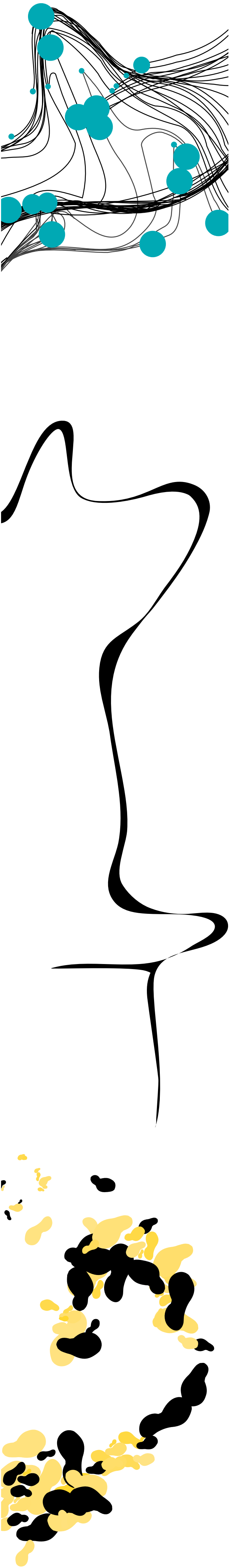


RAM ROBOTICS AND MECHATRONICS

Design of 3D printed biopsy robot and control using mobile phone application

M. (Mart) Bluiminck

BSC ASSIGNMENT

Committee:
prof. dr. ir. S. Stramigioli
dr. F.J. Siepel
dr. V. Groenhuis
ir. J.A.J. Brand

July 2020

022RaM2020
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

Abstract

Breast cancer is a disease with significant worldwide impact. Many hand-operated biopsies have been performed by radiologists to treat a patient suffering from a cancer tumor. Robotic systems have been developed and demonstrated that these may increase the quality and resolution of biopsies. However, with the introduction of robotics, the radiologist's biopsy device is often replaced by a non-intuitive interface such as sliders and buttons on a control panel or a computer program. Furthermore, a biopsy performed by a biopsy robot is costly. This paper therefore proposes a design of a robotic biopsy system, which combines a rapid prototypeable biopsy end-effector with control using a hand-held smartphone.

This method uses low-cost servo motors to generate rotational motions in two dimensions of freedom, plus a stepper motor used in a screw-thread mechanism for a translational injection of the needle. All components are assembled in a 3D printed enclosure. The system uses the built-in accelerometer and gyroscope sensor of the mobile phone to retrieve the phone's orientation, which is reproduced by the robot. A mobile application is designed to retrieve the phone's sensor data and communicate wireless to a controller. In the system several modes are implemented to increase the workability: in biopsy mode, the angular movements are attenuated to enhance precision.

Results show that the designed robot had an angular accuracy of 1.034 degrees based on targeting predefined points in space using an automated program. The phone's orientation was captured with less than 0.30 degrees deviation from the real orientation in all directions. The full system was tested by targeting lesions inside a transparent phantom with 85% success rate (17 hits out of 20 attempts). Using the robotic system developed in this research, it is possible to control a biopsy needle using a handheld mobile phone, and target lesions sized 1 cm and up to 9 cm deep with 85% success rate under visual guidance.

Contents

1 Introduction

1.1 Context

Breast cancer is a disease with significant worldwide impact, especially for women. (Bray et al., 2018). Early detection increases the chance to survive this disease. To detect breast cancer, several techniques such as magnetic resonance imaging (MRI), ultrasound (US), and mammography have been used to detect deviating cells (tumors) inside the breast (Mallapragada et al., 2008). To analyze those cells, biopsies have been performed a lot in the last decades by biopsy specialists using manual breast biopsy needles. However, this approach is dependent on the expertise of the specialist and therefore can require several attempts to obtain a good tissue sample. To solve this problem, many robots have been designed which perform an automated or guided biopsy (Mahmoud et al., 2018). As a drawback, the control is often replaced by a non-intuitive interface, such as sliders on a control panel. Furthermore, those automated robotics are still expensive nowadays (Bélanger-Barrette, 2016).

1.2 Project Goal

Instead of making the full step to automated biopsy robotics, a smaller change is possible. For a biopsy specialist, a radiologist, it would be very intuitive to have a concrete device in hand rather than a non-intuitive interface such as sliders on a control panel or mouse. Also, the absence of necessary physical presence would be beneficial. The *daVinci Surgical System* dealt with this problem and applied a pair of haptic feedback joysticks to control a surgical robot. Also, the physical presence was could be moved, by the surgeon operating with its control device sitting in another room. However, this system mainly focuses on increasing the overview of the operation rather than performing operations on distance (Panteleimonitis et al., 2020). Again, the costs of such a robot are very high. Therefore the search for a cheaper alternative utilizing a concrete input device will be a step ahead. However, making a cheaper alternative should not affect high medical standards. Therefore the question raises:

"How can a mobile device be used together with a biopsy robot to perform a biopsy, which is safe, lower in cost and still accurate enough?"

This research question has some underlying requirements. The system should be able to hit tumors. Typical breast tumors that need to be biopsied range from 1 to 5 cm in diameter and can occur everywhere inside the breast (Mayo Foundation for Medical Education and Research, 2020). As most important, the system, containing biopsy robot and control mechanism, should guarantee the radiologist to perform a biopsy with at least the same precision and safety as a manual biopsy.

1.3 Approach

To meet the requirements, the project aims to use a smartphone, due to too its wide integration nowadays, as a concrete instrument for the radiologist. A smartphone consists of a lot of sensors, such as a gyroscope, accelerometer, and magnetometer. The project aims to extract the current phone's orientation from the built-in sensors. The mobile phone can easily be connected to other wireless devices and data can be transmitted to the biopsy robot. No low-costs, ready-to-go biopsy robot is available. The project, therefore, aims to design a 3D printed biopsy robot, which approaches medical standards and performances of other biopsy robots. The tumor location is defined by a point in space, so a robot with at least three dimensions of freedom (DOFs) is required to approach different tumors. To reduce complexity, the project only aims to design this 3 DOFs end-effector rather than a robot arm with 5 or more DOFs. The project purposes to control this end-effector with a microcontroller. The sensor data gathering, transmissions, and other inputs will be covered by a mobile application.

1.4 Outline

The remaining section of this report are organized as follows: Chapter 2 describes some state-of-art robotic systems and visualization of tumors, Chapter 3 describes the individual parts of the solution, Chapter 4 merges the best approaches and describes the design of the biopsy robot, Chapter 5 tests the build quality and accuracy of the robot, Chapter 6 tests the implemented control algorithm, Chapter 7 tests the complete system by targeting tumors inside a transparent phantom visually and finally, Chapter 8 describes the conclusion, the improvements for further research and wraps up the paper.

2 Background

Before the design of a biopsy robot and the control, a look at the current state of art biopsy robots and mobile phone applications should be taken. Also, the usual visualizations of tumors are stated.

2.1 Types of Automatic Biopsy Robots

In the last decade, a lot of semi- and full automatic biopsy robots are made. The two most common types are the table model and the robotic arm. This research only focused on electrical driven robots. Pneumatic and hydraulic driven robots are not considered, because the project aims for an electric robot.

Table model

One way to do a biopsy is by creating a table model shown in Figure 2.1, which is also used in 3D printers. This model requires the phantom with a tumor positioned on the flat (grey) surface. The needle holder is moved around the phantom using two stepper motors at a constant height. After positioning, the injection is injected by a third stepper motor.

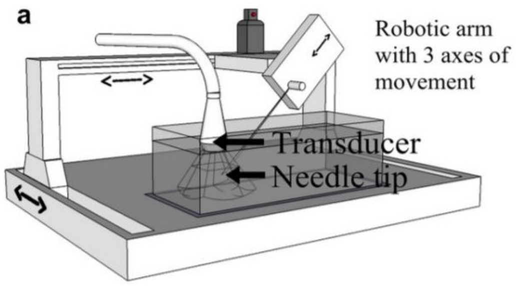


Figure 2.1: The table model biopsy robot of Liang et al. (2010)

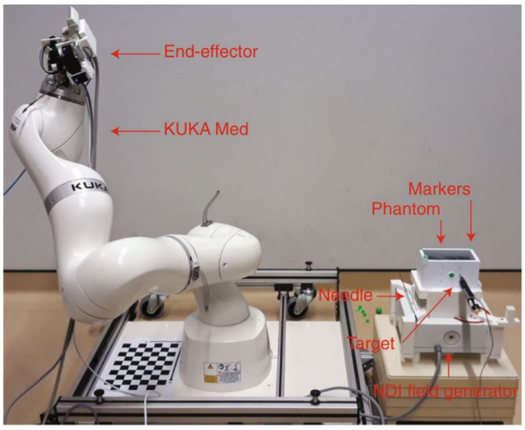


Figure 2.2: The robotic arm of Welleweerd et al. (2020)

Robotic arm

The robotic arm is more employed compared to the table model. It is used in different configurations, with its base on the ground, shown in Figure 2.2, and with its base attached to a cantilever. The more human-likeness of the robotic arm is desirable from a patient's point of view. The robotic arm is nowadays mostly used to serve as a guidance robot, were still someone manually has to insert the needle through an automatic positioned holder (Welleweerd et al., 2020).

2.2 Types of Needle Injection Mechanisms

Several applications for needle injection mechanisms are present, all based on the biopsy needle. The biopsy needle consists of two parts, an inner and outer needle. Both are carefully inserted and stopped in front of the tumor. A biopsy gun is fired, which penetrates the inner needle through a tumor keeping the outer needle stationary. After shooting, the inner needle is kept stationary, while the outer needle slides over the inner needle to collect the tissue sample. Afterward, both are rejected simultaneously.

Few electronic techniques are used in previous researches to inject the needle. One of the few techniques is a linear actuator. A fixed nut, attached to a biopsy needle, is moved up and down on a rotating screw-thread. The rotation is performed using a DC motor Nelson et al. (2012).

The needle is often guided, but also sliding mechanisms are tested to insert a needle over a predefined path by external pressure. Tanaiutchawoot et al. (2014) used this technique in combination with a 5 DOFs robot, shown in Figure 2.3. After the needle box is positioned, the needle can be injected straight. A close-up of the insertion mechanism is shown in Figure 2.3b. The sliding mechanism consists of two linear bushes connected to the needle box, which slide over a metallic tube. Linear bushes (linear-motion bearings) provides movement in only one direction and exists of an outer hollow cylinder with small bearing balls on the inside. These balls make contact with an external tube and provides low frictional movement. Two bushes are used in the robot to restrict rotation around the shafts.

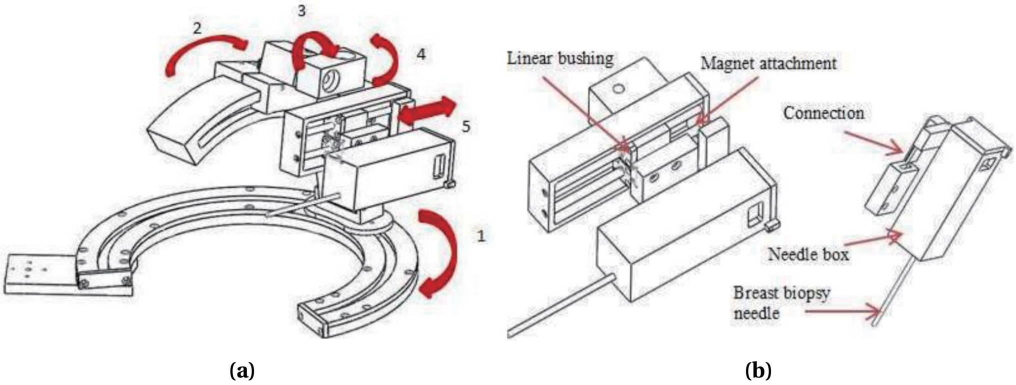


Figure 2.3: Visualisation of 5DOF robotic arm. (a) Schematic representation of whole robot with all directions of movement indicated by red arrows. (b) Zoomed in on needle insertion mechanism containing linear bushing and magnetic connections. Figures from Tanaiutchawoot et al. (2014).

During needle injection, the phantom could deform due too the needle pressure on the phantom. This effect could be minimized using linear actuators to counteract the deformation and reduces the number of attempts to get a sufficient sample(Mallapragada et al., 2008).

2.3 Ways to Visualize a Tumor

The ways needle biopsies are visualized can be subdivided into three commonly used techniques. These techniques are stereotactic mammography, magnetic resonance imaging (MRI), or ultrasound (US) imaging. Ultrasound is the most applied technique because it is cheap and real-time (Mallapragada et al., 2008). Ultrasound enables to determine the position of a tumor in a 3D space. MRI is a precise way to determine the position but requires the lack of metallic components in a scan.

2.4 Sensors In Mobile Phone

The mobile phone contains lots of sensors, which are broadly used in applications. Specifying position related sensors, The mobile phone sensors measure six DOFs, three translation, and three rotation. An accelerometer, gyroscope, and magnetometer are combined to measure the six DOFs. For this, the mobile phone combines an accelerometer, gyroscope, and magnetometer.

2.4.1 Accelerometer

The accelerometer inside a mobile phone is a micro-electro-mechanical system (MEMS) using tiny springs to balance a mass. The gravitational force attracts the mass towards the center of the earth. This attractive gravitational force is measured in three dimensions. Due to a stable (barely) changing gravitational force the accelerometer is often used for measuring the angle of a device, which is a combination of the three accelerometer components. Due too the mass-spring system inside, the measurements result in shaky movement, which can lead to inaccurate measurements looking at one specific time instance. However, over a long time, the gravitational force did not change and therefore the measurements do not drift away over time. Another major drawback is the disability to measure yaw, rotation around the vertical axis. The accelerometer is utilized a lot, but for orientation tracking in combination with the gyroscope and or magnetometer.

2.4.2 Gyroscope

The gyroscope inside a mobile phone includes a mass. This mass has an outer shell configuration, which enables the mass to spin around its axis independent off the tilt of the outer frame. The gyroscope measures the speed the frame is rotating. A major advantage is a capability of measuring in all dimensions of freedom, containing the yaw, which could not be measured by the accelerometer. Also the short time instant measurements are accurate because no mass-spring system is present. To achieve the orientation, the data, in rad/s should be integrated. Instead of a constant gravitational force, the gyroscope does not have a stable reference and therefore easily drifts over time, (within seconds).

2.4.3 Magnetometer

The magnetometer inside a mobile phone measures the magnetic field in three axes. One application is the compass, which requires the measurements of the earth's magnetic field. The earth's magnetic field changes on the time scale from milliseconds to millions of years. Short term changes are due to currents in the ionosphere and longer changes are due to changes in the earth's iron-rich core (Merrill et al., 1998). The short term changes are small enough to barely affect the measurements and therefore the magnetometer acts without drift, as a stable reference point. All axis can be measured and therefore it can measure also yaw. A major drawback is the location dependency because of the magnetic field change from location to location. Also to use the magnetometer for orientation-based projects, the data points should span a sphere around zero, which requires a good calibration. Tilt compensation is also required to be useful.

2.5 Current Mobile Phone Applications

Several applications are available on the internet such as SensorKinetics and Androsensor which visualize the build-in phone sensors. Another application is called SensoDuino. This app was designed with the purpose to log the data and send it by Bluetooth to an Arduino microcontroller. In this application, some parameters can be tuned, but the ability is very limited. To achieve more tuning ability and be able to add functionalities, MIT App Inventor can be used. These are programs that allow using simple building blocks from a menu, which is a very intuitive and easy way of creating a simple application. The sensor readout is integrated and can easily be used. Also, connectivity via Bluetooth or WiFi is integrated very well. Furthermore, a big community online is available, which can help.

3 Analysis

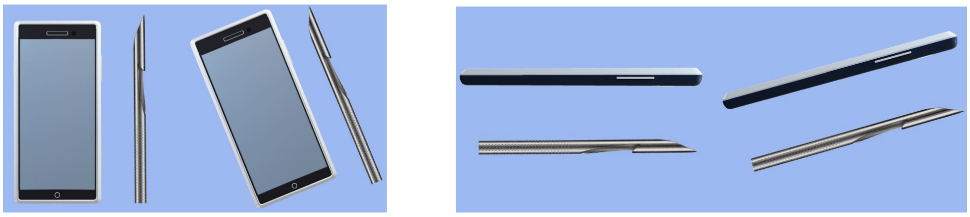
The possible solution can be separated into different parts. First, the robot design and choices are discussed. The robot design again can be separated into reaching a tumor in 3D space and performing a needle injection. The stability and additive hardware such are motors are also discussed. Afterward, the control possibilities of the robot are explained, containing the sort of controller, the differences inside a control group, and how the data should be processed. Last but not least, the implementations to guarantee the radiologist to perform a biopsy with at least the same precision and safety as a manual biopsy are stated, containing some useful modes and some safety regulations.

3.1 Robot Design

3.1.1 Reaching tumor in 3D space

Reaching a tumor can be done in two ways, using the table model or a robotic arm (Chapter 2, Background). The robotic arm has preference above the table model because, in the table model, the patient has an unpleasant construction around him. Also, the robotic arm is much more human-looking.

