Simulation and Modelling of Tissue Deformation for Laparoscopic Surgeries

Alice Maria Catalano

DIBRIS - Department of Computer Science, Bioengineering,
Robotics and System Engineering

University of Genova

Supervisors:
Prof. Fulvio Mastrogiovanni
Prof. Alessandro Carfì

In partial fulfillment of the requirements for the degree of

Laurea Magistrale in Robotics Engineering

October 18, 2023

Declaration of Originality

I, Alice Maria Catalano, hereby declare that this thesis is my own work and all sources of information and ideas have been acknowledged appropriately. This work has not been submitted for any other degree or academic qualification. I understand that any act of plagiarism, reproduction, or use of the whole or any part of this thesis without proper acknowledgment may result in severe academic penalties.

Acknowledgements

This acknowledgments section is just a small gesture to extend my gratitude to the people who were next to me during these two years and in the thesis work period, so to everyone who is reading this page I want to say thank you.

I extend my deepest gratitude to Professor Fulvio Mastrogiovanni, whose expertise, understanding, and teachings added knowledge to my graduate experience but also pushed me toward academic achievement that I was not expecting to direct on, as the Ph.D. pursuit.

Special thanks go to Dott. Alessandro Carfi, who will always be a referential figure for my future academic and work life. Thanks to his dedication, patience, and critical thinking I was always encouraged to do more and go through the next steps, keeping my feet on the ground and teaching me to divide problems into smaller issues with a certain solution.

To TheEngineRoom group and the Emaro lab that hosted me goes my gratitude, in particular to Valerio Belcamino and Simone Macciò, who were there in moments of frustration and always helped in moments of crisis, with precious advice or a simple laugh.

Between the people that were next to me as friends and companions I say thank you to Hussein, my first and dearest friend, The efforts and the challenges that you had to face through these years are not explainable, but you always found time to support and take care of your friends. I am grateful to consider myself one of your friends and I hope that through the years that will come, you will feel the same with me.

I want to thank Youssef whose help was pivotal in every one of my steps in these two years and my academic efforts. Your cold analysis and straightforward solutions finding a tool that you effortlessly exhibit and I try to grasp. I will make good treasure of all the advice you gave me, hoping that in some way my presence in your life brought an improvement as you did in mine.

A huge thank you and deepest gratitude goes to Giulia and Letizia. I can't even start to say the treasure I found in you girls, the support and the sparkle between us is something I cherish with all my heart. Sharing the residence with you is one of my best luck.

A thank you that extends for all the two years of this degree goes to Piya and Angelica. The physical distance between us couldn't anyway break our bond. The moments we shared were periods of joy, in which everything seemed light and fine. I see in you girls a safe place and I hope it will stay like this for a long time.

In the last piece of this acknowledgment, I want to thank those people who have and always had a special place in my heart, starting from my family.

I thank my mother and my father as they are the propellers for every one of my achievements. These two years have been tough though you were always next to me. For my brother, who I found next to me in moments of loss and confusion. I faced challenges that I would never expect to happen and yet you were there, with your advice and with your hugs that heated those cold periods in which everything seemed lost. I love you from the bottom of my heart.

To my best friends, Barbara, Nicol, and Pierfrancesco, whom I try to thank every time I get the chance and I am grateful that in my life I will always have these three people to rely on. You are the piece of stability and reliability that until now I haven't found in my life.

Thank you to everyone that is not clearly cited in these pages, but that I know are the ones that most deserve my gratitude.

I dedicate this work to my family. In a world full of challenges, the only place where nothing seems hard is with you.

Abstract

This research endeavors to address the pivotal challenges inherent in somatic sensing and organ localization during teleoperated surgical procedures, specifically in devices lacking integrated haptic feedback, such as the da Vinci® Surgical System. A perception model, aimed at acquiring and visually replicating the organ's surface deformations during surgery, is proposed. It employs Inertial Measurement Units (IMUs) anchored to the kidney's peripheral surface, intending to guide surgeons through partial nephrectomy by providing reliable displacement data during manipulations, augmented with the device's stereovision. The articulated problem-solving approach encompasses sensing and modeling organ movements and subsequently implementing the model in a simulated realistic scenario.

