Characterization and Investigation into the Accuracy of Touch-Based Registration Using Fiber Optic Shape Sensing

By

MeiLissa McKay

Thesis

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

MASTER OF SCIENCE

in

MECHANICAL ENGINEERING

May 10, 2024

Nashville, Tennessee

Approved:

Eric Barth, Ph.D.

Robert Webster, Ph.D.

Kevin Galloway, Ph.D.

Acknowledgements

I would first like to express my deepest thanks and appreciation to my advisor Dr.Eric Barth, I am sincerely thankful first for giving me the opportunity to learn and work with him as well as for his guidance and understanding throughout this project. I would also like to thank my labmates in the DCES Lab and MedLab both for their help during my project as well as for making that time more fun. And I would like to express my thanks to the members of my committee, Dr. Robert Webster and Dr. Kevin Galloway for their helping during this project and for serving on my committee.

Additionally, I want to thank my parents and sister for their love, support, and encouragement throughout my research at Vanderbilt and in life.

Finally, thank you to the NSF EFRI grant for funding this work.

Contents

List of Figures

Chapter 1

Overview

1.1 Introduction and Motivation

Currently, soft robots show much promise for use in surgical robotics. Because of their inherently soft structure, soft surgical robotics could lead to improvements in safety during surgery. And their use could be advantageous for minimally invasive or more patient specific procedures since they can follow unique and complex curved paths under actuation.

However, a successful surgical robot also needs to have a highly accurate sensing system for detecting its own continuous shape, 3D position, and environment throughout a surgery as well as for using that information for its controls and guidance. Unfortunately this is currently a weak spot in the field of soft surgical robotics.

Unlike with rigid robots, traditional robot kinematics cannot be used to determine a soft robot's shape and end effector position due to their non rigid structure. And many established technologies such as optical and magnetic tracking that can be used to effectively track soft robot position, also have drawbacks that limit the surgical applications for which they can be used. For example, optical tracking requires a direct line of sight to the object tracked which is often limited or not possible once the robotic actuator is inside the patient. On the other hand, magnetic tracking is both sensitive to any environmental factors that can interfere with its magnetic field such as certain surgical tools or medical imaging and can only provide discrete measurement points rather than continuous 3D shape data.

And so the overall goal of this work was to serve as a test bed project to show that soft robotic actuators can achieve highly accurate position control for surgical applications and investigate fiber optic shape sensing (FOSS) as a possible sensing method for the sensing and controls of image guided soft surgical robot.

For this, a form of knee cartilage repair the osteochondral autograft transfer system (OATS) was chosen as our target surgical application. The OATs procedure, depicted in Figure 1.1, is used to restore damaged articular cartilage in the knee [1]. During surgery, a cylindrical region containing the damaged articular hyaline cartilage and the subchondral bone underneath is removed. A cylindrical donor graft is then taken from a lower weight bearing region of the knee and fit into the original site of the defect. The donor and recipient sites are then left to heal on their own [1,2]. For the procedure to succeed, the graft must be slightly smaller than that of the recipient site yet close enough in size and shape to be press-fit into that site [1].

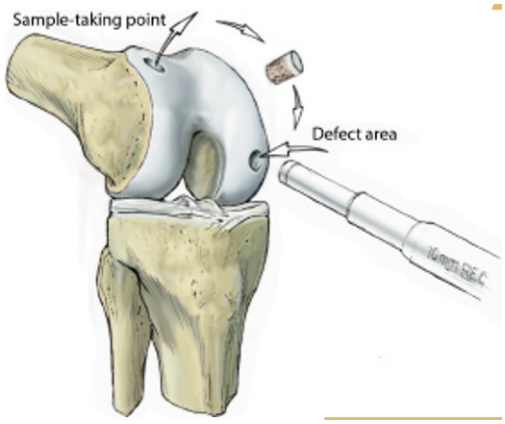

Figure 1.1: OATS procedure method

We imagine that a successful final goal for our project Would be the creation of soft robotic actuator that could first burrow to the locations of damage and of the donor graft and then use a laser osteotome located on the robot's end effector to remove the damage and create the donor regions. Here, Figure 1.2 shows the imagined structure and path of the soft robot we aimed to design to carry out this task.

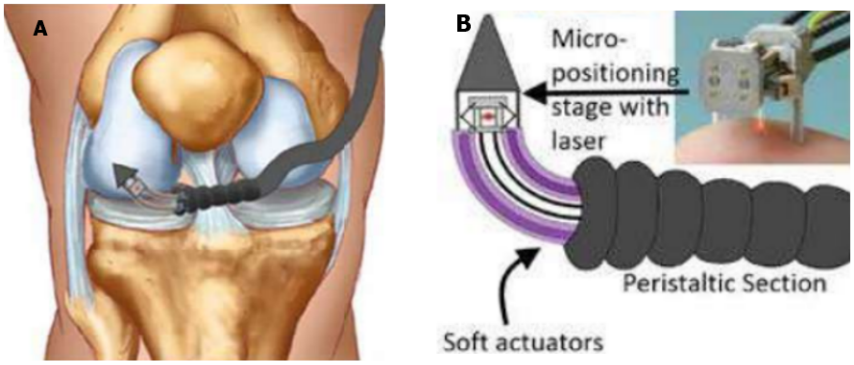

Figure 1.2: A. Envisioned goal for A. the soft robotic platform B. integration of laser osteotome into the robot

This procedure was chosen because the robot would have to trace out two highly accurate shapes with its tip to remove the damage and create the donor. And a robot that could successfully carry

out this task would demonstrate the soft actuator's ability for accurate position control as well as continuous actuator 3D shape and position sensing.

The work presented in this thesis focuses on testing a commercially available Fiber Optic Shape Sensing (FOSS) system to determine whether it is capable as serving as the main sensing for the shape detection and position control of a soft robotic surgical platform. FOSS systems provide 3D position measurements virtually continuously along the length of a flexible fiber optic cable. Therefore, we imagine that a FOSS system that is accurate and fast could be integrated into the body of a soft robotic actuator and provide the data needed for our project. This work first investigates the limitations of the sensing system by testing how it responds to common conditions from the environment or method of use such as light, temperature, twist and sensor configuration changes. It then investigates methods of using the FOSS for robotic image guidance that aim to mitigate the effects of those conditions. And lastly, it explores a potential method of correcting for measurement errors via sensor fusion between the FOSS and a magnetic tracking system

1.2 Literature Survey

The following literature review will discuss previous studies that investigated touch-based registration as an effective method for surgical image guidance as well as studies that investigated the use of fiber optic shape sensing (FOSS) for effectively detecting the shape and environment of soft robotic actuators and surgical needles.

Surgical image guidance assists surgeons in localizing their surgical tools relative to the patient anatomy being operated on continuously during a procedure. It thereby holds a lot of promise for increasing the safety, accuracy, and overall outcome of the surgery for the patient. Touch-based registration is a method of determining the transformation between the physical space on which the surgeon is operating and the space of a preoperative image of the surgical site necessary for image guidance. There have been multiple past studies investigating how the da Vinci surgical system can be used for touch-based registration for robotic assisted partial nephrectomy [3,4]. During these studies, the instrument tip was lightly traced over the surface of a phantom representing the patient anatomy. Tip positions during tracing were then determined by recording the robot's joint values during the tracing and reconstructing a the tip position via the robot kinematics of the da Vinci. The findings of these studies showed that this method was able to determine a proper registration between the surgical space and the preoperative image space. And ultimately, in trials with a surgeon, using this method of image guidance was shown to increase the accuracy of localizing specific anatomy.

The work in this thesis employs a similar method of image guided surgery in that it uses the tracing

of an anatomical surface with the tip of a tracked instrument to construct the surgical space point for registration. However, this thesis aims to build upon this past work and determine a method of touch based registration that would be compatible with soft robotic actuators. Unlike when working with rigid robotic systems like the da Vinci, traditional robot kinematics and joint angle values cannot be used to reconstruct a soft robotic actuators tip position during a tracing experiment. Instead, when working with a soft robotic surgical system, a separate sensing system must be integrated into the soft actuator to measure the tip position. Therefore, this work investigates whether a soft flexible fiber optic shape sensing (FOSS) system can be used to accomplish a registration task similar to those in previous studies.

Previous studies have integrated similar FOSS systems that work from the principle of optical frequency domain reflectometry (OFDR) into different types of actuators to test how well the FOSS could track the shape or environment of the actuator into which the sensor was integrated. For example, in existing work, a FOSS system from Luna Innovations Inc. (Blacksburg, VA) was integrated into a fiber reinforced C-bend soft actuator [5]. And this work then found that the FOSS and soft actuator system was able to effectively measure parameters such as the actuator's tip position while moving along a linear path, the shape of a surface while the actuator tip was moved over it, and the position of an obstacle in the system's environment. Additionally, certain groups have developed their own FOSS systems and tested them for surgical applications. For example, one group has developed a system fabricated from a triplet of UV exposed fibers and with the same OFDR principle. Their system has been tested in early studies with surgical needles and later studies in catheters for intra-arterial image guidance [7,8]. When integrated into surgical needles, their system was determined to accurately reconstruct shape with about 1mm RMS error [7]. And when integrated into a catheter for image guidance, and system was determined to have low tip position errors of around 1.2 mm during in vitro studies and low target registration errors of around 2.1 mm during in vivo studies[8].

Overall, these previous studies show a lot of potential towards the future adaption of fiber optic shape sensing in both soft robotic systems as well as for surgical image guidance. However, because the field of fiber optic shape sensing is a lot less established than typical object technologies such as optical or magnetic tracking these systems are also less available. And many systems are either developed in academic labs or directly for a company's specific robotic platform such as Intuitive's Ion platform which was given FDA clearance in 2019. Therefore, given the success of the previous studies described here, this work aimed to explore whether a commercially available FOSS system could also be used for touch-based registration successfully.

1.3 Organization of the Document

This thesis is divided into three main chapters. Chapter I presents an introduction and motivation of the work shown as well as a literature survey of related work. Chapter II discusses an outline of the author's work with characterizing the fiber optic shape sensing system which identified the causes of common errors and explored how to correct these issues. Next, Chapter III is the manuscript covering the author's final work at Vanderbilt. And lastly, Chapter IV discusses potential future directions to build off of the author's work presented in this thesis.

Manuscript I: Characterization and Investigation of the Accuracy of Touch-Based Registration Using Fiber Optic Shape Sensing

This paper presents a characterization study of a commercially available fiber optic shape sensing system from The Shape Sensing Company (Austin, TX, USA) as well as an evaluation of how well the system can be used for touch-based registration for image guided surgery. It first investigates how conditions common during the touch-based registration procedures such as sensor twist and configuration changes along the proximal end of the fiber influence the FOSS's measurements as well as a potential method of actively undoing the position error caused by these conditions. The paper then presents the results of FOSS based touch-based registrations procedures with and without reducing sensor twist and orientation changes.

1.4 References

[1] R. F. LaPrade, J. C. Botker, "Donor-site morbidity after osteochondral autograft transfer procedures", Arthroscopy: The Journal of Arthroscopic Related Surgery, vol. 20, issue 7, pp. e69-e73, Sept. 2004

[2] A. Pareek, P. J.Reardon, T. G. Maak, B. A. Levy, M. J. Stuart, A. J. Krych, "Long-term Outcomes After Osteochondral Autograft Transfer: A Systematic Review at Mean Follow-up of 10.2 Years". Arthroscopy: The Journal of Arthroscopic Related Surgery, vol. 2, issue 6, pp. 1174-1184, June 2016

[3] J. M. Ferguson, E. B. Pitt, A. A. Remirez, M. A. Siebold, A. Kuntz, N.L.Kavoussi, E.J.Barth, S.D.Herrell, and R. J. Webster, "Toward practical and accurate touch-based image guidance for robotic partial nephrectomy,"IEEE Transactions on Medical Robotics and Bionics, vol. 2, no. 2, pp. 196–205, may 2020.

[4] N. L. Kavoussi, B. Pitt, J. M. Ferguson, J. Granna, A. Remirez, N. Nimmagadda, R. Melnyk, A. Ghazi, E. J. Barth, R. J. Webster, S. D. Herrell, "Accuracy of Touch-Based Registration During Robotic Image-Guided Partial Nephrectomy Before and After Tumor Resection in Validated

Phantoms", Journal of Endourology, vol. 35, no. 3, pp. 362–368 , march 2021

[5] Kevin C. Galloway, Yue Chen, Emily Templeton, Brian Rife, Isuru S. Godage, and Eric J. Barth, "Fiber Optic Shape Sensing for Soft Robotics," Soft Robotics. Oct 2019

[6] Francois Parent, Sebastien Loranger, Koushik Kanti Mandal, Victor Lambin Iezzi, Jerome Lapointe, Jean-Sébastien Boisvert, Mohamed Diaa Baiad, Samuel Kadoury, and Raman Kashyap, "Enhancement of accuracy in shape sensing of surgical needles using optical frequency domain reflectometry in optical fibers," Biomed. Opt. Express 8, 2210-2221 (2017)

[7] F. Parent et al., "Intra-Arterial Image Guidance With Optical Frequency Domain Reflectometry Shape Sensing," in IEEE Transactions on Medical Imaging, vol. 38, no. 2, pp. 482-492, Feb. 2019, doi: 10.1109/TMI.2018.2866494.

Chapter 2

Outline of FOSS Characterization and Registration Studies

Over the course of the work presented in this thesis, we discovered a number of challenges with the FOSS system used in this work. These challenges were unexpected given the promising system specifications reported by The Shape Sensing Company when the sensor was purchased. However, some of the specifications reported by TSSC were experimentally determined to be inaccurate. For example, the acquisition rate of the system was reported to be 60 Hz, but was really 16 HZ and acknowledged by TSSC to be 16 Hz in actuality. And the FOSS system was determined to be incredibly sensitive to certain conditions that were not reported by TSSC as major sources of error.

These challenges significantly influenced the direction this work took. And much of the work leading up to and in the presented manuscript was focused on identifying the sources of unexpected errors found during our initial touch-based registration experiments. Ultimately, the work in this thesis was able to lower these errors, but did not find a solution to meet our goal of a submillimetric registration error. Therefore, this section serves to document what methods have already been investigated and their outcomes. This section presents a chronological report of these challenges, including how they were discovered, the characterization experiments conducted to identify their causes, and the conclusions from the final explored registration methods.

2.1 General Method for All Registration Methods Presented

All touch-based registration experiments presented in this Outline section align two point clouds: the first being a 3D scan of a 3D printed knee model taken with a FARO Scan Arm (Quantum S, FARO Technologies, Lake Mary, FL, USA) representing the space of a patient's preoperative image and the

second being a 3D tracing of the knee model taken with the tip position of the FOSS representing the surgical robot's space as seen in Figure 2.1. These two point clouds were then aligned with a MATLAB based ICP algorithm. And the registration accuracy was evaluated as the root-mean square target registration error (RMS TRE) from a certain number of physical targets on the surface of the knee model where TRE is the euclidean distance between the ground truth and registered 3D position of a Target.

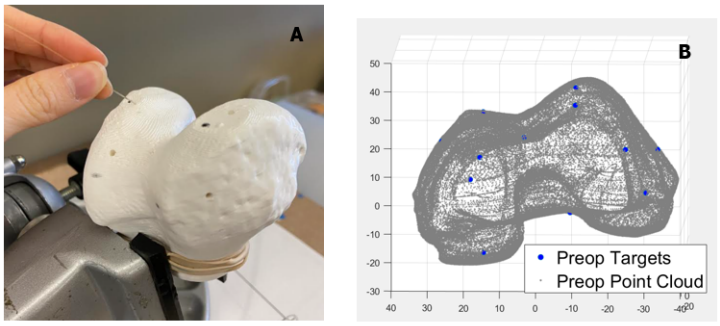

Figure 2.1: A. Surgical point cloud acquired by tracing a 3D printed knee model with the FOSS tip B. Preoperative point cloud acquired by scanning the model with a Faro scan arm

Additionally, it is known that with FOSS technologies position error propagates down the length of the sensor's fiber. In other words, a small error a certain length proximal to the fiber's tip will cause an even larger error in the position measurement of the fiber's tip. Therefore, during these experiments, the fiber's sensing length, or the length of the fiber actively collecting measurements, was set to the minimum length required to trace over the top surface of the knee model while safely avoiding damaging the fiber. This was usually around 35-45 mm of the fiber or about a third of its' full length.

2.2 Initial Touch-Based Registration Results

The initial touch-based registration experiments were conducted prior to completing any major sensor characterization experiments in house. Therefore, they were conducted under the assumption that the FOSS operated within the specifications provided by TSSC at the time of the experiments. These stated that the system had a 60 HZ acquisition rate, a 2 cm bending radius, and was twist compensated. And it was believed that with a proper ICP registration code, the system should have been able to be used as the sole sensing system for touch-based registration.

2.2.1 Methods: Initial Touch-Based Registration Results

During these first registration experiments, the FOSS fiber was held and traced over the knee model surface in the configuration shown in Figure 2.2. In this way, the FOSS's set origin was secured flat onto a table, the knee model was secured upright onto the same table as the FOSS, and in order to trace the model, the working length of the FOSS fiber was arched up and over the knee model. The fiber's tip was then traced across the top surface of the model in a typical drawing motion to acquire a point cloud of that surface to represent the surgical space for registration.

