Final Report

Haptic Feedback Training

for

Pediatric Laparoscopic Surgery Simulators

Supervisors: Professor Carlos Rossa and Professor James R. Green

Second Reader: Professor Chao Shen

Group: 20

Mahtab Mohammed - 101052701

Reyad El Mahdy - 101064879

A report submitted to meet requirements of the SYSC-4907

Engineering Project

Department of Systems and Computer Engineering

Faculty of Engineering

Carleton University

Date: April 12, 2022

Abstract

The present project involved incorporating vibrotactile haptic feedback mechanism in a box trainer type Pediatric Laparoscopic Surgery (PLS) simulator, which has sensors instrumented in it to track the movements of the laparoscopic surgical instruments inserted into it within four Degrees of Freedom (DOF). These sensors include a nine DOF Inertial Measurement Unit (IMU) used to measure the pitch and yaw of said instruments and an optical sensor to measure the trust and roll of the same. The vibrotactile haptic feedback mechanism involves using a grid of Linear Resonant Actuator (LRA) type vibration motors attached to a wearable such that the laparoscopic surgical trainee wearing them during a laparoscopic surgical training task will feel these motors vibrate in particular patterns when their instruments veer off course from the expected regions of operation. These vibration patterns are designed to intuitively guide said surgical trainees to move their instruments back within said regions of operation.

Acknowledgements

We would like to express our gratitude to the following people to help us finalize the project. First, we wish to thank our supervisors, Professor Carlos Rossa and Professor James R. Green, for giving us the opportunity to work on this project. We also thank them for their support and guidance throughout the project. Next, we would like to thank Carleton Undergraduate Engineering Students' Equipment Fund (CUESEF) for funding the crucial and most expensive components of the project, i.e., Pediatric Laparoscopic Surgery (PLS) box trainer and the PLS instruments. On that note, we would also like to thank the Systems and Computer Engineering (SCE) department for funding the technical components of the project such as the sensors and the haptic motors. Furthermore, we would like to thank Michael Marsland from the Augmented Reality (AR) PLS team for helping design the 3D model of the trocar that housed the sensors. In the same vein, we would also like to thank Connor McDermott from the BioMechatronics laboratory at Carleton University and the lab itself for 3D printing assistance and services. In addition, we would like to thank the SCE Technical Support team for facilities and IT equipment support throughout the project. Finally, we would like to thank Carleton University engineering graduates Sami Hammoud and Farhan Chowdhury for offering circuit soldering and testing equipment for the project, without which the project would not have progressed to its current state.

Table of Contents

1. Introduction

1.1. Problem Background & Motivation

Engineering and medicine are two of the most rapidly evolving mainstream areas of study known to man. With new technologies and medical techniques that challenge the current conventions emerging every day, education and training become particularly troublesome tasks. As doctors and engineers alike are forced to keep up with the latest trends that may end up replacing the conventional. Surgery in particular is a field where staying on the cutting edge is paramount as it can potentially be the difference between life and death. This project discusses a surgical technique known as laparoscopy, and the challenges and solutions associated with the education and training of its practitioners.

Laparoscopy is a surgical technique which involves inserting thin rods (typically 3-5 mm thick) called laparoscopes that are fitted with cameras through small incisions on the patient's body [1]. The physician then performs the surgical procedure through the small incision using other laparoscopes fitted with surgical tools [1]. This allows for a minimally invasive surgical procedure, especially when compared to traditional open surgery. Open surgery requires incisions that are large enough for the surgeon to have a full view of the operating area [2]. This simplifies the procedure from the surgical perspective, but unfortunately causes much longer recovery times and higher infection rates [2]. As a result, laparoscopic surgery is used whenever possible, as the smaller incisions allow for a much faster recovery [2]. This also reduces pressure on in-patient care due to patients leaving the hospital much sooner than they would after open surgery [2].

However, laparoscopy does not come without disadvantages; one of the biggest issues with laparoscopy is training. Laparoscopic procedures differ greatly from open surgery both in terms of technique and perspective [2]. As a result, special training is required to ensure that surgeons are properly equipped to conduct laparoscopic surgery [3]. This is especially true in the case of pediatric surgery, as the laparoscopic instruments are smaller in diameter and the pediatric procedure is more sensitive than an adult procedure, leading to a greater need for precision [4].

1.2. Problem Statement

Traditionally, laparoscopy training is done by observing and assisting experienced surgeons in their procedures [3]. This unfortunately leads to long turnaround times when it comes to training new surgeons, as opportunities for hands-on practice with laparoscopy are limited due to the risks involved. Subsequently, a solution that accelerates the training process for novices is necessary to be able to meet the growing personnel requirements of the surgical field. A particularly prominent solution is the use of simulators to allow novices to practice in a risk-free environment [3]. As a result, the primary objective of this project is to design and implement a fully functional pediatric laparoscopic surgery simulator for the purpose of accelerating the process of training novice surgeons. This project is also intended to serve as a foundation for future work by engineering students to further improve the efficacy of the final product until it is ready to be used for its intended purpose.

1.3. Proposed Solution

Surgical simulators generally fall under one of two categories: VR (Virtual Reality) trainers, and box trainers. A VR trainer is a device that uses virtual reality technology to immerse the user in a virtualized simulation of the surgical procedure they want to practice [3]. In this case, the user uses VR goggles to view the procedure, and controls the virtual tools through physical tools that are tracked and mapped onto the virtualized area so that the movement of the physical objects reflects proportional movement within the simulation (see figure 1) [3].

Figure 1.3.1 A trainee practicing laparoscopy on a virtual reality trainer [5]

These simulators can be very effective, but at the same time very expensive to produce, program, and maintain, making it a less scalable solution than the alternative: the box trainer [3]. The box trainer on the other hand, is a much simpler device which features laparoscopic tools going into a box that blocks vision of the surgical area. The box is then fitted with a camera that provides the user with a video feed showing the position of the laparoscopes as one would find in a real laparoscopic procedure. Within the box are tasks that are meant to provide the trainee with the opportunity to practice basic surgical techniques such as suturing and transferring objects between laparoscopes.

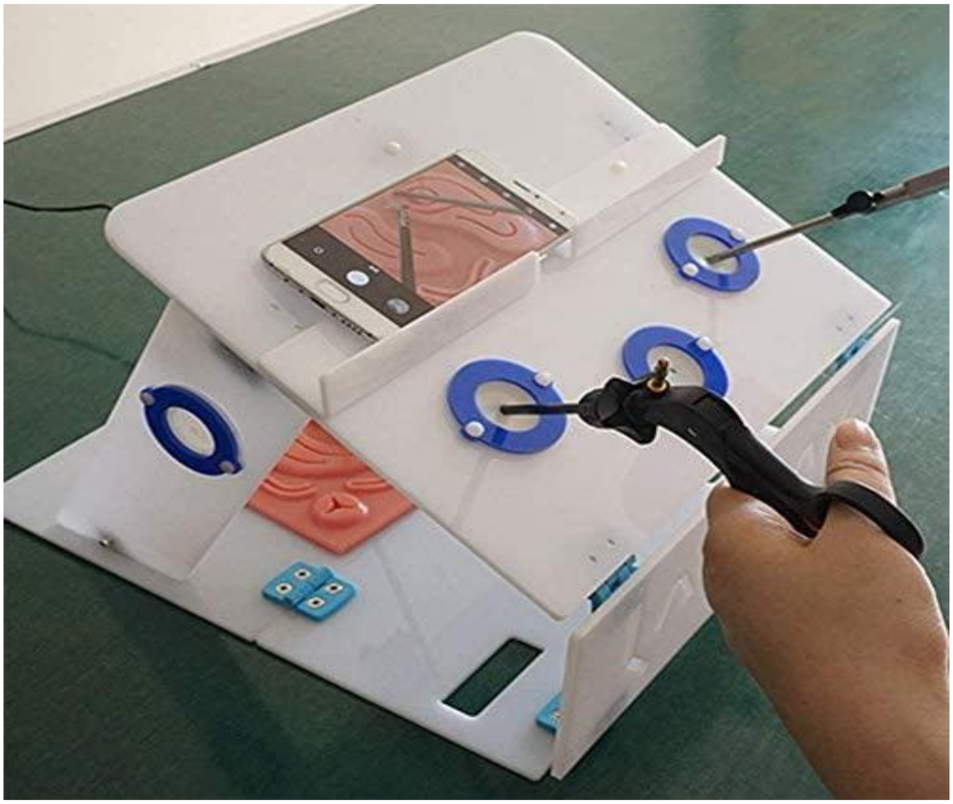

Figure 1.3.2 example of a simple box trainer performing a suturing task [6]

Due to their simplicity, box trainers are inexpensive and effective methods for training novice surgeons on performing laparoscopic procedures. A potential flaw in the box trainer is that it lacks the ability to provide feedback; the user is still dependent on an expert's assessment for feedback. While this is still a viable solution, it lacks convenience for both the user and the expert conducting the assessment. This highlights the need for the administration of automated feedback by the simulator, conveniently providing the novice with feedback until they are ready to be assessed by an expert. The two most common forms of automated feedback in the industry are visual feedback and haptic feedback [7].

Visual feedback involves the use of sensors and/or computer vision to determine when and where feedback is warranted and provide said feedback in the form of visual stimuli which augment the video feed with helpful information [8]. This technology is also known as augmented reality and is often used to guide the user towards (or away from) specific paths by overlaying a video feed with the necessary feedback. Haptic feedback on the other hand, involves similar logic to determine if feedback is needed, except the feedback administered is perceived through the user's sense of touch [9]. These two forms of feedback are not exclusive and can potentially be combined within a single simulator and are fully compatible with both VR simulators and box trainers. A combined solution would have been ideal in producing intuitive feedback and thus reducing training turnaround times. However, due to time and personnel constraints, a singular approach had to be chosen from these potential solutions.

For the purpose of this project, the type of simulator chosen is a box trainer which was chosen for its simplicity and cost. A VR simulator would have been effective, but the high cost associated combined with the engineering requirements for such a project caused it to be unfeasible. It was then decided that the simulator's feedback would be delivered haptically as haptic simulators have been found to be effective in past research [10]. Augmented reality is also a very effective delivery method, but this was avoided as another group was working on a similar simulator that uses augmented reality instead of haptics.

1.4. Accomplishments

The project succeeded in designing and implementing a functional laparoscopic simulator that provides the user with automated feedback based on input from sensors. Consequently, this result covers most of the functional metrics for success defined by the project proposal attached in Appendix A of the report. However, due to the logistical constraints that come from the ongoing pandemic, the simulator has not yet been assessed by an expert (or novice) surgeon who can properly determine whether or not this satisfies the ultimate goal of the simulator, which is to accelerate training. Apart from the basic project requirements, there are several potential improvements that can be made to enhance the effectiveness of the simulator and these improvements will be elaborated upon in later sections within this report.

1.5. Overview of Report

The remainder of the report is split into 5 chapters. Chapter 2 of this report describes the engineering project in terms of health and safety, engineering professionalism, project management style, suitability for the degree program, as well as provides a list of individual contributions for each of the group members. Next, Chapter 3 provides technical details that are relevant to the problem or the project in itself. Chapter 4 then provides a detailed technical overview of the project accomplishments and explains the rationale behind each part and how everything fits together. Then, Chapter 5 outlines the group's recommendations for future work on the project, which involves various potential improvements for later contributions. Finally, Chapter 6 concludes the report and provides some final reflections from the group. The original project proposal is also attached within Appendix A, and a description of the GitHub repository can also be found in Appendix B.

2. The Engineering Project

2.1 Health and Safety

Despite the bulk of project work being unsupervised, proper safety precautions were taken where applicable throughout the entire project. Much of the project work was done through software posed minimal risk to the workers (group members) as no outside software was used. In addition to this, none of the hardware running the programs was left unattended while active/operational. Proper safety precautions were taken to ensure that any active components were completely disconnected from power while left unattended. Project members have also worn protective equipment during potentially hazardous tasks such as soldering. The project member in charge of soldering would wear protective glasses and a mask to prevent the ingestion or inhalation of lead or any other hazardous material. Finally, any exposed wiring was covered in shrink tubing to ensure proper electrical safety.

Apart from project specific precautions, the group members had also acted in compliance with all public health and university regulations regarding the COVID-19 pandemic. All personnel involved with the project have maintained a safe distance from each other throughout the entire academic year by meeting virtually both with the project supervisors and between group members when possible. Social distancing was also exercised during periods where common spaces such as laboratories were used. Surgical masks have also been worn at all times when using common spaces or handling project hardware to ensure COVID-19 safety at all times.

2.2 Engineering Professionalism

With regards to engineering responsibility, the entire project was conducted with not only the end user, but also any potential future contributors in mind. In compliance with the Code of Ethics, the simulator in no way endangers the public or the end user as they would never be in direct contact with any of the active components [11]. The ultimate purpose of this project is to improve training for surgical novices, effectively improving public safety when it comes to laparoscopic procedures. The project also does not falsely advertise itself as a finished product, as the project is intended to serve as a foundation for future work by engineering students on the further enhancement of the simulator. Finally, the group members of this project are not and do not claim to be professional engineers; they are engineering students working under the supervision of professional engineers. Consequently, the foundation provided by this project is intended to be further built upon future engineering students working under licensed professional engineers until the project is deemed fit to be handed over to clinical experts for further evaluation. Therefore, this engineering project is in compliance with all Canadian engineering regulations to the best of the authors' knowledge [12].

2.3 Project Management

Management of this project was done by periodically assessing the project management triangle, shown in figure 2.3.1, throughout the progression of the project [13]. At the beginning, the project scope and time was outlined based on the potential funding of the project. For instance, building of the Pediatric Laparoscopic Surgery (PLS) box trainer was planned such that work on it would begin after the decision dates of the funding applications made for this project. This was done so that further project decisions can be made by reassessing the impact of cost, scope and time on the quality of the project based on how much funding the project received. If the project had not received funding for a commercially bought PLS box trainer, the timeline and scope of the project would be reassessed such that the quality of the project would not be undermined. However, the project did receive funding for the PLS box trainer, and the initial timeline was then reassessed for shipping and included in the proposal as seen in section 8 of Appendix A.

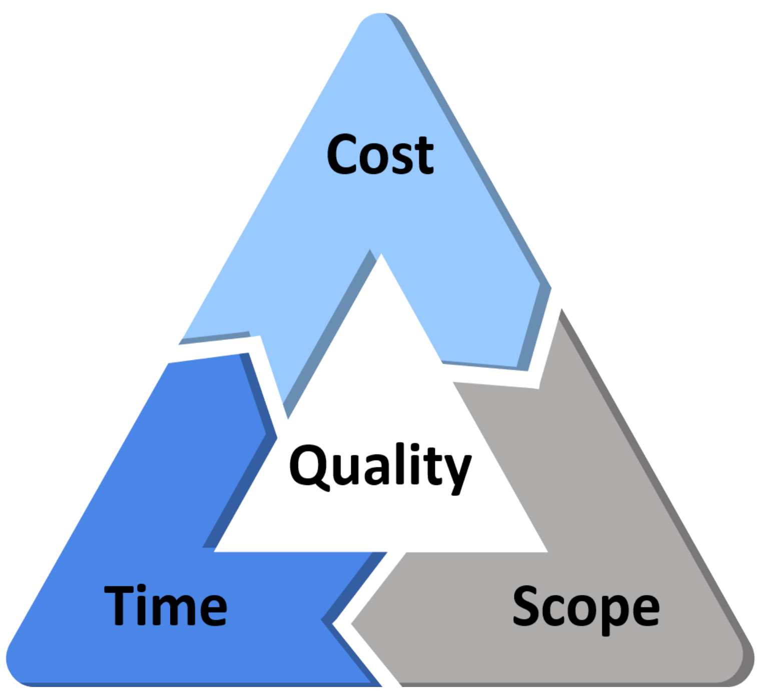

Figure 2.3.1 Project management triangle [13]

To further ensure quality, the timeline was designed such that each member would be primarily responsible for two things per stage of the project such that each member would be responsible for working on their part of the project while also validating a part of the project that was worked on by the other team member.

Later in the project, it became apparent that the timeline of the project would be disrupted due to extenuating circumstances and obligations of the team members. As such, the scope of the project was negotiated and regularly reassessed in the weekly meetings held with the supervisors to account for these circumstances. Project design input and recommendation form the supervisors were also offered during these weekly meetings that were each about thirty minutes long. Furthermore, one of the above-mentioned circumstances involved a project team member to be out of the country during the project. As such, the project workload was divided into in-person and remote work using recommendations from the supervisors and delegated accordingly.

2.4 Justification of Suitability for Degree Program

Both team members of this project are in the Biomedical and Electrical Engineering degree program. As such, the suitability of this project for said program is near perfect. This is because building PLS simulators and corresponding research on the subject matter fall under the category of Biomedical Engineering. Similarly, incorporating sensors and haptic motors into circuits and implementing the data acquisition and control mechanisms for them fall under the category of Electrical Engineering. Therefore, both aspects involved in the degree programs of the project team members are covered by this project.

Knowledge and experience acquired throughout the Biomedical and Electrical Engineering degree program was used in this project. For example, knowledge of Inter-Integrated Circuit (I2C) and Serial Peripheral Interface (SPI) communication protocols for microcontrollers was used in this project to drive the haptic motors and acquire data from the sensors respectively. This helpful knowledge was obtained from the ELEC 4601 course, i.e., Microprocessor Systems. Similarly, knowledge of C coding from SYSC 2006, i.e., Foundations of Imperative Programming, was applied in this project to build C++ based Arduino code that controls the inputs and outputs of the microcontroller.

In addition, programming knowledge of Python from SYSC 2010, i.e., programming project and version control systems knowledge from ELEC 4700, i.e., Modeling of Integrated devices, was used in this project to do essential programming and version control of the code for haptic feedback in the PLS simulator. Furthermore, signal processing knowledge from SYSC 4405, i.e., Digital Signal Processing, was used to process the data acquired from the sensors. Finally, work experience from ELEC 3999, i.e., Co-operative Work Term, offered transferable debugging skills that were used throughout this project.

2.5 Individual Contributions

Following subsections outline individual contributions from each team member of the project. Contributions to the project are itemized in a bulleted list under the names of each team member of the project. Report contributions, on the other hand, are conveniently outlined in a table showing each chapter and section of this report, as well as the corresponding author.

2.5.1 Project Contributions

Mahtab:

- PLS Box Trainer -
  - Assembled the initial PLS box trainer
  - Integrated the microcontrollers and sensors via custom 3D printed trocar into the PLS box trainer
  - Integrated haptic feedback wearable via Pulse Width Modulation (PWM) driver into the PLS box trainer to complete final assembly of project
- Sensors -
  - Tested reading in data from the optical sensors into the microcontroller via SPI using script 2.6 in Appendix B
  - Integrated the Inertial Measurement Units (IMU) and optical sensors into one circuit as per SPI protocol
  - Tested sensor data collection from the above SPI based circuit using program 2.1 in Appendix B

- Haptic Feedback Wearable -
    - Tested functioning of individual Linear Resonant Actuators (LRA) using program 2.3 in Appendix B
    - Soldered LRAs to multi conductor cables to elongate connection points
    - Instrumented LRAs onto wearables
    - Connected the elongated LRAs to the PWM driver and the PWM driver to the microcontroller via I2C. Tested functioning of the PWM driver connections using program 2.4 in Appendix B
- System Programming -
    - Tested Arduino interface with Python and vice-versa using program 2.7 in Appendix B
    - Programmed final Arduino code for haptic feedback in program 1.2 of Appendix B. This program activated or deactivated LRAs via PWM driver based on input string received from program 1.1 in Appendix B while sending sensor data to the same

Reyad:

- PLS Box Trainer -
    - Adjusted the custom trocar initially designed by Michael Marsland from the AR team to fit the simulator's dimensions and house the pediatric instruments through 3D modeling
    - Took measurements and made the geometric calculations needed to evaluate the thresholds used by the Python script to determine whether feedback is necessary

- Sensors -
    - Wrote the initial version of the Arduino program in charge of collecting data from the IMUs (see program 2.2 in Appendix B), this was later integrated into a broader program which controlled both types of sensors and triggered the feedback patterns (program 2.7 in Appendix B)
    - Determined the format for samples of sensor data being sent across the serial connection
- Haptic Feedback Wearable -
    - Created and tested the intuitive haptic feedback patterns triggered by the Arduino after receiving instructions for feedback
- System Programming -
    - Designed, implemented, and tested the final Python code responsible for the reading, windowing, and reformatting sensor data (Appendix B - program 2.5)
    - Designed and wrote the logic and code for the Python state controller which is the main decision-making component of the PLS simulator (Appendix B - program 2.5)
    - Developed the system for sending instructions from Python to the Arduino depicting the pattern and direction of the feedback warranted

2.5.2 Report Contributions

Report Contributions		
Chapter	**Section**	**Author**
Abstract & Acknowledgements		Mahtab
1. Introduction		Reyad
2. The Engineering Project	2.1 Health and Safety	Reyad
	2.2 Engineering Professionalism	Reyad
	2.3 Project Management	Mahtab
	2.4 Justification of Suitability for Degree Program	Mahtab
3. Technical Background		Mahtab
4. Detail of the Design	4.1 Pediatric Laparoscopic Surgery Box	Mahtab
	4.2 Sensors and Custom Trocar Design	Mahtab
	4.3 Haptic Feedback Wearable Design	Mahtab
	4.4 Feedback Delivery	Reyad
	4.5 Project Development	Reyad
5. Recommendations for Future Work	5.1 Recommendations for Improvements	Reyad
	5.2 Recommendations for Additional Features	Mahtab
6. Conclusions and Reflections		Reyad
Appendix B: GitHub Repository Description		Mahtab

3. Technical Background

As mentioned in the introduction, visual or haptic feedback in laparoscopic surgical simulators has a positive impact on surgical trainee performance [10]. This project focuses on haptic feedback mechanisms for such simulators based on the notion that excessive visual stimuli may prove to be distracting for laparoscopic surgical trainees. Haptic feedback in the context of this project does not involve simulating the feel and resistance of tissue material during laparoscopic surgery. Rather haptic feedback in the context of this project involves the use of a wearable fitted with a grid of vibrotactile actuators that vibrate against the skin of surgical trainees when given certain input. This is also referred to as vibrotactile haptic feedback. Unless otherwise mentioned, haptic feedback in this report refers to vibrotactile haptic feedback. This concept was adapted from [14] where this form of haptic feedback was offered to guide needle steering in brachytherapy. Similarly, for this project such haptic feedback helps guide surgical trainees to change the movements of their surgical instruments during laparoscopic training if they are veered off course. The incorporation of haptic feedback for this project via wearables is further discussed in section 4.3 of this report.

Consequently, offering such haptic feedback enabled guidance requires a mechanism to track the laparoscopic surgical instruments. Review of the literature offers solutions to implement this tracking mechanism using sensors in surgical simulators. For instance, 6-axis Inertial Measurement Units (IMU) and laser range sensors were used in [15] to track the gyroscopic movement and depth respectively of the laparoscopic surgical instruments.

Also, in [16] optical mouse sensors were used instead of the above-mentioned laser range sensors which allowed tracking roll of the shaft of the instruments in addition to the depth. As such, this project made use of a similar setup as [16] to track the movements of surgical instruments and establish boundaries for said movements. In addition, a structure is required to hold these sensors in place during movement of the laparoscopic surgical instruments. This is done using a custom trocar which is discussed further along with the sensors in section 4.2 of this report.

Furthermore, an integration system is required to synchronize the tracking information from the sensors and the activation of the haptic feedback. This was mainly done by programming a microcontroller in this project and discussed more in section 4.4 of this report. Lastly, incorporating all the above discussed elements into a single Pediatric Laparoscopic Surgery (PLS) simulator would require said simulator to be easily modifiable. Therefore, a simple box trainer type PLS was leveraged for this project as discussed in section 4.1 of this report.

