ_{IP}), along the Z-axis of the US probe, as illustrated in Figure 2.2. Then, the end-effector will start translating along the Z-axis towards the IP frame. At the same time, a pre-defined region in the confidence map that is produced by the US images will be monitored, as seen in Figure 2.3. Once the confidence of

this region is increased, indicating contact with the breast, the robotic arm will stop. In the case that the contact exceeds this region, the end-effector will move back, as to constraint the contact to this region only. If contact is determined on the right or left side of this region, the end-effector will rotate in such a way as to keep the target in the center of its frame. Once the above rotational and translation criteria are met, the robotic arm will stop moving. At this point, the expected location of the target inside the US image will start to be tracked. Subsequently, the end-effector will be moved further in, until there is full contact between the end-effector and the phantom. Since the robot arm is moved by impedance control, it is possible that the end-effector is not properly aligned with the target. This can happen because of an imperfect model of the robot or end-effector. For that reason, an PI controller will be constantly correcting the trajectory of the end-effector by pulling the springs of the impedance controller, in such a way as decrease the error between the actual position and wanted position of the end-effector. As a result, the first US images that will be received should contain the target in them.

During the experimentation period, where the above algorithm was tested, no MRI machine was available. Because of that, a minor adjustment had to take place. Instead of generating the IP frame from the MRI scan, it was manually calculated. In other words, the only step that was neglected was the calculation of the IP frame from the MRI scan, which was necessary for the experiments to not be delayed any further.

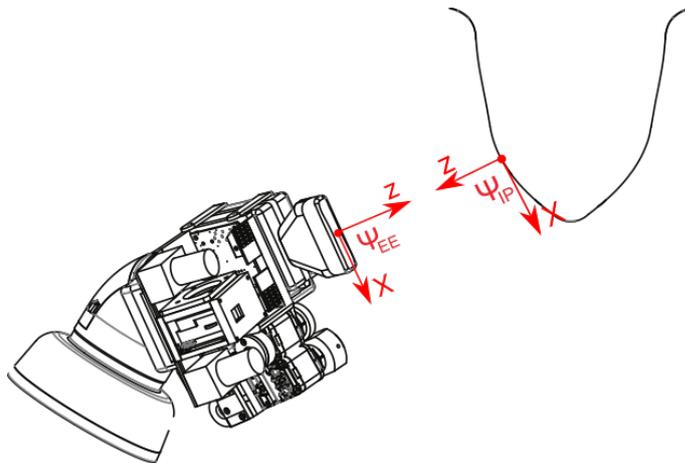


Figure 2.2: Alignment of Ψ_{EE} and Ψ_{IP}

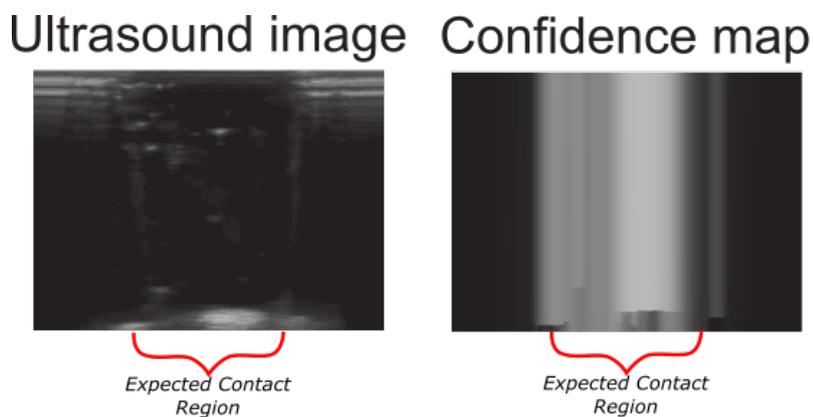


Figure 2.3: Region in which when something is indicated, the robotic arm will freeze.

2.2.1 Implementation of adjusting initial contact using confidence maps

At this point we will elaborate more on details about how the confidence map is used for moving the US probe.

Firstly, how the probe is rotated using the confidence map will be addressed. When the first contact occurs with the phantom, it cannot be guaranteed that the center of the probe first come in contact with the phantom, since the SAIP controller (Looijer, 2018) performs impedance control and not position control. The confidence map is divided into three parts, the left region, the right region, and the center region, as depicted in Figure 2.4. A white section means the probe is in contact with the phantom, whereas a black section means there is no contact. In other words, the aim is for the white vertical lines to be in the center of the image. Additionally, the location of the average values along the column dimension of the confidence maps, for a set depth, is calculated with each new iteration. The depth values used is an adjustable variable. When the average value is in the left or right region, a rotation of the end-effector frame around the target is performed. The probe moves around the target's coordinate, expressed in the end-effector frame, to maintain the target position centered in the US image. This transformation is given by equation 2.1, where d is the distance between the end-effector frame and the target along the Z -axis, and θ is the predefined angle by which it will rotate. The angle should be kept small, as to allow for fine-tuning rotations.

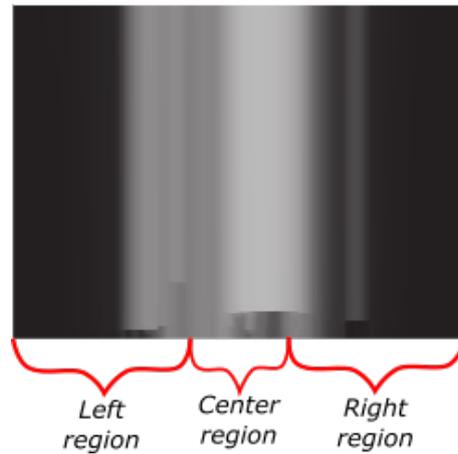


Figure 2.4: Divided regions of Confidence Map

$$H_{EE}^t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -d \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ \sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

The second way the confidence map is used is for determining how much the US probe should be translated along the Z -axis. For this to be done, the percentage of the confidence along the columns is calculated for each new confidence map. A region is then defined of how much percentage along the columns should exist in the confidence maps for the algorithm to determine that the contact is good. In particular, a lower limits and a higher limit is defined. If the confidence percentage is under the lower limit, the robot moves forwards along the Z -axis of the end-effector frame, while if it is above the higher limit, it moves backwards along the Z -axis of the end-effector frame. When the percentage is in between, the robotic arm stays in place. A problem that can occur here is that due to the significant delay that it takes for the confidence map to be calculated, the probe may end up move back-and-forth forever. That is why, in the implementation of this system, the more back-and-forth motions happen, the smaller the

translation along the Z-axis of the end-effector frame will be. In the end, the translation will be so small that the confidence map will be able to keep up with the motion of the robot arm.

In the end, when all the confidence is located in the center of the image, and the amount of confidence which is desired is reached, then the Tissue Deformation Tracking system is activated.

2.3 Design of the tissue deformation tracking algorithm .

During the biopsy process, a US probe will be used to help guide the needle. Because it is often the case where a lesion is so small that it can be seen only in MRI, it is assumed that no target can be seen in the US images. In other words, tracking is done "blindly". Furthermore, the tracking will happen without the need for a tissue deformation model. Creating such a model is difficult and the model's accuracy can significantly affect the results. For those reasons, what possibilities exist for tracking the target without such a model are to be explored.

A number of different approaches were seen in literature review. Using an active model (Hong et al., 2004) or feature tracking algorithms (Abolmaesumi et al., 2002) is not possible for this application. This would require the target to be distinguishable from the US images, which is not the case. The other two options are Speckle Tracking or Optical Flow. Both these approaches have a lot of potential. There has been a lot of work done in Speckle Tracking so far. However, in this thesis, a control system and a possible workflow for it, is to be designed. Implementing a Speckle Tracking algorithm, based on recent work, would require a lot of time. If a simple version of Speckle Tracking from earlier years was implemented, which would require less time, the results would not present any new insight on the use of this method. On the other hand, good Optical Flow implementations already exist. Furthermore, no work was found that used it for this specific application. As a result, using Optical Flow would be faster to implement and would present more interesting scientific results.

With all the above in mind, a Tissue Deformation Tracking algorithm is designed, with Optical Flow as its core. Specifically, the (Lucas and Kanade, 1981) Optical Flow was used. Details about Optical Flow can be found in the appendix chapter C. The idea is that the tissue at the area where the target is expected to be, will be tracked. In Figure 2.1 the position of this algorithm in the control loop, can be seen. Knowing the motion of the tissue is equivalent to knowing where the target is itself. This is, in a sense, speckle tracking. The difference is that in Speckle Tracking, the distance between the region of interest is calculated, while in the Optical Flow case, the velocity along both axes is what is of interest. Lastly, some pre-processing will take place in an attempt to reduce the sensor noise while maintaining tissue speckles.

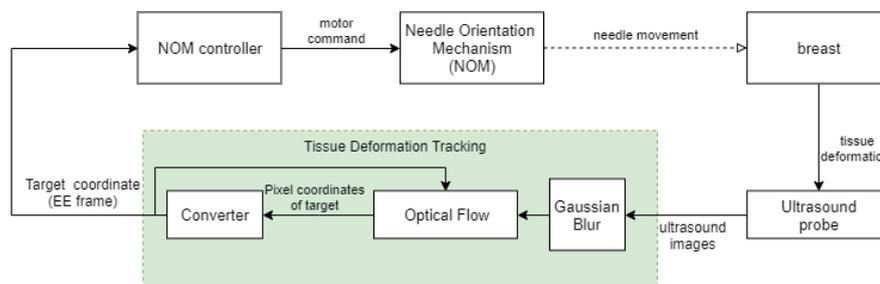


Figure 2.5: NOM Control Loop when Tissue Deformation Tracking is performed.

2.4 Design of the NOM controller.

Once the target is properly tracked, the Needle Orientation Mechanism (NOM) will start aiming the needle holder towards the estimated target. The following closed-loop system is proposed for that task. Given the insertion point and the estimated target location, the orientation of

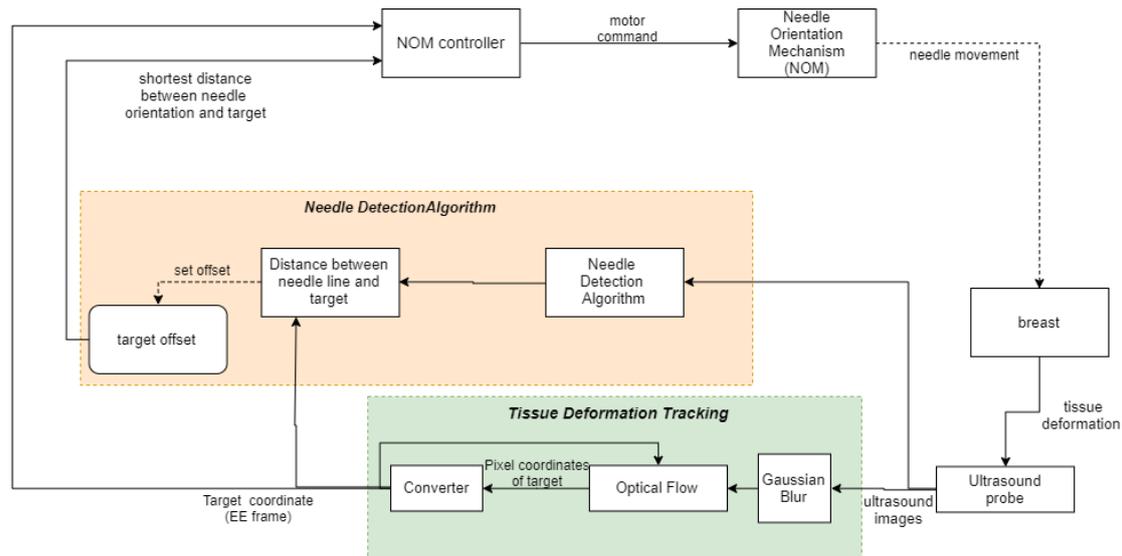


Figure 2.6: NOM Control Loop when Needle Detection is activated

the needle can be calculated. The difference between the desired angle and the current angle is sent to the NOM controller. Then, using the inverse kinematic model of the NOM, the required positions can be calculated and sent to the motors of the NOM. The kinematic models were derived using screw theory. More details about how it was calculated can be found in the appendix chapter D.

However, an issue may occur in this proposed control loop. Depending on the tissue stiffness, it is possible that some bending of the needle can occur. This can produce an error which affects the accuracy of the NOM controller. For that reason, an additional step would be to track the needle in the US image, to determine its current orientation.

The method presented (Neubach and Shoham, 2010) would require a relatively clean US image, which the current machine does not provide. For that reason, it was preferred to detect the line that is shaped in the US images by the needle. This would work since the orientation of the needle is what is sought for. For that reason, the Hough Transformation (Duda and Hart, 1972) will be used, as proposed in (Hong et al., 2004) and (Wijata et al., 2018). Implementation details about the Hough Transform can be found in appendix chapter C. However, it is mentioned in (Wijata et al., 2018) that the line detection can be easily lost because of noise or because the needle does not create strong enough reflections of sound as to register with the US probe. For that reason, this functionality will work as to assist the NOM controller, without making the controller dependent on the line detection.

In Figure 2.1 the above control loop is presented. The needle detection algorithm will detect the line of the needle and then calculate the shortest distance between it and the target. That value is then set in the "target offset". This offset is taken into account by the controller when orienting the needle. Using the design, if the needle is not detected or is lost, the offset will stay where it was set from the last time the line was detected. In other words, it will not affect the NOM controller, which can carry on with orienting the needle as already mentioned.

2.5 Impedance Controller.

The above-proposed control loop will direct the needle to the target, based on the tracking algorithm. However, there is a possibility that an error will be introduced inside the tracking algorithm. In those cases, the radiologist may be able to notice that and be able to determine the correct location of the target. For that reason, it would be useful to the radiologist to be able to move the needle to the correct orientation by hand. But that is not possible in the

above control system where the NOM uses position control for the needle orientation. For that reason, an impedance controller is suggested. For more details about how impedance control works, refer to appendix chapter E. This would allow the radiologist, if wanted, to move the needle to the orientation that the radiologist want. Additionally, it allows the radiologist to determine to which degree the robot has control.

To do this, the controller will be as followed. Since the needle can only move in a 2D plane, there are three degrees of freedom, two transnational and one rotational. What is desired is that after the distance between the last NOM joint and the breast is determined, along the direction of the needle, that only rotations along that point will take place. This is depicted in Figure 2.7. For that reason, two very stiff springs will be placed for the transnational part, as to insure that the needle will be always passing through the insertion point. Additionally, a rotational spring will be placed at that same point. However, the stiffness of this spring will be varying based on what the radiologist wants. The stiffer the spring, the more control is given to the robot, and vice versa.

The proposed impedance strategy will be implemented and simulated in 20-sim as the establish correct behavior. Real-life implementation is not possible yet, with the current motors that are used in the NOM.

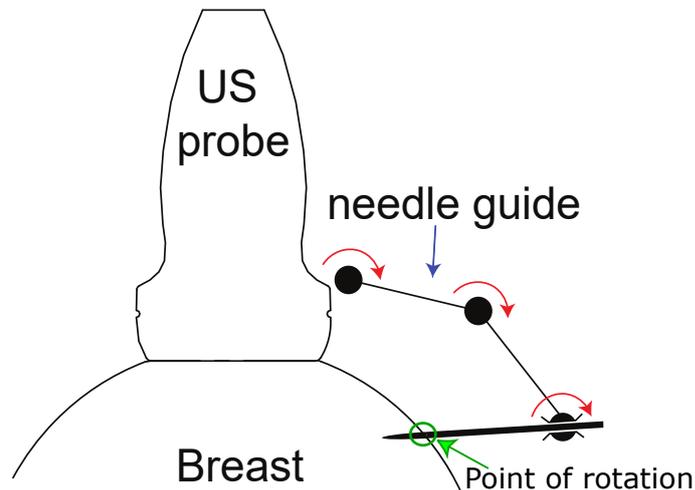


Figure 2.7: Point of rotation for impedance controller

3 Setup

In this chapter all component used for testing out the proposed workflow will be presented. Initially, all hardware used for the implementation and validation is described, followed by the software architecture. Lastly, the US phantom design will be explained.

3.1 Hardware

In the setup, the End-Effector, which contains the NOM and the US probe, is attached to the flange of a KUKA robotic arm. An overview of the hardware components involved are illustrated in Figure 3.1. In particular:

1. KUKA LBR Med
2. KUKA Robot Controller
3. Windows computer
4. Linux computer
5. SmartPAD
6. End-Effector
7. US Machine

Additionally, in order the measure the location of the target which the system would attempt to extract, and the location of the needle tip, the Aurora system by NDI was used. An overview of the components for this system can be seen in Figure 3.2 In particular:

8. System Control Unit
9. Aurora Field Generator
10. Needle sensor
11. target sensor
12. logging computer

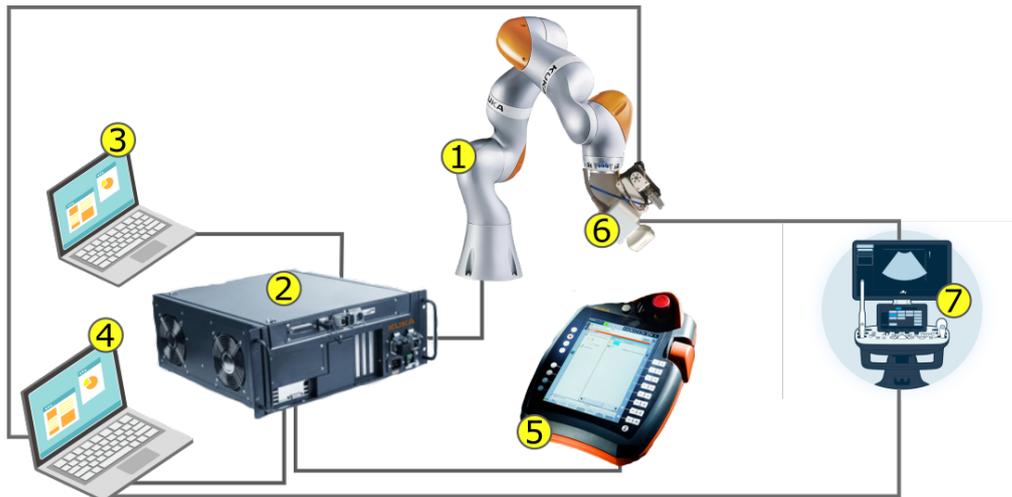


Figure 3.1: Overview of the Robotic setup, with (1) KUKA LBR Med, (2) KUKA Robot Controller, (3) Windows computer, (4) Linux computer, and (5) SmartPAD

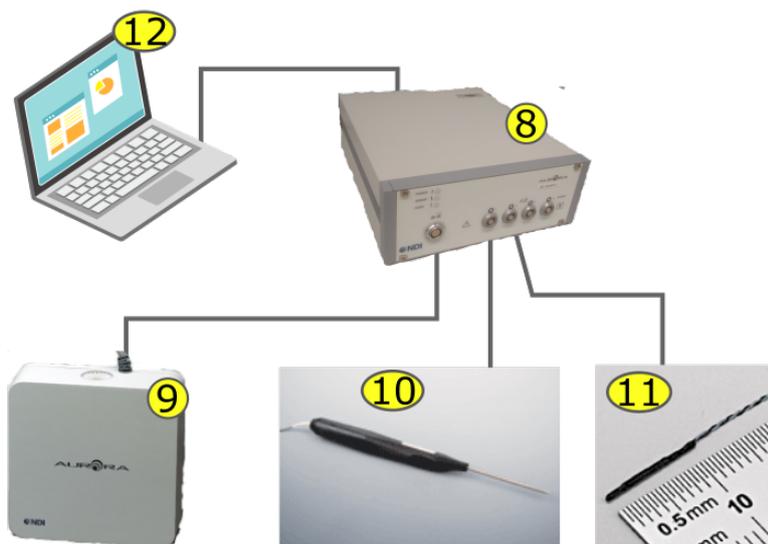


Figure 3.2: Overview of the Aurora setup, with (6) System Control Unit, (7) Aurora Field Generator, (8) Needle sensor, (9) target sensor, and (10) logging computer

3.1.1 Robotic Manipulator

KUKA LBR Med

The robot used is the KUKA LBR Med, which is a medically certified articulated robot with 7 DOFs, as seen in Figure 3.3. This robot is especially designed for medical applications, such that it meets the medical safety requirements. The robot has one redundant DOF, which gives it more dexterity and helps avoiding typical singularities of a 6-DOF manipulator. Each joint is equipped with position and torque sensors, such that it can be operated with position, velocity or torque control.