For reaching a tumor in 3D space, the least complex mechanism consists of two rotational movements and one translational DOF, the needle injection (details in Section 3.1.3). These rotational movements, the yaw movement around the z-axis and the roll movement around the y-axis, need to be captured by the phone and are shown in Figure 3.1.



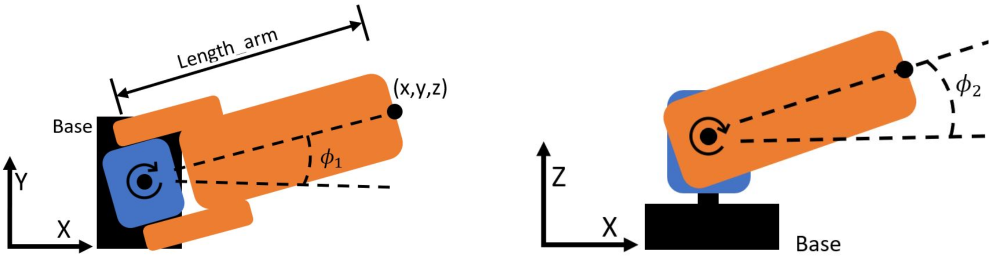
(a) Yaw movement around Z-axis

(b) Roll Movement around Y-axis

Figure 3.1: Schematic representation of desired mobile motion and the corresponding needle movement, (a) rotational movement in X-Y direction and (b) rotational movement in X-Z direction

The mapping of the two rotational movements into a robotic arm is shown in Figure 3.2. The third translational movement is pointed out later on. This robotic arm can span a sphere with the lowest possible complexity, compared to more arms attached. The robot consists of three parts, a black base, a blue mid-base, and an orange arm. The blue body can rotate in the x-y plane around the center of the fixed black base and therefore spans a circle by varying θ_1. Connected to the center of the blue body is the orange arm, which can rotate in the x-z plane. Varying θ_2 spans a circle in this plane. Combining those two movements make it possible to let the tip of the needle move among a sphere with a diameter of length_arm. The (x,y,z) position of the tip of the orange arm is given in Equation 3.1.

$$\begin{aligned} (x) =& \text{length\_arm} \cdot \cos(\theta_1) \cdot \cos(\theta_2) \\ (y) =& \text{length\_arm} \cdot \sin(\theta_1) \cdot \cos(\theta_2) \\ (z) =& \text{length\_arm} \cdot \sin(\theta_2) \end{aligned} \tag{3.1}$$



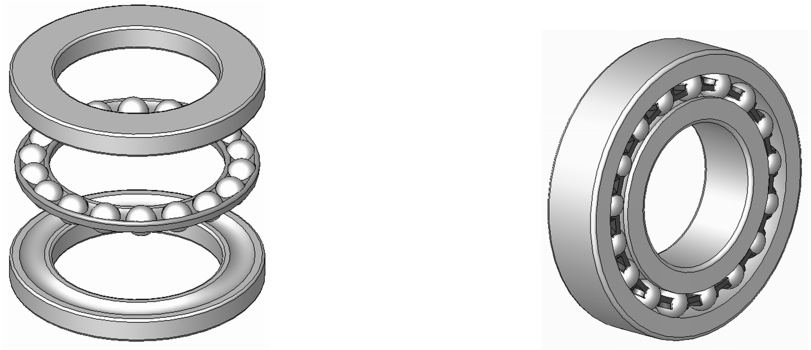
(a) Top View

(b) Side view

Figure 3.2: Schematic representation of (a) top view and (b) side view of needle positioning mechanism. Movement of parts indicated by circular arrows.

3.1.2 Stable Rotation

The two rotational motions described above should be stable and should be perpendicular to the axis of rotation. Therefore connecting one side of the arm to a single rotational point is not desirable. To create stable rotation (of Figure 3.2a) without interrupting the rotation can be done in several ways. One way is by 3D printing two much bigger circles, with a hole in one the middle where the motor connection can be made. 3D printing is smooth in x-y direction but not in the z-direction, where the layer lines are visible. Therefore two smooth flat plastic layers sliding over each other can be printed. This ensures stability, but also creates friction. A better implementation of the same idea is adding metallic balls between the two layers, which is called a thrust bearing. The metallic balls can roll and distribute the force, reduces friction, and enables smooth rotation. Such a bearing is shown in Figure 3.3a, were in the bottom and upper part a notch is made to fit the bearing balls. These bearing balls are distributed evenly over the whole circle using a ring. To reduce weight, the bearing can be 3D printed, which requires printing in the z-direction, which is not smooth. However, the running surface is flat so there are no radially extending layer lines, but only concentric, which does not impede the ball rolling motion.



(a) Trust Bearing

(b) Regular Bearing.

Figure 3.3: Two types of bearings to increase stable rotation. (a) Trust bearing for parallel rotation and (b) regular bearing for concentric rotation. Figures from Wikipedia contributors (2020c)

Next to the horizontal rotation (of Figure 3.2b), the second motion needs to rotate in the vertical plane. To ensure the perpendicular rotation, both sides of the orange arm in Figure 3.2 should be supported. A 3D printed solution is not possible, because a 3D printed cylinder connected to the blue base, and a hole printed in both sides of the orange arm do not allow smooth rotation around each other due to the impurities while printing in the z-direction. Also over time, the plastic will sand off due to friction. Using a bearing, shown in Figure 3.3b, on both sides between the 3D printed cylinder and hole allows a stable perpendicular movement. The bearing is made out of metal, which is significantly heavier than plastic.

3.1.3 Perform Needle Injection

The third DOF to reach a tumor is the translation of the needle. The biopsy needle consists of a shooting mechanism. This mechanism is a complex system containing carefully selected springs, which therefore is not achievable and necessary in this research. Therefore a simple needle (size of biopsy needle) with a sharp tip will act like a biopsy needle in this research.

To perform a biopsy, a needle has to be inserted in a forward direction. One possibility is using a long small notched piece in combination with a gear, driven by a rotational motor. This principle is shown in Figure 3.4. The rotational movement is translated into linear movement. The object slides through a guide and can move back and forth. The principle in Figure 3.4 applied a servo motor (more details in Section 3.1.4), which has limited rotational range of 180 degrees. For a deep insertion, a large gear with many teeth is required. The disadvantage is the large linear displacement by rotating one degree, which leads to less accurate injection depth.

Another method is a screw thread driven by a DC motor (Gollapudi et al., 2020) shown in Figure 3.5. When the DC motor rotates, the screw thread also rotates with the same speed, due too the shaft coupling. On this screw-thread, a nut is attached, which is fixed and cannot rotate. This moves the nut forwards or backward, depending on the direction of rotation of the motor. Sometimes gears are placed in between the motor and the screw-thread to increase torque applied on the screw thread. The number of components and corresponding weight is a drawback, because metal is significant heavier than plastic.

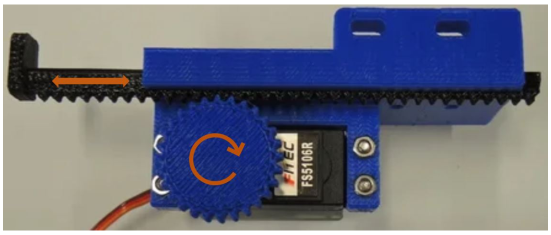


Figure 3.4: The schematic diagram of a geared linear servo driven actuator. Figure from Printables (2018)

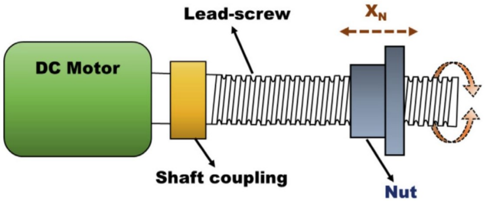


Figure 3.5: The schematic diagram of a geared linear electromechanical actuator. Figure from Gollapudi et al. (2020)

As a safety feature, the needle requires limitation on the injection depth. Two possibilities are available, software- and hardware limitations. The software solution, in combination with one of the two injection mechanisms, could be a limitation on the angle to rotate (for the solution in Fig 3.4) or the number of rotations (for mechanism in Fig 3.5), which translate to a distance to inject. Nevertheless, motors and control may for some reason not function. In that case, a physical backup limit that lets two parts collapse against each other restricting the injection depth is desirable. This will lead to high currents inside the motors but ensures that in no case the needle injects too far. This high current could also be detected and feedback that the endpoint has been reached. However, detecting this current is more difficult and requires more electronics compared to the physical end stop or software limitation. Another possibility is an electric end switch which triggers an external stop mechanism or circuit breaker. Although it would be beneficial, it increases the complexity of the system. Combining software and hardware limitation to ensure safety is the best feasibility.

3.1.4 Type of motor

In the previous sections, the injection mechanisms and end-effector are analyzed. In all cases, rotational motors are required. Two often-used motors are servo and stepper motors and are discussed below. DC motors are also very applicable but requires gears and more difficult control in combination with an encoder. Therefore the solution is based on the motors described below, which are easier and more simple to control.

Servo Motor

A servo motor uses a DC motor in combination with gears and control hardware inside the motor. The servo requires a pulse width input which determines the position. The control hardware consists mostly of a position encoder or rotary encoder which measures the position (and speed) and uses a PID (or bang-bang) control algorithm to let the error between the measured angle and setpoint reduce to zero, which stops the servo motor from rotating (Wikipedia contributors, 2020a). Servo motors have limitations due to the fact that they can only rotate 180 degrees, which is useless in applications such as the screw-thread earlier. An advantage is that the servo motor does not require any external driver, due to the integrated control hardware. Servo motors are widely used, such as in the research about a nine-piece snake from Song et al. (2011).

Stepper Motor

The second option for rotational motion is a stepper motor. Stepper motors consist of a shaft with precise increments and coils around it. The shaft with increments is a permanent magnet. By powering a pair of the coils, magnetic north and south are created, which attracts the closest aligned increment of the permanent magnet. The order the coils are powered results the permanent magnet to rotate (Wikipedia contributors, 2020b).

Stepper motors are often used in 3D printers and the table robot (robot of Liang et al. (2010) in Chapter 2). Stepper motors have some advantages and some disadvantages compared to servo motors. One disadvantage is that stepper motors approximately have ten times more magnetic pairs inside and therefore are much heavier (Burris, 2020). The torque is often a little bit lower but may vary from model to model. However, an advantage is that the position can be easily be controlled because the different steps can be counted. Another advantage is that the stepper motor can rotate more than 360^o, which makes it possible to use in needle injection applications earlier shown in Figure 3.4 and 3.5. The benefit is that for a longer injection in Figure 3.4, the gear not have to increase, but more revolutions can be made, which increases the injection resolution. One drawback is that the stepper motor requires a driver to be able to work and requires much more current, compared to servo motors.

3.2 Communication and Control

3.2.1 Controller

Combining the controller wirelessly to the phone is essential. For wireless communication, several techniques can be used, such as Wi-Fi, Bluetooth, or similar wireless communication techniques. Wi-fi has the ability for high-speed data via the internet. On the other side, connecting a biopsy robot to the internet enables lots of other users to interact with the robot, and therefore require more difficult safety regulation. Bluetooth on the other side has a more limited range of at most 30 meters outside nowadays. Indoor the range is even less. To intervene with this signal, the disturber has to be much closer to the robot compared to Wi-Fi. One downside of Bluetooth is the lower bit-rate. The required bit-rate will be low because the number of transmitted sensor data is low and therefore is no problem. All in all, the Bluetooth has the preference above Wi-Fi, because of its easier use and safety.

Combining Bluetooth with the controller is necessary. Also, the computational power and ease of use makes are important, which fits a microcontroller. Microcontrollers are fast in executing operations in a sequence one by one and are easy to code. A very popular microcontroller is the Arduino UNO. Arduino offers lots of microcontrollers for different purposes. Also lots of libraries, such as `<Servo.h>`is available for easy motor control of servo motors.

The `<Servo.h>` library, is an easy way to control the servo motors. It automatically creates the pulse-width modulation (PWM) signal with the setpoint angle as input with `myservo.write(angle)`. This angle is converted to a pulse-width which is used to rotate the motor to the angle. A pulse-width of 600 µs results in 0° and 2400 µs results in 180°. Also, directly the pulse-width can be set by `myservo.writeMicroseconds(pulse-width)`. Both allow only integer input values and therefore setting the pulse-width directly increases the range from 0-180 degrees to 600-2400 µs, which is 10 times more precise.

The combination between the micro-controller and Bluetooth can simply be made using a variety of breakout Bluetooth modules available. Nevertheless, Bluetooth integrated micro-controllers are also available, such as the Huzzah ESP32. The Huzzah ESP32 has beneficial features over the Arduino UNO, such as more memory and flash. Another advantage is the dual-core availability. This dual-core can be used to run two tasks parallel, such as positioning and data acquisition on one core and injecting a needle on the second core. A detailed specification comparison is shown in Table 3.1.

	nr of Cores	Nr of bits [bit]	CPU [MHz]	Wifi/ Bluetooth	RAM [Kb]	Flash [Kb]	GPIO PINS	ADC Pins	DAC pins
Arduino Uno	1	8	16	NO	2	32	13	6	0
Huzzah ESP32	2	32	160	YES	512	16	36	18	2

Table 3.1: Comparison between Arduino UNO and Huzzah ESP32

3.2.2 Data Conversion

The mobile phone not directly outputs the required angles but requires some processing. The definitions of the mobile frame around the phone are important, which are shown in Figure 3.6. The two motions of Section 3.1.1 can be achieved by converting the phone's output data to the roll angle (around X-axis) and yaw angle (around Z-axis).

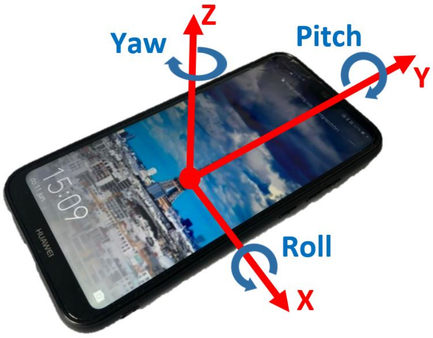


Figure 3.6: Definition of frame around mobile phone

Rotating around X-axis (Roll)

For rotations around the X-axis, an accelerometer can be used. To obtain the roll angle (in radians) from the accelerometer data, the formula shown in Eq. 3.2 (Pedley, 2013) is used, which has range of [-π,π].

$$\text{roll} = \text{atan2}\left(\frac{\text{acceleroY}}{\sqrt{(\text{acceleroX})^2 + (\text{acceleroZ})^2}}\right) \tag{3.2}$$

The stability over the long run is a useful property of the accelerometer. However, the drawback of the accelerometer is, that it is too noisy and inaccurate in the short-run (Chapter 2). Therefore a combination will be made with a gyroscope.

The gyroscope data outputs velocity, radians per second. To get an angle this gyroscope data should be integrated and afterward, high pass filtered. Gyroscope data drift over time but is stable in the short run. The combination with the accelerometer is therefore made, which requires a complementary filter (Douglas, 2018). For rotations around x-axis, the block diagram is shown in Figure 3.7. By tuning the factors 0.95 and 0.05 in this figure, the contribution from each sensor can be tweaked. This data may require to be smoothed out, which requires a low pass filter, which is explained later on in this section.

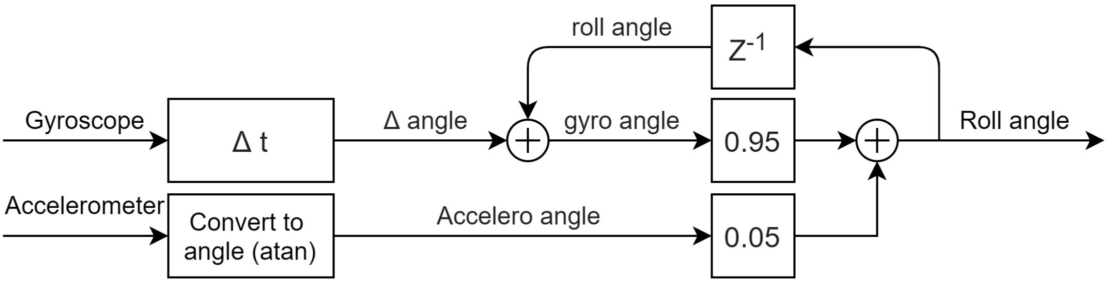


Figure 3.7: Block diagram for summing the gyroscope and accelerometer data using a discrete complementary filter

Rotating around Z axis (Yaw)

For rotating around the Z-axis, the accelerometer is useless because such movements cannot be captured. This due to the way an accelerometer is built up. Flat rotations lead not to any change in acceleration measured by the three individual axes. However, the gyroscope data can measure yaw motion, but again, this measurement drifts over time. Also, tilt compensation is required to obtain absolute angles. A magnetometer could be added to obtain a constant reference point. Also, this magnetometer required tilt-compensation. Both problems can be (partly) bypassed using a function similar to the follow function by *daVinci Surgical Systems* (more details in Section 3.2.3).