The deformation model is realized and conveyed to the surgeon using Digital Twin technology, enabling real-time visual representation of the organ's biomechanical interactions with surgical manipulators. The long-term vision for this approach anticipates extending these displacement patterns to reveal internal structures of the organ, presenting surgeons with critical internal vessel details. The preliminary experiments utilize a dataset derived from an elastic 3D kidney phantom to uphold ethical and research integrity, serving as a ground truth for analysis. It's noteworthy that the da Vinci® robot, inspiring this research and commercially operational, cannot be modified and thus organ displacement is detected solely via surface observations, proposing an effective measurement approach through direct IMU implementation on the renal capsule.

Contents

List of Figures

Chapter 1

Introduction

In the realm of surgical procedures, the integration of diverse subjects to achieve new techniques has continuously been paramount to enhancing precision, reducing invasive measures, and improving patient outcomes. There proofs that these integration have been an important step forward in the healthcare field are: robotics, sensing technology, and simulation software have not only transformed surgical practice but has also elevated the capabilities of training, research, and patient communication in this domain. This thesis aims to delve deep into this intersection, focusing specifically on the possibility of creating a Digital Twin of a specific organ, created through open-source platforms. The main objective is to optimize the perception of organ movements, during the surgeon manipulations through surgical robots that don't allow haptic feedback in the control loop of the device. To implement such optimization the choice of sensors to use wanted to escape the usual cameras and image processing algorithms. Inertial Measurement Units were chosen for this reason and other more technology-specific advantages will emerge during the thesis.

The rise of surgical robotics heralded a new era of minimally invasive procedures, yet it also introduced complexities that demand intricate planning, precise control, and an in-depth understanding of both the human anatomy and the robotic tools. Modern surgical practices now rely on creating accurate digital replicas of anatomical structures to allow surgeons to visualize, plan, and even rehearse complex procedures. The bridge from the physical to the digital realm, particularly the process of creating such replicas, presents its challenges and opportunities, an area this research seeks to explore in-depth, as it is already happening with algorithms that are based on the usual behavior of surgeons and living organs develop a pattern recognition and event prediction. The innovative side of this research stands on trying not to rely on these tools, but to provide real-time data of the behavior of the interested organ and the anatomical parts involved in the surgical procedure.

While several proprietary solutions exist, the emphasis of this work is on the role of IMU sensors in the perspective of creating stable and reliable real-time sensing of the deformations of the organ. The pivotal aim is to capture real-world data and translate it into the digital realm.

The experimentation involves two distinct models: the intricate kidney model, representing the complexity of real-world anatomical structures, and a polyurethane cube model serving as a simplified geometric reference. Both models, when juxtaposed, offer insights into the challenges of mesh density, force field representations, and the complications arising from the transition between different modeling techniques within SOFA, the open-source framework chosen as a representation tool. Additionally, this work seeks to address the challenge of creating a model with mixed properties, particularly starting from a deformable model, creating rigid planes that would represent the position of the IMUs and through these rigid planes create a deformation mirroring the real-world deformation, based on the IMUs orientation solely. A task that SOFA presents as a complex endeavor.

This thesis, structured into multiple sections, begins with a comprehensive overview of the state of the art in surgical robotics, sensing, modeling, and the bridging of physical to digital paradigms. Following this, a detailed exploration of the materials and methods used, with a focus on the experimental protocol, the intricacies of the SOFA framework, and the role of the IMU sensor, is presented. The subsequent sections unveil the results and discussion, shedding light on the various phases of experimentation and the insights garnered. Finally, the work concludes by projecting future work and overarching conclusions.

With this research, the intention is not only to contribute to the body of knowledge on surgical simulations but also to pave the way for further exploration in this domain, especially regarding the creation of Digital Twins using open-source platforms. The journey is one of challenges, learning, and continuous evolution, and it is hoped that this thesis will serve as both a foundation and an inspiration for further exploration in this exciting intersection of medicine and technology.

Chapter 2

State of the art

2.1 Surgical Robotics

The employment of Robot-Assisted Surgery (RAS) is becoming increasingly relevant in multiple and complex surgical procedures, establishing its potential during the years almost reversing the curability of untreatable diseases and decreasing risks and consequences of peculiar surgical operations [Zhao *et al.* (2021)].