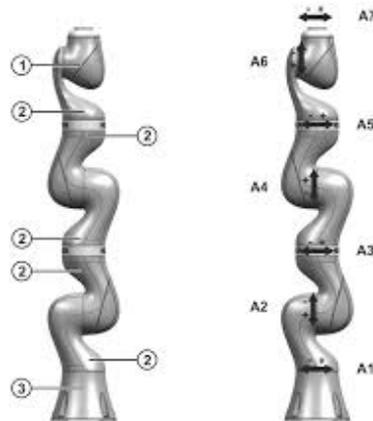


Figure 3.3: KUKA LBR with indicated joint frames

KUKA Robot Controller

The KUKA Robot Controller (KRC) directly controls the robot. Applications created on the Windows computer can be transferred to the KRC through the KUKA Line Interface (KLI). The Linux computer can connect to the KRC through the KUKA Option Network Interface (KONI), which is a UDP-based interface that allows data exchange.

KUKA SmartPad

A human operator can communicate with the robot controller through the KUKA SmartPad, which provides all the operator control and displays functions required for manipulating the robot. It can be used for manually rotating joints, information about the current state of the robot, teaching frames to the robot, or running Sunrise Workbench applications, among other things. The KUKA SmartPad and the KRC can also be accessed through Remote Desktop.

3.2 MURAB End-Effector

The end-effector used in this thesis is presented in (Welleweerd, 2018). It is composed of a holder for the US probe and a 3 DOF serial robotic manipulator on its side. The manipulator is the NOM of this end-effector. At the tip of the serial manipulator, the needle holder it place, from which the needle goes through. It can be seen in Figure 3.4. The motors used are the HerkuleX DRS-0201¹.

¹https://wiki.dfrobot.com/HerkuleX_DRS-0201_SKU_SER0033



Figure 3.4: MURAB end-effector

3.2.1 Ultrasound Device

The Siemens ACUSON X300 ultrasound system (Figure 3.5a) with the VF13-5 linear transducer (Figure 3.5b) are used in this project for US imaging. The ACUSON X300 is a US system that facilitates accurate diagnosis and provides an operator-friendly interface.



(a) Siemens ACUSON X300



(b) VF13-5 Linear Transducer

Figure 3.5: US device components.

Video Capturing Device

The Magewell Pro Capture DVI ² device is used to transfer images from the US device to the Linux computer. It can be seen in Figure 3.6. The Tissue Deformation Tracking algorithm, the Needle Detection algorithm and the confidence maps can then be calculated on the Linux computer, and give the appropriate command to the KUKA robot or the NOM. The US device is connected to the Pro Capture by DVI interface. The Pro Capture is connected on the PCI bus of the Linux computer.

²<https://www.magewell.com/products/pro-capture-dvi>



Figure 3.6: Magewell Pro Capture DVI

3.2.2 Aurora tracking system

The Aurora tracking system allows for tracking multiply targets simultaneously. There are two types of field generators, the "Planar Field Generator" and the "Tabletop Field Generator". The "Planar Field Generator" was used for the setup. This field generator has two modes, in which a different volume about the generator is tracked. These two modes can be seen in Figure 3.7. The cube volume covers a smaller space but has higher accuracy. The space that is needed for this experiment is well inside the cube volume. For that reason, that mode was used.

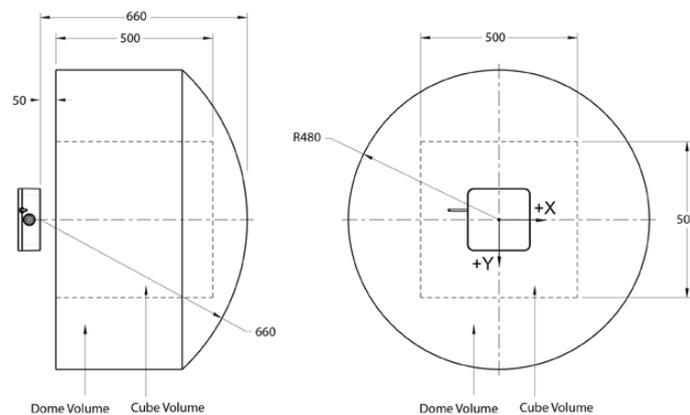


Figure 3.7: Volume in which the NDI sensors can be measured. All number are in millimeters.

3.2.3 3D Printed Test Structure

In Figure 3.8 the 3D printed structure that was used for executing the experiment can be seen. This structure allows for the phantom to be slides in and out the setup, when needed. Furthermore, a distance exists between the phantom and the Aurora field generator. This is because the minimum distance from which the field generator can start tracking is 50mm (as depicted in Figure 3.7).



Figure 3.8: 3D printed structure with Aurora sensor and phantom.

3.3 Software Architecture

In this section we will describe the software architecture as used in this project. A schematic representation of the communication between the computers, the robotic arm, the US device, and the End-Effector, can be seen in Figure 3.9. The Windows and Linux computers are both connected to the KRC through Ethernet. The Windows computer is used to program applications in Sunrise Workbench, which can be executed on the KRC. The Linux computer can be used for real-time data exchange between a robot application on the KRC and a FRI client application on itself.

In our approach, the Sunrise application moves the robot to a desired initial configuration and commands a position hold from there. The FRI client then takes control by sending torque commands to the KRC. The KRC directly controls the robot, which has the End-Effector attached to its flange. The US system captures images and provides them to the Linux computer. A C++ program then uses those images for computing the confidence map, the Tissue Deformation Tracking algorithm, or Needle Detection. Based on the information computed there, the program provides torques to the KRC, via the SAIP (Looijer, 2018) controller, or position control to the NOM.

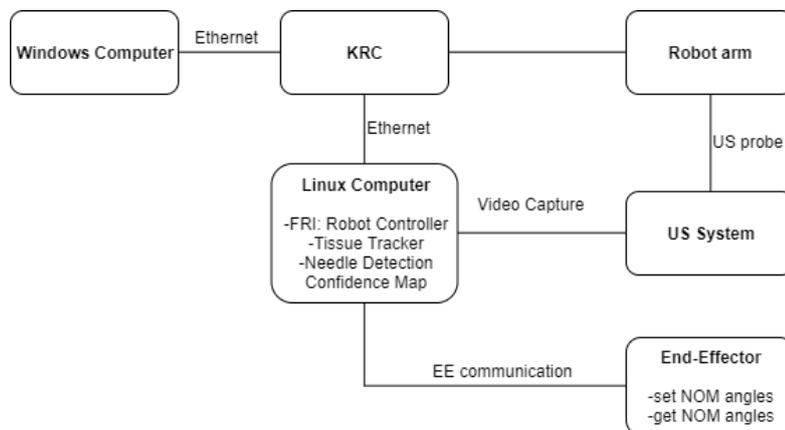


Figure 3.9: Schematic representation of the software setup, from the Robot's side

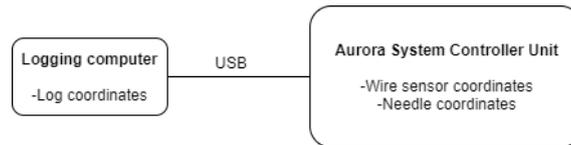


Figure 3.10: Schematic representation of the software setup, from the Aurora's side

3.3.1 Sunrise Workbench

Sunrise Workbench³ is a tool to program robot applications in Java, which can be executed on the KRC. It allows management of coordinate systems and motions, and provides a programming interface for integration of external libraries and functionalities. The robot can be commanded to execute a linear motion, point-to-point motion, circular motion, spline motion, or position hold. During these motions, the robot can use position control, axis-specific impedance control, and Cartesian impedance control.

3.3.2 Fast Research Interface

The Fast Research Interface (FRI) facilitates continuously and real-time-capable data exchange between a robot application on the KRC and an FRI client application on an external system. A robot application on the KRC that is programmed with Sunrise Workbench, can be overlaid by the FRI with a position, wrench or torque overlay. This allows the user to create C++ applications for real-time control of the robot.

The FRI is a state machine that has four states:

- *MONITORING WAIT*: The KRC has opened the FRI connection and is waiting for real-time-capable data exchange.
- *MONITORING READY*: The KRC is performing real-time capable data exchange with the FRI client application.
- *COMMANDING WAIT*: The KRC initializes the motion that is commanded by Sunrise Workbench and synchronizes itself with the FRI client.
- *COMMANDING ACTIVE*: The KRC applies the commanded values from the FRI client application for superposing the robot path.

The FRI client cyclically commands position, wrench or torque overlays to the KRC at a maximum rate of 1KHz. The method receives information about the current state as input. When the FRI state has changed, it calls a callback function to react on the state change. The FRI state machine recognizes the current state of the FRI and cyclically calls the corresponding callback function.

3.4 Ultrasound Phantom

This section describes the design and production of the US phantom. Phantoms that mimic human body parts are used for experimentation in the MURAB project, because the safety of the human subject cannot always be guaranteed in the experimental phase. Developing a phantom in-house also provides more flexibility on the shape and structure of the phantom compared to using a commercial phantom.

3.4.1 Shape of phantom

The aim of the project is to extract cells from a breast. With that in mind, the phantom should partially resemble a breast. A female human breast is composed of the skin tissue, the fat tissue

³website: <https://www.kuka.com/en-de/products/robot-systems/software/system-software/sunriseos>

and the glandular tissue. Producing a phantom with all of these types of tissues is challenging. For that reason, our phantom will only have the skin and fat tissue. It is expected that should be sufficient for the experiments. Additionally, since the main focus of the thesis is controlling the needle, we only care about a 2D slice of the breast shape, which is what the US probe can see. What was decided to be the final design is a plane of a breast, expanding in a out-of-plane direction. Figure 3.11 presents a CAD model of what the desired shape is. This way, there are multiple "breast planes" as to conduct many experiments on. This is important because the more a needle is inserted inside the phantom, it will start to tear up and become unusable.

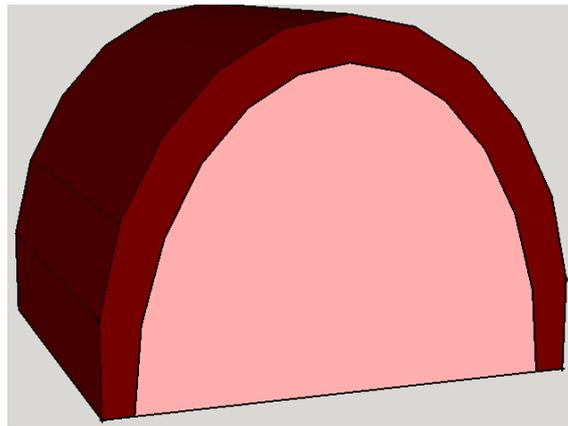


Figure 3.11: Basic shape of phantom. Pink is the fat layer while dark red is the skin layer.

3.4.2 Material

Choosing the correct material for creating a phantom, is not a simple task. The two main properties that are of interest when it comes to tissue-mimicking material is its speed of sound and its attenuation coefficient. The reason for that is because the US machine depicts distances based on these properties for human tissue. So the material of choice should be similar to human tissue in these two properties.

The proposed phantom design has mostly fat tissue, with a think layer of skin tissue around it. According to (Thouvenot et al., 2016) the speed of sound for fat tissue is usually between 1540ms^{-1} and 1465ms^{-1} , while for stiffer tissue around 1630ms^{-1} . Furthermore, the appropriate range of attenuation coefficient for material used for medical US is $0.3\text{dB MHz}^{-1}\text{cm}^{-1}$ to $0.7\text{dB MHz}^{-1}\text{cm}^{-1}$. In this same paper, they found that polyvinyl chloride plastisol (PVC-P) had similar properties to fat tissue, depending on its stiffness. For tissue like skin, it was not as close. (Maggi et al., 2013) and (Spirou et al., 2005) found similar results, concerning the relationship of PVC-P and fat tissue.

Since the fat layer is the biggest part of the phantom, it was decided to use PVC-P for this process. For adjusting the stiffness of the material, assouplissant plastileurre⁴ was used. Additionally, created speckles are essential, since that is what is used for tracking tissue deformation. Silica gel was used for that purpose.

3.4.3 Mold

A mold was designed in SketchUp⁵ and consist of two components. These are presented in Figure 3.12. First, for the creating of the skin layer, the material is poured inside the left mold, and then the left piece is pushed inside the mold. When this piece is fully inside, there is a distance of 10mm between them. The material spreads around this space, which creates the

⁴<https://www.bricoleurre.com/product/assouplissant-plastileurre>

⁵<https://www.sketchup.com/>

skin layer. Following that, the left piece is pulled out and the fat layer material is poured on the skin layer.

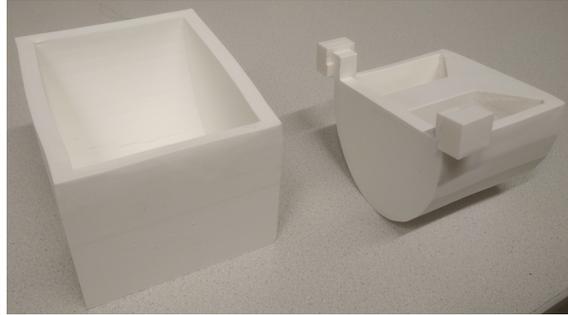


Figure 3.12: 3D printed mold for the casting of the phantom

3.4.4 Final result

The final result is presented in Figure 3.13. The dimensions of the bottom surface of the phantom are 130x100 mm. For the experimentation, the aurora wire sensor (number 9 in Figure 3.2) is placed inside, from the side of the phantom.

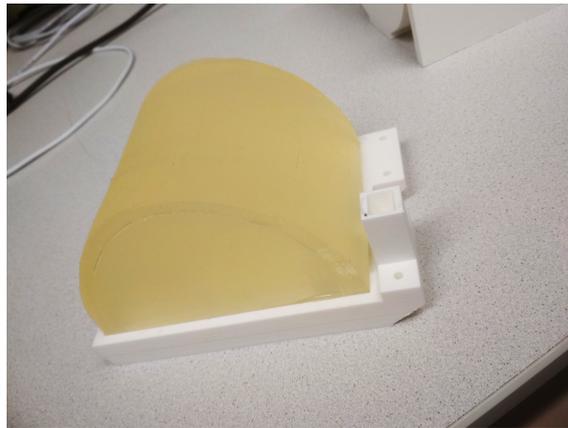


Figure 3.13: Final Phantom

4 Experiment Design

In this chapter, the experiments that are performed, to validate the designed system, are presented. For each experiment, the different parameters of the setup are presented, how the experiments are set up and executed, and which information is recorded in order to evaluate if the given design achieved the milestones of this thesis.

The structure of the chapter is as follows. Each section is for a different experiment. In each section, the first thing discussed is what is the purpose of this experiment and how it correlates with the research objectives of this thesis. Following that, an analysis takes place. Here all the system parameters, test setups, and test variables are presented. The system parameters are the different parts of the setup that are expected to influence the results of said experiment. We present which ones are identified and the reasoning behind why they are expected to affect the results. Each test setup is an execution of the experiment using a different combination of system parameters. The test variables are the quantities that are measured during each execution. They are used to evaluate the quality of the system. Lastly, in each section, the physical setup is presented.

In the following experiments, the already presented tissue deformation tracking algorithm will be used. For the implementation of the image processing components, the OpenCV library¹ was used. The two main settings that affect the results of optical flow is the window size and the number of pyramid layers. These are to be kept constant throughout all the experiments, as to allow to properly compare results from different experiments. When it comes to the window size, it needs to be large enough to have a sufficiently unique speckle pattern, yet small enough to contain only pixels with about the same disparity. From a practical point of view, if the point of interest is not properly tracked, the window is too small, and if the tracked motion does not seem to reassemble the motion of the area around that point, the window is too big. An additional issue that big windows can have, which do not relate with accuracy, is there need for more processing power. While using the default 21x21 window size, that OpenCV uses as a default value, on some quick intuitive experiments with US images, none of the above issues were observed. It is, therefore, it is assumed that this window size gives acceptable results. Regarding the number of pyramid layers, they are needed to ensure that the small motion assumption is kept. If the assumption is not met, the tracking is lost (refer to appendix A for details about the small motion assumption). With that in mind, a quick experiment was performed where US images were taken from a phantom, several points were tracked and a needle was inserted to allow to move the material around. In the beginning, the number of layers was at zero and increased whenever the tracking was lost. To ensure that the number of layers was sufficient, the motions with the needle was much faster than what is to be expected from any biopsy process. If the small motion assumption is not broken with these motions, it is safe to assume that during a proper procedure, the assumption would still be insured. The best value found was 3 layers.

Additionally, the PI settings of the trajectory controller are also kept constant. The tuning of the parameters happened manually. The testing for the tuning parameters had the robotic arm trying to keep a position, and then when changing the tuning parameters, observing the changes in error between wanted position and actual position. Firstly, the proportional element was examined. The initial value was 0.1 for both translation and rotation. This value was increased by 0.1 until vibration like motion was observed. Then, that value was decreased by 0.2. After those parameters were set, the I element was tuned. Again, we started at 0.1 and slowly increased. While the permanent error was being reduced, the value was incremented. When overshooting

¹OpenCV - version 3.2.0, <https://docs.opencv.org/3.2.0/>

was observed, the value was again decreased by 0.2. The final values that seem to work well for the trajectory controller, are:

Elements	Proportion	Integration
Translation	0.8	0.4
Rotation	0.2	0.3

Table 4.1: PI elements for translation and rotation

4.1 Experiment 1 - Tracking tissue deformation due to US probe contact

In this experiment, we will examine how well the presented tissue tracking algorithm performs when the cause of deformation is the pressure applied by the US probe. Deformation is caused by the pressure that the US probe applies to the breast and by the needle when inserted. In this experiment, only the former is examined.

There are two reasons why the two sources of tissue deformation are examined separately. Firstly, it will allow us to properly assess if the tracking algorithm performs differently to global deformation, like that caused by the US probe, and local deformation, like that caused by the needle insertion. Secondly, this information correlates to the performance of the initialization phase. As presented in the Design Analysis chapter, during this phase, when the end-effector has a small contact with the phantom, the tracking algorithm will be activated, and then the end-effector will be pressured in further so that the US probe has full contact with the phantom. In this experiment, the possible error that this motion may introduce into the system is to be examined.

The experiment will take place as follows. The US probe will be manually placed on the phantom, in such a way for it to have full contact. Then, the tissue tracking algorithm will be activated. A few pre-selected targets will start to automatically be tracked. Following that, the robotic arm will translate along the Z-axis of the end-effector frame. First, forward motion will take place and then a backward motion, to the position it started from. This motion can be repeated as much as needed.

4.1.1 Analysis

At this point, the experiment will be broken down into several system parameters, test setups, and test variables.

System Parameters

- *US probe speed:* This is the speed in which the motion of the US probe will be. Higher speed, resulting in faster deformation, would be more challenging for the tissue tracking algorithm. However, the speed is not allowed to be too high as this may cause harm to the patient. This is also ensured by the SAIP controller (Looijer, 2018), for the same reason.
- *Angle of pressure:* The direction from which the US will pressure the phantom. If the US probe is directly above the phantom, the pressure to the phantom will result in compression of the material. On the other hand, if pressure is applied from one side, since there is nothing on the other side of the phantom, there will be more motion of the material than compression. This means that the direction of the US probe will determine if the deformation is due to compression or translation of the material.
- *Depth of target:* Targets can be located at different depths in the breast, relative to its surface. The further the target is from its closest surface point, the fewer speckles will be seen. This implies that tracking targets further inside the phantom should be more challenging.

- *Number of in-and-out motion:* This determined how many times the US probe will be pushed forward and backward along the Z-axis of the end-effector. This motion should be repeated a few times to determine if the target is consistently being tracked along the same path.

Test Setups

Two different test setups will take place. The parameter that will be changed, in an attempt to observe how it affects the tracking system, is the angle of pressure. In the first test case, the pressure will be applied from the center of the phantom, while in the second test case it will be applied from the side.

Regarding the other parameters, the speed of the probe will be chosen to be a relatively high one, but not too high, as for it to be allowed by the SAIP controller. Furthermore, 3 points will be simultaneously tracked, with distances 1cm, 2.5cm and 5cm from the US probe. This will allow assessing if different depths affect the quality of the tracking algorithm. The optical flow settings will be those mentioned in the introduction of the chapter. Lastly, the number of in-and-out motions will be set to 3.