4. Detail of the design

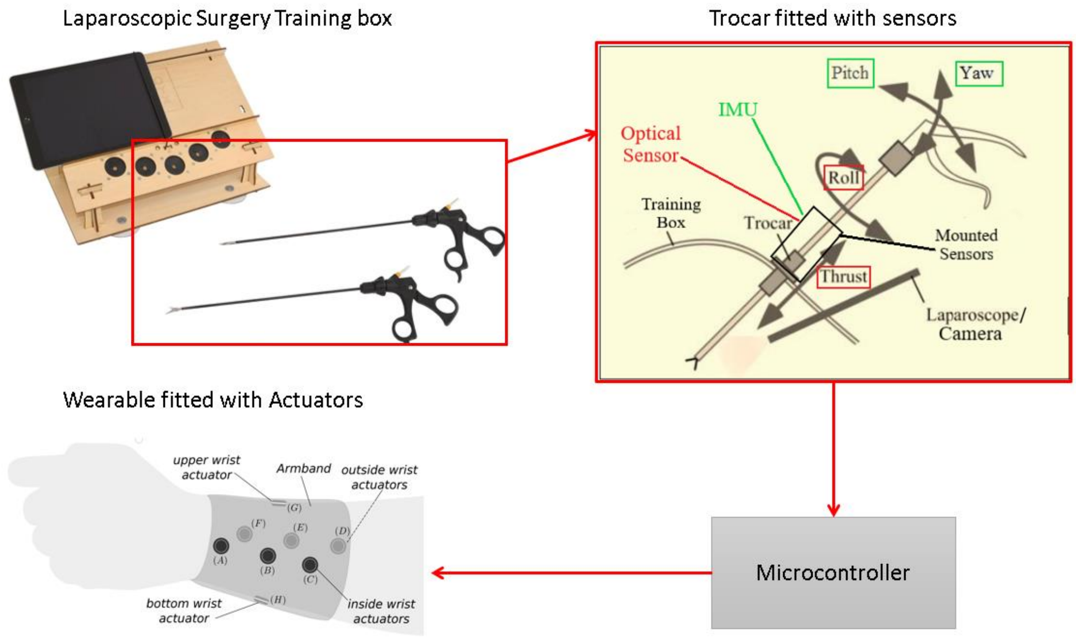

Figure 4.1 Overall concept of project[1]

The following sections offer detailed explanations of the main elements of the project. Figure 4.1 illustrates an overview of how these elements were put together. First, a box trainer type Pediatric Laparoscopic Surgery (PLS) simulator was built, and sensors connected to a microcontroller were attached to said simulator by means of custom built trocars. These sensors provided real-time information on the movements of the surgical instruments involved in the PLS simulator. The PLS box trainer and sensors including the custom trocars housing them are discussed further in sections 4.1 and 4.2 of this report respectively. In addition, a wearable, such as a wrist brace, was instrumented with a grid of vibrotactile actuators connected to the above-mentioned microcontroller. This wearable is further discussed in section 4.3 of this report.

[1] Figure 4.1 has been adapted from [14], [17], and [18]

These vibrotactile actuators, henceforth referred to as actuators or motors, provided tactile haptic feedback in specific patterns based on specific inputs from the microcontroller. Section 4.4 offers further insight into the input and output mechanisms of the sensors, actuators, and the Arduino based microcontroller that connects them all.

4.1 Pediatric Laparoscopic Surgery (PLS) Simulator Box

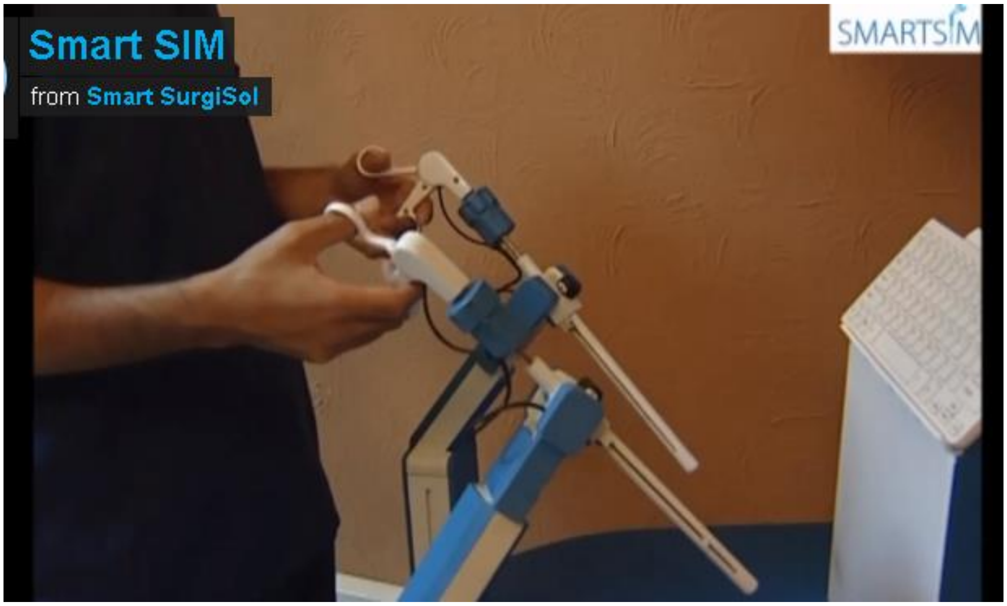

Figure 4.1.1 Virtual Reality laparoscopic surgery trainer [19]

The choices for laparoscopic surgery trainers were between that of a simple box trainer or a virtual reality (VR) trainer. Most studies where participants compared classical box trainers to VR trainers seemed to favor the former [4, p254] [10, p254] [8] [20]. This may be the general sentiment due to lack of natural haptic feedback in VR trainers such as those in figure 4.1.1 [19]. Some VR trainers that incorporated simulated haptic feedback received complaints of added friction during training [10, p254].

Furthermore, a simple box trainer was deemed easier to modify for incorporation of custom trocars and sensors for the purposes of this project. In contrast, some of the existing features of a VR trainer may have had to be removed to adapt it for the purposes of the project. Thus, for the above discussed reasons a box trainer was used in this project.

Figure 4.1.2 Laparoscopic box trainer and 3mm laparoscopic instruments [17]

A commercially sold box trainer, as shown in the left image of figure 4.1.2, was used for this project as the product offered instructions and components to build it specifically for PLS. For example, additional components were provided for the height of the box such that it could be built shorter. In addition, 3 mm laparoscopic instruments suitable for PLS were also provided as shown on the right image of figure 4.1.2. Moreover, this box was designed by a clinician thus providing more dimensional reliability for PLS training [17]. The box trainer was assembled as shown in figure 4.1.3. A USB webcam was attached to the box which allowed the trainee to view the state of the surgical instruments during training. An example of the webcam feed is shown in figure 4.1.4 displaying the object transfer surgical training exercise kit.

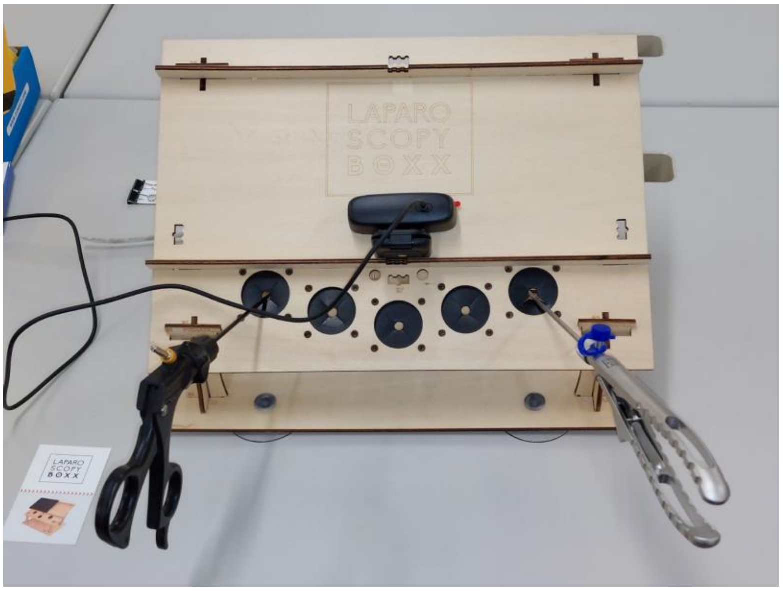

Figure 4.1.3 Assembled laparoscopic box trainer

Figure 4.1.4 Object transfer surgical training exercise kit

The objective of this exercise kit is to transfer objects from pegs on one side of the board to pegs on the other side using surgical instruments. This kit was used for the entirety of this project. Sensors were used to track the movements of the inserted laparoscopic instruments during this exercise to enable haptic feedback when said instruments exceeded expected bounds. The mechanism for such feedback and sensors involved are discussed further in the following subsections.

4.2 Sensors and Custom Trocar Design

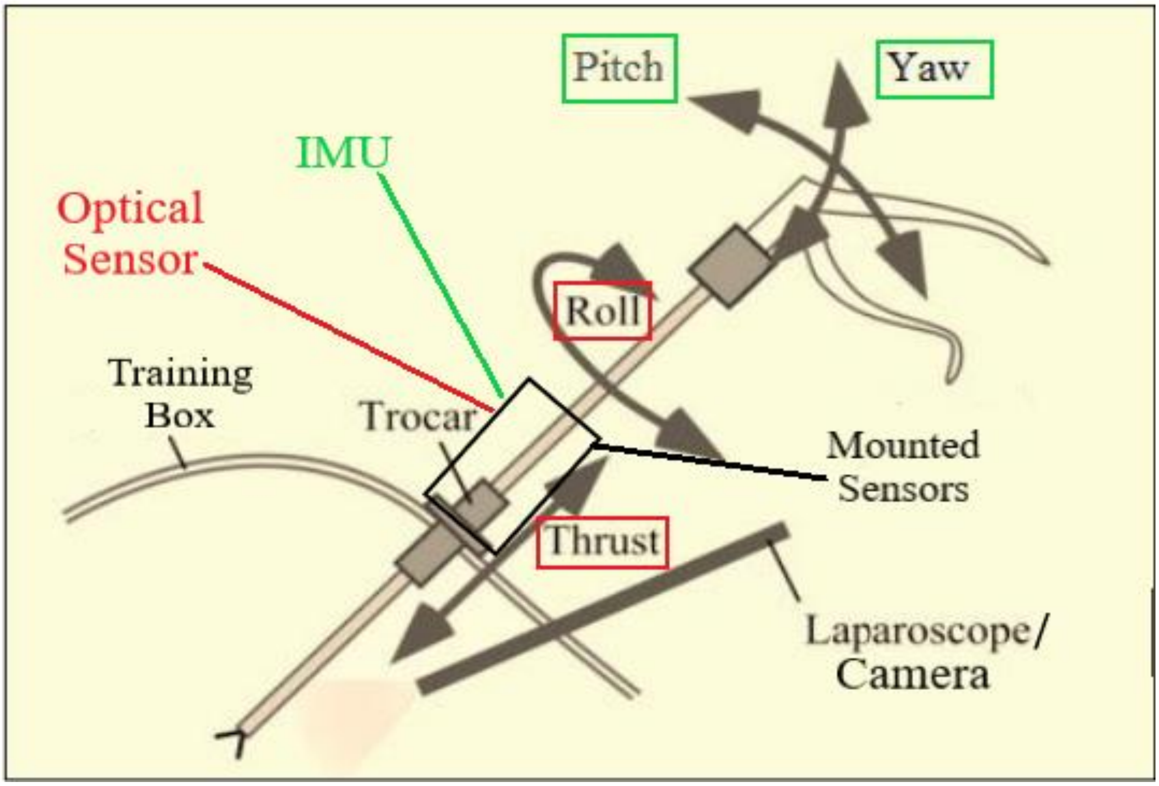

Figure 4.2.1 Proposed concept for sensors in surgical instrument of PLS simulator[2]

Laparoscopic surgical instruments inserted in the above discussed PLS box are allowed to move within four Degrees of Freedom (DOF). These degrees of freedom consist of pitch and yaw, highlighted in green in figure 4.2.1, as well as roll and thrust, highlighted in red in figure 4.2.1. As such, two types of sensors were used to track these degrees of freedom. Inertial Measurement Units (IMU) were used to track the pitch and yaw and optical mouse sensors were used to track the roll and thrust. As further highlighted in figure 4.2.1, these sensors were positioned along the shaft of the instruments by means of custom trocars for movement tracking purposes.

[2] Figure 4.2.1 was adapted from [18]

4.2.1 Inertial Measurement Unit (IMU)

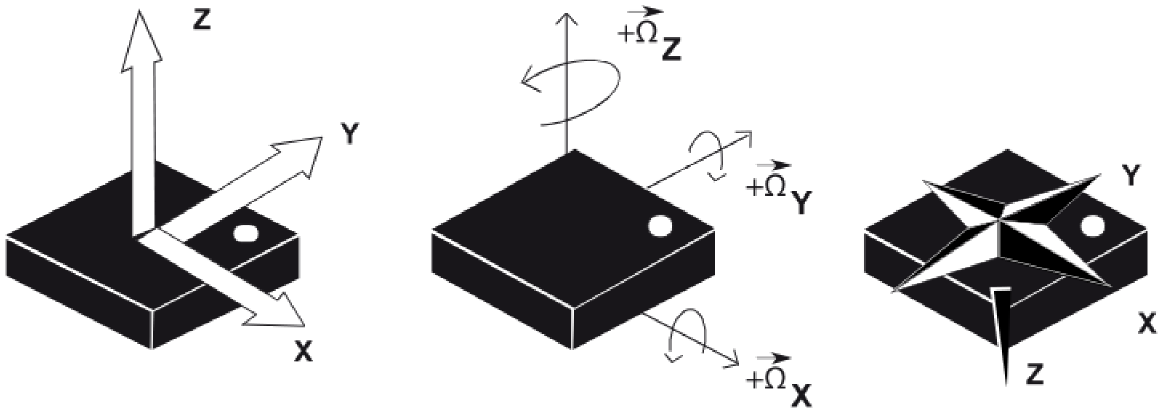

Figure 4.2.2 Nine Degrees of Freedom measured by LSM9DS1 IMU [21]

As discussed in the technical background, the most common sensor used to track pitch and yaw of laparoscopic surgical instruments in laparoscopic simulators were Inertial Measurement Units (IMU). Therefore, nine DOF IMUs were used for the same in this project. To be specific, the LSM9DS1 inertial chip in a STEVAL-MKI159V1 board was used as the nine DOF IMU for this project [22]. Each LSM9DS1 IMU includes an accelerometer, a gyroscope, and a magnetometer which measure the acceleration, angular rotation and magnetic force in 3 axes respectively. An illustration of these measurements is shown in figure 4.2.2 [21]. Therefore, each IMU was able to transmit nine measurements to the microcontroller at a time, either via Serial Peripheral Interface (SPI) or Inter-Integrated Circuit (I2C) communication protocol.

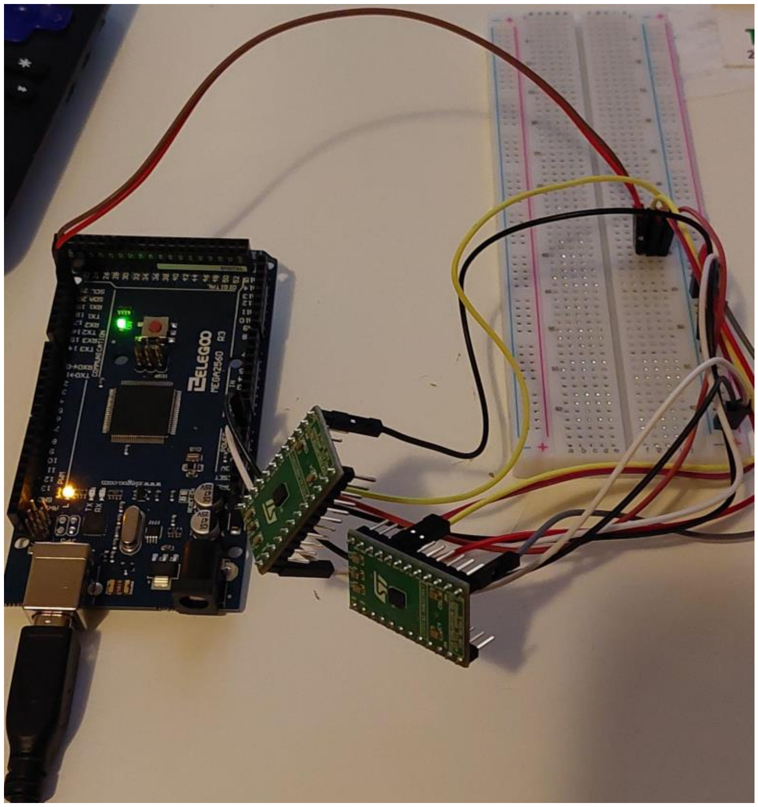

Figure 4.2.3 IMU testing circuit

Initially, the IMUs were intended to be used via I2C protocol and were tested as such using the IMU testing circuit in figure 4.2.3. I2C was preferred due to the use of fewer wires, however the protocol used for IMUs in the final stage of the project was SPI due to reasons further discussed in subsection 4.2.3. Raw data obtained from the sensors were converted to respective units such as m/s^2 for accelerometer readings using vendor provided Arduino libraries for the IMU [23]. This converted data was read in and organized in the serial port such that the first 6 readings were that of the gyroscope of the first and second IMU. Similarly, the next 6 were readings of each accelerometer, and the last 6 were that of the magnetometers. This organization of IMU data was displayed on the Arduino serial monitor as shown in figure 4.2.4.

```
COM5

02:21:21.075 -> -0.25,1.17,0.94,-1.93,1.01,-0.98,-0.04,-1.05,0.04,0.52,0.73,-0.45,0.06,-0.91,0.32,0.64,-0.40,0.51
02:21:21.169 -> -0.11,1.18,0.95,-2.17,0.52,-1.08,-0.04,-1.04,0.04,0.52,0.73,-0.44,0.06,-0.92,0.32,0.64,-0.40,0.51
02:21:21.263 -> -0.20,1.16,1.05,-2.11,0.81,-0.99,-0.04,-1.04,0.04,0.52,0.74,-0.44,0.06,-0.92,0.32,0.65,-0.40,0.51
02:21:21.403 -> -0.32,1.35,0.95,-1.93,0.72,-0.78,-0.04,-1.04,0.04,0.52,0.73,-0.45,0.06,-0.91,0.32,0.65,-0.40,0.51
02:21:21.497 -> -0.39,1.31,0.77,-1.82,0.63,-0.77,-0.04,-1.04,0.05,0.52,0.73,-0.44,0.06,-0.91,0.32,0.64,-0.40,0.52
02:21:21.589 -> -0.25,1.04,0.80,-1.82,0.60,-0.96,-0.04,-1.05,0.04,0.52,0.74,-0.45,0.06,-0.92,0.33,0.64,-0.40,0.51
02:21:21.729 -> -0.15,1.31,1.05,-1.96,0.74,-1.01,-0.04,-1.04,0.04,0.52,0.74,-0.45,0.06,-0.92,0.32,0.64,-0.41,0.52
02:21:21.822 -> -0.26,1.16,0.82,-2.21,0.81,-1.08,-0.04,-1.05,0.05,0.52,0.73,-0.44,0.06,-0.92,0.32,0.65,-0.41,0.51
02:21:21.964 -> -0.31,1.28,0.86,-2.12,0.88,-0.93,-0.04,-1.04,0.05,0.52,0.73,-0.44,0.06,-0.92,0.32,0.64,-0.41,0.52
02:21:22.056 -> -0.31,1.13,0.81,-1.85,0.91,-0.93,-0.04,-1.04,0.04,0.52,0.74,-0.44,0.06,-0.91,0.32,0.64,-0.41,0.51
02:21:22.195 -> -0.05,1.28,0.83,-1.82,0.62,-0.91,-0.04,-1.04,0.04,0.52,0.74,-0.45,0.05,-0.92,0.32,0.64,-0.40,0.51
02:21:22.288 -> -0.25,1.20,0.81,-2.15,0.70,-0.91,-0.04,-1.04,0.05,0.52,0.73,-0.44,0.06,-0.92,0.32,0.64,-0.41,0.51
02:21:22.382 -> -0.09,1.37,0.95,-2.08,0.81,-1.04,-0.04,-1.04,0.04,0.52,0.73,-0.44,0.06,-0.92,0.32,0.64,-0.41,0.51
02:21:22.523 -> -0.33,1.22,0.79,-1.86,0.60,-0.84,-0.04,-1.04,0.04,0.52,0.73,-0.44,0.06,-0.92,0.32,0.65,-0.41,0.51
02:21:22.617 -> -0.28,1.19,0.80,-1.90,0.55,-1.14,-0.04,-1.05,0.04,0.52,0.73,-0.45,0.06,-0.92,0.32,0.64,-0.41,0.51
02:21:22.711 -> -0.17,1.40,0.89,-1.56,0.73,-0.96,-0.04,-1.05,0.04,0.52,0.73,-0.44,0.06,-0.92,0.32,0.64,-0.41,0.51
02:21:22.850 -> -0.24,1.21,0.96,-2.06,0.57,-1.14,-0.04,-1.04,0.04,0.52,0.74,-0.44,0.05,-0.92,0.33,0.64,-0.41,0.51
02:21:22.944 -> -0.45,1.12,0.81,-1.95,0.83,-1.02,-0.04,-1.05,0.05,0.52,0.73,-0.45,0.06,-0.92,0.32,0.64,-0.41,0.51
02:21:23.037 -> -0.26,1.07,0.98,-1.87,0.74,-0.76,-0.04,-1.04,0.04,0.52,0.73,-0.44,0.06,-0.92,0.32,0.64,-0.41,0.51
02:21:23.180 -> -0.26,1.08,0.88,-2.02,0.93,-0.85,-0.05,-1.05,0.04,0.52,0.74,-0.45,0.06,-0.92,0.32,0.65,-0.41,0.51
02:21:23.272 -> -0.14,1.27,0.88,-2.39,0.64,-1.07,-0.04,-1.05,0.04,0.52,0.74,-0.44,0.06,-0.92,0.32,0.64,-0.41,0.51
02:21:23.411 -> -0.17,1.04,0.87,-2.12,0.93,-0.99,-0.04,-1.04,0.05,0.52,0.73,-0.45,0.05,-0.92,0.32,0.64,-0.41,0.51
02:21:23.504 -> -0.13,1.07,0.84,-2.23,0.84,-0.99,-0.04,-1.04,0.04,0.52,0.73,-0.44,0.06,-0.92,0.32,0.64,-0.41,0.51
02:21:23.645 -> -0.35,1.10,0.84,-1.74,0.92,-0.94,-0.04,-1.05,0.04,0.52,0.73,-0.44,0.06,-0.92,0.32,0.64,-0.41,0.51
02:21:23.739 -> -0.18,1.15,0.88,-2.41,1.00,-1.01,-0.04,-1.05,0.04,0.52,0.72,-0.44,0.06,-0.92,0.32,0.65,-0.41,0.50
02:21:23.834 -> -0.28,1.13,0.83,-1.97,0.68,-1.05,-0.05,-1.05,0.04,0.52,0.73,-0.45,0.06,-0.92,0.33,0.64,-0.41,0.50
02:21:23.966 -> -0.32,1.34,0.86,-1.97,0.78,-1.07,-0.04,-1.04,0.04,0.52,0.73,-0.44,0.06,-0.92,0.31,0.64,-0.40,0.50
02:21:24.057 -> -0.39,0.92,0.89,-2.06,0.88,-0.95,-0.04,-1.05,0.05,0.52,0.73,-0.44,0.06,-0.93,0.32,0.64,-0.41,0.50
02:21:24.154 -> -0.21,1.37,0.87,-1.85,0.63,-0.89,-0.05,-1.05,0.04,0.52,0.74,-0.44,0.05,-0.92,0.32,0.64,-0.41,0.51
```

Figure 4.2.4 Data obtained from IMUs

4.2.2 Optical Mouse Sensors

The nine DOF IMU would have sufficed to track all four DOF of the laparoscopic surgical instruments if it could offer accurate positional measurements long-term. However, this is not the case due to the accumulation of “drift” error over time which is caused by inherent imperfections and noise within such devices [24]. With time, this consequently causes calculation errors in the IMU measurements at each sampling instant, which further leads to inaccurate positional data. Therefore, the approach in [16] was used where the IMUs were solely relied on for pitch and yaw measurements and optical sensors were incorporated to measure the roll and thrust of the laparoscopic instruments. To be specific, the PMW3389 optical sensor chip manufactured for high-end gaming mice was used for this project [25].