In the RAS systems category, the Robotic Minimally Invasive Surgery (RMIS) systems acquired the keen interest of surgeons and researchers. It's a technology that was adopted in hospitals in the early 2000s, right after the beginning of the related research. One of the most outstanding examples is the daVinci® Surgical System (Intuitive Surgical Inc., Sunnyvale) figure 2.1, which pioneered this new sector of research and market and almost monopolized it for a long time. The biggest drawback of this technology is the lack of "somatic sensory feedback", as Enayati *et al.* (2016) defines the natural sensing of both force and touch, a deficiency that was already perceived in Minimally Invasive Surgery (MIS) and magnified in RMIS due to the teleoperation set up and the consequent physical distance between the surgeon and the operating instruments represented by the robotic arms [Joice *et al.* (1998)]. In MIS surgeons still hold in their hands the laparoscopic instruments, having the possibility to gain a higher perception feedback. Due to the teleoperation setup of RMIS and the total external control of the surgical tools employed by the robotic arms, the perception of the surgical environment is totally absent. For this reason, it must be noted that surgeons must endure intensive training to develop and master the technical competencies necessary to maneuver the device and control movements' coordination [Kühnapfel *et al.* (2000)]. Another challenge for surgeons is preserving the spatial perception during the teleoperation of the RMIS on soft and moving organs. For instance, in the urology field, where the da Vinci system has had the most significant impact, organs are small and pliable with a homogeneous surface.

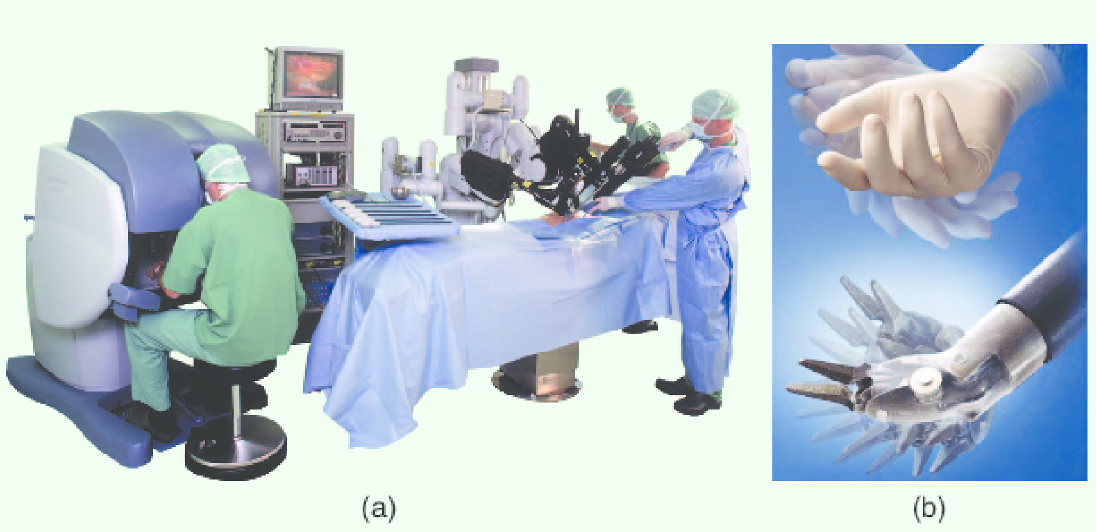


Figure 2.1: (a) The da Vinci surgical system consists of a master console and teleoperated patient-side robot. (b) Dexterous instruments enable fine manipulation inside the body. Okamura *et al.* (2010)

For these reasons, teleoperation maneuvering without proper sensation feedback will always represent an obstacle to overcome for surgeons. The possibility to smooth down these obstacles is still there thanks to the extensive research aiming to build up the human-computer interaction and improve the cooperation between surgeon, machine, and patient. An ideal solution for such problems would be the possibility to have a real-time dynamic model of organs, which can be used in simulation with training objectives and as real-time guidance to the surgeon during the operation through Augmented Reality (AR) support [Nakao & Minato (2010)].