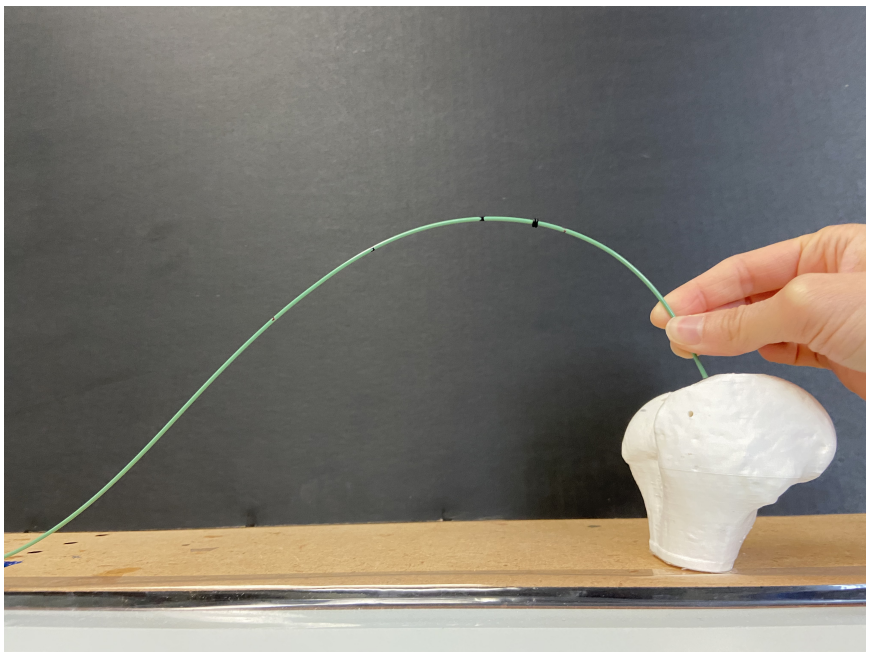

Figure 2.2: FOSS configuration to get the surgical point cloud in the first touch-based registration experiments

2.2.2 Results: Initial Touch-Based Registration Results

This method of touch-based registration yielded unexpectedly high errors. The RMS TRE for the trials shown in Figures 2.3 and 2.4 was 16.7 mm and subsequent trials yielded results in the 10-20 mm range as well. Additionally, by looking at the individual target measurements in Figure 2.4, it can also be seen that the FOSS tip measurements had low precision as there is a lot of variation for the readings for each registered target location. These should have all measured in approximately the same 3D position as they were recorded while holding the tip of the FOSS fiber in the same position on the model.

Conclusions: Initial Touch-Based Registration Results

The resulting TRE values from the initial touch-based registration experiments were much higher than expected given our understanding of the FOSS system at the time of the experiments. Therefore,

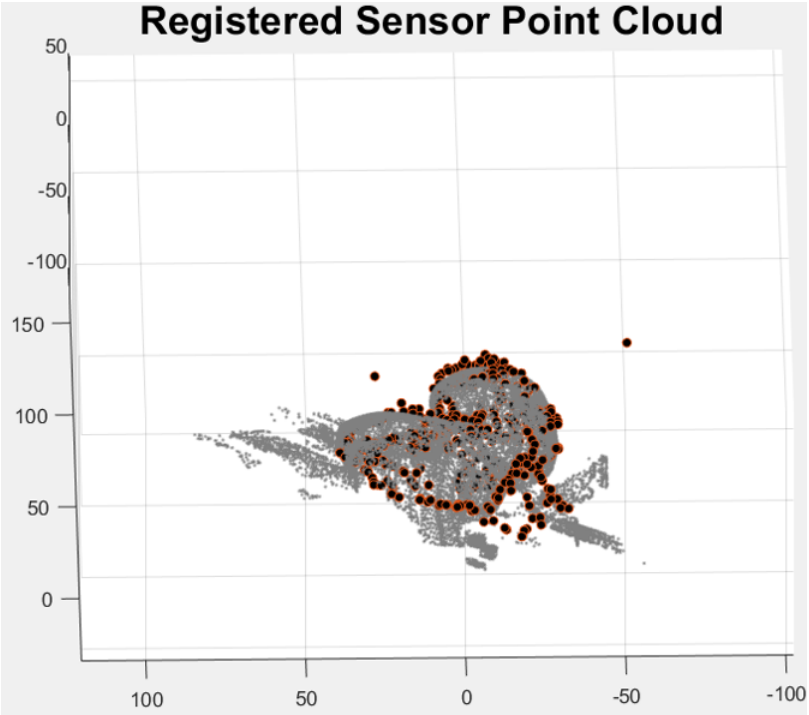

Figure 2.3: FOSS surgical point cloud (Gray points) registered to FARO Arm preoperative point cloud (Black points)

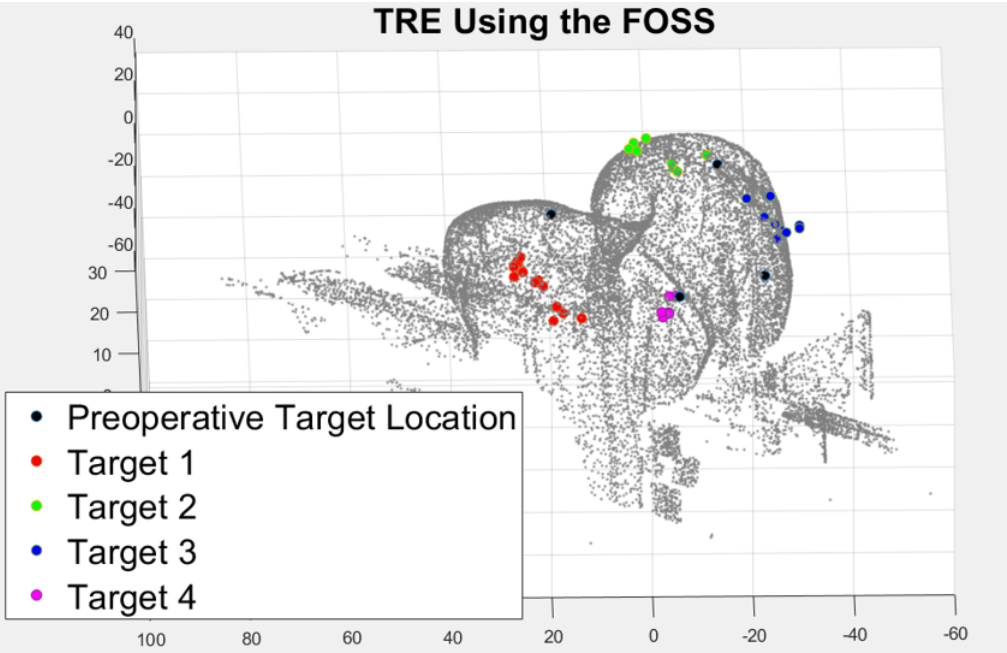

Figure 2.4: FOSS TRE results for a single trial of the drawing method shown in Figure 2.2

there had to be something incorrect with our understanding of the sensing system, the ICP registration code used for registration, or the method used to collect tip position measurements with the FOSS fiber.

2.3 Identifying Sources of Error within Initial Registration Experiments

Following the initial registration experiments in Figures. 2.2-2.4, there were a number of possible sources of error that were identified and tested. These included possible errors in the MATLAB ICP code, the fact that the FARO scan representing the preoperative point cloud contained excess points from the external surroundings of the knee model, or issues inherent to the FOSS. Therefore, each possible source of error was investigated to determine which could have caused the high registration errors observed.

2.3.1 Investigating Possibilities of Coding Errors

To eliminate the possibility of coding errors, the same touch-based registration code and procedure described in Chapter II. section 1. was used and repeated except with the surgical point cloud collected with an Electro-magnetic (EM) tracker rather than the FOSS system. The EM tracking system (Aurora NDI, Inc.) used for this test is well known to provide accurate 3D position results. And so, if carrying out this procedure with the EM tracker still yielded high TRE values, it was likely that there was some error in our own registration method. However, as shown in Figure. 2.5, the registration experiments using the EM tracker consistently yielded low RMS TRE values in the range of 2-3 mm. And looking to the TRE for individual targets, our registration method with the EM tracker, yielded low errors even for points that lay outside of the surgical point cloud. Therefore, this indicated that the errors encountered during the initial registration experiments with the FOSS were likely not caused by the ICP registration code written for this work.

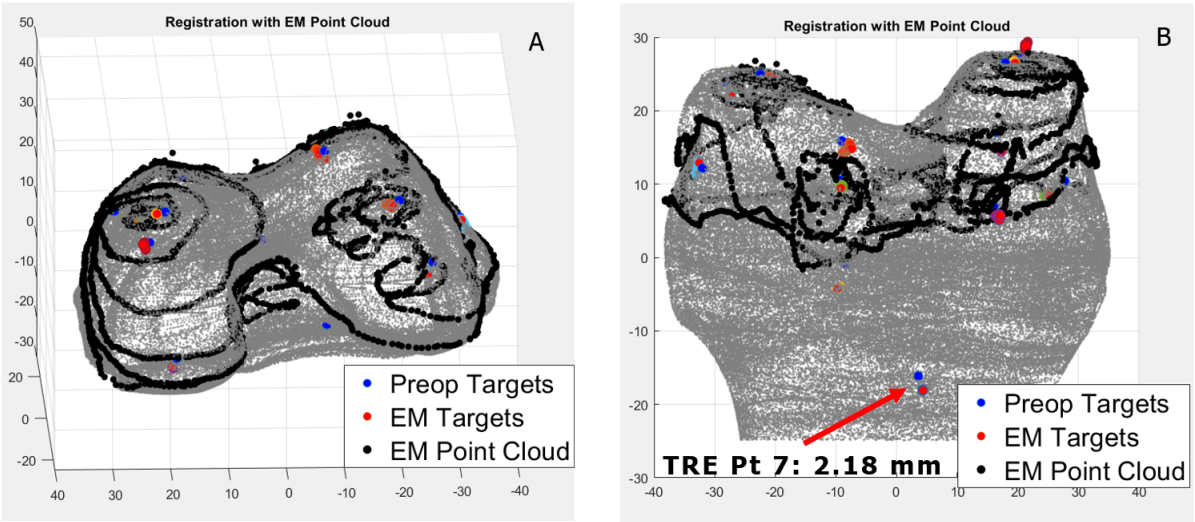

Figure 2.5: A and B. FOSS surgical point cloud (Gray points) registered to FARO Arm preoperative image point cloud (Black points)

2.3.2 Investigating Errors Caused by Incorrect Preoperative Point Cloud

The preoperative point cloud used in the initial registration experiments included points from the knee model's surrounding environment when it was scanned with the FARO arm. This meant that during a registration experiment, the ICP algorithm would attempt to register the surgical point cloud to points that were not actually located on the knee model. And ultimately, this could cause the registration experiment to yield a transformation that incorrectly shifts the point cloud to points outside of the knee model surface.

To test how this effected the initial registration experiments, a new 3D FARO scan of the knee model with 12 targets was made that did not contain any extra points. And this new scan was then used in a second set of registration experiments. In this way, if this second set of registration experiments did not have those high TRE values, it was likely that the extra points in the preoperative point cloud had caused the errors in the initial registration experiments.

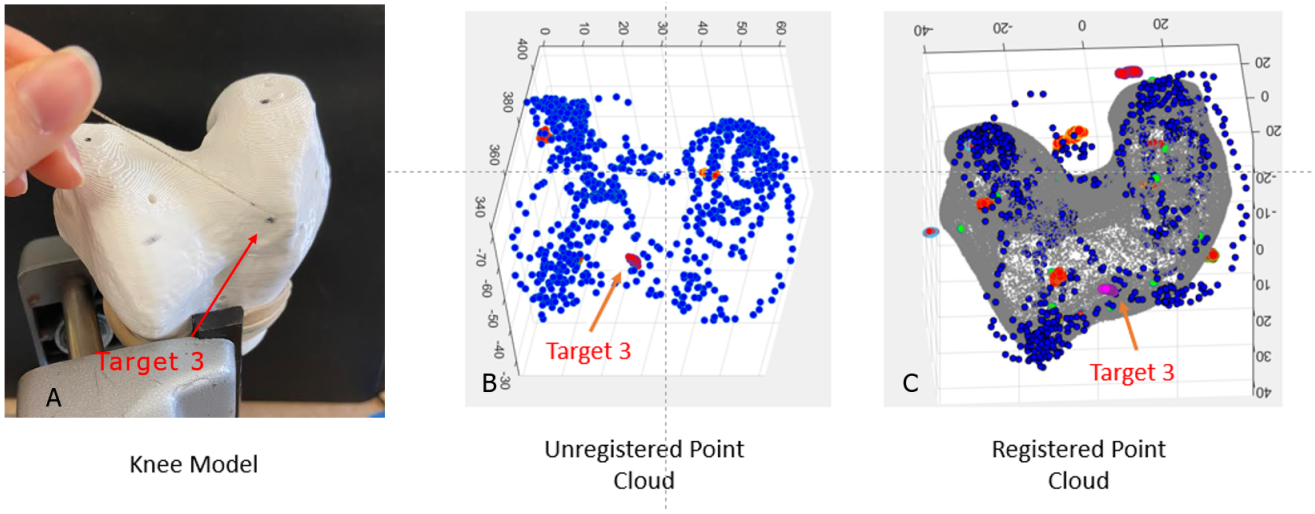

Figure 2.6: A. Physical knee model with the tip of the FOSS pointing to Target 3 B. Measured location of Target 3 (red) within the unregistered surgical point (blue) C. Measured location of Target 3 within the surgical point cloud registered to the preoperative point cloud (gray)

The results of this second set of registration experiments are shown in Figure 2.6 A-C. By visually comparing the positioning of the registered surgical point cloud on top of the FARO preoperative point cloud in Figures 2.3 and 2.6, it can be seen that in Figure 2.6, the registered point cloud is no longer shifted towards the position of the FARO point cloud points representing the surrounding area to the knee model like in Figure 2.3. Therefore, it was concluded that the inclusion of extra points in the iniital preoperative point cloud, likely increased the TRE results in experiments where it was used. However, these registration experiments also yielded visibly high TRE values. So, there was still an unresolved issue within the registration method.

The experiments in this section did provide the first clear insight into what was causing the high

errors seen in the first registration experiments. As seen when comparing Figures 2.6 A and B, the the measured location of Target 3 relative to the unregistered surgical point cloud was not consistent with its physical location on the knee model. Instead, for the images shown here, Target 3 measured about 1 cm to the left of where it should have been. Therefore, when the surgical point cloud is registered to the preoperative one as shown in Figure 2.6 C, Target 3 remains in that incorrect location.

During these experiments, it was observed that when measured multiple times the same location in physical space would frequently measure at different locations within the FOSS system's space. This meant that the raw Target position measurements were not correctly aligned within the raw surgical point cloud before any registration transformation was applied. Additionally, it is important to note that at this point in the investigation, it seemed that the raw tip position errors were more random rather than being due to a steady drift in measurements as they did not seem to follow a pattern in the direction or timings of shift. And overall, these results proved that the high TRE values found during these initial experiments were largely due to issues inherent to the FOSS. Therefore, the FOSS system itself needed to be investigated before moving forward with any more touch-based registration work.

2.4 FOSS Characterization Experiments

There are a number of factors to which FOSS systems are known to be sensitive including light in its environment, temperature, twist, and changes in FOSS configuration towards the proximal end of the fiber during use. Each of these factors were first isolated and tested in a series of stationary characterization experiments to determine whether they could cause measurement errors similar in nature to those encountered during the touch-based registration experiments in sections 2 and 3. The factors that were seen to cause measurement errors were then further tested in another series of tracing characterization experiments to determine how these factors influenced FOSS readings when used in a more similar way to the one used during a touch-based registration experiment.

2.4.1 FOSS Stationary Characterization Experiments

To isolate each of these potential factors for error, stationary characterization experiments were conducted to test how each factor individually influenced FOSS measurements when the fiber was secured in either a straight line or a curved configuration. During the initial touch-base registration experiments, the FOSS was used to trace the irregular surface of the knee model which made it difficult to visibly determine when a measurement error occurred and under which condition it happened. Therefore, these methods were chosen as FOSS readings in these configurations should yield easily recognizable 3D shapes. And the results of these experiments visibly show which factors cause FOSS

measurement errors and which did not. The first factors tested during these characterization experiments were light in the FOSS's environment and changes to the sensor's internal temperature over time as these were the two environmental factors The Shape Sensing Company warned could cause measurement errors. After these experiments, twist along the FOSS fiber was tested. As per the sensor specifications, the system should have been twist compensated. However, twist is a known source of error in FOSS systems and was therefore tested anyways.

Methods: Light in FOSS Environment and Internal Temperature

Here, the FOSS fiber was laid flat against the top of a lab bench and secured in a straight line configuration. To test for the system's light sensitivity, the 3D position of the sensing length of the fiber was then recorded in five minute intervals first with the room lights on and then with them off. Experimental results were analyzed for any steady drift or jump in position readings at the fiber's tip position. Next, to test the effects of internal temperature, the fiber was again secured in the same straight line configuration as in the light experiments and the sensing length's 3D position was recorded in 5 back-to-back 30 minute intervals with the first trial starting from the time the system was first turned on.

Results: Light in FOSS Environment and Internal Temperature

When testing for the effects of light in the environment, the FOSS tip measured in the same position throughout all experiments. And when testing for the effects of internal temperature, FOSS tip position readings exhibited drift within the first half an hour while the sensor was warming up. However, once past this time, it reached an equilibrium point and the tip position readings consistently remained in the same location afterwards.

Conclusions: Light in FOSS Environment and Internal Temperature

Because the tip position remained constant during the environmental light experiments, this indicated that the room lights were not the cause of the tip position errors during touch-based registration. And for the internal temperature experiments, it should be noted that The Shape Sensing Company had initially instructed us to let the system warm for around 30 minutes before use and that all touch-based registration experiments were carried out after this warming time. Therefore, these results also indicate that the internal temperature was not the cause of the high TRE values initially encountered.

Methods: Stationary Twist Applied to the FOSS Fiber

The twist FOSS characterization experiments were carried out next in a similar fashion to the light and internal temperature ones. In this way, the FOSS fiber was secured flat onto the lab-bench table

in a straight line configuration and its tip position was marked with an x. A piece of tape was then stuck perpendicularly onto the fiber so that it pointed upwards from the tabletop to visually determine the angle of twist at which the fiber was measured.

During the experiment, the tip position of the FOSS was first measured in that configuration and location with 0 deg of twist applied. The fiber was then moved from that position, manipulated, and then secured back into the exact same location without twist to test whether the FOSS could measure the same position in space twice accurately. From there, the FOSS was again removed from that position twisted and secured to the same straight line configuration and location and the tip position was again measured. These experiments were conducted while adding +90 and +180 deg twist to a 45 cm sensing length.

