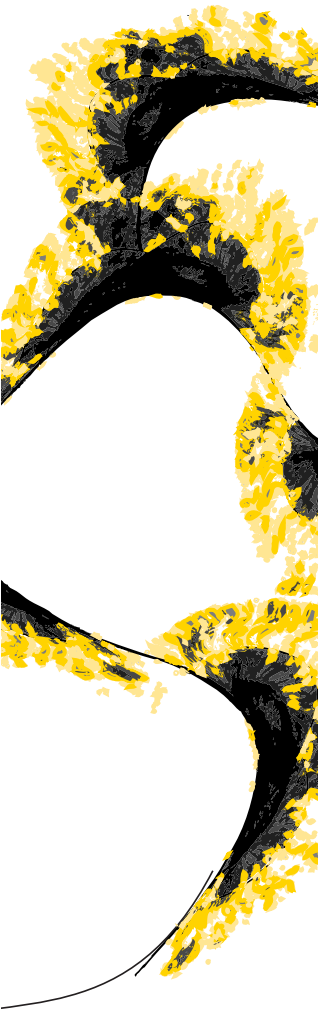


RAM ROBOTICS AND MECHATRONICS

DESIGN OF A USER-FRIENDLY OPTICAL CONTROLLER FOR THE SUNRAM 7 WITH SAFETY FEATURES TO ASSIST RADIOLOGISTS IN MRI-GUIDED BREAST BIOPSY

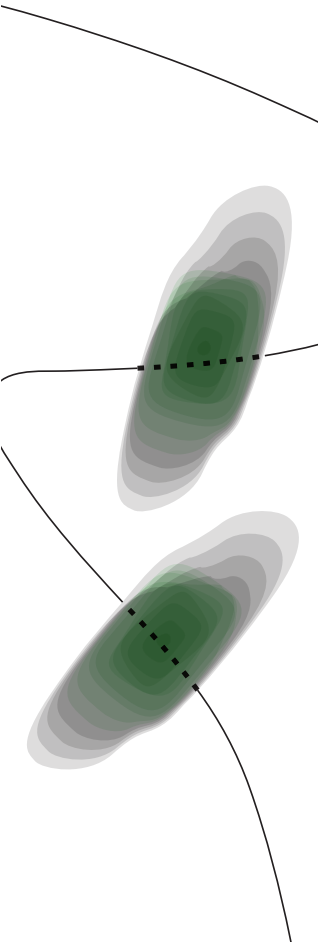
D.N. (Desiree) Pietersma

BSC ASSIGNMENT

Committee:
dr. F.J. Siepel
dr. V. Groenhuis, MSc
dr. ir. W.M. Brink

July, 2023

033RaM2023
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands



UNIVERSITY OF TWENTE. | TECHMED CENTRE UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Summary

Magnetic Resonance Imaging (MRI)-guided biopsies play a crucial role in the diagnosis and treatment of breast cancer, particularly for detecting abnormalities that may be missed by other imaging techniques. However, the traditional MRI-guided biopsy procedure is time consuming and complex. To address these challenges, the Sunram 7 was designed. The Sunram 7 is a biopsy robot that can be placed inside an MRI. Currently, the Sunram 7 is controlled using three joysticks, which can make targeting the robot both complex and unintuitive.

This research paper focuses on the design of a safe and user-friendly optical tracking system for MRI-guided breast biopsies with the Sunram 7.

The methodology includes the design and implementation of an optical tracking system in MATLAB for the biopsy robot. The optical tracking system utilizes infrared-based tracking and integrates certain safety features and additional features to enhance the overall usability and the ease of use of the controller.

To assess the performance of the optical controller, latency and accuracy experiments were conducted. The user-experience was tested with the System Usability Scale (SUS).

The latency measurement resulted in an average value of 352.5±11.2ms, indicating acceptable performance without significant degradation. User feedback indicated a positive overall user experience, with a SUS score of 70.4, indicating good usability, intuitiveness, and efficiency. The accuracy measurements revealed an average Root Mean Square (RMS) error of 8.55±2.27mm, indicating a need for improvement to effectively target smaller lesions.

In conclusion, the designed optical tracking system offers a safe and user-friendly approach to controlling the Sunram 7 for MRI-guided breast biopsies. While improvements are needed to enhance accuracy, the system shows promise for future applications. Recommendations are provided to address the identified code issues and suggest areas for further experimentation, such as conducting a more comprehensive user evaluation and accuracy tests in free air and in tissue.

Overall, this research contributes to the advancement of optical tracking systems in image guided procedures and provides insight for future developments in biopsy robot control.

Contents

1 Introduction

Worldwide breast cancer is one of the most commonly diagnosed types of cancer and has one of the highest mortality rates among cancers. In 2020, 2.3 million people were diagnosed with breast cancer and it accounted for 685 thousand deaths (Arnold et al., 2022). Mortality of breast cancer can be reduced by early diagnosis and treatment. Different screening methods can aid in the detection of breast cancer. The standard for screening is mammography and ultrasound, although sometimes a different technique is necessary (Zhang et al., 2021).

Magnetic resonance imaging (MRI) can detect abnormalities in breast tissue that other screening methods are unable to detect. MRI uses a strong magnetic field, making ferromagnetic materials a hazard near an MRI. To determine if a lesion is malignant, a biopsy can be taken from the tissue with an MRI guided biopsy. This procedure is time consuming, complex and requires well-trained and experienced radiologists (Papalouka et al., 2018).

Taking the previous factors into consideration, the Sunram 7 biopsy robot was designed. It is designed with no metal except for the non-ferromagnetic titanium needle and works on air pressure to drive the motors. Currently, the Sunram 7 is controlled via 3 joysticks, which makes targeting the robot to a specific spot complex and unintuitive.

In this thesis, research will be done to find the optimal way to control the Sunram 7 robot to perform biopsies in a breast (phantom) in a way that could potentially be used in hospitals.

1.1 Sunram 7

The Sunram 7 is the seventh version of the MR safe breast biopsy robot, previously known as the Stormram. It is fully made of non-metallic, non-magnetic and non-conductive materials such as polycarbonate (PC) and polylactic acid (PLA). Only the titanium biopsy needle is metallic. It is originally from an MR conditional hand-held biopsy gun made by Invivo Corp. The robot is powered by five pneumatic stepper motors with a step size of 0.462mm. The motors drive four revolute joints for the positioning of the robot and one prismatic joint for the forward motion of the needle. The five joints give the robot five degrees of freedom. The first full prototype of the Sunram 7 can be seen in Figure 1.1, with the five joints. This prototype has an accuracy of 2-3mm in free air and an approximate workspace of 200x100x100mm.

Figure 1.1: *Sunram 7 first full prototype*

The Sunram 7 is developed by Vincent Groenhuis at Robotics and Mechatronics (RaM), University of Twente, Enschede, The Netherlands.

1.2 State of the Art

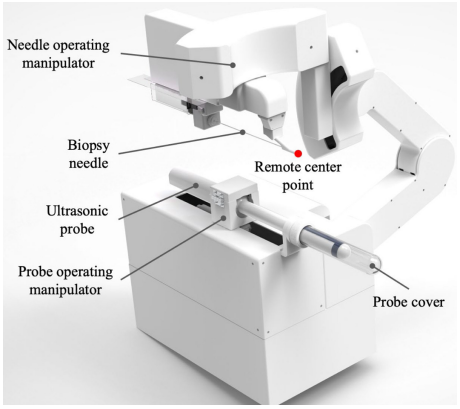
Current state of the art for image guided biopsy robots are mainly for breast, prostate, lung and liver biopsies. There also exist biopsy robots that can be used on multiple organs. The next section briefly discusses some of these robots.

A prostate biopsy robot can be seen in Figure 1.2a. It has a transrectal ultrasound (TRUS) probe and a biopsy needle. It contains two image guidance modes, an image fusion of MRI and TRUS and image independent TRUS (Wang et al., 2023, 2021).

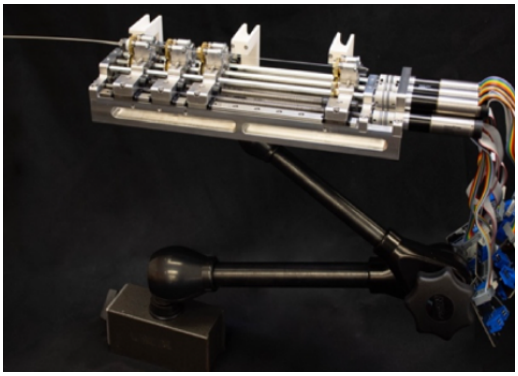
For a lung biopsy robot, a flexible aspiration needle is employed. This can be seen in Figure 1.2b. This needle is guided through the patient's mouth to take a biopsy. Under computed tomograpy (CT) guidance can a biopsy be taken (Amack et al., 2019).

In Figure 1.2c, a liver biopsy robot controlled by four pneumatic cylinders labeled A, B, C, and D is shown. This robot utilizes MRI scans for guidance (Minchev et al., 2017).

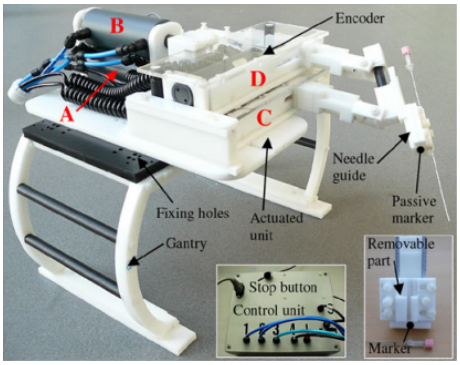
A robot that can be used for various biopsies in the abdominal area and the shoulders, is shown in Figure 1.2d. The robot is mounted on the body and can be used inside an MRI or a CT. The needle is moved by two scissor mechanisms (Patel et al., 2018).



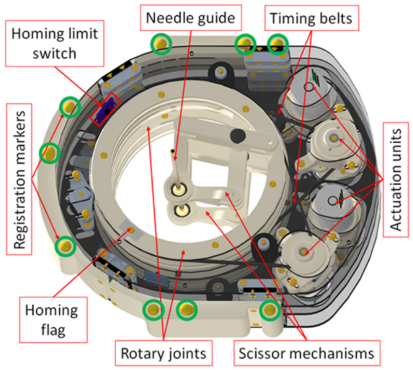
(a) *Ultrasound-guided prostate biopsy robot (Wang et al., 2023)*



(b) *Transbronchial lung biopsy robot (Amack et al., 2019)*



(c) *Liver biopsy robot (cylinders A,B,C and D) (Minchev et al., 2017)*



(d) *Body-mounted biopsy robot (Patel et al., 2018)*

Figure 1.2: *Biopsy robots for different organs*

1.3 Techniques

Teleoperation or telesurgery is the term used for controlling a robot remotely for surgery. The operator holds a sensor/device that translates the inputs from the operator to movement of the robot. On monitors the real-time movements or a simulation of the robot can be viewed (Leal Ghezzi and Campos Corleta, 2016; Xia and Lu, 2021).

In the case of the Sunram 7, with real-time MRI scanning, the robot can be controlled in the MRI. There can also be immediate feedback to see if the targeted spot for the biopsy was hit. For telesurgery there are multiple techniques, such as haptic feedback devices, electromagnetic tracking and optical tracking (Patel et al., 2022). To make the controlling of the Sunram 7 intuitive and simple to use, different controllers with these techniques will be considered. When considering these types of controllers, the accuracy, user-friendliness and workspace are taken into account.

1.3.1 Technique 1: Haptic Feedback Device

A recent technique for teleoperation is the implementation of haptics into the controller. Haptics create a sense of touch for the surgeon. Instead of only visual feedback on the monitors, the surgeon feels how much pressure is exerted on the tissue of the patient. There are multiple types of haptic feedback but the most common is force feedback and will be the only focus in this study, since most operators are made with only force-feedback. Research has shown that the added effect of force-feedback to the controller improves the performance of surgeons, with shorter procedure times and less excessive forces on tissue (Patel et al., 2022).

For this project, the Omega.6 and the Omega.7 Haptic Device are available. The Omega.6 has a pen-shaped end-effector, shown in Figure 1.3a. The pen-shaped end-effector provides translational as rotational movements. The Omega.7 has the same specifications as the Omega.6 and only differs in the end-effector. The Omega.7 has a thicker end-effector and a gripper extension, shown in Figure 1.3b. Both haptic devices have a translational resolution of <0.01mm (Dimension, 2022). The translational workspace is ⌀ 160x110mm and the rotational workspace is 240x140x320 degrees.