Filtering in Arduino IDE

As earlier stated, filtering is required to suppress noise and shaky movements. The filtering cannot be done on the mobile phone (application), which directs the problem to the microcontroller. To apply a filter on the microcontroller, a first-order continuous transfer function (TF) is made (Eq. 3.3) with the designed cutoff frequency. An example with a cutoff frequency of 4 Hz is shown below. Afterward, this TF should be converted to discrete-time with a sample time of 0.01 [s] (see Eq. 3.4). This discrete-time TF can be rewritten to obtain of formula which can be used in the microcontroller shown in Eq. 3.5.

$$H(s) = \frac{\omega_c}{s+\omega_c} = \frac{2\cdot\pi\cdot f_c}{s+2\cdot\pi\cdot f_c} \qquad H(s) = \frac{25.13}{s+25.13} \tag{3.3}$$

$$H(z) = \frac{0.2222}{z-0.7778} \qquad H(Z) = \frac{y[z]}{x[z]} \tag{3.4}$$

$$y[n] = y[n-1]\cdot 0.7778 + 0.2222\cdot x[n] \tag{3.5}$$

3.2.3 Guarantee Precision, Workability and Safety for Radiologist

To guarantee the radiologist to perform a biopsy to the best of their ability, the system has to be minimize delays and includes additional modes.

System Introduced Delays

The elemental requirement for the radiologist for a workable system leads back to the time delay. The wireless communication introduces timing delays. Those delays should under no conditions hinder the radiologist performing a biopsy. The control on the other hand could introduce slower responses due to filtering. Less filtering and therefore faster responses result in more shocking and abrupt movements, which are also not prohibited in a medical application. A trade-off should be made and should be optimized for the best results.

Follow Movement Function

From an user perspective, it is not desirable to always let the robot follow the phone's input. In that case, the user should be focused all the time and cannot move around the patient to look from different angles, because this movement will then be followed by the robot. To overcome this problem, a similar mode of the *daVinci Surgical Systems* is implemented. A foot pedal lets the robot stop follow the movement and enables the radiologist to re-position the joysticks without moving the robot. The robot uses relative positions, rather than absolute positions. A similar function should be integrated into the designed mobile application, only exactly the other way around.

The robot should only follow the movement when a button is pressed and stop when the button is released. This implementation also ensures safety when the phone slips out of the radiologist's hand because the button automatically is released. This button also solves the drift and tilt-compensation described in Section 3.2.2. The effect of drift disappears when only the gyroscope data is integrated when this follow button is pressed. The tilt effect slightly decreases, because the radiologist will naturally re-positioned the phone almost horizontal after releasing the follow button. In a horizontal position, the z-component of the gyroscope aligns with the z-axis in the outer frame.

Resolution Increasing Function

Moving the phone's orientation with small differences is difficult and therefore exactly positioning the robot to a specific position is also problematic. A possible solution is a specific function in the mobile application, such as a *"biopsy mode"*. Outside this mode, *"the normal mode"*, the mapping from phone position to robot position is one-to-one. A change of 1 degree of the phone's orientation results in a change of 1 degree of the robot orientation. However, inside the specific function, a mapping factor is introduced, which changes the one-to-one mapping to for example four-to-one. In that case, a change of 4 degrees of the phone's orientation results in a change of only one degree of the robot orientation. This mapping factor increases the resolution and could be set any integer value and therefore will decrease the possibility of damage to a patient.

Emergency Function

Errors or failures can occur in the system and the robot should reduce the damage to the patient. The user should be able to activate an emergency function in that case. This emergency function could contain several safety features, where freezing is one of the possibilities. Especially when the needle is injected, the position should be held and the needle should be rejected as fast as possible over the same trajectory as injected. Another option is that after activating the function, the needle manually can be rejected while still freezing the position of the rest of the robot.

4 Design

The possibilities and requirements described in Chapter 3 are considered, discussed, and result in the real design. Every choice is discussed and implementation is explained below in this chapter.

4.1 Robot Design

The base, mid-base, and arm described in Section 3.1.1 are combined into a concrete design of the robot, which is shown in Figure 4.1. The goal of the robot is to reach a tumor in 3D space. However, this research has a defined a limited work range -25° to 25° for the roll angle. This range enables the end-effector to insert the needle from different angles, because height positioning is necessary to act as a full biopsy robot. More range is therefore not required. The three robot parts are discussed separately below.

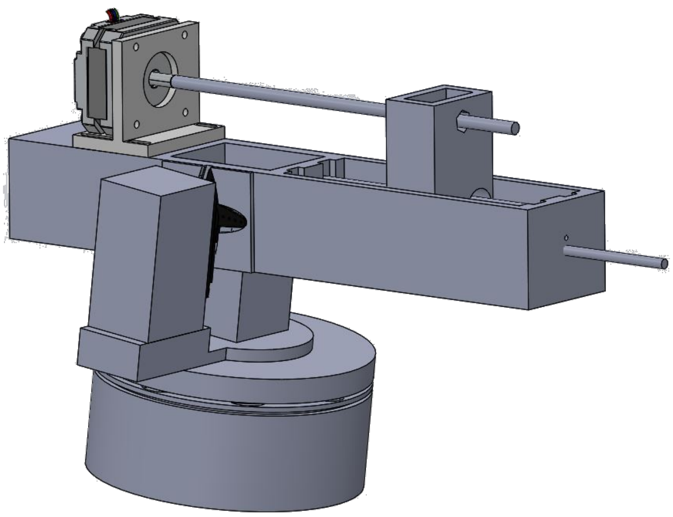
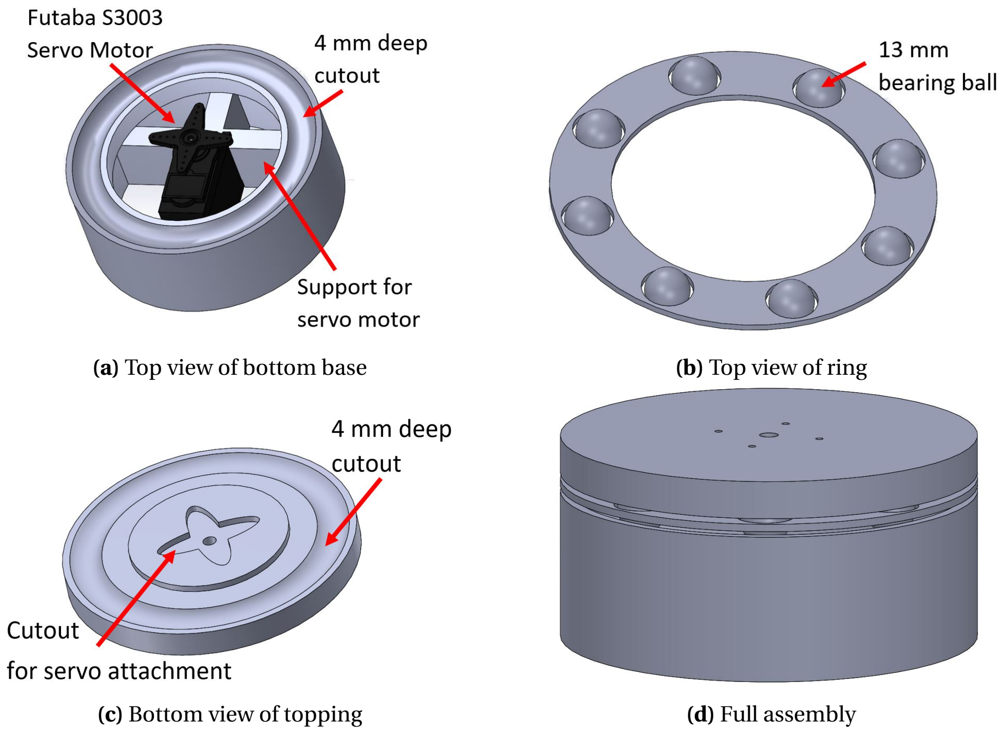


Figure 4.1: Assembly of the final design of the biopsy end-effector

4.1.1 Base

The yaw rotation is integrated into the base of the robot. The thrust bearing is implemented in a 3D printed design because it improves stability and reduces friction. This base consists of three parts, the bottom base, a ring with bearing balls, and a topping. The bottom base, shown in Figure 4.2a, has an inner diameter of 90 mm and an outer diameter of 115 mm. Inside the inner diameter, a servo motor is placed on support for fixing servo to the base. A servo motor, the Futaba s3003, is chosen for its easy control. The estimated torque the servo motor has to deliver equals 38 N-cm. This number is only stated here and details can be found in Appendix A. The Futaba s3003 operating on 6.0 volts can deliver the required torque, because the maximum torque equals 40.9 N-cm at 6.0 volts. The Futaba s3003 is shown in Figure 4.3 and the specifications are shown in Table 4.1.



(a) Top view of bottom base

(b) Top view of ring

(c) Bottom view of topping

(d) Full assembly

Figure 4.2: 3D model of bottom base consisting (a) bottom base, (b) ring with bearing balls, (c) topping and (d) assembly of three parts

Figure 4.3: Futaba s3003. Picture from Tower Hobbies (2020)

Torque [N-cm]	4.8V	31.1
	6.0V	40.9
Speed [$s/60^o$]	4.8V	0.23
	6.0V	0.19
Weight [g]		37.0
Pulse Cycle [ms]		30
Pulse Width [µs]		500-3000

Table 4.1: Specification of Futaba s3003

In the bottom base, a circular cutout is printed of 4 mm deep to ensure bearing balls of 13 mm in diameter staying within the cutout during operation. Eight balls are, after a feasibility study, the lowest possible amount which prevents tilting of the topping. Four balls would also prevent most of the tilting, but would still allow tilting of the topping if pressure is applied on top of the topping between the balls. Increasing the amount to eight reduces the spacing between the balls, which decreases the tilting. Increasing the number of balls to 16 has no significant stability benefit. Those eight bearing balls divide the pressure applied of the topping and are spaced evenly over a ring, shown in Figure 4.2b. The same cutout as in the bottom base is printed inside the bottom of the topping, shown in Figure 4.2c. This enables the bottom base and topping to move parallel with an offset of 5 mm (=13mm - 2x 4mm), as can be seen in the assembly of the three parts in Figure 4.2d. A cutout in the shape of the crosshead of the servo and a hole in the center is made to allow a screw mounted from the other side through the crosshead into the servo motor. This ensures a rigid connection between the servo motor and the topping of the base.

4.1.2 Mid-Base

The roll movement is integrated into the mid-base. This mid-base consists of the mid-base itself, shown in Figure 4.4, and the rotating arm, shown in Figure 4.5. The mid-base consists of a tower and a motor holder, which fit inside each other. This tower ensures that the arm is positioned at a height, which makes roll angles between -25 and 25 degrees (relative from perpendicular start position) possible. A 8 mm hole is drilled into the mid-base, where a 8 mm diameter aluminum tube can fit through. The tube itself, not shown in Figure 4.4, expands the width of the tower on both sides which makes it possible to fit a regular bearing on both sides. Those two bearings should be placed inside the 22 mm holes in the arm. These bearings in combination with the aluminum tube ensure stable rotation. A servo motor, the Futaba s3003, is chosen to drive this roll motion, because of its easy control and weight. The weight is important to reduce the torque to deliver by the other base servo. On one side of the arm, an extra layer is printed with a cutout of the head of the servo motor. This makes a rigid connection between the servo, which is placed inside the motor holder, and the arm. The roll servo has to deliver a torque of 24 N-cm, which fits inside the range of the Futaba s3003. Torque calculation shown in Appendix A. The arm with the needle injection mechanism is balanced to make movement more smooth.

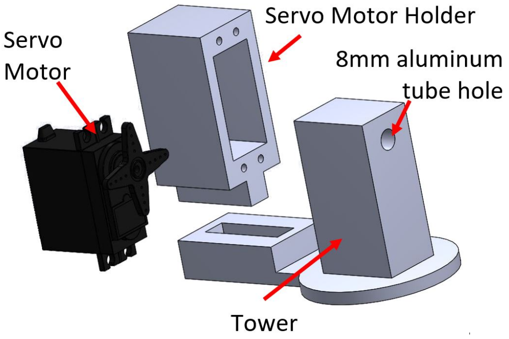


Figure 4.4: 3D model of mid-base

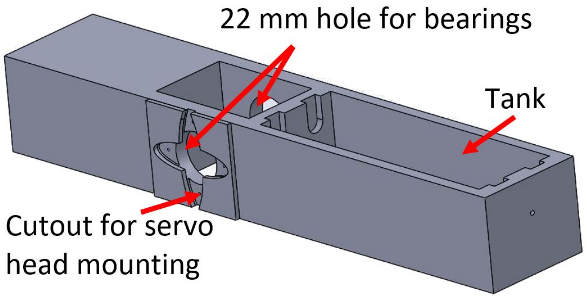


Figure 4.5: 3D model of arm

4.1.3 Arm and Needle Insertion Mechanism

With the arm already stated above, the rest of the design will be explained below. The rest of the arm consists of a tank, which holds the sliding mechanism for the needle injection mechanism. The whole mechanism is shown in Figure 4.6. The screw-thread mechanism with a stepper motor is implemented because a more precise injection is possible compared to the other options. Also, the injection can be more stable. The nut of the screw-thread mechanism is connected to the movable part, the needle box shown in Figure 4.7. In the top of the needle box, a cutout for an M5 nut is made, where an M5 screw-thread can fit. To ensure stability during injection, a sliding mechanism is used containing linear bushes. This principle has already proved stable translational movement in the research of Tanaiutchawoot et al. (2014) (Chapter 2). The available linear bushes have an inner diameter of 8 mm and an outer diameter of 15 mm. The needle box has two holes for the linear bushes. Inside the tank of the arm, four notches (two on both sides) are made where two aluminum tubes of 8 mm diameter can slide into. The linear bushes fit around these tubes. Two tubes (and bushes) are used to restrict the concentric movement when only one of the tubes (and linear bushes) was placed inside the needle box. This needle box furthermore has a hole on the front outside where an external 2 mm diameter biopsy needle can be slid through and be glued in place. This needle is 12 cm long (10 cm outside needle box). To allow the needle be inserted 9 cm, the length_arm in Figure 3.2 equals 160 mm.

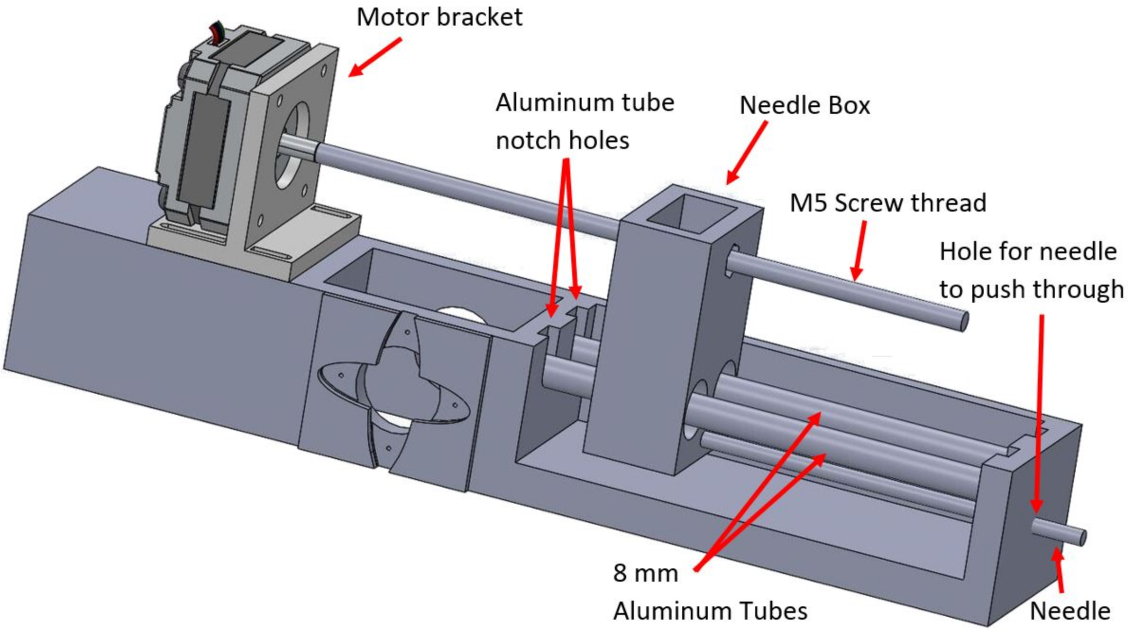


Figure 4.6: Assembly of needle injection mechanism op top of the arm. Note that front side of the tank is made transparent to show the inside of the tank, but is is closed in real design.