Both the above test setups will be done 10 times, which will allow assessing the accuracy and precision of each test case. A synopsis of the above tests can be seen in table 4.2.

System Parameters	Test 1	Test 2
US probe speed	2mm/sec	2mm/sec
Angle of pressure	center	side
Depth of target	1cm, 2.5cm and 5cm	1cm, 2.5cm and 5cm
number of in-and-out motions	3	3

Table 4.2: Synopsis of all Test Setups for Experiment 1

In Figure 4.1a the setup of test 1 is shown. The probe initial position is on the top of the phantom. The motion of the probe will be downwards. In Figure 4.1b the setup of test 2 can be seen.



(a) Setup for test 1



(b) Setup for test 2

Figure 4.1: Experiment 1 setup for each test.

Test Variables

- *Estimated Target Location:* The only value recorded in the estimated tracked location, expressed in the end-effector frame.

4.2 Experiment 2 - Initialization Phase

In this experiment, the effectiveness of the proposed initialization phase is examined. The goal is to minimize the deformation caused by the initial contact. Determining what error is introduced into the tracking algorithm by this initial contact, is the purpose of this experiment. For this experiment, the whole initialization phase will be performed.

4.2.1 Analysis

System Parameters

- *Contact percentage region*: This is the amount of contact the US probe should have with the phantom before the optical flow starts tracking the target. The exact percentage of contact does not matter. A few percentages up-and-down have virtually the same result. Also, trying to achieve an exact percentage, while having a significant delay from the calculation of the confidence map, is unnecessarily challenging. This is why a small region of acceptable contacts is defined. The higher the average value of this region is, the more robustness exists in ensuring that the target of interest will be in an area where the US probe has a visual. The reason is that even though the target is expected to be in the center columns of the image, it is not always guaranteed. Hence the more contact the probe has, the more tolerance there is for the target not to be exactly in the center columns of the image. However, the higher the average of this region, the more deformation will take place which will not be tracked by the optical flow algorithm.
- *US probe speed*: This parameter is important relative to the calculation of the confidence map. That is because of the delay the calculation of the confidence map produces.
- *NDI Target location*: The location of the target does not have any notable effects on the results. However, it should be kept constant for all test setups of this experiment.

Test Setups

In this experiment, one test will take place. The only thing possible is to challenge the algorithm. This can happen by giving a relatively high velocity for the probe but not too high as to create issues with the SAIP controller. That is why 2mm/sec is chosen for that. For the contact region, 20% to 35% is chosen. These values allow having very small deformation on the phantom while having a region big enough for the target to be in when contact is established. Lastly, the target coordinates do not matter as long as they are located inside the phantom, in a depth not bigger than 5 cm. In table 4.3 the synopsis of this test can be seen. Figure 4.2 shows a possible setup for this experiment.

System Parameters	Test
Contact percentage region	20%-35%
Probe speed	2mm/sec
NDI Target location	(0.55, -0.18, 0.38)

Table 4.3: Synopsis of all Test Setups for Experiment 2

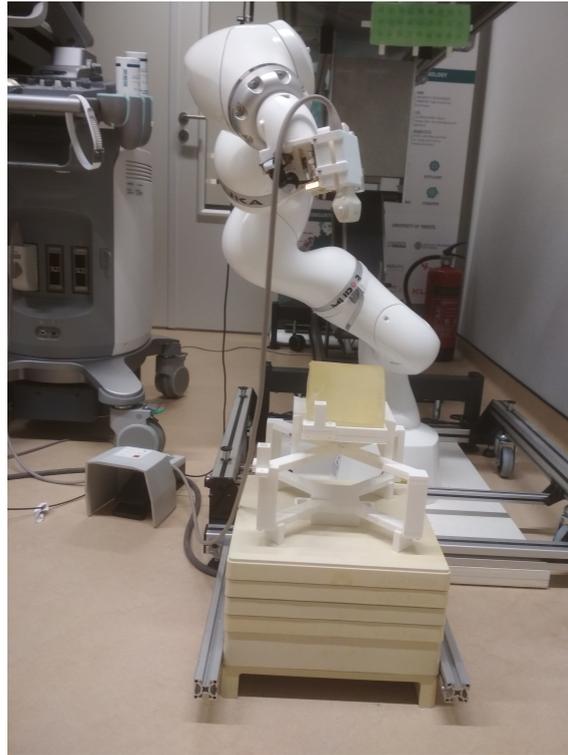


Figure 4.2: Initial configuration of Experiment 2. The position of the robotic arm was randomly put there.

Test Variables

- *Estimated Target Location:* The estimated location of the target, from when the US probe comes in contact with the phantom.
- *NDI Target Location:* The locating of the NDI target, from the beginning of the experiment.

4.3 Experiment 3 - Deformation due to Needle insertion

In this experiment, the tissue deformation tracking algorithm is tested out against the locale deformation presented by the needle insertion.

For this experiment, the US probe is manually placed on the phantom, as to have full contact with it. Then a target is selected manually on the screen, as to activate the tracker. The needle will then be inserted via the NOM but with it not being activated. This is just to ensure that the needle is inside the X-Z plane of the end-effector frame. The needle can then be guided above, below or through the target. At that point, the needle should be moved up and down, as to have a lot of tissue deformation. Lastly, the needle should be taken out.

4.3.1 Analysis

System Parameters

- *Needle location:* This refers to if the needle will be placed above, underneath, or through the target.
- *Target location:* This is the location of the target in millimeters, expressed in the end-effector frame. It does not have any notable effects on the results. However, it should be kept constant for all test setups of this experiment.

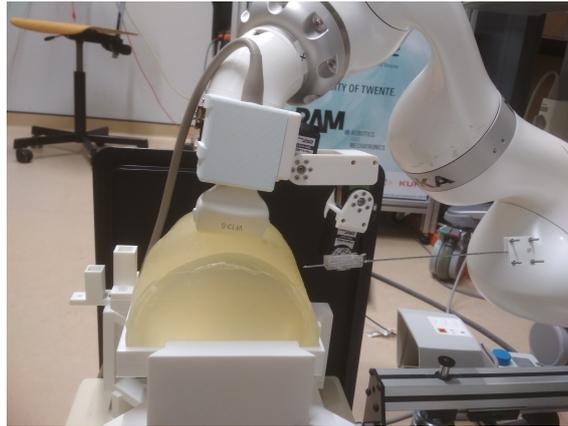


Figure 4.3: Initial configuration of Experiment 3.

Test Setups

In this experiment, two different tests will be done, in which the needle location will change. In the first test, the needle will go underneath the selected target location. In the second test, it will go through it. The first test will show how well local deformation created by the needle can be handled by the proposed tissue deformation tracking algorithm. The second test is of interest because the needle will "break" up the speckle pattern that the tracking algorithm is following. Passing the needle from above will not be tested since - apart from a small change in image intensity - it gives similar results as passing the needle below the target. The only difference that could be noted is that when passing above, the speckles in the tracked region will be less intense since some US waves will be cut-off from the needle itself. However, this does not offer any additional information about the tracking algorithm and during a biopsy process, the needle is not expected to go above the target. In table 4.4 has the synopsis of this test. Figure 4.3 presents the initial configuration of both test cases.

System Parameters	Test 1	Test 2
Location of insertion	Below	Through
T _{target} location	(0, 0, 25)	(0, 0, 25)

Table 4.4: Synopsis of all Test Setups for Experiment 3

Test Variables

- *Estimated Target Coordinates:* The estimated location of the target is of interest and how well that point is tracked from the local deformation of the tissue. Of particular interest is the difference of the tracked location in the beginning (before the needle is inserted) and at the end (when the needle is extracted).

4.4 Experiment 4 - Needle Orientation Controller

This experiment is to validate the quality of NOM controller. By combining the errors measured in Experiment 4.3, the error due to the controller can be determined.

For the experiment, the US probe will be placed manually on the phantom to have full contact. Following that, the NDI target will be placed inside the phantom. The coordinates given by the sensor will be expressed in the zero/robot frame, and consequently given to the NOM controller. Said controller will be then activated. The NDI needle will be then placed inside the NOM and slowly inserted into the phantom. The needle will be pushed until the target is reached. Following that, the needle is extracted from the phantom.

4.4.1 Analysis

System Parameters

- *Needle detection method*: Two methods have been implemented in this thesis for determining the needle's orientation. The first is using the motor encoders of the NOM in combination with its forward kinematics. The other is the use of the Hough Transformation by using the US images.
- *NDI target location*: This is the location of the NDI target in meters, expressed in the zero/robot frame. It does not have any effects on the results. However, it should be kept constant for all test setups of this experiment.

Test Setups

Two test setups exist for this experiment. In the first case, the NOM encoders will be used as the feedback signal for the NOM controller. In the second case, the Hough Transformation will be used to detect the needle inside the US images, and then that information will be used as the feedback for the NOM controller. Because the Hough transform is dependent on the US image having a relatively clear line in it, the second test will be broken down to two sub-test. The only difference is that the target will change between then. This is because after inserting the needle 5 times, it is expected that too many marks will be left in the phantom, that will be seen in the US image as lines. The location of the target is not relevant but should be constant throughout the whole experiment. Table 4.5 presents a synopsis of the tests. The setup of the material is the same as in the previous experiment (Figure 4.3).

System Parameters	Test 1	Test 2a	Test 2b
Needle detection method	NOM encoders	Hough Trans.	Hough Trans.
NDI target location	(0.36, -104.0)	(0.2, -95.3)	(0.6, -95.1)

Table 4.5: Synopsis of all Test Setups for Experiment 4

Test Variables

- *Needle tip coordinates*: The coordinates of the tip of the needle that is inserted.
- *NDI target coordinates*: The coordinates of the NDI target.

4.5 Experiment 5 - Complete Workflow

In this experiment, the whole process, from beginning to end, will take place, in one go.

For this experiment, the NDI target will first be placed inside the phantom. The initial position of the robotic arm is not relevant. Once the target is given, the whole process will start. After the robotic arm has placed the US probe on the phantom, the NOM controller will be activated. At this point, the NDI needle will be placed inside the phantom, via the NOM. After the target is reached, the needle is taken out.

4.5.1 Analysis

System Parameters

Since this experiment is basically performing Experiments 2 and 4 sequentially, all of those parameters can be consider parameters of this system. We can assume that the system parameters that worked best for those individual experiments, will work the best in this one as well. With that in mind, the parameters with the best results will be used for this experiment.

Test Setups

The test setup will start as the one seen in 4.2. Once proper contact with the phantom happens and the tissue deformation tracking algorithm is initialized, the configuration depicted in Figure 4.3 will follow. Table 4.6 present the chosen parameters.

System Parameters	Test
Needle detection method	NOM encoders
Contact percentage region	20%-35%
Probe speed	2mm/sec
Target location	(-8.5, 0.4, -108.3)

Table 4.6: Synopsis of all Test Setups for Experiment 5

Test Variables

- *Needle tip coordinates:* The coordinates of the tip of the needle that is inserted.
- *NDI target coordinates:* The coordinates of the NDI target.

4.6 Experiment 6 - Needle Impedance Controller Simulation

In this final experiment, the Needle Impedance Controller is simulated. The idea is show that the idea about the impedance control will suit the needs that it is intended for.

4.6.1 Analysis

System Parameters

- *Rotational spring stiffness:* The stiffness value of the rotation spring placed in the insertion point.
- *Disturbance force:* The value of the linear force that represent the force the radiologist would apply to the back part of the needle.
- *Desired coordinates:* The 2-D Cartesian coordinates of the Insertion point and the rotation of the needle around said point.

Test Setups

Three simulations will take place. In all cases, the desired coordinates will stay the same. In the first simulation, no external forces will be applied, as to see if the controller reaches the desired position and rotation. In the second and third simulation, the same linear force will be applied to the needle but with two different stiffness parameters for the rotating spring. Everything is expressed in the End-Effector frame. The linear forces will be along the Z axis.

System Parameters	Test 1	Test 2	Test 3
Rotational spring stiffness	1	1	20
Disturbance Force	0N	3N	3N
Desired Coordinates $[X, Z, \theta]$	[0.065, 0.02, -2.79]	[0.065, 0.02, -2.79]	[0.065, 0.02, -2.79]

Table 4.7: Synopsis of all Test Setups for Experiment 5

Test Variables

- *Needle Orientation:* The needle orientation, in the end-effector frame, from the point it is inserted.
- *Virtual Needle Orientation:* The Cartesian coordinates of the point of the needle that should be positioned at the insertion point.

5 Results

In this chapter, the results of the experiments will be presented.

5.1 Experiment 1 - Tracking tissue deformation due to US probe contact

In figures 5.1, 5.2 and 5.3 the tracked position, in test 1, of point 1, 2 and 3 are plotted, expressed in the end-effector frame. Each figure has the tracked position of that point for all 10 experiments. Figures 5.4, 5.5, and 5.6 present the same quantities but for test 2. The mean absolute differences between the initial and final location of the target is presented in tables 5.1 and 5.2, for each test. Lastly, the difference between the initial and final values of each execution, for each point, can be found in Appendix F.

It should be pointed out that for the X-axis, because of a bug found in the code after the completion of the experiments, the initial values given was not the same. That being said, what is of interest is the error between the first and final measurement, for each execution. In that regard, the plots show that the tracing is consistent with what was expected.

Looking at the mean absolute errors, the effect that the distance between the target and the US probe can be observed. The bigger the distance, the more error is introduced into the tracking algorithm. This is expected since when force is applied to the phantom, which make the tissue move around, the microscopic structures between the US probe and the target change position, resulting in a change of the speckle's intensity around the area of interest. Changes in speckle intensity, make Optical Flow's tracking more difficult.

Lastly, the effect of the location and direction of the US probe's motion can be seen by comparing the mean absolute error between test 1 and test 2. Although the results of test 1 are a bit better, the difference is very small, almost insignificant in most cases.