(a) *Omega.6 haptic device* ***(b)*** *Omega.7 haptic device*

Figure 1.3: *Force Dimension Haptic Devices (Dimension, 2022)*

1.3.2 Technique 2: Electromagnetic Tracking

An electromagnetic (EM) tracker consists of an EM field emitter and an EM receiver. The receiver can in the case of teleoperation be in the shape of a pen or a medical instrument. EM sensors work when their line of sight is obstructed, but with increasing distance between the EM emitter and receiver, the accuracy decreases (Cormier and Bouslimani, 2021; Qiu et al., 2019). EM tracking is also generally less accurate than other methods. There is a concern with EM tracking that other medical devices, such as an MRI or an US scanner will influence the EM field and decrease the accuracy of the tracking (Cormier and Bouslimani, 2021).

For this project, the Aurora NDI EM tracker tabletop is available, shown in Figure 1.4. From the technical sheet of the Aurora tabletop it shows that the translational resolution for a 5 degrees

of freedom (DoF) sensor is 1.2mm. The workspace is a 420x600mm ellipse and 600mm high (NDI, 2023).

(a) *NDI Aurora needle* ***(b)*** *NDI Aurora EM field generator*

Figure 1.4: *NDI EM tracker (NDI, 2023)*

1.3.3 Technique 3: Optical Tracking

There are two main types of optical tracking, infrared-based tracking and image-based tracking. Both use markers that are attached to the medical device which can be easily localized by the camera. The markers for infrared tracking are retroreflective (Qiu et al., 2019). Retroreflection is the reflection of light back towards its source. It is most commonly used during surgery to track where the surgical instrument is in the patient. But should also work outside of the body (Xu et al., 2023). For this thesis, multiple IR stereocameras are available. The Visualeyez VZ4000, the XBOX kinect v2 and the KAYETON stereo webcam. The accuracy of the cameras depend strongly on the manufacturer (Xu et al., 2023; PTI, 2014; Edmunds et al., 2016; KAYETON, 2018). The three different cameras can be seen in Figure 1.5.

(a) *VZ4000 (PTI, 2014)* ***(b)*** *Kinect v2 (Edmunds et al., 2016)* ***(c)*** *KAYETON (KAYETON, 2018)*

Figure 1.5: *Cameras*

More high end optical tracking systems such as the Visualeyez VZ4000 can reach a translational accuracy of <0.5mm (PTI, 2014). At the low end of the spectrum is the kinect v2 camera with an accuracy of 1.1mm (Edmunds et al., 2016). The exact accuracy of the KAYETON webcam cannot be found in literature, but it is a higher end camera than the kinect v2 so should have a higher accuracy. The exact workspace size of an optical tracking system is hard to determine since the non-linear nature of a camera is complex. For the available systems it can however be said that the workspace will be large enough for the Sunram 7, since all cameras have a depth of at least 0.5m and at least Field Of View (FOV) angles of 45x45 degrees (PTI, 2014; Edmunds et al., 2016; KAYETON, 2018).

1.3.4 Consideration

For the Sunram 7 the objective is to find an optimal controller so that it could potentially be used in hospitals. It should further be safe, simple to use, intuitive and have an accuracy better than the robot.

One approach to ensure intuitive usability is by shaping the controller like a pen, to mimic a needle. This design choice aims to create a more natural experience for the surgeon, enabling them to use it effectively without extensive training.

For the pen-shaped controller, the Omega.7 is impractical. The surgeon will have more precision control over a thinner controller compared to the thick and heavy cylinder of the Omega.7. All the other controllers have a needle-like device.

In most surgical robots, the haptic feedback system would be the best option. This is due to its high accuracy and ability to provide feedback when tissue is touched. For the Sunram 7 this aspect of haptic feedback is less relevant since the robot is put into the position in front of the targeted spot and the needle gun then shoots the needle towards the target to take a biopsy. The Sunram 7 also has no force feedback device on the needle, making accurate force feedback unattainable. Additionally, the high accuracy of the haptic devices of 0.01 mm is not necessary for the robots intended purpose. The robot itself has an accuracy of 2-3 mm. The unnecessary accuracy will only increase the price of the controller.

For the use of the EM tracker there is a concern that the EM field will be distorted due the magnetic field of the MRI. Since the purpose of the robot is to be teleoperated by a surgeon in a room next to the MRI. Furthermore, the accuracy is the lowest of the available options and the EM tracker is still quite expensive.

For this project optical tracking will be used, for the reason that the optical tracking is less expensive than the Haptic Feedback System and it is more accurate than the EM tracking. Optical tracking systems also have a larger workspace, so there is no concern that the full workspace of the robot can be used.

From the available optic devices the KAYETON webcam will be used, this camera has better accuracy than the kinect and is smaller than the VZ4000. Furthermore, this camera is compact, relatively cheap compared to the haptic devices and the EM device and easy to install which makes the implementation of the controller effortless. The instrument that will be used by the controller can also be easily changed depending on the preferences of the user.

1.4 Research Question and Hypothesis

Given the consideration above the following research question with subquestions was drafted:

How can the Sunram 7 be controlled by an optical controller in an optimal way that is safe, user-friendly and accurate in MRI-guided breast biopsies?

- What are the key aspects necessary to ensure safe use?
- What are the key aspects necessary to ensure user-friendliness?
- What are the accuracy and latency of the optical controller?

Based on the previous research questions the following hypothesis was formulated: The use of an optical controller to control the Sunram 7 during MRI guided-breast biopsies will be safe, user-friendly and accurate.

Implementing safety aspects such as starting the needle at a safe distance from the patient and ensuring that during a biopsy the needle only moves in the intended direction will ensure the safe use of the Sunram 7. To ensure user-friendliness, a controller with a pen-shaped end-effector and intuitive controls is essential. A clear overview of the movement of the robot on monitors will also be necessary. To measure the accuracy of the controller experiments will be performed that compare the intended location of needle of the robot and the actual location. Latency can be measured by timing with a camera, how fast the output is given to the robot.

2 Methods

To answer the research question and subquestions from Chapter 1, a methodology has been written. The first part of the methodology covers the implementation of the MATLAB code to control the Sunram 7 with optical tracking. The second part of the methodology presents the experiments that were conducted with the developed controller.

2.1 Method: Code

Controlling the Sunram 7 with an optical controller is done with infrared(IR)-based tracking with IR markers instead of image-based tracking. Image-based tracking relies on consistent lighting, while IR tracking is less affected by varying lighting levels, making it more reliable and robust. Additionally, IR tracking requires less computational power, which provides an advantage for real-time applications like tele-operation. The higher computational power and complex algorithms required for image-based tracking are due to tasks such as pattern recognition, feature extraction and matching. On the other hand, IR tracking often involves simpler algorithms as the IR markers used are easy to detect, due to their brightness relative to the surroundings on the camera footage (Ren et al., 2018; Xu et al., 2023).

The available IR markers are highly retroreflective spheres for infrared light and can be attached to any instrument. To be able to control the robot with this instrument with the markers, a program was designed which tracks these markers in space with the chosen KAYETON stereocamera.

When the tracking of the markers has been achieved, an app was designed that makes it clear how to operate the controller and perform biopsies. In the app, the coordinates of the markers are transformed to joint rotations of the robot so that the point of the needle is at the targeted position of the controller. The design of this app is discussed in Chapter 3.

2.1.1 Camera Calibration

The KAYETON stereocamera has to be calibrated before use. A stereocamera was used since this has 2 cameras. Depth can be calculated from the difference in angle that an object has on the two cameras. The calibration of the camera is done in MATLAB with StereoCameraCalibrator. For the calibration multiple pictures were taken of a checkerboard pattern with differing distances and angles to the camera, the used checkerboard can be seen in Figure 2.1. Since there is an unequal amount of squares on both sides, the x and y direction can be distinguished. The program then calculated the distortion that both cameras have and also the distance between the cameras. The program also determines a conversion matrix that converts the pixel positions of the markers to XYZ coordinates in space.

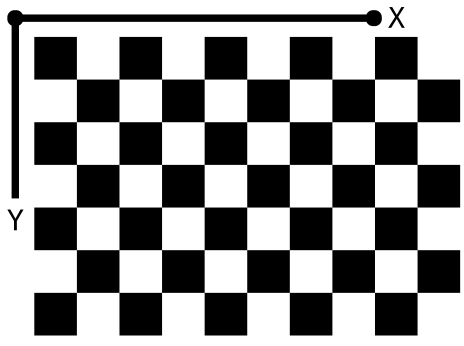


Figure 2.1: *Checkerboard pattern for camera calibration*

2.1.2 Camera Types

The marker tracking with the KAYETON webcam is happening once per trigger of the cameras. The time duration of the tracking is therefore important to be kept short. The cameras film in 30Hz and both have an IR LED positioned next to their lens. The KAYETON webcam provides two options to use, one with 1280x480 pixels and 2560x960 pixels. Since the webcam has two cameras, each camera provides either 640x480 or 1280x960 pixels.

The higher pixel option would have better accuracy but would possibly take longer to run the code. It was determined that with the the higher pixel option for the camera the duration of the marker tracking was too long and caused additional delay. The increase in accuracy was deemed less important of the higher pixel variant than the shorter time duration of the lower pixel variant. Therefore, the 1280x480 pixels option for the KAYETON webcam was chosen.

2.1.3 Marker Tracking

The markers were tracked based on their ability to reflect infrared light more than the surroundings. From the pixel image that the KAYETON camera provided the positions of the markers had to be found. To find the pixel positions of the markers, multiple methods could be used. Three methods were compared during testing.

The first method was with Blob Analysis, a function in MATLAB. In Figure 2.2 can the detected positions of the markers be seen and the pseudocode of this method can be seen in Algorithm 1. To use this function the image had to be converted to a binary image. The luminance threshold to binarize was adjusted depending on the environment until only the markers remained visible. Since this value varied depending on the time of day, regular checks were necessary. The images were then separated into the images of the left and the right camera.

In the binary image the function detected connected pixels called blobs. The minimum and maximum amount of connected pixels could be specified. For the used markers, the minimum was set at 100 pixels and the maximum at 1500 pixels. The function then gave out the centroid (geometric centre) position of the markers in pixel values. Since the pixelvalues from the two camera were all in one matrix, these had to be separated to find the markers on the right camera and the left camera. Per camera the maximum number of blobs that could be found was set at two, for the two markers.

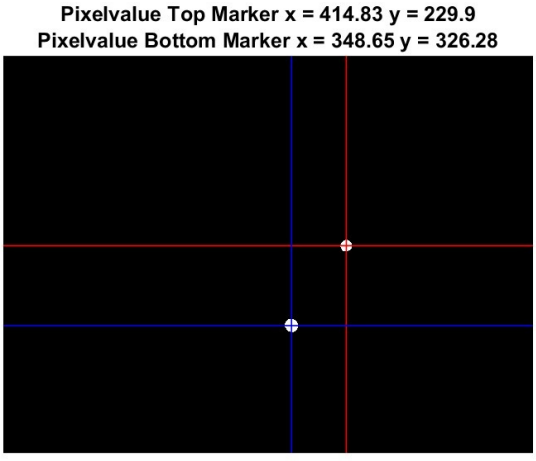


Figure 2.2: *Marker detection with method 1*

Algorithm 1 Pseudocode: Method 1 Marker Detection

```
Image = CameraTrigger()
Image = Binarize(Image, LuminanceThreshold)
[ImageLeft, ImageRight] = SeperateImage(Image)

BlobFunction = BlobAnalysis('MinBlobArea', 100, 'MaxBlobArea', 1500, 'MaxCount', 2)
PositionLeftMarkers = BlobFunction(ImageLeft)
PositionRightMarkers = BlobFunction(ImageRight)
```

The second method first binarized the frame of the cameras, in the same manner as the first method. The detected marker positions can be seen in Figure 2.3 and the algorithm used for this method can be seen in Algorithm 2. This algorithm was used for both the left and the right camera separately. From the resulting black and white frame, the sum of all columns from the camera was taken. Afterwards, the findpeaks function was used to identify the x positions of the marker centres. Since the column with the highest value was supposedly the centre of the marker. Similarly, the sum of all rows was calculated, and the findpeaks function was used to determine the y-positions of the centres. A minimum peak prominence was set at 8, so the clearest and most prominent peaks were taken from the data. With this method it is unclear which column peak value belonged to which row peak value. Therefore, additional checks had to be implemented. An x and y position were combined and checked. If their pixel position had a value of one, it indicated a valid marker position. With these checks implemented, the centre positions of the markers were identified successfully.