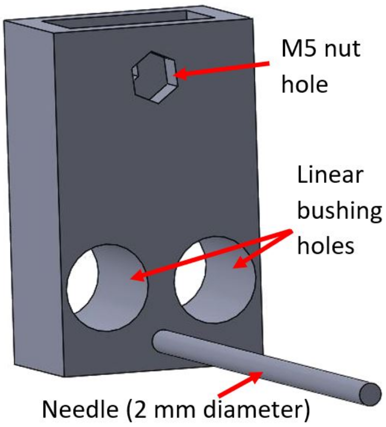


Figure 4.7: Needle Box. Needle is drawn, but in the real design there is only a cutout of 2 mm diameter printed

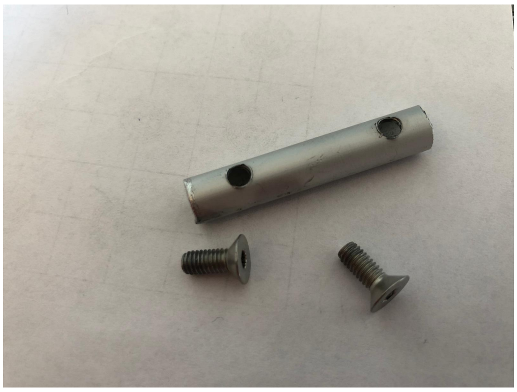


Figure 4.8: Connection shaft between screw-thread and motor shaft, where two bolts clamp the shaft to the tube

To drive the screw-thread, a motor bracket is mounted on top of the arm. This motor bracket has slots to shift the position of the motor a little bit after printing to ensure a good balance. A stepper motor is chosen above a DC motor because the number of steps for the stepper motor can be counted to know the injection depth. The DC motor requires an encoder for this. A servo motor is no possible option, due too its limited rotational range of 180 degrees. Thus a stepper motor can be attached to this motor bracket. The shaft of the motor is connected using a motor coupling, which is a hollow 8 mm tube with two holes with internal thread, shown in Figure 4.8. The required torque the motor has to deliver to move the needle box is difficult to estimate, due to the lack of a detailed simulation model and uncertain parameters, such as friction between the linear bushes and aluminum tubes. The needle box with including M5, needle, and linear bushes only weight 50 grams and therefore the assumption is made that a light stepper motor is able to produce the required torque. The stepper motor used is a Nema17 with a motor length of 27 mm, which specification is shown in Table 4.2. This stepper motor is the lightest stepper motor available, to keep the within the torque boundaries of the earlier two servo motors.

Shaft	Diameter [mm]	5
	Length [mm]	18
Connector		JST
Steps	[per revolution]	200
Holding Torque	[kg-cm]	13.1
Voltage	[V]	4.8
Current	[A]	1.2
Dimensions	[mm]	42x42x27
Weight	[g]	150

Table 4.2: Specifications Nema17 stepper motor

As earlier stated, stepper motors require drivers. A driver for the chosen stepper motor is the single stepper motor driver, the Stepstick DRV8825. This driver can deliver 2.5 amps and features several protection circuits, such as over-current, short circuit, and over-temperature protection.

To ensure the motor does not insert the needle box to deep into a phantom, the combination of hardware and software limitation is used. A physical limitation is the simplest solution in combination with a maximum number of steps to move. The simplicity is decisive because an end switch would have been better. This end switch never creates high currents and high pressures on parts or connections, because the circuit is already disconnected.

4.1.4 Microcontroller & Wiring

Connecting all hardware to the controller is important. The controller itself is the Huzzah ESP32, due to its integrated Bluetooth, dual-core system and higher RAM and Flash memory. To use the Huzzah ESP 32 with the easy to use Arduino IDE, Adafruit provides a setup manual online. To address the different cores, a specific task should be made. This is shown in Listing 4.1. The normal `void loop()` is standard executed on core 0 and therefore only the second core has to be defined, core 1. These different cores are both individually executing all statements straight against each other without waiting for synchronization, due to the `for(;;)` in task 2.

```
TaskHandle_t Task2;

void setup(){
  xTaskCreatePinnedToCore(
        Task2code,   // Task function.
        "Task2",     // name of task.
        10000,       // Stack size of task
        NULL,        // parameter of the task
        1,           // priority of the task
        &Task2,      // Task handle to keep track of created task
        1);          // pin task to core 1
}
void Task2code( void * pvParameters ){
for(;;)
  // code to be executed on core 1.
}
```

Listing 4.1: Create task to perform a task on a specific core of controller

To connect the hardware to the controller, the wiring diagram is shown in Figure 4.9. Also, the power regulation is shown, which makes powering the circuit with only a single 12v adapter possible. Note that an external switch between the 12V adapter and the rest of the circuit is not explicitly shown, but is included inside the 12V power supply block.

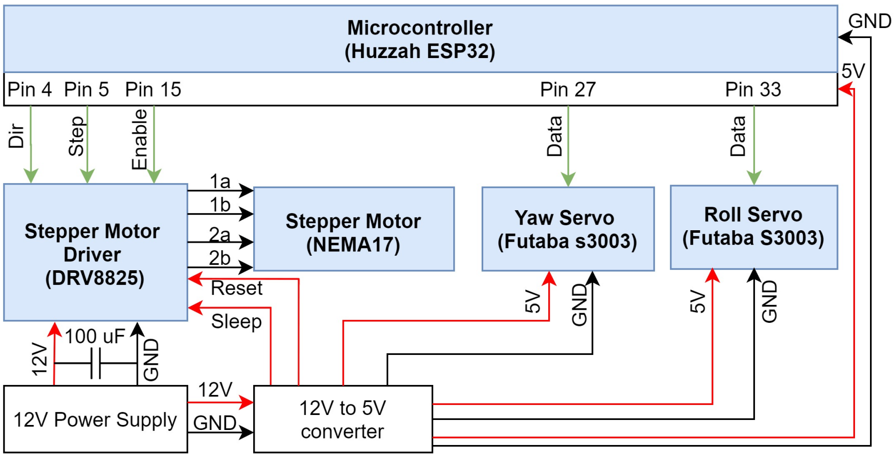


Figure 4.9: Wiring diagram for connecting motors to micro controller and power regulation

4.2 Communication and Control

The mobile application is build using the MIT app inventor software and an overview of all building blocks is shown in Appendix B. Below those blocks will be explained at a higher level. Some modes are defined, which make four states possible; a startup, normal-, biopsy- and emergency mode. To get from one state to another, certain thresholds should be met. These thresholds and modes are shown in Figure 4.10, which can be triggered in the mobile application. One important note is that the flow of data is one-directional, only towards the controller. The application never knows what the controller is doing. The basis of the program is the use of two timers is used. This timer runs every 100 ms (10 Hz) through a list of executions, which is mode dependent. Only one mode at a time is active, so only one timer is running at the same time.

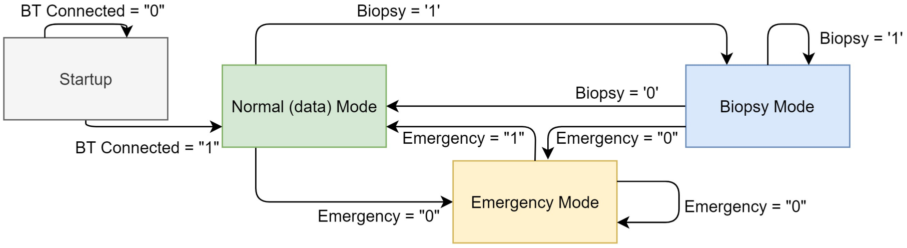
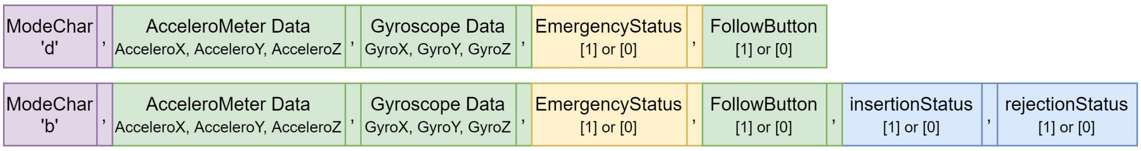


Figure 4.10: State diagram connecting the states and showing threshold to go to other state

4.2.1 Communication

Due too the different modes, not always the same data is sent. The data is sent over a serial connection at a baud rate of 921600. To distinguish between these modes, a status char is sent first, and afterward the data. For the normal mode, the char is a "d" (from 'data') and for the biopsy mode, a "b" (from 'biopsy') is sent. This structure is shown in Figure 4.11 and shows which parameters are sent. The data is sent in once, separated with commas.

The code to receive the data is straightforward. It first checks whether something is sent, afterward checks the char, and writes the data received to the corresponding variables. This code is shown in Listing 4.2.



ModeChar 'd'	,	AcceleroMeter Data AcceleroX, AcceleroY, AcceleroZ	,	Gyroscope Data GyroX, GyroY, GyroZ	,	EmergencyStatus [1] or [0]	,	FollowButton [1] or [0]

ModeChar 'b'	,	AcceleroMeter Data AcceleroX, AcceleroY, AcceleroZ	,	Gyroscope Data GyroX, GyroY, GyroZ	,	EmergencyStatus [1] or [0]	,	FollowButton [1] or [0]	,	insertionStatus [1] or [0]	,	rejectionStatus [1] or [0]

Figure 4.11: Structure of the data send to microcontroller

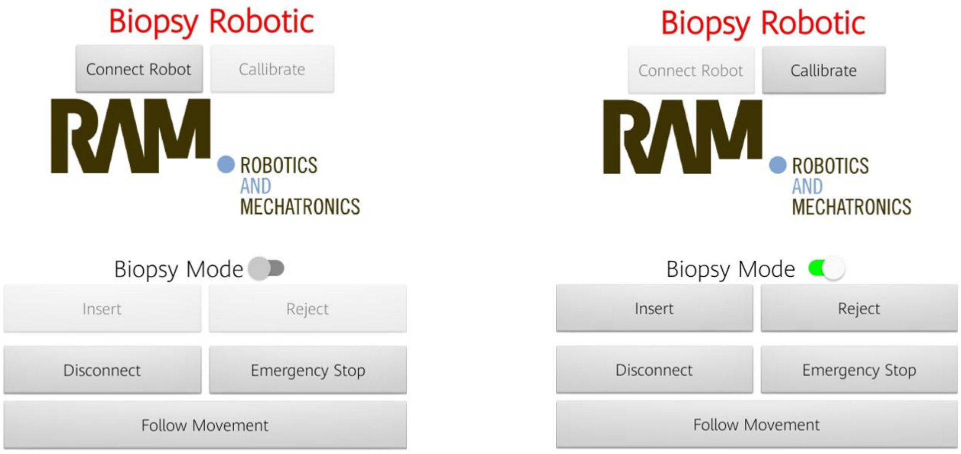
```
#include "BluetoothSerial.h"           // Use correct Bluetooth pins for Huzzah
BluetoothSerial SerialBT;              // Set name to use Bluetooth

if (SerialBT.available() < 1) return; // Wait for nonzero input
getChar = SerialBT.read();             // Read ModeChar to determine datatype

if (getChar == 'd')                    // Selects in which order data to read
   acceleroX = SerialBT.parseFloat(); // Read out float accelerometer X data
   acceleroY = SerialBT.parseFloat(); // Read out float accelerometer Y data
   ...                                 // Read out the intermediate parameters
   FollowButton = SerialBT.parseInt();// Read out last parameter (int FollowButton)
```

Listing 4.2: Code for receiving the data via bluetooth

The processing of the the received data is mode dependent. So therefore every mode is explained below.

4.2.2 Startup

While powering the system, none of the motors receive a control signal. The systems wait for the mobile application to start, which is only possible after successfully connecting and pressing the start button in the application. The startup screen is shown in Figure 4.12a, where the possibility to click a button is regulated making buttons transparent when they should not be pressed. Pressing the start button, after connecting, sets both servo motors in the central position (90 degrees), which ensures the robot to direct always in the same direction as a start point. Afterward, the mobile application writes variable "BT Connected" high, which directs the application to go into normal 'data' mode.



(a) Startup Screen

(b) Biopsy Mode Enabed

Figure 4.12: Screenshots of the graphical user interface showing (a) the startup screen and (b) the biopsy mode enabled screen. The button in both modes which should not be pressed are made partly transparant

After the startup, mode independent, the accelerometer data is processed using the formulas described in Chapter 3, which is shown in listing 4.3. Due to startup problems, or the gravity being along the y-vector, results in an angle of NaN. When this occurs, the angle is forced to zero.

The normal mode and biopsy mode are very similar containing both the "following movement" button, which will first be discussed in detail.

```
// Independent of mode, Calculate Accelerometer Data
RawAccelerometerAngle = atan(acceleroY / sqrt(pow(acceleroX,2) + pow(acceleroZ,2))) *180/ PI;

// Apply Complementary filter and lowpass afterwards depending on the mode
RealRollAngle =((RealRollAngle+gyroIntegratedX)*0.95) + 0.05*RawAccelerometerAngle;

// Problem with NaN solved
if (isnan(RealRollAngle))
    RealRollAngle = 0;
```

Listing 4.3: Processing of Accelerometer data for roll angle

4.2.3 Following Movement Function

The following mode is part of the normal and biopsy mode. The movement of the robot is only possible when the "following movement" button is pressed inside the application. First note the difference between the roll angle, which is absolute, and the yaw angle to be relative. First, the roll mode is explained, and afterward, the difference with the yaw is explained. When this button is not pressed, the last angle with the button pressed is stored and the motors are frozen. The code is shown in Listing 4.4. After converting the angle from degrees to pulse-width, the setpoint is always checked whether it is inside the range of ± 25 degrees from the central point, which is 1250 to 1750 [µs].

```
if (FollowButton < 1)
   RelativeRollAnglePulseWidth = (-RelativeRollAngleDegrees*10)+1500;
   if (RelativeRollAnglePulseWidth > 1250 && RelativeRollAnglePulseWidth < 1750)
        myservoRoll.writeMicroseconds(RelativeRollAnglePulseWidth);

   SetDeltaRollAngle = 0;      // Variable to check whether follow button is released in previous cycle or for a longer time
   if (RelativeYawAnglePulseWidth > 1250 && RelativeYawAnglePulseWidth < 1750)
        myservoYaw.writeMicroseconds(RelativeYawAnglePulseWidth);
```

Listing 4.4: Following movement function implementation when disabled

The first time the "following movement " button is pressed again, the new real phone's orientation is compared to the last saved one and the difference in roll angle is calculated. This difference is only calculated when after the first enabling of the following button and afterward constantly subtracted from the real measured angle. Only when the button is released and again pressed will give a new delta difference. This method ensures one to one movement but allows an offset. In this way, the phone can be moved by the radiologist to a desirable position and the radiologist can move around the phantom to check the needle position. The whole calculation is done using floats of degrees and just before setting the servo angle, it is converted into the pulse-width. This conversion consists of multiplying by ten and apply an offset of 1500 [µs]. This part of the code is shown in Listing 4.5.

```
1  if (FollowButton > 0)
2      // Set only the difference once and not every iteration
3      if (SetDeltaRollAngle < 1)
4          DeltaAngleRoll = RelativeRollAngleDegrees − RealRollAngleFiltered;
5          SetDeltaRollAngle = 1;    // Prevent next iteration to change difference
6
7      // Constantly apply the offset to the real phones orientation
8      RelativeRollAngleDegrees = RealRollAngleFiltered + DeltaAngleRoll;
9
10     // Convert angle to PulseWidth and check validity of setpoint
11     RelativeRollAnglePulseWidth = (−RelativeRollAngleDegrees*10)+1500;
12     if (RelativeRollAnglePulseWidth > 1275 && RelativeRollAnglePulseWidth < 1775)
13         myservoRoll.writeMicroseconds(RelativeRollAnglePulseWidth);
```

Listing 4.5: Following movement function implementation when enabled

As stated above, the yaw angle is controlled slightly differently. By using only the gyroscope for the yaw, the relative angle is measured. Instead of comparing it with the real angle, which is unknown, the angle is only integrated when the follow button is pressed. Adding the magnetometer is therefore not preferable, because it would more code, more data to be transmitted, but do not improves the control of the robot. The same could be done for the roll angle by removing the accelerometer data and create relative angles. However this is not integrated here, due to design development, which required first an absolute roll angle, but afterward, only the relative angle would be already sufficient.

4.2.4 Normal Mode

The normal mode uses the "following movement" function. The filter used in normal mode has a cutoff of 3.5 Hz, which is chosen according to personal desirable speeds. The code for this filtering is shown in Listing 4.6.

```
RealRollAngleFiltered = 0.8*RealRollAngleFiltered + 0.2*RealRollAngle;
yawAngle              = 0.8*yawAngle              + 0.2*gyroAngleZ;
```

Listing 4.6: Low pass filter implementation in normal mode with cutoff of 3.5 Hz

In the normal mode, the biopsy needle cannot move. To move the needle requires a slider inside the application (shown in Figure 4.12) to be enabled, which sets "biopsy" high to go to the biopsy mode.