The PMW3389 chip was bought from a vendor that provided a functioning breakout board for the chip with the required lens attached to it [26]. As shown in figure 4.2.5, this lens allows the light from the infrared source to illuminate the surface under the optical sensor [27]. In addition, this lens helps focus the image of the illuminated surface on the picture element array in the optical sensor as shown in figure 4.2.6 [27].

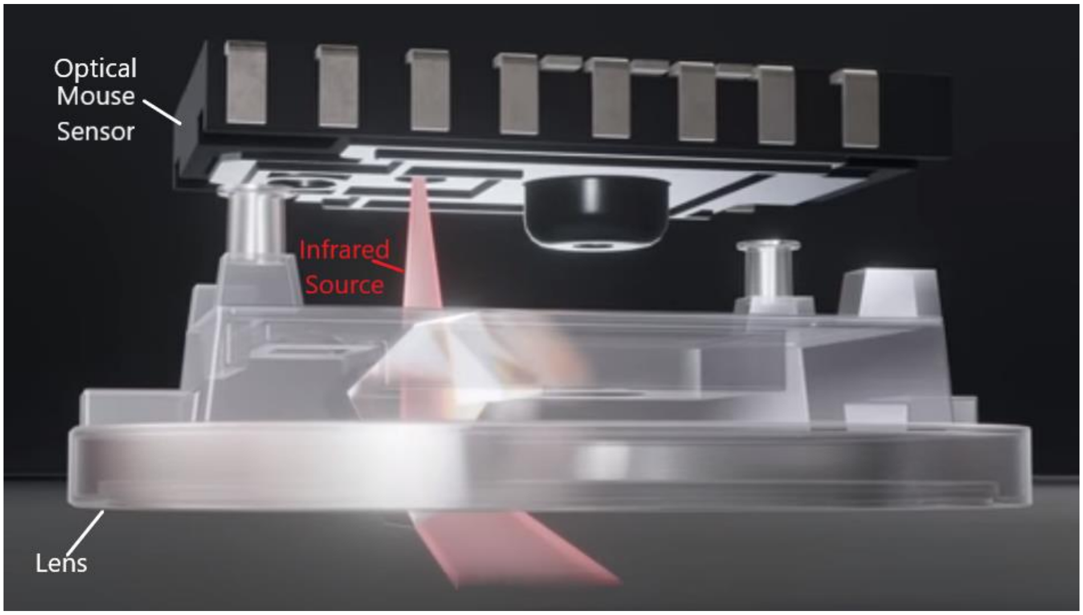

Figure 4.2.5 Infrared illumination of surface via Lens of Optical Sensor [27]

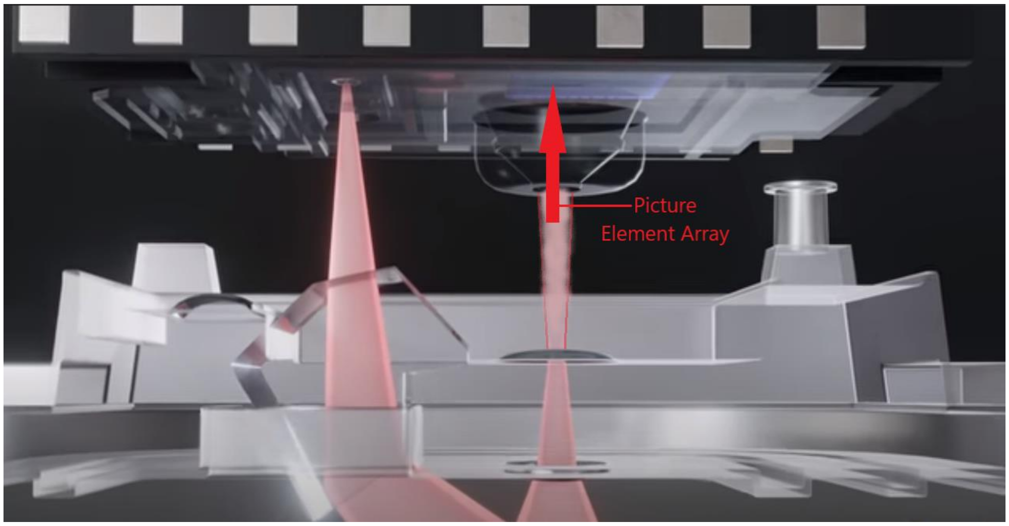

Figure 4.2.6 Picture Element Array captures surface image focused by Lens [27]

Since the picture element array is able to capture multiple consecutive images within miniscule intervals, it leverages a Digital Signal Processing (DSP) block in the optical sensor to compute the difference between these images [25]. This difference between consecutive images is translated to incremental changes in x and y position of the optical sensor, i.e., Δx and Δy [25]. This concept is also illustrated in figure 4.2.7 [28].

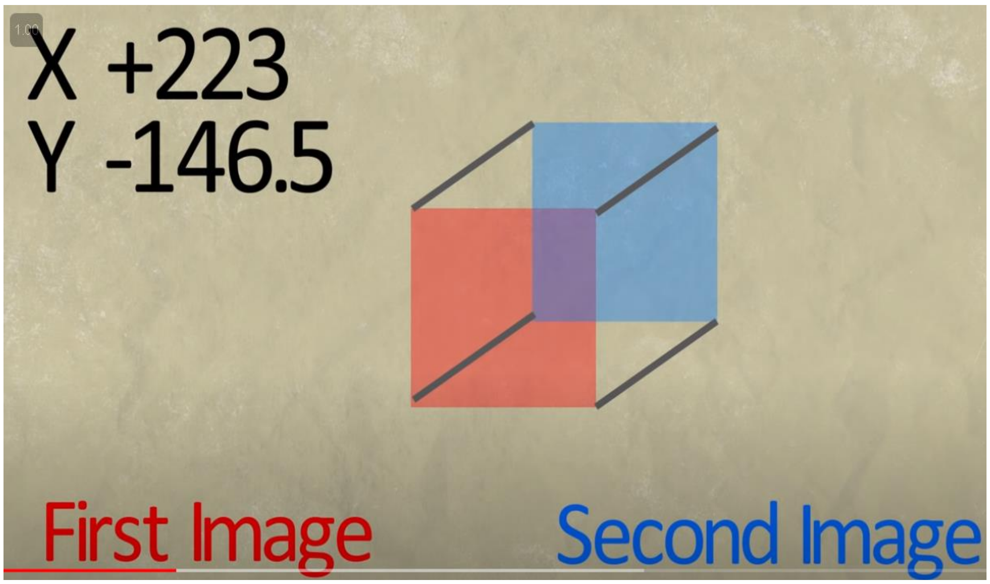

Figure 4.2.7 Image comparison under optical mouse sensor [28]

The PMW3389 has a maximum resolution of 16000 counts per inch (CPI) with step size of 50 CPI [105] [25]. This means that the Δx and Δy data received from the sensor will increment by up to 16000 in steps of 50 if the optical sensor is moved an inch in one direction. This metric is useful for thrust and roll tracking purposes. For instance, if Δx is shown to have incremented by 50 the sensor may have moved 1/320 inches in the positive x direction. However, the optical mouse sensor has a fixed position in this project and it is the surface beneath the optical sensor that moves, i.e., the surface of a laparoscopic tool. As such, the Δx and Δy data from the sensors inform the roll and thrust measurements of the laparoscopic surgical tool.

Data from the optical sensor, i.e., Δx and Δy measurements, were read in by the microcontroller using SPI protocol as the optical sensor is only compatible with this protocol. The viability of each optical sensor was tested using the circuit in figure 4.2.8, which also shows the use of a prototype to test the functionality of the sensor with a potential laparoscopic instrument. This prototype was built using a generic pill box, which served as a replacement for a trocar, and a wooden chopstick, which was a replacement for a laparoscopic instrument. Analysis using this prototype helped visualize potential Δx and Δy data that could be obtained from the optical sensor when a laparoscopic instrument is thrusted into the PLS simulator or rotated on its axis.

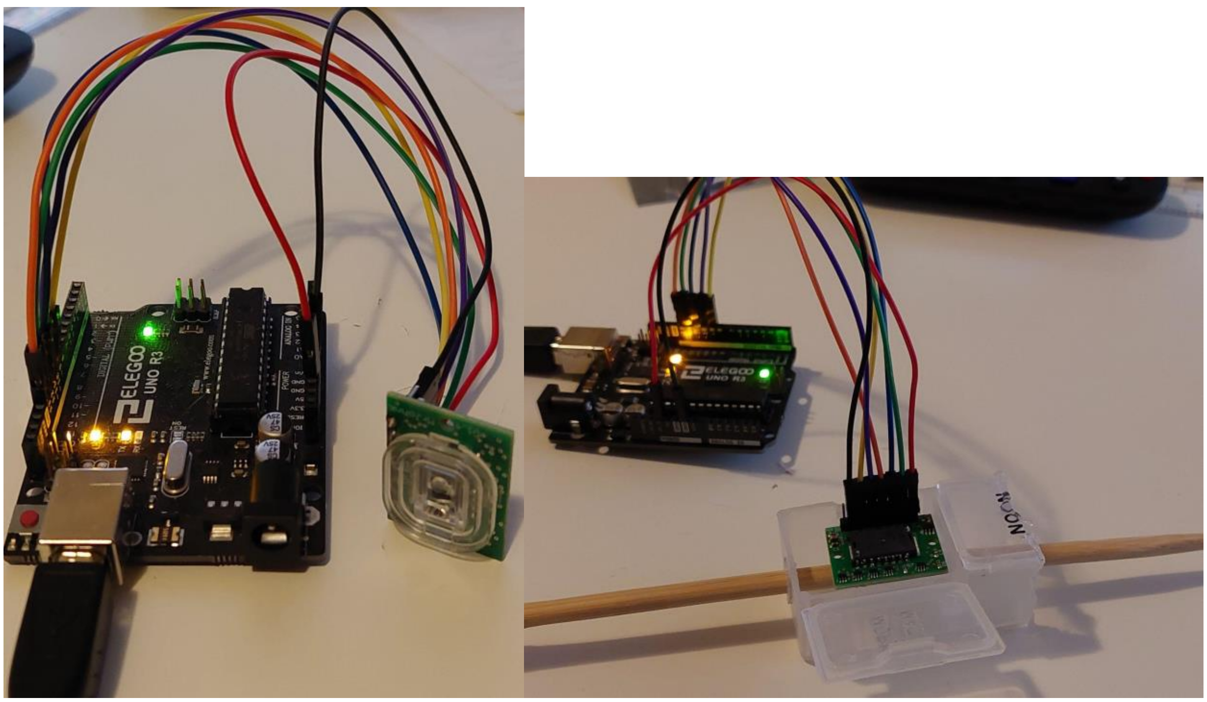

Figure 4.2.8 Optical sensor testing circuit and analysis prototype

4.2.3 Integration of Sensors in Microcontroller

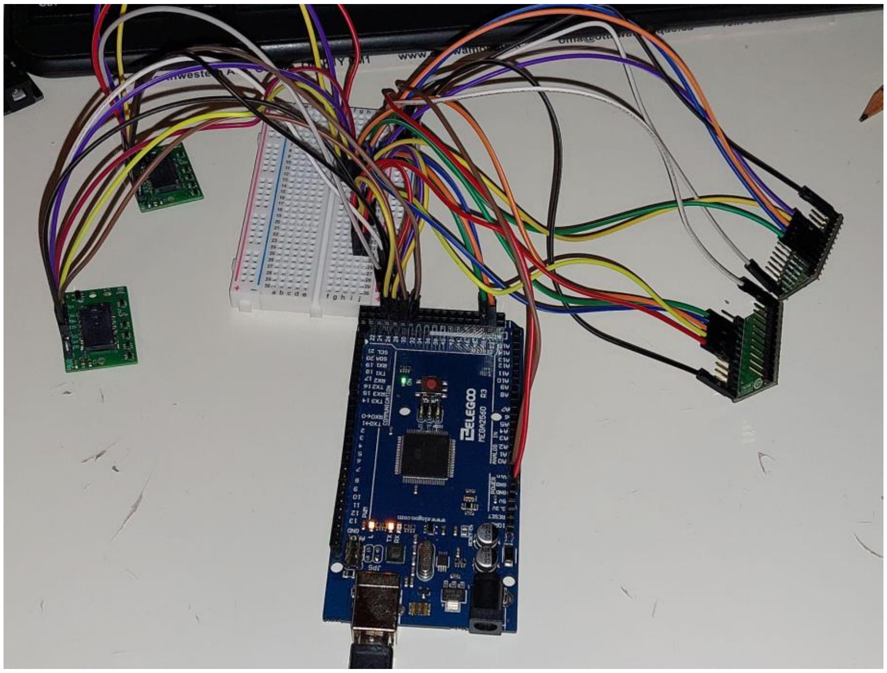

Figure 4.2.9 SPI sensors integration testing circuit

In the previous subsections, both the IMUs and the optical sensors were tested individually. However, the project required the microcontroller to receive data from both types of sensors at the same time. Therefore, a circuit was built to integrate both sensors in the microcontroller such that it receives synchronized data from the IMUs and the optical sensors. Initially, the microcontroller received data from the IMUs and the optical sensors using I2C and SPI protocol respectively. However, there were difficulties synchronizing the received data during integration. This may have been due to the use of separate protocols for both types of sensors because the issue was resolved when data from both sensors were received using SPI. Therefore, SPI protocol was also used to receive data from the IMUs as opposed to I2C. Figure 4.2.9 shows the circuit used to test the integration of both types of sensors using SPI protocol.

An overview of the pin connections for this testing circuit is shown in figure 4.2.10. The Serial Clock Line (SCL), Master-In-Slave-Out (MISO), Master-Out-Slave-In (MOSI), 3.3 V, and the ground (GND) pins for all the sensors and the microcontroller had parallel connections. Each of the IMUs require two separate Chip Select (CS) pins, i.e., one each for the accelerometer (Acc) and gyroscope (Gyro) and the other for the magnetometer (Mag). The optical sensors only require one CS pin each. These CS pins are independent connections and were connected to separate digital pins on the microcontroller.

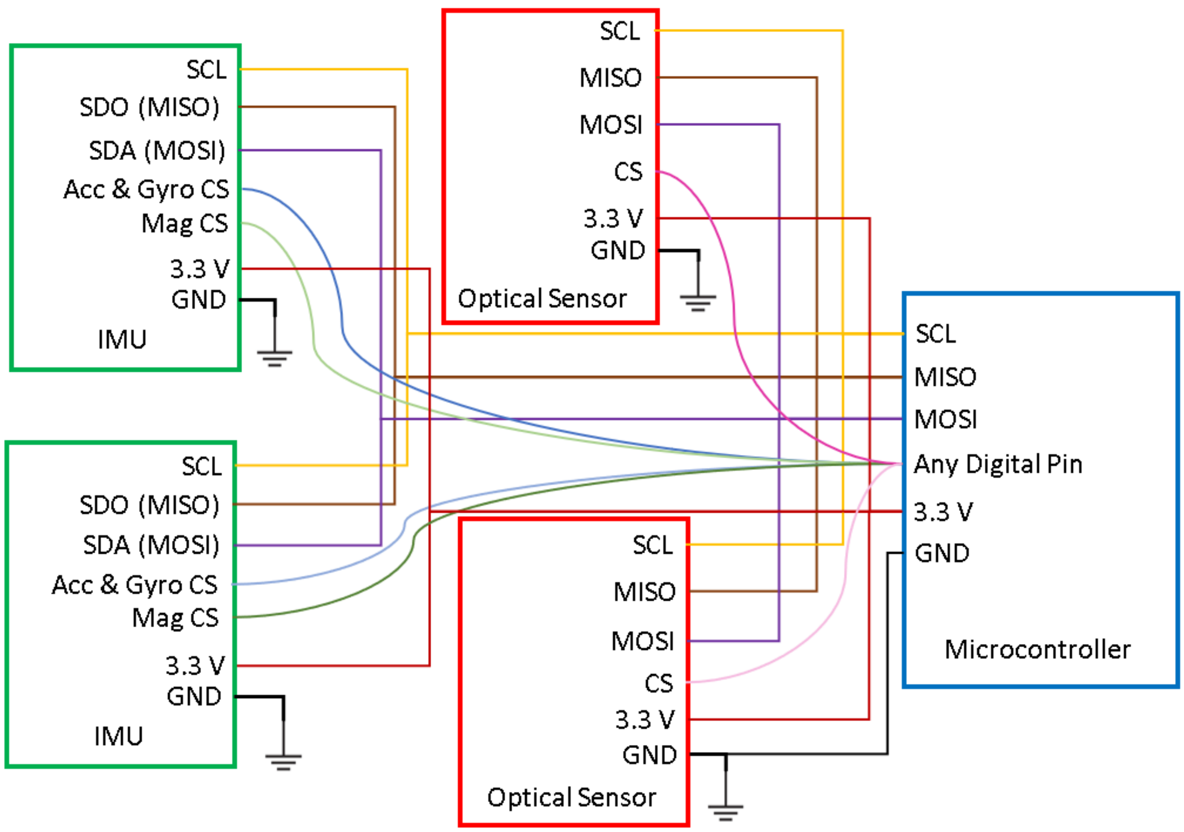

Figure 4.2.10 Pin connections for SPI sensors integration circuit

4.2.4 Custom Trocar Design for Sensor Placement

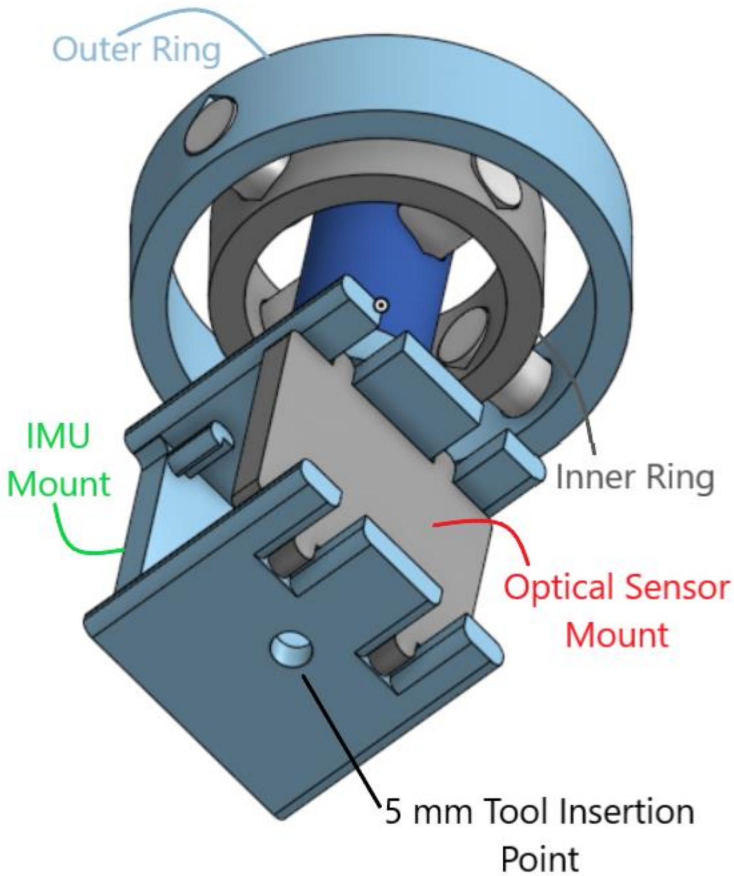

Figure 4.2.11 Original Custom Trocar 3D Model [29]

Once the technique for receiving data from the sensors involved was established, an apparatus was required to place the sensors on the laparoscopic instruments to track their movements. As such, a custom trocar was designed and 3D modeled as shown in figure 4.2.11 [29]. This original model was created by a colleague working on a project that explores augmented reality (AR) applications for PLS simulators. It includes a sensor housing component where the IMU can be mounted on one surface and the optical sensor can be mounted on the sliding surface. The sliding surface allows the positioning of the optical sensor to be adjusted such that the lens sits on top of the laparoscopic instrument.

A tool insertion point allows laparoscopic instruments to be inserted through the trocar and into the PLS box. The custom trocar was designed in this manner because the laparoscopic instruments need to be interchangeable and therefore independent of the sensors. Furthermore, the base of this custom trocar is fitted with an outer and inner ring to allow the surgical instruments freedom of movement with regards to pitch and yaw. For the purposes of this project, the original model had to be modified such that the custom trocar could be attached to the PLS box. As a result, the base of the 3D model was modified as per figure 4.2.12 where the square attachment base could be screwed onto the opening of the PLS box that allows insertion of laparoscopic instruments.

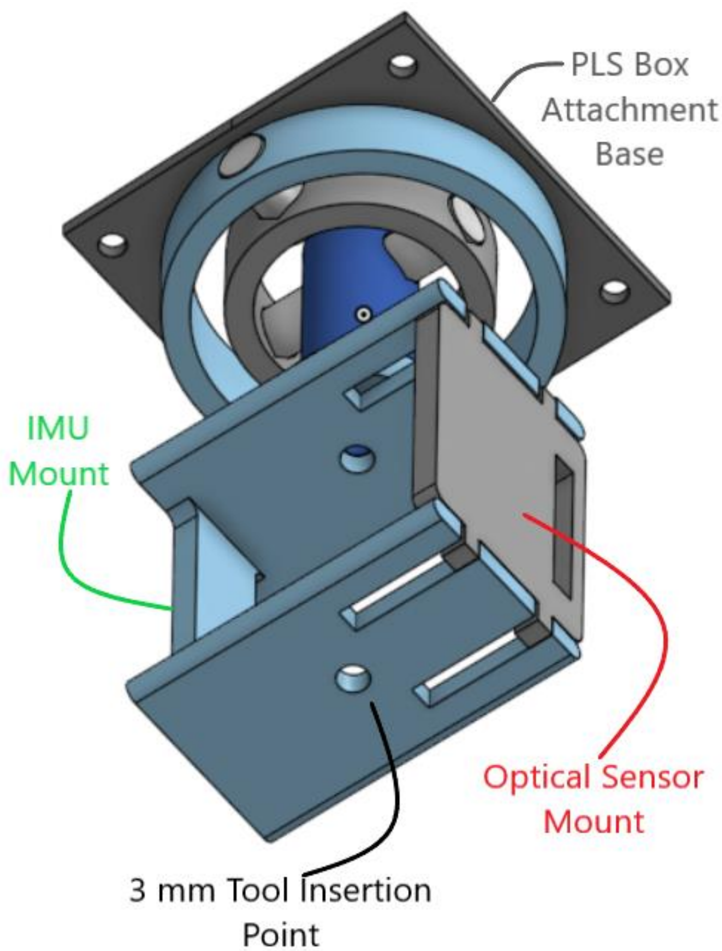

Figure 4.2.12 Modified Custom Trocar 3D Model [29]

The dimensions of the original custom trocar 3D model were also modified to fit the dimensions of the sensors and the laparoscopic tools used in this project. After the above-mentioned modifications were completed, the modified custom trocar was 3D printed as shown in figure 4.2.13. Next, this 3D printed model was then fitted with sensors and attached to the PLS box as shown in figure 4.2.14.