A problem with this method was that if the markers were in the same column or row, only one point would be detected. To resolve this issue, a function was used for cases where only one marker was detected. Algorithm 3 shows this function. In this function a check is performed to see if markers overlap. This is done by checking the size of the x or y positions. If there was one peak detected, then there was overlap. The centre was then copied and made a new peak value, since the centre positions of the markers were almost the same.

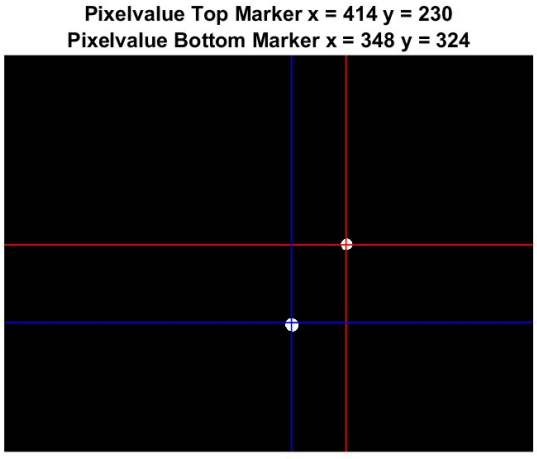


Figure 2.3: *Marker detection with method 2*

Algorithm 2 Pseudocode: Method 2 Marker Detection

```
SumColumns = sum(Image, Columns)
SumRows = sum(Image, Rows)
XPositions = FindPeaks(SumColumns. 'MinPeakProminence', 8)
YPositions = FindPeaks(SumRows, 'MinPeakProminence', 8)

if XPositions(1), YPositions(1)) in Image == 1 then
  PositionMarkers = [XPositions(1), YPositions(1); XPositions(2), YPositions(2)]
else if (XPositions(1), YPositions(2)) in Image == 1 then
  PositionMarkers = [XPositions(1), YPositions(2); XPositions(1), YPositions(2)]
else
  Warning('Markers not found')
end if
```

Algorithm 3 Pseudocode: Marker overlap function

```
if size(XPositions) == 1 then
  XPositions = [XPositions, XPositions]
else if size(YPositions) == 1 then
  YPositions = [YPositions, YPositions]
end if
```

The third marker detection method can be seen in Algorithm 4. This algorithm first converted the image to a grayscale. A mask was created with the binarize function with a certain threshold, to just contain the markers. In the grayscale image all the pixels outside the mask were set to 0. The markers remained in the grayscale image with differing values of their illumination. The modified grayscale image with the mask was then used as input for the FindPoints function. The FindPoints function calculated the average column or row value of the markers. The first time this function was used, the average row between the two markers was calculated. The image of a camera was then split into two frames along the average row. A bottom and a top frame was obtained. In each frame resided now one marker. The FindPoints function could then be used to find the average marker positions along the columns and rows. In Figure 2.4 can the detected positions of the markers be seen, the gray marker consists of different gray values.

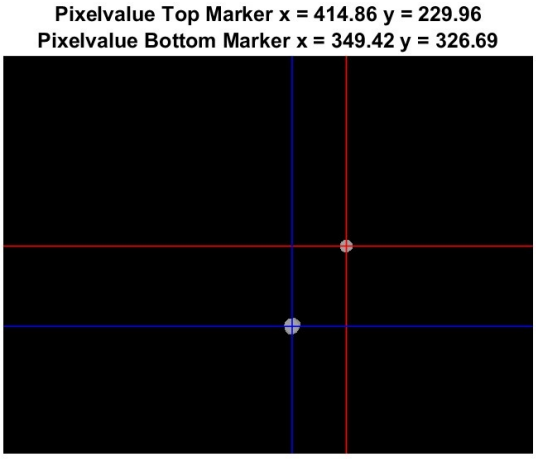


Figure 2.4: *Marker detection with method 3*

Algorithm 4 Pseudocode: Method 3 Marker Detection

```
Image = CameraTrigger()
GrayScaleImage = Grayscale(Image)
MaskImage = Binarize(Image, LuminanceThreshold)
Mask = MaskImage == 1
GrayscaleImage(not Mask) == 0
AverageY = FindPoints(GrayscaleImage, Rows)
[ImageTop, ImageBottom] = split(Grayscale, AverageY)
MarkerPositionTop = FindPoints(ImageTop, Columns and Rows)
MarkerPositionBottom = FindPoints(ImageBottom, Columns and Rows)
```

The FindPoints function calculated the average column or row value of the markers. This represents the pixel value for the x or y position. First the columns or rows were summed, resulting in an array. Next, an index array was created, ranging from one to the length of the array. Each value in the array was multiplied by its corresponding element in the index array. The resulting array was then summed, referred to as the total. To obtain the average column or row index representing the pixel value for the x or y position of the markers, the total value was divided by the sum of the original grayscale frame with the mask. The pseudocode for this function can be seen in Algorithm 5.

Algorithm 5 Pseudocode: FindPoints function

```
SumArray = sum(Image, Columns or Rows)
IndexArray = 1 : length(SumArray)
MultipliedArray = IndexArray .* SumArray;
Total = sum(MultipliedArray);
AveragePosition = Total/sum(SumArray);
```

When one of the previously described methods was used, the pixel values of the two markers on both cameras became known. For the marker detection in the app, the first method was chosen due to its shorter duration and higher accuracy. From Figure 2.2, 2.3 and 2.4 it can be seen that the first and third method barely differ from one another in accuracy, only the third method had a longer duration. The second method does not seem to grab the middle of the marker and can only be a full column or row number.

To utilize the markers effectively, it was necessary to distinguish between the two markers. The top marker would be used as the tip of the needle. This differentiation was made by examining the pixel values and determining which marker had a lower value thereby identifying it as the top marker.

2.1.4 Pixel Conversion to XYZ Coordinates

The pixel positions of the markers were converted to XYZ coordinates by using triangulation. The difference in the angle of the two markers on both cameras could be used to determine the depth. Before using triangulation, the points were first undistorted with the parameters obtained during the calibration of the cameras. With the XYZ coordinates known of the markers the azimuth angle between x and y was calculated and the elevation angle between y and z.

The XYZ coordinates were then known in the frame of the camera. With an H-matrix these coordinates could be converted from the frame of the camera to the frame of the robot. The two frames can be seen in Figure 2.5. The frame of the robot is on the base of the robot. The frame of the camera is 380mm above the frame of the robot, 127mm in front of the robot and 30mm to the right of the robot, these values are the offset.

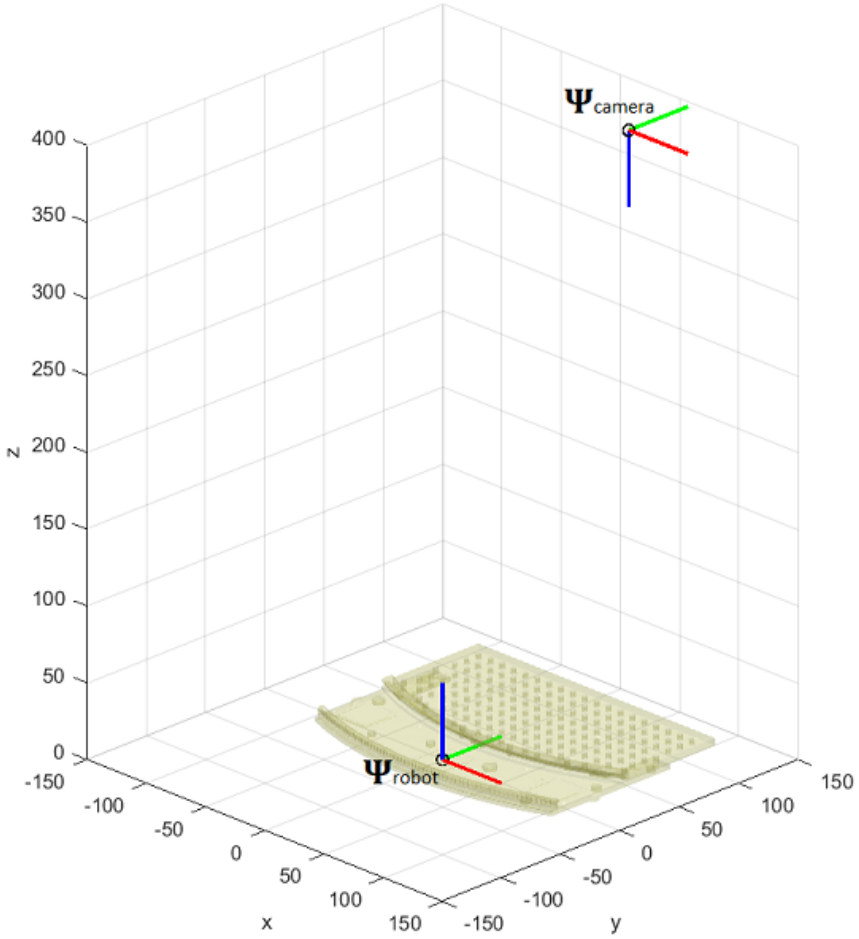


Figure 2.5: *Positions of the frame of the camera and frame of the robot*

The used H-matrix from camera to robot can be seen in the following Equation 2.1:

$$H_c^r = \begin{bmatrix} 0.25 & 0 & 0 & 30/4 \\ 0 & 0.25 & 0 & 127/4 \\ 0 & 0 & -0.25 & 380/4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.1}$$

The x, y and z coordinate are multiplied with 0.25. This was done to make the the controller work 4:1. So 4mm movement of the controller was equal to 1mm of movement of the robot. This gave the user more precise control over the Sunram 7 with the controller. The last column values are the offset between the robot and the camera in Equation 2.1, these values also have to be multiplied with 0.25. The z-axis is flipped when going from camera to robot frame. With a Ray function and inverse kinematics function from the inventor of the Sunram 7, Vincent Groenhuis, the coordinates in the camera frame were converted with the H-matrix to the robot frame and then to the joint vectors of the robot.

2.1.5 Filters

A user will always have small involuntary movements of the hand. To bypass these small movements of the hand when holding the instrument stationary, a threshold was implemented. There should at least be a two pixel difference with the current pixel position, to calculate a new joint vector.

To avoid these involuntary movements during the movement of the robot a moving average could be used or a weighted average of the pixel position.

From the different filters tested, the moving average with the threshold was used in the final version. The moving average ensured that during movement, the movement was not jerky, and the threshold ensured that when the controller was held in a position, it stayed in that position and did not jerk around due to the small involuntary movements of the hand. A weighted average was not used since the function used to calculate the coordinates of the markers used a binary image, a weighted average assigns importance to certain pixels. When the image is converted to binary there is no more importance of certain pixels over others. So to use this filter a different tracking method was necessary. The third method from Section 2.1.3 was used but this method ultimately made the tracking slower and not more accurate than the moving average. So the weighted average was not used in the final version.

2.1.6 Flowdiagram

To illustrate every process that happens during a trigger of the stereocamera, the following flowdiagram in Figure 2.6 was designed. In this flowdiagram, the begin and end steps are represented by a red ellipse, the process steps are by a white square and the decision steps are by a blue rhombus. The previously described processes of marker tracking, pixel conversion and filtering are also implemented.

For each camera trigger, the snapshot was converted to a grayscale and then binarized. In the beginning the option could be chosen to calibrate the contrast of the camera. By adjusting the luminance threshold, the image could be calibrated so only the markers were visible.

Once the right luminance threshold had been found, the marker positions in pixel values could be found with the previously determined method in Section 2.1.3. Then, with a threshold filter, involuntary hand movement were filtered. The marker points were then undistorted from the lens distortions of the camera and then converted to cartesian coordinates in the frame of the camera. A moving average filter was applied to make the coordinates smoother. The ray of the robot, representing the coordinates in the robot's frame, was then calculated. The ray was then put into the inverse kinematics to get the joint vectors.

After the joint vectors had been calculated, the biopsy and the shooting decision steps were reached subsequently. If the biopsy option had been selected then a biopsy was performed, and if the shooting option was chosen only the fifth joint, the forward motion of the needle, could move. If the shooting option was not chosen, the robot could move freely in space using the first, second, third, and fourth joints. This was implemented for safety and is further explained in Section 2.1.7.