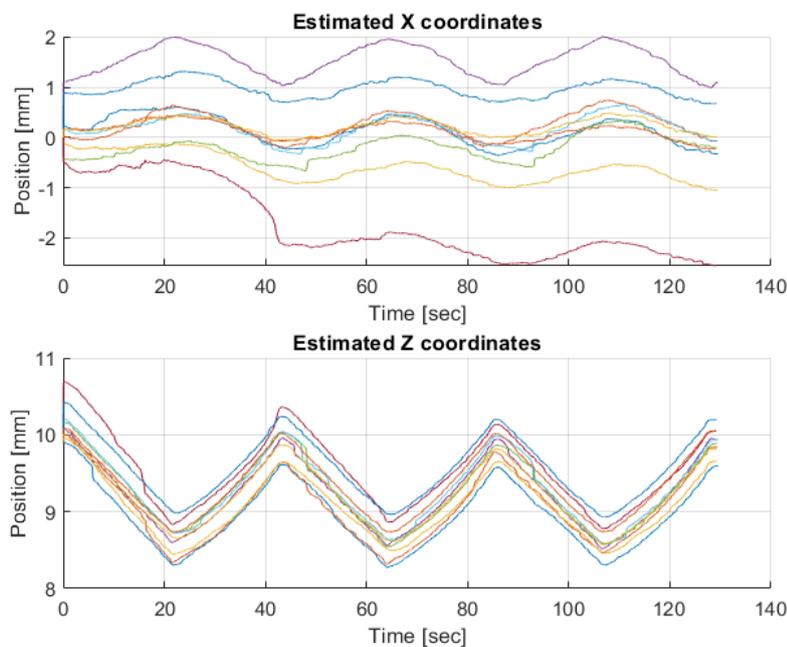


Figure 5.1: Experiment 1 - Test 1 - Point 1

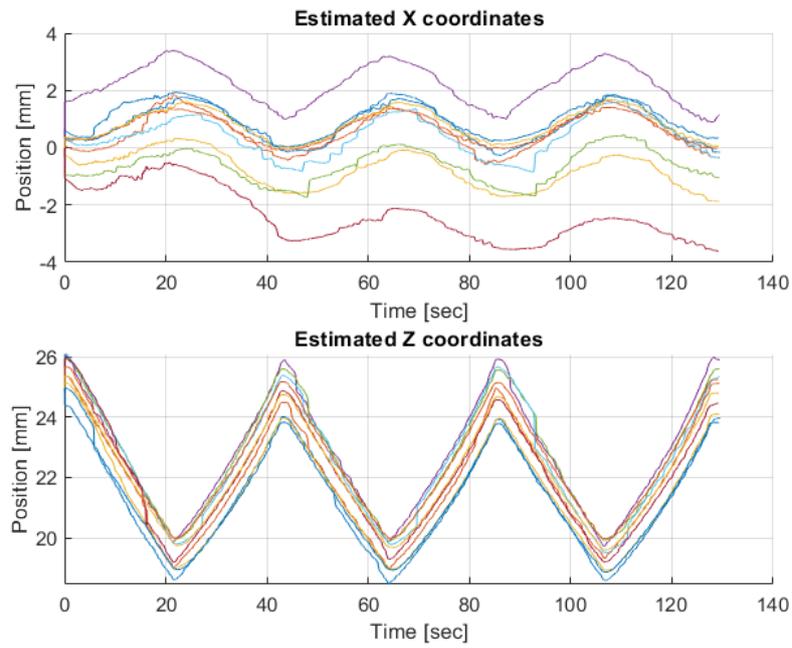


Figure 5.2: Experiment 1 - Test 1 - Point 2

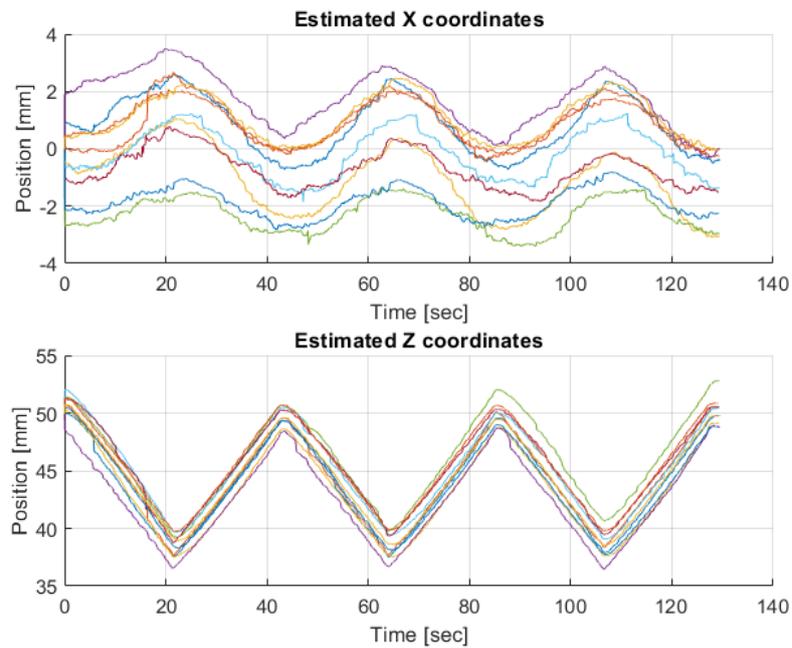


Figure 5.3: Experiment 1 - Test 1 - Point 3

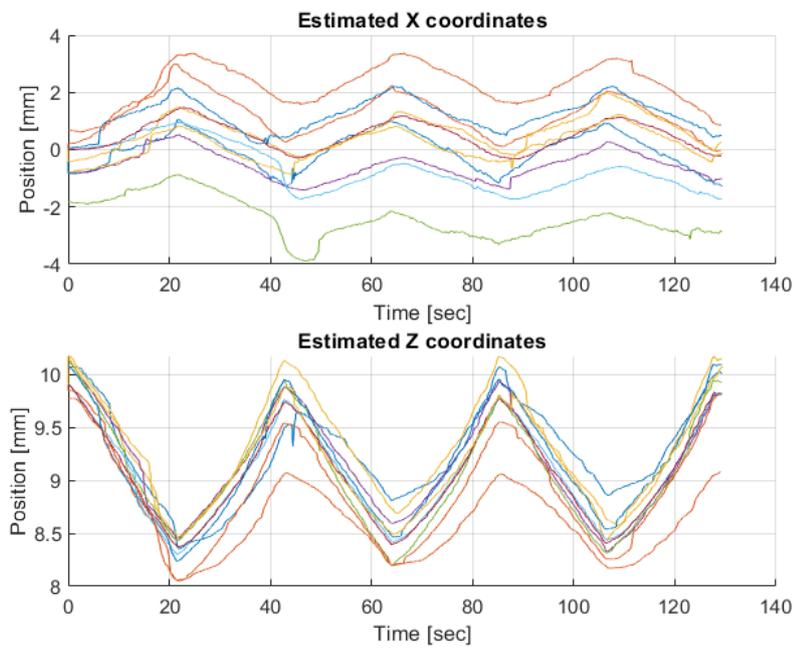


Figure 5.4: Experiment 1 - Test 2 - Point 1

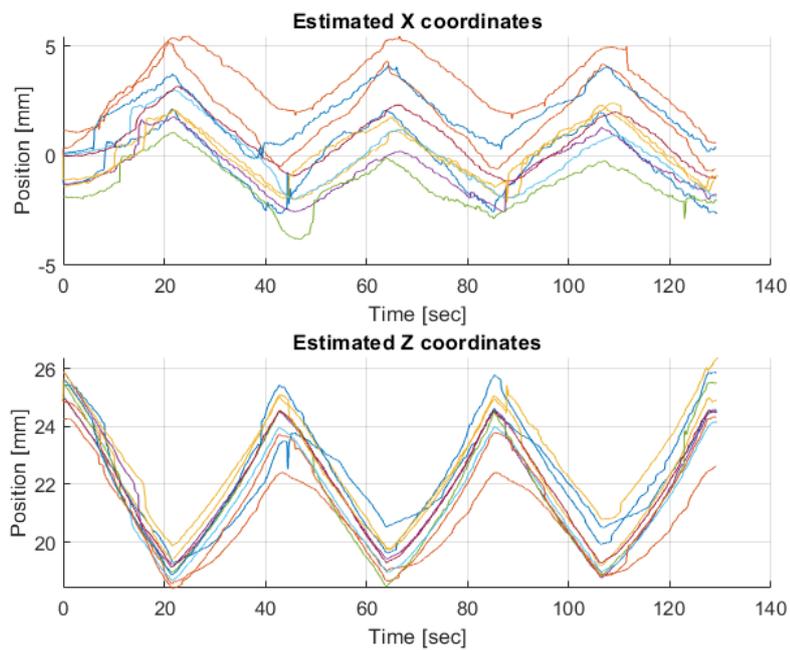


Figure 5.5: Experiment 1 - Test 2 - Point 2

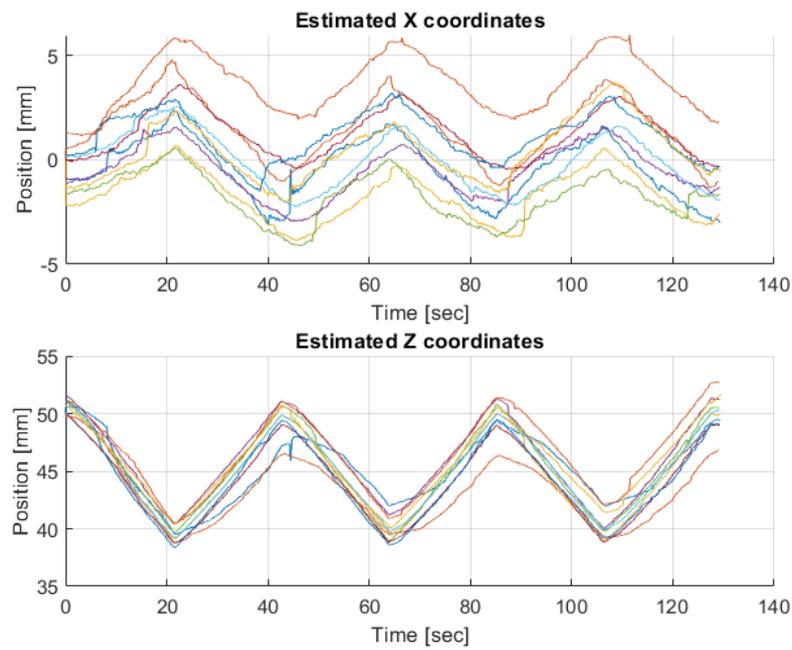


Figure 5.6: Experiment 1 - Test 2 - Point 3

Point	X mean [mm]	Z mean [mm]
1	0.4857	0.2485
2	0.7007	0.6552
3	0.8606	0.8274

Table 5.1: Test 1 - Mean absolute difference between initial and final value

Point	X mean [mm]	Z mean [mm]
1	0.5857	0.1648
2	0.7251	0.7352
3	0.8537	1.0748

Table 5.2: Test 2 - Mean absolute difference between initial and final value

5.2 Experiment 2 - Initialization Phase

In Figure 5.7 the estimated position of the target by using the tissue deformation tracking algorithm is presented. In Figure 5.8 the position given by the NDI target sensor is presented. The blue squares represent the final values. For all plots, each color represents a different execution of the experiment. All data are expressed in the robot base frame. The measurements of each execution, which share the same color, for both the estimated position and the NDI target position, start and end at the same time.

Regarding the optical flow estimation, the initial values, seen as a small line on the left side of the plots, are the initial position of the target once the first contact between the US probe and the phantom is established, as discussed in the initialization phase algorithm. The final values are seen at the end of the measurement of each execution as a small straight line. Tables E.7 and E.8 show the initial and final values in meters for each execution, respectively.

Because of the noise presented in the NDI target, it is not clear what is the exact initial and final value. For that reason, the final value of each execution is defined as the mean values of the

NDI measurements for the time period where the US probe is not moving. This period is the same time for which the initial and final values seen in Figure 5.7 are straight lines. The initial and final values of the NDI target in meters are seen in tables E.9 and F.10, respectively.

In tables F.11 and F.12 the difference between the estimated target location and the NDI measurements is presented, for both the initial and final values, respectively. The Mean Absolute Error for the initial position and final position is seen in table 5.3.

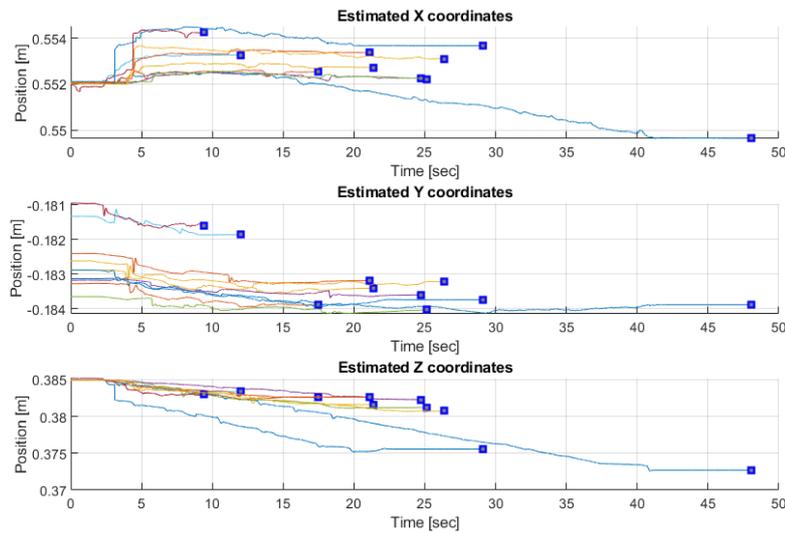


Figure 5.7: Experiment 2 - Optical Flow estimation

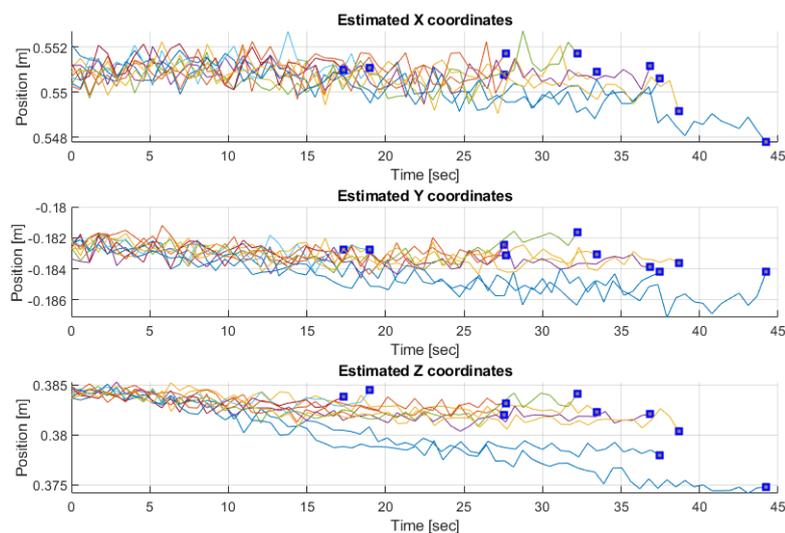


Figure 5.8: Experiment 2 - NDI coordinates

	X coordinate [mm]	Y coordinate [mm]	Z coordinate [mm]
Initial Position	1.0279	0.5925	1.2334
Final Position	2.1233	0.8029	0.9654

Table 5.3: Mean Absolute Error of Initial and Final values

5.3 Experiment 3 - Deformation duo to Needle insertion

The estimated coordinates of the target can be seen in Figure 5.9 for test 1, where the needle was placed underneath the target, and Figure 5.10 for test 2, where the needle went through the target. The blue squares represent the final values. The Mean Absolute Error between initial and final estimation for each test can be seen in table 5.4. The final values for each execution, of Test 1 and Test2, are in tables F.13 and F.14, respectively.

The results of test 1 show a consistent behavior from the tissue deformation tracking algorithm, even when the deformations are done locally by the needle.

Comparing the absolute mean error between what is seen in test 1 of this experiment and both tests of Experiment 1, it could be argued that local deformations due to the needle present more error than uniform deformation of the whole phantom does. It should be kept in mind that during this experiment, the needle's motions were much more sudden and aggressive than what is expected of a biopsy process. This was so as to test the limits of the algorithm. As a result, this mean error can be considered the maximum error that will be introduced in the tracking algorithm due to the needle.

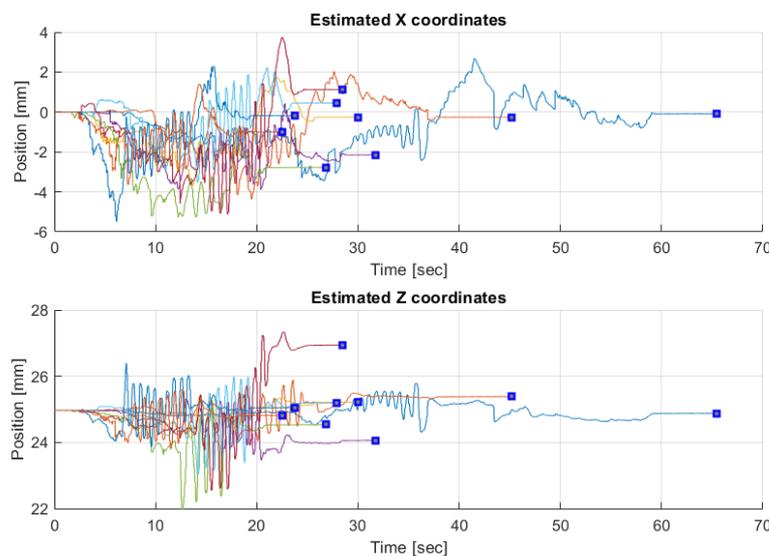


Figure 5.9: Experiment 3 - Test 1 - Estimated Coordinates

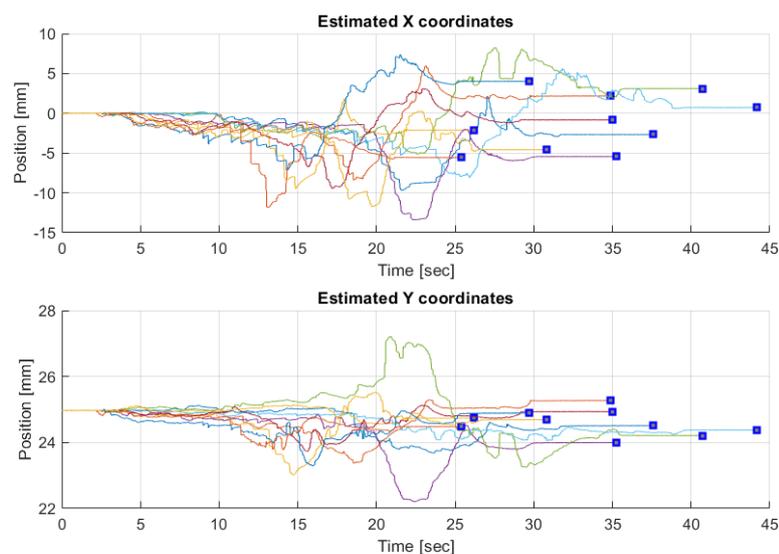


Figure 5.10: Experiment 3 - Test 2 - Estimated Coordinates

Test	X coordinate [mm]	Z coordinate [mm]
1	0.9137	0.4999
2	3.1178	0.4190

Table 5.4: Mean Absolute Error between initial and final position

5.4 Experiment 4 - Needle Orientation Controller

In Figures 5.11 and 5.12, the measured location of the target and needle tip for test 1 can be seen. These Figures are expressed in the frame of the Aurora sensor. Similarly, in figures 5.13 and 5.14, the measured locations of the target and the needle tip, for test 2, can be seen. The blue squares represent the final values. In Figures 5.12, 5.13, and 5.14, some of the sub-plots have been zoomed in. This is because final position, e.g. when the needle reaches the target, is of interest, and plotting the whole path makes it hard to see it. In table 5.5, the Mean Absolute Error between the final position of the target and of the needle tip, for both tests, can be seen. In table 5.6 and average value of the shortest distance between the line orientation and the target for each execution is shown. The final values of the needle and NDI target can be found in detail in the appendix tables F.15, F.16, F.17, and F.18.

Comparing the final position of the target with the needle tip in figures 5.11 and 5.12, the order, from top to bottom, for each axis, generally match with each other. This implies relative consistence in the behaviour of the controller. Furthermore, the NOM controller's commands affects mostly the motion along the Z-axis, while the Y-axis is more affected by how far the needle is inserted. This explains why in the Z-axis of the needle tip, the final locations are more close to each other, compared with those of the Y-axis that are more spread out. Additionally, a bit of bias can be noticed along the Z-axis. The target averages around -104.1 mm, while the needle does so around -104.7 mm. This error could be due to the experiment setup or improper calibration of the sensors.

In test 2, where the Hough Transform is used, the motion of the needle tip, as seen in Figure 5.14, has more oscillation compared to test 1. Regarding the order of the final values, between figures 5.14 and 5.13, it is relatively consistent again but not as much as what was observed in test 1. Looking at table 5.5, the error is much higher. About 0.8 millimeter.

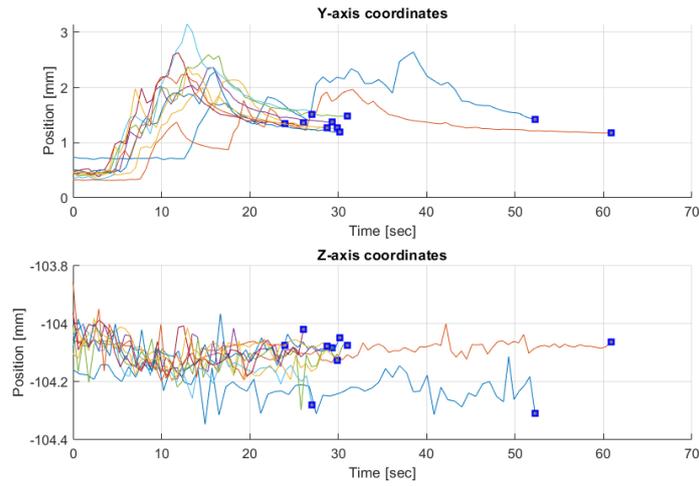


Figure 5.11: Experiment 4 - Test 1 - Coordinates of NDI target

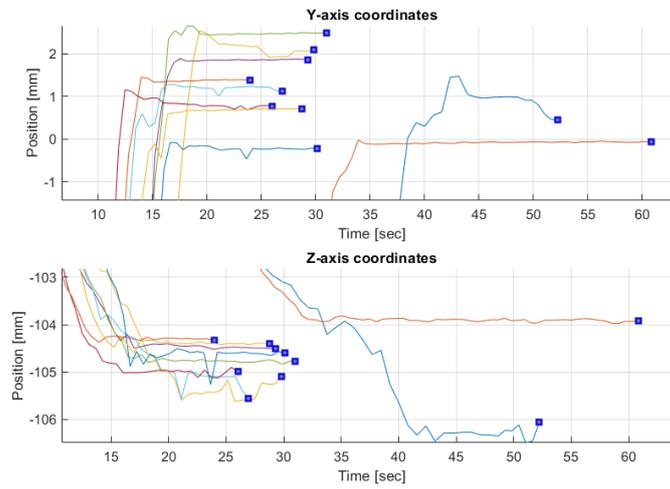


Figure 5.12: Experiment 4 - Test 1 - Coordinates of needle tip

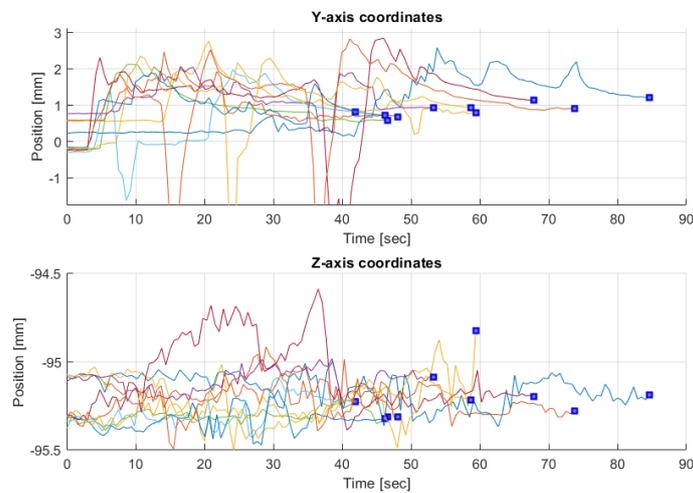


Figure 5.13: Experiment 4 - Test 2 - Coordinates of NDI target

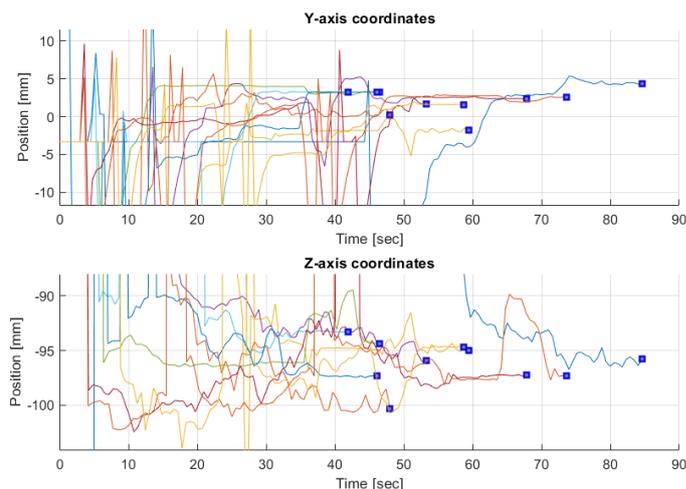


Figure 5.14: Experiment 4 - Test 2 - Coordinates of needle tip

Test	Y coordinate [mm]	Z coordinate [mm]
1	0.7229	0.7573
2	1.8091	1.6131

Table 5.5: Mean Absolute Error between needle tip and NDI target, after needle is fully inserted.

Test	Average Distance [mm]
1	0.7610
2	1.5347

Table 5.6: Shortest average distance between target and needle line.

5.5 Experiment 5 - Complete Workflow

In Figures 5.15 and 5.16, the measured location of the target and needle tip, can be respectively seen. The blue squares represent the final values. These Figures are expressed in the frame of the Aurora sensor. In Figure 5.16, all axis coordinate have been zoomed in since the final position (e.g. when the needle reaches the target) is of interest. In table 5.7, the Mean Absolute Error between the target and the needle tip location can be seen. In table 5.8 and average value of the shortest distance between the line orientation and the target for each execution is shown. Furthermore, a bias can be noticed at the Z-axis, of about 2.5 mm. This would suggest that the system is accurate along this axis but not as precise as the other two axis. The final values of the needle and NDI target can be found in detail in the appendix tables F.19 and F.20.