Results: Stationary Twist Applied to the FOSS Fiber

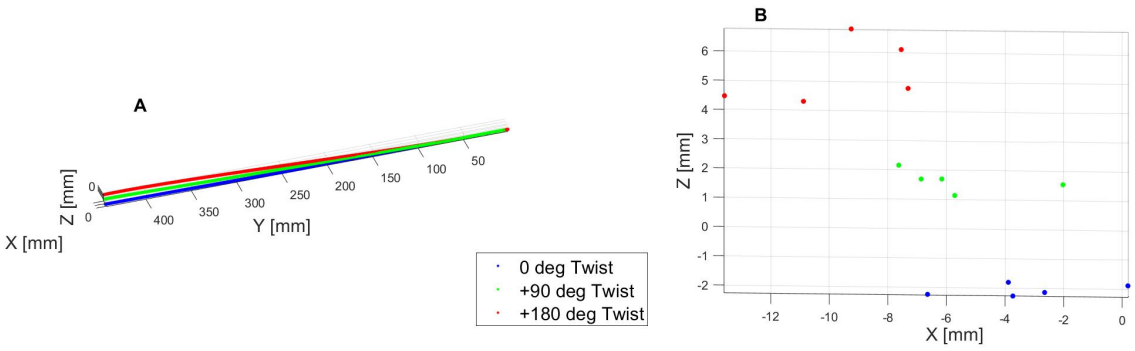

Figure 2.7: (A) Full FOSS data in a straight line configuration and 0,+90, and 180 deg of twist (B) Mean tip position with the FOSS under 0,+90, and 180 deg of twist

Conclusions: Stationary Twist Applied to the FOSS Fiber

It can be seen from both figures, that adding twist caused the tip position to measure at a distincitly different 3D location in the space of the FOSS despite the fact that the tip was located at the same location in physical space for all trials. As noted in section 3.2, the FOSS measurement errors observed during the initial touch-based registration experiments were not that of a constant drift in measurement. But instead, the tip position measurement had distinct jumps in 3D space from where they logically should have been measured relative to each other. Therefore, the results here are consistent with the measurement errors known to be the cause of the high TRE values in the initial registration experiments. Lastly, it should be noted that the The Shape Sensing Company markets their Pathfinder Platform used in these studies as able to measure twist and twist compensated. However, after conducting these experiments, the author found by reading the company's blog that the fiber should not be twisted near 360 deg or one full rotation of twist and that twist could cause measurement errors of up to 1 cm. Therefore, even though we expected the sensing system to be twist compensated going into our touch-based registration work, the results found during this study is consistent with the manufacturer's own testing.

2.4.2 FOSS Tracing Characterization Experiments

This section of work aimed to test how the factors identified during the stationary characterization experiments impacted tip position measurements when the fiber's tip was traced along the outline of a square printed on a flat surface. These experiments were chosen to test these factors under conditions more similar to those during a registration experiment. Because during those, the fiber tip is traced across the surface the knee model to create the surgical point cloud.

Methods: Tracing Characterization Experiments

A second series of characterization experiments were conducted to test the behavior of the FOSS system when used in a tracing motion similar to the one used in the creation of the surgical cloud during touch-based registration. Here, the tip of the FOSS was traced along the outline of a 6x6in square and a point cloud of the measured tip position throughout tracing was collected to determine how successfully the FOSS could measure the square's shape and dimensions. Additionally, unlike when tracing along the irregular 3D surface of the knee model, when tracing the square the proximal end of the FOSS could be held at relatively the same configuration and without adding twist. Therefore, to isolate and test the effect of twist and configuration changes on tip position error, these tracing experiments were conducted first while keeping the proximal end steady, and then subsequently while purposely adding

changes in twist or configuration to the proximal end of the FOSS. Lastly, early testing suggested that the positioning of the square relative to the FOSS effected how well the proximal end of the FOSS could be held steady while tracing. Therefore, this process was first carried out with the square facing horizontally and parallel along the length of the FOSS similar to how the knee model was positioned during the initial registration experiments, and then again with the printed square facing vertically to the FOSS and perpendicular to the length of the fiber.

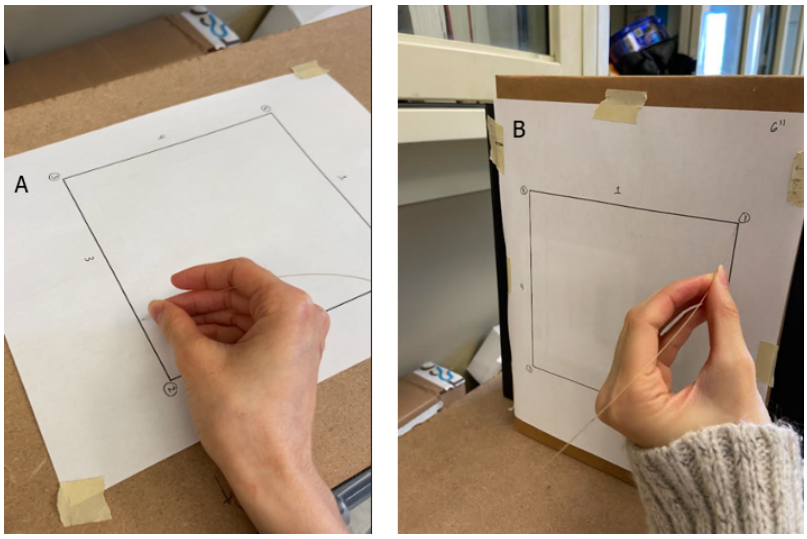

Figure 2.8: A. Top view of tracing the fiber along the square positioned horizontally along the tabletop B. Side view of tracing the fiber along the square positioned vertically to the tabletop

Results: Tracing Characterization Experiments

Figures 2.9 and 2.10 show the point clouds measured with the square placed horizontally on the test bench and parallel to the FOSS. Here, Figure 2.9 shows the tracings taken while maintaining the most stable proximal end FOSS configuration as possible. The measured point clouds here indicate that when used in this manner, the FOSS consistently measures the tip position at relatively the correct shape and remains at relatively the same 3D position and orientation within FOSS space during all three trials. However, the point clouds also have the same incorrect dimensions as the three point clouds shown here are rectangles with dimensions of 7.8x6.7 in, rather than three squares with dimensions of 6x6 in.

Figure 2.10 compares the tip position point clouds collected when tracing the horizontal squares without twist or configuration changes, with those collected with twist and configuration changes. It shows that introducing either twist or configuration changes during tracing cause significant errors in the tip position measurements, as the resulting point clouds are not shaped like the 2D square that was traced and all measure at different 3D positions and orientations within FOSS space. This is unlike

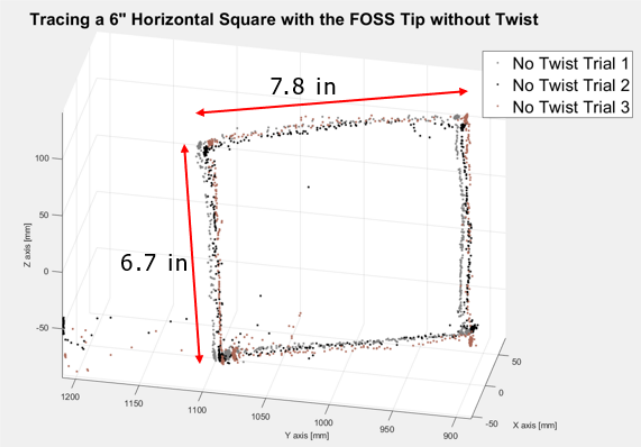

Figure 2.9: Three point clouds of the FOSS tip positions when tracing along a 6x6in printed square placed horizontally along the tabletop

the resulting point clouds collected without introducing these factors which were shaped relatively like squares and measured consistently at the same 3D position and orientation

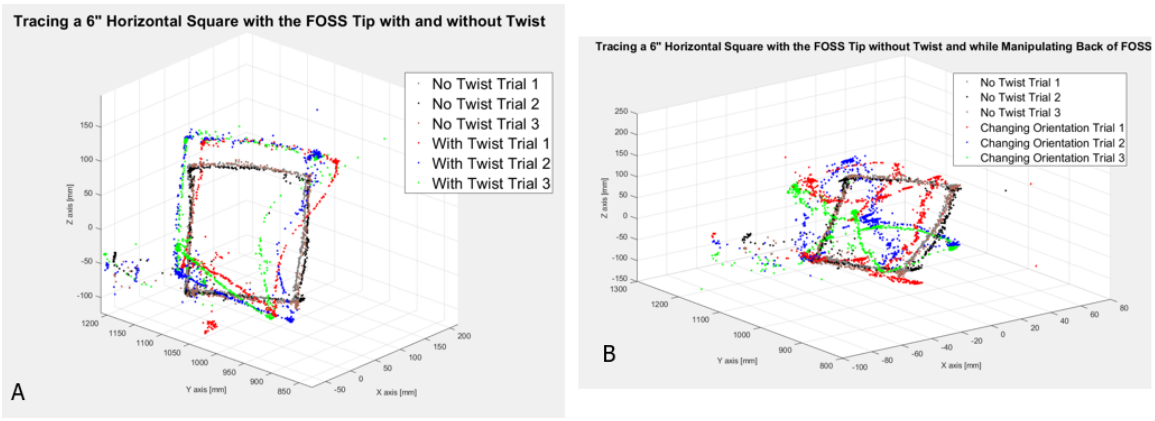

Figure 2.10: Six point clouds measured with the square placed horizontally along the test bench with and without A. twist or B.configuration changes

Figure 2.11 shows the point clouds measured with the square placed vertically to the test bench and perpendicularly to the FOSS and compare the point clouds without twist or configuration changes with those collected with twist and configuration changes. As seen in both figures, unlike with the horizontal trials, here when the proximal end of the FOSS was kept steady, the measured tip position consistently formed point clouds of the correct 6x6in square dimension and measured at a consistent 3D position and orientation within FOSS space.

And similar to the horizontal measurements, Figure 2.11 shows that tracing the vertical square while twisting the FOSS's proximal end led to errors in point cloud shape and dimension as well as caused the tip positions to be measured at different 3D positions and orientations in comparison to each other and the original vertical measurements taken without twist. However, as seen in Fig 13 B,

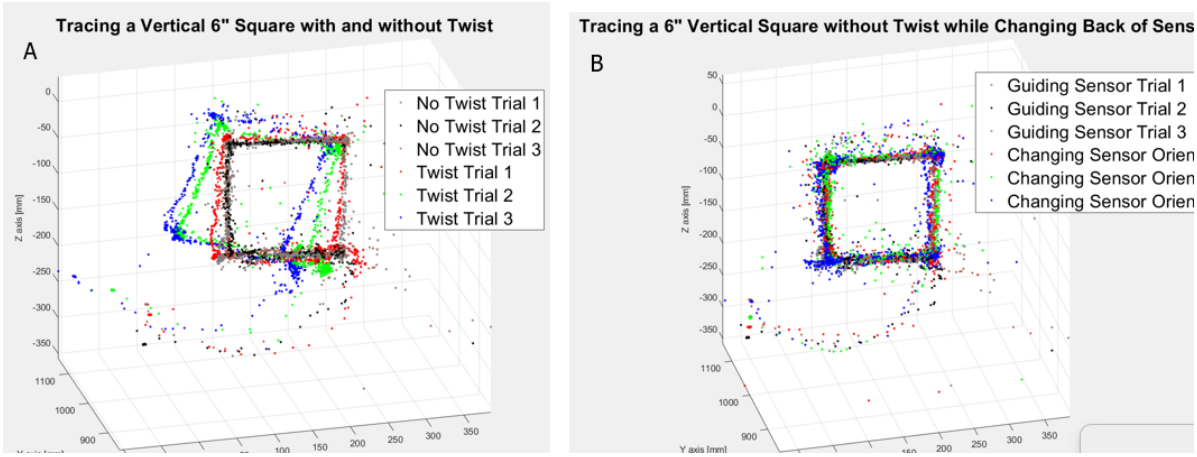

Figure 2.11: Six point clouds measured with the square placed vertically along the test bench with and without A. twist or B.configuration changes

when tracing the square vertically, introducing configuration changes only increased the noise in the tip position measurements, and here the shape, dimension, 3D position and orientation were consistent with the measurements taken vertically without twist or configuration changes.

Conclusion: Tracing Characterization Experiment

The tracing characterization experiments confirmed that introducing twist or configuration changes to the FOSS could cause measurement errors as well as changes in the 3D position and orientation within FOSS space at which the tip was measured at for a given position in task space. During the initial registration experiments before FOSS characterization, the FOSS tip positions were observed to measure at different 3D positions within FOSS space for a given position on the physical knee model in task space. And these changes in FOSS space position were largely what had caused the high TRE values during those experiments, Therefore, given these results as well as how easy it is to introduce twist or configuration changes during knee model tracing, these results show that both twist and changes in FOSS configuration are likely causes for the jumps in position during the initial registration experiment.

Lastly, these experiments also suggest that the positioning of the knee model relative to the FOSS during the initial registration experiments could have contributed to the high TRE values measured. During those experiments, the surface of the knee model was positioned parallel to the length of the FOSS similar to the positioning of the square during the horizontal tracing experiments. As shown in Figure 2.10, when in the horizontal positioning, the FOSS tip position measurements had high error even when the proximal end of the fiber was held steady. In comparison, when tracing the square in the vertical position like in Figure 2.11, the square dimensions were measured accurately when the proximal end was held steady. Additionally, as seen by comparing figures 2.10 and 2.11, changing

the fiber's proximal end configuration during tracing caused less error in tip position measurements when tracing in the vertical position than the horizontal. And unlike when in the horizontal position, configuration changes more lead to an increase in noise rather than high measurement errors or changes in 3D position and orientation when in the vertical position.

2.5 Experiments Testing Methods to Reduce the Effects of Twisting and Changing the Configuration of the Proximal End of the Fiber on Tip Position Measurements

Both the stationary and tracing characterization studies identified twist and configuration changes along the proximal end of the fiber as major sources of the tip position measurement errors observed during the prior touch-based registration experiments. And unfortunately, due to the irregular surface of the knee, it is difficult to avoid either condition when tracing the surface and measuring target positions. Therefore, this section of experiments aimed to explore methods of actively avoiding or undoing the negative effects of these two factors.

2.5.1 Methods: Experiments Testing Methods to Reduce the Effects of Twisting Proximal End of the Fiber on Tip Position Measurements

To reduce twist, the FOSS fiber was sheathed in different types of tubing and then the typical touch-based registration method described in sections 1 and 2 of this outline were carried out. For this, Polyamide, Teflon, pneumatic, and PEEK tubing were used. And the tubing's inner diameters were chosen so that there was a close fit between the tubing and fiber, but so that the fiber could freely rotate within the tubing. Therefore, when twist was inevitably applied while tracing the surface of the model, the twist would be applied to the tubing and not the fiber. An image of this method is shown in Figure 2.12.

2.5.2 Methods: Experiments Testing Methods to Reduce the Effects of Changing the Configuration of the Proximal End of the Fiber on Tip Position Measurements

These experiments aimed to reduce the effects of configuration changes rather than the occurrence of these changes, because in order to move the fiber tip position, the proximal end configuration must also change to some extent. To do this, rather than move the fiber along a C-bend. The sensor was configured in an S curve as shown in Figure 2.13. The idea being that because an S-curve is made

Figure 2.12: Top view of tracing the surface of the knee model with the FOSS fiber sheathed in polyamide tubing

from two identical C-bends in opposite directions, the errors from the two bends might be equal and opposite and therefore, might have close to a net zero error at the tip position.

To do this, a piece of paper with one straight line and five S-shaped curves that all began and ended at points along the straight line was printed. The paper was secured flat along the same tabletop as the FOSS system. And the position along the length of the FOSS was then recorded first while the FOSS fiber was held stationary along the straight line, and then when held stationary with the fiber bent along each S-curve.

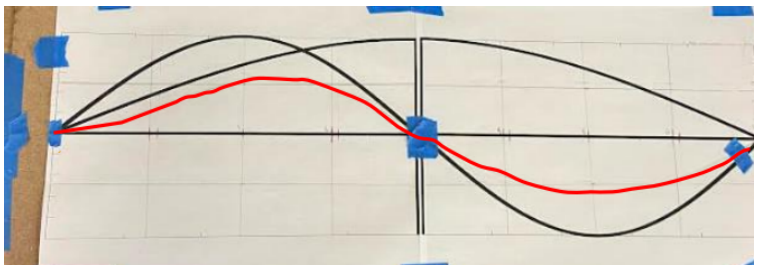

Figure 2.13: Top view of the printed line and S-curves the FOSS fiber was measured along

2.5.3 Results: Experiments Testing Methods to Reduce the Effects of Twisting Proximal End of the Fiber on Tip Position Measurements

Figures 2.14 and 2.15 show two examples of the surgical point clouds and target positions measured when carrying out the touch-based registration experiments with the fiber sheathed in some type of tubing. Similar errors were seen with all tubing tested in this work.