4.2.5 Biopsy Mode

In biopsy mode, the same principle about the "following movement" function is implemented. However, there are differences. The first difference is the filtering. In biopsy mode, the cutoff frequency is chosen to be higher, to measure small changes better. The frequency is set to 11 Hz. Furthermore, in biopsy mode, the resolution increasing function is implemented, which increases the resolution by four. This function only requires minor changes to the earlier code, shown in Listing 4.5, containing line 8. The adapted code line in biopsy mode is shown in Listing 4.7.

```
RelativeRollAngleDegrees =  PreviousRelativeRollAngleDegrees + DeltaAngleRoll  + (
    RealRollAngleFiltered − PreviousRelativeRollAngleDegrees)/4;
```

Listing 4.7: Change of line eight in Listing 4.5 for biopsy mode

After positioning the needle, the needle needs to be inserted. This is done by checking whether biopsy mode is enabled and whether the injection or rejection button is pressed. When the injection button is pressed, one step will be made and the number of steps is counted. The same holds for the rejection button, but now steps are subtracted from the counted number of steps. When both buttons are pressed simultaneously, the motor will not do anything by default. Also, the max injection length is stated to be 22240 steps, which captures the 9 cm of translational movement. The injection and rejection of the needle are executed on core 1 which may be faster due to fewer tasks to perform compared to core 0, which could lead to multithreading effects, such as writing and reading the same variable at the same time but is not taken into account during design. The code is shown in Listing 4.7.

```
if (biopsyModeStatus > 0)
  digitalWrite(enablePin,LOW);       // Enables current to flow to stepper motor to rotate

  if (insertionStatus > 0 && insertionStatus != rejectionStatus && count < 22240){
    count = count + 1;
    digitalWrite(dirPin,HIGH);
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(SteppermotorSpeedDelay);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(SteppermotorSpeedDelay);

  else if(rejectionStatus > 0 && insertionStatus != rejectionStatus &&  count >= 0){
    count = count - 1;
    digitalWrite(dirPin, LOW);
    digitalWrite(stepPin, HIGH);
    delayMicroseconds(SteppermotorSpeedDelay);
    digitalWrite(stepPin, LOW);
    delayMicroseconds(SteppermotorSpeedDelay);

else
  digitalWrite(enablePin,HIGH);      // Disables current to flow to stepper motor to rotate
```

Listing 4.8: Code to insert or reject the needle by creating PWM signal for stepper motor driver and also count the number of steps taken

4.2.6 Emergency Stop Function

As earlier stated, errors may occur for whatever reason. The choice is made to freeze the motors and reject the needle because disconnecting the needle to reject it manually requires more complex design and does not increase the safety. When in the mobile application the emergency button is pressed, an emergency stop bit (*EmergencyStatusApp*) is transmitted to the controller and another emergency variable (*EmergencyStatusTask2*), is set low (variable is high by default). Before every loop, the controller (on core 0) checks both variables and when one or both are low, the whole biopsy or normal mode is bypassed until both variables returned high again, which is only the case when the needle is fully rejected. This code is shown in Listing 4.9.

```
if (EmergencyStatusApp < 1 || EmergencyStatusTask2 < 1)
    EmergencyStatusApp = 0;
    myservoRoll.writeMicroseconds(RelativeRollAnglePulseWidth);
    myservoYaw.writeMicroseconds(RelativeYawAnglePulseWidth);
    return;
```

Listing 4.9: Code checked by core 0 to check whether emergency from application is received or system is busy rejecting the needle

This *EmergencyStatusTask2* is introduced because after releasing the emergency button in the application, the emergency status would disable without the needle already been fully rejected.

Constantly pressing the emergency button in the application is also not desirable. *EmergencyStatusTask2* is set ’low’ when the emergency button in the application is pressed, but can only return to its original ’high’ state when the injection core (core 1) detects that the needle is fully rejected. Before every iteration of the injection core (core 1), the emergency status sent by the application is checked, and when ’low’, it set the second emergency variable low. Next, it checks the number of steps taken and creates a loop to reject that number of steps. After rejecting the number of steps, the emergency variable two is set ’high’ again, which, in combination with the button released in the application, makes the robot movable again. The code is shown in Listing 4.10.

```
if (EmergencyStatusApp < 1){
    EmergencyStatusTask2 = 0;
    digitalWrite(enablePin,LOW);
    digitalWrite(dirPin, HIGH);
    for (int i = count; i < 0; i++){
        digitalWrite(enablePin,LOW);
        digitalWrite(stepPin, HIGH);
        delayMicroseconds(SteppermotorSpeedDelay);
        digitalWrite(stepPin, LOW);
        delayMicroseconds(SteppermotorSpeedDelay);
    }
    EmergencyStatusTask2 = 1;
}
```

Listing 4.10: Code checked by core 1 to check whether emergency from application is received and start rejection of needle

5 Results: Biopsy Robot Validation

The research aims to hit tumors in 3D space. The designed robot, which should be able to hit a point in 3D space, is tested and the results are shown here. First, the hardware of the robot design, including results from 3D printing and assembling of the different components are validated and afterward discussed. Also, the quality of the motions is discussed. Afterward, when the build accomplishes the requirements, the accuracy of the robot is tested and discussed.

5.1 Build Validation

5.1.1 Method

The robot is assembled as described in Chapter 4. The first check is done by hand, by moving the tip of the robot in space and inspected to see any obvious mistakes. Afterward, a program is made for the controller, which lets the servos first one by one rotate from -25 ° to 25 °, with a speed of 10 and 60 degrees per second, to simulate slow and fast movements. Afterward, both servos operate simultaneously. In all cases, the vibrations of the arms top are visually criticized. Also, the bearing balls are checked if constant pressure is applied on each ball and therefore roll constantly.

The needle injection is tested separately, by setting both servo motors to 90 degrees (initial start point) and by injection the needle at different speeds ranging from 500 to 2000 steps per second using a program.

5.1.2 Results

The printing of the robot turned out good as is shown in Figure 5.1. Due too the slightly wider holes and gaps taking into mind, every part fits perfectly together. During sweeping, moving the arm upwards and downwards occurs with the same magnitude of vibration, after shifting the motor bracket more to the front in the designed slots. After adjusting the screw in the bottom servo, all bearing balls roll smoothly, and applying external pressure by hand between the balls does not tilt the base.

The injection at the speed of 2000 steps per second significantly show more vibration than 500 steps per second. By visual inspection, the speed of 1430 steps per second is the best trade-off between speed and vibration.

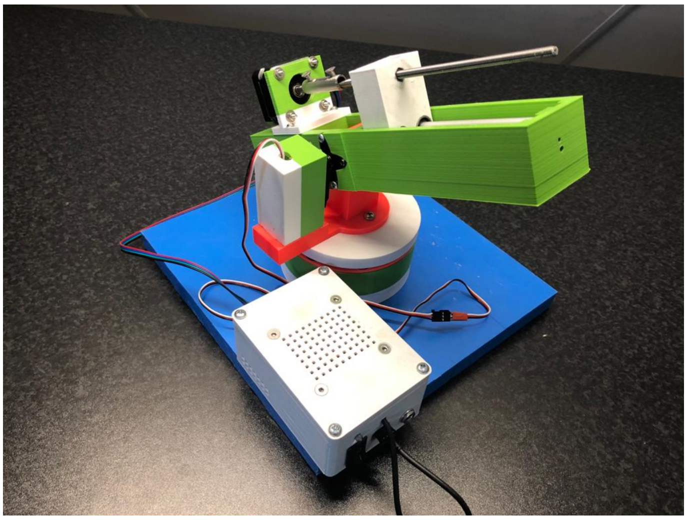


Figure 5.1: Assembly of all (printed) parts

5.1.3 Discussion

Due too the designed slots in the motor bracket, the balance could easily be adjusted. Without balance, the servo for roll movement approaches its maximum torque moving the arm in the upward direction. The bottom servo had less trouble, because of the thrust bearing.

During the injection of the needle box, small movement is still possible between the aluminum tubes and the linear bushes. The needle box is moved by pressing or pulling the nut by the screw-thread. However, this pressure is put onto the top of the needle box, which therefore is tilted slightly forwards or backward, which, due to the tiny movement possible, clamps the linear bushes to the aluminum bars. This clamping results in a choppy insertion, because it looks like the box is moved step by step, rather than continuous.

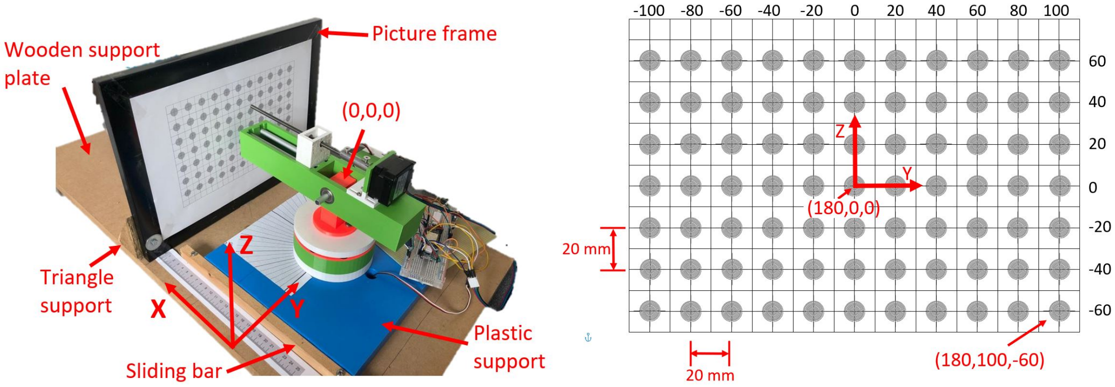
Furthermore, the needle is vibrating constantly speed independent. This is caused by the vibrations produces by the motor, the imbalanced motor shaft, and the screw-thread. These vibrations travel through the component towards the needle, which therefore vibrates. The hole where the needle leaves the robot could be smaller to reduce the vibration. The vibration by the screw-thread is caused by only attaching one side to the motor and let the other side float. Due to too the long-distance (about 10 cm), this enables the screw-thread to swing.

5.2 Accuracy Validation

5.2.1 Method

Hardware setup

The accuracy test is performed with the setup shown in Figure 5.2. As can be noticed, the robot is mounted with two screws in the center of a rectangular piece of blue plastic. This blue plastic is placed onto a wooden support plate. Onto this plate, two wooden sliding bars are screwed which allows the blue plastic with the robot to move over the x-axis, forwards or backward. Perpendicular to the x-axis, a picture frame is mounted parallel to the y-axis with wooden triangle blocks with bolts and nuts. Onto the picture frame, a piece of paper containing targets is taped. Due too the sliding mechanism, the distance from the center of the robot can be initialized and is set using a ruler taped on both sides on the wooden support plate.



(a) Overview of setup

(b) Target on paper with coordinate system (x,y,z)

Figure 5.2: Test setup for measuring accuracy of robot. (a) An overview of setup and (b) target paper

Target paper

The target piece of paper is shown in Figure 5.2b. This paper is constructed out of 77 targets (matrix 11x7), which are spaced 20 mm in the horizontal and vertical direction. The system center point (x,y,z), (0,0,0), is indicated in Figure 5.2a. The robot center is placed 180 mm away from the target paper to position the tip of the arm 20 mm from the picture frame. The picture frame is parallel to the y-axis and therefore all x-coordinates of the target points are 180. The y-coordinates range from -100 to 100 mm and the z-coordinates range from -60 to 60 mm. Each target consists of 10 circles with diameters ranging from 1 to 10 mm. After targeting, the target piece is scanned and the distance between each hit and the corresponding center of the target is measured by using a computer drawing program.

Automated targeting using forward kinematics

The predefined points on the target paper require three variables to reach the point, the angles of the servo motors and the translational depth by the injection mechanism. An excel sheet is used to calculate the required angles using forward kinematics and needle translation to reach the center of each target. To calculate the needle translation, the difference between the robot_radius, defined in Figure 5.3 as the total length from the tip of the needle to the center of rotation, and the length_arm is calculated. This robot_radius, the length to a point in 3D space, is calculated using by taking the square root of the sum of the squared coordinates, shown in Equation 5.1.

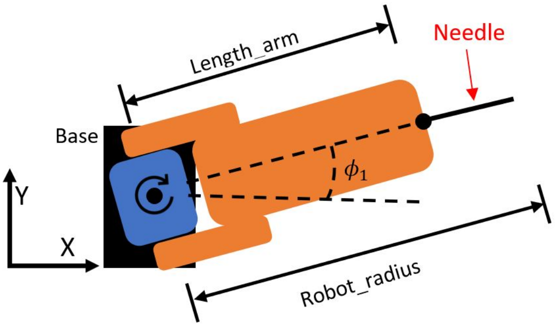


Figure 5.3: Eleborated schematic of Figure 3.2 indication the length of the tip of the needle from the center of rotation as robot_radius

$$\text{robot\_radius} = \sqrt{x^2 + y^2 + z^2} \tag{5.1}$$

The angles of the two servo motors are set using the formulas in Equation 5.2. After calculating the angles of the servo motors, the injection depth is extended with one cm, to push through the paper instead of touching, which would occur with the calculated insertion length. The two angles and the insertion length are combined in a matrix, which is fed into an automated program. This program uses these predefined points and does not use the mobile phone.

$$\begin{aligned} \theta_1 &= \arctan(y/x) \\ \theta_2 &= \arctan(z/\text{robot\_radius}) \end{aligned} \tag{5.2}$$

5.2.2 Results

After targeting all 77 targets, the targeting error of the designed biopsy robot was measured to be 3.25 mm (range from 0.94-6.01 mm) ±1.27 (SD). This distance of 3.25 mm on an x-distance of 180 mm results in an angular accuracy of 1.034 degrees (arctan(3.25/180). In 73 cases (95%), the error was to the left side of the target's center and in 41 cases (53%), the error was below the target's center. In 9 out of 77 cases (11.7 %), the deviation was more than 5 mm. The distribution of the error is shown in Figure 5.4.

5.2.3 Discussion

The angular accuracy of the servo turned out to be 1.034 degrees, which cannot be compared to the data-sheet because its accuracy of the included encoder is not specified. The error may

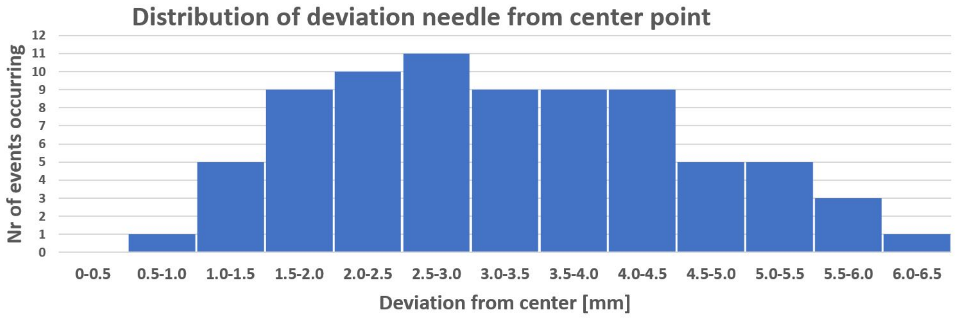


Figure 5.4: Distribution of deviation of accuracy test

partly be a static error because the hit was mostly to the left. This static error could be explained by misplacing the center of rotation at the analytical desired point, stated in Figure 5.2b by (0,0,0). The total error can be explained due to the vibrations of the motor, screw thread, and linear bushes discussed in Section 5.1.3.

Also, the tip of the needle was not sharp enough in combination with the tension in the paper, which resulted in the needle sliding over the paper pointing to wider angles. The sliding causes the tip to penetrate not at the first touch with the paper, but more to the area around it, always more to the outwards direction. This is due to the flat surface of the paper, which could be solved by placing the paper in half a sphere around the robot. This will make the penetration angle always close to 90 degrees. However, this would require new coordinates for the target points. Also, with the available tools, it is hard to construct a sphere surface out of a paper sheet.

Despite all those remarks, the typical breast tumor that needs to be biopsied ranges from 1 to 5 cm, which with the achieved accuracy could be hit successfully in at least 88.3 % of all attempts.

6 Results: Control Validation

In addition to the robot hardware, the control is at least as important. For this control, the built-in sensors of a smartphone, the Huawei P20 Lite, are used, which carry an accelerometer and gyroscope from STMicroelectronics, with corresponding resolutions of 0.00958 [m/s^2] and 1.75·10^{-5} [rad/s]. These sensors are first tested for small and big range variations and discussed. Afterward, the use of these sensors is tested using the introduced functions, the "follow movement", "resolution increasing" and the "emergency stop". Afterward, each function will be evaluated.

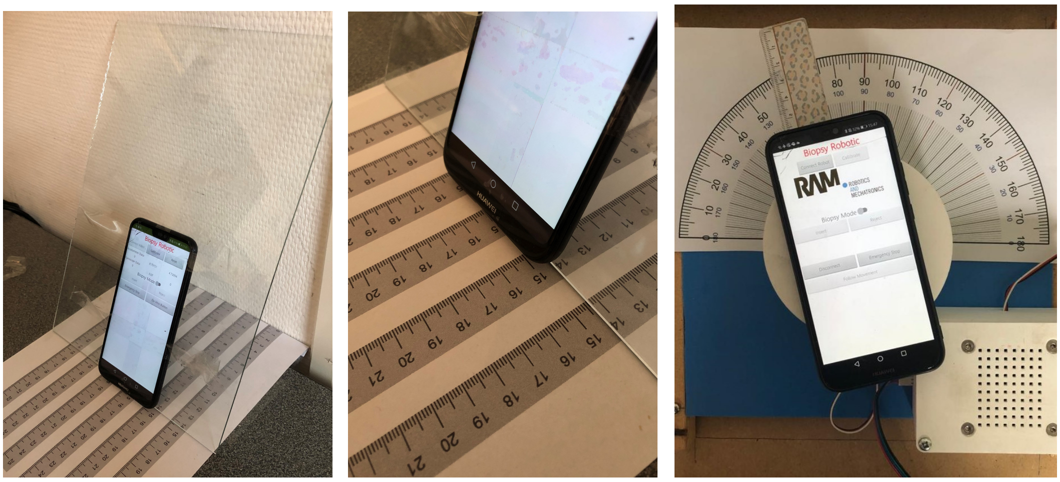
6.1 Sensor Validation

6.1.1 Method

For the validation of the sensors two parts have to be measured, the resolution and the accuracy. To test the accuracy of the roll angle, the mobile phone will be taped to a much bigger glass plate of 295 mm x 210 mm using adhesive tape. A leveled table is provided with measuring tape on the horizontal surface. The setup is shown in Figure 6.1a. This glass plate afterward is placed on eight distances ranging from 100 to 275 mm, corresponding approximately 70 to 21 degrees and the sensor readout is noted. This experiment is performed five times and the average values are used to calculate the average difference from the real angle and average SD.