Modeling the organ is a fundamental step to address the issues pointed out. Having an organ modeled means having the possibility to analyze the organ in a virtual environment and having a full 3D view of the specific case improves time and efficiency of the surgical procedure. The objective is to add a step forward to the model, synchronizing it with the real organ inside the patient, allowing pharmaceutical tests, surgery training and if the real-time component is added, to have a clear 3D view of the organ movements and changes during any surgical operation, so that the surgeon's point of view is not limited from the dark and not always a clear view of the RMIS camera.

In the panorama of surgical robotics, aside from RMIS, several other techniques and technologies have evolved, reflecting the vast potential and dynamic growth of this field. One example is microsurgical procedures which require a level of precision and stability that is often challenging for human hands alone, especially during prolonged operations where even the slightest tremor can compromise results. In constrained surgical sites, such as the eye in ophthalmic surgeries, this precision is crucial. The robot's arms can manipulate instruments at scales and with a steadiness that a human hand might struggle to achieve.

Two exemplary platforms in this domain are RAVEN Li *et al.* (2019) and MiroSurge Hagn *et al.* (2010). Developed by the University of Washington, the RAVEN system, figure 2.2, is designed with multiple arms and has found applications in telesurgery, where surgical procedures are performed remotely. On the other hand, MiroSurge, figure 2.3 a product of DLR - German Aerospace Center, is designed for delicate manipulations. Its modularity allows for the integration of various instruments, making it suitable for a wide range of procedures.

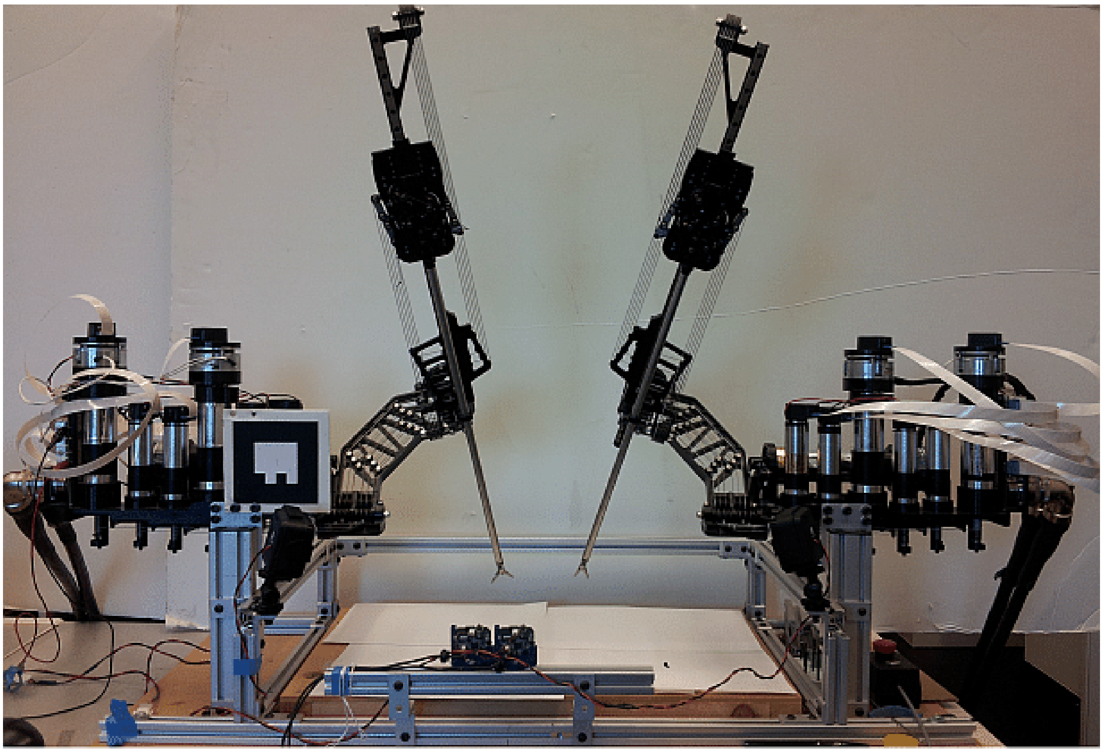


Figure 2.2: (a) The raven II surgical robot is the target system for the cable research by Miyasaka *et al.* (2015)