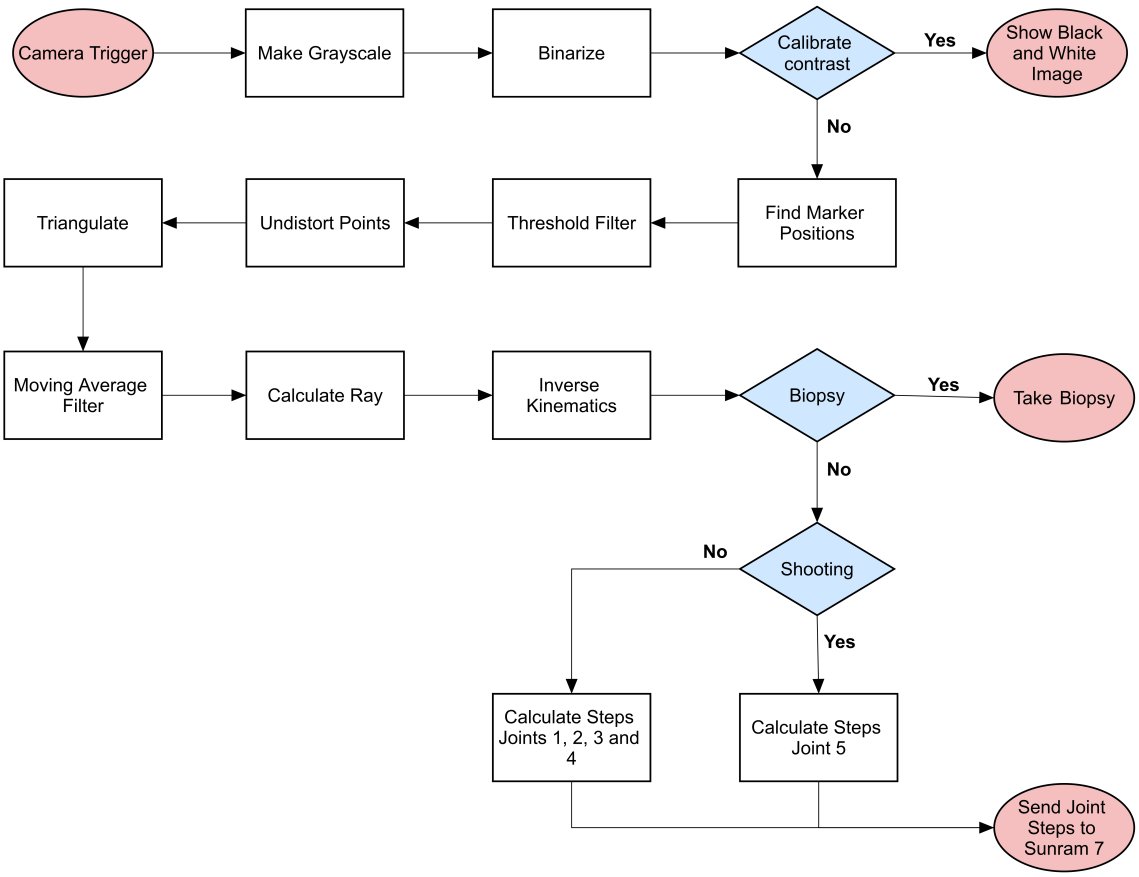


Figure 2.6: *Flowdiagram of the steps per camera trigger. In the red ellipses are the start and end steps, in the white squares are process steps and in the blue rhombuses are decision steps.*

2.1.7 Safety Features

To ensure the safety of the patient during the biopsy and to minimize the amount of tissue that is damaged, two states were added to the robot: The movement state and the shooting state. During the movement state the robot could be moved around with the controller, but the needle was in the position furthest away from the patient. This way the lesions could be targeted with the robot without touching the patient. When a lesion was targeted the operator could go into the shooting state.

In the shooting state, the needle of the robot could only move using the fifth joint. The fifth joint is a prismatic joint. This restriction ensured that the needle could not move sideways while performing the biopsy, therefore avoiding unnecessary damage to more tissue. After the shooting state the biopsy state could be selected. In this state a biopsy was taken. After the biopsy was taken, the state switches back to the shooting state.

The following function determined which state the robot was in. The pseudocode can be seen in Algorithm 6. To switch to the shooting state, a checkbox had to be checked. If the checkbox was not selected, the fifth joint would be positioned all the way back, and the current value of the joint vector would be saved as the shooting joint vector. When the robot was in the shooting state, all the joints remained in the same position except for the fifth joint, which was updated. The current y value was updated to start at the retracted value of the needle, serving as the offset. During a biopsy, the needle was not allowed to move forward.

Algorithm 6 Pseudocode: Shooting function

```
if ShootingSelected and not BiopsySelected then
    Joint 1, 2, 3 and 4 = not Changed
    Joint 5 = y position - offset
else if ShootingSelected and BiopsySelected then
    Biopsy()
else
    Joint 1, 2, 3 and 4 = Changed
    Joint 5 = Retracted
end if
```

To also ensure the safe movement of the robot during the start up, the operator had to move the instrument with the IR markers to the start position. The robot would not take input of the controller until this start position had been reached. The starting position was within 5mm of all joint steps of joints 1, 2, 3 and 4. The starting position was necessary to ensure that when the controller input was taken, the robot's first position was not far from the zero position.

2.1.8 Additional Functionality

To illustrate to the user where the needle would end up during the robot's positioning, a trajectory line was implemented. This line moved around during the movement state but not during the shooting state. In Figure 2.7 the projection line and biopsy line can be seen. The blue line is the trajectory of the robot and the red line indicates the section where the biopsy will be taken.

In the movement state, the robot's joint vector was converted into a ray using forward kinematics. This ray represented the needle's point. By performing a calculation from spherical coordinates to cartesian coordinates, the endpoint of the projection line could be determined. The needle had the capability to move forward by 135mm, which was used as the radius for the calculation.

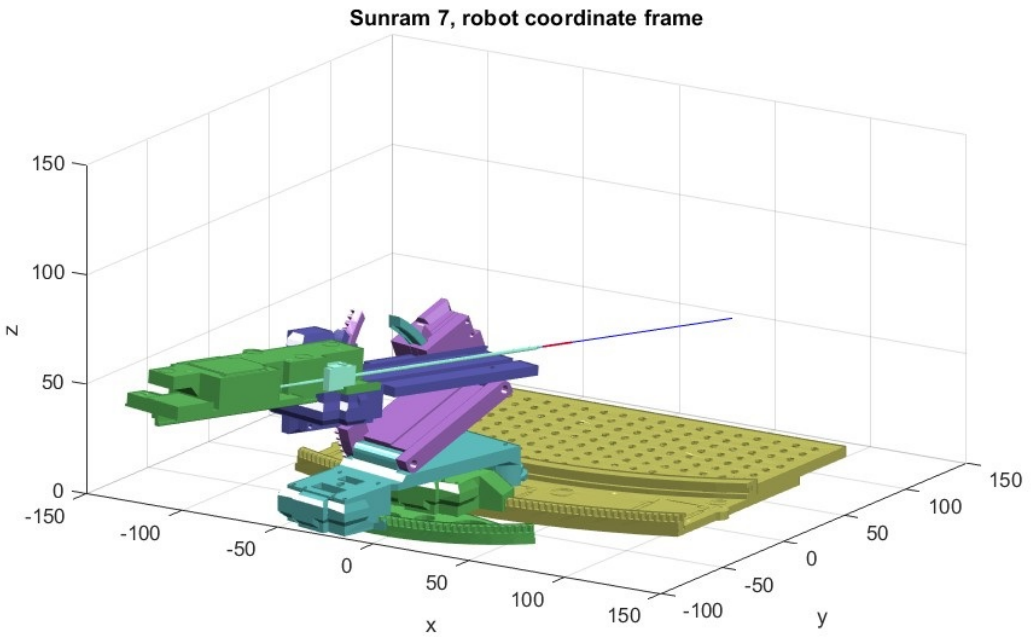


Figure 2.7: *Sunram 7 simulation with Projection line*

Another feature was the end simulation function. This function was used at the end of the biopsy. When the biopsy was taken and the needle was in a safe position, the end simulation function was used. This function moved the Sunram 7 back to the start position where all the revolute joints are zero. This way when another biopsy needed to be taken, the robot did not need to be manually reset to this position.

2.2 Method: Experiments

To test the designed controller three experiments were performed. These experiments tested the latency, the user experience and the accuracy of the controller.

2.2.1 Latency

The latency of the designed controller was assessed through an analysis of response time using a camera. Both the controller and the Sunram 7 were within the camera's field of view. The Sunram 7 was controlled using the designed controller. To capture more frames for accurate analysis, the camera recorded the movements in slow motion.

Six tests were conducted while the camera recorded the movements of both the controller and the Sunram 7. By examining the video footage, the number of frames between the movements of the controller and the Sunram 7 was counted. Utilizing the camera's framerate, the latency was calculated.

The camera that was used is the camera of the Samsung Galaxy S9 plus, with a framerate of 240Hz in slowmotion.

From sources it was found that a latency of 300ms would not degrade the performance of the surgeon. Latencies above 500ms caused overcorrection and therefore degraded performance Rakita et al. (2020); Rogers et al. (2017). To validate the controller and to say that the latency was acceptable, the delay had to be below 500ms.

2.2.2 User Experience

To test the user experience of the designed optical controller for the Sunram 7, an experiment was performed with different users. The users were able to use the controller and control the Sunram 7, afterwards they filled in a System Usability Scale (SUS) questionnaire.

Users were tasked with targeting a marker in a phantom breast to perform a biopsy on. The Sunram 7 robot was placed inside a Esaote G-Scan Brio 0.25T scanner with a breast phantom as target. In this phantom, markers were present as locations of interest to target with the controller. With a serial connection from a laptop to the robot, coordinates were sent. After using the controller a few times, the users were asked to fill out the SUS form, as can be seen in Table 2.1. In this form there are 10 questions with five different responses ranging from strongly disagree to strongly agree. The questions are drafted in such a way that the tone switches from positive to negative in each question and vice versa. This is done to avoid response bias, so the users read each question carefully (Lewis, 2018; Brooke, 2013).

The final score was calculated with Equation 2.2, with the conversion: strongly disagree is 1 point, disagree is 2 points, neutral is 3 points, agree is 4 points and strongly agree is 5 points (Lewis, 2018; Brooke, 2013).

$$\begin{aligned} \text{SUS score} &= (X + Y)2.5 \\ \text{with } X &= (\textstyle\sum \text{odd numbered questions}) - 5 \\ \text{and } Y &= 25 - \textstyle\sum \text{even numbered questions} \end{aligned} \tag{2.2}$$

A passable grade is between 50 and 70, a good grade is above 70 and an excellent grade is above 85 (Lewis, 2018).

Table 2.1: *SUS questionnaire*

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
1. I think that I would like to use this system frequently.					
2. I found the system unnecessarily complex.					
3. I thought the system was easy to use.					
4. I think that I would need the support of a technical person to be able to use this system.					
5. I found the various functions in this system were well integrated.					
6. I thought there was too much inconsistency in this system.					
7. I would imagine that most people would learn to use this system very quickly.					
8. I found the system very cumbersome to use.					
9. I felt very confident using the system.					
10. I needed to learn a lot of things before I could get going with this system.					

2.2.3 Accuracy

The accuracy of the controller was tested by analyzing the error between a marker location in the phantom breast and the targeted location with the Sunram 7.

Two distinct markers in a phantom breast were targeted with the controller. Both markers were targeted thrice, resulting in triplicate measurements. When a marker was targeted, an MRI scan was taken. Since the needle is made of metal and artifacts will be present in the slices, the azimuth and elevation angles were determined and the width of the artifact. With a conversion of spherical to cartesian coordinates, the offset of the artifact on top of the needle point could be calculated. With a MATLAB function called orthosliceViewer, the coordinates of the tip of the artifact in ijk coordinate frame could be determined. The ijk coordinate frame is the voxel coordinate frame of the MRI volume. The i represents the column index, the j the row index and the k the slice number. In Figure 2.8, an artifact can be seen of the needle and the ijk values of the tip of the artifact.

With a H-matrix from the ijk coordinate frame to the robot frame the XYZ position of the artifact tip could be determined. The point of the needle could then be calculated by subtracting the offset of the artifact from the artifact tip.