This experiment is repeated for small angles differences, ranging from 0.39 to 2.33 degrees (in 6 equally spaced steps). In this case, the glass plate was placed flat on a table and glass plates of 2 mm thick were once at a time placed under one side of the glass, to create slightly increased angles. Again the average differences and standard deviations are calculated.

The yaw angle is validated by placing the phone on the trust bearing built in this research, shown in Figure 6.1c. The rest of the robot was not assembled for this validation. A ruler is placed below the phone creating an indication line in the middle of the phone and going through the center of rotation. A big protractor was printed and fitted around the base of the robot. The thrust bearing is manually rotated and the angles are set visually every 15 degrees. Two seconds the position is remained to let the low-pass filter reach the measured angle, which afterwards is noted. Next, the experiment is repeated, but now by stopping at every degree for a range of -5 to 5 degrees to test whether small changes can be observed.



(a) Roll Experiment **(b)** Close-up of Roll Experiment **(c)** Yaw Experiment

Figure 6.1: Test setup for performing (a) roll angle and (c) yaw angle validation. A close-up on the measurement tape for the roll validation is shown in (b).

6.1.2 Results

Roll Angle

For the roll angle, fourteen angles, six in the small range from 0.4 to 2.5 degrees and eight in the big range from 20 degrees to 70 degrees, have been set. The average difference in the small range was 0.06 degrees (SD ± 0.025 degrees) and the big range was 0.30 degrees (SD ± 0.325 degrees).

Yaw Angle

For the yaw angle, after setting 14 angles ranging from -90 degrees to 90 degrees in steps of 15 degrees, the angles angle difference was at average 1.67 degrees and has a range of (0.229 to 2.9 degrees) with an SD of 1.03 degrees. For angles changes about 1 degree, all five cases, the difference in signal could be observed, with an average difference of 0.11 degrees (range -0.18 degrees to 0.04 degrees).

6.1.3 Discussion

Roll Angle

For the roll angle, the data approaches the real angle good enough to detect small changes as well as big changes. The difference can be explained by the glass plate's thickness of 2 mm, which can create differences in the placed angle. Nevertheless, the phone has shown that the roll angle can be measured accurately and with a certain resolution, which allowed to continue to test the complete system.

Yaw Angle

The results for the yaw show that, for the big range, the sensor deviates 1.67 degrees from the setpoint. The difference is mainly due to the rough setting of the angle. Also, the angle is visually directed to the desired angle but may vary a little bit due to the point of view. To decrease the deviation introduced by setting the angle visually using the protractor, the yaw servo could be used in an automated program to set the angles. This would require a validation of the servo motor. All in all, the phone has shown that small changes of 1 degree can be measured and because smaller angles are barely possible to make as a user, this allowed to continue to test the complete system.

6.2 Follow Motion Button

6.2.1 Method

The follow button algorithm for selecting which movements should be followed is implemented as described in Chapter 4 and tested moving the phone constantly and sometimes press the follow button and sometimes releases the button.

6.2.2 Result

The result can be observed in Figure 6.2. Two parts are shown, in the upper part, the roll angle captures by the phone and the setpoint for the motor. In the lower part, the "follow motion mode" status is shown. When the "follow motion mode" is enabled, a "1" is shown, and a "0" is shown when disabled. As can be seen from one to approximately six seconds, the movement is followed adequately. Between 6 and 8.5 seconds, the motor setpoint stays constant, where the actual phone angle decreases. After 8.5 seconds the "follow motion mode" is enabled again, and the motor setpoint follows the actual roll angle again, but now only with an offset.

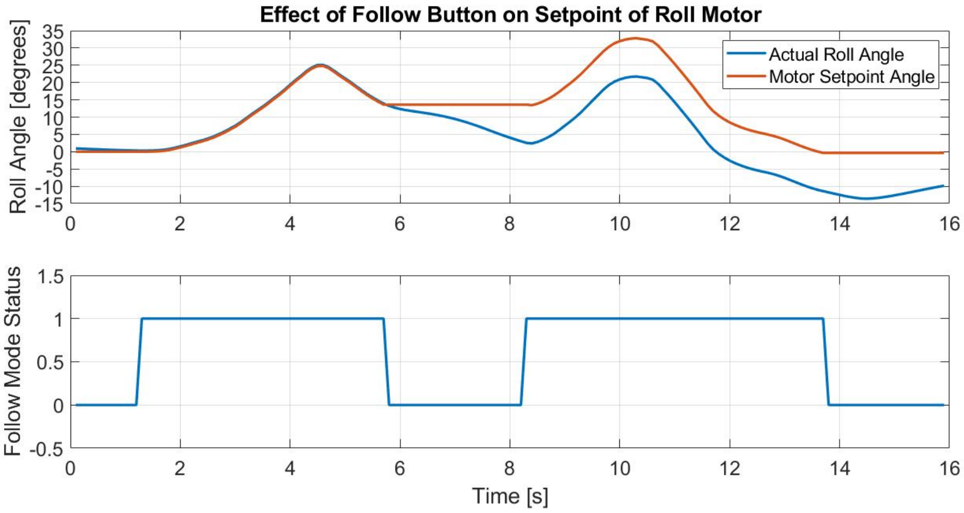


Figure 6.2: Effect of follow button on setpoint of roll motor

6.2.3 Discussion

The overall principle works as designed, but some tiny details need attention. The transition from disabled to enabled follow mode status while moving the phone is not instantly fast. The roll angle has a low-pass filter applied, which needs some time to reach the actual angle, but when the following mode is enabled (after disabling), the filtered roll angle is used to calculate the difference between the robot's angle and real angle. Due too the delay by the low-pass filter, the angle can deviate, which resulted in (small) deviation from real difference and leads to a small shaky robot movement. This problem is very visible when rotating the phone very fast because the differences between the real phones and filtered angles are bigger. For slow and precise movements, the error is much smaller and did not cause visible shaky movement. This could be solved by increasing the cutoff frequency in biopsy mode but would make the robot more sensitive for shaky movements while moving the phone and pressing the button.

6.3 Reduced Motion in Biopsy mode

6.3.1 Method

The algorithm for increasing the movement resolution (by a factor four) is implemented and tested by varying the angle with biopsy mode on and off.

6.3.2 Results

The result is shown in Figure 6.3, which consists of two parts. In the lower part, the enabling ("1") or disabling ("0") of the biopsy mode is shown over time. In the upper part, the real phones roll angle is plotted in blue and the setpoint for the robot is plotted in orange. As can be seen, between 0 and 12.5 seconds, the setpoint angle adequately follows the movement of the phone's roll angle, while the biopsy mode is disabled. An offset can be observed, which is caused by pressing the "follow motion mode" before starting the measurements, but is constant over time. After 12.5 [s], the biopsy mode is enabled and the movements afterward differ from each other. By looking at the values at instant 11 [s], where the mode is disabled, and at the peaks at instant 16.4 [s], their corresponding values measured are shown in Table 6.1. Also, the difference difference in degrees between the angles values at time instant 12.5 [s] and 16.4 [s] are shown. Dividing the difference of the actual angle over the setpoint angle results in 4.051.

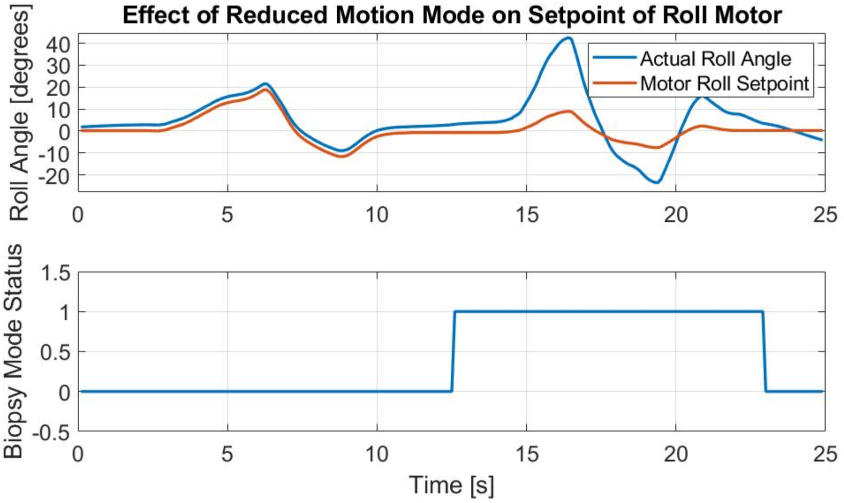


Figure 6.3: Effect of reduced motion mode on setpoint of roll motor

Table 6.1: Data values from time instances in Figure 6.3.

Time Instant [s]	Actual Roll Angle [degrees]	Motor Roll Setpoint [degrees]
12	3.54	-0.68
16.4	42.55	8.95

Difference	39.01	9.63

6.3.3 Discussion

The hard-coded factor 4 deviates slightly from the measured 4.051. This deviation can be explained by the measurement data not being precise enough. Also, a deviation in the deltaAngle may be present, discussed in Section 6.2.3. Nevertheless, the function shows improved resolution and workability of the system. The hard-coded factor could be adapted regarding test experiences by experts.

6.4 Emergency Stop Mode

6.4.1 Method

The emergency stop is implemented and tested as described in Chapter 4. The robot is moved to an arbitrary position and the needle is injected. With the needle partly injected, the emergency button is pressed inside the application. The reaction of the robot is visualized.

6.4.2 Results

By positioning, injecting the needle and afterward pressing the button makes the robot uncontrollable. The motors are instantly frozen and no reaction from inputs from application are visible. The needle is rejected slowly, 5cm in 8 seconds (1430 steps per second), and the same vibration is observed as the experiment described in Section 5.1. Afterward, pressing a button and moving the phone made the robot movable again.

6.4.3 Discussion

The goal of the function is to reduce damage to the phantom. The rejection speed is stable and vibrations are visible, as discussed in Section 5.1.3. The speed could be increased but would result in more vibration and more damage. At a lower rejection speed, the vibrations do not decrease significantly and would create a longer nervous moment for the patient and surgeon.

7 Results: Targeting Lesions inside Transparent Phantom

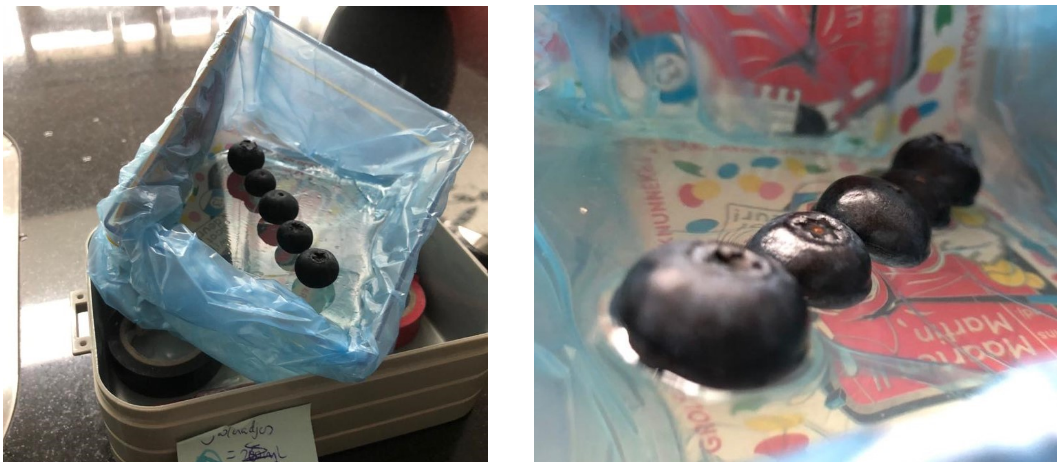
This research aims to be able to use a mobile device to hit tumors with a biopsy robot. The results of this test are shown here. A medical phantom is not available at the test location. Therefore, first, a representative phantom with tumors will be made. Afterward, the robot will be used to hit those tumors.

7.1 Production Of Transparent Phantom

7.1.1 Method

Having no MRI or ultrasound system at the location available, a solution is making a transparent phantom. In that way, visual feedback is used to determine whether the needle has hit (or missed) a tumor. To simulate a typical breast tumor which need to be biopsied, which is one to five cm in diameter, blueberries are used because they are circular, hard but still pierceable. Five blueberries are selected with a diameter of approximately 1 cm because these are the smallest possible breast tumors. These 5 blueberries are placed inside a 10x10x10 cm gelatin block. Gelatin has the feature to be transparent to ensure that the lateral offset of the needle in blueberry can be estimated. By placing the 5 blueberries approximately equally spaced on the inner diagonal makes every blueberry visible from each side. In this way, it can be verified whether the needle is inside the tumor and by rotating four times a quarter turn, 20 biopsies can be performed.

A cube mold, with a plastic bag inside, is rotated 45 degrees in two directions to level the diagonal of the cube parallel to the horizontal surface. This procedure can be seen in Figure 7.1a. Gelatin is added up to the diagonal. This gelatin is solidified and afterward, the five blueberries are roughly positioned equally over the diagonal. A little bit gelatin is added to place the blueberries partly (≈ 50%) inside this layer, to prevent them from floating, which would occur by directly filling the rest of the cube once. This step can be seen in Figure 7.1b. After this small layer is also solidified, the cube has taken out of the setup and placed flat on a horizontal surface. Afterward, the rest of the gelatin is added to complete the cube. The real positions are afterwards measured with measuring tape.



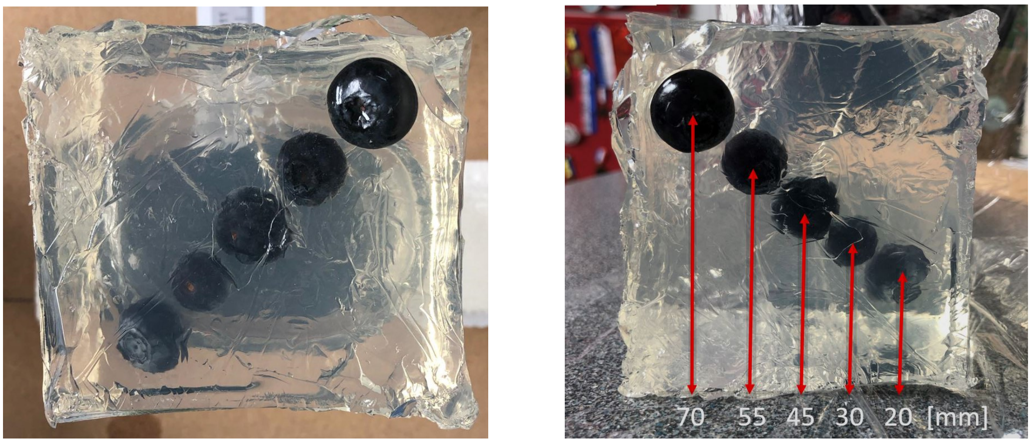
(a) Setup overview

(b) Close-up during production of phantom

Figure 7.1: Setup for producing a gelatin phantom. (a) Setup to level liquid parallel to flat surface and (b) close up after insertion the blueberries and adding a small amount to cover the blueberry partly in gelatin

7.1.2 Results

The production has been performed and resulted in Figure 7.2. The space between the tumors is enough to see each blueberry from each side without an overlay. The edges are very rough and not making a square cube. The transparency is high enough to see each blueberry, but the blueberry furthest away from the edge is already vaguely visible. The blueberries are located as noted in Figure 7.2b, where the distance measured by measurement tape from the edge to the origin of the blueberry was 20, 30, 45, 55 and 70 mm (± 2 mm).



(a) Top View

(b) Side View with measurements

Figure 7.2: 10x10x10 cm gelatin phantom with five blueberries as breast tumors

7.1.3 Discussion

The produced phantom is not as transparent as expected. Straight out of the mold, the transparency was even worst, which is caused by the wrinkles of the plastic bag inside the mold. The transparency was increased by trimming down all edges using a large sharp knife. These impurities caused by the wrinkles created lots of light beam distortion. After trimming, still, no square cube was achieved, but the transparency and positioning of the tumors had accomplished the necessities to continue testing. Due to stacking layers of gelatin on each other resulted in visible planes, which did not stick properly to the other layers. The blueberries positioned on the diagonal keeps the two parts connected, but the two (neglecting the small layer) can quite easily be separated by hand.