All these technologies support the affirmation that surgical robotics and RAS systems have become increasingly powerful technology that took its space in healthcare proving its potential directly in the field of application. In this section, a summary of the literature findings that prove how surgical technology can be helpful will be presented. In table 2.1 some of the major key points are organized. It's important to notice that these benefits are explored comparing with both

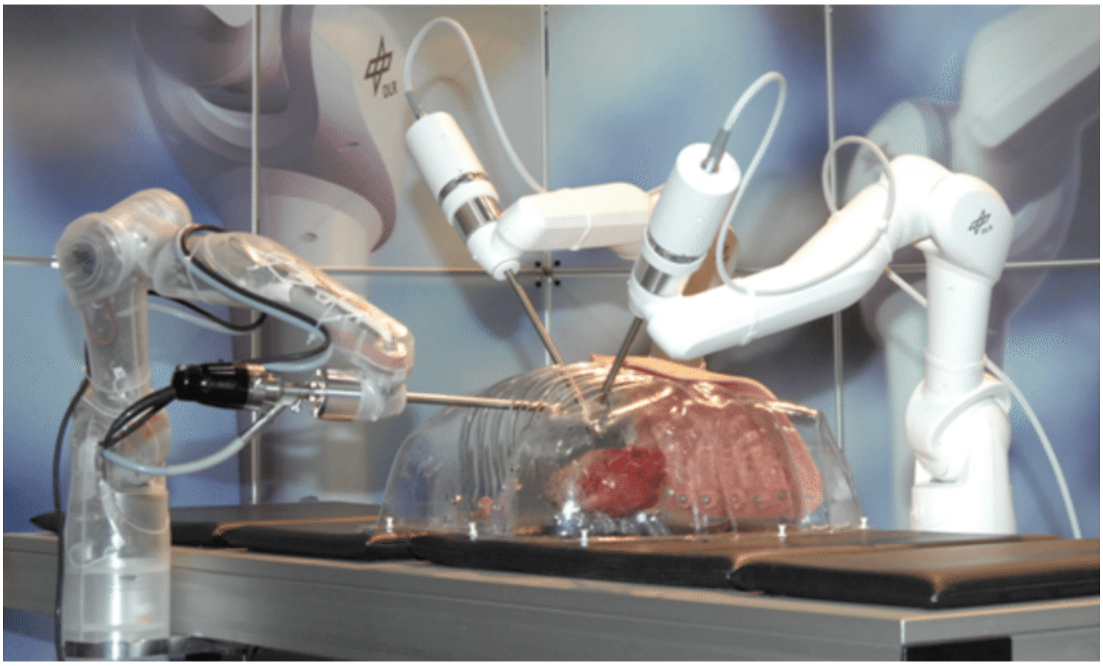


Figure 2.3: MiroSurge system developed at DLR with table-mounted manipulators. Figure by Yip & Das (2017)

laparoscopic and open surgery techniques and some of these advancements are achieved solely thanks to the new technologies of RAS i.e. the conjunction on near-infrared fluorescence (NIRF) and indocyanine green dye (ICG), that showed to reduce the probability of risks by 10.6% in the reconstruction of the upper and lower urinary tract.

Table 2.1: Key points of RAS benefits

Key points	Reference code
• Robotics improves visual accuracy, beyond usual laparoscopy • Possibility to conjunct NIRF and ICG dye.	Zhao *et al.* (2021)
• Reduction of post-operative, surgical and medical complications	Vittori (2014)
• 7 DoF dexterity versus 4 DoF of usual laparoscopic technique • Tremor filtration and 3D visualization	Hughes *et al.* (2023)

The RAS technologies offer a major dexterity in visceral anatomical regions, endorsing the RMIS field. This characteristic has prompted comparisons with traditional open surgery, in which RAS outperforms conventional approaches across multiple domains as Hughes *et al.* (2023) documented. Additionally, thanks to the less-invasive nature of the the RMIS procedure, it infers recovery benefits for patients, conveying a smoother and faster healing, avoiding the serious complications an open wound might arise.