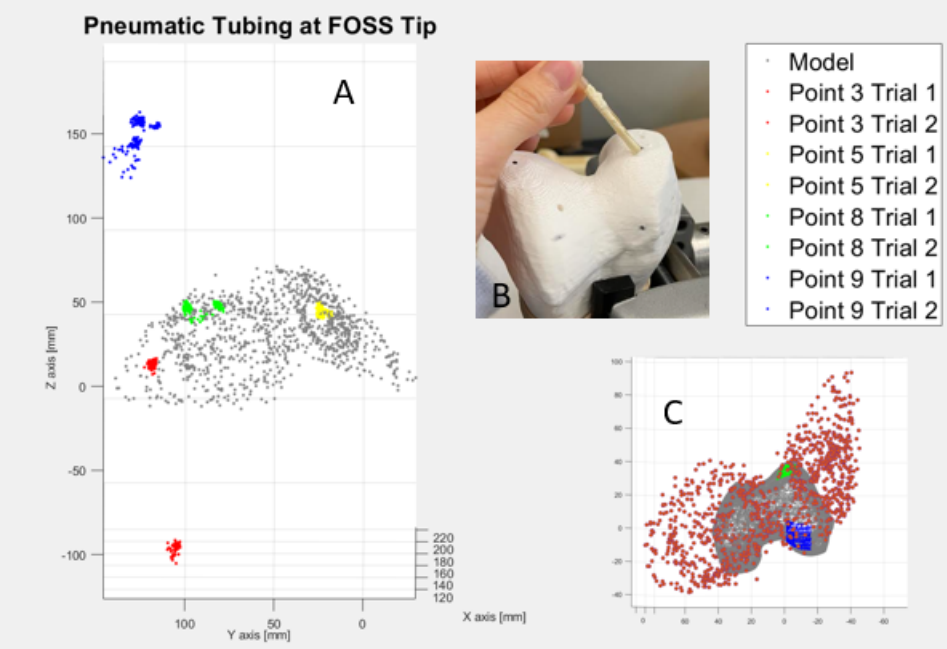

Figure 2.14: A. Surgical point cloud and target positions acquired when sheathing the FOSS fiber with pneumatic tubing B. Top view of tracing with the FOSS in tubing C. Measured surgical point cloud registered against the preoperative point cloud (both clouds are at the same scale)

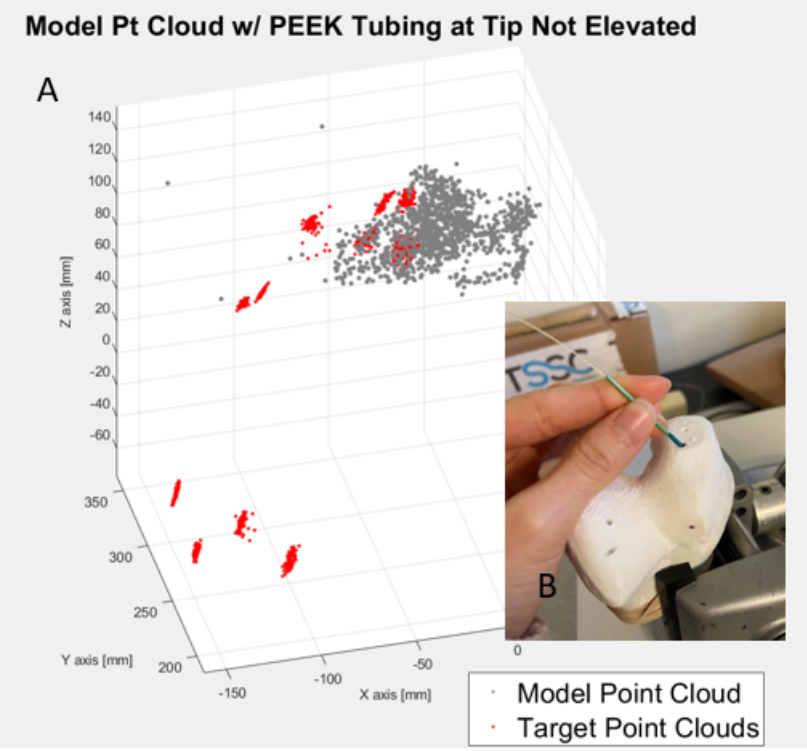

Figure 2.15: A. Surgical point cloud and target positions acquired when sheathing the FOSS fiber with PEEK tubing B. Top view of tracing with the FOSS in tubing

2.5.4 Results: Experiments Testing Methods to Reduce the Effects of Changing the Configuration of the Proximal End of the Fiber on Tip Position Measurements

Figure 2.16 visually shows the measured fiber 3D shape and position for the straight line and S-curves in one trial. And also reports the error as the distance between the tip position of each S-curve and the closest point on the straight line. This figure only depicts one trial, however, three subsequent trials all yielded similar errors

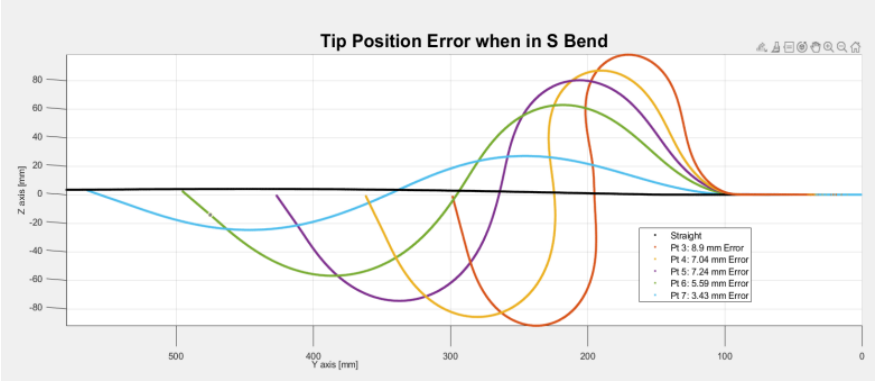

Figure 2.16: Measured positions of the length of FOSS in the straight line and S-curve positions. Distances between the tip of the S-curve and the closest point on the straight line are also reported

2.5.5 Conclusions: Experiments Testing Methods to Reduce the Effects of Twisting and Changing the Configuration of the Proximal End of the Fiber on Tip Position Measurements

Unfortunately, both methods actually increased the error of the tip position measurements. As seen in Figures 2.14 and 2.15, sheathing the FOSS fiber during testing actually increased the tip position measurement error to such a high extent that the resulting surgical point cloud did not represent the surface of the knee model in size or shape at all, Additionally, the raw measured target positions often were not located within the measured surgical point cloud. And similar issues were seen across multiple trials when sheathing the fiber in all four types of tubing tested. These high errors might have occurred because when tracing the model, the fiber likely interacted with the sides of the tubing and interfered with the measurements along the fiber. Due to these results, this method was determined be an ineffective way of reducing twist and was abandoned.

Similarly, as seen in Figure 2.16, the method of measuring the length of the fiber along S-curves was also not effective at counteracting the errors caused by configuration changes. During this test, the tip of each measured S-bend should have measured along the measured straight line if this method yielded results consistent with how the fiber was positioned in physical space. However, this was not the case and the tip position instead measured anywhere from 3-9 mm off of the line. It should also be noted that this error did not increase, decrease, or follow a pattern with how sharp the S-curve was in any experimental trial. Therefore, this method was also abandoned.

2.6 Touch Based Registration Experiments: Knee Model Targets Perpendicular to the FOSS

The initial touch-based registration methods presented in sections 1 and 2 were modified to address the findings of the FOSS characterization studies and then tested. The new registration method aimed to reduce the twist and proximal end configuration changes introduced when tracing the knee model as those factors were found to increase error during both the stationary and tracing characterization studies. Additionally, the knee model was positioned so that its top surface was perpendicular to the length of the FOSS fiber as during the tracing characterization study, this configuration was found to have less tip position error.

2.6.1 Method: Touch-Based Registration with Perpendicular FOSS Fiber and Knee Model Position

During these experiments, the knee model was secured so that it's top surface was perpendicular to the length of the FOSS fiber and directly facing the tip so that the knee model and fiber were in the positioning shown in Figure 2.17. To get the surgical point cloud, the tip of the FOSS was traced along the top surface while keeping the first two centimeters of the fiber from the tip perpendicular to the knee model surface. The distal end of the FOSS was held in this perpendicular position so that the fiber approached the model surface at the same angle throughout the tracing. And when moving the FOSS to create the surgical point cloud, the middle of the fiber was compressed and elongated along the same direction to reduce proximal configuration changes and twist along the fiber.

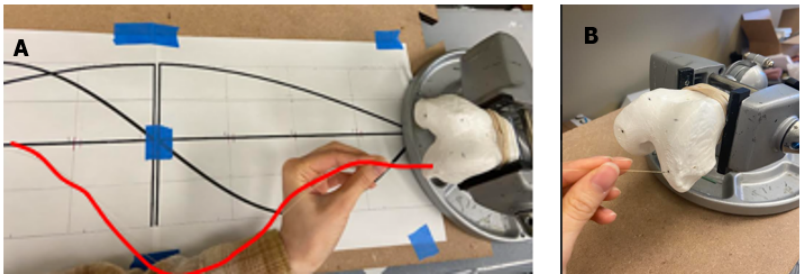

Figure 2.17: A. Top view of how the FOSS fiber was held during the perpendicular registration method trials with the fiber highlighted in red. B. Side view of how the last 3-4 cm was held normal to the model's top surface when tracing the surgical point cloud

Additionally, only 40 cm or around a third of the length of the FOSS fiber was used to trace the FOSS. This was determined to be around the minimum length of the FOSS fiber required to trace the entire surface of the model with this registration method while also not putting too much force on the fiber.

Like in section 2, the preoperative point cloud was a FARO arm scan of the knee model and the registration accuracy was measured as the Target Registration Error from twelve physical targets on the knee model of known locations within the preoperative point cloud.

2.6.2 Results: Touch-Based Registration with Perpendicular FOSS Fiber and Knee Model Position

Figure 2.18 here shows two examples of surgical point clouds and their corresponding registered target locations that were collected with this registration method. It should be noted that because the FOSS fiber became more rigid when manipulated in this manner, three targets became unreachable on the surface. Therefore, the two targets in Fig 20.A that are not located on the surface of the surgical point cloud actually did measure as expected and were not included in the overall RMS TRE values for each

trial with this method. The RMS TRE distribution from the nine trials taken with this method is shown in Figure 2.19.

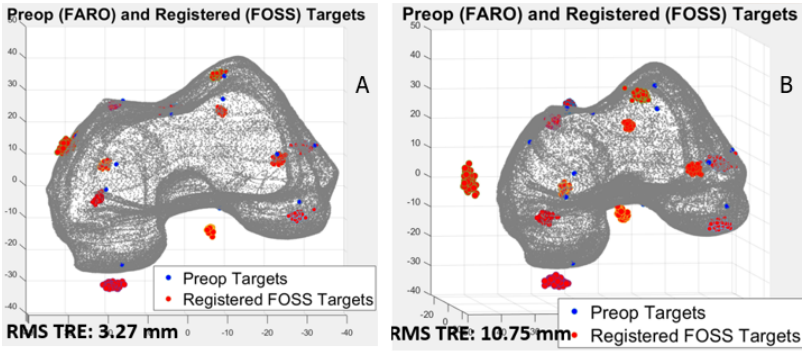

Figure 2.18: A. Example of a surgical point cloud and corresponding registered target points that A. mostly avoided large FOSS measurement errors with this method B. still had large FOSS measurement errors with this method

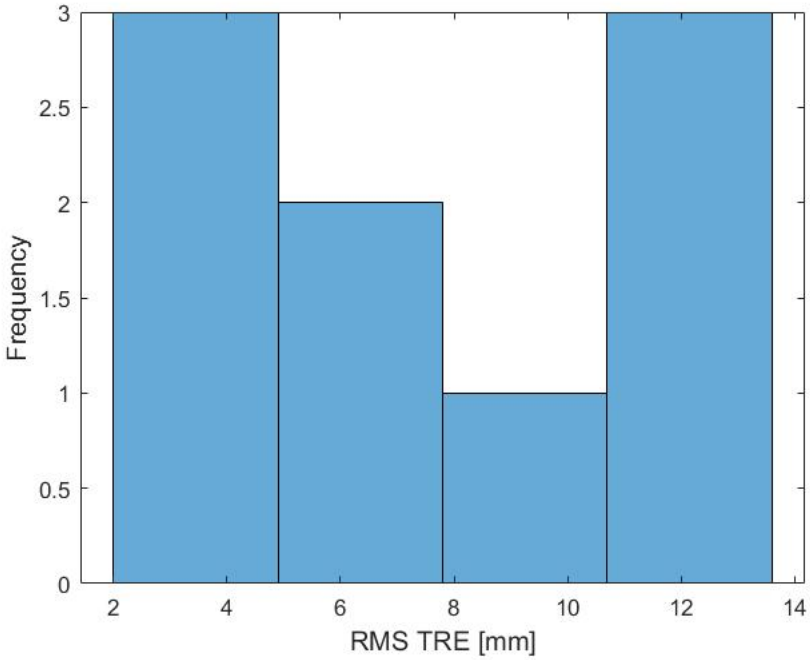

Figure 2.19: Distribution of RMS target registration errors for 9 trials

2.6.3 Conclusions: Touch-Based Registration with Perpendicular FOSS Fiber and Knee Model Position

These registration experiments yielded lower TRE values than the touch-based registration methods presented in sections two and three that did not mitigate twist or proximal end configuration changes during tracing. And of the nine trials, there were three with RMS TRE values in the 3-4.5 mm range

which, while not within our overall submillimetric error goal, were much lower than the 10-20 mm registration errors observed during the experiments in sections two and three. However, as can be seen in the spread of the data, the results were still inconsistent and this method also yielded TRE values around 10 mm.

This registration method attempted to reduce twist and configuration changes by holding the 40 mm length of the fiber in as close to a straight line as possible. However, holding and manipulating the FOSS fiber in this method is quite difficult to physically maintain. The surface of the knee model is quite irregular. Therefore, the proximal end fiber configuration needs to change in order for the tip to trace over the surface. Also, in order to keep the fiber in as straight of a configuration as possible while tracing over that irregular surface, the user conducting the test must work against the fiber as this is not a configuration that the fiber can hold naturally. This caused a palpable amount of tension to build in the fiber as the fiber would occasionally try to reconfigure itself into a more natural configuration. This likely caused fiber twist and inadvertently led to how inconsistent the results were.

Lastly, this method caused the fiber to became a lot more rigid in comparison to when it was traced freely during the methods shown in sections two and three. One of the main advantages of FOSS technologies is that the sensor is flexible and could be integrated into a soft robot manipulator. Therefore, this method was also not compatible with our intended soft robotics application.

2.7 Touch Based Registration Experiments: FOSS Fiber Configured in a Backwards Loop towards Knee Model Surface

The last FOSS only registration method was developed to reduce twist and proximal end configuration changes similarly to the method in section 6. However, it also aimed to avoid the build up in tension along the fiber observed during that method. Therefore, rather than keeping the fiber as straight as possible, this method intended to have the FOSS fiber bend at relatively the same curvature along the length of the fiber and follow relatively the same angle of approach to the model. Throughout this method, the fiber could be held loosely and it still maintained the same curvature. This ensured that the fiber remained flexible, and avoided the build-up tension and resulting twisting issues observed during the previous registration experiments.

2.7.1 Methods: Touch-Based Registration with the FOSS Fiber Configured in a Backwards Loop towards Knee Model Surface

As seen in Figure 2.20, during the backwards loop registration experiments, the origin of the fiber's sensing length was secured at an upwards 45 deg angle rather than secured flat on the same tabletop

as the knee model like in all previous registration methods. The fiber was then curved backwards along the curvature set by the angle of the fixture at the origin. And the knee model was clamped in a vice positioned between the secured fiber origin and the optical network of the FOSS as well as right underneath where the tip of the FOSS naturally fell when wrapped along the backwards loop. During testing, the fiber tip was traced along the top surface of the knee model to create the surgical point cloud and collect target positions while the length of the fiber remained in this backwards loop configuration.

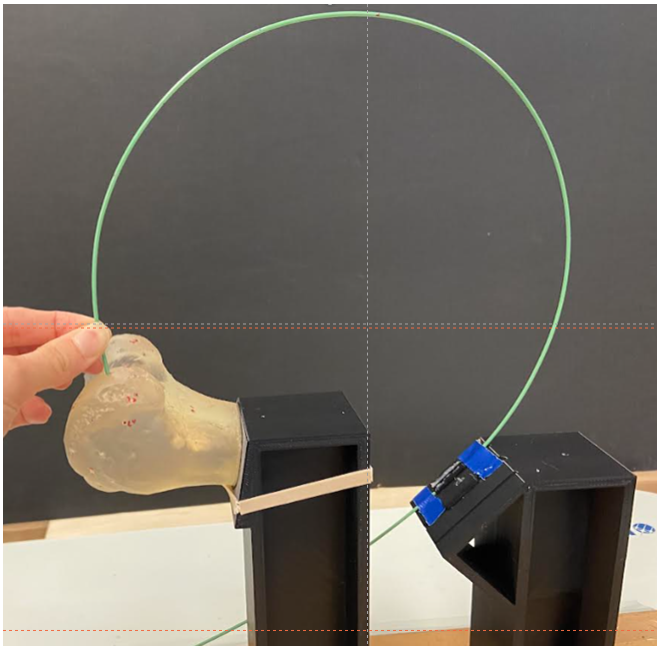

Figure 2.20: Side view of the FOSS and knee model in the backwards loop configuration

Again, only 40 cm or around a third of the length of the FOSS fiber was used to trace the model. This was determined to be around the minimum length of the FOSS fiber required to trace the entire surface of the model while not bending the FOSS fiber at a curvature lower than the minimum 2 cm radius of curvature defined by the The Shape Sensing Company.

2.7.2 Results: Touch-Based Registration with the FOSS Fiber Configured in a Backwards Loop towards Knee Model Surface

Figure 23 shows the histogram of the RMS TRE results for 120 experimental trials using the FOSS backwards loop registration method.

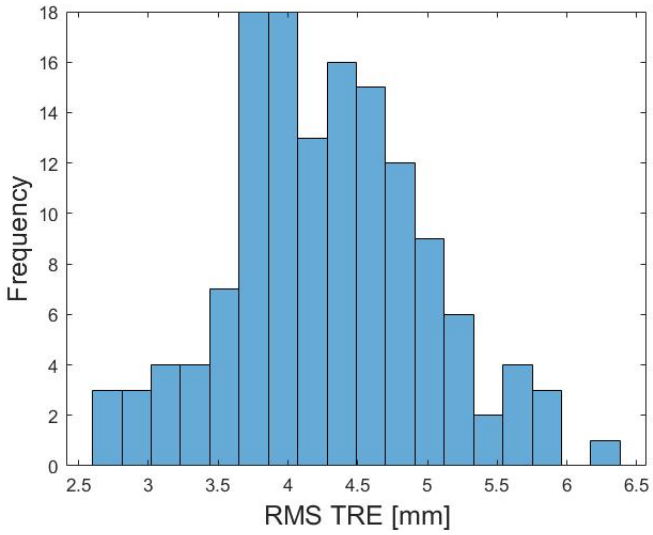

Figure 2.21: Distribution of RMS target registration errors for 120 trials

2.7.3 Conclusions: Touch-Based Registration with the FOSS Fiber Configured in a Backwards Loop towards Knee Model Surface

The RMS TRE results from the backwards loop FOSS registration method were the lowest and most consistent of any registration method experimented with during the course of this thesis work. The mean RMS TRE values across all trials was 4.27 mm which is much lower than the error values in the 10-20 mm range that were observed in all of the previous registration methods. And out of the 120 trials presented in Figure 2.21, there were no trials that yielded RMS TRE values around the 10 mm range which was observed from every other registration method.

During this method, the tip of the fiber is both able to reach every point along the the model's top surface and does not sacrifice any of the FOSS fiber's flexibility. By bending the fiber backwards from an origin set at an upwards angle, the fiber is naturally forced to follow a similar curvature throughout tip tracing rather than needing to be forcibly manipulated into a particular configuration by the user. And from a more qualitative perspective, this causes the fiber to feel the least tense during tip tracing in comparison to every other registration method explored during this thesis. This is important because when tension builds along the fiber it is harder to manipulate smoothly, and it then becomes very easy for the FOSS to twist or change proximal end configuration because the fiber is working against whatever is manipulating it.