7.2 Targeting Blueberries in Gelatin

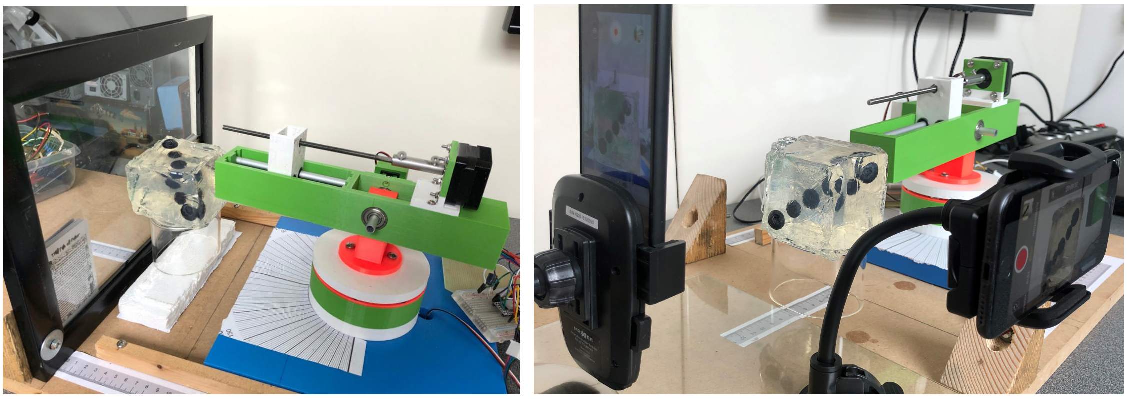
All individual parts are produced and successfully tested before starting performing 20 biopsies. Out of the earlier experiment of the robot's accuracy in Section 5.2, the average accuracy was measured to be 3.25 mm (range 0.94-6.01 mm), converted an angular accuracy of 1.034 degrees. In 11.7% of the cases, the error was more than 5 mm, which would result in a miss. Therefore the expected number of hits is 18 out of 20 (88.3%).

7.2.1 Method

The targeting will be done placing the middle blueberry inside the phantom straight in line with the biopsy robot, which is the same location as target (180,0,0) earlier shown in Figure 5.2. The gelatin phantom needs some support in the form of a plastic tray to level the robot at the desired height. On the left side and in front of the phantom, two cameras are placed to verify the needle's position. This setup can be seen in Figure 7.3. For every tumor, as many attempts are executed until the tumor has been hit. Every injection is measured and the number of attempts to get a hit is noted. The observation of whether a hit or miss occurred is done visually. Afterward, the phantom is rotated 90 degrees to redo the experiment from another side.

This experiment is first executed with only the plastic tray support but afterward redone on a new phantom with an extra glass plate placed against the tumor from the back. A new phantom is made to have no injection routes from earlier injections present to have an equal chance of hitting. This experiment is done to test the influence of deformation during injecting.

For this experiment, the blueberries in the produced transparent phantom (shown in Figure 7.2b) are defined from 1 to 5. Blueberry 1 is the less deep (20 mm) tumor in the phantom and blueberry 5 is the deepest (70 mm) tumor in the phantom. The four biopsy sides of the phantom are defined as the front, right side, left side, and the back. From the top and bottom, no injection is performed because these sides have the most transparency, which allowed to visual observations.



(a) Setup with glass plate

(b) Setup without glass plate

Figure 7.3: Setup for targeting lesions and measuring hit-percentage. (a) Setup with glass plate as back support, cameras not shown and (b) setup without glass plate as back support, showing camera positioning

7.2.2 Results

Without glass support

Out of 20 biopsies without glass support targeting only one attempt per tumor resulted in 15 hits and 5 misses, which is a hit-percentage of 75%. The result is shown in Table 7.1. After a second attempt, 18 hits were made. The misses are the deepest blueberries, 70 mm deep. The last two were hit after four attempts. The damage done to the gelatin was neglectable and no rips are made during the biopsy, which can be observerd in the supported video.

		Blueberry				
		1	2	3	4	5
Biopsy side	Front	X	X	X	X	X
	Right	X	X	X	X	O
	Bottom	X	X	X	O	O
	Left	X	X	X	O	O

(a) First attempt

		Blueberry				
		1	2	3	4	5
Biopsy side	Front	X	X	X	X	X
	Right	X	X	X	X	X
	Bottom	X	X	X	X	O
	Left	X	X	X	X	O

(b) Second attempt

Table 7.1: Result from targeting 5 blueberries from four sides without glass support. Showing hits, indicated with a cross, and misses, indicated with a circle of the (a) first attempt and (b) second attempt

With glass support

After placing the glass plate behind the phantom, the experiment was redone. Out of 20 biopsies with glass support targeting only one attempt per tumor resulted in 17 hits and 3 misses, which is a hit-percentage of 85%. After a second attempt, 20 hits were made.

		Blueberry				
		1	2	3	4	5
Biopsy side	1	X	X	X	X	X
	2	X	X	X	X	X
	3	X	X	X	X	O
	4	X	X	X	O	O

(**a**) First attempt

		Blueberry				
		1	2	3	4	5
Biopsy side	1	X	X	X	X	X
	2	X	X	X	X	X
	3	X	X	X	X	X
	4	X	X	X	X	X

(**b**) Second attempt

Table 7.2: Result from targeting 5 blueberries from four sides with glass support. Showing hits and misses of (a) attempt one and (b) attempt two

7.2.3 Discussion

The expected hit-percentage of 88% is in both cases not met, but the experiment with glass support approaches the expected value. The glass experiment had a hit-percentage of 85%. The deformation, which has been taken into account, indeed caused problems. While injecting the needle to the two deepest blueberries (55 and 70 mm), the phantom is pushed away, and because only attachment to the bottom, it is deformed by tilting away from the robot. This tilting results in the blueberries being pushed backwards and downwards after starting injection. This causes misses, while the robot was aligned for a hit. This explains the two fewer hits without a glass plate compared to the experiment with a glass plate.

Adding the glass plate increased the hit-percentage. The miss could also be the effect of the camera as a viewpoint, which slightly differs from a view perpendicular to the edge. The human effect on misses is also significant. Working with the system quickly learns the user how to work with it and increases the chances of hitting tumors. The needle could also be not perfectly straight, because its a small in diameter piece of metal, and during assembling and testing it could be deformed a little bit. The rest of the misses can be explained by the vibrations, as earlier discussed in Section 5.1.3. The vibration was mainly visible without using the glass plate support. Although these vibrations, no negative effect on the phantom is observed. The vibrations of the needle did make the phantom vibrate but did not lead to rips in the phantom. The vibration was significantly reduced after attaching the back glass plate but is not an applicable solution for real biopsies.

The multithreading was not taken into account during designing, but also did not lead to any problem during targeting lesions. The controller may have built-in regulation software to neglect one or both, such as read and write of the same variable when both are requested at the same time. A clear confirmation of this presumption is not found. If that is the case, the effect is reasonably small, because the update rate of the sensors is 10 Hz. One measurement missing is not directly catastrophic.

8 Conclusions and recommendations

8.1 Conclusion

In this bachelor thesis, a 3 DOFs biopsy end-effector is designed and controlled using a smartphone application. The use of two low-cost servo motors enabled the robot, integrating a 3D printed thrust bearing, to rotate stable and position a needle injection mechanism, a screw-thread actuator including a stepper motor. This injection mechanism moves a needle box with a needle attached in a straight line using linear bushes and can reach tumors up to 9 cm in-depth, where afterward its physically limited. The robot has been tested targeting 77 predefined setpoints. An average accuracy of 3.25 mm is achieved, equals an angular accurcy of 1.034 degrees. The control is performed using a Bluetooth integrated microcontroller, which communicates to a phone application. The application offers different control modes, which enables the user to perform a safe and accurate biopsy. The robotic system is tested targeting 20 tumors in a transparent phantom under visual guidance, where 17 tumors were hit the first attempt. Using the robotic system developed in this research, it is possible to control a biopsy needle using a handheld mobile phone, and target lesions sized 1 cm and up to 9 cm deep with 83% hit-percentage under visual guidance.

8.2 Recommendations

Although the system already showed its capabilities, the system should be tested intensively by radiologists (and other users) to gain feedback to improve the system.

For the 3D printed robot, the vibration often stated in this paper is a problem. The sliding of the needle box on the two parallel rods still allowed tilting, which enables out of plane movements by the needle box. Support for the end of the screw thread or slightly bigger aluminum rods should be included to suppress this vibration. Also, the connector between the motor and screw-thread should be rotational balanced and lighter, which is not the case in the current design. An aluminum shaft is therefore preferable.

The robots appears not medical. A more rounded and elegant design is desirable and also a closed design would be a step ahead, where the rough screw thread and motor are not visible. The cable management has less priority in the design process but requires some attention for further research. The motors have to deal with quite some load, so a more efficient and lighter design increases performance.

For the needle box itself, it would be beneficial if the point which pushes or pull would be at the same vertical level as the guidance rods. The screw-thread should than go through the mid-base. As stated, the used injection mechanism had not the desired stability, which was the reason. The current injection mechanism also includes much volume. Therefore it could be replaced by the other mechanism stated in this paper containing a rotational motor with gear attached to drive a small notched piece of plastic with needle inside, shown in Figure 3.4. A stepper motor could be used. The main benefits is the drastically reduced volume and only a motor straight above the center of rotation is required, which lowers the weight of the total "arm" and lowers the required torque of the other motors. Also using the stepper motor, still precise insertion is managed. This mechanism is also able to reject a needle much faster than the screw-thread mechanism. The injection and rejection stability is uncertain and needs to be evaluated to test if this increases the performance.

The performance is determined by the type of motor. The Futaba s3003 servo motor had a specific accuracy of approximately one degree, which can be improved by using a more accurate and stronger servo. For example, the Herkulex DRS-0201, provides a resolution of 0.325°, a weight of 60 grams and 235 N-cm torque, compared to the 1.034°, 37 gram and 41 N-cm of the Futaba s3003. The improved performance comes on the cost of increased weight and costs. Applying a stepper motor with micro-stepping possibilities (up to 1/32 of a step size, which is often 1.8 degree) could theoretical have a resolution of 0.05625°. This performance comes with the costs of weight. Another option to increase performance is by using gears. In the Futaba s3003, there is already a small gear build in. The motor therefore could be replaced by a DC motor (or stepper motor), with a selection of gears. An external encoder should be implemented to keep track of the angle. Gears also may have backlash, except when using toothed belts. Inside the base, these gears could easily be fit, but for the roll motor, this would be more complex.

To approach the functionalities of other biopsy robots, the implementation of a real biopsy needle is required. The range of 9 cm should also be increased to about 15 cm to match other robots. Also, an extensive comparison with other robots should be done in future research.

The update rate of the sensor data is now 10 Hz. In combination with the selected filter lengths, the delay is acceptable but less delay is preferred. The update rate could in the current setup not much increased, because the number of executions by the application inside the timer loop is simply not faster. The number of executions could be reduced by omitting the accelerometer data. Besides, the main part is the optimization of the filters. The implementation of a PID controller could increase performance.

A Torque Calculation

For all calculations the densities for the material are taken as follows, steel equals 7.85 g/cm^3, aluminum = 2.70 g/cm^3 and 3D printed material equals water, which is 1 g/cm^3. For all parts with the center of mass at the center of rotation, the part ils spitted up into several parts, with each individual center of mass and spacing from the center of origin. Although the 3D printed parts are often 25% hollow with 1.2 mm thick walls, the calculation is based on solid objects, which gives a little too pessimistic estimation of the required torque.

1. Bottom base
The bit round plate has a diameter of 110 mm and is 5 mm thick, the centre of mass is at the center of rotation. Approximating each quarter plate to have a weight of 12 grams (total is 47.5 gram) results in a force of 0.12 Newton. By multiplying with the average radius of 27.5 mm (110mm/2/2) and by multiplying by four results in a torque of 0.33 N-cm.

2. Midbase
The midbase is attached to the bottom base via a round place and a rectangle. This round plate has a diameter of 60 mm and the rectangle is 60mm x 30 mm. Both parts have a height of 10 mm. By dividing the round plate in four pieces results in a torque of 0.42 N-cm. The rectangular plate has 18 grams at 4.5 cm from center. This results in a torque of 0.81 N-cm. In total is this 1.23 N-cm.

On top of this plate, the base for on which the arm is attached is placed. The dimensions of this base are 33x25x60 mm. By again dividing in four parts results in a torque of 1.4 N-cm.

The motor holder base has dimensions of 30x35x80 mm, which results in a volume of 84 cm^3. There is a hole made in it with dimensions 20x41x28 mm, which results in 23 cm^3. The weight therefore is the difference between the two, which is 50 grams. The servo motor placed in the gab has a weight of 37 grams, which make a total of 87 grams. This base with motor is located 60 mm from the center of rotation and therefore the torque is 5.22 N-cm.

One hollow aluminum tube of 8 mm diameter (1mm thick) with a length of 50 mm has a weight of 1.1 grams, which equals 0.01 N-cm.

3. Arm
A stepper motor is located 2 cm from the center of axis, which has a weight of 150 grams. The motor bracket is also located at that position. The motor bracket has a basis of 37.5x39x5 mm and a 5x49x42 mm upstanding wall, which makes a weight of 17.6 gram. This 167.6 gram result in a torque of 3.35 N-cm.

The arm itself is 259x49x40 mm, which equals 506 gram. There are two gabs, one of 45x35x40 mm and one of 45x124x30 mm. The weight therefore equals 275 gram. The center of mass is balanced to be at the center of rotation, which makes a torque of 0 N-cm.

The two hollow aluminum tubes are 8 mm thick (1 mm thick) and 110 mm each long. This results in a weight of 4.8 grams, which are positioned 110 mm from the center of rotation which creates a torque of makes 0.53 N-cm.

The two linear bushes have a weight of 16 grams each located 17 cm from the center, which makes a torque of 2.72 N-cm. The two bearings are 50 grams together. The two bearings are 50 grams together, which make a torque of 0.35 N-cm taking friction not into mind. This therefore sums up to 3.07 N-cm.

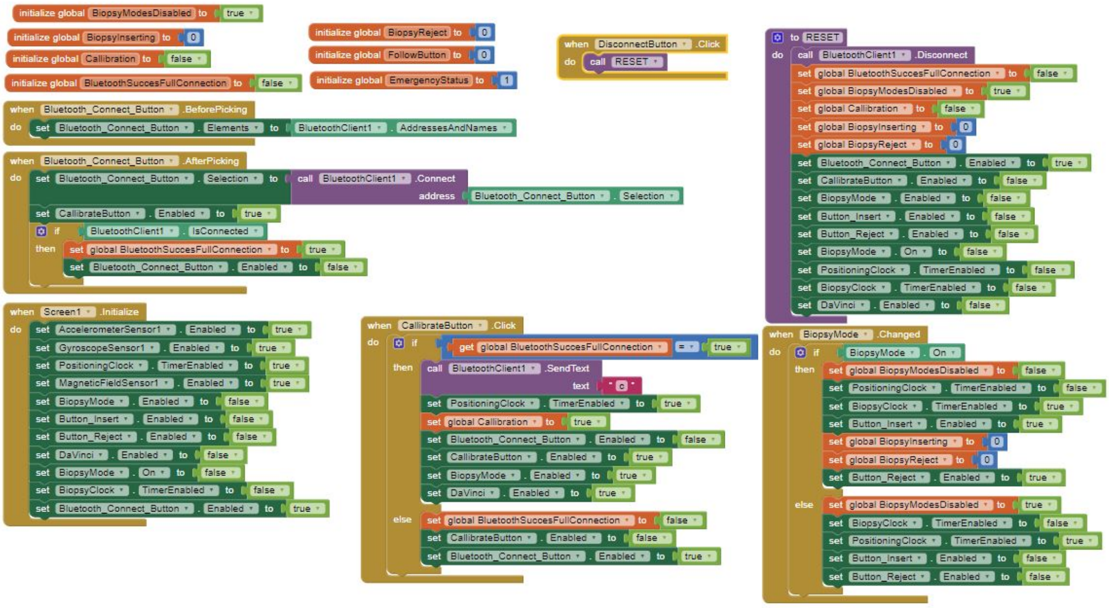
The needle box is 41x24x50 mm, with a gab of 31x14x50, which makes a weight of 43.5 gram, which is located 160 mm from the center of rotation. This makes 6.9 N-cm.

The needle itself is 2 mm in diameter and 12 cm length, which makes 12 grams at 220 mm from the center of rotation, which makes 2.64 N-cm.

The M5 screw thread has a diameter of 5 mm, a length of 12 cm, which makes 73.6 grams. On a average distance of 6 cm from the center of rotation, this makes 4.4 N-cm.