In the analysis proposed by Vittori (2014) an interesting comparison is made between open and robotics partial nephrectomy (respectively OPN and RPN), exploiting also a summary on the advancement over the classic laparoscopic approach in which the LPN (Laparoscopic Partial Nephrectomy) results more difficult and less reproducible because of technical complications [Benway *et al.* (2009); Khalifeh *et al.* (2013); Pierorazio *et al.* (2011)].

Another interesting point of this analysis is the differentiation between post-operative complications (RPN=8.6%, OPN=24.2%), and medical complications which infers the arising of other medical events like *urinary fistula* (RPN=2.9%, OPN=3%), and surgical complications (RPN=5.7%, OPN=21.2%). From the statistical data gathered it is noticeable how the RPN score never approaches the two integer digits.

Although sensory feedback is a significant issue, RMIS systems like da Vinci® Surgical System, do not support haptic feedback because it would introduce instabilities in the closed loop control process, and in a field like laparoscopic surgery is not acceptable to have even the slightest degree of unsafety [Freschi *et al.* (2013); Gettman *et al.* (2004); Yates *et al.* (2011)]. For this reason, despite the obvious weight of the haptic feedback, the system was never integrated with any force or movement sensors. Instead, alternative methods to provide the force feedback to the surgeon were explored.

2.2 Sensing and modeling the organ and its movements

One of the alternative methods to improve the surgeon's sensing during the operation is the modeling of the organ's deformations, which aims to recreate a visual real-time or online projection of the organ's mechanical properties while interacting with the robotic manipulators. However, gathering data to model the biomechanical properties of an organ represents significant challenges due to the case-specific anatomy of each patient, and creating a general model might be not efficient. Furthermore, the model shall be applied in vivo, so in a real-time scenario with the patient involved, increasing the sensibility of the task [Amparore *et al.* (2022)].

The literature reports different strategies to depict organs' motion, with a particular emphasis on motion compensation during spontaneous breathing, frequently with the goal of enhancing the effectiveness of radiotherapy in cancer treatment [Hostettler *et al.* (2010); Strbac *et al.* (2015)]. To address the issue of capturing organ motion during surgical procedures the use of medical imaging has been the primary approach.

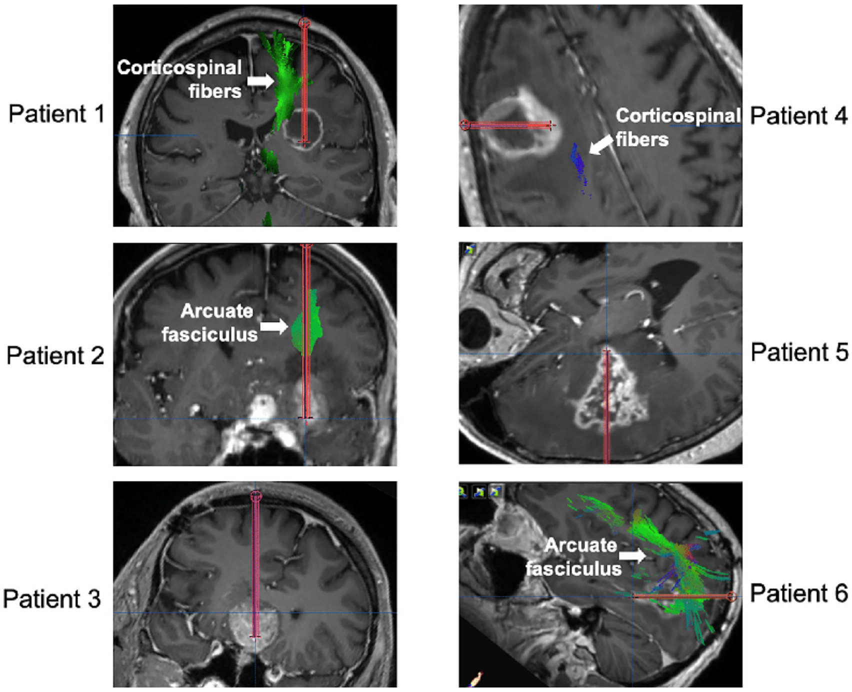


Figure 2.4: Preoperative imaging and planned trajectories, it shows the tumor targets and planned trajectories for each patient included in the study by Hajtovic *et al.* (2020)

Preoperative imaging (Figure 2.4) is utilized to plan the operation strategy, assess the safest path to access the area of interest, and determine the method of action i.e. in a tumor excision as the partial nephrectomy. This offline technique relies heavily on the surgeon's experience and does not involve any technological assistance after the imaging process.