Additionally, one reason this backwards loop configuration was explored is because the C-bend shape the FOSS bends along on is very similar to the type of shape C-bend fluid powered soft robots form under actuation. Therefore, a FOSS fiber could be integrated into a C-bend like actuator and be used to accomplish this registration method.

However, the RMS TRE values observed here were still much higher than our initial goal of submillimetric errors. Therefore, this method still needs to be modified in order for the FOSS to be used as a viable sensing method for surgical touch-based registration image guidance.

2.8 FOSS - EM Tracking Fusion Registration Method

At this point in the project, it was concluded that the FOSS system would likely not be viable as the sole sensing system for a touch-based registration project that required submillimetric accuracy. And the project group decided that a FOSS and EM tracking based sensor fusion method should be explored to develop a touch-based registration method that met the required accuracy. Overall, this sensing method aimed to maintain the flexibility and continuous nature of the FOSS system that was advantageous for soft robotic shape sensing, but could hopefully use the EM tracking measurements to correct for the high 3D position errors from the FOSS system by providing accurate measurements at certain points along the fiber.

2.8.1 Methods: FOSS - EM Tracking Fusion Registration Method

For this method, one or two EM tracking coils were secured at known points along the length of the FOSS fiber to provide trustworthy 3D position measurements at the position of the corresponding FOSS readings. And the FOSS and knee model were secured at the positioning shown in Figures 2.22 and 2.23 similarly to what was used in the backwards loop registration methods. To collect the surgical point cloud, the fiber was then curved along the surface of the model and held in place while collecting FOSS and EM tracker data. This was repeated at multiple positions along the model's surface in the medial and lateral condyle regions to collect point lines from the surface of the knee model in FOSS space and one or two positions in EM tracker space that corresponded to known positions along the surface of the FOSS fiber. Target positions were then collected with an EM tracking coil.

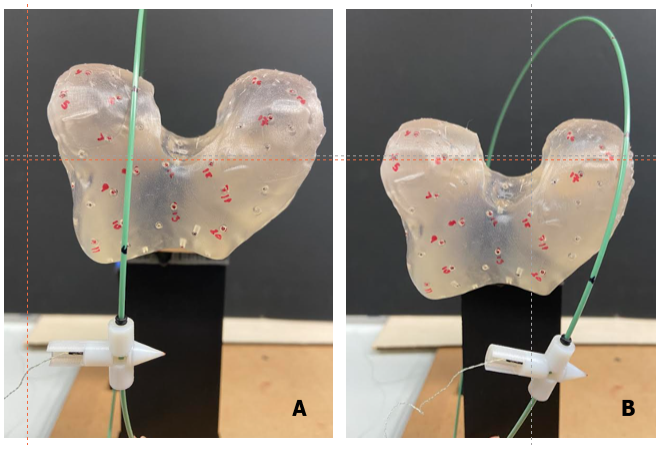

Figure 2.22: Front view of the FOSS in green PEEK tubing and connected EM tracking coil with working region of the FOSS manually held against the (A) medial and (B) lateral femoral condyle regions

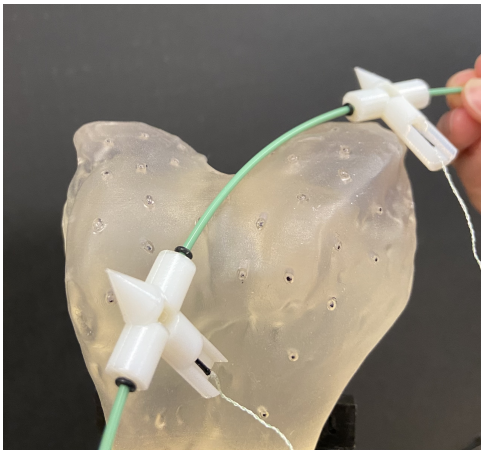

Figure 2.23: Top view of sensor fusion method with two magnetic tracking coils before and after region in contact with model

2.8.2 Results: FOSS - EM Tracking Fusion Registration Method

The experimental results from using one EM tracking sensor with the FOSS is shown in Figure 2.24.

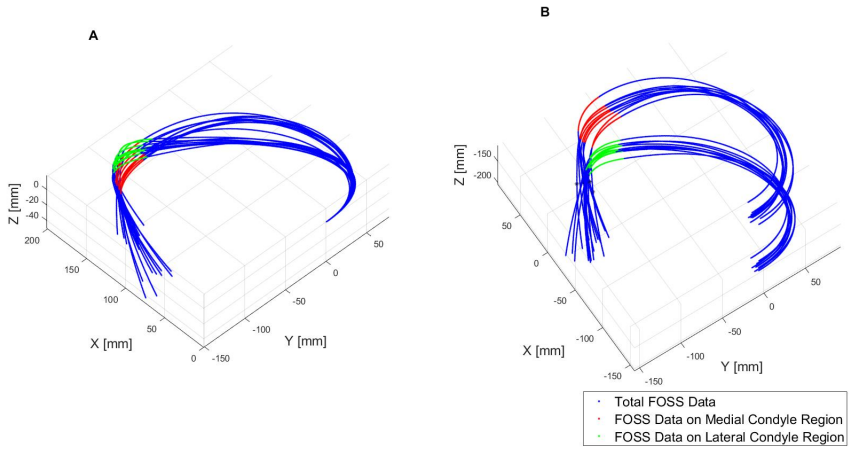

Figure 2.24: Full FOSS measurements with working lengths in contact with knee model highlighted (A) without (B) with position correction from EM tracker data

And the experimental results from using two EM tracking sensors with the FOSS is shown in Figure 2.25.

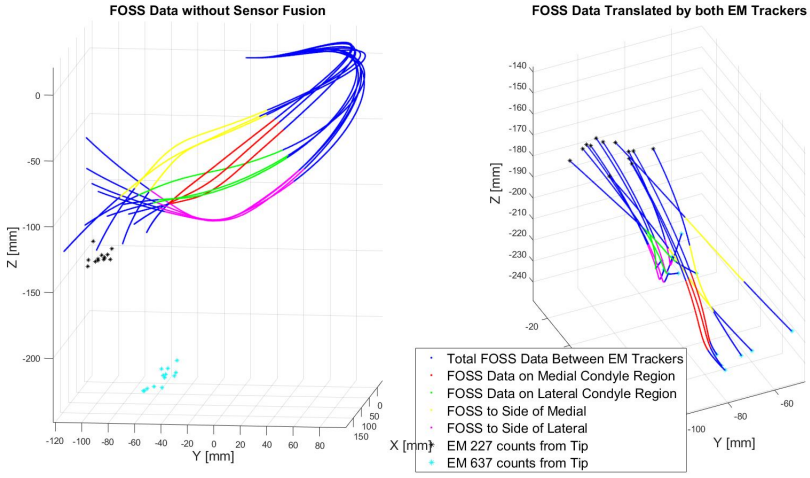

Figure 2.25: Full FOSS measurements with sensing lengths in contact with the knee model highlighted (A) without (B) with position correction from two EM trackers.

2.8.3 8.3 Conclusions: FOSS - EM Tracking Fusion Registration Method

The experiments with one EM tracker did not provide a successful registration method. However, they did provide insight into the errors encountered during the registration experiments with only the FOSS system. Given the physical locations of the FOSS fiber shown in Figure 2.22 used to create the surgical point cloud, the red and green regions in Figure 2.24 should be two distinct regions located adjacent to each other. However, without 3D position correction from the EM tracker, they measure

crossed over each other. This is because when the FOSS fiber is curved in a particular direction, a position error then builds with each measurement down the fiber from that curve in the direction that the fiber was curved. And when the fiber is curved along a different direction, the position error builds in the direction of that curve. During the collection of the surgical point cloud for the FOSS only registration methods, the FOSS fiber had to be bent at different curvatures in different directions in order to trace the fiber tip along the surface of the model. Therefore, given the results shown here, that was likely leading to the collection of a point cloud and target point positions made from points of different errors in different directions which is consistent with how the tip position measurements seemed to have random position errors during these experiments.

And as seen in Figure 2.24. B, using the EM trackers to correct for FOSS position can improve results. Because the point lines from the medial and lateral regions are uncrossed with EM tracker position correction, the results are more consistent with the physical system. However, it can be visibly seen that this method does not fully correct for position errors. As seen by how the FOSS fiber was placed along the knee model surface in Figure 2.22, these lines should form one continuous surface. However, as seen in these figures, some of these point lines are still located above or below the surface formed by the other point lines in their region. And so these methods do not correct for FOSS measurements well enough to create a point cloud that accurately describes the surface of the knee model.

Additionally, Figure 2.25 shows the results of adding two EM tracking coils onto the fiber to provide accurate position readings before and after the highlighted regions in contact with the knee model. Like in Figure 2.24, Figure 2.25 A shows how the point lines incorrectly measure as being crossed due to the FOSS curvature errors. And Figure 2.25 B shows the results of fitting the FOSS measurements between the two coils to the 3D locations of the EM coils. Like in 2.24 B, this method uncrossed the data. However, it also warped the data and so the resulting highlighted regions that should represent the surface of the knee model were not oriented properly to form that surface.

And unfortunately, these methods were never fully explored or completed before the conclusion of this thesis work for two reasons. Firstly, there were difficulties determining the orientation of the FOSS readings in EM tracker space. Secondly, given the limited flexibility of the sensor, this method did not allow for the collection of enough points along the surface of the knee model to create an understandable point cloud. And instead generated a point cloud made from lines of constant curvature that could not capture the whole surface or the changes in curvature along model surface that make it humanly recognizable or recognizable for the ICP algorithm used. This can be seen by comparing the point clouds Figures 2.24.b and Figure 2.25.b to that in Figure 2.6, as the point clouds in this experiment are not nearly as distinguishable as those used in 8. Therefore, for FOSS-EM tracking sensor fusion to

be usable for our touch-based registration experiment, the algorithm as well as possibly how the EM tracking coils are used would need to be modified to correct for FOSS orientation as well as position, and the method of how we are collecting the surgical point cloud and target points would also need to be modified so that it can properly collect the irregular surface of the model.

2.9 Overall Conclusions from FOSS Background Work

The FOSS characterization studies and modified registration methods determined which factors contributed to the high FOSS tip position errors observed during the initial registration experiments, proved that using a touch-based registration method that mitigated these factors led to a decrease in TRE values, and provided some early results that suggest supplementing the FOSS system with an EM tracking system could correct for FOSS measurement errors with future work. However, the final FOSS only registration method explored did not decrease RMS TRE enough to meet our submillimetric goal. And the sensor fusion method explored here was unable to produce a surgical point cloud with enough measurement correction or one that covered a sufficient amount of the model surface for successful a successful ICP point cloud match needed for touch-based registration.

As seen in this Outline section, the issue encountered at every step in this work was that how the FOSS system's fiber was measured in FOSS space was just highly inconsistent with how the fiber was positioned, or moved around in physical space. For example, measuring the same position in physical twice or measuring two lines that intersect could yield positions in FOSS space that were over a centimeter away from each other. And during these experiments, this could happen because of something as common as being measured from a different angle of approach or there was some twist on the fiber. Ultimately, because this is an error in the raw data, no ICP point cloud match algorithm can be used to calculate an accurate registration until a proper solution to these issues can be found.

Chapter 3

Chapter III: Manuscript I

Characterization and Investigation into the Accuracy of Touch-Based Registration Using Fiber Optic Shape Sensing

MeiLissa McKay

November 10, 2023

3.1 Abstract

Image guidance is increasingly used by surgeons to locate specific anatomy more accurately during surgery. Unfortunately, many of the sensing systems currently used for image guidance, such as optical or magnetic tracking, have limitations that make them incompatible with certain surgical techniques or medical imaging technologies. Recently, fiber optic shape sensing has emerged as an alternate technology that could overcome these challenges. This work investigates whether a commercially available Fiber Optic Shape Sensor (FOSS) can feasibly be used for image guided surgery via touch-based registration. It first aims to characterize the FOSS by testing the effect that conditions common during touch-based registration have on the sensor's shape and tip position measurements. Lastly, a method of using the FOSS for touch-based registration that aims to mitigate the effects of these conditions is presented and evaluated.

3.2 Introduction

Image guided surgery (IGS) can improve surgical outcome by assisting surgeons in accurately locating surgical instruments relative to patient anatomy continuously during a procedure. To accomplish this, a point cloud of the patient's anatomy must be found by tracing a tracked surgical tool over the relevant anatomy and then registered to a preoperative image i.e. a transformation between the physical patient anatomy and the preoperative image space is found. However, for this, the location and shape of the surgical instrument must be tracked accurately throughout the surgery. And for surgical robotic systems using soft flexible actuators, there is much difficulty in determining a sensing method that is both compatible with their soft flexible nature and is also able to capture actuator shape and location accurately in real time.

Multiple methods of collecting the surface point cloud of the patient's anatomy have been investigated in previous studies. For example, surface points can be sampled by tracing the tip of a surgical robot across the relevant anatomy and estimating the position via a robot's kinematic model [1,2] , or optical sensors such as a laser scanners [3] or an endoscopic camera [4] can be used.

However, these methods are not without limitations. When touching anatomy with a rigid probe and determining its position via a kinematic model, the rigid probe can either cause the surface of soft tissue structures to deflect and thus acquire an incorrect reading or it could damage the tissue being sampled. And in the case of optical tracking systems, a direct line of sight between the camera used for tracking and the object being tracked is always required. Therefore, this method is not compatible with certain surgical applications especially during minimally invasive surgery. Additionally, while electromagnetic (EM) tracking is also widely used for surgical tracking and avoids the issues above, its

readings can be compromised by medical scanners i.e CT or MRI or ferromagnetic surgical instruments [5,6].

Therefore, methods of fiber optic shape sensing (FOSS) have emerged as a possible alternative method of object tracking that does not require rigid contact with the surface being sampled, a direct line of sight, and is not affected by magnetic field distortions. For example, studies have explored the use of FOSS sensors that work from the principle of optical frequency domain reflectometry (OFDR). These have investigated how well these sensors can detect shape and tip position when integrated into soft robotic actuators [7] as well as how they can improve the accuracy of tracking the shape of surgical needles [8].

This paper first investigates the characterization of a commercially available fiber optic shape sensor that operates on the principles of OFDR, and then evaluates the accuracy of a touch-based registration method using the FOSS. For this work, the registration experiments are completed against a 3D printed model of the distal end of the femur to simulate how the the FOSS could be used for image guidance during knee cartilage repair surgery.

3.3 FOSS Characterization Experimental Methods

Several characterization experiments were first carried out to isolate how different conditions common during sensor use would influence the FOSS's measurements. For this work, we use the FOSS pathfinder platform from The Shape Sensing Company (Austin, TX, USA). And during these experiments, only 0.45m of the sensor's total 1.2m length was used in order to increase the accuracy of its measurements. And the sensor's shape data was received at a rate of 17 Hz in MATLAB Simulink (The MathWorks, Inc., Natick, MA, USA).

3.3.1 FOSS Tip Measurement with Manually Added Sensor Twist Experiment

During the manual twist experiments, we evaluated the effect that manually adding twist to the body of the FOSS had on the measured position of the FOSS's tip. The FOSS was secured in the same straight line shape configuration and with its tip held and measured at the same marked position throughout the experiment. The tip of the FOSS was first measured when the sensor was untwisted. Following this, tip position was measured again with positive 45 deg and positive 90 deg of twist applied to the length of the FOSS. This process was repeated five times.

3.3.2 Changing FOSS Configuration Experiment

During the FOSS configuration experiments, we evaluated the effect of changing FOSS configuration on both the measured shape and orientation of the FOSS. A 3D printed testing jig was made to hold the FOSS in two different fixed positions while its 3D shape was measured along its entire sensing length. As shown in figure 1, the testing jig sat flat on the lab bench and had two grooves that follow the paths of identical positive and negative sine waves located on the same plane. Then, during each experimental trial, the FOSS was secured in one of the two grooves and the 3D shape of the sensing length of the FOSS was measured at both of the different configurations. This process was also repeated five times.

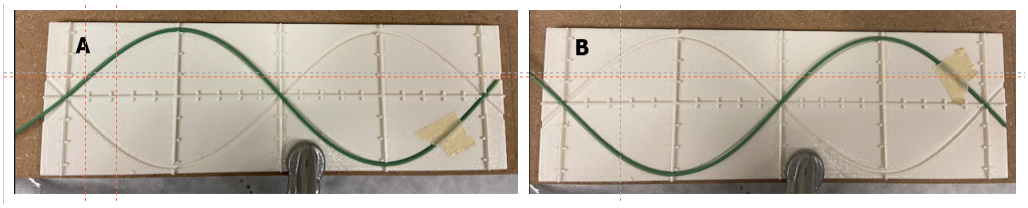

Figure 3.1: Top View of FOSS configuration testing jig with the FOSS in the (A) positive sine and (B) negative sine configurations

3.3.3 FOSS and EM Tracker Sensor Fusion Manual Alignment along Knee Model

During these experiments, we investigated how the configuration in which the FOSS is held influences how its shape and position are measured when stationary and in contact with a 3D printed knee model. Additionally, an EM tracking coil was also fixed perpendicularly to the FOSS at a known location near the tip. It is known that FOSS position error builds down its length away from the origin, so the EM tracker measurements were used to provide a more accurate 3D position at a discrete location towards the tip of the FOSS.

As shown in Figure 2, the FOSS, EM tracking coil, and knee model were held together with the test set up shown. The sensing origin was tightly secured at a 45 deg angle, and the length of the sensor was wrapped around to make contact with the top of a knee model representing the distal end of a femur bone. A 3 cm known length of the FOSS near the tip was then wrapped around the model at different locations within the the medial and lateral femoral condyle regions and the 3D position of the full sensing length was measured at each location. This yielded 10 separate measurements of the combined FOSS and EM tracker readings in each region.