Figure 4.2.13 3D printed custom trocar

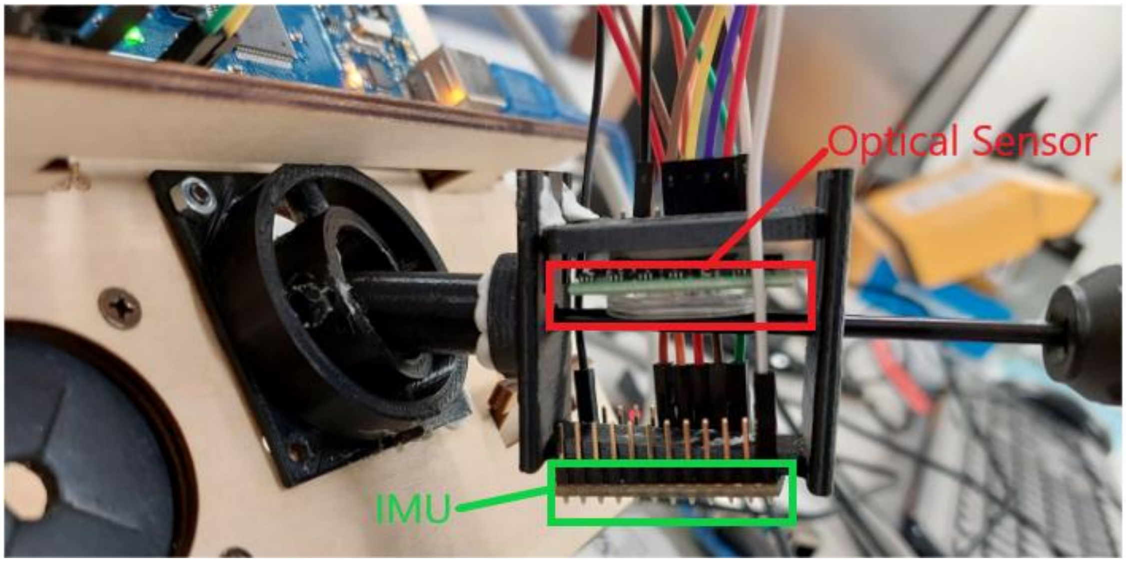

Figure 4.2.14 3D printed custom trocar attached to PLS box

4.3 Haptic Feedback Wearable Design

Figure 4.3.1 High-fidelity robotic 3D haptic controller [30]

Two types of haptic feedback options were considered for this project, forced feedback and vibrotactile feedback. The former would involve the use of a high-fidelity robotic 3D haptic controller, as shown in figure 4.3.1, which updates its motors 1000 times per second to simulate natural 3D haptic feedback offered during real life surgery [30]. Using this device to build a box-type laparoscopic simulator would require the use of complex kinematics. The idea to use such an interface was further demoted as similar interfaces in VR trainers were complained about for adding friction during surgical training [10, p254]. In addition, building a laparoscopic trainer with this device would require two of them, i.e., one for each scope, which significantly drives up the cost and risk involved as they are expensive. Therefore, given the time and budget constraints of this project, the forced feedback option was ignored.

On the other hand, the approach of using vibrotactile feedback via wearables as seen in [14] was considered. The vibrotactile feedback approach involves a wearable fitted with a grid of actuators on each hand. This may feel less intuitive than forced feedback because surgical trainees may have to learn the vibrational patterns to understand the corrective actions required or assess their mistakes. However, the cost and risk involved is significantly reduced and it is slightly easier to implement due to the use of simpler signals. If proven significantly useful, this option offers scalability in the long run. As such, the use of vibrotactile feedback was chosen for this project.

4.3.1 Wearable Design

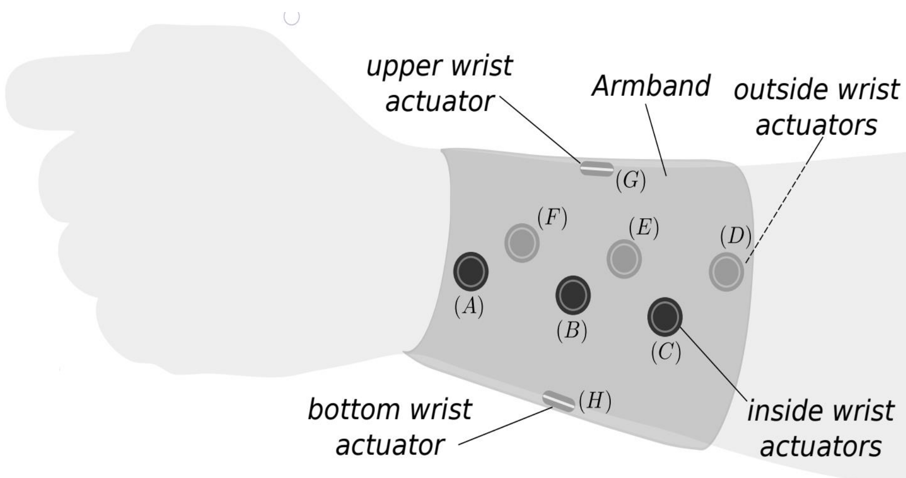

Figure 4.3.2 Positioning of haptic feedback actuators on wearable [14]

For this project, the number of actuators and their positioning on the wearable was directly adapted from [14], as shown in figure 4.3.2. This was chosen because placing actuators in the vicinity of the hands via wearables allows for easier perception of different types and patterns of vibrations as suggested in [31]. In addition, this configuration of actuators allows for the vibrations to be felt more because sensation in skin is greater closer to the joints and when more area of the skin is covered [32], [33].

This design is also advantageous because it keeps the fingers and hands free from any obstruction to perform the surgical training tasks. However, initially a modified version of the configuration in [14] was proposed for the project but it was not implemented due to limitations of the Pulse Width Modulation (PWM) driver. As such, the project makes use of a wrist brace wearable, which is in line with what was proposed, as opposed to the armband used in [14]. The original proposed design has been added to section 5.2 as a recommendation for future work on this project.

Figure 4.3.3 Adafruit 1201 mini LRA [34]

Two other types of actuators apart from the Linear Resonant Actuator (LRA) type vibration motors used in [14] were considered for this project. The first alternative type of motors to be considered were the Eccentric Rotating Mass (ERM) vibration motors. These motors would have been heavier and bulkier, thereby making them a bad option for this specific application because configuring them in an array on the wearable would make it heavier. ERM motors also consume more power to produce larger vibrations [35]. As such, the ERM motors were ignored for this project. The next type of motors to be considered were piezoelectric actuators. This option was also ignored as these motors require more layers to induce perceptible vibrations which increases their weight and power requirements [35].

Therefore, the low power consuming LRA vibration motors were chosen for this project because of their small and flat structure, which easily fit in the wearable without making them bulky or heavy. To be specific, the Adafruit 1201 mini LRA motors, as shown in figure 4.3.3, were used as the actuator for this project [34].

4.3.2 Haptic Motors Implementation

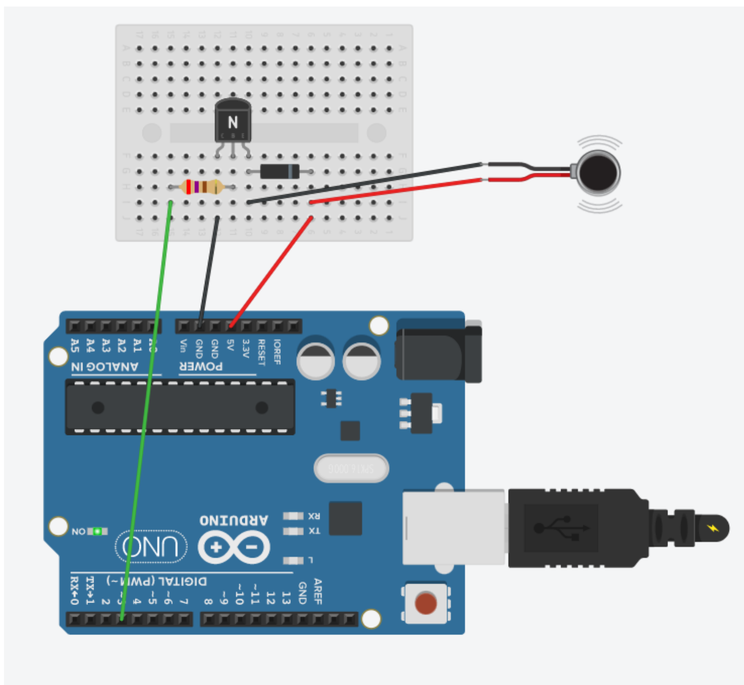

Figure 4.3.4 LRA testing circuit [36]

At the beginning, the LRA motors were tested using the LRA testing circuit, as shown in the simulation figure 4.3.4. This circuit and the simulation diagram were built using the circuit tutorial and Arduino code provided by Adafruit [36]. A resistor, diode, and a PN2222 transistor was used to drive the motor since the motor cannot be directly driven using the PWM pins of the microcontroller. These components ensure that little current from the Arduino was used to control the motor and damage to said PWM pins were avoided.

As seen from the LRA testing circuit, implementing multiple motors using the PWM pins would require more of the above-mentioned components per motor. This would make the final circuit involving sixteen motors messy and prone to errors. Therefore, a PWM driver module was leveraged to avoid using many components in the circuit. The on-board circuitry of the PWM driver handled all the power supply needs of the connected motors without putting any Arduino pins at risk. Thus, the need for additional components is eliminated. To be specific, the PCA9685 16-Channel 12-bit PWM driver module, as shown in figure 4.3.5, was used for this project [37]. This module allowed sixteen motors to be controlled simultaneously with varying input from the microcontroller via I2C protocol [38].

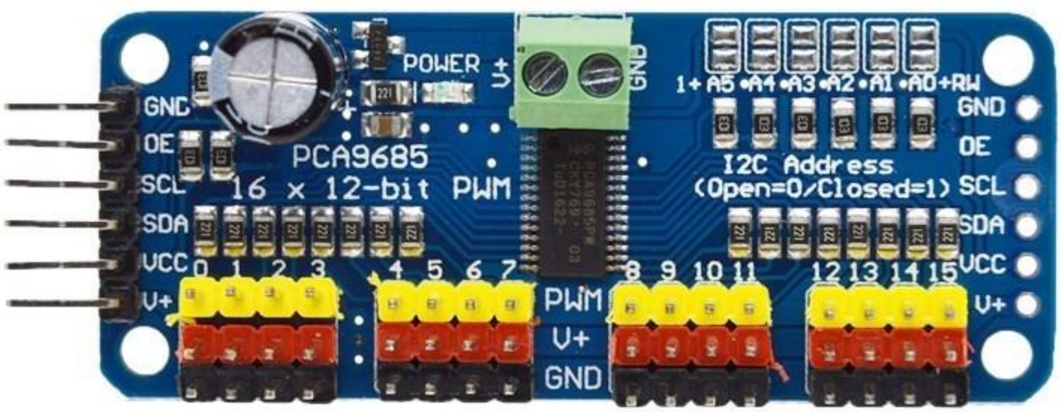

Figure 4.3.5 PCA9685 16-Channel 12-bit PWM driver [37]

Next, the wiring of the motors to the PWM driver posed some challenges. The connection from the motors to the PWM driver needed to be long such that attachment of motors to the wearables allows surgical trainees freedom of movement during surgical training tasks. As such, 9-conductor multi conductor cables were leveraged to elongate said connection. By default, the wires on the vendor provided motors were short and fragile. This made the task of connecting the default motor wires to the multi conductor cable difficult.

Therefore, the default motor wires were elongated by soldering them to cut pieces of Arduino jumper wires as shown in figure 4.3.6. This elongation made the task of connecting the motors to the multi conductor cable easier. Each connection was tested for continuity using the multimeter and all exposed wiring was later insulated via heat-shrink tubing.

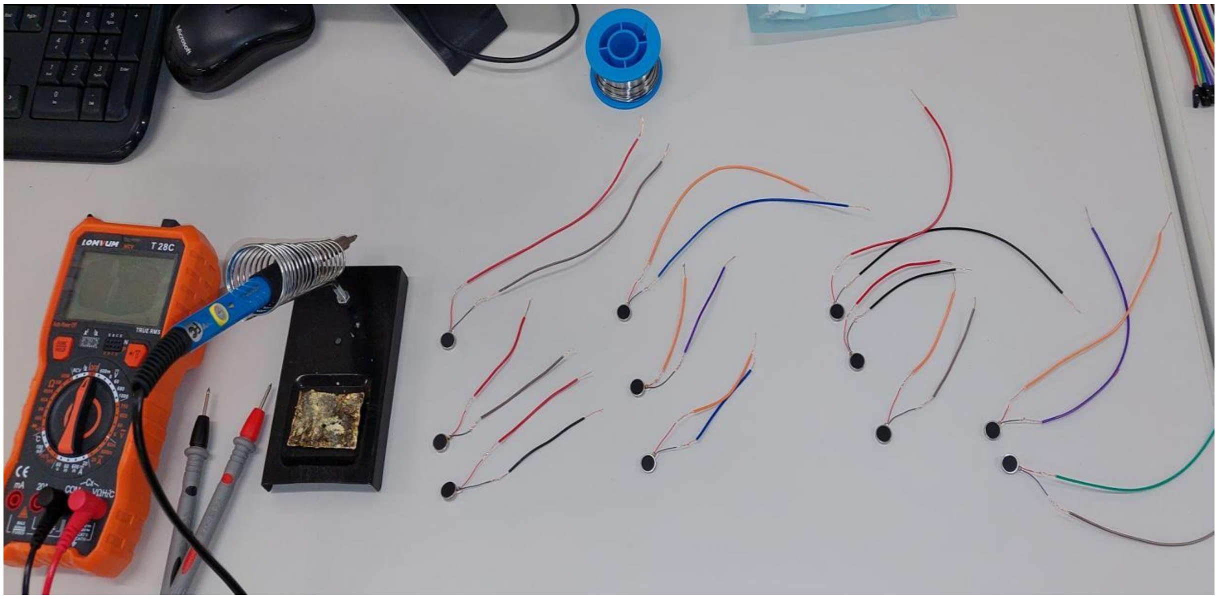

Figure 4.3.6 Elongation of default motor wires

As shown in figure 4.3.7, for attachment to one wearable, eight of the above modified wires were soldered to one end of a 9-conductor multi conductor cable. To form this connection, all the ground terminals of the motors were connected to a single wire on the multi conductor cable and power terminals of the motors were connected to individual wires of the multi conductor cable. Similarly on the other end of the multi conductor cable, the ground terminal connected wire was soldered to a male lead Arduino jumper wire and the others were connected to female leads for PWM driver connection purposes. Next, the PWM driver module was connected to the microcontroller via I2C protocol. The input to the PWM driver and output to the motors is further discussed in the following subsections.

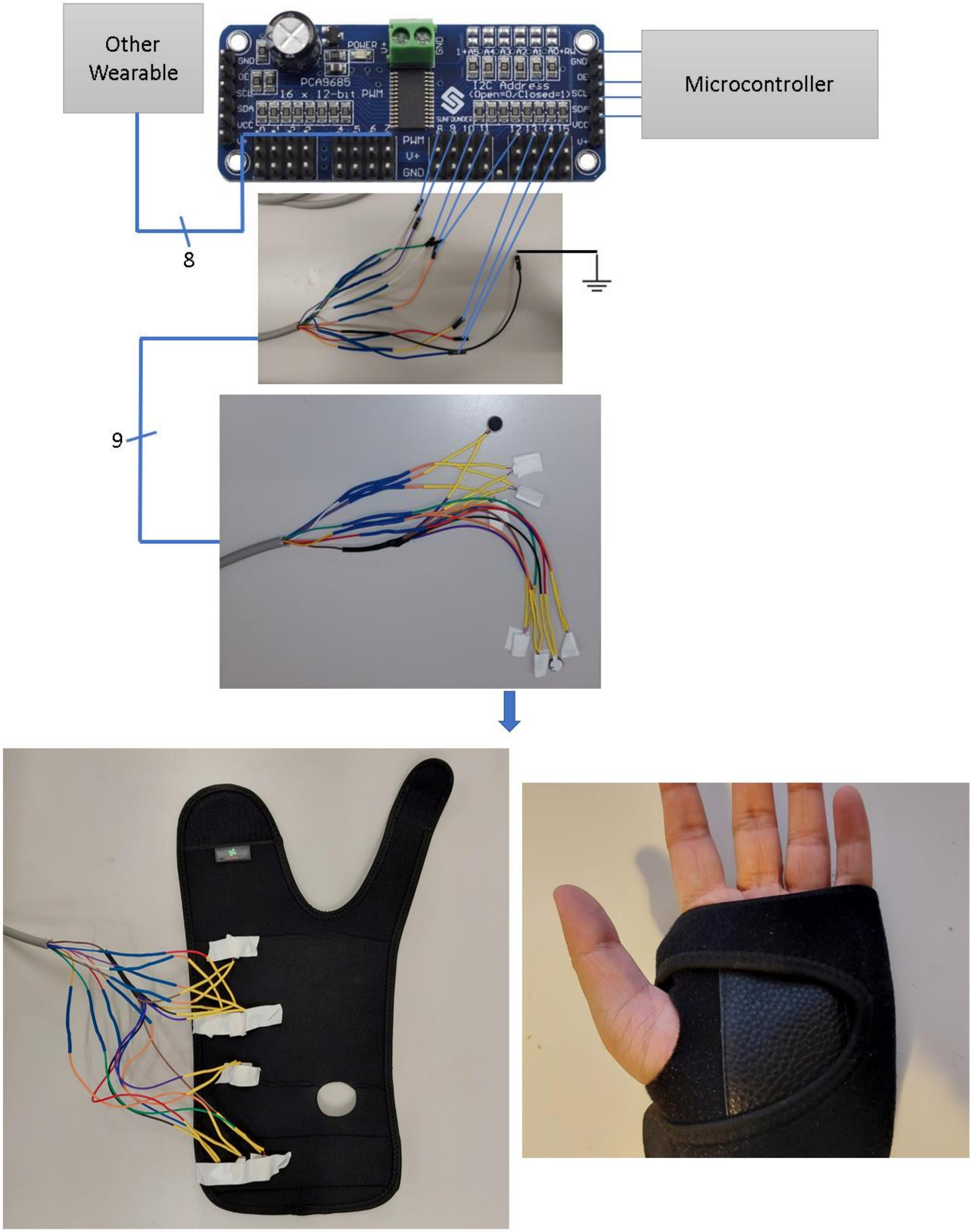

Figure 4.3.7 Wiring of motors to wearable via PWM driver

4.4 Feedback Delivery

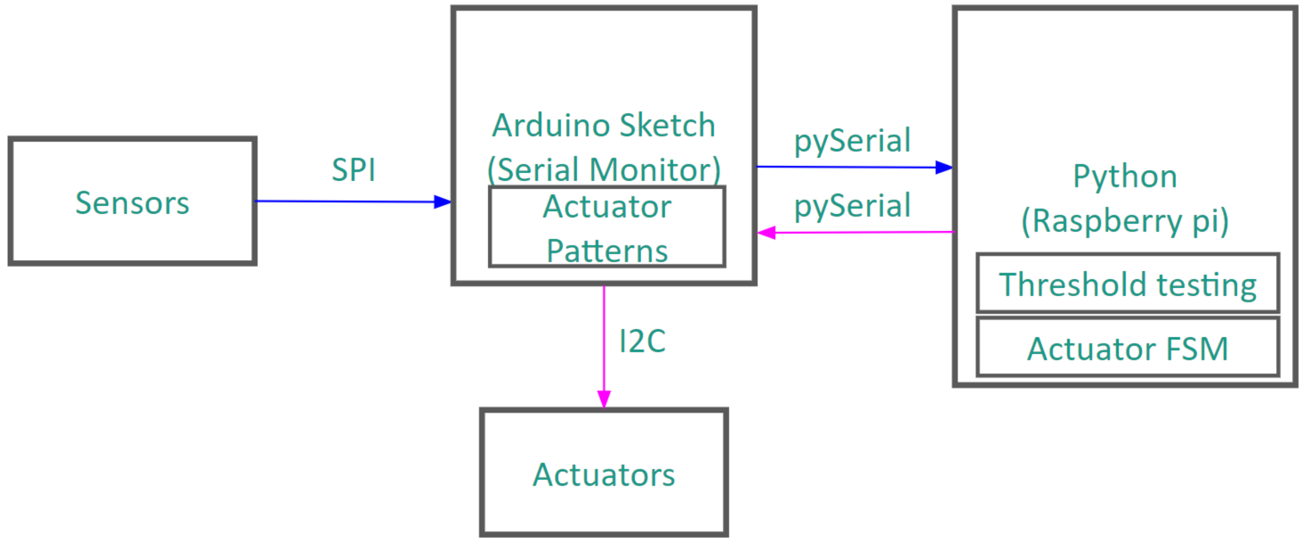

Figure 4.4.1 High level schematic of system interface

After integrating the sensors and the haptic wearable into the simulator, the next step was to create a decision-making system that uses the data from the sensors to determine whether feedback is warranted, and signals the microcontroller to activate the corresponding actuators. Figure 4.4.1 shows a high-level schematic of electronic components of the PLS and how they connect together. As shown, the sensors send data on the position and orientation of the laparoscopes to the Arduino using a combination of I2C and SPI communication protocols. The Arduino program then receives this data and sends it over to a Raspberry Pi which is a small single board computer used to activate, deactivate, and reprogram the simulator. The Raspberry Pi communicates with the Arduino through a serial connection by reading and writing using the PySerial[3] Python serial connection library. Next, the Raspberry Pi uses Python to perform threshold testing on the sensor data and tracks any violation through an actuator finite state machine.

[3] A python module that enables Python to access and use data from the serial port [39]

The state machine ultimately decides whether feedback is warranted, and sends a message through the serial connection to the Arduino, indicating the feedback required. The following subsections will begin by going into detail on the threshold testing within the Pi, followed by a thorough description of the actuator state machine. Finally, the feedback patterns and the Arduino actuation code will be discussed.

4.4.1 Threshold Testing

During training, the user is asked to perform specific tasks using the PLS simulator instruments while equipped with the haptic feedback wearable. The data acquired by the sensors during the tasks is then processed by the microcontroller and inspected in real-time by a Python script that works to ensure that the movement of the instruments involved stays within specific bounds. The thresholds are set based on the bounds defined by the area of interest of the task at hand. If the laparoscopes leave the trajectory of the area of interest, this constitutes a threshold violation; the state machine is updated accordingly, providing an effective mechanism for the detection of threshold violations. The area of interest defined is based on 2D rectangular area encompassing the 'operating area' of each laparoscope as shown in the image below.