In certain cases, intra-operative imaging can also be used to guide the surgeon to the tumor's location, understand the organ's movements during the procedure, and provide a view of its internal structures based on the imaging method. With the development of Virtual Reality (VR) and Augmented Reality (AR) technologies, organs can be modeled from bioimages, then coupled with the standard information of biomechanical and physical characteristics of the interested organ and transferred to a virtual environment, enabling surgeons to practice a specific procedure on them or see the anatomical structures in during surgery, as shown in figure 2.5. This possible setup increases the demand for a functional model

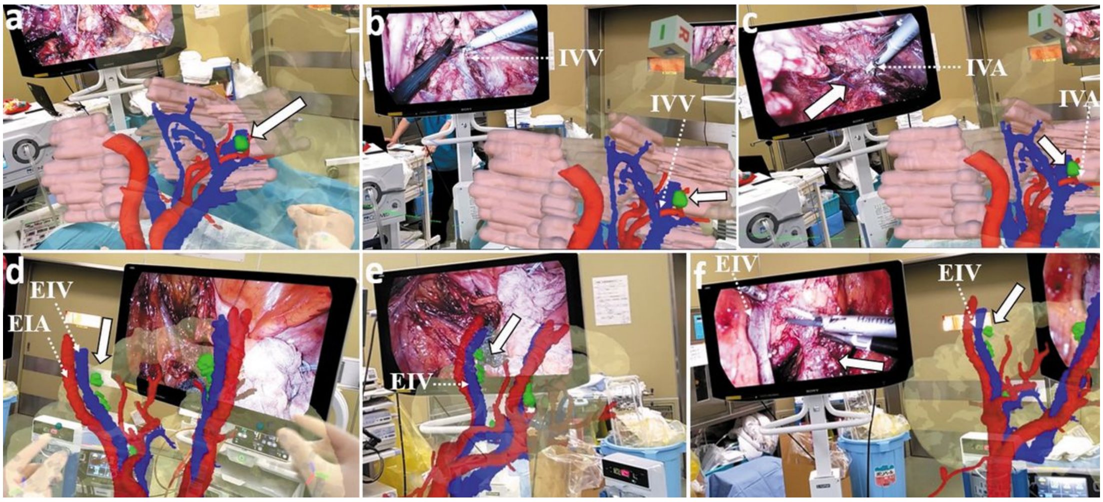


Figure 2.5: Lateral lymph node dissection. a) HoloLens2 allowed the surgeons to intraoperatively view the 3D model of veins and arteries as if they were floating in the air in the operating field. The white line indicates lymph node metastasis around the inferior vesicle artery (IVA). b) Dissection around the inferior vesicle vein (IVV). c) Dissection around the IVA. d) The white line indicates lymph node metastasis around the external iliac vein (EIV). e) The surgeon can manipulate the image, such as magnifying and changing the angle, while maintaining sterility. f) Dissection around the EIV. EIA: External iliac artery. Study by RYU *et al.* (2022)

with the subsequential possibilities that it can offer opening the research to a variety of performance options: starting from simple observation of the 3D model in various orientations, to manipulation of the virtual organ, to the possibility for the surgeon to practice their skillset through simulation with a maximum realistic response to movement like the real organ.

In this scenario though, the movements happening on the organ are not synchronized with the model. To implement this characteristic there is the necessity to perceive the movements through sensors. The stereo camera used in RMIS systems 2.6 is known not to have good visibility, especially in case of bleeding and suturing by electrocauterization, a technique that uses heat generated by an electric current to coagulate blood vessels and seal tissues and produces smoke. A small and non-intrusive sensor with the ability to recognize motion and changes of orientation would be a good option, and IMUs sensors have these abilities.