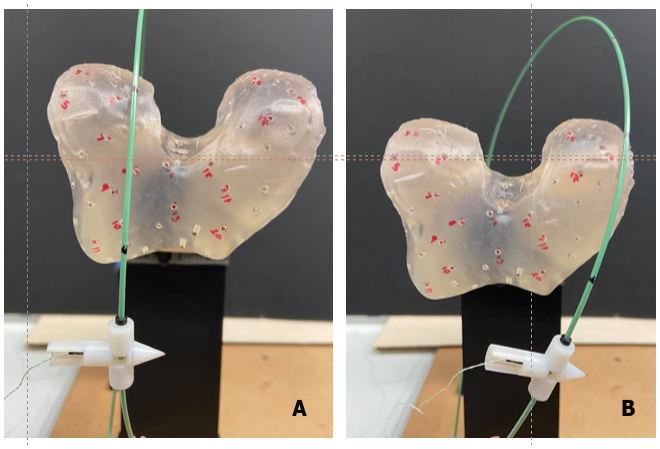

Figure 3.2: Side view of the FOSS in green PEEK tubing and connected EM tracking coil when with a specific region of the FOSS manually held against the (A) medial and (B) lateral femoral condyle regions of the knee model

3.4 FOSS Based Image Guided Surgery Experiments

A method of touch-based registration that aimed to mitigate the negative effects of twist and sensor configuration changes was developed and tested for its accuracy. A FOSS sensing length of 38 cm was used as that was the minimum length necessary to reach all needed regions for registration. These experiments were repeated 120 times in total.

3.4.1 Touch Based Registration

Touch-based registration is used to align two point clouds representing the surface of a patient's knee measured during a preoperative scan and during surgery. Experimentally, the point cloud from a preoperative image is taken as the surface of a 3D printed model of the distal end of the femur as scanned with a FARO ScanArm (Quantum S, FARO Technologies, Lake Mary, FL, USA). And the surgical data is taken from tracing the surface of the model with the tip of the FOSS with the final point cloud taken as the 3D locations of the sensor tip during tracing. Prior to registration, the surgical point cloud is manually aligned via a MATLAB GUI to the preoperative one to provide the registration algorithm with a realistic initial guess for the alignment. The ICP algorithm from MATLAB's Computer Vision Toolbox is then used to register the two point clouds and determine the transformation matrix that aligns the surgical point cloud to the preoperative one.

When collecting the surgical point cloud, the FOSS was held in the configuration shown in figure 3. The point along the FOSS set as the origin was fixed in place at a 45 deg angle and the sensing region of the FOSS was wrapped in a loop backwards from the origin so that the sensor tip was in contact with the knee model surface. And in doing this, the FOSS maintained a similar shape and curvature throughout the surface tracing process used to create the surgical point cloud.

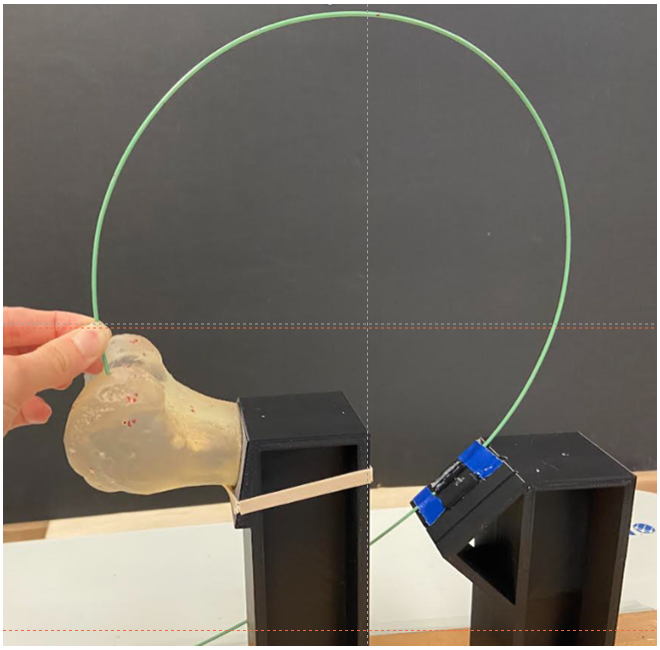

Figure 3.3: Side view of the FOSS and knee model in the angled loop configuration tested during all touch-based registration experimental trials

3.4.2 Measuring the Accuracy of Registration

To evaluate the accuracy of registration, eleven physical targets were added onto the surface of the 3D-printed knee model. Their ground truth 3D locations within the preoperative point cloud were found with the FARO software after scanning the model. And experimentally, their locations within the surgical point cloud were found by touching the tip of the FOSS to each target and recording its 3D location. Targets within the surgical point clouds were then registered to the preoperative cloud using the same transformation matrix found via the MATLAB ICP algorithm initially used to register the two point clouds. The full touch based registration process was repeated 120 times.

3.5 Results

3.5.1 FOSS Tip Measurement with Manually Added Sensor Twist

Figures 4A and B plot the full sensing length and the tip position of the FOSS when under a manually applied twist respectively. During these experiments, the FOSS was measured in the same straight line position for all trials and twist angles. Therefore, the full FOSS shape and tip position should measure as relatively the same for each one. However, Figures 4A and B show that increasing twist consistently caused the FOSS tip position measurement to change. This is further shown in Table 1 which showed that adding 45 deg of twist caused the tip position to change by a mean of 5.68 mm and 9.79 mm for 90 deg of twist,

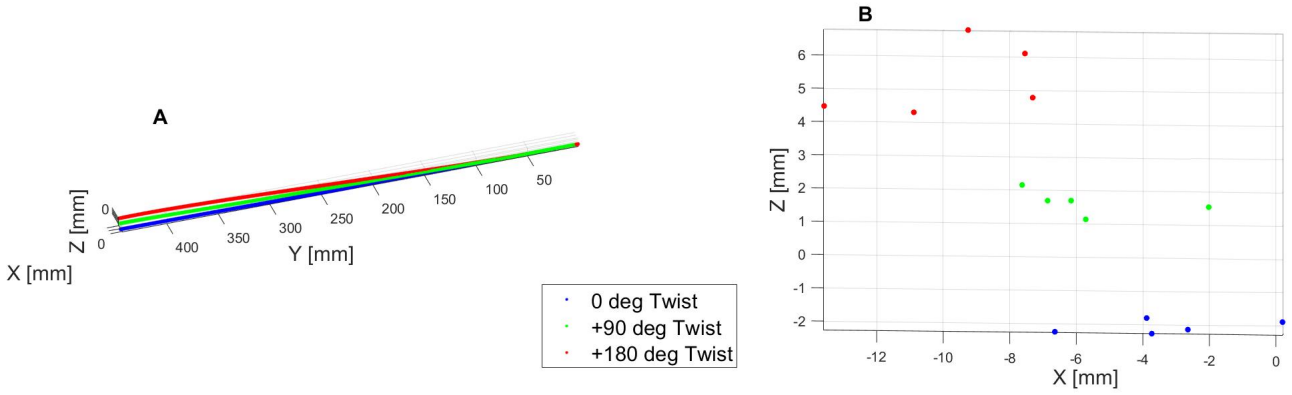

Figure 3.4: (A) Full FOSS data in a straight line configuration and 0,+45, and 90 deg of twist (B) Mean tip position with the FOSS under 0,+45, and 90 deg of twist

FOSS Twist Angle	Mean Error from 0 deg Position [mm]
45 deg	5.68
90 deg	9.79

Table 3.1: Manual Twist Experimental Results

3.5.2 Changing FOSS Configuration

Figure 5 shows how the FOSS measured the 3D shape and position of the positive and negative sine waves on the testing jig. As shown in figure 1, physically the sine waves are located on the same plane. However, an error with how the FOSS measured its' own position caused the sine waves to be measured at an angle to one another.

3.5.3 FOSS and EM Tracker Fusion Manual Alignment along Knee Model

Figure 6 presents the FOSS measurements (A) without and (B) with position correction from the EM tracking coils. These were taken as the FOSS was wrapped around the top surface of the medial and lateral femoral condyle regions of the knee model as seen in figure 2. Without position correction, the medial and lateral regions were incorrectly measured as crossing over each other rather than following along the model's surface next to each other. Additionally, when using the EM tracking coil for position correction, the FOSS readings of the condyle surfaces no longer measure as crossed. However, even

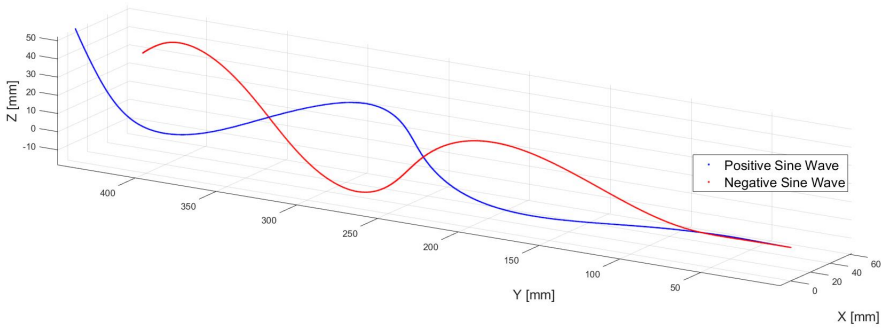

Figure 3.5: Experimental results of measuring the 3D shape and position of the FOSS configured at planar positive and negative sine curves

with the position correction, many of the individual FOSS readings do not follow along one cohesive surface. And the individual measurements lines in the condyle regions of both data sets therefore cannot combine to form a logical point cloud of the surface of the knee model.

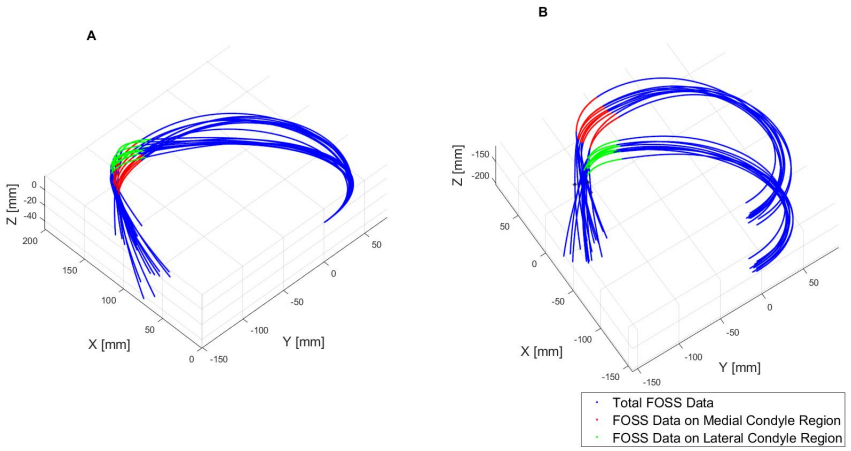

Figure 3.6: Full FOSS measurments with sensing lengths in contact with knee model highlighted (A) without position correction from EM tracker data (B) with position correction from EM tracker data

3.5.4 Touch Based Registration

Accuracy of each registration trial was measured as the root mean square (RMS) of the Target Registration Error (TRE). TRE for each individual target was calculated as the Euclidean distance between the targets' ground truth location within the preoperative point cloud, and their measured location within the registered surgical point cloud. Here, the RMS TRE value reported is the RMS value for

all of the TRE values for each of the eleven targets during every trial. Overall, the 120 RMS TRE values from each of the 120 trials are reported in the histogram in Figure 7.

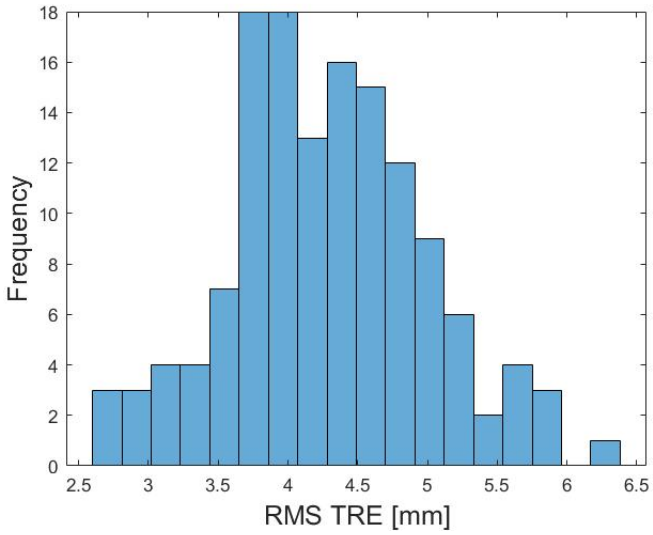

Figure 3.7: Distribution of RMS target registration errors for 120 trials

3.6 Discussion

In this paper, we presented a set of characterization experiments and an evaluation of a touch-based registration image guidance method for a commercially available fiber optic shape sensor. The manually applied sensor twist and sensor configuration experiments showed that the FOSS was sensitive to changes in both conditions. Additionally, adding changes to either condition repeatedly caused incorrect readings in both FOSS position and orientation not consistent with the FOSS's shape and position in physical space. These errors indicate that any successful use of this system for surgical image guidance would need to either reach the surgical site by manipulating the FOSS in ways that cause minimal twist and configuration changes or implement a method of undoing the measurement errors caused by these conditions.

The touch-based registration method presented here attempted to implement a method of using the sensor that reduced FOSS twist and configuration changes enough to avoid measurement errors. This method was visually able to well avoid twisting and changing the configuration of the FOSS. However, the resulting RMS TRE values were higher than the required submillimetric values for successful image registration. This suggests that a method of using the FOSS for image guided surgery needs to actively correct for the errors caused by twist and configuration changes in order to be successful.

The manual knee model alignment experiments also provide insight into the errors encountered during the touch-based registration experiments and how to actively correct for the FOSS measurement

errors. As seen here, the FOSS needed to be bent along an angle to capture data along the lateral region which caused those measurements to incorrectly measure as crossing over the top of the medial condyle in the same direction as that angle used to collect data. Ultimately, a point cloud is a collection of hundreds of measurements like the stationary ones in this experiment. Therefore, since the FOSS needs to be held at multiple different angles when creating a registration point cloud, those like the ones shown in this work are then comprised of points with errors of different degrees and directions leading to the high and unpredictable registration errors seen in this work.

Lastly, the sensor fusion experiments suggest the need for furthering our method of actively correcting for errors from twist and configuration changes. Just as the position correction here was shown to uncross the data taken from the medial and lateral condyle regions, it is possible that furthering this method could better correct for position errors and actively correct for orientation errors as well.

3.7 Conclusion

In this paper, we first presented a characterization study of a commercially available FOSS that tested its sensitivity to twist and changes in configuration. We then presented an evaluation of a method of using the FOSS for touch-based registration surgical image guidance that aimed to reduce FOSS measurement errors from twist and configuration changes. The results of this work indicate that our FOSS system is too prone to error to be used as the sole sensing system for an image guidance application like the one presented in this study, However, there is some promise from our results that sensor fusion with the FOSS can correct for position errors. And future work will be needed to investigate whether sensor fusion with an EM tracker can correct for both position and orientation errors to a significant enough degree that the FOSS can be used for touch-based registration.

3.8 References

[1] J. M. Ferguson, E. B. Pitt, A. A. Remirez, M. A. Siebold, A. Kuntz, N.L.Kavoussi, E.J.Barth, S.D.Herrell, and R. J. Webster, "Toward practical and accurate touch-based image guidance for robotic partial nephrectomy,"IEEE Transactions on Medical Robotics and Bionics, vol. 2, no. 2, pp. 196–205, may 2020.

[2] N. L. Kavoussi, B. Pitt, J. M. Ferguson, J. Granna, A. Remirez, N. Nimmagadda, R. Melnyk, A. Ghazi, E. J. Barth, R. J. Webster, S. D. Herrell, "Accuracy of Touch-Based Registration During Robotic Image-Guided Partial Nephrectomy Before and After Tumor Resection in Validated Phantoms", Journal of Endourology, vol. 35, no. 3, pp. 362–368 , march 2021

[3] A. Raabe, R. Krishnan, R. Wolff, E. Hermann, M. Zimmermann, and V. Seifert, "Laser surface

scanning for patient registration in intracranial image-guided surgery," Neurosurgery, vol. 50, no. 4, pp. 797–803, apr 2002.

[4] D. Burschka, M. Li, M. Ishii, R. H. Taylor, and G. D. Hager, "Scale-invariant registration of monocular endoscopic images to CT-scans for sinus surgery," Medical Image Analysis , vol. 9, no. 5, pp. 413–426, oct 2005

[5] W. Birkfellner, F. Watzinger, F. Wanschitz, R. Ewers and H. Bergmann, "Calibration of tracking systems in a surgical environment," in IEEE Transactions on Medical Imaging, vol. 17, no. 5, pp. 737-742, Oct. 1998

[6] A. M. Franz, T. Haidegger, W. Birkfellner, K. Cleary, T. M. Peters and L. Maier-Hein, "Electromagnetic Tracking in Medicine—A Review of Technology, Validation, and Applications," in IEEE Transactions on Medical Imaging, vol. 33, no. 8, pp. 1702-1725, Aug. 2014

[7] Kevin C. Galloway, Yue Chen, Emily Templeton, Brian Rife, Isuru S. Godage, and Eric J. Barth, "Fiber Optic Shape Sensing for Soft Robotics," Soft Robotics. Oct 2019

[8] Francois Parent, Sebastien Loranger, Koushik Kanti Mandal, Victor Lambin Iezzi, Jerome Lapointe, Jean-Sébastien Boisvert, Mohamed Diaa Baiad, Samuel Kadoury, and Raman Kashyap, "Enhancement of accuracy in shape sensing of surgical needles using optical frequency domain reflectometry in optical fibers," Biomed. Opt. Express 8, 2210-2221 (2017)

Chapter 4

Future Directions

The EM Tracker sensor fusion and registration experiments presented here suggest some immediate directions for improving the accuracy of touch-based registration with the FOSS. For example, the sensor fusion method and experiments presented in this thesis show that this method has potential for correcting position errors in the FOSS measurements. However, further testing is needed to determine how well it can correct for position errors. Additionally, this method should be expanded to account for correcting the FOSS orientation errors caused by twist and sensor configuration changes. One potential method to expand on the method of using two EM tracking coils attached to the FOSS before and after the region in contact with the knee model to also correct for orientation. This would provide for two accurate points of position and orientation along the sensor, and the sensing length in contact with the knee model could be adjusted to the positions and orientations of both. Because, the EM tracker also provides the quaternion values for the orientation of the tracking coil, if the tracking coil was attached parallel to the FOSS rather than perpendicularly like it is in this work, the same EM tracker orientation would match at least a section of the FOSS and could be used to correct for orientation errors.