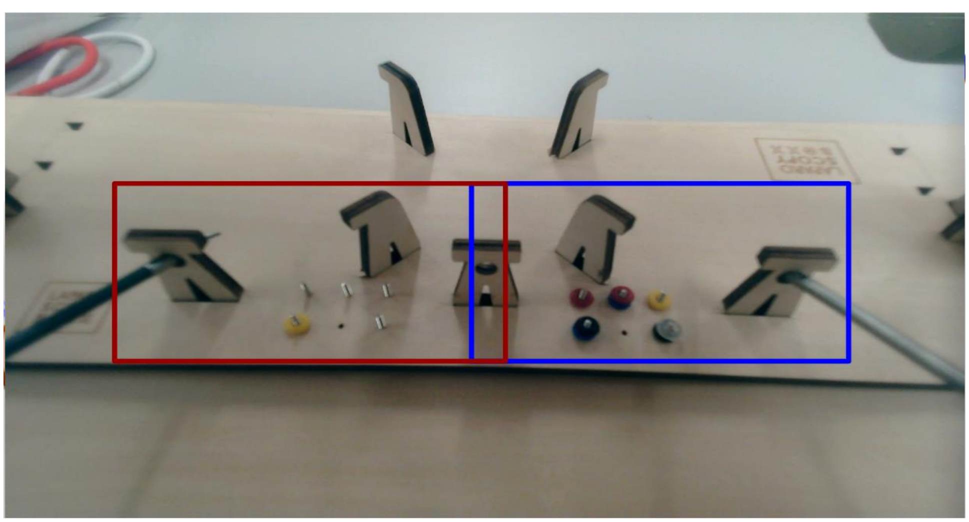

Figure 4.4.2 Rectangular areas of interest for object transfer

For this task, each laparoscope is not meant to leave the trajectory leading from its point of entry into the box to its corresponding rectangular area of interest. A small overlap between the regions exists to allow for object transfer without triggering any violations. To quantitatively map the trajectory so that threshold values can be added, the acceptable trajectories of the laparoscopes are mapped as oblique pyramids, the apex being the point of entry of the laparoscope and the base being the rectangular area of interest as illustrated in the image below.

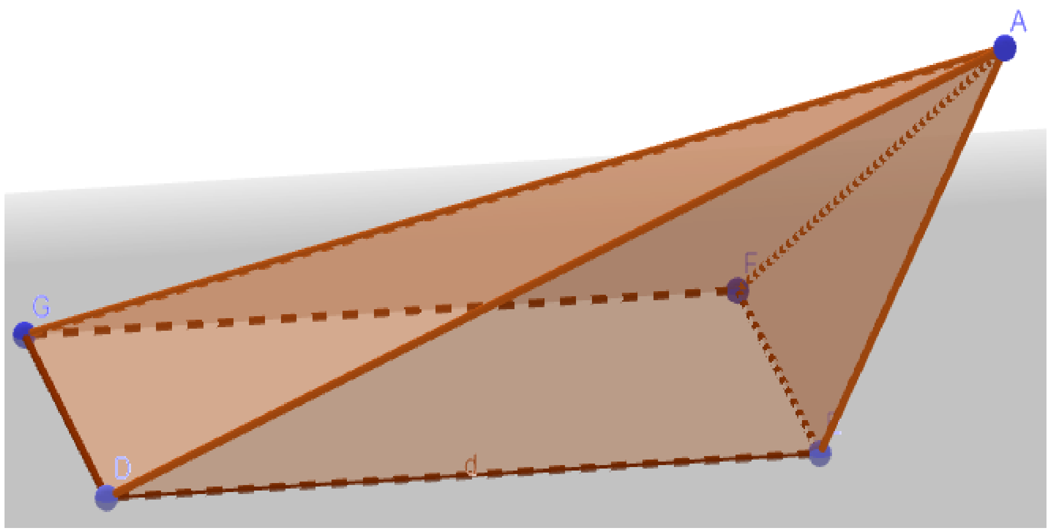

Figure 4.4.3 Oblique pyramid representing the laparoscope trajectories

After defining the allowable trajectory, getting the thresholds becomes a simple matter of geometry as the thresholds are defined by values that are attainable from the pyramid. Plugging the dimensions of the simulator into the pyramid allows us to determine static and dynamic thresholds based on the sensor data. The PLS simulator only allows for 4 degrees of freedom for movement thus thresholds need to be defined for each of these degrees of freedom. To stay within the pyramid, the minimum and maximum values for pitch are constant and come from the angle between the AGF and ADE planes in Figure 4.4.3. From the yaw perspective, the thresholds are also constant, and depend on the angle between AGD and AEF planes in the figure.

Next, the surge (depth) threshold is set so that the laparoscopes stay within an arbitrary distance from the base of the simulator. This makes the surge threshold dynamic, as it is a function of the current pitch and yaw of the laparoscopes. Finally, the roll thresholds are more flexible as the roll of the instrument has no significant effect on the object transfer task. As a result, the roll thresholds are decided based on arbitrary parameters that are derived by calibration. The Python script essentially monitors the position and orientation of the laparoscopes for any violation of the aforementioned thresholds, and sends the result of the threshold test to the state machine.

4.4.2 The Python Script

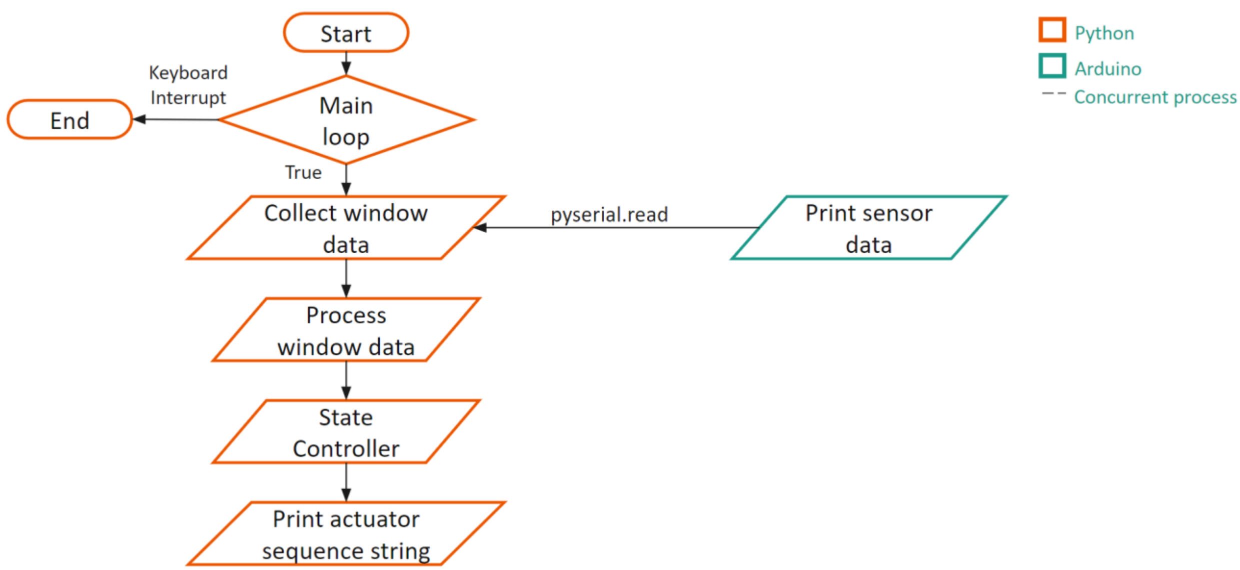

Figure 4.4.4 Flow chart describing the steps taken by the Python script

Before getting into the inner workings of the state controller that controls the actuators, the structure of the Python script should first be discussed. After running the script, an infinite loop is entered which only ends after a keyboard interrupt is detected. This allows the script to continuously repeat its processing on every sample of sensor data received. The main loop begins by collecting enough samples of sensor data from the Arduino to fill a window.

A window size of 3 was selected for the Python script, as it allows the script to determine the direction of motion while also not being so large as to cause delay. In this stage, the script listens for new samples that may appear on the serial port. The sample is then validated by checking the length and the data types of each entry within the sample. After collecting 3 valid samples, a window is formed by sending the samples into arrays that indicate which hand the window corresponds to. At this point, the processed window becomes ready to be received by the state controller. This part of the process is outlined in the form of a flowchart within figure 4.4.4 at the beginning of this section.

4.4.3 The State Controller

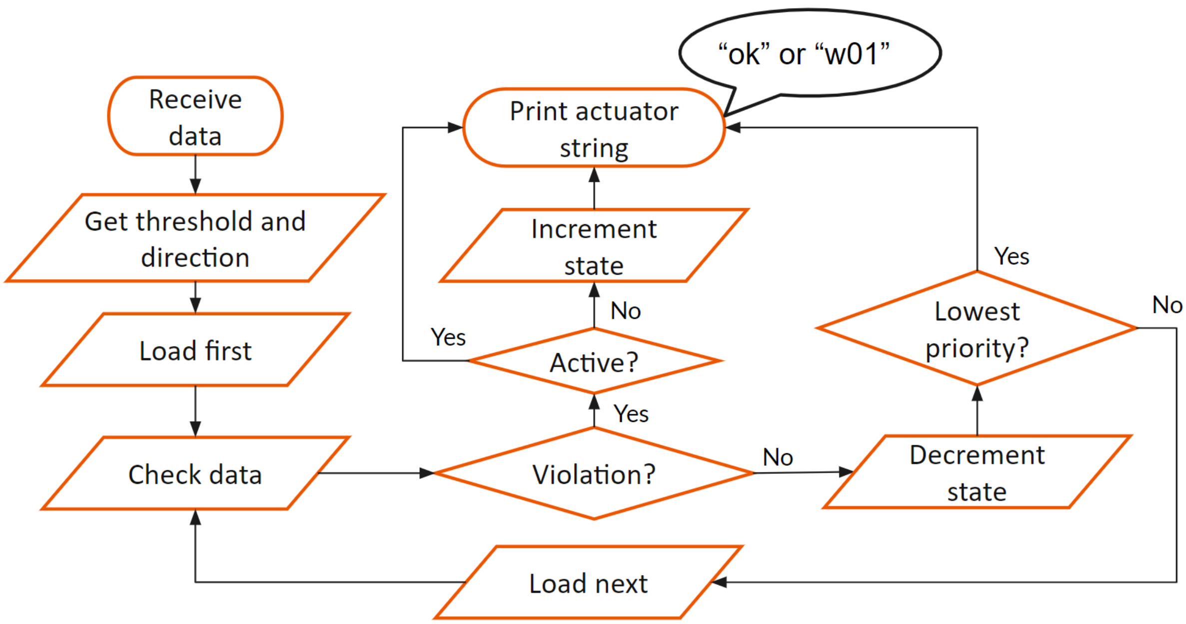

Figure 4.4.5 Flowchart depicting the state controller portion of the script

After preparing the windows, the data is ready to enter the state controller. As shown in figure 4.4.5, the state controller starts by receiving the windowed data and calculating the corresponding surge threshold for that window based on the average values for pitch and yaw within the window.

The direction of motion for each degree of freedom is also determined by comparing the position and orientation of the laparoscope at the beginning of the window with the data at the end of the window. This is important as the direction of motion affects the haptic feedback patterns triggered by the state controller.

Since the system supports 4 degrees of freedom for motion, a priority system is needed so that more important violations are allowed to interrupt feedback that is determined to be less critical. The chosen priorities from highest to lowest are as follows: Surge, yaw, pitch, and roll. The priorities were chosen based on importance during the object transfer task as well as safety during a real procedure. Since the instruments can pierce the patient if pushed too deep, it was determined that surge needed to have the highest priority. Similarly, movements in the roll axis seemed to be the least likely to cause irreversible damage, thus roll was deemed to have the lowest priority. Extreme movements in the pitch and yaw axes seemed to be equally likely to cause damage in a surgical environment. However, the object transfer task involves more longitudinal movement than lateral movement, leading to the decision to place yaw at a higher priority than pitch.

After the program determines the thresholds and the direction of motion, the first data points in the priority line are loaded and compared against the calculated thresholds. If no violation is found, the program updates (decrements) the current state variable accordingly and loads the next data point in the priority line and repeats this until the lowest priority data point is reached. If a violation is detected at any point, the program checks if any feedback is currently active for the current metric. However, if no feedback is active, then the state controller increments the current state variable and prints an 'ok' message to the serial port if no feedback is currently warranted. If feedback is warranted, then an encoded message like 'w01' is printed instead which triggers a specific feedback sequence when read by the Arduino.

If feedback is already active at this point, then the state controller would print the same message that originally triggered the feedback. The encoded messages printed to the Arduino follow the same format where the message is 3 characters long and consists of a letter followed by 2 binary digits. The letter describes what action the Python code wants the user to make. The first binary digit specifies which arm (0 for right, 1 for left) the feedback is intended for, and the last digit specifies the direction of motion (0 for positive, 1 for negative). In the case of "w01", the program is asking the Arduino to trigger a vibration sequence that asks the user to withdraw their right arm backwards.

Within all of this code, the actual mechanism that decides whether a situation warrants feedback is essentially a group of four simple finite state machines that track violations within different degrees of freedom. As shown in the below figure, each finite state machine consists of a reset/base state and three buffer states before reaching the active state where the feedback is triggered.

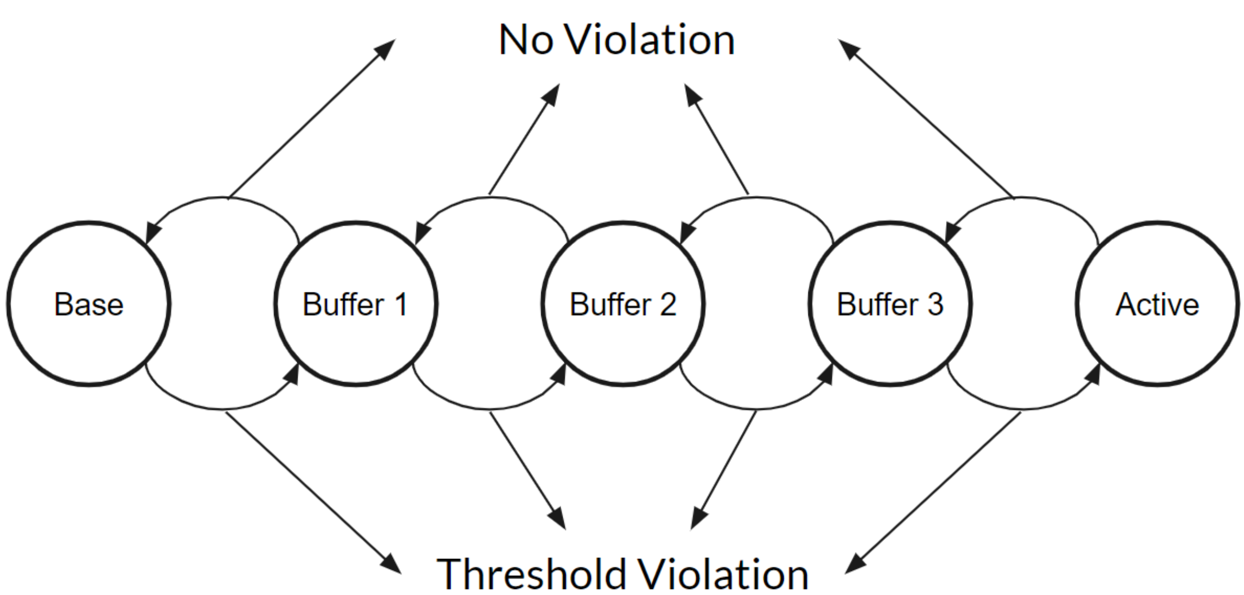

Figure 4.4.6 The finite state machine controlling the actuators

As the state controller finds new violations, the state variables progress forwards towards the actuation/active state. However, windows that are not found to violate the thresholds will decrement the state, causing the corresponding state variable to move backwards towards the base state. During the buffer states, no feedback is activated or deactivated; these states only serve as delay. This mechanism provides a layer of protection against false activations (or lack of activation) due to hazards or glitches and synergizes well with the overall architecture of the state controller. This essentially causes the system to require consecutive violations to trigger feedback, ensuring that temporary violations or those caused by glitches do not falsely trigger any vibration. After reaching the active state, the state variable remains in that state until the violation is resolved, and the feedback pattern repeats until either the violation is no longer detected, or a higher priority state variable reaches the active state.

4.4.4 Haptic Feedback Patterns

After the state controller prints the encoded message to the serial port. The message is received by the Arduino which is running its own infinite loop at this stage and is awaiting the message from Python. In the steady state, the state controller only prints “ok” messages, indicating no violations and thus no feedback necessary. Due to this, the Arduino only needs to listen for changes in the message as the feedback (or lack of feedback) would just repeat itself otherwise. If a change is detected in the string, the Arduino program stops the current feedback sequence (if there is one) and signals the PWM drivers to activate the new sequence and then the infinite loop starts anew. This design is further illustrated by the flowchart in the figure below, which shows the Arduino logic in addition to the Python logic from the previous sections.

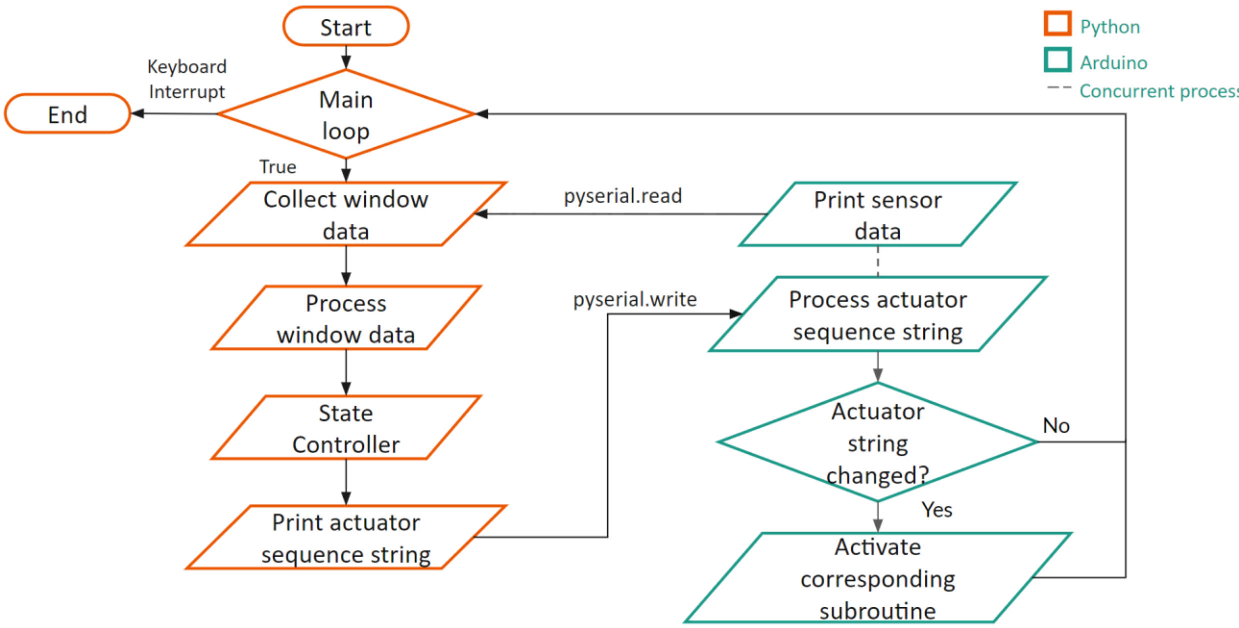

Figure 4.4.7 Complete version of the 4.4.4 flowchart

One of the main goals in terms of project functionality is to be able to administer intuitive feedback that the user can potentially understand without having to spend too much time memorizing the feedback patterns. Otherwise, the simulator could potentially be counterproductive in a sense that more time is spent learning how to use it than the time it saves from a surgeon's training. As a result, the feedback patterns needed to be as intuitive as possible, leading to the current design. It was decided that the vibration patterns would follow the direction of the motion the simulator is asking the user to make. For example, figure 4.4.8 below shows the pattern triggered when the surge of the instrument exceeds the negative threshold. In this example, the user needs to push their arm forward to return the surge to normal values, thus the vibration pattern moves in the forward direction by spending 0.5 seconds in each state of the pattern.

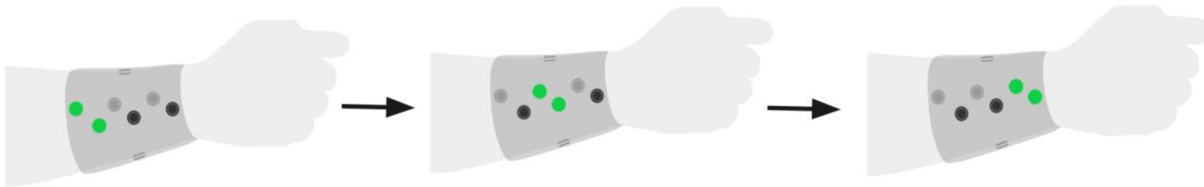

Figure 4.4.8 Haptic feedback pattern example [14]

There are a total of 8 vibration patterns supported by the simulator, 2 for each degree of freedom of motion. The simulator chooses between the 2 patterns depending on the direction of the violation.

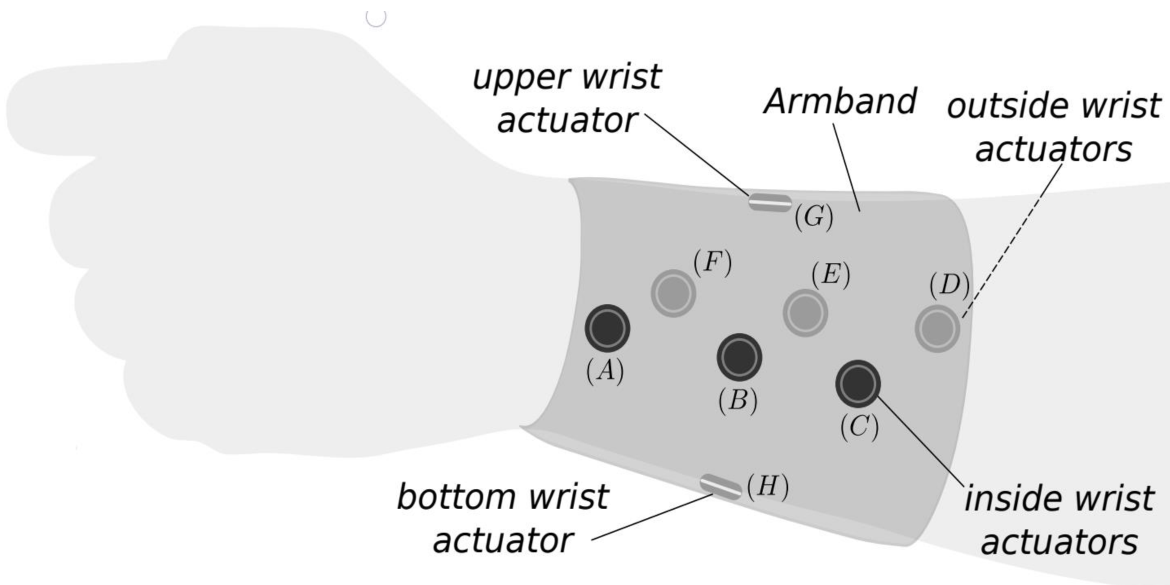

Figure 4.3.2 from page 39 [14]

Consequently, based on actuator labeling from figure 4.3.2 (reposted above), the feedback patterns for surge violations are C->B->A with F->E->D activating concurrently as shown in figure 4.4.8 [14]. For positive violations, the instruction to withdraw is the same as the pattern for advancing except reversed, so the pattern changes to A->B->C + D->E->F. Similarly, the instruction for clockwise roll follows the pattern B->G->E->H which follows the movement requested by the simulator. For counterclockwise movement, this pattern is reversed.

Next, for violations in the yaw axis the instructions are similar to negative violations in the surge axis, but the feedback is only triggered on one side of the arm. To tilt the instrument right, the pattern C->B->A is applied as the hand needs to move left for the instrument to cross over to the right. Similarly, the instruction to tilt left triggers the pattern F->E->D. Finally, instructions in the pitch axis follow the pattern (C+D) -> G for downwards tilt, and (C+D)-> H for upwards tilt.

This implementation makes it possible for the PLS simulator to trigger intuitive feedback, making learning to use the device a simple task. Overall, the way the project software is designed makes making changes or upgrades to the software an easy task as it was implemented with programmability in mind. The Arduino code allows for future developers to easily add new feedback patterns. The Python script also makes adding state variables or tasks into the simulator a simple matter, as all the developer would need to do is add their state variables and their position in the priority line and define the thresholds for the new task.