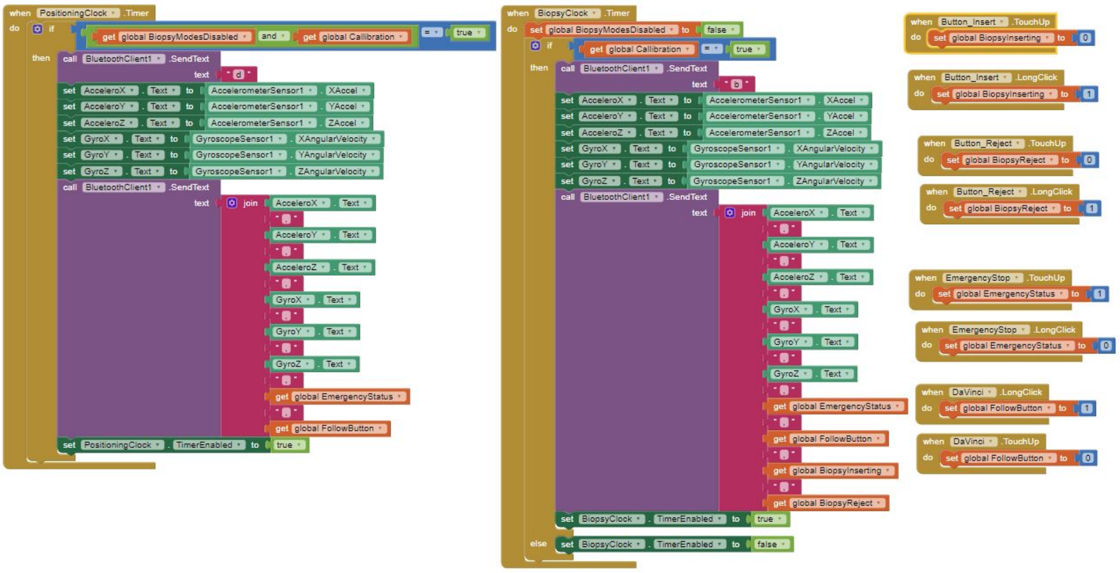
This sums up to 38 N-cm to be driven by the base Servo and 14.75 N-cm be driven by the arm Servo. However this calculation is done for all parts being at maximum distance from the center, which in practice is not the case.

B MIT App Inventor Blocks



```
initialize global BiopsyModesDisabled to true
initialize global BiopsyInserting to 0
initialize global Callibration to false
initialize global BluetoothSuccesFullConnection to false

initialize global BiopsyReject to 0
initialize global FollowButton to 0
initialize global EmergencyStatus to 1

when Bluetooth_Connect_Button .BeforePicking
do  set Bluetooth_Connect_Button . Elements to BluetoothClient1 . AddressesAndNames

when Bluetooth_Connect_Button .AfterPicking
do  set Bluetooth_Connect_Button . Selection to call BluetoothClient1 .Connect
                                                              address Bluetooth_Connect_Button . Selection
    set CallibrateButton . Enabled to true
    if BluetoothClient1 . IsConnected
    then set global BluetoothSuccesFullConnection to true
         set Bluetooth_Connect_Button . Enabled to false

when Screen1 .Initialize
do  set AccelerometerSensor1 . Enabled to true
    set GyroscopeSensor1 . Enabled to true
    set PositioningClock . TimerEnabled to true
    set MagneticFieldSensor1 . Enabled to true
    set BiopsyMode . Enabled to false
    set Button_Insert . Enabled to false
    set Button_Reject . Enabled to false
    set DaVinci . Enabled to false
    set BiopsyMode . On to false
    set BiopsyClock . TimerEnabled to false
    set Bluetooth_Connect_Button . Enabled to true

when DisconnectButton .Click
do  call RESET

when CallibrateButton .Click
do  if get global BluetoothSuccesFullConnection = true
    then call BluetoothClient1 .SendText
                              text " c "
         set PositioningClock . TimerEnabled to true
         set global Callibration to true
         set Bluetooth_Connect_Button . Enabled to false
         set CallibrateButton . Enabled to true
         set BiopsyMode . Enabled to true
         set DaVinci . Enabled to true
    else set global BluetoothSuccesFullConnection to false
         set CallibrateButton . Enabled to false
         set Bluetooth_Connect_Button . Enabled to true

to RESET
do  call BluetoothClient1 .Disconnect
    set global BluetoothSuccesFullConnection to false
    set global BiopsyModesDisabled to true
    set global Callibration to false
    set global BiopsyInserting to 0
    set global BiopsyReject to 0
    set Bluetooth_Connect_Button . Enabled to true
    set CallibrateButton . Enabled to false
    set BiopsyMode . Enabled to false
    set Button_Insert . Enabled to false
    set Button_Reject . Enabled to false
    set BiopsyMode . On to false
    set PositioningClock . TimerEnabled to false
    set BiopsyClock . TimerEnabled to false
    set DaVinci . Enabled to false

when BiopsyMode .Changed
do  if BiopsyMode . On
    then set global BiopsyModesDisabled to false
         set PositioningClock . TimerEnabled to false
         set BiopsyClock . TimerEnabled to true
         set Button_Insert . Enabled to true
         set global BiopsyInserting to 0
         set global BiopsyReject to 0
         set Button_Reject . Enabled to true
    else set global BiopsyModesDisabled to true
         set BiopsyClock . TimerEnabled to false
         set PositioningClock . TimerEnabled to true
         set Button_Insert . Enabled to false
         set Button_Reject . Enabled to false
```

Figure B.1



```
when PositioningClock .Timer
do  if get global BiopsyModesDisabled and get global Callibration = true
    then call BluetoothClient1 .SendText
                              text " d "
         set AcceleroX . Text to AccelerometerSensor1 . XAccel
         set AcceleroY . Text to AccelerometerSensor1 . YAccel
         set AcceleroZ . Text to AccelerometerSensor1 . ZAccel
         set GyroX . Text to GyroscopeSensor1 . XAngularVelocity
         set GyroY . Text to GyroscopeSensor1 . YAngularVelocity
         set GyroZ . Text to GyroscopeSensor1 . ZAngularVelocity
         call BluetoothClient1 .SendText
                              text join AcceleroX . Text
                                        " , "
                                        AcceleroY . Text
                                        " , "
                                        AcceleroZ . Text
                                        " , "
                                        GyroX . Text
                                        " , "
                                        GyroY . Text
                                        " , "
                                        GyroZ . Text
                                        " , "
                                        get global EmergencyStatus
                                        " , "
                                        get global FollowButton
         set PositioningClock . TimerEnabled to true

when BiopsyClock .Timer
do  set global BiopsyModesDisabled to false
    if get global Callibration = true
    then call BluetoothClient1 .SendText
                              text " b "
         set AcceleroX . Text to AccelerometerSensor1 . XAccel
         set AcceleroY . Text to AccelerometerSensor1 . YAccel
         set AcceleroZ . Text to AccelerometerSensor1 . ZAccel
         set GyroX . Text to GyroscopeSensor1 . XAngularVelocity
         set GyroY . Text to GyroscopeSensor1 . YAngularVelocity
         set GyroZ . Text to GyroscopeSensor1 . ZAngularVelocity
         call BluetoothClient1 .SendText
                              text join AcceleroX . Text
                                        " , "
                                        AcceleroY . Text
                                        " , "
                                        AcceleroZ . Text
                                        " , "
                                        GyroX . Text
                                        " , "
                                        GyroY . Text
                                        " , "
                                        GyroZ . Text
                                        " , "
                                        get global EmergencyStatus
                                        " , "
                                        get global FollowButton
                                        " , "
                                        get global BiopsyInserting
                                        " , "
                                        get global BiopsyReject
         set BiopsyClock . TimerEnabled to true
    else set BiopsyClock . TimerEnabled to false

when Button_Insert .TouchUp
do  set global BiopsyInserting to 0

when Button_Insert .LongClick
do  set global BiopsyInserting to 1

when Button_Reject .TouchUp
do  set global BiopsyReject to 0

when Button_Reject .LongClick
do  set global BiopsyReject to 1

when EmergencyStop .TouchUp
do  set global EmergencyStatus to 1

when EmergencyStop .LongClick
do  set global EmergencyStatus to 0

when DaVinci .LongClick
do  set global FollowButton to 1

when DaVinci .TouchUp
do  set global FollowButton to 0
```

Figure B.2

C Supportive Pictures of system

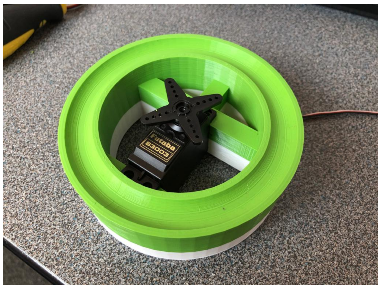


Figure C.1: Bottom base and placement of servo motor

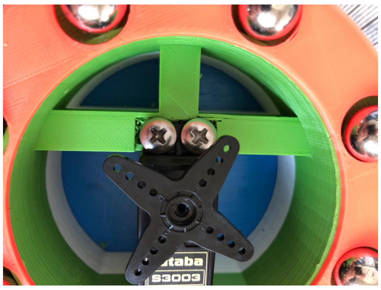


Figure C.2: Attachment onto support of bottom servo motor

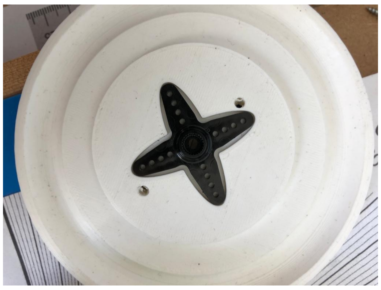


Figure C.3: Fit of servo motor head into topping

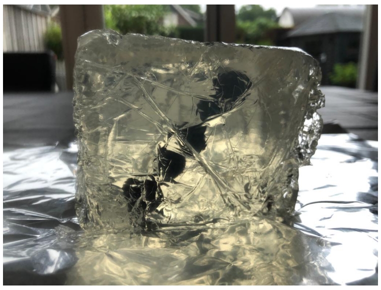


Figure C.4: Rough phantom (1)

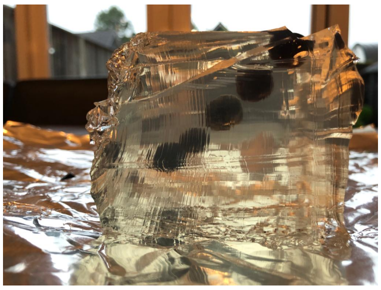


Figure C.5: Rough phantom (2)

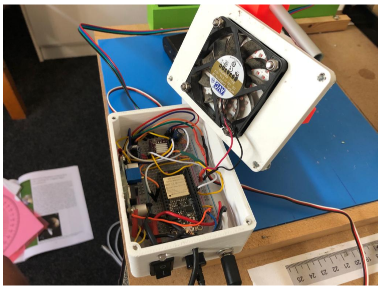


Figure C.6: Box for electronics

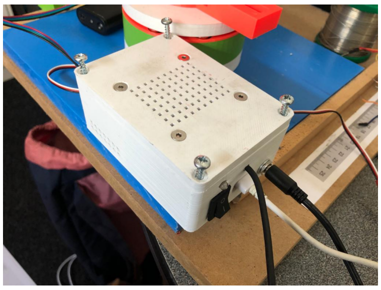


Figure C.7: Box for electronics

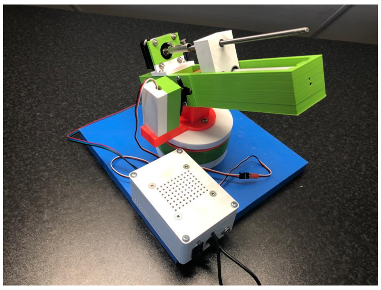


Figure C.8: Overview (1)

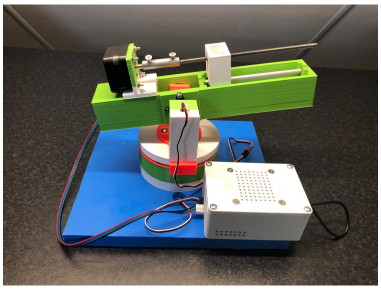


Figure C.9: Overview (2)

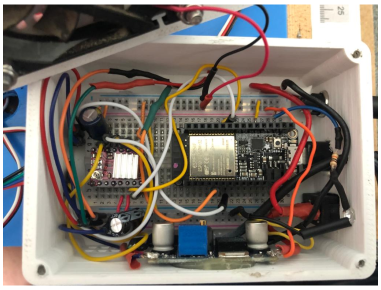


Figure C.10: Closeup Electronics

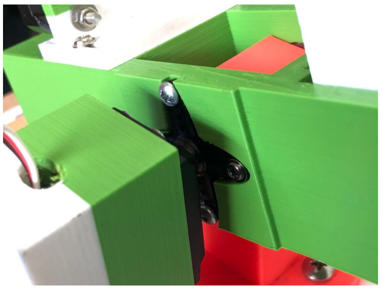


Figure C.11: Roll attachment

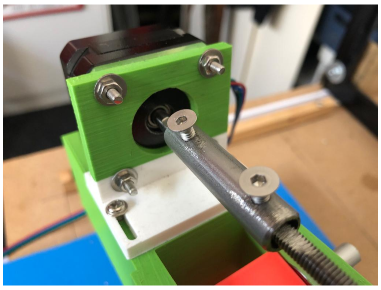


Figure C.12: Motor shaft

Bibliography

Bray, F., J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre and A. Jemal (2018), Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries, **vol. 68**, no.6, pp. 394–424.

Burris, M. (2020), Stepper Motors vs. Servo Motors, accessed: April 23, 2020. https://www.lifewire.com/stepper-motor-vs-servo-motors-selecting-a-motor-818841

Bélanger-Barrette, M. (2016), What is an Average Price for a Collaborative Robot?, [Online; accessed 8-June-2020]. https://blog.robotiq.com/what-is-the-price-of-collaborative-robots

Douglas, B. (2018), Drone Control and the Complementary Filter. https://www.youtube.com/watch?v=whSw42XddsU

Gollapudi, A. M., V. Velagapudi and S. Korla (2020), Modeling and simulation of a high-redundancy direct-driven linear electromechanical actuator for fault-tolerance under various fault conditions, *Engineering Science and Technology, an International Journal*, doi:https://doi.org/10.1016/j.jestch.2019.12.007.

Liang, K., A. J. Rogers, E. D. Light, D. von Allmen and S. W. Smith (2010), Three-Dimensional Ultrasound Guidance of Autonomous Robotic Breast Biopsy: Feasibility Study, **vol. 36**, no.1, pp. 173–177.

Mahmoud, M. Z., M. Aslam, M. Alsaadi, M. A. Fagiri and B. Alonazi (2018), Evolution of Robot-assisted ultrasound-guided breast biopsy systems, **vol. 11**, no.1, pp. 89 – 97, ISSN 1687-8507, doi:https://doi.org/10.1016/j.jrras.2017.11.005.

Mallapragada, V. G., N. Sarkar and T. K. Podder (2008), Robotic system for tumor manipulation and ultrasound image guidance during breast biopsy, in *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS'08 - "Personalized Healthcare through Technology"*, pp. 5589–5592.

Mallapragada, V. G., N. Sarkar and T. K. Podder (2008), Robotic system for tumor manipulation and ultrasound image guidance during breast biopsy, in *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5589–5592.

Mayo Foundation for Medical Education and Research (2020), Tumor size, [Online; accessed 8-May-2020]. https://www.mayoclinic.org/diseases-conditions/breast-cancer/multimedia/tumor-size/img-20006260

Merrill, R. T., M. W. McElhinny and P. L. McFadden (1998), Chapter Two The present geomagnetic field: Analysis and description from historical observations, in *The Magnetic Field of the Earth*, volume 63 of *International Geophysics*, Academic Press, pp. 19 – 68, doi:https://doi.org/10.1016/S0074-6142(96)80003-9.

Nelson, T. R., A. Tran, H. Fakourfar and J. Nebeker (2012), Positional calibration of an ultrasound image-guided robotic breast biopsy system, **vol. 31**, no.3, pp. 351–359.

Panteleimonitis, S., O. Pickering, M. Ahmad, M. Harper, T. Qureshi, N. Figueiredo and A. Parvaiz (2020), Robotic rectal cancer surgery: Results from a European multicentre case series of 240 resections and comparative analysis between cases performed with the da Vinci Si and Xi systems, **vol. 3**, no.1, pp. 6 – 11, ISSN 2468-9009, doi:https://doi.org/10.1016/j.lers.2019.12.002.

Pedley, M. (2013), Tilt Sensing Using a Three-Axis Accelerometer.
https://www.nxp.com/docs/en/application-note/AN3461.pdf

Printables, P. (2018), Linear Servo Actuator, accessed: April 28, 2020.
https://www.thingiverse.com/thing:3170748

Song, T., L. Zao and X. Shen (2011), The Research on Remote Control Robot Snake Driving by Servo, *Procedia Engineering*, **vol. 15**, pp. 762 – 766, ISSN 1877-7058, doi:https://doi.org/10.1016/j.proeng.2011.08.142, cEIS 2011.

Tanaiutchawoot, N., B. Treepong, C. Wiratkapan and J. Suthakorn (2014), A path generation algorithm for biopsy needle insertion in a robotic breast biopsy navigation system, in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*, pp. 398–403.

Tower Hobbies (2020), Futaba s3003 Standard Servo, [Online; accessed 8-May-2020].
https://www.towerhobbies.com/cgi-bin/wti0001p?&I=LXH288

Welleweerd, M. K., F. J. Siepel, V. Groenhuis, J. Veltman and S. Stramigioli (2020), Design of an end-effector for robot-assisted ultrasound-guided breast biopsies, *International Journal of Computer Assisted Radiology and Surgery*.

Wikipedia contributors (2020a), Servomotor, [Online; accessed 11-June-2020].
https://en.wikipedia.org/w/index.php?title=Servomotor&oldid=955109101

Wikipedia contributors (2020b), Stepper motor, [Online; accessed 11-June-2020].
https://en.wikipedia.org/w/index.php?title=Stepper_motor&oldid=958488838

Wikipedia contributors (2020c), Thrust bearring, [Online; accessed 8-May-2020].
https://en.wikipedia.org/w/index.php?title=Thrust_bearing&oldid=939395810