The registration experiments showed in this thesis are also for touch-based registration over the entire knee model. This was chosen so that our work could focus on reducing error caused by the FOSS system and not on those caused by the ICP registration algorithm. However, the full knee does not need to be accessed during the OATS procedure. If future work focused on only tracing the two regions of the knee that are operated on during the surgery, the the FOSS active sensing length required would likely be shorter than the 35-45 cm length used during the registration experiments here. And since error builds down the length of the sensor, using a shorter sensing length could reduce errors in the FOSS measurements. It should be noted by future groups that it is important to be very careful with the sensing fiber when using shorter lengths. According to emails with The Shape Sensing Company,

the fiber has a 2 cm bending radius. However, from our in-lab experience the fiber feels like it is going to break and has broken at bend radii larger than 2cm especially when working near the fiber's tip. Additionally, the FOSS does hold a lot of tension when using a sensing length of around 25 cm or less.

Lastly, this work focused on mainly the image guidance via fiber optic shape sensing portion of the original plan for the surgical soft robotic system described, And if this area continues to cause difficulties, other areas of the project such as establishing a method of highly accurate position control compatible with soft actuators or developing the design of the soft robotics actuator for the OATS procedures should be explored instead.

For position control, it is unlikely the FOSS system described here could be used on its own unless the soft robotic system moved incredibly slowly. This is because even though the FOSS is currently marketed as having a 60 Hz acquisition rate, it was determined in lab and confirmed by the company that the actual acquisition rate is only around 16 Hz. However, an alternative sensing system such as the EM tracker used in the sensor fusion experiments in this thesis could be used instead.

And while this thesis did not explore any potential actuator designs, this work does give some guidance for what a FOSS compatible soft actuator would look like. It was consistently seen throughout the characterization and registration experiments in this thesis that in order to give the FOSS the best chance of delivering reliable measurements the FOSS had to be guided along paths where it could maintain one similar shape and curvature. And that any sharp bends would cause visibly obvious measurement errors. Therefore, any actuator that can successfully use the FOSS should guide the FOSS along a path that adheres to these parameters such as a C-bend soft actuator that holds the FOSS fiber in the backwards loop path shown in the registration experiments in this thesis.

Chapter 5

Appendices

5.1 ICP Registration Matlab Code

5.1.1 Main Code

```
%% registrationGUI.m
% This script contains a GUI for registration using a fiber-based shape
% sensor. Updated to include multiple initial guesses

% tabula rasa
clear, clc

%% Parameters to set
% specify experiment folder
exrun = 'Registration_38cm_angleback_ortho_11_4_2022';
% specify stl file name
%stlfilename = 'femur_end.stl';

%load Faro Scan
load('Knee_Model_200xDecimate_3-22-2022.mat')

% specify name of the simulink model
modelname = 'TSSC_whole_sensor';
```

```
% specify the name of Simulink output blocks that contain the x, y, and z
% data of the fiber sensor
blocknameXout = 'xdata';
blocknameYout = 'ydata';
blocknameZout = 'zdata';
% specify whether to enable sensor point cloud filtering and downsampling
% of both sensor and stl point cloud
enableFilter = true;
enableDownsampling = true;

%% Load data
% read STL file
%stlData = stlread(stlfilename);
%preopPtCloud = pointCloud(stlData.Points);

%Read in point cloud from Faro Arm
preopPtCloud = pointCloud(Faroscan_200x);



% preprocess preop data
R = eye(3);
tformpreop = rigid3d(R,-[median(preopPtCloud.Location(:,1)), median(preopPtCloud.Location(:,2)), medi
preopPtCloudorigin = pctransform(preopPtCloud,tformpreop);


if enableDownsampling
    preopPtCloudorigin = pcdownsample(preopPtCloudorigin,"random",0.3);
end

% initialize sensor data point cloud (for displaying initial pcshowpair
% plot
sensorPtCloudorigin = pointCloud([0 0 0]);
%% Load Points for donor and damaged regions
```

```
load('GUI_surgery_sites_Registered_6_30_22')
% Donor_PtCloud = pointCloud(Donor_Site);
% Damaged_PtCloud = pointCloud(Damaged_site);
% DonorPtCloudorigin = pctransform(Donor_PtCloud,tformpreop);
% DamagedPtCloudorigin = pctransform(Damaged_PtCloud,tformpreop)

%Color Code by pre-registered region
DonorPtCloudorigin = pointCloud(Donor_Registered);
DamagedPtCloudorigin = pointCloud(Damaged_Registered);




%% Prepare GUI figure, buttons, ...
f = figure;
f.Position = [50 50 1000 600];
%ax = pcshowpair(preopPtCloudorigin, sensorPtCloudorigin);
s1 = scatter3(preopPtCloudorigin.Location(:,1), preopPtCloudorigin.Location(:,2),preopPtCloudorigin.L
ax = gca;
hold on
s3 = scatter3(DonorPtCloudorigin.Location(:,1),DonorPtCloudorigin.Location(:,2),DonorPtCloudorigin.Lo
hold on
s4 = scatter3(DamagedPtCloudorigin.Location(:,1),DamagedPtCloudorigin.Location(:,2),DamagedPtCloudori
hold on
s2 = scatter3(sensorPtCloudorigin.Location(:,1), sensorPtCloudorigin.Location(:,2),sensorPtCloudorigi
axis equal
hold off
ax.Units = 'pixels';
ax.Position(1:2) = [-50 50];
axislimits = [-100 100 -100 100 -100 100]*0.75;
axis(axislimits)

p = uipanel('Title','Registration controls', 'Position',[0.6 0 0.4 1]);
```

```
preg = uipanel('Parent',p,'Title','1. Acquire sensor data for registration','Units','pixels','Positio
b9 = uicontrol('Parent',preg,'Style','pushbutton','String','Load existing data','Position',[5 30 100
b9.Callback = @(es,ed) loadDataCallback(es.Value);

b10 = uicontrol('Parent',preg, 'Style','togglebutton','String','Record new data','Position',[120 30 1
b10.Callback = @(es,ed) recordSensorDataCallback(es);

peuler = uipanel('Parent', p, 'Title','2. Initial guess for rotation: Euler angles','Units','pixels',

% add controls to figure to rotate about euler angles alpha, beta and gamma (ZYZ sequence)
b1 = uicontrol('Parent',peuler,'Style','slider','Position',[80,130,200,15], 'Value',0, 'Min',-180,'Ma
b1t = uicontrol('Parent',peuler,'Style','text','String','alpha','Position',[5, 120, 75,25]);
b2 = uicontrol('Parent',peuler,'Style','slider','Position',[80,80,200,15], 'Value',0, 'Min',-180,'Max
b2t = uicontrol('Parent',peuler,'Style','text','String','beta','Position',[5, 70, 75,25]);
b3 = uicontrol('Parent',peuler,'Style','slider','Position',[80,30,200,15], 'Value',0, 'Min',-180,'Max
b3t = uicontrol('Parent',peuler,'Style','text','String','gamma','Position',[5, 20, 75,25]);

% Callback functions for controls
b1.Callback = @(es,ed) trafoCallback('alpha',es.Value);
b2.Callback = @(es,ed) trafoCallback('beta',es.Value);
b3.Callback = @(es,ed) trafoCallback('gamma',es.Value);

ptrans = uipanel('Parent', p, 'Title','3. Initial guess for translation','Units','pixels','Position',
% add controls to figure to translate along axes
b5 = uicontrol('Parent',ptrans,'Style','slider','Position',[80,130,200,15], 'Value',0, 'Min',-100,'Ma
b5t = uicontrol('Parent',ptrans,'Style','text','String','x','Position',[5, 120, 75,25]);
b6 = uicontrol('Parent',ptrans,'Style','slider','Position',[80,80,200,15], 'Value',0, 'Min',-100,'Max
b6t = uicontrol('Parent',ptrans,'Style','text','String','y','Position',[5, 70, 75,25]);
b7 = uicontrol('Parent',ptrans,'Style','slider','Position',[80,30,200,15], 'Value',0, 'Min',-100,'Max
b7t = uicontrol('Parent',ptrans,'Style','text','String','z','Position',[5, 20, 75,25]);

% Callback functions for controls
b5.Callback = @(es,ed) trafoCallback('x',es.Value);
b6.Callback = @(es,ed) trafoCallback('y',es.Value);
```

```
b7.Callback = @(es,ed) trafoCallback('z',es.Value);

% Control for Registration
b4 = uicontrol('Parent',p,'Style','pushbutton','String','4. Register','Position',[15, 75, 100, 25]);
b4.Callback = @(es,ed) registerPtCloud(es.Value);

% Toggle button to display sensor shape
b8 = uicontrol('Parent',p,'Style','togglebutton','String','5. Show Sensor','Position',[15 30 100 25])
b8.Callback = @(es,ed) displaySensor(es.Value);

%% Global variables and other variables setup
% define global variables as there is no possibility to pass variables by
% reference to callback functions
global eulerangles;
eulerangles = [0 0 0];
global translation;
translation = [0 0 0];
global enableSensorDisplay;
enableSensorDisplay = 0;
global inittform;
inittform = rigid3d(eye(3),[0 0 0]);
global loadData;
loadData = 0;
global recordSensorData;
recordSensorData = 0;
global registerSensorData;
registerSensorData = 0;
global updatePlot;
updatePlot = 0;
global initTrafoChanged;
initTrafoChanged = 0;
recordednewdata = 0;

% hack to enable changing scatter data in the plot with changing size of
```

```
warning('off','MATLAB:handle_graphics:exceptions:SceneNode');

%% Main loop
while true

    %% Set Data for ICP
    if loadData
        % load sensor data mat-file
        load([exrun, '\FOSS5_angleback_11_4_2022.mat']);
        FOSS_mm = FOSS5_angleback_11_4_2022*1000;
        FOSS_PtCloud = pointCloud(FOSS_mm);
        loadData = 0;

        % Data pre processing
        % get rid of initial offsets -> shift to origin
        tformsens = rigid3d(R,-[median(FOSS_PtCloud.Location(:,1)), median(FOSS_PtCloud.Location(:,2)
        sensorPtCloudorigin = pctransform(FOSS_PtCloud,tformsens);

         if enableFilter
            [sensorPtCloudorigin,inliers, outliers] = pcdenoise(sensorPtCloudorigin);
         end

        if enableDownsampling
            sensorPtCloudorigin = pcdownsample(sensorPtCloudorigin,"random",0.2);
        end
%
        if enableFilter
            [sensorPtCloudorigin,inliers, outliers] = pcdenoise(sensorPtCloudorigin);
        end




        % update Plot
        ax.Children(1).XData = sensorPtCloudorigin.Location(:,1);
```

```
    ax.Children(1).YData = sensorPtCloudorigin.Location(:,2);
    ax.Children(1).ZData = sensorPtCloudorigin.Location(:,3);

end
%%

if recordSensorData
    recordednewdata = true;
    sensortipdata = [0 0 0];
end

while recordSensorData
    % here everything to record data
            % IMPORTANT: install Simulink Coder and disable "Signal storage
    % reuse" in the simulink model configuration parameters -> Code
    % Generation -> Optimization
    %
    % The RuntimeObject might return wrong Data if blocks are allowed to share memory
    % see https://ch.mathworks.com/help/simulink/ug/accessing-block-data-during-simulation.html
    xblock = [modelname '/' blocknameXout];
    outputObjectX = get_param(xblock, 'RuntimeObject');
    sensorlivedata.x = outputObjectX.InputPort(1).Data*1000;
    yblock = [modelname '/' blocknameYout];
    outputObjectY = get_param(yblock, 'RuntimeObject');
    sensorlivedata.y = outputObjectY.InputPort(1).Data*1000;
    zblock = [modelname '/' blocknameZout];
    outputObjectZ = get_param(zblock, 'RuntimeObject');
    sensorlivedata.z = outputObjectZ.InputPort(1).Data*1000;
    %nanidx = isnan(sensorlivedata.x);
    [~,tipindex] = max(cumsum(~isnan(sensorlivedata.x)));
    sensortipdata = [sensortipdata; sensorlivedata.x(tipindex) sensorlivedata.y(tipindex) sensorl
    pause(0.01)
end
```

```
if recordednewdata
    sensorPtCloud = pointCloud(sensortipdata);
    % Data pre processing
    % get rid of initial offsets -> shift to origin
    tformsens = rigid3d(R,-[median(sensorPtCloud.Location(:,1)), median(sensorPtCloud.Location(:,
    sensorPtCloudorigin = pctransform(sensorPtCloud,tformsens);

    if enableDownsampling
        sensorPtCloudorigin = pcdownsample(sensorPtCloudorigin,"random",0.2);
    end

    if enableFilter
        [sensorPtCloudorigin,inliers, outliers] = pcdenoise(sensorPtCloudorigin);
    end

    % update Plot
    ax.Children(1).XData = sensorPtCloudorigin.Location(:,1);
    ax.Children(1).YData = sensorPtCloudorigin.Location(:,2);
    ax.Children(1).ZData = sensorPtCloudorigin.Location(:,3);
    recordednewdata = 0;
end

%% Iterate Through Multiple initial guess
%inittform is the initial guess that comes from the GUI sliders

% Perform Transformation on sensor data if sliders were changed
if initTrafoChanged
    sensorPtCloudToShow = pctransform(sensorPtCloudorigin,inittform);
    updatePlot = 1;
    initTrafoChanged = 0;
end

%% Rigid Registration using ICP - update to include multiple initial guesses
if registerSensorData
```

```
%GOICP parameters
perturbations = 200;
angles = deg2rad(5);
lengths = 3;

%TRE parameters
tform_mat = [];
tform_ICP = [];
rmse_vec = [];
TRE_vec = [];
%random angles and lengths for iterations
ran_angle1 = [0;-angles+(angles+angles)*rand(perturbations,1)];
ran_angle2 = [0;-angles+(angles+angles)*rand(perturbations,1)];
ran_angle3 = [0;-angles+(angles+angles)*rand(perturbations,1)];

ran_length1 = [0;-lengths+(lengths+lengths)*rand(perturbations,1)];
ran_length2 = [0;-lengths+(lengths+lengths)*rand(perturbations,1)];
ran_length3 = [0;-lengths+(lengths+lengths)*rand(perturbations,1)];

%Find registration transform and RMS TRE for each perturbation
for i = 1:perturbations
eul_pt = [ran_angle1(i) ran_angle2(i) ran_angle3(i)];
length_pt = [ran_length1(i) ran_length2(i) ran_length3(i)];
rot_pt = eul2rotm(eul_pt);

%define point cloud that's transformed by initial transform as
%defined by sliders in GUI
sensorPtCloudinit = pctransform(sensorPtCloudorigin,inittform);

sensor_pt = sensorPtCloudinit.Location*rot_pt+length_pt;
sensorPtCloud_pt = pointCloud(sensor_pt);

[tform,movingReg,rmse] = pcregistericp(sensorPtCloud_pt,preopPtCloudorigin,'Extrapolate',true
```

```
        regtform = rigid3d(tformsens.T*inittform.T*tform.T);

        tform_mat = [tform_mat;regtform.T]; %transformation including move to origin, initial from GU
        tform_ICP = [tform_ICP;tform.T]; %transform from ICP function

        TRE_RMS = Target_Registration(regtform,exrun,preopPtCloud);
        rmse_vec = [rmse_vec;rmse];
        TRE_vec = [TRE_vec;TRE_RMS];

        end
         rmse_vec = rmse_vec
         TRE_vec = TRE_vec
        %Determine which perturbation has the lowest TRE

        [min_rmse, I] = min(rmse_vec); %minimum TRE and index it refers to
        disp(['Vector Index of Min RMSE',num2str(I)])

        ICP_trans_mat = tform_mat((I*4-3:I*4),(1:4)); %transformation matrix w/ min TRE
        tform_regICP = tform_ICP((I*4-3:I*4),(1:4))
        %tform_regICP_max = tform_ICP((I_max*4-3:I_max*4),(1:4))

        %Find RMS TRE and TRE for all points for transformation with lowest
        %error
        finaltform = rigid3d(ICP_trans_mat)
        [TRE_RMS_mean, TRE_mean_vec] = Target_Registration(finaltform,exrun,preopPtCloud)


%         sensorPtCloudToShow = pctransform(sensorPtCloud,regtform);
        sensorPtCloudToShow = pctransform(FOSS_PtCloud,finaltform);
        %disp(rmse)
        registerSensorData = 0;
        updatePlot = 1;
    end
```

```
    %%
    % update the point cloud plot
    if updatePlot
%         camerapos = ax.CameraPosition;
%         pcshowpair(preopPtCloudorigin,sensorPtCloudToShow)
%         campos(camerapos)
%         axis(axislimits)
        ax.Children(1).XData = sensorPtCloudToShow.Location(:,1);
        ax.Children(1).YData = sensorPtCloudToShow.Location(:,2);
        ax.Children(1).ZData = sensorPtCloudToShow.Location(:,3);
        updatePlot = 0;
    end

    %% Display sensor in plot
    hold(ax,'on')
    if enableSensorDisplay
        % IMPORTANT: install Simulink Coder and disable "Signal storage
        % reuse" in the simulink model configuration parameters -> Code
        % Generation -> Optimization
        %
        % The RuntimeObject might return wrong Data if blocks are allowed to share memory
        % see https://ch.mathworks.com/help/simulink/ug/accessing-block-data-during-simulation.html
        xblock = [modelname '/' blocknameXout];
        outputObjectX = get_param(xblock, 'RuntimeObject');
        sensorlivedata.x = outputObjectX.InputPort(1).Data*1000;
        yblock = [modelname '/' blocknameYout];
        outputObjectY = get_param(yblock, 'RuntimeObject');
        sensorlivedata.y = outputObjectY.InputPort(1).Data*1000;
        zblock = [modelname '/' blocknameZout];
        outputObjectZ = get_param(zblock, 'RuntimeObject');
        sensorlivedata.z = outputObjectZ.InputPort(1).Data*1000;
        validindx = ~isnan(sensorlivedata.x);
```

```
    sensorlivedata.x = sensorlivedata.x(validindx);
    sensorlivedata.y = sensorlivedata.y(validindx);
    sensorlivedata.z = sensorlivedata.z(validindx);

    % Transform sensor data
    livesensorPC = pointCloud([sensorlivedata.x' sensorlivedata.y' sensorlivedata.z']);
    livesensortform = rigid3d(tformsens.T*tform.T);
    livesensorPC = pctransform(livesensorPC,livesensortform);


    if ~exist('ax3','var') % if plot for sensordata doesn't already exist -> create it
        ax3 = plot3(livesensorPC.Location(:,1), livesensorPC.Location(:,2), livesensorPC.Location
    else
        ax3.XData = livesensorPC.Location(:,1);
        ax3.YData = livesensorPC.Location(:,2);
        ax3.ZData = livesensorPC.Location(:,3);
    end
else
    if exist('ax3','var')
        delete(ax3); % clear sensor plot when toggle button is released
        clear('ax3');
    end
end


while enableSensorDisplay
    % here everything to record data
            % IMPORTANT: install Simulink Coder and disable "Signal storage
    % reuse" in the simulink model configuration parameters -> Code
    % Generation -> Optimization
    %
    % The RuntimeObject might return wrong Data if blocks are allowed to share memory
    % see https://ch.mathworks.com/help/simulink/ug/accessing-block-data-during-simulation.html
    xblock = [modelname '/' blocknameXout];
```

```
        outputObjectX = get_param(xblock, 'RuntimeObject');
        sensorlivetipdata.x = outputObjectX.InputPort(1).Data*1000;
        yblock = [modelname '/' blocknameYout];
        outputObjectY = get_param(yblock, 'RuntimeObject');
        sensorlivetipdata.y = outputObjectY.InputPort(1).Data*1000;
        zblock = [modelname '/' blocknameZout];
        outputObjectZ = get_param(zblock, 'RuntimeObject');
        sensorlivetipdata.z = outputObjectZ.InputPort(1).Data*1000;
        %nanidx = isnan(sensorlivedata.x);
        [~,tipindex] = max(cumsum(~isnan(sensorlivetipdata.x)));
        sensortargetdata = [sensortipdata; sensorlivetipdata.x(tipindex) sensorlivetipdata.y(tipindex
        pause(0.01)
    end


    hold(ax,'off')
    % slow down loop
    pause(0.01)

end

function trafoCallback(whichtrafo,trafovalue)
    global initTrafoChanged;
    initTrafoChanged = 1;
    global eulerangles;
    if strcmpi(whichtrafo,'alpha')
        eulerangles(1) = trafovalue*pi/180;
    elseif strcmpi(whichtrafo,'beta')
        eulerangles(2) = trafovalue*pi/180;
    elseif strcmpi(whichtrafo,'gamma')
        eulerangles(3) = trafovalue*pi/180;
    end
    global translation
    if strcmpi(whichtrafo, 'x')
```

```
        translation(1) = trafovalue;
    elseif strcmpi(whichtrafo, 'y')
        translation(2) = trafovalue;
    elseif strcmpi(whichtrafo, 'z')
        translation(3) = trafovalue;
    end
    R = eul2rotm(eulerangles);
    global inittform;
    inittform = rigid3d(R,translation);
end

function registerPtCloud(buttonstate)
    global registerSensorData;
    registerSensorData = buttonstate;
end

function displaySensor(buttonstate)
    global enableSensorDisplay;
    enableSensorDisplay = buttonstate;
end

function loadDataCallback(buttonstate)
    global loadData
    loadData = buttonstate;
end

function recordSensorDataCallback(es)
    global recordSensorData
    buttonstate = es.Value;
    recordSensorData = buttonstate;

    if buttonstate
        es.String = 'Recording...';
    else
```

```
        es.String = 'Record new data';
    end
end
```