4.5 Project Development

In terms of project development, most of the actual project implementation occurred during the second/winter semester. Implementation in the first semester was mainly impeded by shipping and logistical delays incurred by the ongoing pandemic. As a result, most of the project work done in the first semester was applied in the form of design work involving the high-level structure of the simulator, as well as some high-level Arduino software architecture which was made to control the sensors. After various shipping delays, the PLS box finally arrived towards the end of January, which marked the beginning of the implementation phase for the project. At this stage, the PLS box trainer was assembled almost immediately, and the Arduino software which controlled the sensors was concurrently in development.

Figure 4.5.1 below shows the state of the PLS simulator at this stage of development, before the sensor software was fully implemented. The simulator was essentially a simple box trainer at this stage and was fitted with a camera and ready for use without any means of automated feedback ready.

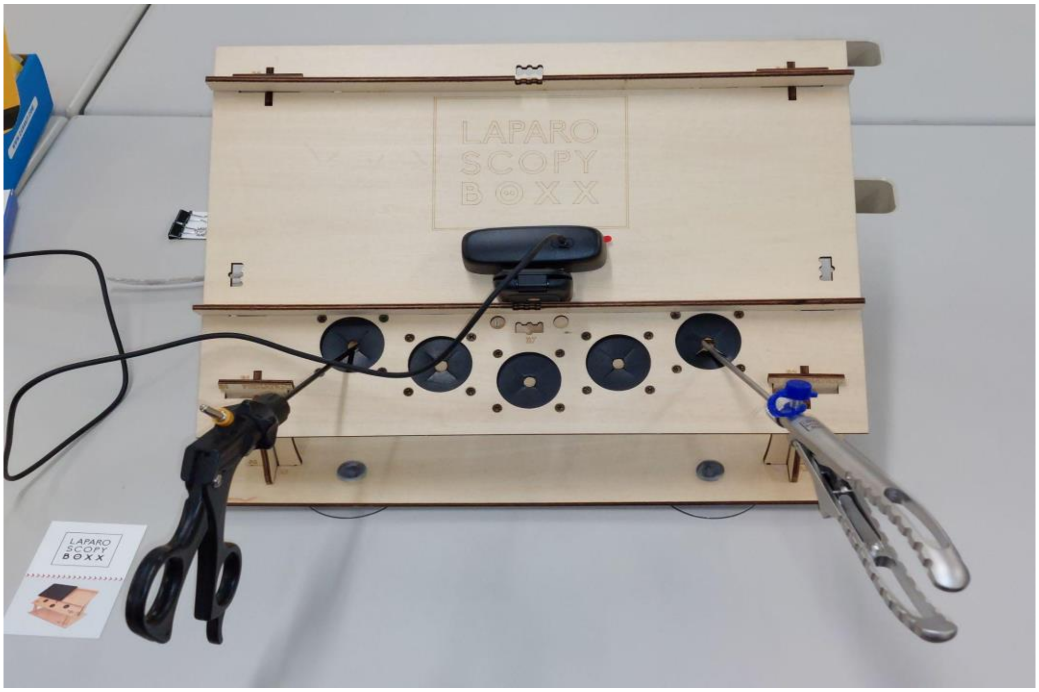

Figure 4.5.1 Initial assembly of the PLS Simulator Box

After assembling the box and beginning the implementation for the sensor software, it was apparent that the project had fallen behind relative to the timeline outlined in the project proposal (see Appendix A). This caused several revisions to be made to the technical requirements of the project so that the deadlines could still be met. A lack of availability from the clinical partners' side made it impossible to perform the thresholding based on the originally intended project design due to both time and logistical constraints. Likewise, a lack of experience with surgical procedures within the group made it infeasible to implement more advanced surgical tasks, leaving only the object transfer task of the simulator which allowed for the thresholding based on a predefined trajectory of interest.

Despite this, the project requirements from the sensor side of the simulator remained the same, and the Arduino sensors were fully integrated into the box trainer by mid-February, leading to the state shown in figure 4.5.2 below.

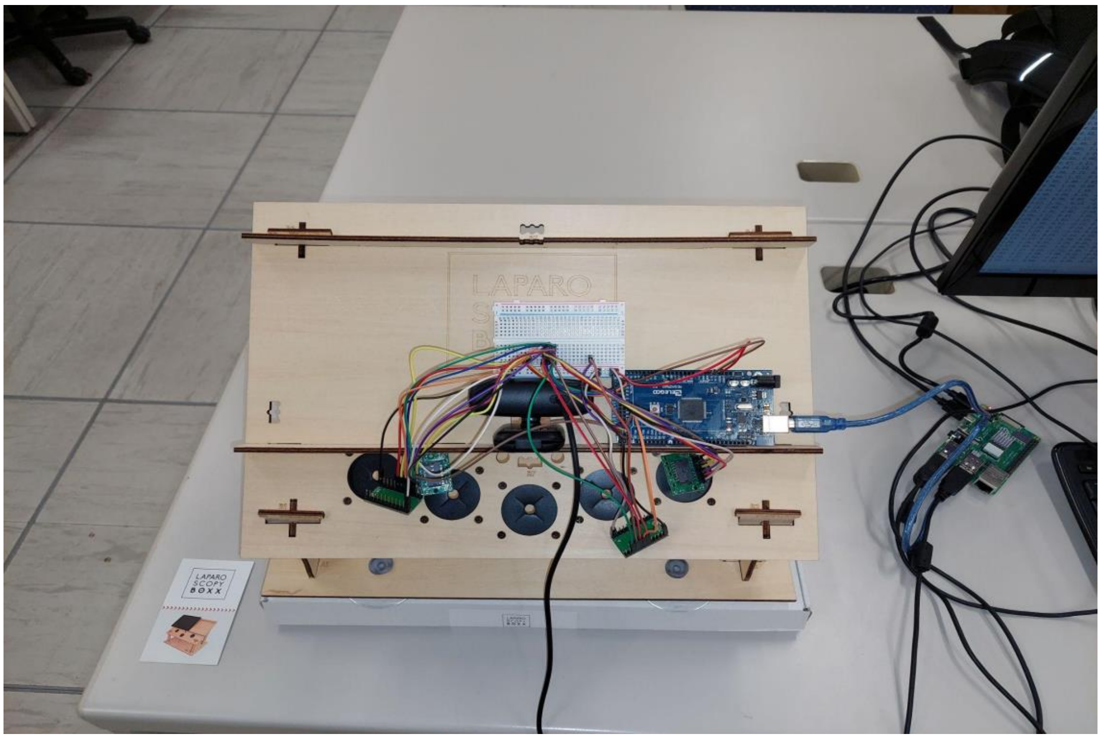

Figure 4.5.2 PLS Simulator after adding the Arduino and the sensors to the box

At this stage, the implementation of the project had approximately reached the halfway mark, and the remaining tasks involved implementing the haptic wearable, the custom trocars, and the Python in charge of programming and controlling the feedback delivery. Around mid-February, both haptic wearables were completed and fitted 8 actuators each. After this was completed, the remaining project components were in their design phases. The state diagram and functionality of the state controller were still in early design stages, and the focus was directed towards repurposing the custom trocar to fit the box dimensions as that was blocking further progression. During this time, the Arduino programs in charge of triggering the motors were also in development so as not to waste time.

By early March, the custom trocars and the actuation code had been fully implemented and the Python code and feedback patterns had passed the design stages and moved on to implementation. Within the week, these last components had finally passed implementation and verification, leading to the final product assembly shown in figure 4.5.3 below.

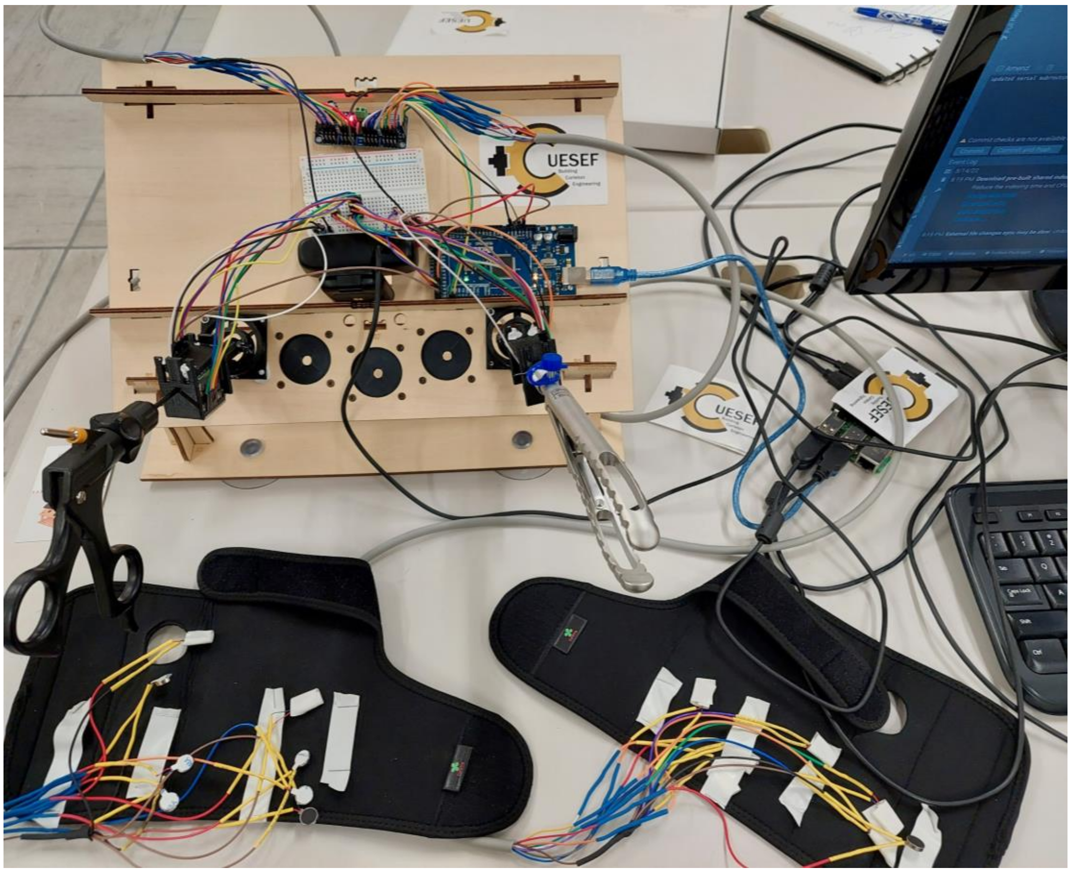

Figure 4.5.3 Final product assembly of the PLS Simulator

5. Recommendations for Future Work

While the PLS simulator that was built serves the purpose of a minimum viable product well, both the simulator and the haptic feedback mechanism implemented into it can be expanded upon. This can include improvements and additional features which are both discussed in the following sections. Since some of the original ideas for the simulator could not be implemented, some of the potential improvements were mentioned earlier in the report and will be expanded upon in this chapter of the report.

5.1 Recommendations for Improvements

As mentioned, while the implemented PLS simulator meets the minimum requirements for the project, it is far from being a market-ready product. There are several areas where the simulator has room for improvement. For example, the quality of the sensors chosen for this project needs attention. The optical sensors in particular provide inconsistent data for the depth and roll of the instruments, which causes issues during threshold testing in the Python code as the actual surge and roll of the instruments cannot be accurately determined. The Arduino also has room for optimization, as the rate of output of the sensor data to the serial connection seems to vary in length, likely due to the runtime of the program. This can sometimes cause delays when this sampling delay is combined with the delay caused by windowing. These runtime delays can also have adverse effects on the response time of the feedback administered and can even cause the duration of the vibration patterns to vary.

Another significant improvement that is worth implementing in future work is the implementation of dynamic thresholding that changes the allowable bounds based on the profile of the current user or based on a calibration process that automatically determines the bounds for feedback. The improvements to thresholding can also be taken a step further by basing the thresholds on a quantitatively mapped ‘ideal’ procedure, which allows the system to personalize the thresholds by recognizing specific actions within the task such as grabbing, moving, or handing over the object in the task at hand. For example, having the system detect when a laparoscope is currently holding an object and changing the thresholds once it detects that the object has been picked up to guide the laparoscopes towards one another.

Additionally, digital signal processing methods can be very suitable in cleansing the sensor data of any noise or unwanted artifacts. The two options for measuring the pitch and yaw of the instruments are using the gyroscopes and the magnetometers within the IMUs. Neither of these options are immune to noise as the magnetometers are susceptible to electromagnetic interference and the sensitivity of the gyroscopes can cause the sensor output to reflect motion artifacts. A combination of suitably designed high-pass and low-pass (or band pass) filters would reduce the impact these have on the overall product. The filtering was not implemented in the project due to faults within the sensors that overshadow most of the noise the sensors should normally experience. In other words, this improvement should be explored after replacing the faulty sensors.

Finally, a Python-based GUI (Graphical User Interface) would be a worthy quality of life improvement with respect to the programmability and user-friendliness of the simulator. Such a GUI could be used to set parameters for the Python script such as the threshold tolerance or even choose between tasks if new tasks are implemented into the simulator. This could potentially be used to allow the user to fully reprogram the simulator to their own desired specifications, without the need to even look at the Python code.

5.2 Recommendations for Additional Features

In the discussion of the haptic feedback wearable design, it was mentioned that modifications to the number of actuators used and their configuration was initially proposed but not implemented. These modifications included adding two additional motors on the wearable such that they sit on the inside and outside of the thumb basal joint, also known as the carpometacarpal (CMC) joint. Adding these two motors as shown in Figure 5.2.1 can allow the implementation of more haptic feedback patterns which may guide the surgical trainees more intuitively. Therefore, it is recommended that two or more motors be added to the haptic feedback wearable along the lines of Figure 5.2.1 for implementation of more intuitive haptic feedback patterns. This can be done using PWM pins with the PWM driver or daisy chaining multiple PWM drivers to control more motors.

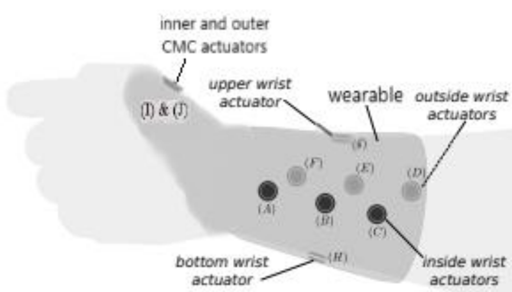

Figure 5.2.1 Originally proposed concept for actuators in haptic feedback wearable[4]

[4] Figure 5.3.1 was adapted from [14]

Furthermore, the current haptic feedback mechanism of the PLS simulator box was developed using a single surgical training exercise kit, i.e., object transfer surgical training exercise. Additional surgical training kits, as shown in Figure 5.2.2, for tasks such as laparoscopic needle threading, cutting, and suturing are provided with the PLS box [17]. Training for these tasks would also benefit from the currently developed haptic feedback. As such, it is recommended that additional thresholds be established specific to these tasks such that surgical trainees can be intuitively guided via the haptic feedback to perform these tasks correctly. This may require implementation of additional motors as previously recommended to enable more types of haptic feedback patterns depending on the task being performed.

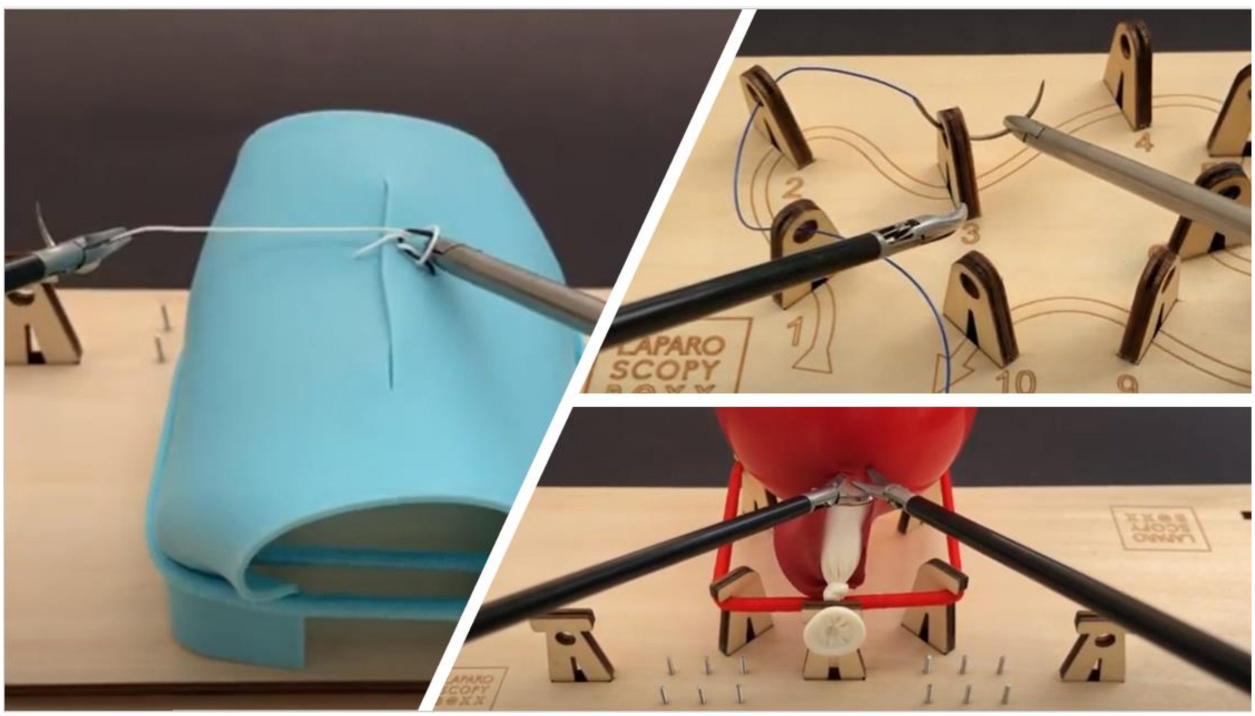

Figure 5.2.2 Suturing, needle threading, cutting PLS training kits [17]

In addition, the weight of the wearables is increased due to the wires and multi conductor cables used to currently attach the motors to it which may be a hindrance during surgical training. Such hindrances will further increase with the addition of more motors in the same manner. Therefore, a solution by means of attaching the PWM driver to a Bluetooth module as shown in Figure 5.2.3 is recommended [14].

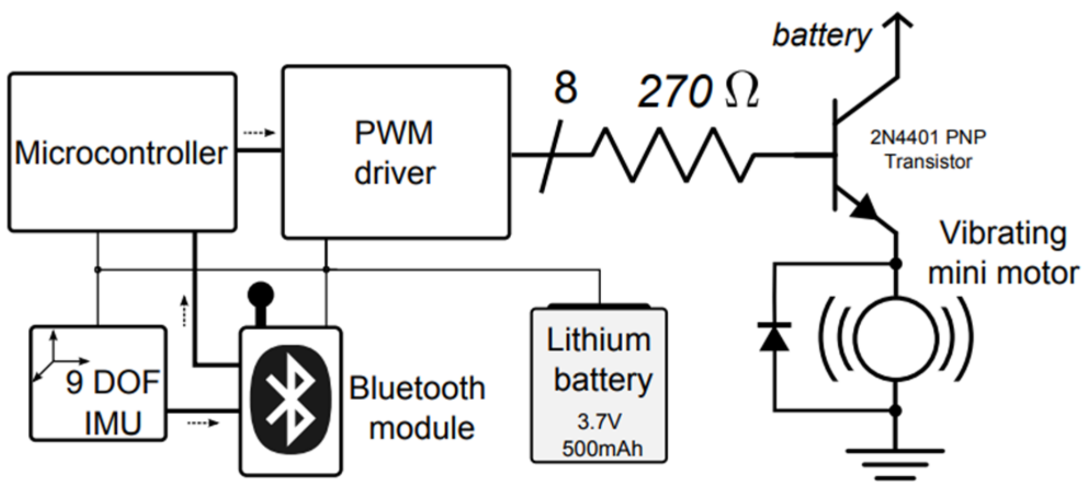

Figure 5.2.3 Bluetooth solution for haptic wearable [14]

Moreover, deviation from thresholds of a training task in the current PLS simulator elicits a binary response from the motors of the haptic feedback wearable. To be specific, these motors are either on when the instruments deviate from the threshold, regardless of the extent of deviation, or off otherwise. Therefore, it is recommended that the vibration strength of the motors increase in proportion to the extent of deviation of the laparoscopic instruments from the threshold. This can be done by making the PWM frequencies and duty cycles of each motor dependent on the deviation of thresholds.

Finally, adding force measurement sensors to the PLS simulator and the laparoscopic instruments using flex sensors such as the one shown in the left image of Figure 5.2.4 is also recommended [40]. This is because a study has shown that training with force parameters benefits laparoscopic surgeons of various experience levels [41]. In many surgical simulators, force parameters are also obtained using a 3D mouse [16], [42]. Therefore, exploring the use of a 3D mouse, as shown in the right image of Figure 5.2.4, to assess force parameters during surgical training tasks in the current PLS simulator is also recommended [43].

Figure 5.2.4 Flex sensor and 3D mouse [40], [43]

6. Conclusion and Reflections

In conclusion, a basic PLS simulator was built for this project while enabling it with the capability of offering vibrotactile haptic feedback. As such, most of the metrics of success outlined in the proposal were met (see subsection 4.7 of Appendix A). For instance, the proposed metric of success for the PLS simulator was defined as its ability to gather information from all the sensors by means of a microcontroller. This metric of success has been achieved as data from both the IMUs and optical sensors is consistently read in by the microcontroller. In addition, the proposed metric of success for the haptic feedback wearable was defined as the ability of individual motors to vibrate in specific patterns given specific input. This metric of success was also achieved by means of programming inputs to the PWM driver using the microcontroller.

Lastly, it is the hope of the authors that collaborative efforts between future team members of the project and clinical experts will allow further development of haptic feedback mechanisms for PLS simulators and other surgical teaching devices at a reasonable cost such that they can be used in both resource-rich and resource-limited settings for the benefit of surgeons and patients alike.

Over the course of the year, the PLS simulator project had seen various complications that affected the project's development in some way or another. Initially, the simulator's design was relatively optimistic in terms of the intended features and the timeline for implementation. As complications arose, the project design and timeline had to be adjusted accordingly, which explains some of the deviation between the actual implementation of the simulator and the intended PLS simulator and project timeline.

As mentioned in section 4.1 of this report, it was decided that a simulator box that is commercially sold would be preferable to building one from scratch as this would save more time for the development of the haptic feedback delivery system and the more functional parts of the simulator. Unfortunately, the project experienced various shipping delays mainly due to COVID-19, which caused various revisions to be made to the project timeline and functionality.

These revisions include lowering the number of tasks supported by the simulator to one. Additionally, the initial plan for the project thresholding had to be changed as it was no longer feasible to quantitatively map an expert's ideal procedure to create a more dynamic thresholding mechanic. Though these were mainly due to a lack of support from the project's medical partners, it was unlikely that they would have been completed by the end of the academic year due to the time constraints imposed by the initial delays. The design of the haptic wearable was also changed to only use 8 motors per arm rather than 10 as the project could not be delayed further to service the limitations of the PWM driver board. Finally, the project initially planned to create thresholds that were based on components like acceleration, speed, angular velocity, and smoothness. All of these features were initially outlined in the project proposal provided in Appendix A, but the aforementioned constraints made these features unfeasible for completion within the academic year. Most of these missing features have been mentioned in Chapter 5 as recommendations for future work since their implementation is still feasible in a more unconstrained environment.

Despite these complications, most of the original project metrics for success have been met. The project has succeeded in designing and implementing a functional PLS simulator which tracks movement in 2 laparoscopes and is capable of providing intuitive haptic feedback when warranted. Due to the lack of feedback from the project's clinical partners, the efficacy of the simulator as a solution to the problem of training surgeons is yet to be assessed. As a result, the simulator will likely require modifications over the coming years, but the programmability provided by the architecture of the software should make making modifications to the project a smooth process.