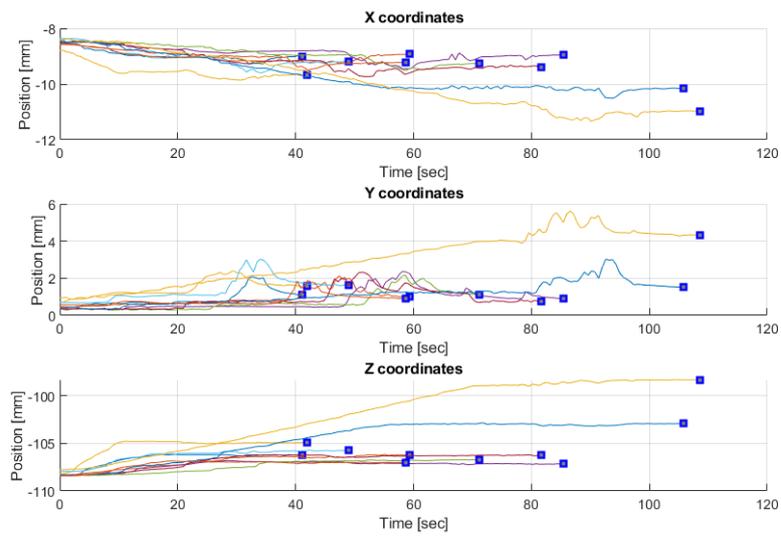


Figure 5.15: Experiment 5 - Coordinates of NDI target

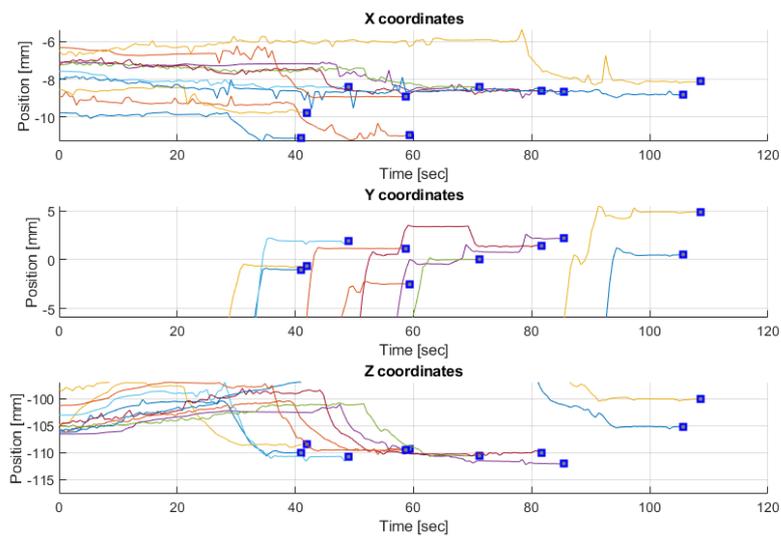


Figure 5.16: Experiment 5 - Coordinates of Needle tip

X coordinate [mm]	Y coordinate [mm]	Z coordinate [mm]
1.1472	1.3094	3.4739

Table 5.7: Mean Absolute Error between needle tip and NDI target, after needle is fully inserted.

Distance [mm]
2.8922

Table 5.8: Shortest average distance between target and needle line.

5.6 Experiment 6 - Needle Impedance Controller Simulation

The final values that the coordinates reached can be seen in table 5.9. The angle of the needle, the position of the needle that is at the insertion point, and the external forces applied to the needle, can be seen in Figures 5.17, 5.18, and 5.19, for each simulation. It can be seen that regardless of if an external force is applied, the Cartesian coordinates stay in position, as is desired. In test 1, where no external force is applied, it reaches the desired angle, performing like the NOM controller would. Lastly, in test 2 and 3, where the rotational spring stiffness changes, the final angle of the needle is different.

Coordinates	Test 1	Test 2	Test 3
X	0.065	0.065	0.065
Y	0.020	0.020	0.020
θ	-2.79	-3	-2.83

Table 5.9: The final values of the coordinates.

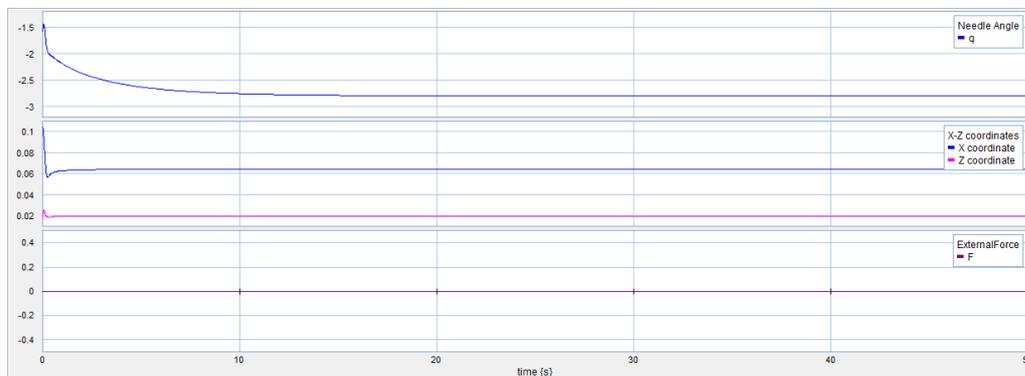


Figure 5.17: Test 1 - Simulation of the angle of the needle, the coordinates of the needle at the insertion point, and the external force

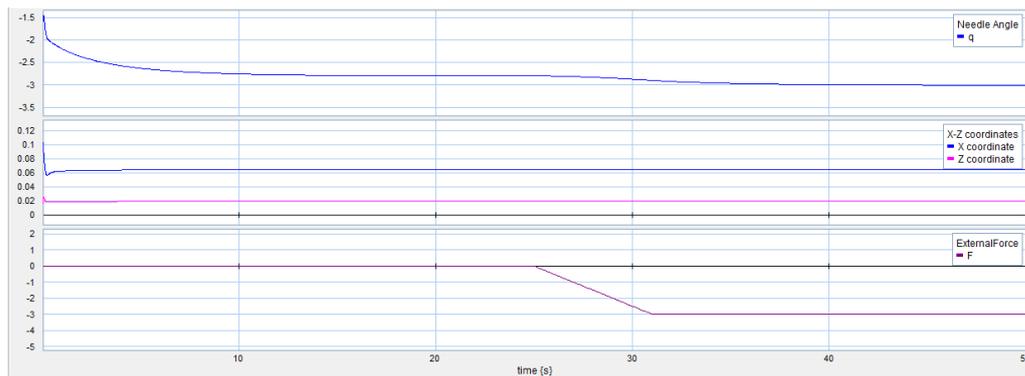


Figure 5.18: Test 2 - Simulation of the angle of the needle, the coordinates of the needle at the insertion point, and the external force

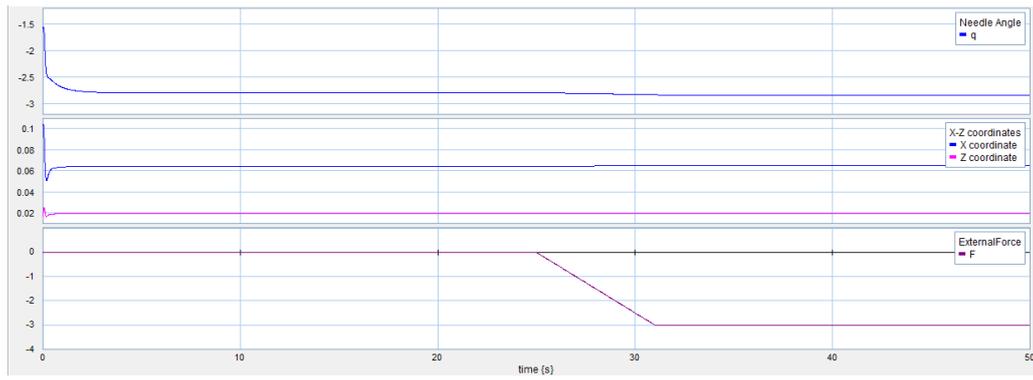


Figure 5.19: Test 3 - Simulation of the angle of the needle, the coordinates of the needle at the insertion point, and the external force

6 Discussion

In this chapter, the results are discussed, a few outliers are explained, and an overall assessment of the control design is made.

In experiments 1 and 3 the tissue deformation tracking algorithm was examined under very intense conditions. Intense in the sense that the pressure that the US probe applied to the phantom, and the motions of the needle, were far more than what could be expected from an actual biopsy session. But even in those conditions, there was sub-millimeter accuracy from the algorithm. This shows that Optical Flow is indeed a valid option for tracking tissue motion and has potential to be used in the final MURAB robotic system.

At this point, the quality of the initialization phase will be examined. Looking at table 5.3, the initial position error is key here, as this is when tracking first starts. In other words, this is the error that is introduced by the initialization phase. The mean absolute error is around 1mm, which means that the combined error by the tissue tracking algorithm and the NOM controller needs to be at most 1mm, if 2 mm targets are to be successfully extracted. An interesting observation is that the Z estimation has more error than that of the X estimation. At the same time the tissue deformation tracking algorithm's error for the Z estimation is usually less than that of the X estimation. If the X errors and the Z error are added, it is less than 2 mm.

When looking at the error introduced to the tracking system between the initial and final position (table 5.3), it is noticed that X received a disproportionately bigger amount of error than what Y or Z receives. This can be partially explained by the small error that is introduced by the tissue deformation tracking algorithm. However, not all the error can be accounted by that. The reason more than 1 mm of average error is presented in the X-axis is that when the first contact is done, the estimated target is not perfectly in the center of the US probe. This results in the target being close to the unseen areas from the US probe, as demonstrated in Figure 6.1. Because of that, when the US probe is pushed further in, the black walls on the right and left side of the US image move away from the center. The optical flow, thinking this is part of region it should track, follows this motion for a bit. It is expected that if the target is more in the center, the increase of error along the X-axis will be more close to what was seen at the Z-axis. The reason for the target not being in the center is most likely due to a code bug. Unfortunately, this bug has not yet been identified. Additional debugging of the implementation is needed, which is expected to reduce the final error that the initialization phase introduces to the system. Furthermore, because of the nature of the error in the X-axis, it is assumed that the error seen in the other two axes is a better representation of the capabilities of the initialization phase. Based on that, the overall error that the initialization phase introduces is low. Additionally, integrating the MRI scan in the process is expected to reduce the error even more.

The NOM controller presented some promising results. The Cartesian Mean Average Error is not only sub-millimeter, as seen in table 5.5, but also consistent with the results found in (Welleweerd, 2018). Furthermore, in table 5.6 the average distance between the target and the orientation of the needle allows examining how the error is while neglecting error due to the needle being inserted too much or too little. It is not very different from what would be expected based on the error of table 5.5. This implies that the way the target is indicated on the screen allows for the needle to be properly inserted by the eye.

When applying the needle detection algorithm to compensate for the artificial error added to the system, most of it was properly compensated but not enough to match the results of the properly calibrated NOM controller. The error is still under the 2mm wanted result but taking in to account that some error will also be introduced by the initialization phase and the tissue tracking algorithm, it is unlikely that the 2mm target will be reached in the current implement-

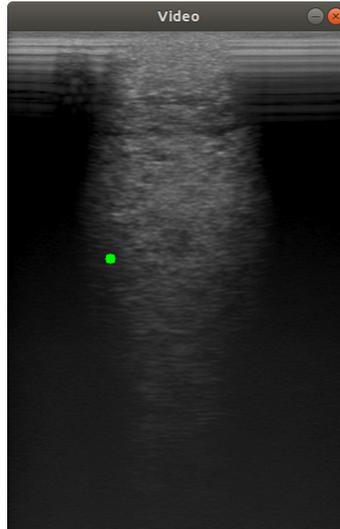


Figure 6.1: Tracked target (green dot) located near an area where the US probe does not have a visual

ation. Additionally, the average shortest distance between the needle and the target (table 5.6) is higher than what was seen when the NOM controller was used on its own. However, during this test, the needle was starting with a significant error, of about 10 millimeters. Based on what is seen, the Needle Detection algorithm can reduce that error significantly. That being said, the error is not reduced to a degree that it would make a significant difference if the needle is slightly bent. It will, however, probably help to reduce the error that could be there from bad settings of the NOM motors.

Looking at Experiment 5, a judgment of the overall system can be made. According to table 5.7, the Z-axis has a significantly higher error than the X-axis and Y-axis. Generally speaking, it is expected that the Z-axis will have the highest error. This is because of how each part of the workflow contributes an error to a different axis:

- *Y-axis errors* = initialization phase error.
- *X-axis error* = initialization phase error + Optical Flow error + small part of NOM controller error.
- *Z-axis error* = initialization phase error + Optical Flow error + most part of NOM controller error.

So the Z-axis will most likely have a higher error than any of the other two axis. However, if the bias seen in the data is accounted for, this error is reduced further.

Additionally, the average shortest distance observed in table 5.8 shows the error of the system while neglecting the error due to inserting the needle too much or not enough.

Even with the bias seen in the Z-axis, it is able to accurately hit targets as low as 6 mm diameter. These results are comparable with other robotic biopsy systems, like what was seen in (Megali et al., 2001) and (Abayazid et al., 2015).

Finally, the simulation of the impedance controller gives results that would allow the radiologist to have more control over the direction of the needle. That being said, it should be kept in mind that the simulation does give ideal results. Impedance control cannot be as accurate as position control. So the position and orientation of the needle will probably not achieve this accuracy in a real implementation. Regarding the accuracy of orientation, this would mean that the accuracy of hitting the target will also fall. In that case, the radiologist will have a

bigger responsibility for the accuracy of the biopsy process than the robotic system. Since the radiologists are responsible for the success of the biopsy process, it is good to give them the option to, in a sense, override the robotic system.

7 Conclusion

The goal of this thesis was to develop a control strategy for properly orientating the biopsy needle towards the desired target. The research objective set for this was:

Design and implement an initialization protocol for the robotic arm and a control strategy for tracking the desired target, while properly orientating the needle towards that direction

To achieve this research objective, the following tasks were set:

- Design and implement the motion of the Robotic arm in such a way as to allow the US probe to start properly tracking the given target.
- Design and implement a tracking tissue deformation algorithm using a model-free approach.
- Design and implement the controller of the needle orientation mechanism
- Design and implement compliant behaviour for the needle orientation.

The initialization phase, which moves the robotic arm and starts tracking the target, was designed and implemented.

We developed and implemented the algorithm that correctly positions the US probe on the phantom and starts properly tracking the target. To do this, two controllers would adjust the rotation of the US probe and also the amount of force that the US probe applied to the phantom, using confidence maps. Additionally, since the motion of the robotic arm was based on impedance control, a PI controller was created to minimize the trajectory error. Experiments have shown that an error around 1mm was achieved. However, the implementation of the part that controls the rotation of the US probe still needs improvement.

Since the target is assumed to be too small to directly distinguish from the US images, the tissue, located at the position of the target, will be tracked. Using optical flow as the core of the tissue deformation tracking algorithm, proved to be a good choice. The results from the experiments show that a sub-millimeter accuracy was achieved, for both deformations caused by the US probe and the needle.

Using the aforementioned tracking system, the controller that aimed the needle, by using the NOM, was designed. Results gave sub-millimeter accuracy when it came to hitting the target. An issue that could have occurred, was that the controller did not take in to account possible needle deformation. For that reason, a needle detection algorithm was tested out. It uses the Hough transformation in its core, which detects the line that represented the needle in the US images. Experimenting with the method showed that it was able to reduce significant errors, which the NOM controller was not able to do on its own. However, the error was not reduced to acceptable levels, which shows that further optimization of the needle detection algorithm is needed.

Furthermore, an impedance controller was designed, that would allow the radiologist to have more control over the direction of the needle. This was unfortunately not implemented on the real system, because the currently used motors are not the most appropriate for this task. However, a simulation proved that in principle, using impedance control is a valid way to share the control of the needle between the radiologist and robot.

Lastly, when combining the above implementations in one full workflow, the overall control strategy was able to present accuracy that would successfully hit down to 6 mm targets. The proposed solutions show that it has the potential of leading to a highly accurate control strategy

that can be used in real examinations. Together with the proposed impedance controller, this can lead to a system which accurately targets detected lesions and still allows the radiologist to be fully in control. As a result, the system will be more easily integrated by radiologists in real medical situations.

7.1 Future work

To support on going work in the MURAB project, we propose the following recommendations to further improve the proposed control strategy:

1. **Integration of MRI scans:** When presenting the initialization phase, the MRI scan was part of the process. However, because of not guaranteed access to an MRI scanner, it was chosen not to integrate that step into the implementation, an manually get that information. It is recommended that the MRI scan is integrated into the process, as described in the initialization phase, to increase the accuracy of the needle control and automate the biopsy process even more.
2. **Orientation controller of US probe:** As was mentioned already, in the experiment of the initialization phase, the significantly bigger error observed in the x-axis, was expected to be due to how the US probe is orientated around the target. This is most likely an implementation issue that needs to be investigated and solved.
3. **Optimize needle detection algorithm:** The needed detection algorithm, although work in principle, does not reduce the error enough for acceptable results. This part of the system needs to be further examined as to create better results. Additionally, it may be that a more sophisticated needle detection algorithm is needed.
4. **Implementation of impedance controller:** The impedance controller has been proven to work in principle. The only thing left is for the physical implementation to take place.

A Tissue Deformation Tracking

In this appendix chapter, some technical information are presented, regarding algorithms used in the tissue deformation tracking system.

A.1 Optical Flow

Optical flow is the motion of objects, surfaces or edges, between consecutive frames sequences, caused by the relative movement between the object and camera. A different way to see optical flow is as the distribution of apparent velocities of movement of brightness patterns in an image. There are two main assumptions on which all Optical Flow solutions are based on. First, each pixel's brightness value stays relatively constant, throughout the whole process and secondly that very small motions take place between two frames. While US speckles are usually considered to be noise from an imaging point of view, they are in fact resulting from the coherent reflection of microscopic structures contained in soft tissue (Krupa et al., 2009). As such, speckles are coherent. That means that they can be used to satisfy the first Optical Flow assumption. The second assumption will be addressed later on.

The general problem of Optical Flow can be seen in Figure A.1, where $I(x, y, t)$ is the pixel's intensity value, with coordinates (x, y) at time t and (dx, dy) is the displacement of those pixels after time dt .

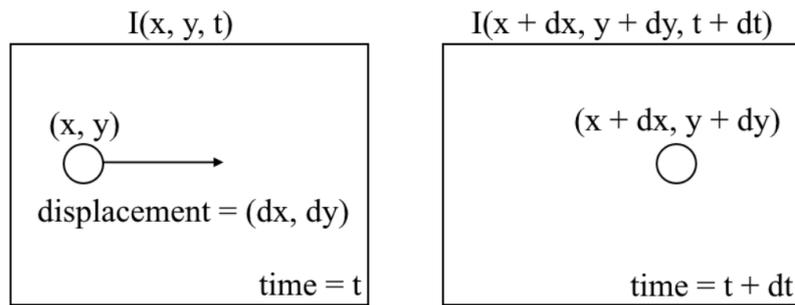


Figure A.1: Optical Flow problem

In Figure A.1, the pixel of interest is located at coordinates (x, y) at time t and that same pixel is located at $(x + dx, y + dy)$ at time $t + dt$. This would give the following relationship:

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (\text{A.1})$$

Then if the Taylor Series Approximation is applied to the right side element, the following is reached:

$$I(x + dx, y + dy, t + dt) = I(x, y, t) + \frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt + H.O.T.^1 \quad (\text{A.2})$$

By combining A.1 and A.2, the following equation is produced:

$$\frac{\partial I}{\partial x} dx + \frac{\partial I}{\partial y} dy + \frac{\partial I}{\partial t} dt = 0 \quad (\text{A.3})$$

¹High Order Terms

Lastly, by deviding with dt , the final equation is achieved:

$$\frac{\partial I}{\partial x} u + \frac{\partial I}{\partial y} v + \frac{\partial I}{\partial t} = 0 \quad (\text{A.4})$$

where $u = dx/dt$ and $v = dy/dt$. The variables (u, v) are the estimation velocity of the given pixel, along the x-axis and y-axis, equivalently. Equation A.4 is what is referred to as the basic Optical Flow problem. The reason being, there are two unknown variables that are wanted, (u, v) , while there is only one equation. There have been different proposals on how to solve that, like (Farnebäck, 2003) and (Horn and Schunck, 1981). A popular one solution, and the one that will be used in this thesis, is the Lucas-Kanade method (Lucas and Kanade, 1981).