The difference in position of the centre of the marker and the point of the needle could then be determined. To determine the accuracy a root mean square (RMS) was taken from the difference in positions between the centre point of a marker and the point of the needle. A RMS was used, since this is often used as a measure for accuracy in research Christie and Neill (2022). To validate the accuracy, the accuracy was compared to the Sunram 7's own accuracy of 2-3mm in free air.

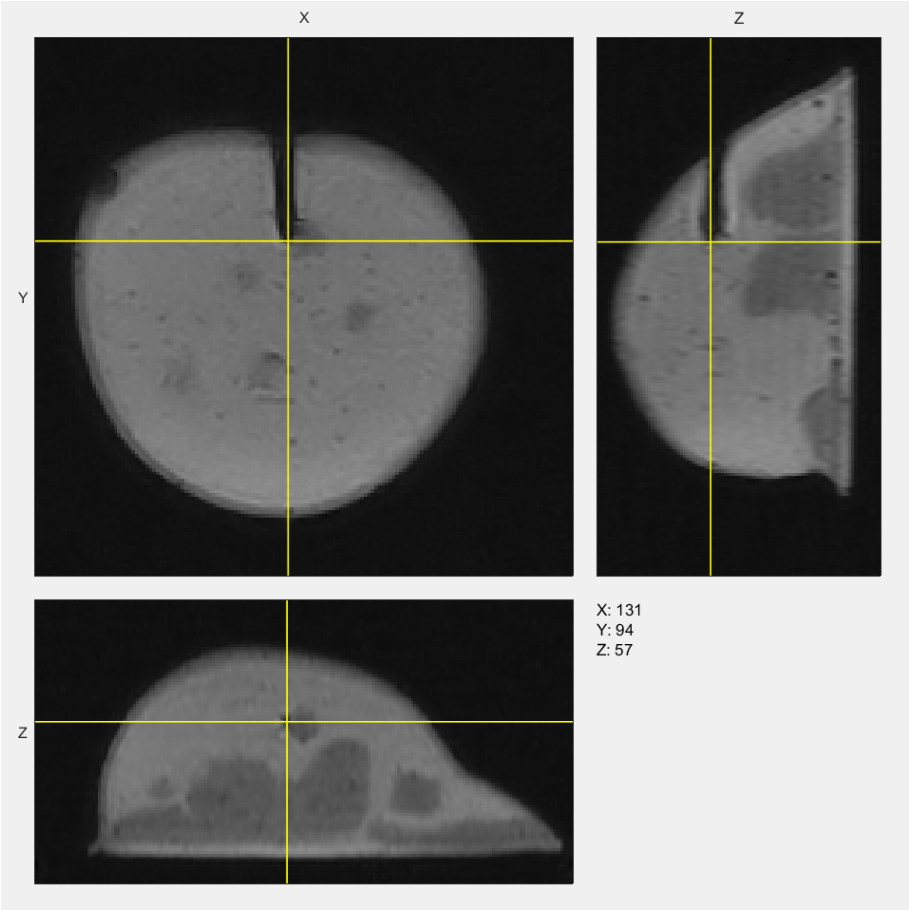


Figure 2.8: *Breast phantom mri scan with needle artifact and ijk coordinates of the tip of the artifact*

The phantom breast on which the accuracy is measured is made of PVC Plastisol strong, plastileurre soft and assouplissant plastileurre. The outer layer is made of a 50% mixture of PVC Plastisol strong and 50% plastileurre soft. The inside is made of 72% plastileurre soft and 28% assouplissant plastileurre. There are 6 markers inside the breast positioned at, but only two were targeted. In Table 2.2 can the positions of the two targeted markers be seen in the frame of the robot. The markers are made of PVC Plastisol strong and will be noticeable on the MRI scans.

Table 2.2: *Marker positions*

Markers	Marker XYZ Position (mm)
1	(1.01, 99.67, 73.12)
2	(-37.47, 99.24, 51.29)

The following MRI scanning sequence was used with the the Esaote G-Scan Brio 0.25T scanner: 3D Hyce with a balanced gradient echo, a TR of 10ms and a TE of 5ms.

3 Design

This chapter covers the design aspects of the controller for the Sunram 7, such as the app design and instrument design. The app design includes the different tabs in which the functions are divided and the monitor interface. The instrument design addresses the different designed models.

3.1 App Design

For the user-friendliness, an app was designed around the algorithms to allow the user easy access to all the different functionalities. In clearly defined tabs the user could navigate throughout the app.

3.1.1 Tabs

Different tabs can be selected to access different parts of the app, such as the start up, calibration and simulation.

When opening the app, the first screen that pops up is the Start up tab. This can be seen in Figure 3.1. This tab allows the user to select the KAYETON stereocamera and to either connect to the serialport or not. Not connecting to the serialport will allow the user to only move the simulation of the Sunram 7 with the controller. Connecting to the serial port allows the control of the real robot and the simulation. When the camera has connected, a red circle will become green, the same thing happens for a red circle in the serialport section.

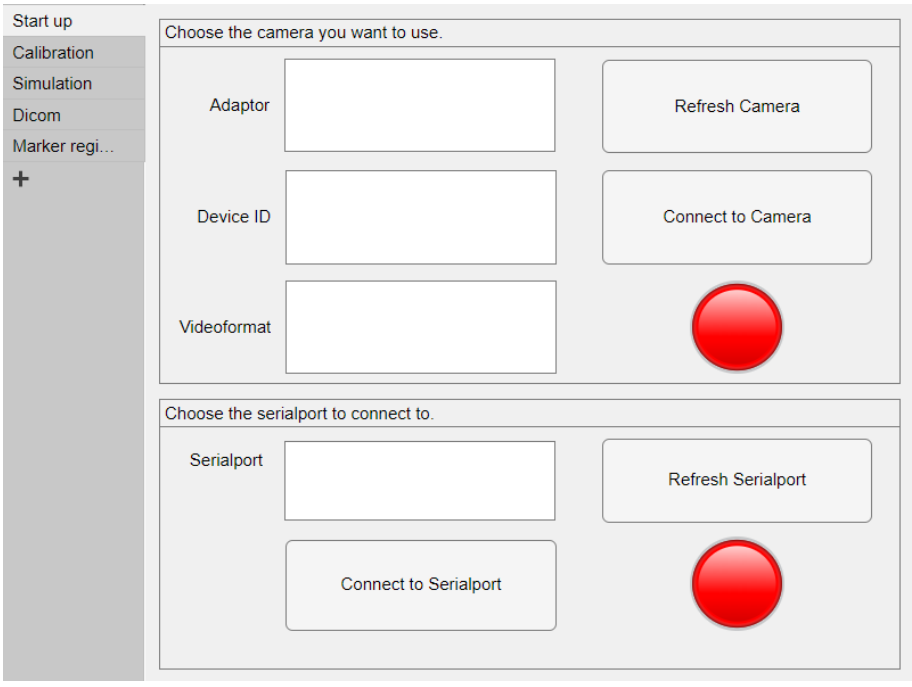
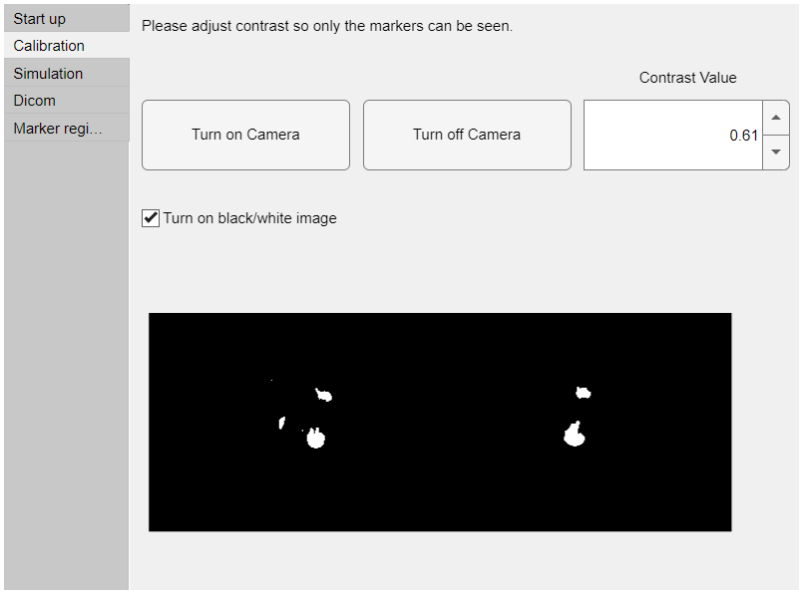
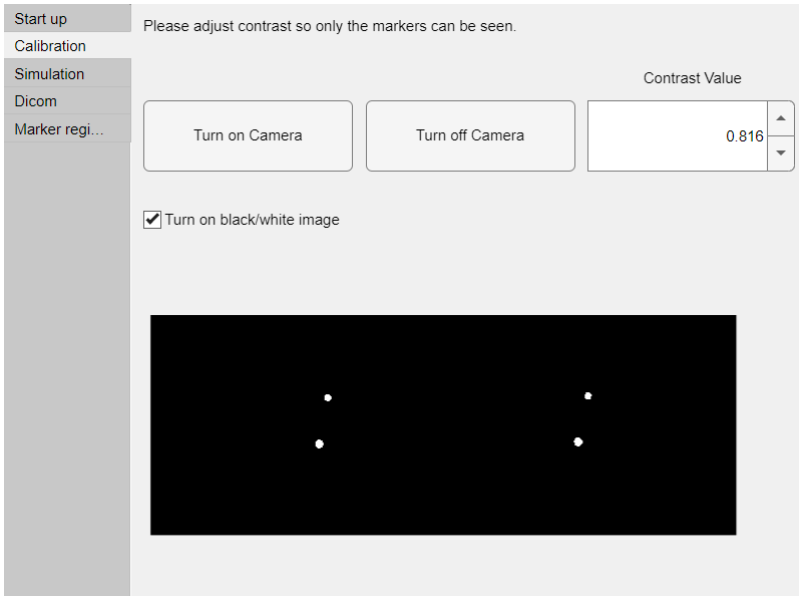


Figure 3.1: *Start up tab: Camera connection and Serialport connection*

The Calibration tab allows the user to choose a luminance threshold with a slider for the detection of the IR-markers. When the camera is turned on, live camera feed is shown to support the user in choosing the right threshold. In Figure 3.2 the influence of a change in luminance threshold can be seen in the calibration tab.



(a) *Calibration tab with low threshold*



(b) *Calibration tab with high threshold*

Figure 3.2: *Calibration tab*

The Simulation tap gives the option to turn on the camera. When the camera is turned on, first a pop-up for the starting position comes on the screen. The starting position was earlier explained in Section 2.1.7. The pop-up can be seen in Figure 3.3. To be able to start controlling the Sunram 7, the user first has to move the controller inside the square on the monitor and make sure the two circles representing the two markers overlap. When the circles overlap the azimuth and elevation angles are approximately zero.

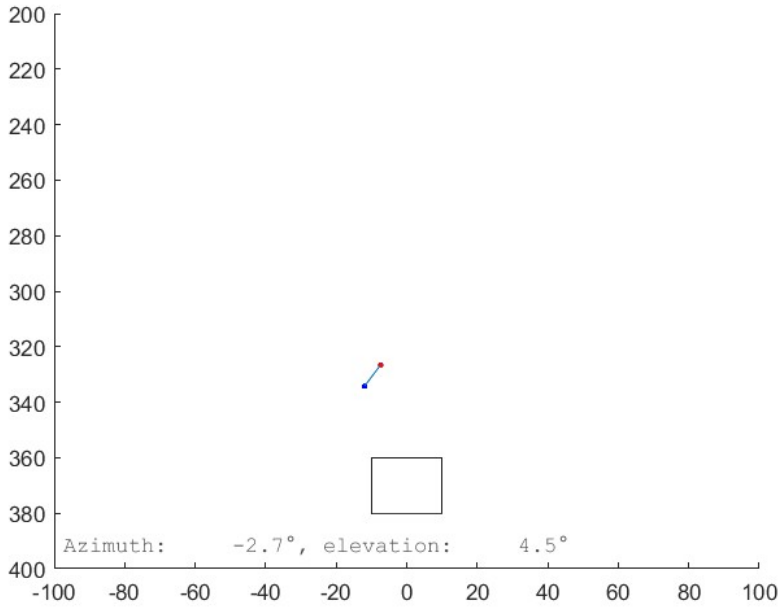


Figure 3.3: *Starting position pop-up*

In the Simulation tab, in Figure 3.4, there is the options to switch states. The standard state the controller is in, is the movement state. When selecting the shooting checkbox, the state switches to the shooting state. When a spot has been targeted from which a biopsy needs to be taken, the biopsy checkbox can be selected. A biopsy will be taken and after it will deselect itself and move back to the shooting state. To end the simulation, the end simulation button can be pressed. This will move the robot back to the starting position. The white circle with kinematics will tell the user if the kinematics are possible or not by either turning green or red.