While the project can be called a success, the amount of unexpected complications that came at a detriment to the final result cannot be overlooked. Several factors, both political and logistical, ended up directly and indirectly affecting the time allotted for actual project development. Had the group members been working with a higher course load, the project would likely not have been feasible as the original project requirements did not take the external factors affecting the project timeline into account. Moreover, the added uncertainty provided by the ongoing pandemic further aggravated this as access to the campus was restricted and inconsistent with no definitive end in sight. In addition, the group was originally intended to have more members contributing towards the project. Having a small team made collaboration easier, but not enough to cover the volume of work required which came at an overall detriment to the project. In retrospect, if the project were to be restarted from scratch, the scope of the project would likely have to be decreased, or a greater effort in regards to the recruitment of extra contributors would have been made so that the requirements can be satisfied.

References

[1] A. Buia, F. Stockhausen, and E. Hanisch, *Laparoscopic surgery: A qualified systematic review*, 26-Dec-2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4686422/. [Accessed: 22-Mar-2022].

[2] M. Prasad, “Open surgery vs laparoscopic surgery: Which is the best procedure?,” Far North Surgery, 27-Feb-2022. [Online]. Available: https://www.farnorthsurgery.com/blog/open-surgery-vs-laparoscopic-surgery [Accessed: 12-Apr-2022].

[3] J.-U. Stolzenburg, M. C. Truss, R. Rabenalt, M. Do, T. Schwalenberg, P. F. Katsakiori, A. McNeill, and E. Liatsikos, “Training in Laparoscopy,” Training in Laparoscopy - Science Direct, 22-Dec-2006. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1871259206000852. [Accessed: 12-Apr-2022].

[4] P. K. H. Tam, “Laparoscopic surgery in children,” *Archives of Disease in Childhood*, 01-Mar-2000. [Online]. Available: https://adc.bmj.com/content/82/3/240. [Accessed: 22-Oct-2021].

[5] “Trainee using LAP mentor simulator, wearing VR goggles” Simbionix 2021. https://simbionix.com/simulators/lap-mentor/lap-mentor-vr-or/ [Accessed: 23-Mar-2022].

[6] “Laparoscopic Surgery Training Box Simulator Kit” SUAI-Medical Training. 2019. https://www.amazon.ae/Laparoscopic-Surgery-Training-Box-Simulator/dp/B07YB4N6GV [Accessed: 23-Mar-2022].

[7] R. V. Chirurgii, Existing laparoscopic simulators and their benefit for the surgeon, 01-Jan-2016. [Online]. Available: https://europepmc.org/article/med/26982186. [Accessed: 12-Apr-2022].

[8] H. Fuchs, M. A. Livingston, R. Raskar, D. Colucci, K. Keller, A. State, J. R. Crawford, P. Rademacher, S. H. Drake, and A. A. Meyer, "Augmented reality visualization for laparoscopic surgery," Augmented reality visualization for laparoscopic surgery | SpringerLink, 01-Jan-1998. [Online]. Available: https://link.springer.com/chapter/10.1007/BFb0056282. [Accessed: 12-Apr-2022].

[9] R. R. Postema, L. A. van Gastel, S. F. Hardon, H. J. Bonjer, and T. Horeman, "Haptic exploration improves performance of a laparoscopic training task," Surgical endoscopy, Aug-2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8263408/. [Accessed: 12-Apr-2022].

[10] E. M. Overtoom, T. Horeman, F.-W. Jansen, J. Dankelman, and H. W. R. Schreuder, "Haptic feedback, force feedback, and force-sensing in simulation training for Laparoscopy: A systematic overview," *Journal of Surgical Education*, 04-Aug-2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1931720418300394. [Accessed: 21-Oct-2021].

[11] "Public guideline on the code of ethics," Engineers Canada. [Online]. Available:https://engineerscanada.ca/publications/public-guideline-on-the-code-of-ethics [Accessed: 12-Apr-2022].

[12] The Professional Engineers Act, Professional Engineers Ontario. [Online] R.R.O. 1990, Reg. 941 Available: https://www.ontario.ca/laws/statute/90p28 [Accessed: 12-Apr-2022].

[13] S. Tee, “How to be an effective project manager?: BlackStorm Consulting,” *BlackStorm Consulting | We Scale Businesses*, 02-Feb-2022. [Online]. Available: https://blackstormco.asia/how-to-be-an-effective-project-manager/. [Accessed: 12-Apr-2022].

[14] C. Rossa, J. Fong, N. Usmani, R. Sloboda and M. Tavakoli, "Multiactuator Haptic Feedback on the Wrist for Needle Steering Guidance in Brachytherapy," in IEEE Robotics and Automation Letters, vol. 1, no. 2, pp. 852-859, July 2016, doi: 10.1109/LRA.2016.2528295.

[15] L. H. Olivas-Alanis, R. A. Calzada-Briseño, V. Segura-Ibarra, E. V. Vázquez, J. A. Diaz-Elizondo, E. Flores-Villalba, and C. A. Rodriguez, “LAPKaans: Tool-motion tracking and gripping force-sensing Modular Smart Laparoscopic Training System,” *Sensors*, vol. 20, no. 23, p. 6937, 2020.

[16] A. A. Moorhead, D. Nair, C. Morison, N. J. Cook, S. W. Beasley, and J. M. Wells, “Development of an instrumented thoracoscopic surgical trainer for objective evaluation of esophageal atresia/tracheoesophageal fistula repair,” *Medical & Biological Engineering & Computing*, vol. 58, no. 3, pp. 601–609, 2020.

[17] “Laparoscopic training box, simulator box, surgery simulator kit, for Training,” *LAPAROSCOPYBOXX - PRO*. [Online]. Available: https://laparoscopyboxx.com/products/laparoscopyboxx-pro. [Accessed: 22-Oct-2021].

[18] K. Ikuta, T. Kato, H. Ooe and Kazuhiko Shinohara, "Surgery Recorder System acquiring position/force information of surgical forceps," 2008 World Automation Congress, 2008, pp. 1-6.

[19] M. Sarwar, “Locally developed simulator promises to train surgery students,” *DAWN.COM*, 18-May-2017. [Online]. Available: https://www.dawn.com/news/1333907. [Accessed: 12-Apr-2022].

[20] S. M. Botden, F. Torab, S. N. Buzink, and J. J. Jakimowicz, “The importance of haptic feedback in laparoscopic suturing training and the additive value of virtual reality simulation,” Surgical Endoscopy, vol. 22, no. 5, pp. 1214–1222, 2007.

[21] J. BLOM, “LSM9DS0 Hookup Guide,” *LSM9DS0 Hookup Guide | learn.sparkfun.com*. [Online]. Available: https://learn.sparkfun.com/tutorials/lsm9ds0-hookup-guide. [Accessed: 12-Apr-2022].

[22] “Steval-MKI159V1,” *STMicroelectronics*. [Online]. Available: https://www.st.com/en/evaluation-tools/steval-mki159v1.html. [Accessed: 12-Apr-2022].

[23] S. Wizer, H. Yun, and R. Hunke, “Sparkfun/lsm9ds1_breakout: Breakout Board for ST's LSM9DS1 accelerometer/magnetometer/gyroscope system-in-package.,” *GitHub - Sparkfun/lsm9ds1_breakout*. [Online]. Available: https://github.com/sparkfun/LSM9DS1_Breakout. [Accessed: 12-Apr-2022].

[24] I. Beavers, “The case of the misguided gyro,” *The Case of the Misguided Gyro | Analog Devices*, Mar-2017. [Online]. Available: https://www.analog.com/en/analog-dialogue/raqs/raq-issue-139.html. [Accessed: 22-Oct-2021].

[25] PIXART, “PMW3389DM description and datasheet,” *ALLDATASHEET | PIXART/PMW3389DM*. [Online]. Available: https://pdf1.alldatasheet.com/datasheet-pdf/view/1179020/PIXART/PMW3389DM.html. [Accessed: 12-Apr-2022].

[26] Jack Enterprises, “PMW3389 motion sensor by Jack Enterprises on tindie,” *Tindie | pmw3389-motion-sensor*. [Online]. Available: https://www.tindie.com/products/jkicklighter/pmw3389-motion-sensor/. [Accessed: 12-Apr-2022].

[27] Branch Education, “How does a mouse know when you move it? - youtube.com.” [Online]. Available: https://www.youtube.com/watch?v=SAaESb4wTCM. [Accessed: 12-Apr-2022].

[28] Basics Explained, H3Vtux, “How Does a Mouse Work? - youtube.com.” [Online]. Available: https://www.youtube.com/watch?v=eccSwn9QVxo. [Accessed: 12-Apr-2022].

[29] M. Marsland, “3D model of custom trocar design”, OpenCAD, 2021. [Online]. Available: https://cad.onshape.com/documents/4f77fd0d4e1591bcd8c49638/w/52dfe7e862b263b65d4385a5/e/d830297ff98ed1836737242b

[30] “Novint's* Falcon Haptic device,” HapticsHouse.com. [Online]. Available: https://hapticshouse.com/pages/novints-falcon-haptic-device. [Accessed: 22-Oct-2021].

[31] R. Ackerley, I. Carlsson, H. Wester, H. Olausson, and H. Backlund Wasling, “Touch perceptions across skin sites: Differences between sensitivity, direction discrimination and pleasantness,” *Frontiers in Behavioral Neuroscience*, vol. 8, 2014.

[32] S. Paneels, M. Anastassova, S. Strachan, S. Van, S. Sivacoumarane, and C. Bolzmacher, “What’s around me? multi-actuator haptic feedback on the wrist,” in World Haptics Conference (WHC), 2013, April 2013, pp. 407–412.

[33] R. W. Cholewiak and A. A. Collins, “Vibrotactile localization on the arm: Effects of place, space, and age,” Perception & psychophysics, vol. 65, no. 7, pp. 1058–1077, 2003.

[34] Adafruit, “Adafruit 1201 vibrating mini motor disc,” *Rapid Online | adafruit-1201*. [Online]. Available: https://www.rapidonline.com/adafruit-1201-vibrating-mini-motor-disc-73-5414. [Accessed: 12-Apr-2022].

[35] B. Schweber, “Haptics components, pt 1: LRA, ERM, and piezo actuators,” *Power Electronic Tips*, 02-Jun-2021. [Online]. Available: https://www.powerelectronictips.com/haptics-components-pt-1-lra-erm-and-piezo-actuators/. [Accessed: 22-Oct-2021].

[36] S. Monk, “Arduino lesson 13. DC Motors,” *Adafruit Learning System*. [Online]. Available: https://learn.adafruit.com/adafruit-arduino-lesson-13-dc-motors/breadboard-layout. [Accessed: 12-Apr-2022].

[37] Adafruit, “16 channel 12-bit PWM Servo Motor driver,” *Aus Electronics Direct*. [Online]. Available: https://www.auselectronicsdirect.com.au/16-channel-12-bit-pwm-servo-motor-driver-module-pc. [Accessed: 12-Apr-2022].

[38] Adafruit, “PCA9685 Datasheet by Adafruit Industries LLC,” *Digi-Key*. [Online]. Available: https://www.digikey.com/htmldatasheets/production/1640697/0/0/1/pca9685-datasheet.html#:~:text=PCA9685%20Datasheet%20by%20Adafruit%20Industries%20L

LC%20f%201.,for%20Red%2FGreen%2FBlue%2FAmber%20%28RGBA%29%20color%20ba%20cklight%20ing%20applications. [Accessed: 12-Apr-2022].

[39] PySerial's Official Documentation , Pyserial. [Online] Available: https://pyserial.readthedocs.io/en/latest/ [Accessed: 12-Apr-2022].

[40] "Flex Sensor," *Digi-Key*. [Online]. Available: https://www.digikey.ca/en/product-highlight/s/spectra-symbol/flex-sensor. [Accessed: 12-Apr-2022].

[41] M. S. Raghu Prasad, M. Manivannan, and S. M. Chandramohan, "Effects of laparoscopic instrument and finger on force perception: A first step towards Laparoscopic Force-skills training," *Surgical Endoscopy*, vol. 29, no. 7, pp. 1927–1943, 2014.

[42] B. Gavrilovic, A. S. Fahy, B. Carrillo, A. Nasr, J. T. Gerstle, and G. Azzie, "Development of an open-source laparoscopic simulator capable of motion and force assessment: High tech at low cost," *Journal of Laparoendoscopic & Advanced Surgical Techniques*, vol. 28, no. 10, pp. 1253–1260, 2018.

[43] 3Dconnexion, "SpaceMouse wireless | 3Dconnexion," *3DMouse.ca*. [Online]. Available: https://www.3dmouse.ca/collections/popular-products/products/spacemouse-wireless-and-case. [Accessed: 12-Apr-2022].

Appendix A - Proposal

Haptic Feedback Training for Pediatric Laparoscopic Surgery Simulators

Supervisors: Carlos Rossa and James R. Green

Mahtab Mohammed - 101052701

Reyad El Mahdy - 101064879

1. Introduction

The purpose of this proposal is to seek approval to design and develop a Pediatric Laparoscopic Surgery Simulator enabled with haptic feedback as a 4th year undergraduate capstone project. This proposal will discuss some of the relevant aspects of laparoscopic surgery and explain the project's objectives in improving the currently mainstream training process for surgical trainees new to laparoscopy. This will be done by providing a brief background on the procedure, an outline of the project objectives, and the planned methodology in achieving those objectives. A project timeline and a list of required components and facilities will be provided at the end of the proposal.

2. Background

Laparoscopy is a surgical technique which involves inserting a laparoscope fitted with a camera through a small incision on the patient's body [1]. The surgical procedure is then performed through the small incision using other laparoscopes fitted with surgical tools [1]. This technique is used to minimize the invasiveness of surgical procedures while also benefiting from a significantly reduced patient recovery time [1]. Laparoscopy differs greatly in terms of technique from traditional surgery, special training is required for surgeons performing laparoscopy. This is especially true in the case of pediatric surgery, as the laparoscopic instruments used are smaller in diameter and the pediatric procedure is more delicate than in normal laparoscopy, leading to a greater need for precision [2]. Since the risks that accompany ad-hoc training for laparoscopic surgery are high, the current convention is to use simulated environments to train novice surgeons on laparoscopic procedures [3].

Originally, laparoscopic surgical simulators were not enabled by any form of haptic or visual feedback. Instead, feedback would be given manually by an experienced surgeon, but studies have shown that the addition of haptic or visual feedback positively affects trainee performance [4]. While visual feedback has proven to be beneficial, too many visual stimuli may detrimentally affect the training process by serving as distractions. In these cases, the implementation of haptic feedback into the simulator may prove beneficial by acting as an intuitive form of feedback that is less distracting to the user. A study by Overtoom et al found that the implementation of haptic feedback into laparoscopic simulators resulted in a significant improvement in speed, performance, and skill transfer and was deemed necessary to convey surgical elements such as tissue consistency [4]. Consequently, this project aims to explore the use of haptics in laparoscopy simulators for novice surgeons.

3. Objectives

The purpose of this project is to provide proof of concept for an emerging biomedical engineering technology, while also demonstrating our level of technical proficiency as 4th year engineering students. The project aims to implement the use of wearable technology and haptics to provide intuitive haptic feedback for surgical trainees so that they can accelerate their learning in a simulated environment with minimal risk. At completion, the simulator should be able to read accurate measurements for various surgical metrics based on the movement and positioning of the laparoscopic instruments and use signal processing methods to deliver real-time intuitive haptic feedback through a pair of wearable devices. These metrics include velocity, acceleration, position, smoothness, and time taken for task completion. The simulator is expected to be able measure a trainee's deviation from the metrics of an ideal procedure and deliver haptic feedback accordingly with the purpose of alerting the user of these deviations so that they can be corrected. The simulator in this project is not intended for clinical use, it instead aims to serve as proof-of-concept future experiments and studies in the use of haptics in laparoscopic surgical training simulators, and possibly laparoscopic tools in real surgical environments.

4. Planned Methodology

4.1 Overview of the design and feedback process:

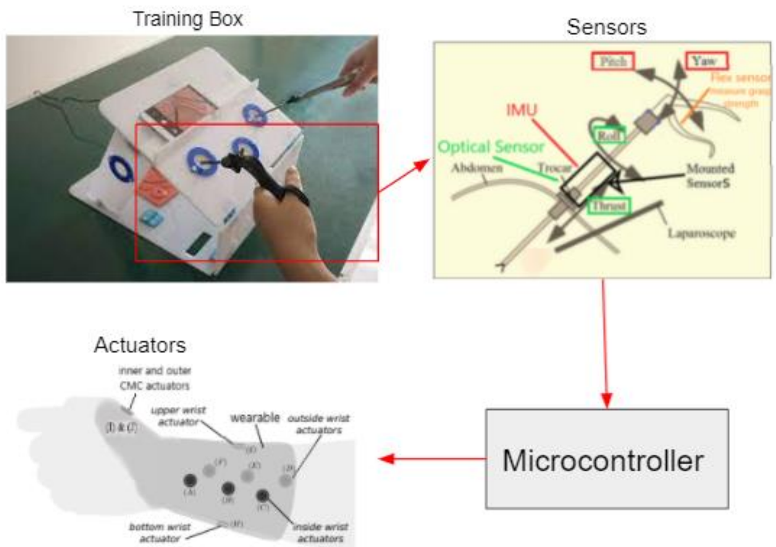

Fig. 1 Proposed concept of project[5]

To begin, this subsection offers a brief overview of how the main elements of the proposed project will be designed and implemented as shown in Figure 1. First, a box-type Pediatric Laparoscopic Surgery (PLS) simulator will be built and instrumented with sensors connected to a microcontroller. These sensors will provide real-time information on the movements of the surgical instruments involved in said PLS simulator. In addition, a wearable, such as a wrist brace, will be instrumented with a grid of vibrational actuators connected to the same microcontroller as above. The actuators on said wearable will provide tactile haptic feedback in specific patterns based on specific inputs from the microcontroller. Subjects will then be asked to perform specific surgical training tasks using the PLS simulator while equipped with the haptic feedback wearable.

[5] Figure 1 has been adapted from [5], [6], and [7]

Information acquired by the PLS simulator sensors during said tasks will be processed and inspected in real-time to ensure that the movement of the instruments involved are within set bounds. These bounds will be set based on data from an expert surgeon. If the movement of the instruments go out of bounds, actuators on the haptic feedback wearable will vibrate in a specific pattern so as to intuitively guide the surgical trainee back within the above-mentioned bounds. Finally, metrics such as time taken for completion of task and the smoothness with which it was completed will be assessed and compared to data from the surgical expert. Particular make and model of the components discussed in the following subsections are listed in section 6 of this proposal.

4.2 Pediatric Laparoscopic Surgery (PLS) Simulator Design

Types of laparoscopy surgery trainers that are currently available include Virtual Reality (VR) trainers, Augmented Reality (AR) trainers, and Box Trainers (BT). According to a study, a BT was generally preferred over a VR trainer by all participating surgeons [8]. This may be the sentiment in general due to lack of natural haptic feedback in VR trainers and simulated haptic feedback in VR trainers are not perfect as they have been proven to be disadvantageous due to additional friction in the haptic interface [4, p254]. Furthermore, most of the studies comparing BTs to VR trainers favor the former [4, p254]. Although AR trainers are promising, as they include the natural haptic feedback of BTs and virtual surroundings of VR trainers, only very limited studies have been conducted investigating their performance [4, p253]. As such, we are proposing the use of a box-type pediatric laparoscopic surgery (PLS) trainer for this project. This will be achieved by attaching a USB camera to a clear plastic box that will have holes drilled into it to fit 3 mm pediatric laparoscopic surgical instruments via custom-made trocars.

Each custom trocar will be instrumented with a 9 Degrees of Freedom (DOF) Inertial Measurement Unit (IMU) which will measure the pitch and yaw of the surgical instrument. We are proposing the use of 9 DOF IMUs because they each include an accelerometer, a gyroscope, and a magnetometer. In particular, the measurements of the magnetometer combined with that of the accelerometer allows accounting for “tilt” of the IMU, where the sensor can display the North direction regardless of its orientation [9]. This will prove crucial in setting a proper frame of reference for the surgical instrument measurements.

In addition, the custom trocars will be further instrumented with optical sensors to measure roll and thrust of the surgical instrument. The 9DOF IMU would have sufficed if it could offer accurate positional measurements long-term. However, this is not the case due to the accumulation of “drift” error over time which is caused by inherent imperfections and noise within the device [10]. With time, this will consequently cause calculation errors in the measurements of the IMU at each sampling instant, which will further lead to inaccurate measurements. As such, an optical sensor can offer updated position and orientation information to correct such inaccuracies over time. Therefore, we are proposing the use of an optical sensor in conjunction with the IMU, as shown in figure 2.

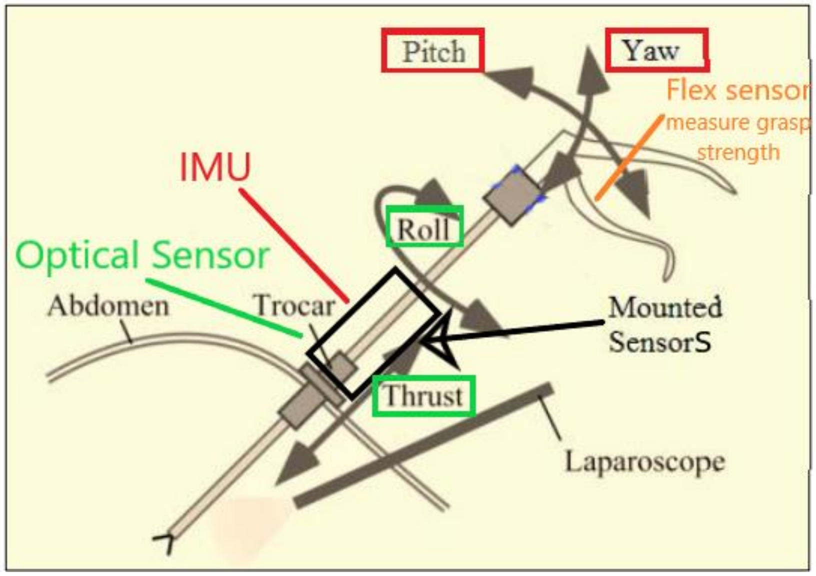

Fig. 2 Proposed concept for sensors in surgical instrument of PLS simulator[6]

Moreover, the IMUs and optical sensors will be connected to a microcontroller for data collection and data processing purposes. The use of a microcontroller with a large number of I/O and PWM pins is proposed as multiple sensors and actuators will be used in this project. Additional sensors depicted in figure 2 may be added to the PLS simulator and are further discussed in subsection 4.7. Finally, boards will be developed to allow performance of specific surgical training tasks, which will be placed at the base of the PLS simulator. These surgical tasks are further discussed in the next subsection.

[6] Figure 2 was adapted from [6]

4.3 Surgical Training Tasks in the PLS

A comprehensive web-based education module, named Fundamentals of Laparoscopic Surgery (FLS) was partially used as the basis for the proposed surgical training tasks to be performed in the PLS simulator [11]. Specific laparoscopic surgical tools are required for these tasks, which include a pediatric laparoscopic Maryland dissector, a pediatric laparoscopic needle holder, and pediatric laparoscopic scissors. Henceforth, the prefix defining these tools for pediatric laparoscopic surgery will not be used, but it should be understood that they refer to the 3mm pediatric laparoscopic tools. The description of each proposed surgical training tasks is as follows:

Task 1: Object Transfer - A needle holder is held in the dominant hand and a Maryland dissector is held in the non-dominant hand to transfer ring-shaped objects from one peg placed on the board at the base of the PLS simulator to another peg placed at a different location on the board. This task trains hand–eye coordination, spatial perception, and grasping skills [12].