A.2 Lucas-Kanade method

This method was developed by Bruce D. Lucas and Takeo Kanade. There are two assumptions that are the base of this method. Firstly that the flow in a local neighbourhood of pixels is very small between to sequential frames. Secondly, that all optical flow equations for all the neighbourhood of pixels, are solved using the least square criterion.

This method resolves the inherent ambiguity of the optical flow equation by combining information from several nearby pixels. A window is defined, with its center being the pixel that is being tracked, which defines which neighboring pixels will be used. The bigger the window, the more robust the tracking is to noise but with the introduction of more motion blur (Chunke et al., 1996). Given that the chosen window contains n pixels, the following Optical Flow equations can be written:

$$\begin{aligned} \frac{\partial I(q_1)}{\partial x} u + \frac{\partial I(q_1)}{\partial y} v &= -\frac{\partial I(q_1)}{\partial t} \\ \frac{\partial I(q_2)}{\partial x} u + \frac{\partial I(q_2)}{\partial y} v &= -\frac{\partial I(q_2)}{\partial t} \\ &\vdots \\ \frac{\partial I(q_n)}{\partial x} u + \frac{\partial I(q_n)}{\partial y} v &= -\frac{\partial I(q_n)}{\partial t} \end{aligned} \quad (\text{A.5})$$

where $q_n = (x_n, y_n)$ are the pixels inside the window.

These equations can be written in the matrix form $Ak = b$, where:

$$A = \begin{bmatrix} \frac{\partial I(q_1)}{\partial x} & \frac{\partial I(q_1)}{\partial y} \\ \frac{\partial I(q_2)}{\partial x} & \frac{\partial I(q_2)}{\partial y} \\ \vdots & \vdots \\ \frac{\partial I(q_n)}{\partial x} & \frac{\partial I(q_n)}{\partial y} \end{bmatrix} \quad k = \begin{bmatrix} u \\ v \end{bmatrix} \quad b = \begin{bmatrix} -\frac{\partial I(q_1)}{\partial t} \\ -\frac{\partial I(q_2)}{\partial t} \\ \vdots \\ -\frac{\partial I(q_n)}{\partial t} \end{bmatrix} \quad (\text{A.6})$$

The above system has more equations than unknowns and thus it is over-determined. The Lucas-Kanade method obtains a compromise solution by the least squares principle, namely it solves the 2x2 system:

$$A^T A k = A^T b \implies k = (A^T A)^{-1} A^T b \quad (\text{A.7})$$

That is, it computes:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_i \left(\frac{\partial I(q_i)}{\partial x}\right)^2 & \sum_i \frac{\partial I(q_i)}{\partial x} \frac{\partial I(q_i)}{\partial y} \\ \sum_i \frac{\partial I(q_i)}{\partial x} \frac{\partial I(q_i)}{\partial y} & \sum_i \left(\frac{\partial I(q_i)}{\partial y}\right)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i \frac{\partial I(q_i)}{\partial x} \frac{\partial I(q_i)}{\partial t} \\ -\sum_i \frac{\partial I(q_i)}{\partial y} \frac{\partial I(q_i)}{\partial t} \end{bmatrix} \quad (\text{A.8})$$

where $i = 1, 2, \dots, n$.

A.3 Solving the small motion assumption

As mentioned, one of the assumptions of Optical Flow is that the motion of the pixel being tracked between two frames has to be small. In many cases, however, that is not the case. To solve this problem, the so-called "Pyramids" can be used.

A pyramid is a type of multi-scale signal representation in which a signal or an image is subject to repeated smoothing and subsampling. Pyramid representation is a predecessor to scale-space representation and multiresolution analysis. Thanks to this, big motions are scaled down to seem as if they are small. An illustration of it can be seen in Figure A.2.

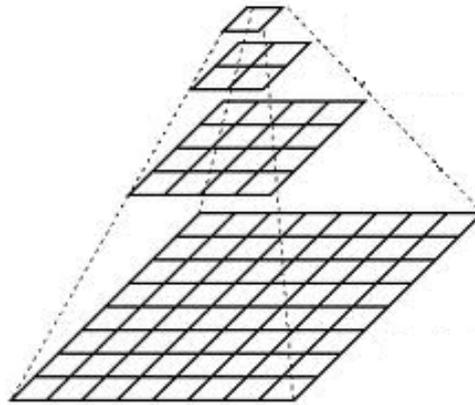


Figure A.2: Illustration of a pyramid in image processing terms

The use of a pyramid is defined by how many layers are to be used. When the image moves from one layer to the next, first the image is smoothed out and the subsampling occurs. Depending on the number of layers chosen, this process is repeated. The more layers exist, the smaller all motions will be perceived as. A variate of different smoothing kernels can be used for each layer, like the Gaussian blur, the Laplacian kernel, and the binomial kernel. The subsampling can also be adjusted to any ration desired but a typical value is 0.5.

A.4 Implementation of Tissue Deformation Tracking in the NOM controller

In this section, how the above tracking system was implemented into the NOM controller will be described. During the initialization phase, when the US probe has centered the target with minimum contact, the tracking algorithm initializes. Afterward, when the NOM controller is fully initialized, the following loop takes place:

1. Tracked point is converted from pixels to millimeters and expressed in the end-effector frame.
2. By combining the target coordinates and the defined insertion point, the orientation and location of the needle is defined.
3. By using the inverse kinematics of the NOM, the angles of the NOM are calculated.
4. Angles are sent to motors which perform their own position control.

5. Motion of the orientation of the needle or further insertion of the needle will result in tissue deformation.
6. Tracking algorithm estimates new position.
7. Back to 1

The above loop is done indefinitely, until needle is removed and biopsy process is completed. In Figure 2.5 the control system using the needle detection algorithm is illustrated.

B Confidence Map

In the appendix chapter, a small description about confidence map is presented.

Confidence maps were introduced in (Karamalis et al., 2012). They graphically represent what is the confidence for each pixel in the image field of view (Ω) of an US probe, which emphasizes uncertainties in attenuated and shadow regions. Confidence is depicted with 1.0 (white) for complete confidence in that pixel and 0.0 (black) for a complete lack of confidence. The confidence estimation is created by using a random walks framework, by taking into account US specific constraints. By definition, all elements of the top row, which indicate the beginning of the scan, are set to 1.0. They represent the virtual transducer elements. All elements at the end of the scan, i.e. the bottom row, have their value set to 0.0. The confidence $C(p)$ of all other pixels $p \in \Omega$ is defined as the probability that an echo, originating from pixel p , reaches one of the transducer elements. An estimation of this probability is performed within the random walk framework (Grady, 2006). A simplified model of the physics of US propagation in soft tissues, including attenuation and absorption, is used to calculate the probability that a random walk - starting from a certain pixel - reaches a virtual transducer element.

C Needle Detection

In this section the process of detecting the needle from the US images will be described.

For the NOM controller to work properly, the location of the needle needs to be known at all times. That can happen by reading the encoders. However, it is possible that due to inaccuracies of the kinematic model or because the needle has bent, that the estimated position is incorrect. A way to solve this problem is by detecting the location of the needle directly by the US images. The needle appears as a white line in the US images. With that in mind, the line detection algorithm Hough Transform (Duda and Hart, 1972) is used to determine the location of the needle. Additionally, some pre-processing is required for the Hough Transform to work properly with the US images. As to make a clear distinction where the needle is located, the edge detection algorithm Canny (Canny, 1986) is first applied on the image feed, before given to the Hough Transformation algorithm. This helps enhance the contrast between the needle and the tissue.

C.1 Canny Edge Detection

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a wide range of edges in images. It helps to extract useful structural information from images and dramatically reduces the amount of data to be processed afterwards. John Canny found that the requirements for the application of edge detection in different types of images are relatively similar. As a result, an edge detection algorithm that addresses these requirements can be implemented in several different situations. The general criteria for edge detection include:

- Detect accurately as many existing edges as possible from the image.
- The edge point that is detected should be located in the center of the actual edge.
- An edge should be detected only once and image noise should not be interpreted as edges.

The Canny edge detection algorithm satisfies the above requirements by using the calculus of variations, a technique that finds the function which optimizes a given function. The optimal function used is described by the sum of four exponential terms. Alternatively, it can be approximated by the first derivative of a Gaussian.

The Canny edge detection algorithm can be broken down to the following five steps:

1. Apply Gaussian filter to smooth the images and reduce noise.
2. Find the intensity gradients of the image.

The smoothed image is filtered with a Sobel kernel in both horizontal and vertical direction to get the first derivative in horizontal direction (G_x) and vertical direction (G_y). From these two images, we can find the edge gradient and the direction for each pixel as follows:

$$Edge_Gradient(G) = \sqrt{G_x^2 + G_y^2}$$

$$Angle(\theta) = \tan^{-1} \frac{G_y}{G_x}$$

3. Apply non-maximum suppression to get rid of spurious response to edge detection.

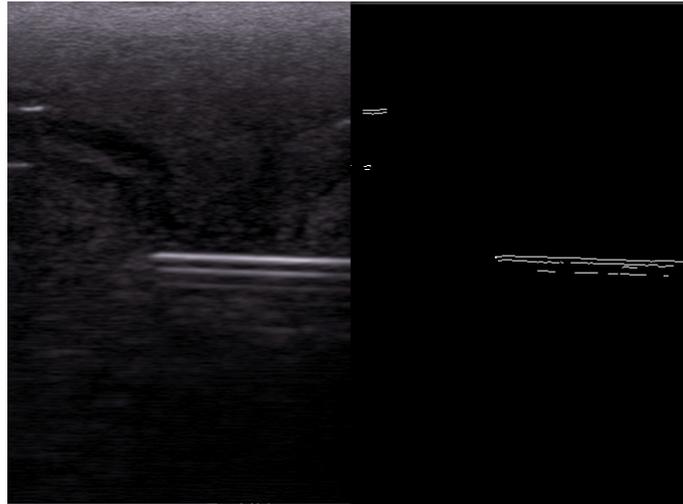


Figure C.1: Result of Carry Edge Detection. Left image is input and right image is result

After getting gradient magnitude and direction, a full scan of the image is performed to remove any unwanted pixels which may not constitute an edge. For this, each pixel is checked if it is a local maximum in its neighborhood in the direction of the gradient.

4. Apply double threshold to determine potential edges.

After the application of non-maximum suppression, remaining edge pixels provide a more accurate representation of real edges in an image. However, some edge pixels remain that are caused by noise and intensity variation. To account for them, edge pixels with a weak gradient need to be filtered out while edges with a high gradient values should be preserved. This is done by selecting high and low threshold values. If an edge pixel's gradient value is higher than the high threshold value, it is marked as a strong edge pixel. If an edge pixel's gradient value is smaller than the high threshold value and larger than the low threshold value, it is marked as a weak edge pixel. If an edge pixel's value is smaller than the low threshold value, it will be suppressed.

5. Track edge by hysteresis: Finalize the detection of edges by suppressing all the other edges that are weak and not connected to strong edges.

From the previous step, some of the detected weak edges may be due to noise. Usually, a weak edge pixel caused by true edges will be connected to a strong edge pixel while noise responses are unconnected. With that in mind, blob analysis is applied by looking at a weak edge pixel and its 8-connected neighborhood pixels. If there is at least one strong edge connected to the weak edge, it is preserved, otherwise, it is discarded.

In image C.1 the Canny edge detection algorithm is applied to an US image with the needle inserted.

C.2 Hough Transform

The Hough Transform is a feature detection technique invented by Richard Duda and Peter Hart in 1972, based on the 1962 patent of Paul Hough. Its purpose is to detect simple imperfect shapes, usually lines, cycles or ellipses. When having imperfect shapes in images, due to noise or incorrect edge detection, associating which pixels are associated with which shape in non-trivial. The Hough Transform addresses this problem by making it possible to perform grouping of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects.

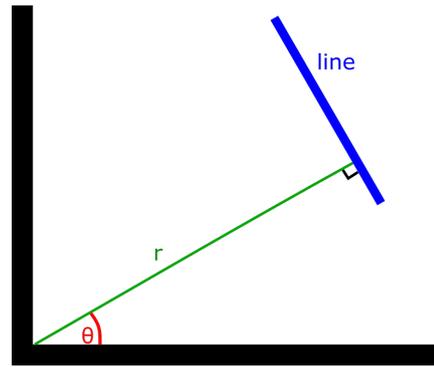


Figure C.2: Representation of a line in polar coordinates

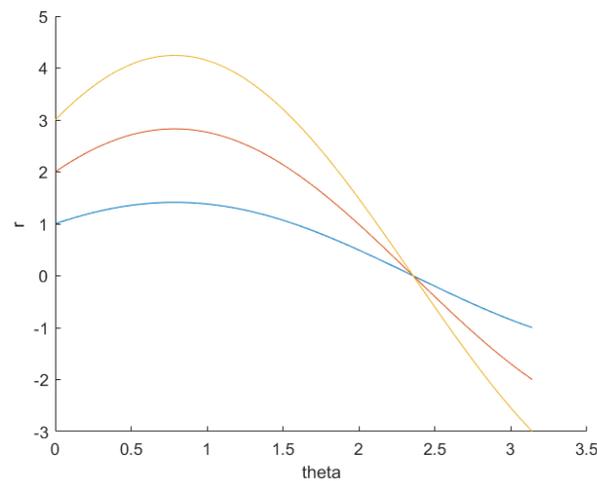


Figure C.3: The sinusoids produces in the (r, θ) from points with coordinates $(1,1)$, $(2,2)$ and $(3,3)$

For the purposes of this application, we only focus on line detection. The reason for choosing to detect a line is because the part of the needle that can be seen by the US probe will virtually have no deformation. This means that the orientation that this part of the needle has, is a result of deformation presented in the rest of the needle. The equation of a straight line in 2D is represented as:

$$y = m \cdot x + b \quad (\text{C.1})$$

with a point (b, m) in the parameter space. However, vertical lines pose a problem as they would give rise to unbounded values of the slope parameter m . For that reason, (Duda and Hart, 1972) propose the use of the following form:

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (\text{C.2})$$

where r is the distance from the origin to the closest point on the straight line and θ is the angle between the x axis and the line connecting the origin with that closest point, as depicted in Figure C.2. Therefore, it is possible to associate each line in the image with a pair of (r, θ) . Knowing that, for a single point, all possible straight lines going through it will create a sinusoidal curve in the (r, θ) space. Each point on that curve represents a different line. A set of two or more points that form a straight line will produce sinusoids which cross at the (r, θ) for that line. This is illustrated in Figure C.3 for three points located on the same line. Thus, the problem of detecting collinear points can be converted to a problem of finding concurrent curves.

C.3 Incorporate Needle Detection in Control Loop

In this section, how the needle detection algorithm is incorporated inside the control loop is described. The challenges with embedding this algorithm inside the control loop are mainly two:

1. The needle needs to be first inserted inside the phantom and be in the field of view of the US probe
2. The needle may not be capable of being detected the whole time, as stated in (Kaya et al., 2015). For that reason, the NOM controller should still be able to function with these small gaps of knowledge.

With that in mind, it needs to assist the existing NOM controller, rather than replacing it. To do that, the algorithm tries to correct the orientation of the needle by setting a correction offset to it. While the needle is not detected, the NOM controller will aim the needle using the NOM's inverse kinematic model. However, when the needle is detected, and an error is calculated, an offset will be set to compensate for that error. If later on the needle is lost again because of low quality US images, the offset will stay as is, to help the NOM controller correctly hit the target. In Figure 2.6 the control system using the needle detection algorithm is illustrated.

D Mathematics of Needle Orientation Mechanisms

This chapter explains how to derive the kinematic and dynamic models of a serial manipulator, in the framework of Screw Theory. These models will be used to design the NOM controller and also to simulate the behavior of the impedance controller.

The basic idea of screw theory is for the analysis of spatial mechanisms. A screw consists of two three-dimensional vectors. A screw can be used to represent the position and orientation of a spatial vector, the linear and angular velocity of a rigid body, or the forces and torques applied to a rigid body. Therefore, the concept of a screw is convenient for calculating the kinematics and the dynamics of a mechanism. Additionally, it is easy to transform between the screw-based methods and the vector and matrix methods. When applied in mechanism analysis, screw theory has the advantages of clear geometrical concepts, explicit physical meaning, simple expression, and convenient algebraic calculation.

D.1 Kinematics of Serial Manipulators

This section describes how to derive the kinematics of a serial manipulator using Screw Theory. Two categories can be distinguished, direct (or forward) kinematics and differential kinematics. Direct kinematics produce the position of the defined End-Effector in the Cartesian space, given the angles/positions of the joints. Differential kinematics gives the velocity of the End-Effector, given the angular/linear velocity of the joints. Another useful term is inverse kinematics, where the angles/positions of the joints are to be calculated given a desired End-Effector position.

D.1.1 Direct Kinematics

A generalized serial manipulator can be seen in Figure D.1. It consists of the base, n links, and n joints. The base frame is Ψ_0 which does not move and the End-Effector frame is Ψ_{ee} .

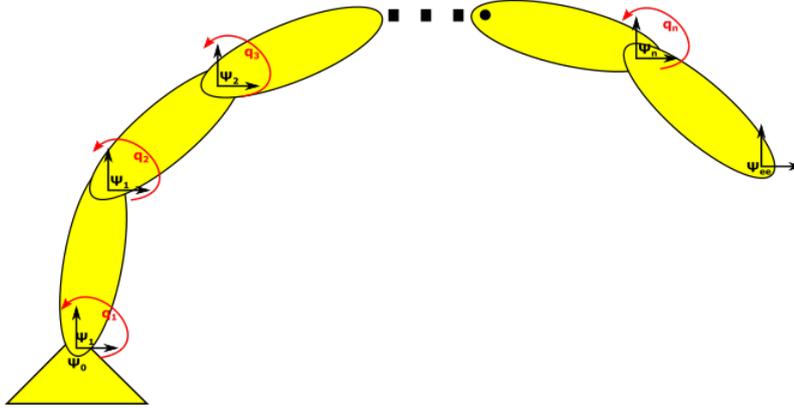


Figure D.1: Serial Manipulator

To change the frame in which a point is expressed, a Homogeneous Transformation Matrix can be used. Given that there are the frames Ψ_i and Ψ_j , and point p^i needs to be expressed in frame Ψ_j , the following matrix can be used:

$$\begin{pmatrix} p^j \\ 1 \end{pmatrix} = \begin{pmatrix} R_i^j & p_i^j \\ 0_{1 \times 3} & 1 \end{pmatrix} \cdot \begin{pmatrix} p^i \\ 1 \end{pmatrix} \implies P_j = H_i^j P_i \quad (\text{D.1})$$

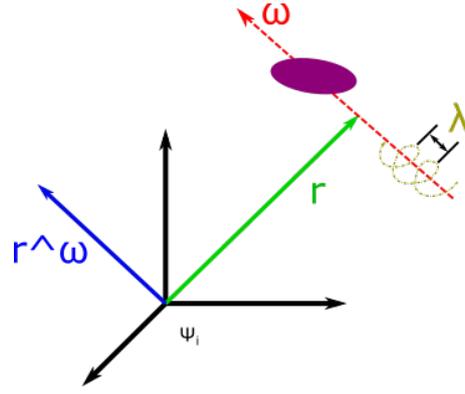


Figure D.2: A Twists based on Mozzi's Theorem

Where $p^i \in \mathbb{R}^3$ is the point expressed in frame Ψ_i , $p^j \in \mathbb{R}^3$ is the point expressed in frame Ψ_j , $P^i \in \mathbb{R}^4$ is the projected space of p^i , $P^j \in \mathbb{R}^4$ is the projected space of p^j , $p_i^j \in \mathbb{R}^3$ is the translation needed to go from the origin of Ψ_j to the origin of Ψ_i and $R_i^j \in \mathbb{R}^{3 \times 3}$ is the rotation between the same to frames. R_i^j belong to the Special Orthonormal Group, $SO(3)$, which is a Lie Group of orthonormal matrices with determinant 1:

$$SO(3) = \left\{ \mathbf{R} \in \mathbb{R}^{3 \times 3}; \mathbf{R}^{-1} = \mathbf{R}^T; \det(\mathbf{R}) = 1 \right\} \quad (\text{D.2})$$

H_i^j is the Homogeneous Transformation Matrix that defines the pose of frame Ψ_i relative to Ψ_j . This is part of the Special Euclidean Group, $SE(3)$, which is also a Lie Group:

$$SE(3) = \left\{ \begin{pmatrix} R_i^j & p_i^j \\ \mathbf{0}_{1 \times 3} & 1 \end{pmatrix}; R_i^j \in SO(3), p_i^j \in \mathbb{R}^3 \right\} \quad (\text{D.3})$$

A useful property of Homogeneous Transformation Matrices is the chain rule, as seen in equation D.4. This means that if a series of frames exist, all that is needed to connect the first and last frame is to calculate the Transformation Matrix of each frame relative to each previous one.

$$H_n^0 = H_1^0 H_2^1 H_3^2 \dots H_n^{n-1} \quad (\text{D.4})$$

According to Mozzi's theorem (Mozzi, 1763), a generalized way of representing the displacement of rigid bodies is by a translation along a line, which is preceded by a rotation along an axis which is parallel to said line. This is illustrated in Figure D.2 This is referred to as a Twists:

$$T = \begin{pmatrix} \omega \\ v \end{pmatrix} = \begin{pmatrix} w \\ r \wedge w \end{pmatrix} + \lambda \begin{pmatrix} 0 \\ \omega \end{pmatrix} \implies \tilde{T} = \begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix} \quad (\text{D.5})$$

where $T \in \mathbb{R}^6$ is the Twists, $\omega \in \mathbb{R}^3$ is the angular velocity of the screw, $v \in \mathbb{R}^3$ is the translation along the screw axis, $r \in \mathbb{R}^3$ is the closest distance between the origin of the frame in which the screw is expressed and the screw axis, and λ is a scalar defining the translation for each full rotation.