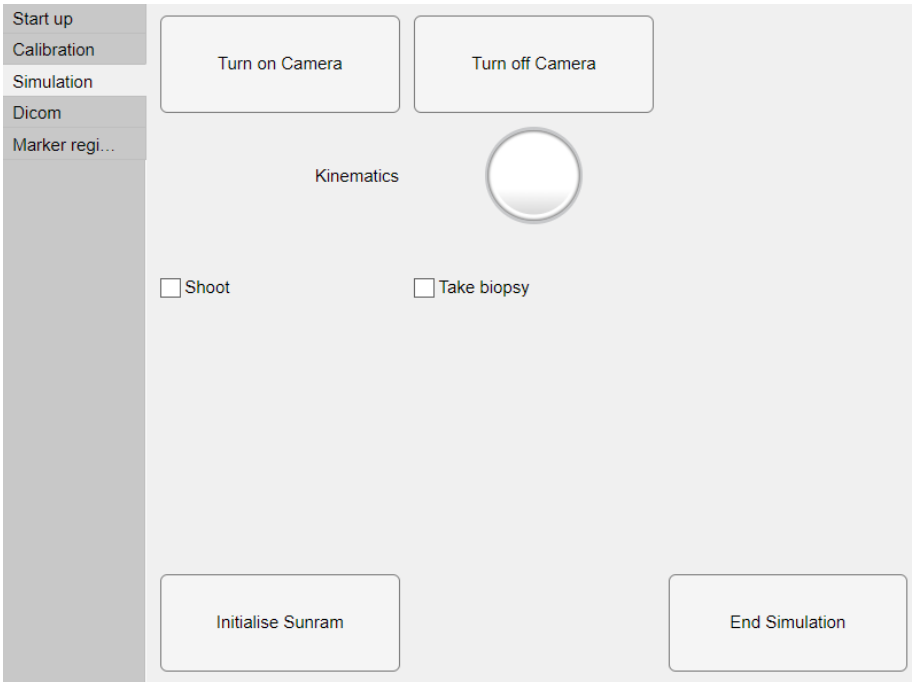


Figure 3.4: *Simulation tab*

The Dicom tab allows the users to load in MRI scans of a phantom breast. The tab can be seen in Figure 3.5. After selecting the right files from the DICOM browser, the files have to be imported into the app. Depending on the MRI scanner used, corrections for distortions could be selected.

The scans of the phantom breast are made on a plateau with six markers on the sides. To detect these markers the scans have to be binarized and can then be detected. The threshold for the binarization can be changed depending on the MRI scans and the number of markers can also be changed.

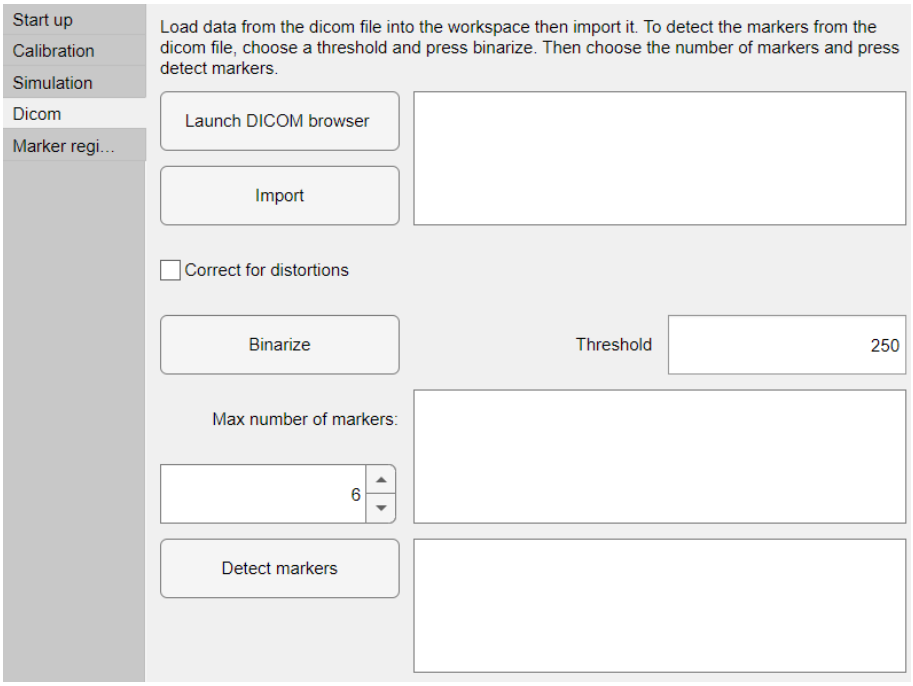


Figure 3.5: *Dicom tab*

The final tab is the Marker registration tab, shown in Figure 3.6. Here the detected markers from the MRI scanner are linked to the reference positions of the markers in the frame of the robot. The reference positions can be imported from a CSV file and then the markers can be coregistered.

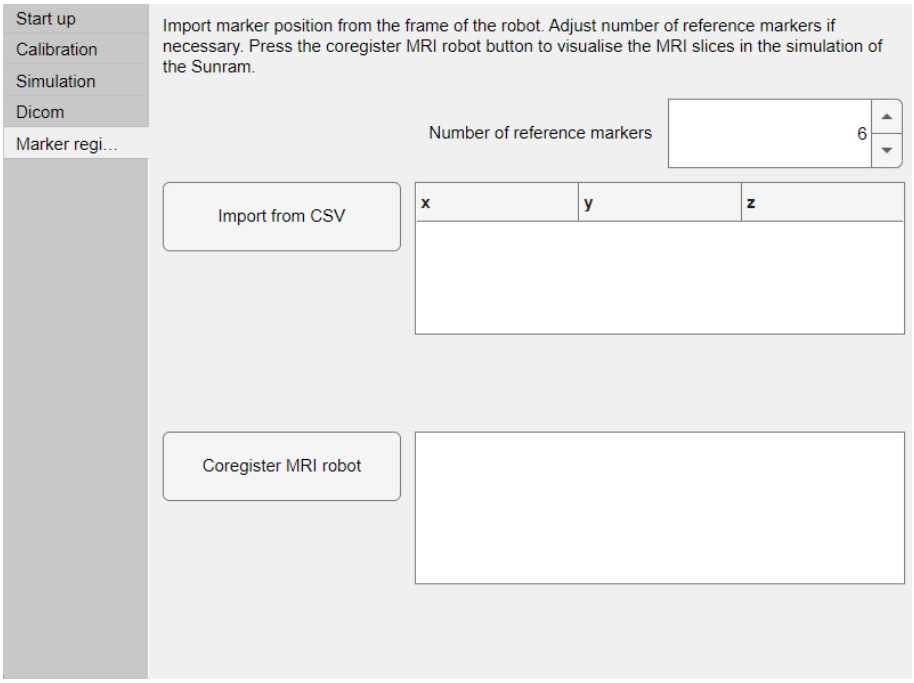


Figure 3.6: *Marker recognition tab*

3.1.2 Monitor Interface

The app can be used with the real Sunram 7 and a simulation with the same inputs. This makes teleoperation with the Sunram 7 possible. When using teleoperation, the simulation can be seen by the user while the real Sunram 7 is in the MRI. To view the phantom in the simulation, the phantom has to be positioned in the simulation exactly where the real phantom is in front of the Sunram 7. To do this, an MRI scan was taken of the phantom breast on the plateau in front of the sunram. The scans of the phantom can then be put in the right position in front of the simulation robot with the Dicom tab and Marker registration tab.

A function was used made by Vincent Groenhuis, to view the MRI scan of the phantom breast inside the simulation. Pseudocode of this function can be seen in Algorithm 7. To visualize the MRI scans of the phantom in the simulation a horizontal intersection plane and a vertical intersection plane is rendered on the projection line of the needle. These planes can be seen in Figure 3.7. For both planes, the function calculates the coordinates of the four corners of the nominal plane in the robot coordinate frame based on the target location. Additionally, it creates a rotation matrix, using the needle elevation and azimuth values. This matrix is called H_rotate_target.

The nominal plane coordinates in the target coordinate frame are obtained through matrix operations with the H_rotate_target matrix. These coordinates are then transformed to the ijk coordinate frame of the MRI.

Next, two vectors are created that represent the directions of the x-axis and the y-axis in the ijk coordinate frame. The function then creates a bitmap with dimensions that represent the rendered slice, this is done by generating a meshgrid of coordinates for the bitmap.

To convert the bitmap coordinates to the ijk coordinate frame, a transformation matrix called H_ijk_bmp is constructed with the two vectors. The matrix H_ijk_bmp can be used to convert bitmap coordinates to ijk coordinates.

The ijk coordinates are rounded to the nearest integers to determine the voxel indices within the MRI volume that correspond to each pixel in the bitmap. The values of these voxels are then assigned to the indices in the bitmap and the slice can be rendered.

Algorithm 7 Pseudocode: Render MRI slices

```
for HorizontalPlane and VerticalPlane do
    Dicom = DicomBrowser()
    MarkerPositions = DetectMarkers(Dicom)
    LinkedMarkers = LinkMarkers(MarkerPositions, ReferencePositions)
    TargetCorners = CalculateCorners(Plane, Azimuth or Elevation)
    H_Rotate_Target = MakeHMatrix(TargetCorners, LinkedMarkers)
    TargetCornersIJK = ConvertIJK(TargetCorners, H_Rotate_Target)
    XVector = CreateVector(TargetCornersIJK)
    YVector = CreateVector(TargetCornersIJK)
    Slice = FillSlice(XVector, YVector, Dicom)
    RenderSlice(Slice)
end for
```

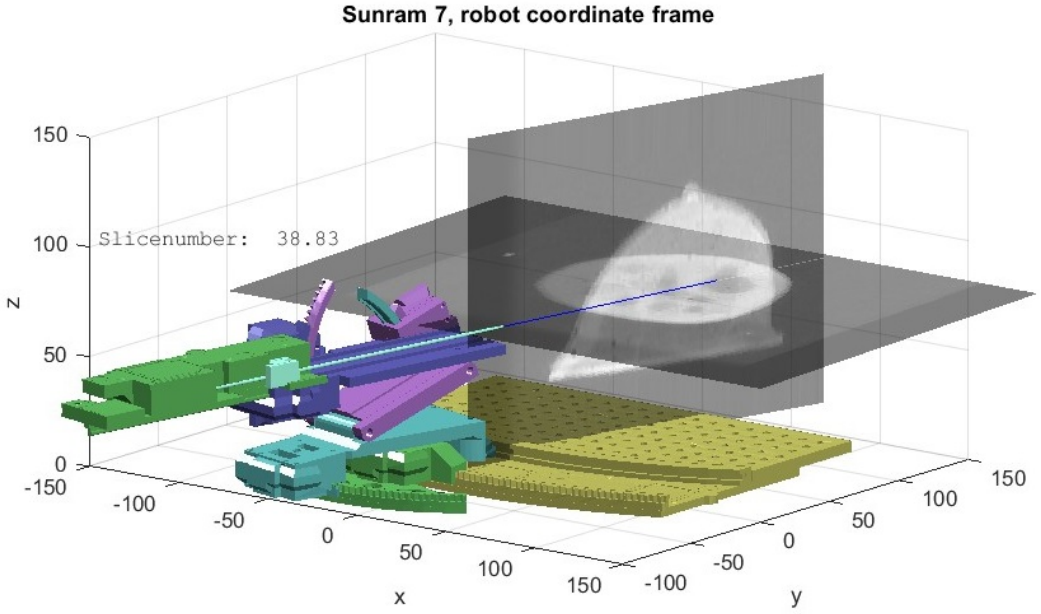


Figure 3.7: *MRI horizontal and vertical plane rendered in Sunram 7 simulation with slicenumber*

3.2 Instrument

For the design of the instrument, the types of IR markers and the 3D printed instrument to which the markers will get attached to could be chosen.