5.1.2 Position Based Sensor Fusion with Two EM Tracking Coils

```
clear all
close all
%% Sensor Fusion Code to make knee model surgical point cloud from FOSS and EM Tracker Data

%EM Tracker Located 637 sensors from tip or the the sensor length
%FOSS segment in contact with the Knee Model between
%For Current Data the last sensor is at a position of sensor_length in
%Calibration_chx data

%% Set Variables to Read Translated and Untranslated Position data to for each section of the knee mo
FOSS_model_trans_leftside = [];
FOSS_trans_leftside = [];
FOSS_model_leftside = [];
FOSS_leftside = [];

EM_conn_227 = [];
EM_conn_637 = [];




%% Translate and Save Data for Each Section of the knee model: Left Side, Left Top , Right Side, Righ
%Left is medial region
%Right is lateral region
%% Read and Average Shape Data from FOSS
%sensor_length = 1201; %column position of last sensor in Calibration_chx vectors
for i_ls=0:2
fileX = sprintf('%s_%d_%s.csv','C:\Users\MeiLissa\Documents\Sensor Fusion 7_11_2023\FOSS Fusion 7_11_
fileY = sprintf('%s_%d_%s.csv','C:\Users\MeiLissa\Documents\Sensor Fusion 7_11_2023\FOSS Fusion 7_11_
```

```
fileZ = sprintf('%s_%d_%s.csv','C:\Users\MeiLissa\Documents\Sensor Fusion 7_11_2023\FOSS Fusion 7_11_
Coord1x = load(fileX);
Coord1y = load(fileY);
Coord1z = load(fileZ);

[r,c] = size(Coord1x);

FOSS_avg = [mean(Coord1x(1:end,7:end))'*1000,mean(Coord1y(1:end,7:end))'*1000,mean(Coord1z(1:end,7:en
%scatter3(mean(ch1)',mean(ch2)',mean(ch3)')


FOSS_leftside = [FOSS_leftside;FOSS_avg];

%Remove regions not in contact with knee model
startregion = c-637;
endregion = c-227;

FOSS_model = FOSS_avg(startregion:endregion,1:end);
FOSS_model_leftside = [FOSS_model_leftside;FOSS_model];

%scatter3(FOSS_model(1:end,1),FOSS_model(1:end,2),FOSS_model(1:end,3));

%% Read and Average Magnetic Tracking Position Data
%% Load Position and Quaternion Data from Magnetic Tracker
 j_ls = i_ls;

fileEM = sprintf('%s%d.csv','C:\Users\Meilissa\Documents\Sensor Fusion 7_11_2023\EM Sensor Fusion_7_1
EM_data = readtable(fileEM);

[r2,c2] = size(EM_data);

%Determine averaged position of EM tracking coil nearer to the tip of the FOSS

EM_227_x = [];
```

```
EM_227_y = [];
EM_227_z = [];
Quat1_227_0 = [];
Quat1_227_x = [];
Quat1_227_y = [];
Quat1_227_z = [];

%Weed out points designated as bad fit points
for k_ls = 1:r2
    state = EM_data{k_ls,20};
    if strcmp('OK',state)
        EM_227_x = [EM_227_x;EM_data{k_ls,25}];
        EM_227_y = [EM_227_y;EM_data{k_ls,26}];
        EM_227_z = [EM_227_z;EM_data{k_ls,27}];
        Quat1_227_0 = [Quat1_227_0;EM_data(k_ls,21)];
        Quat1_227_x = [Quat1_227_x;EM_data(k_ls,22)];
        Quat1_227_y = [Quat1_227_y;EM_data(k_ls,23)];
        Quat1_227_z = [Quat1_227_z;EM_data(k_ls,24)];
    end

end

EM_227_avg = mean([EM_227_x, EM_227_y, EM_227_z]);
Quat_227_avg = mean([table2array(Quat1_227_0), table2array(Quat1_227_x), table2array(Quat1_227_y), ta

% Transform Measured EM Position to a point located on FOSS

EM_227_position = [0,0,4.5]'; %distance in z from position of EM tracker point to connected location

quat_227 = quaternion(Quat_227_avg(1),Quat_227_avg(2),Quat_227_avg(3),Quat_227_avg(4));
Rot_227 = rotmat(quat_227,'point');
xyz_227_adjust = Rot_227*EM_227_position;
EM_227_connect = xyz_227_adjust'+EM_227_avg;
EM_conn_227 = [EM_conn_227;EM_227_connect];
```

```
%Determine averaged position of EM tracking coil farther from the tip of
%the FOSS (637 sensor counts from tip)

EM_637_x = [];
EM_637_y = [];
EM_637_z = [];
Quat1_637_0 = [];
Quat1_637_x = [];
Quat1_637_y = [];
Quat1_637_z = [];

%Weed out points designated as bad fit points
for k_ls = 1:r2
    state = EM_data{k_ls,6};
    if strcmp('OK',state)
        EM_637_x = [EM_637_x;EM_data{k_ls,11}];
        EM_637_y = [EM_637_y;EM_data{k_ls,12}];
        EM_637_z = [EM_637_z;EM_data{k_ls,13}];
        Quat1_637_0 = [Quat1_637_0;EM_data(k_ls,7)];
        Quat1_637_x = [Quat1_637_x;EM_data(k_ls,8)];
        Quat1_637_y = [Quat1_637_y;EM_data(k_ls,9)];
        Quat1_637_z = [Quat1_637_z;EM_data(k_ls,10)];
    end

end

EM_637_avg = mean([EM_637_x, EM_637_y, EM_637_z]);
Quat_637_avg = mean([table2array(Quat1_637_0), table2array(Quat1_637_x), table2array(Quat1_637_y), ta

% Transform Measured EM Position to a point located on FOSS


EM_637_position = [0,0,6.5]'; %distance in z from position of EM tracker point to connected location
```

```
quat_637 = quaternion(Quat_637_avg(1),Quat_637_avg(2),Quat_637_avg(3),Quat_637_avg(4));
Rot_637 = rotmat(quat_637,'point');
xyz_637_adjust = Rot_637*EM_637_position;
EM_637_connect = xyz_637_adjust'+EM_637_avg;

EM_conn_637 = [EM_conn_637;EM_637_connect];

%% Transformations for each sensor count between EM tracker 1 and 2

%Find change in x,y,z position applied to each sensor count between the two EM
%tracking coils


connect_pos = c-227;
EM_FOSS_translation = EM_227_connect-FOSS_model(1, 1:3);


EM_diff = EM_637_connect-(EM_FOSS_translation+FOSS_model(end, 1:3));
EM_diff_ct = EM_diff/410;


%Transform data in contact with knee model
FOSS_model_trans = zeros(411,3);
for i =1:411
    FOSS_model_trans(i,1:3) = FOSS_model(i,1:3)+EM_FOSS_translation+(EM_diff_ct*(i-1));
    x = (EM_diff_ct*(i-1))
end


%FOSS_trans = FOSS_avg+EM_FOSS_translation+EM_diff_ct;


FOSS_model_trans_leftside = [FOSS_model_trans_leftside;FOSS_model_trans(157:302,1:end)];
FOSS_trans_leftside = [FOSS_trans_leftside;FOSS_model_trans];
```

```
%%
end


%% Load and Translate Data from Left Top Region of Knee Model

%Read and Average Shape Data from FOSS
%sensor_length = 1201; %column position of last sensor in Calibration_chx vectors
for i_lp=9:11
fileX = sprintf('%s_%d_%s.csv','C:\Users\MeiLissa\Documents\Sensor Fusion 7_11_2023\FOSS Fusion 7_11_
fileY = sprintf('%s_%d_%s.csv','C:\Users\MeiLissa\Documents\Sensor Fusion 7_11_2023\FOSS Fusion 7_11_
fileZ = sprintf('%s_%d_%s.csv','C:\Users\MeiLissa\Documents\Sensor Fusion 7_11_2023\FOSS Fusion 7_11_
Coord1x = load(fileX);
Coord1y = load(fileY);
Coord1z = load(fileZ);

[r,c] = size(Coord1x);

FOSS_avg = [mean(Coord1x(1:end,7:end))'*1000,mean(Coord1y(1:end,7:end))'*1000,mean(Coord1z(1:end,7:en
%scatter3(mean(ch1)',mean(ch2)',mean(ch3)')

FOSS_LeftTop = [FOSS_LeftTop;FOSS_avg];

%Remove regions not in contact with knee model
startregion = c-637;
endregion = c-227;

FOSS_model = FOSS_avg(startregion:endregion,1:end);
FOSS_model_LeftTop = [FOSS_model_LeftTop;FOSS_model];
%scatter3(FOSS_model(1:end,1),FOSS_model(1:end,2),FOSS_model(1:end,3));
```

```
% Read and Average Magnetic Tracking Position Data
% Load Position and Quaternion Data from Magnetic Tracker
j_lp = i_lp-9;

fileEM = sprintf('%s%d.csv','C:\Users\Meilissa\Documents\Sensor Fusion 7_11_2023\EM Sensor Fusion_7_1
EM_data = readtable(fileEM);

[r2,c2] = size(EM_data);
EM_227_x = [];
EM_227_y = [];
EM_227_z = [];
Quat1_227_0 = [];
Quat1_227_x = [];
Quat1_227_y = [];
Quat1_227_z = [];

%Weed out points designated as bad fit points
for k_lp = 1:r2
    state = EM_data{k_lp,20};
    if strcmp('OK',state)
        EM_227_x = [EM_227_x;EM_data{k_lp,25}];
        EM_227_y = [EM_227_y;EM_data{k_lp,26}];
        EM_227_z = [EM_227_z;EM_data{k_lp,27}];
        Quat1_227_0 = [Quat1_227_0;EM_data(k_lp,21)];
        Quat1_227_x = [Quat1_227_x;EM_data(k_lp,22)];
        Quat1_227_y = [Quat1_227_y;EM_data(k_lp,23)];
        Quat1_227_z = [Quat1_227_z;EM_data(k_lp,24)];
    end

end

EM_227_avg = mean([EM_227_x, EM_227_y, EM_227_z]);
Quat_227_avg = mean([table2array(Quat1_227_0), table2array(Quat1_227_x), table2array(Quat1_227_y), ta
```

```
% Transform Measured EM Position to a point located on FOSS


EM_227_position = [0,0,4.5]'; %distance in z from position of EM tracker point to connected location


quat_227 = quaternion(Quat_227_avg(1),Quat_227_avg(2),Quat_227_avg(3),Quat_227_avg(4));
Rot_227 = rotmat(quat_227,'point');
xyz_227_adjust = Rot_227*EM_227_position;
EM_227_connect = xyz_227_adjust'+EM_227_avg;

EM_conn_227 = [EM_conn_227;EM_227_connect];

%Determine averaged position of EM tracking coil farther from the tip of
%the FOSS (637 sensor counts from tip)

EM_637_x = [];
EM_637_y = [];
EM_637_z = [];
Quat1_637_0 = [];
Quat1_637_x = [];
Quat1_637_y = [];
Quat1_637_z = [];

%Weed out points designated as bad fit points
for k_ls = 1:r2
    state = EM_data{k_ls,6};
    if strcmp('OK',state)
        EM_637_x = [EM_637_x;EM_data{k_ls,11}];
        EM_637_y = [EM_637_y;EM_data{k_ls,12}];
        EM_637_z = [EM_637_z;EM_data{k_ls,13}];
        Quat1_637_0 = [Quat1_637_0;EM_data(k_ls,7)];
        Quat1_637_x = [Quat1_637_x;EM_data(k_ls,8)];
        Quat1_637_y = [Quat1_637_y;EM_data(k_ls,9)];
        Quat1_637_z = [Quat1_637_z;EM_data(k_ls,10)];
```

```
    end

end

EM_637_avg = mean([EM_637_x, EM_637_y, EM_637_z]);
Quat_637_avg = mean([table2array(Quat1_637_0), table2array(Quat1_637_x), table2array(Quat1_637_y), ta

% Transform Measured EM Position to a point located on FOSS


EM_637_position = [0,0,6.5]'; %distance in z from position of EM tracker point to connected location

quat_637 = quaternion(Quat_637_avg(1),Quat_637_avg(2),Quat_637_avg(3),Quat_637_avg(4));
Rot_637 = rotmat(quat_637,'point');
xyz_637_adjust = Rot_637*EM_637_position;
EM_637_connect = xyz_637_adjust'+EM_637_avg;
EM_conn_637 = [EM_conn_637;EM_637_connect];
%% Transformations for each sensor count between EM tracker 1 and 2

%Find change in x,y,z position applied to each sensor count between the two EM
%tracking coils
connect_pos = c-227;
EM_FOSS_translation = EM_227_connect-FOSS_model(1, 1:3);


EM_diff = EM_637_connect-(EM_FOSS_translation+FOSS_model(end, 1:3));
EM_diff_ct = EM_diff/410;


%Transform data in contact with knee model
FOSS_model_trans = zeros(411,3);
for i =1:411
    FOSS_model_trans(i,1:3) = FOSS_model(i,1:3)+EM_FOSS_translation+(EM_diff_ct*(i-1));
    x = (EM_diff_ct*(i-1))
```

```
end



FOSS_model_trans_LeftTop = [FOSS_model_trans_LeftTop;FOSS_model_trans(157:302,1:end)];
FOSS_trans_LeftTop = [FOSS_trans_LeftTop;FOSS_model_trans];



%%
end
```

5.2 Solidworks Parts

5.2.1 EM Coil Connection Fixture to FOSS during Sensor Fusion

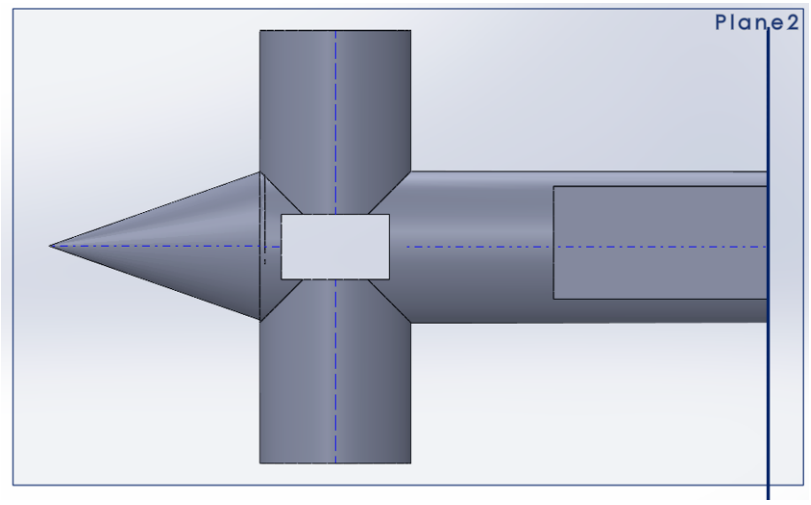

Figure 5.1: Fixture used to connect the EM tracking coils to the FOSS

5.2.2 Backwards Loop and Sensor Fusion Registration Fixture

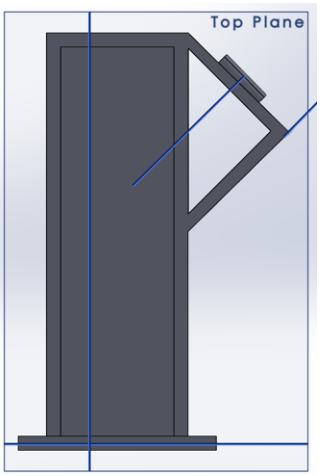

Figure 5.2: Fixture used to hold the origin of the FOSS at a 45 deg angle