On the right side of equation D.5, the tilde form of the Twist is shown. The tilde form of ω is:

$$\tilde{\omega} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix} \quad (\text{D.6})$$

The tilde operator offers the following property:

$$x \wedge y = \tilde{x}y \quad (D.7)$$

The tilde form of ω is part of the $so(3)$, which is the Lie group of $SO(3)$:

$$so(3) = \{\tilde{\omega} \in \mathbb{R}^{3 \times 3} : -\tilde{\omega} = \omega^T\} \quad (D.8)$$

The tilde form of the Twists is part of the $se(3)$, which is the Lie Group of $SE(3)$:

$$se(3) = \left\{ \begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix} : \tilde{\omega} \in so(3), v \in \mathbb{R}^{3 \times 3} \right\} \quad (D.9)$$

This means that if the H matrix is a differential function of time, \tilde{T} can be directly calculated from it:

$$\tilde{T} = \dot{H}_i^j H_j^i \quad (D.10)$$

Twists change depending on the frame in which they are expressed. The notation used for twists is $T_a^{b,c}$, which means that it is the twists of frame a , relative to frame c , expressed in frame b . To change the frame of reference, the adjoint is used:

$$Ad_{H_i^j} = \begin{pmatrix} R_i^j & 0 \\ \tilde{p}_i^j R_i^j & R_i^j \end{pmatrix} \quad (D.11)$$

Where $R_i^j \in \mathbb{R}^{3 \times 3}$ is the rotation matrix from frame Ψ_i to Ψ_j and $\tilde{p}_i^j \in \mathbb{R}^{3 \times 3}$ is the tilde form, as defined in equation D.6, of the translation vector from Ψ_i to Ψ_j . To express the Twists of Ψ_i relative to Ψ_0 , from Ψ_i to Ψ_j , equation D.12 is used.

$$T_i^{j,0} = Ad_{H_i^j} T_i^{i,0} \quad (D.12)$$

The kinematic model of a one joint rotational robot, with base frame Ψ_j and frame Ψ_i attached to the rigid body, is defined as:

$$H_i^j(q_i) = e^{\tilde{T}_i^{j,j} q_i} \cdot H_i^j(0) \quad (D.13)$$

Where q_i is the angel of the joint, $\tilde{T}_i^{j,j}$ is a Tilde unit Twists, $e^{\tilde{T}_i^{j,j} q_i}$ is the exponential of the Twists, and $H_i^j(0)$ is the Homogeneous Transformation Matrix between frame j and frame j when the joint has not rotated (i.e. the initial configuration). The unit Twist for rotational joints is:

$$\hat{T}_i^{j,j} = \begin{pmatrix} \hat{\omega} \\ r \wedge \hat{\omega} \end{pmatrix} \quad (D.14)$$

The exponential of the Twists is defined as:

$$e^{\begin{pmatrix} \tilde{\omega} & v \\ 0 & 0 \end{pmatrix} q} = \begin{pmatrix} e^{\tilde{\omega} q} & \frac{1}{\|\tilde{\omega}\|^2} ((I - e^{\tilde{\omega} q})(\omega \wedge v) + \omega^T v \omega) \\ 0 & 1 \end{pmatrix} \quad (D.15)$$

where $e^{\tilde{T}_i^{j,j} q_i}$ is calculated using the Rodriguez formula (Rodrigues, 1816):

$$e^{\tilde{\omega} q} = I + \tilde{\omega} \sin(q) + \tilde{\omega}^2 (1 - \cos(q)) \quad (D.16)$$

By combining equations D.4 and D.13, Brockett's exponential formula for direct kinematics can be derived:

$$H_n^0(q_1, q_2, \dots, q_n) = e^{\tilde{T}_1^{0,0} q_1} \cdot e^{\tilde{T}_2^{0,1} q_2} \dots e^{\tilde{T}_n^{0,(n-1)} q_n} H_n^0(0) \quad (D.17)$$

This means that if all frames attached to rigid bodies are located at the point through which the Twists of the body passes, the only information required is the direction of the Twist and the $H_i^{i-1}(q_1)$ matrices. All other information can be extracted from these.

D.1.2 Differential Kinematics

Differential kinematics relates the joint velocity with the Twist of the End Effector:

$$T_n^{0,0} = J(q) \dot{q} \quad (D.18)$$

Where $J(q) \in \mathbb{R}^{6 \times n}$ is the Geometric Jacobian and $q \in \mathbb{R}^n$ are the angles of the joints. The Jacobian can be calculated using the following equation:

$$J(q) = (T_1, T_2, \dots, T_n) \quad (D.19)$$

where:

$$T_i = Ad_{H_{i-1}^0} \hat{T}_i^{(i-1), (i-1)} \quad (D.20)$$

To calculate the joint velocities, given a desired End-Effector velocity, the Jacobian needs to be inverted: $\dot{q} = J^{-1}(q) T_n^{0,0}$. However, the kinematic map of a redundant manipulator is a surjective function, not injective, which means that the solution of \dot{q} is not unique. Additionally, it could happen that for some configurations of q , the rank of the Jacobian will drop. If that is the case, its inverse can become very big which leads to high joint velocities, which in turn can be dangerous for the robot and its surroundings.

D.2 Dynamics of Robot Manipulator

In this section, the dynamics of serial manipulators will be described in the context of Screw Theory, which relates the End-Effector forces with the joint torques. The force and torque vectors can be expressed in a screw motion by something called a "wrench". According to Poinsot's Theorem, any forces and torques applied to a body can be written as a pure linear force along a line and a pure moment around that same line, as illustrated in Figure D.3. The equation of a Wrench is:

$$W = \begin{pmatrix} \tau \\ F \end{pmatrix} = \begin{pmatrix} r \wedge F \\ F \end{pmatrix} + \lambda \begin{pmatrix} F \\ 0 \end{pmatrix} \quad (D.21)$$

where $W \in \mathbb{R}^6$ is the Wrench, $\tau \in \mathbb{R}^3$ is the torque of the screw, $F \in \mathbb{R}^3$ is the linear force along the screw axis, $r \in \mathbb{R}^3$ is the closest distance between the origin of the frame in which the screw is expressed and the screw axis, and λ is a scalar defining the translation due to the momentum.

A wrench is a co-vector, which is a linear operator that maps vectors to real numbers:

$$P = W \cdot T \quad (D.22)$$

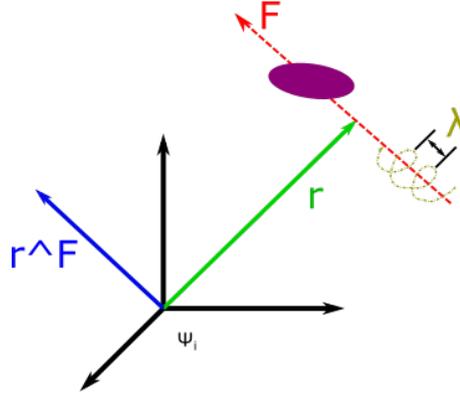


Figure D.3: A Wrench based on Poinso's Theorem

Where P is Power, which is a scalar number that is independent of any coordinate frame, T is the Twist, and W is the wrench. To change the frame in which the wrench is expressed, the transpose of the adjoint, as defined in equation D.11, is used again:

$$(W^i)^T = Ad_{H_i}^T(W^i)^T \quad (D.23)$$

The Lie Group structure allows to describe motions independently of the configuration of the body.

D.2.1 Equation of Motion of a Rigid Body

In this section, the equations of motion of a rigid body, in the context of Screw Theory, will be derived. These equations relate the motion of the principal inertia frame of a rigid body with the sum of forces and torques acting on said body.

It can be shown that any rigid body of any shape, density, and material, behaves as a uniform ellipsoid of mass m and principal inertias j_x, j_y , and j_z , centered and oriented in Ψ_k :

$$\mathcal{J}^i = \begin{pmatrix} J^i & 0 \\ 0 & mI \end{pmatrix} \quad (D.24)$$

where $J^i \in \mathbb{R}^{3,3}$ is a diagonal matrix with the principal moment of inertia on its diagonal and $I \in \mathbb{R}^{3,3}$ is the identity matrix. Similarly to Euler's first law of motion, $p = mv$, the dynamics of a rigid body can be expressed employing a momentum screw:

$$(\mathcal{P}^i)^T = \mathcal{J}^i T_i^{i,0} \quad (D.25)$$

Where $\mathcal{P}^i \in \mathbb{R}^3$ is the moment screw. \mathcal{P} is defined as a row vector because it transforms line a wrench, i.e. it is a co-vector. Furthermore, in a similar way to the rate of change of momentum ($\dot{p} = F$), the rate of change of momentum screw $\dot{\mathcal{P}}^{0,i}$ can be expressed in the inertial frame Ψ_0 :

$$\dot{\mathcal{P}}^{0,i} = W^{0,i} \quad (D.26)$$

The following expression can be used to describe the motion in the principal inertia frame Ψ_k :

$$\dot{\mathcal{P}}^k{}^T = ad_{T_k^{k,0}}^T(\mathcal{P}^k)^T + (W^k)^T \quad (D.27)$$

Where:

$$ad_{T_k^{k,0}} = \begin{pmatrix} \tilde{\omega}_k^{k,0} & 0 \\ \tilde{v}_k^{k,0} & \tilde{\omega}_k^{k,0} \end{pmatrix} \quad (\text{D.28})$$

As a results, equation D.27 can be written as follows:

$$\dot{\mathcal{P}}^k T^T = \mathcal{J}^k \dot{T}_k^{k,0} = \begin{pmatrix} -\tilde{\omega}_k^{k,0} & -\tilde{v}_k^{k,0} \\ 0 & -\tilde{\omega}_k^{k,0} \end{pmatrix} \mathcal{J}^k T_k^{k,0} + (W^k)^T \quad (\text{D.29})$$

D.2.2 General Dynamics of Robotic Manipulators

A serial robotic manipulator can be considered as a system of rigid bodies that are interconnected by joints. These joints introduce constraints for the relative motion of the rigid bodies. The equations of motion of an n-link rigid body manipulator can, in their most simplistic canonical form, be written as the following non-linear second-order differential equation:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q)^T = \tau^T \quad (\text{D.30})$$

where q , \dot{q} , and $\ddot{q} \in \mathbb{R}^n$ are the generalized joint position, velocity, and acceleration vectors of the manipulator, $M(q) \in \mathbb{R}^{n \times n}$ is a symmetric and positive definite mass matrix, $C(q, \dot{q}) \in \mathbb{R}^n$ represents the Coriolis and centrifugal forces, $G(q) \in \mathbb{R}^n$ represents the applied gravitational forces, and $\tau \in \mathbb{R}^n$ is a co-vector of equivalent joint torques due to the motors and interaction forces.

E Interaction Control of Needle Orientation Mechanisms

This chapter explains how implementing an interactive controller for the needle orientation mechanisms, will allow the control of the needle to be simultaneously shared between the robot and the user. The control strategy that will be examined is impedance control. This chapter will describe the implementation of an energy-based impedance controller.

E.1 Dynamic Interaction

Given that the position of the lesion is perfectly known, by using the kinematic model of the orientation mechanism, it is simple to calculate the orientation of the motors as to properly align the needle with said lesion target. However, what happens in the case that the estimate position is incorrect and the radiologist can determine on their own where the target actually is? In this case, the radiologist should be able to properly align the needle with their observed target location. In that case, performing position control would not allow the user to move the needle orientation manually. In that case, applying an interactive controller would allow for the above desired behavior.

Interaction control works by regulating the exchange of power between the controlled system and its environment. In our case, the force applied on the needle by the user can be interpreted as the environment interacting with the orientation mechanisms.

E.2 Impedance Control

Impedance control is an approach to dynamic control relating force and position. This is done by creating a virtual physical system that connects the robotic system with the environment. This virtual system is the actual controller. This way, instead of only controlling one state variable (e.g. force, position, velocity), the impedance of the manipulator is controlled ((Hogan, 1984)). By properly adjusting this relationship, it is believed that the control of the needle can be partially shared by the orientation mechanisms and the radiologist.

E.2.1 Simple Impedance Controller

Initially a simple 1-D mass system will be examined as to illustrate the basic idea of impedance control. In Figure E.1 the system, which is a simple mass, and the impedance controller can be seen. This system corresponds to a second order mass-spring-damper system with mass m , damping b and stiffness k . The controller attempts to minimize the error between the current position of the mass x and the desired position x_d . This is equivalent to a traditional PD controller, where k is the proportional gain and b is the derivative gain.

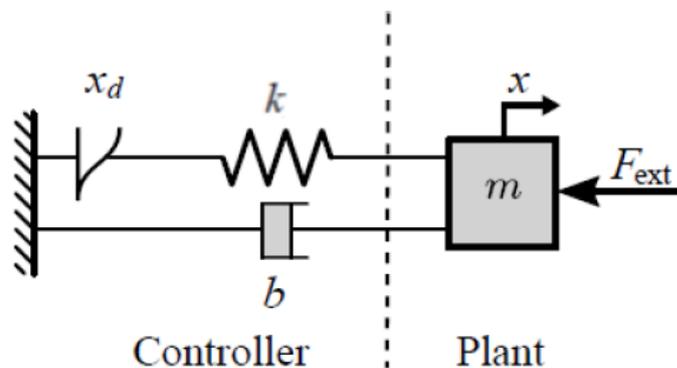


Figure E.1: Simple Mass

The dynamic equation of the this system can be seen in E.1.

$$m\ddot{x} + k(x_d - x) + b\dot{x} = F_{ext} \quad (\text{E.1})$$

With F_{ext} being the external forces acting on the mass. For this controller to work, the position of the mass, x , and the mass's velocity, \dot{x} , are needed. In our system, the motors have encoders that provide the position of the system. However velocity is not provided. The velocity can be calculated from the position by numerical differentiation, state variable filters or observers. That being said, this methods are not very suitable for physical control systems. Differentiation would increase any error embedded in the position measurement, variable filters do not have any direct physical interpretation, and observers require a model of the system and the environment. An other method is one introduced in (Stramigioli, 1996) referred to as the "Damping Injection Framework".

E.2.2 Damping Injection Framework

In Figure E.2 the controller with the Damping Injection can be seen. It is the same system as the one in Figure E.1 with the addition of the spring k_c and the mass m_c . In this system, if $m_c \ll m$ and $k_c \gg k$ then the whole system acts like a second order system ((Stramigioli, 1996)), as seen in E.1. Additionally, since the mass m_c is only a state of the controller, its position can be obtained without the fear of it having error embedded in it. In that case, its derivative can easily be calculated and used for the damping parameter.

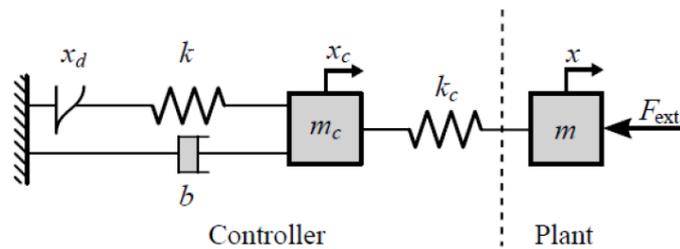


Figure E.2: Impedance Controller with Damping Injection Framework

This framework also allows to passively handle saturation of actuator torques. If the saturation value, F_{sat} , of the motor is known, it is possible to implement a passive non-linear controller spring k_c , as seen in Figure E.3. This ensures that both the controller mass m_c and system mass m will never receive a force from the actuator that is higher than F_{sat} .

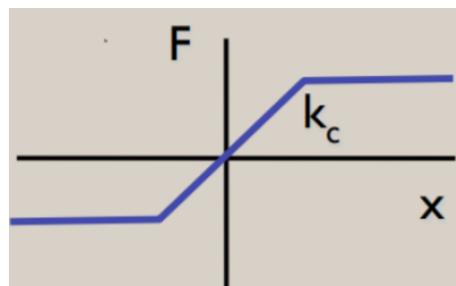


Figure E.3: Passive non-linear spring to account for saturation of actuators

E.2.3 Energy Shaping

Going back to the simple impedance controller introduced in E.2.1, it was mentioned that this virtual system controls the power exchange between the robotic system and the environment.

To do that, the energy functions of the elements should be defined. This introduces the potential energy function of the spring, $V(r)$, and the Rayleigh function for the damper, $R(r)$.

$$V(r) = \frac{1}{2C}r^2 \quad (\text{E.2})$$

$$R(r) = \frac{1}{2}b\dot{r}^2 \quad (\text{E.3})$$

where $r = x - x_d$ and $C = \frac{1}{k}$. C is referred to as the compliance of the spring. In the case that x_d is constant and there are no external forces applied on to the system ($F_{ext} = 0$), the control law would be:

$$F = -\frac{\partial V(r)}{\partial r} - \frac{\partial R}{\partial \dot{r}} = -\frac{1}{C}(x - x_d) - b\dot{x} \quad (\text{E.4})$$

This method is called energy shaping because the elements from above that are chosen determine how the energy is exchanged. In general, energy shaping can be used when the system is back-drivable, has low friction and contains position measurements ((Stramigioli, 2017)).

In the case of a translational spring (as seen in the previous section), the potential energy of $V(r)$ is a positive semi-definite function with a minimum at $r = 0$, as seen in Figure E.4a. In the case of a rotational spring, where a 2π rotation is expected, things are a bit more complex. A desired behavior would be for there to be the same amount of potential energy when $\theta = 0$ and $\theta = 2\pi$. This would not be the case with E.4a. For that reason, the potential energy of the rotation spring has to be a closed cycle, as illustrated in Figure E.4b. That being said, for our case, that will not be an issue. For small angles around $\theta = 0$, the behaviour of both functions is very similar. The needle is expected to only perform small angles. With that in mind, equation E.2 was chosen for the potential energy of the spring, since it has a linear behaviour.