3.2.1 Markers

For the IR markers two variants were tried. There were spherical markers and flat stickers. The spherical markers were the most consistent in the number of pixels and were more reliable in the detection of the marker's position. When the controller was tilted forwards or backwards one of the markers becomes smaller due to it being flat. The spherical markers with a volume did not have this problem due to their omnidirectional visibility. The spherical markers used can be seen in the following Figure 3.8, the markers have a thread inside for M3 screws. The spherical markers are a product of NDI (NDI, 2023).

Figure 3.8: *NDI Disposable Marking Spheres (NDI, 2023)*

3.2.2 First Instrument

The first designed marker instrument can be seen in Figure 3.9. A rectangular design was used with rounded corners so it would lay steady in the hand. With the screw thread the markers can be attached. The screw thread made it so no metal screws were needed in the design. The thread was however flimsy and would easily break.

(a) ***(b)***

Figure 3.9: *First Marker Instrument. a. Side view, b.Front view*

3.2.3 Second Instrument

The second marker has a cylindrical design and includes holes for screws. In Figure 3.10b the cross section of the screw hole can be seen. This design ensured a more secure attachment of the markers to the instrument. However, the cylindrical design resulted in a less stable grip on the instrument. The hand also sometimes blocks some of the markers, since the hand is so close to the markers.

(a) ***(b)***

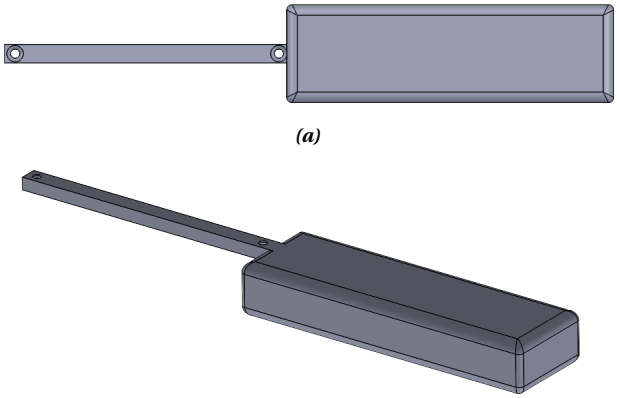
Figure 3.10: *Second Marker Instrument. a. Side view, b. Cross section of screw hole*

3.2.4 Third Instrument

The third design offers a grip for the hand. It is designed after the Magnum Biopsy System, shown in Figure 3.11 (Magnum, 2023).

Figure 3.11: *Magnum Reusable Core Biopsy Instrument (Magnum, 2023)*

This design, shown in Figure 3.12, offers the same screw holes as the previous design. This design improves on the previous design with the hand grip. The addition of the hand grip overcomes the issue of the obstructions caused by the hand close to the markers. However, the grip is not ergonomic for the hand, there is no comfortable way to hold the thick block.



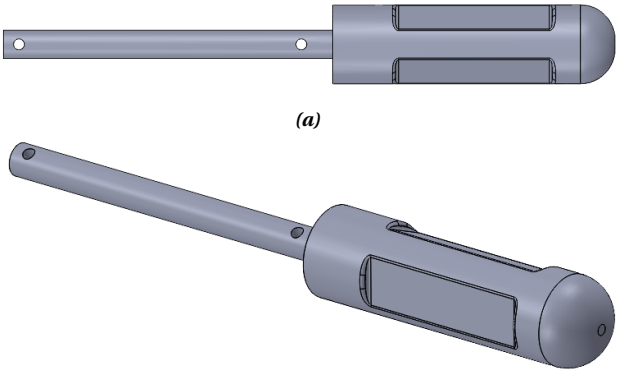
(a)

(b)

Figure 3.12: *Third marker instrument. a. Bottom view, b. Angled view.*

3.2.5 Fourth Instrument

The fourth design and final design of the instrument offers a more ergonomic hold for the hand. It solves the same problems encountered in the third instrument, with the only modification being the improved grip. The instrument is portrayed in Figure 3.13.



(a)

(b)

Figure 3.13: *Fourth marker instrument. a. Top view, b. Angled view.*

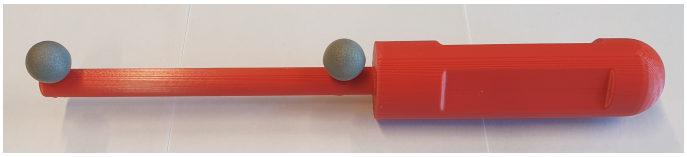


Figure 3.14: *Fourth marker instrument 3D printed with markers*

With the designed app and the fourth edition of the designed instrument with the spherical markers are the experiments performed from Section 2.2. The 3D printed version of the final instrument with the markers attached can be seen in Figure 3.14.

4 Results

Three experiments were performed with the final version of the app in MATLAB app designer. These experiments tested the latency, the user-experience/user-friendliness and the accuracy of the designed controller.

4.1 Latency

The latency of the designed controller was measured through response time with a measurement with the Samsung Galaxy S9 Plus. The recording was done in slowmotion with a framerate of 240Hz. Both the Sunram 7 and the controller were in the field of view of the camera.

Six tests were performed. The video footage of these tests were frame by frame examined, to determine the number of frames between the movement of the controller and the Sunram 7. The number of frames was then converted to seconds. The number of frames and the latency can be seen in Table 4.1.

Table 4.1: *Latency measurements*

Measurement	Number of Frames	Latency (ms)
1	82	342
2	78	325
3	85	350
4	82	342
5	86	358
6	81	338

The mean latency was calculated to be 342.5ms with a standard deviation of 11.17ms.

4.2 User Experience

To test the user experience, users were asked to fill in the SUS scale questionnaire after using the controller and performing a biopsy. This questionnaire is a widely used tool to quickly and easily test the product's usability in human-machine systems. The scores of all users can be seen in Table 2.1. The amount of points each user gave to all the questions can be seen in Appendix A.

Table 4.2: *SUS scores of the users*

User	1	2	3	4	5	6	7
Score	77.5	55	70	82.5	80	35	92.5

The average of the SUS score is 70.4. As mentioned in Section 2.2.2 a good grade is above 70, this makes the overall user-friendliness of the system good.

4.3 Accuracy

As previously explained in the methods Section 2.2.3. The accuracy of the controller was determined by targeting a certain marker with the Sunram 7 via the controller. There were in total 2 targeted markers and each marker was targeted in triplicate. The positions of the targeted markers in the frame of the robot can be seen in Table 2.2.

When a marker was targeted, an MRI scan was taken. To find the point of the needle in the artifact due to the metal, the azimuth and elevation angles were determined. This can be seen in Table 4.3. The width of the artifact is 4.64mm, the needle diameter is 2mm. so the size of an

artifact on one side of the needle is 1.32mm. By using the conversion of spherical to Cartesian coordinates, the artifact length can be calculated to XYZ coordinates. By removing this offset from the tip of the artifact, the needle position can be calculated. The offset in XYZ of the needle artifact can also be seen in Table 4.3.

Table 4.3: *Needle angles of measurements and artifact offsets*

Measurement	Azimuth (deg)	Elevation (deg)	Needle artifact XYZ offset (mm)
1	0.32	-0.83	(0.01, 1.32, -0.02)
2	2.61	-0.95	(0.06, 1.32, -0.02)
3	-0.43	2.43	(-0.01, 1.32, 0.06)
4	2.15	-0.52	(0.05, 1.32, -0.01)
5	1.96	-0.34	(0.05, 1.32, -0.01)
6	-1.30	0.45	(-0.03, 1.32, 0.01)

The tip of the artifact was determined with a MATLAB function called orthosliceViewer and with a H-matrix. In Table 4.4 can the position of the tip of the artifact, the position of the needle and the RMS error be seen.

Table 4.4: *Accuracy measurements*

Measurement	Tip of artifact XYZ Position (mm)	Needle XYZ Position (mm)	RMS Error (mm)
1	(3.82, 94.42, 80.75)	(3.81, 93.10, 80.77)	10.47
2	(4.76, 100.59, 75.07)	(4.70, 99.27, 75.09)	4.20
3	(-4.52, 104.34, 80.54)	(-4.51, 103.02, 80.48)	9.79
4	(-34.15, 99.21, 60.38)	(-34.20, 97.89, 60.39)	9.76
5	(-35.06, 97.60 , 58.62)	(-35.11, 96.28, 58.63)	8.26
6	(-42.56, 101.05, 58.52)	(-42.53, 99.73, 58.51)	8.83

The mean of the RMS error is 8.55mm with a standard deviation of 2.27mm.

5 Discussion

The optical tracking system designed in this thesis offers a method to take biopsies in an MRI. While this controller was designed for the Sunram 7, the optical controller can be used for the other state of the art biopsy robots specified earlier in 1.2 with minimal changes. With a new H-matrix to convert the frame of the camera to the frame of the robot and with inverse kinematics of the robot the controller can be used for other robots.

The discussion further covers the code of the optical tracking system and the results obtained during the experiments.

5.1 Code

The overall code of the optical tracking works as intended. Meaning that the retroreflective markers from NDI (2023) on the designed instrument get tracked with a MATLAB function called BlobAnalysis. With triangulation the pixel values from BlobAnalysis are converted to XYZ coordinates. The coordinates then get calculated from the frame of the camera to the frame of the robot. The jointvectors can be calculated with inverse kinematics from the coordinates in the robot frame. The jointvectors and converted to joint steps and these are sent to the Sunram 7. From the safety features and the additional features everything also works as intended, ensuring increased functionality and safety of the optical controller. Two states, the movement state and shooting state, were introduced. In the movement state, the robot can be maneuvered without touching the patient, while in the shooting state, the needle can only move using the fifth joint to avoid unnecessary tissue damage. The robot also has a starting position requirement for safe movement, which is within a specific range of joint steps. A trajectory line is implemented to show the needle's projected path during robot movement, aiding in positioning. Another feature is the end simulation function. This function moves the robot back to the start position where all the revolute joints are set to zero, eliminating the need for a manual reset.

While the code for the tracking and the added features works as it should, there are a few remarks to be made:

A problem with the used camera is the limiting length of the y direction. The height of a camera is only 480 pixels and the width is 640 pixels. This shorter height than width causes issues in the shooting function. In the shooting function, the current y value is utilized as an offset when the checkbox is selected. This ensures that when the state switches from movement to shooting state, the needle remains stationary. However, when the instrument with markers is held in the middle of the camera and is then moved forwards, the camera's range is insufficient to fully extend the needle in such cases. When the hand is positioned further away from the middle before the shooting state this does not occur. Without a clear marking on the table where to hold the instrument in the movement state, this limitation can be bothersome.

The biopsy check in the code is near the end of a trigger, unnecessarily delaying its execution. The coordinates for the joints are calculated but not used. It would have been more efficient to implement the biopsy check earlier in the code.

Furthermore, the code lacks a preventative measure to handle a specific scenario: when a user for example moves the instrument beyond the right side of the camera's view and then re-enters from the opposite side, causing the new joint vectors to be located at the opposite end of the robot. The robot would need to travel a great distance to reach the new position. Although this situation is unlikely to occur, a preventative measure would have been nice to have implemented.

5.2 Experiments

From the latency experiment, the conclusion can be made that the delay of 342.5±11.2ms is acceptable. The delay is below 500ms, so the performance does not degrade (Rakita et al., 2020; Rogers et al., 2017). The threshold filter might have interfered with the final time of the delay measurements. The threshold filter makes it so only after the position of the marker has changed with two pixels in any direction, only then gets the new position send to the robot. So if the instrument with the marker is moved, then only after two pixel does the position get send to the robot. This might have slightly increased the latency of the controller.

The accuracy of the controller was 8.55±2.27mm. This is more than three times higher than the free air accuracy of the Sunram 7 of 2-3mm. Lesions are generally between 3mm and 70mm Liberman et al. (2006). The controller is not accurate enough for correctly targeting and performing biopsies on smaller lesions of 3mm. The inferior accuracy of the controller compared to the free air accuracy of the Sunram 7 could have happened due to a number of reasons:

Since the controller is controlled by a human, human errors are added to the RMS error. The camera also adds error since it works in discrete pixel values, this however should not have a large influence since the steps of the joints of the Sunram 7 are larger than the distance a pixel of the camera would give. Another cause of the higher accuracy is the decrease in air pressure over the tubes to the Sunram 7. For the accuracy experiments the robot was placed inside the Esaote G-Scan Brio 0.25T scanner. To be able to obtain air from outside the MRI, the cables had to be about 10 meters long. This caused the Sunram 7 to sometimes skip a step. In the simulation the Sunram 7 would be in a different position than the real Sunram 7. So sometimes the targeted position in the simulation was the middle of a marker, but the real location of the needle would be to the side of the marker and not in the middle.