Task 2: Needle Trailing - A needle holder is held in the dominant hand to trail a surgical needle in a specific order through loops placed at various locations on the board at the base of the PLS simulator. In addition, a Maryland dissector is held in the non-dominant hand and used for support to help fix orientation of the needle. This task trains all the same skills as task 1 in addition to needle manipulation skills.

Task 3: Precision Cutting - The scissors are held in the dominant hand to cut a circle with a certain diameter very close to the one drawn on a latex glove stretched out in a plastic base. A Maryland dissector is used to help hold the structure in place using the non-dominant hand, and the surgical trainee is expected to start and end at the same point. This bimanual task trains cutting, grasping, bimanual dexterity, and hand–eye coordination skills [12].

Task 4: Suturing - A needle holder is held in the dominant hand and a Maryland dissector is held in the non-dominant hand to suture a slit in a foam pad placed firmly at the base of the PLS that mimics organic tissue This task trains needle manipulation, management of silk suture, knot tying, bimanual dexterity, and hand–eye coordination skills [12].

Common performance metrics, such as velocity, acceleration, smoothness, and time taken for completion of task, will be measured across all tasks, and compared against data from an expert. Smoothness of task completion is an important metric during surgery and is defined by the rate of change in acceleration of the instrument [13]. However, the use of frequency analysis to investigate this metric is also proposed for this project. Moreover, haptic feedback will also be offered when instruments stray by +/- some standard deviation from expected path during training to intuitively guide the trainee back on track. This feedback is further discussed in subsection 4.5.

4.4 Haptic Feedback Wearable Design

Two types of haptic feedback options were considered for this project, forced feedback and tactile feedback. The former would involve the use of a high-fidelity robotic 3D haptic controller that updates its motors 1000 times per second to simulate natural 3D haptic feedback offered during real life surgery [14]. Using this device to build a box-type laparoscopic simulator would require the use of complex kinematics. The use of such an interface further concerned us, since the use of similar interfaces in VR trainers was complained about for adding friction during surgical training [4, p254]. In addition, building a laparoscopic trainer with this device would require two of them, one for each scope, which significantly drives up the cost and risk involved as each of them are expensive. Therefore, given the time and budget constraints of this project, the forced feedback option was ignored. On the other hand, use of tactile feedback using a wearable fitted with a grid of actuators on each hand may be less intuitive than forced feedback as surgical trainees may have to learn the vibrational patterns to assess their mistakes. However, the cost and risk involved is significantly reduced and it is slightly easier to implement due to the use of simpler signals. If proven significantly useful, this option would offer scalability in the long run. As such, the use of tactile feedback is proposed for this project.

Moreover, the wearable proposed for the project is one that covers the wrist and the base of the thumb, not unlike a wrist brace except for the metal splint, which will be removed for the project. This is proposed because placing actuators on the hands via gloves would allow for easier perception of different types and patterns of vibrations, as suggested in [15], but bulky gloves may also prove to be restrictive during the surgical training tasks. Also, the vibrations felt in the hand may carry over to the scopes which may interfere with the sensing of the IMU and optical sensors.

Therefore, via a wrist brace type wearable the actuators will be placed as far from each other as possible while keeping them as close to the wrists and thumb basal joints as possible. Such a configuration, depicted in Figure 3, is proposed because it keeps the fingers and hands free from any obstruction to perform the surgery. This configuration also allows for the vibrations to be felt more as sensation is greater closer to the joints and when more area of the skin is covered [16], [17].

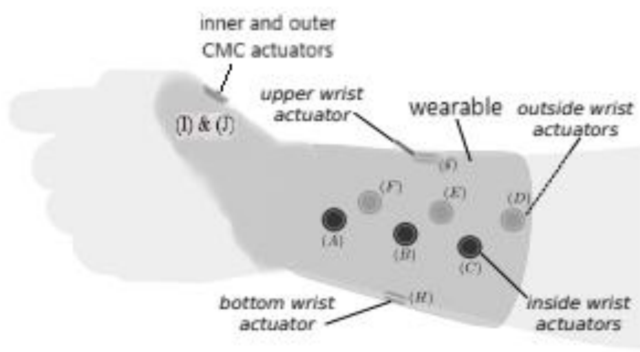

Fig. 3 Proposed concept for actuators in haptic feedback wearable[7]

The type of actuators proposed for this project is the Linear Resonant Actuator (LRA) type vibration motors because of their small and flat structure, which will easily fit in the wearable without making them bulky or heavy. In contrast, Eccentric Rotating Mass (ERM) vibration motors are heavier and bulky, making them a bad option for this specific application as configuring them in an array on the wearable would make it heavier. ERM motors also consume more power to be able to induce larger vibrations, which is also the case for piezoelectric actuators as they require more layers to induce perceptible vibrations [18]. Therefore, LRA vibration motors are proposed as opposed to piezoelectric actuators or ERM motors.

[7] Figure 3 was adapted from [7]

4.5 Wearable Feedback Delivery Method

Following is the proposed method of tactile feedback delivery via the wearable. Pulse Width Modulated (PWM) pins of the proposed microcontroller will have leads connected to the LRAs, which will allow the actuators to vibrate in specific patterns depending on the input. These vibrational patterns will be similar to those in [7], where specific actuators vibrate when specific translational or rotational motion is detected that leads the surgical instruments out of expected bounds. These vibrational patterns will be set so as to intuitively guide the surgical trainee back within the expected bounds.

For example, when a surgical instrument exceeds the bounds of thrusting translational motion, the actuators A, B, C and F, E, D in Figure 3 will repeatedly vibrate in that order to signal the trainee to translationally withdraw the instrument. Actuators I and J on the inside and outside of the carpometacarpal (CMC) joints are proposed to be used for tactile feedback when force applied during the task is too high. Proposed methods of force sensing are further discussed in subsection 4.7. In addition, actuators will be set to vibrate at or around resonance frequency so that their perception is maximized but does not prove to be hindering. This frequency will be subject to change with experimentation and as required for individual surgical trainees.

Above-mentioned expected bounds are proposed to be based on data collected from an expert surgeon experienced with PLS. The scenario where this might not be feasible is discussed in section 7. The PLS expert will be asked to perform the surgical training tasks using the PLS simulator, and information collected from them will then be used to set an average expected path for each training task.

Surgical trainees who train with the PLS simulator thereafter will be allowed some leeway from this average expected path, which will be decided by thresholds set at certain positive and negative standard deviations away from said average.

4.6 Metrics for Success of the Project

Ideally, a study to assess the intuitiveness of the tactile haptic feedback training for PLS simulator would be run with participants of varying surgical skill level. The results of that study then would serve as a metric for success of the project. However, due to the scope and the timing constraints of the project, we are proposing to set the metric for success of the project as the complete functionality of the sensor instrumented PLS simulator and consequent functionality of the haptic feedback wearable when instruments in said PLS simulator exceed set bounds.

To be more specific, success for the design of the instrumented PLS simulator will be defined by its ability to gather information from all the sensors by means of the microcontroller. Similarly, success for the design of the haptic feedback wearable is defined by vibration of individual actuators in specific patterns given specific inputs from the microcontroller. Once these elements work in conjunction as proposed, assessment of the final product from an expert and their feedback will set the basis for future work on the project and subsequent studies or experiments involving it.

4.7 Advanced Sensing in PLS Simulator (given additional funding)

External feedback on the magnitude of force used during a laparoscopic surgery is non-existent, which causes both novices and residents to have misperceptions of the force applied during laparoscopic surgery leading them to apply excessive force [19]. As such, training with force parameters is hypothesized to benefit laparoscopic surgeons of various experience levels [19].

In one study that used sensors in the handles of the graspers, it was found that force applied during laparoscopic pinching using a grasper is significantly affected by surgical experience [20]. Therefore, the use of flex sensors to measure gripping strength is proposed for this project. In addition, these sensors can help set markers for individual portions of the surgical training tasks. For example, object 1 was picked up or released at a certain time because the flex sensor measurements changed at that time.

Furthermore, additional force sensing measurements can be acquired during complex training tasks such as suturing by instrumenting the base of the PLS simulator or the suturing foam with types of force sensors. A modified 3D mouse is used as a base to assess force parameters in various surgical training simulations and is prevalent in literature [21], [22], [23]. Exploration of options that use basic principles of such a device to assess force parameters is proposed.

5. Project Relation to Degree

Since the project involves building a simulator for pediatric laparoscopy which is a device intended for clinical applications, there is a strong association between the project and the biomedical engineering field in general. More specifically, as Biomedical and Electrical Engineering students there are many relevant courses within the degrees that have applications within the project. This includes digital signal processing (SYSC 4405) which is used to sample and process input signals into forms that the microcontroller can read and act upon, as well as working with microcontrollers themselves which is covered in the microprocessor systems course (ELEC 4601). There are several other courses with varying degrees of contribution to the project, however, it is clear that this is a biomedical project that employs mainly electrical engineering solutions to achieve its function. As a result, the project falls under the biomedical and electrical engineering spectrum of our degrees.

6. Project Requirements

Required Material	Proposed Product/Description
Surgery Simulator	
IMU x2	**LSM9DS1 9DOF Development board IMU to measure pitch and yaw of the surgical instruments**
Optical Sensors x2	**PMW3360 Optical motion sensors would be used to track translational movement within the procedure**
Camera	**A Generic HD webcam would likely be used to capture the visual elements of the surgical tasks**
Microcontroller	**An ATMega 2560 Microcontroller would be used to process inputs from the sensors and control actuators based on thresholds. This particular product was chosen for its large number of digital and analog I/O pins**
Surgical Instruments	
Pediatric Laparoscopic Needle Holder	**3 mm Pediatric Laparoscopic Needle Holder training instrument for surgical training tasks 1, 2, and 4**
Pediatric Laparoscopic Maryland Dissector	**3 mm Pediatric Laparoscopic Dissector training instrument for surgical training tasks 1-4**
Pediatric Laparoscopic Scissors	**3 mm Pediatric Laparoscopic Scissors training instrument for surgical training task 3**
Haptic Feedback	
Actuators x20	**1201 Adafruit LRA vibration motors that vibrate to deliver haptic feedback in the wearable**
Wrist brace x2	**Wrist brace type wearable that facilitates delivery of haptic feedback from the LRA vibration motors**
Advanced Sensing in PLS Simulator	
Flex sensors x2	**2.2" Flex Sensor SEN-10264 used to measure gripping strength of the surgical instruments**
Force sensitive sensor	**Sensors similar to ones used in this 3D Mouse can be used in the base of the PLS simulator to measure force metrics during surgical training tasks**

Required Material	Proposed Product/Description
General Requirements	
Working space	A lab or working space will be needed to store, build, and test the simulator.
Power supply	We would need to be able to supply power to all the components of the simulator. USB power from a computer should be enough for the microcontroller and most of the components.
Wiring	Wiring and soldering equipment would be needed to connect all the components together.

7. Risks and Mitigation Strategies

In terms of safety or physical risks, there aren't any notable risks associated with the project as the tasks would be conducted in a simulated environment with no real patients. However, there are some factors associated with the project methodology that have dependencies that might not be able to be satisfied. For example, setting proper activation parameters for the actuators would ideally be done by an experienced surgeon who would know what cases are worthy of triggering haptic feedback. Due to issues with time constraints and availability, it might not be possible to get these parameters directly from an expert. In this case, the parameters will be set based on means and standard deviations gathered from existing literature on the procedure. Another potential risk to the project is the component or more specifically sensor failure. While IMUs can measure positional data using a gyroscope and an accelerometer, the positional data would eventually be skewed by drift error which is undesirable due to the required precision in the project. To mitigate this, we will attempt to use optical sensors to measure depth and use the 9DOF IMU's magnetometer to keep track of rotational movement. If the optical sensor method could not be integrated for any reason, the IMUs would have to be used to measure depth as well and measures, such as Kalman Filtering, would have to be taken to minimize and account for drift error.

Finally, the project timeline in section 8 does not account for unexpected delays in shipping. The effects of delayed shipping on the project would differ depending on the components. In the event of unexpected delays, we may have to opt for a more minimalist design and sacrifice some functionality in order to meet the project deadlines.

8. Proposed Timeline

Task/milestone	Expected start date	Proposed date of completion	Due Date	Member Responsible
Proposal	October 12, 2021	October 21, 2021	October 22, 2021	All
Budget form application	October 24, 2021	October 30, 2021	October 31, 2021	All
Order initial sensors of PLS simulator and microcontroller	October 31, 2021	October 31, 2021	N/A	All
Implement the IMU in microcontroller	November 16, 2021	November 28, 2021	N/A	Reyad
Testing/Validation of IMU	November 28, 2021	December 1, 2021	N/A	Mahtab
Integrate optical sensors into the system	November 30, 2021	December 3, 2021	N/A	Reyad
Testing/Validation of optical sensor integration	December 3, 2021	December 8, 2021	N/A	Mahtab
Progress Report	December 5, 2021	December 9, 2021	December 10, 2021	All
Oral Presentation Form	December 8, 2021	December 9, 2021	December 10, 2021	All
Fall Final Exam + Winter Break (December 14, 2021 to January 7, 2022)				
Order wearable and actuators	December 27, 2021	December 27, 2021	N/A	All

Task/milestone	**Expected start date**	**Proposed date of completion**	**Due Date**	**Member Responsible**
Integrate Actuators into the wearable via microcontroller	January 10, 2022	January 17, 2022	N/A	Mahtab
Verify Actuator functionality via microcontroller	January 17, 2022	January 22, 2022	N/A	Reyad
Oral Presentation	January 14, 2022	January 24, 2022	January 25, 2022	All
Build PLS simulator	January 24, 2022	January 31, 2022	N/A	All
Set actuator parameters	February 1, 2022	February 17, 2022	N/A	Reyad
Digital processing of input/sensor signals from the PLS simulator	February 1, 2022	February 17, 2022	N/A	Mahtab
Final verification and testing of PLS simulator with haptic feedback	February 18, 2022	March 17, 2022	N/A	All
Poster Fair	March 5, 2022	March 17, 2022	March 18, 2022	All
Advanced sensing in PLS simulator (flex sensors on the instrument, force sensing in the base)	March 21, 2022	March 30, 2022	N/A	All
Final Project Validation	March 30, 2022	April 5, 2022	N/A	All
Final Report	March 17, 2022	April 11, 2022	April 12, 2022	All

References

[1] A. Buia, F. Stockhausen, and E. Hanisch, *Laparoscopic surgery: A qualified systematic review*, 26-Dec-2015. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4686422/. [Accessed: 22-Oct-2021].

[2] M. Prasad, "Open surgery vs laparoscopic surgery: Which is the best procedure?," Far North Surgery, 27-Feb-2022. [Online]. Available: https://www.farnorthsurgery.com/blog/open-surgery-vs-laparoscopic-surgery [Accessed: 12-Apr-2022].

[3] P. K. H. Tam, "Laparoscopic surgery in children," *Archives of Disease in Childhood*, 01-Mar-2000. [Online]. Available: https://adc.bmj.com/content/82/3/240. [Accessed: 22-Oct-2021].

[4] M. Varras, N. Nikiteas, V. K. Varra, F. N. Varra, E. Georgiou, and C. Loukas, "Role of laparoscopic simulators in the development and assessment of Laparoscopic Surgical Skills in laparoscopic surgery and gynecology (review)," *World Academy of Sciences Journal*, 01-Mar-2020. [Online]. Available: https://www.spandidos-publications.com/10.3892/wasj.2020.41. [Accessed: 22-Oct-2021].

[5] E. M. Overtoom, T. Horeman, F.-W. Jansen, J. Dankelman, and H. W. R. Schreuder, "Haptic feedback, force feedback, and force-sensing in simulation training for Laparoscopy: A systematic overview," *Journal of Surgical Education*, 04-Aug-2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1931720418300394. [Accessed: 21-Oct-2021].

[6] “Laplay laparoscopic training box, simulator box, surgery simulator kit, for Training • $308.21,” *PicClick CA*. [Online]. Available: https://picclick.ca/Laplay-Laparoscopic-Training-BoxSimulator-Box-Surgery-Simulator-Kit-114936505557.html. [Accessed: 22-Oct-2021].

[7] K. Ikuta, T. Kato, H. Ooe and Kazuhiko Shinohara, "Surgery Recorder System acquiring position/force information of surgical forceps," 2008 World Automation Congress, 2008, pp. 1-6.

[8] C. Rossa, J. Fong, N. Usmani, R. Sloboda and M. Tavakoli, "Multiactuator Haptic Feedback on the Wrist for Needle Steering Guidance in Brachytherapy," in IEEE Robotics and Automation Letters, vol. 1, no. 2, pp. 852-859, July 2016, doi: 10.1109/LRA.2016.2528295.

[9] S. M. Botden, F. Torab, S. N. Buzink, and J. J. Jakimowicz, “The importance of haptic feedback in laparoscopic suturing training and the additive value of virtual reality simulation,” Surgical Endoscopy, vol. 22, no. 5, pp. 1214–1222, 2007.

[10] O. S. Media, “Basics of 6DOF and 9DOF sensor fusion,” *Embedded Computing Design*. [Online]. Available: https://www.embeddedcomputing.com/technology/analog-and-power/basics-of-6dof-and-9dof-sensor-fusion. [Accessed: 22-Oct-2021].

[11] I. Beavers, “The case of the misguided gyro,” *The Case of the Misguided Gyro | Analog Devices*, Mar-2017. [Online]. Available: https://www.analog.com/en/analog-dialogue/raqs/raq-issue-139.html. [Accessed: 22-Oct-2021].

[12] “FLS program description,” Fundamentals of Laparoscopic Surgery, 05-Jan-2021. [Online]. Available: https://www.flsprogram.org/index/fls-program-description/. [Accessed: 22-Oct-2021].

[13] I. Oropesa, F. P. Escamirosa, J. A. Sánchez-Margallo, S. Enciso, B. Rodríguez-Vila, A. M. Martínez, F. M. Sánchez-Margallo, E. J. Gómez, and P. Sánchez-González, "Interpretation of motion analysis of laparoscopic instruments based on principal component analysis in box trainer settings," *Surgical Endoscopy*, vol. 32, no. 7, pp. 3096–3107, 2018.

[14] A. Ghasemloonia, Y. Maddahi, K. Zareinia, S. Lama, J. C. Dort, and G. R. Sutherland, "Surgical skill assessment using motion quality and smoothness," *Journal of Surgical Education*, vol. 74, no. 2, pp. 295–305, 2017.

[15] "Novint's* Falcon Haptic device," HapticsHouse.com. [Online]. Available: https://hapticshouse.com/pages/novints-falcon-haptic-device. [Accessed: 22-Oct-2021].

[16] R. Ackerley, I. Carlsson, H. Wester, H. Olausson, and H. Backlund Wasling, "Touch perceptions across skin sites: Differences between sensitivity, direction discrimination and pleasantness," *Frontiers in Behavioral Neuroscience,* vol. 8, 2014.

[17] S. Paneels, M. Anastassova, S. Strachan, S. Van, S. Sivacoumarane, and C. Bolzmacher, "What's around me? multi-actuator haptic feedback on the wrist," in World Haptics Conference (WHC), 2013, April 2013, pp. 407–412.

[18] R. W. Cholewiak and A. A. Collins, "Vibrotactile localization on the arm: Effects of place, space, and age," Perception & psychophysics, vol. 65, no. 7, pp. 1058–1077, 2003.

[19] B. Schweber, "Haptics components, pt 1: LRA, ERM, and piezo actuators," *Power Electronic Tips,* 02-Jun-2021. [Online]. Available: https://www.powerelectronictips.com/haptics-components-pt-1-lra-erm-and-piezo-actuators/. [Accessed: 22-Oct-2021].

[20] M. S. Raghu Prasad, M. Manivannan, and S. M. Chandramohan, “Effects of laparoscopic instrument and finger on force perception: A first step towards Laparoscopic Force-skills training,” *Surgical Endoscopy*, vol. 29, no. 7, pp. 1927–1943, 2014.

[21] W. K. Susmitha, G. Mathew, S. R. Devasahayam, B. Perakath, and S. K. Velusamy, “Factors influencing forces during laparoscopic pinching: Towards the design of Virtual Simulator,” *International Journal of Surgery*, vol. 18, pp. 211–215, 2015.

[22] T. Horeman, S. P. Rodrigues, F.-W. Jansen, J. Dankelman, and J. J. van den Dobbelsteen, “Force measurement platform for training and assessment of Laparoscopic Skills,” *Surgical Endoscopy*, vol. 24, no. 12, pp. 3102–3108, 2010.

[23] B. Gavrilovic, A. S. Fahy, B. Carrillo, A. Nasr, J. T. Gerstle, and G. Azzie, “Development of an open-source laparoscopic simulator capable of motion and force assessment: High tech at low cost,” *Journal of Laparoendoscopic & Advanced Surgical Techniques*, vol. 28, no. 10, pp. 1253–1260, 2018.

[24] A. A. Moorhead, D. Nair, C. Morison, N. J. Cook, S. W. Beasley, and J. M. Wells, “Development of an instrumented thoracoscopic surgical trainer for objective evaluation of esophageal atresia/tracheoesophageal fistula repair,” *Medical & Biological Engineering & Computing*, vol. 58, no. 3, pp. 601–609, 2020.

Appendix B - GitHub Repository Description

Following is a description of the directories and files that were used during the project. These files are available at https://github.com/meyadelrahdy/pls-simulator.

1. Programs for main functioning of the PLS simulator and haptic feedback:

1.1*pls-simulator/PLS_Actuator.py* - Python script that reads in sensor data from the microcontroller via PySerial, processes it to detect threshold violations via FSM and sends a string to the microcontroller via PySerial signifying which actuation pattern to trigger.

Suggested enhancements:

- Leverage SciPy library to filter received sensor data as required
- Leverage AsyncIO library to incorporate asynchronous functions to send multiple strings to the microcontroller in a parallel fashion.

1.2 *pls-simulator/sketches-to-test-parts/Sensors_Actuator_Interface_test/* - The Arduino program with extension ".ino" in this directory should be loaded on to the microcontroller before running the script in 1.1. This program takes in data from the IMUs, and optical sensors connected to the microcontroller via SPI, converts the data to understandable units, and transmits it to the script in 1.1 via PySerial. This program also parses the string received from the script in 1.1 via PySerial and drives the actuators via the PWM driver.

Suggested enhancements:

- Find ways to add intervals between motors without using delay () function in Arduino.

2. Programs to test parts:

2.1 *pls-simulator/sketches-to-test-parts/Sensors_Integration_SPI_test/* - The Arduino program with extension ".ino" in this directory was loaded into the microcontroller to test the integration of all four sensors, i.e., two IMUs and two microcontrollers via SPI.

2.2 *pls-simulator/sketches-to-test-parts/LSM9DS1_I2C_TestConfig.ino* - This Arduino program was used to test reading in accelerometer, gyroscope, and magnetometer data from the IMUs via I2C protocol

2.3 *pls-simulator/sketches-to-test-parts/sketch_basic_LRA_test.ino* - This Arduino program was used to test the LRA motors individually.

2.4 *pls-simulator/sketches-to-test-parts/test_pwm_driver_and_actuators.ino* - This Arduino program was used to test driving the LRA motors via the PWM driver. The program turns on all 16 motors one after the other. To turn them off, set the PWM command from 4096 to 0 and upload it to the microcontroller.

2.5 *pls-simulator/sketches-to-test-parts/Python/ActuatorFSM.py* - This script was used to progressively design and test the actuator state controller.

2.6 *pls-simulator/sketches-to-test-parts/Python/optical-sensor_serial-test.py* - This script was used to test the PySerial interface by reading in the test data from the optical sensors.

2.7 *pls-simulator/sketches-to-test-parts/Python/sensors-py-arduino-io_interface.py* - This script was used to test the PySerial interface between Arduino and Python by reading in strings from the serial port and writing strings to the same.

2.8 *pls-simulator/sketches-to-test-parts/Python/serialtest.py* - This script was used to test reading in the IMU data.