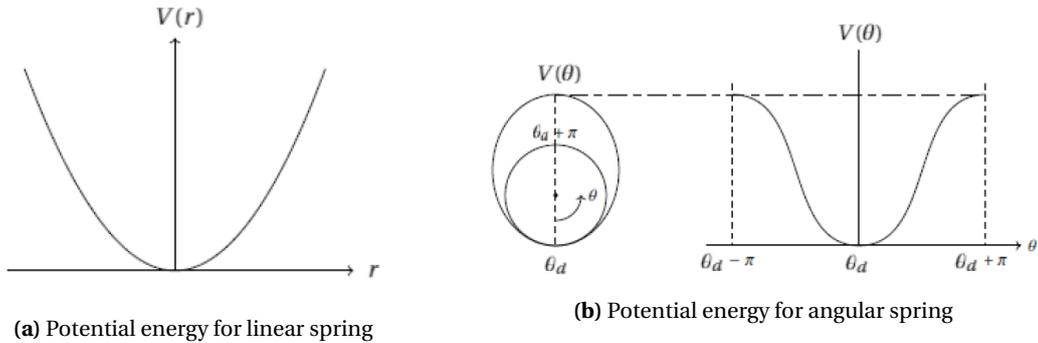


Figure E.4: Potential energy functions

F Detailed Measurement Results

F.1 Experiment 1

Execution number	X coordinate [mm]	Z coordinate [mm]
1	0.54623	0.29888
2	0.089653	0.15865
3	0.92509	0.29207
4	0.0061172	0.14529
5	-0.23464	0.26965
6	0.22871	0.26102
7	2.0673	0.62704
8	0.21547	0.22011
9	0.39243	0.02668
10	0.15098	0.18522

Table F.1: Test 1 - Point 1 - Difference of initial and final value

Execution number	X coordinate [mm]	Z coordinate [mm]
1	0.71257	0.98933
2	0.36922	0.23986
3	1.596	1.0237
4	0.48671	0.18354
5	0.06975	0.32285
6	0.62203	0.76113
7	2.5168	1.4967
8	-0.032368	0.55598
9	0.28667	0.42293
10	0.31507	0.55596

Table F.2: Test 1 - Point 2 - Difference of initial and final value

Execution number	X coordinate [mm]	Z coordinate [mm]
1	1.3032	0.18226
2	0.27648	0.10237
3	2.6102	1.0601
4	1.9178	-0.29258
5	0.26813	-1.4214
6	0.72774	1.5995
7	0.51441	0.66453
8	0.098717	1.5856
9	0.45096	0.40287
10	0.4379	0.96277

Table F.3: Test 1 - Point 3 - Difference of initial and final value

Execution number	X coordinate [mm]	Z coordinate [mm]
1	0.42526	0.05908
2	0.2494	0.7645
3	-0.6552	0.022775
4	0.23666	0.30098
5	1.044	0.19637
6	1.745	0.080591
7	0.2249	0.087663
8	-0.44839	-0.028231
9	-0.19458	-0.041354

Table F.4: Test 2 - Point 1 - Difference of initial and final value

Execution number	X coordinate [mm]	Z coordinate [mm]
1	1.3053	0.88287
2	0.95862	2.2287
3	-0.069733	0.70139
4	0.49121	1.3459
5	0.19291	-0.057055
6	1.9346	0.85322
7	1.0085	0.41332
8	-0.25994	-0.29347
9	0.56646	-0.060673

Table F.5: Test 2 - Point 2 - Difference of initial and final value

Execution number	X coordinate [mm]	Z coordinate [mm]
1	1.8716	1.1635
2	1.5697	2.9331
3	0.4296	-0.073064
4	0.4361	0.26982
5	0.03394	0.47504
6	2.0701	-0.27004
7	0.33428	0.89614
8	0.53137	2.0148
9	-0.50573	-1.7222

Table F.6: Test 2 - Point 3 - Difference of initial and final value

F.2 Experiment 2

Execution number	X coordinate [m]	Y coordinate [m]	Z coordinate [m]
1	0.5520	-0.1831	0.3850
2	0.5520	-0.1833	0.3849
3	0.5521	-0.1829	0.3849
4	0.5520	-0.1832	0.3849
5	0.5521	-0.1837	0.3849
6	0.5521	-0.1813	0.3850
7	0.5520	-0.1810	0.3852
8	0.5521	-0.1829	0.3850
9	0.5521	-0.1824	0.3850
10	0.5520	-0.1826	0.3849

Table F.7: Initial value for Tissue Deformation Tracking algorithm

Execution number	X coordinate [m]	Y coordinate [m]	Z coordinate [m]
1	0.5496	-0.1839	0.3727
2	0.5525	-0.1839	0.3826
3	0.5527	-0.1834	0.3816
4	0.5523	-0.1836	0.3823
5	0.5522	-0.1840	0.3812
6	0.5533	-0.1819	0.3833
7	0.5542	-0.1816	0.3830
8	0.5537	-0.1837	0.3756
9	0.5534	-0.1832	0.3826
10	0.5531	-0.1832	0.3807

Table F.8: Final value for Tissue Deformation Tracking algorithm

Execution number	X coordinate [m]	Y coordinate [m]	Z coordinate [m]
1	0.5513	-0.1829	0.3835
2	0.5508	-0.1824	0.3836
3	0.5512	-0.1827	0.3842
4	0.5511	-0.1830	0.3838
5	0.5508	-0.1827	0.3837
6	0.5512	-0.1827	0.3838
7	0.5513	-0.1828	0.3836
8	0.5509	-0.1831	0.3832
9	0.5509	-0.1825	0.3842
10	0.5508	-0.1826	0.3837

Table F.9: Initial value NDI target

Execution number	X coordinate [m]	Y coordinate [m]	Z coordinate [m]
1	0.5488	-0.1859	0.3748
2	0.5508	-0.1832	0.3823
3	0.5509	-0.1832	0.3826
4	0.5508	-0.1836	0.3819
5	0.5510	-0.1827	0.3827
6	0.5512	-0.1829	0.3833
7	0.5512	-0.1830	0.3832
8	0.5500	-0.1849	0.3786
9	0.5511	-0.1832	0.3830
10	0.5501	-0.1834	0.3814

Table F.10: Final value NDI target

Execution number	X coordinate [mm]	Y coordinate [mm]	Z coordinate [mm]
1	0.7718	-0.2587	1.5172
2	1.2576	-0.8321	1.3379
3	0.8419	-0.2225	0.6629
4	0.9515	-0.1975	1.1519
5	1.2814	-0.9962	1.1947
6	0.8927	1.3359	1.1520
7	0.7004	1.8174	1.6054
8	1.2090	0.1789	1.7063
9	1.1813	0.0698	0.7968
10	1.1912	-0.0162	1.2091

Table F.11: Difference of initial values

Execution number	X coordinate [mm]	Y coordinate [mm]	Z coordinate [mm]
1	0.8392	1.9756	-2.1414
2	1.7833	-0.6873	0.2955
3	1.8515	-0.1689	-0.9509
4	1.4966	-0.0040	0.3966
5	1.2725	-1.3661	-1.5213
6	2.0286	1.0851	0.0526
7	3.0890	1.3837	-0.1915
8	3.6864	1.1488	-3.0475
9	2.2303	-0.0154	-0.3599
10	2.9554	0.1943	-0.6965

Table F.12: Difference of final values

F.3 Experiment 3

X coordinate [mm]	Z coordinate [mm]
-0.0871	24.8800
-0.2581	25.3809
-0.2568	25.2147
-2.1497	24.0633
-2.7711	24.5348
0.4474	25.1862
1.1254	26.9394
-0.1767	25.0491
-0.9971	24.8160

Table F.13: Final Values - Test 1

X coordinate [mm]	Z coordinate [mm]
-2.6801	24.5079
2.1618	25.2697
-2.1213	24.7402
-5.4436	23.9977
3.1000	24.2106
0.7168	24.3787
-0.8358	24.9360
4.0052	24.8886
-5.5661	24.4805
-4.5869	24.6929

Table F.14: Final Values - Test 2**F.4 Experiment 4**

X coordinate [mm]	Z coordinate [mm]
1.5226	-104.2281
1.1840	-104.0766
1.2611	-104.1297
1.4009	-104.0905
1.4966	-104.1039
1.6008	-104.1890
1.3417	-104.0827
1.2223	-104.0781
1.3887	-104.0671
1.2962	-104.0692

Table F.15: Final Values NDI target - Test 1

X coordinate [mm]	Z coordinate [mm]
0.7614	-106.2813
-0.0667	-103.9288
1.9940	-105.4007
1.8614	-104.4888
2.4803	-104.7793
1.1957	-105.1865
0.7471	-105.0231
-0.2226	-104.6034
1.3845	-104.2983
0.7055	-104.4016

Table F.16: Final Values of Needle - Test 1

X coordinate [mm]	Z coordinate [mm]
1.2514	-95.2015
0.7209	-95.1988
0.8089	-95.0966
0.9435	-95.0830
0.5811	-95.2809
0.8700	-95.2188
1.1806	-95.1874
0.7756	-95.3274
0.9128	-95.2860
0.9900	-95.1932

Table F.17: Final Values NDI target - Test 2

X coordinate [mm]	Z coordinate [mm]
4.3735	-96.0268
0.4289	-99.7943
-1.8354	-94.8306
2.1169	-95.0239
3.1280	-92.4661
3.2828	-93.2519
2.3924	-97.2507
3.2441	-97.3497
2.5098	-96.2595
1.6122	-94.6491

Table F.18: Final Values of Needle - Test 2

F.5 Experiment 5

X coordinate [mm]	Y coordinate [mm]	Z coordinate [mm]
-9.0195	1.1994	-106.2265
-8.9409	1.0437	-106.2246
-9.6756	1.5924	-104.9341
-8.9660	0.9528	-107.1732
-9.2760	1.1664	-106.7307
-9.2062	1.6319	-105.7710
-9.3686	0.7739	-106.2437
-10.1563	1.5465	-102.9332
-9.2299	0.9285	-107.0074
-10.9731	4.3197	-98.3299

Table F.19: Final Values NDI target

X coordinate [mm]	Y coordinate [mm]	Z coordinate [mm]
-11.1164	-1.0142	-109.9630
-10.9547	-2.4852	-109.3396
-9.7116	-0.6867	-108.6505
-8.6209	2.1714	-112.0283
-8.3923	0.0144	-110.5654
-8.4063	1.8674	-110.8219
-8.5981	1.4001	-109.9458
-8.7824	0.4709	-105.2442
-8.9129	1.1448	-109.5592
-8.1377	4.8683	-100.1954

Table F.20: Final Values of Needle

Bibliography

- Momen Abayazid, Pedro Moreira, Navid Shahriari, Sachin Patil, Ron Alterovitz, and Sarthak Misra. Ultrasound-guided three-dimensional needle steering in biological tissue with curved surfaces. *Medical Engineering and Physics*, 2015. ISSN 18734030. doi: 10.1016/j.medengphy.2014.10.005.
- Purang Abolmaesumi, Septimiu E. Salcudean, Wen Hong Zhu, Mohammad Reza Sirouspour, and Simon P. DiMaio. Image-guided control of a robot for medical ultrasound. *IEEE Transactions on Robotics and Automation*, 18(1):11–23, 2 2002. ISSN 1042296X. doi: 10.1109/70.988970.
- Yeong Yn An, Sung Hun Kim, Bong Joo Kang, and Jae Hee Lee. Usefulness of magnetic resonance imaging-guided vacuum-assisted breast biopsy in Korean women: A pilot study. *World Journal of Surgical Oncology*, 2013. ISSN 14777819. doi: 10.1186/1477-7819-11-200.
- Nimmi Arora, Tari A. King, Lindsay M. Jacks, Michelle M. Stempel, Sujata Patil, Elizabeth Morris, and Monica Morrow. Impact of breast density on the presenting features of malignancy. In *Annals of Surgical Oncology*, 2010. doi: 10.1245/s10434-010-1237-3.
- Patrizia Baraldi, Alessandro Sarti, Claudio Lamberti, Alessandro Prandini, and Fiorella Sgallari. Evaluation of Differential Optical Flow Techniques on Synthesized Echo Images. *IEEE Transactions on Biomedical Engineering*, 43(3):259–272, 1996.
- Adrian Basarab, Hervé Liebgott, Fabrice Morestin, Andrej Lyshchik, Tatsuya Higashi, Ryo Asato, and Philippe Delachartre. A method for vector displacement estimation with ultrasound imaging and its application for thyroid nodular disease. *Medical Image Analysis*, 2008. ISSN 13618415. doi: 10.1016/j.media.2007.10.007.
- Wendie A. Berg, Jeffrey D. Blume, Jean B. Cormack, Ellen B. Mendelson, Daniel Lehrer, Marcela Böhm-Vélez, Etta D. Pisano, Roberta A. Jong, W. Phil Evans, Marilyn J. Morton, Mary C. Mahoney, Linda Hovanessian Larsen, Richard G. Barr, Dione M. Farria, Helga S. Marques, and Karan Boparai. Combined screening with ultrasound and mammography vs mammography alone in women at elevated risk of breast cancer. *JAMA - Journal of the American Medical Association*, 2008. ISSN 00987484. doi: 10.1001/jama.299.18.2151.
- L. N. Bohs, B. J. Geiman, M. E. Anderson, S. C. Gebhart, and G. E. Trahey. Speckle tracking for multi-dimensional flow estimation. *Ultrasonics*, 2000. ISSN 0041624X. doi: 10.1016/S0041-624X(99)00182-1.
- John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1986. ISSN 01628828. doi: 10.1109/TPAMI.1986.4767851.
- Yi Hong Chou, Chui Mei Tiu, Jiayu Chen, and Ruey Feng Chang. Automated full-field breast ultrasonography: The past and the present, 2007. ISSN 09296441.
- Yang Chunke, Kenji Terada, and Shunitiro Oe. Motion analysis of echocardiograph using optical flow method. In *IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems*, pages 672–677, Beijing, 1996. IEEE.
- Richard O. Duda and Peter E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, 1972. ISSN 15577317. doi: 10.1145/361237.361242.
- Gunnar Farneäck. Two-frame motion estimation based on polynomial expansion. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2003. ISSN 03029743. doi: 10.1007/3-540-45103-x{_}50.
- Barry H. Friemel, Laurence N. Bohs, and Gregg E. Trahey. Relative performance of two-dimensional speckle-tracking techniques: normalized correlation, non-normalized correl-

- ation and sum-absolute-difference. In *Proceedings of the IEEE Ultrasonics Symposium*, volume 2, pages 1481–1484, 1995.
- Leo Grady. Random walks for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006. ISSN 01628828. doi: 10.1109/TPAMI.2006.233.
- Neville Hogan. Impedance Control: An Approach to Manipulation. In *Proceedings of the American Control Conference*, 1984.
- J. Hong, T. Dohi, M. Hashizume, K. Konishi, and N. Hata. An ultrasound-driven needle-insertion robot for percutaneous cholecystostomy. *Physics in Medicine and Biology*, 2004. ISSN 00319155. doi: 10.1088/0031-9155/49/3/007.
- B. K.P. Horn and B. G. Schunck. Determining optical flow. *Computer vision*, 1981. ISSN 00043702. doi: 10.1016/0004-3702(81)90024-2.
- International Agency for Research on Cancer (IARC). Press Release N° 263. Technical report, World Health Organization, 2018. URL <https://www.who.int/cancer/PRGlobocanFinal.pdf>.
- Athanasios Karamalis, Wolfgang Wein, Tassilo Klein, and Nassir Navab. Ultrasound confidence maps using random walks. *Medical Image Analysis*, 2012. ISSN 13618415. doi: 10.1016/j.media.2012.07.005.
- Ebrahim Karami, Mohamed S. Shehata, and Andrew Smith. Tracking of the internal jugular vein in ultrasound images using optical flow. In *Canadian Conference on Electrical and Computer Engineering*, 2017. ISBN 9781509055388. doi: 10.1109/CCECE.2017.7946589.
- Mert Kaya, Enes Senel, Awais Ahmad, Orcun Orhan, and Ozkan Bebek. Real-time needle tip localization in 2D ultrasound images for robotic biopsies. In *Proceedings of the 17th International Conference on Advanced Robotics, ICAR 2015*, 2015. ISBN 9781467375092. doi: 10.1109/ICAR.2015.7251432.
- Kevin M. Kelly and Gary A. Richwald. Automated Whole-Breast Ultrasound: Advancing the Performance of Breast Cancer Screening. *Seminars in Ultrasound, CT and MRI*, 32(4):273–280, 8 2011. ISSN 08872171. doi: 10.1053/j.sult.2011.02.004.
- Alexandre Krupa, Gabor Fichtinger, and Gregory D. Hager. Real-time motion stabilization with b-mode ultrasound using image speckle information and visual servoing. *International Journal of Robotics Research*, 2009. ISSN 02783649. doi: 10.1177/0278364909104066.
- Marla R. Lander and László Tabár. Automated 3-D breast ultrasound as a promising adjunctive screening tool for examining dense breast tissue. *Seminars in Roentgenology*, 2011. ISSN 0037198X. doi: 10.1053/j.ro.2011.06.003.
- S.O.H. De Looijer. Intrinsicly Passive Control for Ultrasound-guided Robotic Scanning of the Breast. 2018.
- Bruce D. Lucas and Takeo Kanade. Iterative Image Registration Technique with an Application to Stereo Vision. volume 2, pages 674–679, 1981.
- Luis Maggi, Guillermo Cortela, Marco Antonio Von Krüger, and Carlos Negreira. Ultrasonic Attenuation and Speed in Phantoms Made of PVCP and Evaluation of Acoustic and Thermal Properties of Ultrasonic Phantoms Made of polyvinyl chloride- plastisol (PVCP). In *IWBIO 2013*, 2013.
- Giuseppe Megali, Oliver Tonet, Cesare Stefanini, Mauro Boccadoro, Vassilios Pappaspyropoulos, Licinio Angelini, and Paolo Dario. A computer-assisted robotic ultrasound-guided biopsy system for video-assisted surgery. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2001. ISBN 3540426973. doi: 10.1007/3-540-45468-3{_}41.
- Giulio Mozzi. Discorso matematico sopra il rotamento momentaneo dei corpi. Technical report, 1763.

- Zipi Neubach and Moshe Shoham. Ultrasound-guided robot for flexible needle steering. *IEEE Transactions on Biomedical Engineering*, 57(4):799–805, 2010. ISSN 00189294. doi: 10.1109/TBME.2009.2030169.
- Yoshihiro Noguchit, Jun Furukawatt, and Hitoshi Kiyatt. A Fast Full Search Block Matching Algorithm for MPEG-4 Video. pages 61–65, Kobe, 1999.
- Claire Pellot-Barakat, Frédérique Frouin, Michael F. Insana, and Alain Herment. Ultrasound Elastography Based on Multiscale Estimations of Regularized Displacement Fields. *IEEE Transactions on Medical Imaging*, 2004. ISSN 02780062. doi: 10.1109/TMI.2003.822825.
- Jie Ren, Tian Cao, Chaowei Tan, and Dong C. Liu. SAD based sensor-less freehand 3D ultrasound reconstruction with adaptive curve correction. In *2010 4th International Conference on Bioinformatics and Biomedical Engineering, iCBBE 2010*, 2010. ISBN 9781424447138. doi: 10.1109/ICBBE.2010.5516742.
- Olinde Rodrigues. De l’attraction des sphéroïdes. Technical report, 1816.
- Gloria M. Spirou, Alexander A. Oraevsky, I. Alex Vitkin, and William M. Whelan. Optical and acoustic properties at 1064 nm of polyvinyl chloride-plastisol for use as a tissue phantom in biomedical optoacoustics. *Physics in Medicine and Biology*, 2005. ISSN 00319155. doi: 10.1088/0031-9155/50/14/N01.
- Stefano Stramigioli. Creating artificial damping by means of damping injection. In *American Society of Mechanical Engineers, Dynamic Systems and Control Division (Publication) DSC*, 1996.
- Stefano Stramigioli. Lecture notes in the course "Modern Robotics" of the University of Twente, 2017.
- Michael Sühling, Muthuvel Arigovindan, Christian Jansen, Patrick Hunziker, and Michael Unser. Myocardial motion analysis from B-mode echocardiograms. *IEEE Transactions on Image Processing*, 2005. ISSN 10577149. doi: 10.1109/TIP.2004.838709.
- Audrey Thouvenot, Tamie Poepping, Terry M. Peters, and Elvis C. S. Chen. Characterization of Various Tissue Mimicking Materials for Medical Ultrasound Imaging. In *SPIE Medical Imaging*, San Diego, California, United States, 2016. doi: <https://doi.org/10.1117/12.2218160>.
- U.S. Breast Cancer Statistics. https://www.breastcancer.org/symptoms/understand_bc/statistics, 2019. URL https://www.breastcancer.org/symptoms/understand_bc/statistics.
- Hong Yan Wang, Yu Xin Jiang, Qing Li Zhu, Jing Zhang, Qing Dai, He Liu, Xing Jian Lai, and Qiang Sun. Differentiation of benign and malignant breast lesions: A comparison between automatically generated breast volume scans and handheld ultrasound examinations. *European Journal of Radiology*, 2012. ISSN 0720048X. doi: 10.1016/j.ejrad.2012.01.034.
- Marcel Welleweerd. MURAB: Final Design Robotic Head. 2018.
- Agata Wijata, Żaneta Ranzosz, Marta Galińska, Jan Juszczak, and Czajkowska Joanna. Detection and Tracking of the Biopsy Needle Using Ultrasound Images. In *Innovations in Biomedical Engineering*, chapter 4, pages 33–41. Springer, 2018. ISBN 978-3-319-70063-2. doi: https://doi.org/10.1007/978-3-319-70063-2_{4}.