In the accuracy experiment, the assumption was made that the artifact would be isotropic. It was assumed that the artifact length on the side of the needle would be equal to the artifact length on the tip of the needle. However, this assumption turned out to be incorrect due to the inhomogeneity of magnetic fields (Papalouka et al., 2018). The non-uniform distribution of the magnetic field leads to variations in the size and shape of the artifacts, making the initial assumption invalid.

Another noteworthy thing with the accuracy experiment, is that most of the measurements are taken with barely any azimuth or elevation angles, not exceeding three degrees. This was caused by the method the dicom files were shown in the simulation. The markers in the phantom could best be seen when the needle was fairly straight, so the biopsies were also taken without any angles.

The user experience experiment ended with a SUS score of 70.4. This is a good score according to the SUS rating. This means that users generally perceive the controller as usable, intuitive, and efficient. User 2 and User 6 from Table 4.2 fall below the score of the majority of users. Their answers deviated the most in question 9, the question in the SUS questionnaire about their confidence in using the system. This might come from the lack of feedback that the controller provides. The movement of the controller can only be seen visually on a monitor. There is no pressure when the needle of the Sunram 7 touches the phantom breast.

During the user experiments, some users noted that the instrument was made to be held in such a way that rotating in the positive azimuth direction was more natural and effortless than in the negative direction. This observation brings to light a design oversight that unintentionally favored one rotational direction over the other. The pen-like shape of the instrument was aimed to provide familiarity and ergonomic benefits. This shape however caused inequality in the ease of rotation and provided an unbalanced experience in the users. Another note by the users, was that the simulation of the Sunram 7 was shown on a 13 inch laptop screen. The screen was too small to easily see where the tumors were on the screen.

6 Conclusions and Recommendations

6.1 Conclusion

The aim of this thesis was to find an optimal way to control the Sunram 7 with an optical controller that was safe, user-friendly and accurate to use in MRI-guided breast biopsies. An account of safety, with the implementation of the safety features, the optical controller can be said to be very safe. With the movement state and the shooting state, no unnecessary tissue gets damaged. The starting position of the robot also ensured safe movement of the robot at the start-up. The user-friendliness was validated with the System Usability Scale (SUS). The final grade from this experiment was good with a score of 70.4. The optical controller was for users, easy and friendly to use with the intuitive controls. To determine the accuracy of the controller, the latency and accuracy were measured. The latency was 342.5±11.2ms. A delay above 500ms degrades performance, so the latency of the controller is acceptable. The Root Mean Square (RMS) accuracy of the controller was 8.55±2.27mm. This accuracy is subpar for the needs of the Sunram 7. Lesions are generally between 3mm and 70mm. With an accuracy of 8.55mm, the small lesions are untargetable. The designed controller is an optimal way to control the Sunram 7 in terms of its safety and user-friendliness. Based on the accuracy, improvements have to be made to say this controller is an optimal method of controlling the Sunram 7.

6.2 Recommendations

The recommendations are divided into three parts: the code, the design and the experiments.

6.2.1 Code

As mentioned earlier in Section 5.1, to avoid the limits of the y direction of the camera, a sticker/paper should be placed below the space were the controller is held with the limits of the camera. A different conversion of the controller to the Sunram 7 could also work instead of 4:1, 2:1 could be used.

A change in the position of the biopsy check in the signal flow during a trigger removes unnecessary calculations when in the biopsy state. The biopsy check could placed be after the moving average filter.

Finally, a preventative measure should be implemented that gives an error when the user moves the instrument outside of the view of the camera and the instrument can then only be moved back in view from that side. This prevents the robot from traveling significant distances when the instrument is moved back in the camera view from a different side.

6.2.2 Design

To enhance the user-experience, an additional monitor to display the simulation would be advantageous. Users noted that the 13 inch laptop screen was too small for clear visualisation of the tumors. Viewing experience would be improved by using a larger screen.

To overcome the problem of uneven rotational ease of the instrument, a design that is more like a real needle and less like a pen would be more suitable. By stepping away from the pen-like design the rotational experience can be enhanced.

6.2.3 Experiments

In the latency experiment, to obtain a more accurate measurement the threshold filter should be turned off. The threshold filter is implemented to filter out hand tremors when the control-

ler is held stationary, but it does not increase the delay when the robot is already moving. So to determine the latency during movement, the threshold filter should have been turned off. Since all experiments were conducted when the controller was moved from a stationary state.

To improve accuracy measurements, it would have been preferable to conduct a test in a free-air environment prior to testing in tissue. This approach would ensure that any incorrect movement of the robot is consistent across both experiments, the influence of tissue could then also be analysed.

In the user experiments, a broader and more comprehensive analysis method should be used. The System Usability Scale is a quick method to test the user-friendliness of the overall system. A different questionnaire could be used to test different parts of the system with more nuance such as: the system usefulness, the information quality and the interface quality.

A Appendix: SUS Questionnaire Results

Table A.1: *SUS questionnaire result*

User	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Score
1	4	1	4	4	4	1	5	2	4	2	77.5
2	2	3	4	2	4	2	5	4	1	3	55
3	4	3	4	2	4	2	5	3	4	3	70
4	3	2	5	2	5	2	5	2	4	1	82.5
5	2	1	4	2	5	2	4	1	4	1	80
6	2	4	2	5	4	2	2	4	1	2	35
7	5	2	5	3	5	1	5	1	5	1	92.5

Bibliography

Amack, S., M. F. Rox, M. Emerson, R. J. Webster, R. Alterovitz, A. Kuntz, J. Mitchell, T. E. Ertop, J. Gafford, F. Maldonado and et al. (2019), Design and control of a compact modular robot for transbronchial lung biopsy, *Medical Imaging 2019: Image-Guided Procedures, Robotic Interventions, and Modeling*, doi:10.1117/12.2513967.

Arnold, M., E. Morgan, H. Rumgay, A. Mafra, D. Singh, M. Laversanne, J. Vignat, J. R. Gralow, F. Cardoso, S. Siesling and I. Soerjomataram (2022), Current and future burden of breast cancer: Global statistics for 2020 and 2040, *The Breast*, **vol. 66**, pp. 15–23, ISSN 0960-9776, doi:10.1016/j.breast.2022.08.010.

Brooke, J. (2013), SUS: a retrospective, *Journal of Usability Studies*, **vol. 8**, pp. 29–40.

Christie, D. and S. P. Neill (2022), 8.09 - Measuring and Observing the Ocean Renewable Energy Resource, in *Comprehensive Renewable Energy (Second Edition)*, Ed. T. M. Letcher, Elsevier, Oxford, pp. 149–175, second edition edition, ISBN 978-0-12-819734-9, doi:10.1016/B978-0-12-819727-1.00083-2.

Cormier, R. and Y. Bouslimani (2021), Electromagnetic and inertial motion sensor fusion, in *2021 IEEE International Symposium on Robotic and Sensors Environments (ROSE)*, pp. 1–7, doi:10.1109/ROSE52750.2021.9611762.

Dimension, F. (2022), Haptic devices, accessed: 18-3-2023.
https://www.forcedimension.com/products

Edmunds, D. M., S. E. Bashforth, F. Tahavori, K. Wells and E. M. Donovan (2016), The feasibility of using Microsoft Kinect v2 sensors during radiotherapy delivery, **vol. 17**, no.6, pp. 446–453, doi:10.1120/jacmp.v17i6.6377.

KAYETON (2018).
http://www.kayetoncctv.com/Product/Info/416?cid=293&page=1

Leal Ghezzi, T. and O. Campos Corleta (2016), 30 years of robotic surgery, **vol. 40**, no.10, p. 2550–2557, doi:10.1007/s00268-016-3543-9.

Lewis, J. R. (2018), The System Usability Scale: Past, Present, and Future, **vol. 34**, no.7, pp. 577–590, doi:10.1080/10447318.2018.1455307.

Liberman, L., G. Mason, E. A. Morris and D. D. Dershaw (2006), Does size matter? positive predictive value of MRI-detected breast lesions as a function of lesion size, **vol. 186**, no.2, p. 426–430, doi:10.2214/ajr.04.1707.

Magnum (2023).
https://www.bd.com/en-us/products-and-solutions/products/product-families/magnum-reusable-core-biopsy-instrument#overview

Minchev, G., G. Kronreif, M. Martínez-Moreno, C. Dorfer, A. Micko, A. Mert, B. Kiesel, G. Widhalm, E. Knosp and S. Wolfsberger (2017), A novel miniature robotic guidance device for stereotactic neurosurgical interventions: Preliminary experience with the isys1 robot, **vol. 126**, no.3, p. 985–996, doi:10.3171/2016.1.jns152005.

NDI (2023), Aurora Field Generators, accessed: 18-3-2023.
https://www.ndigital.com/electromagnetic-tracking-technology/aurora/aurora-field-generators/

Papalouka, V., F. Kilburn-Toppin, M. Gaskarth and F. Gilbert (2018), MRI-guided breast biopsy: a review of technique, indications, and radiological–pathological correlations, **vol. 73**, no.10, pp. 908.e17–908.e25, ISSN 0009-9260, doi:10.1016/j.crad.2018.05.029.

Patel, N. A., J. Yan, D. Levi, R. Monfaredi, K. Cleary and I. Iordachita (2018), Body-mounted robot for image-guided percutaneous interventions: Mechanical design and preliminary accuracy evaluation, *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, doi:10.1109/iros.2018.8593807.

Patel, R. V., S. F. Atashzar and M. Tavakoli (2022), Haptic Feedback and Force-Based Teleoperation in Surgical Robotics, **vol. 110**, no.7, pp. 1012–1027, doi:10.1109/JPROC.2022.3180052.

PTI (2014). http://www.ptiphoenix.com/products/trackers/VZ4000

Qiu, L., C. Li and H. Ren (2019), Real-time Surgical Instrument Tracking in robot-assisted surgery using multi-domain Convolutional Neural Network, **vol. 6**, no.6, p. 159–164, doi:10.1049/htl.2019.0068.

Rakita, D., B. Mutlu and M. Gleicher (2020), Effects of Onset Latency and Robot Speed Delays on Mimicry-Control Teleoperation, in *2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pp. 519–527, doi:10.1145/3319502.3374838.

Ren, S., X. Yang and N. Qi (2018), A Review of Visual tracking Technology based on Infrared Image, doi:10.2991/ncce-18.2018.30.

Rogers, H., A. Khasawneh, J. Bertrand and K. C. Madathil (2017), An investigation of the effect of latency on the operator's trust and performance for manual multi-robot teleoperated tasks, **vol. 61**, no.1, p. 390–394, doi:10.1177/1541931213601579.

Wang, W., B. Pan, Y. Ai, Y. Fu, G. Li and Y. Liu (2023), Ultrasound-guide prostate biopsy robot and calibration based on dynamic kinematic error model with POE formula, *Robotics and Autonomous Systems*, **vol. 166**, p. 104465, ISSN 0921-8890, doi:10.1016/j.robot.2023.104465.

Wang, W., B. Pan, Y. Fu and Y. Liu (2021), Development of a transperineal prostate biopsy robot guided by mri-trus image, **vol. 17**, no.4, doi:10.1002/rcs.2266.

Xia, S.-B. and Q.-S. Lu (2021), Development status of telesurgery robotic system, **vol. 24**, no.3, pp. 144–147, ISSN 1008-1275, doi:10.1016/j.cjtee.2021.03.001.

Xu, L., H. Zhang, J. Wang, A. Li, S. Song, H. Ren, L. Qi, J. J. Gu and M. Q.-H. Meng (2023), Information loss challenges in surgical navigation systems: From information fusion to AI-based approaches, *Information Fusion*, **vol. 92**, pp. 13–36, ISSN 1566-2535, doi:10.1016/j.inffus.2022.11.015.

Zhang, Y.-n., K.-r. XIA, C.-y. LI, B.-l. WEI and B. Zhang (2021), Review of Breast Cancer pathologigcal image processing, *BioMed Research International*, **vol. 2021**, p. 1–7, doi:10.1155/2021/1994764.