The Inertial Measurement Units (IMU) sensors are used for the purpose of detecting object motion and orientation and are a particularly adaptable sensor due to their compact size and ease of measuring position and orientation in all directions. IMUs have diverse applications, ranging from localization of aircraft

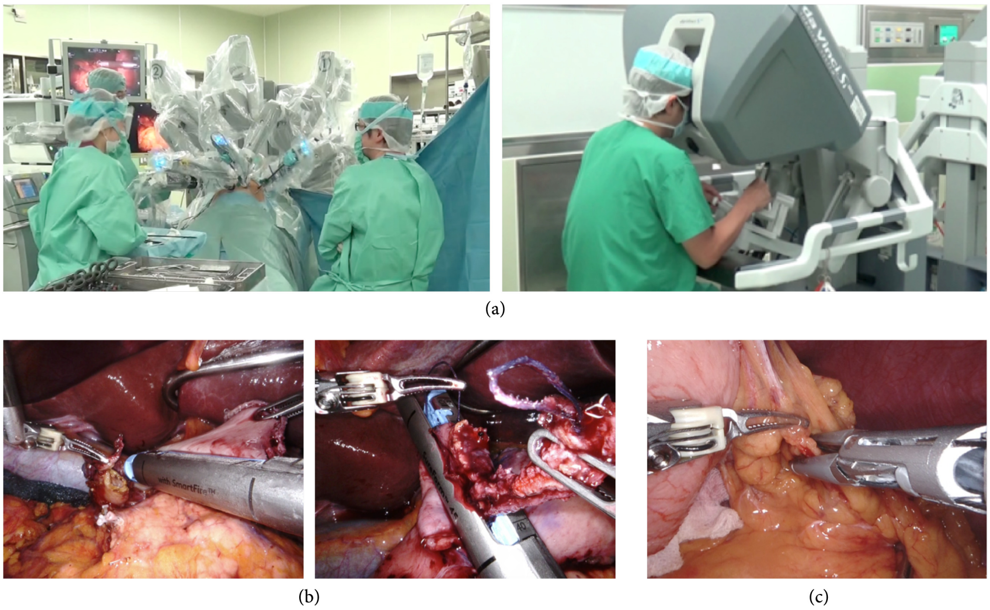


Figure 2.6: (a) The da Vinci Xi system; (b) the developed stapler for the da Vinci system; (c) the developed energy devices for the da Vinci system. Study by Ohuchida (2020)

and Automated Unmanned Vehicles (AUV), measuring deformation and bending of underground pipelines [Ma & Wangr (2022)] to the angular deformation of ships' decks and hulls [Zhang (2022)]. In addition, these small sensors are often used to measure human motion and posture (gait, falls, etc.), breathing or articulation movements [Bernhart *et al.* (2023); Pesenti *et al.* (2023)].

The IMU is a reliable sensor, that typically consists of three key components: accelerometers, gyroscopes, and magnetometers. They make use of the laws of motion and the known initial conditions to continuously update the position, angular velocity, and orientation of an object. However, over time, errors can accumulate due to inherent sensor limitations, such as drift and noise, but is possible to apply algorithms to reduce the error accumulations. The advantages of these devices are the contained dimensions, affordability, ability to measure all movement properties, and compatibility with other sensors or microcontrollers for communication or enhanced functionality purposes. These sensors were not usually used as motion tracking of organs, if not in the case of Abayazid *et al.* (2018) in which an IMU attached to the hub of a reference needle inserted in a liver had the aim to measure the free breathing movements of the organ to train a learning-based approach to estimate its position.

The necessity of a realistic model that accurately represents both the outlook and biomechanical behavior is critical. Through literature, various methods have been developed that evolved the concept of preoperative imaging integrating it with machine learning algorithms to predict displacements based on surgeons' actions. This technique yields a comprehensive model developed entirely offline, with no exact correlation to real scenarios and organ behavior, as data is primarily gathered from phantoms and animal experiments, such as swine within the ethical limits. In some cases, intraoperative imaging is added to the loop as the ground truth of deformation inferred by the algorithm [Heizmann *et al.* (2010)]. However, limitations still exist, including an increase in operation time, potential risk to both the patient and medical staff with imaging methods, and the need for a broad dataset to create a reliable algorithm capable of predicting the organ's behavior with any surgeon manipulation.

On the other hand, in order to initiate the development of deformable models, it was essential to categorize various cases according to anatomies and surgical techniques. Meier *et al.* (2005) offer a concise and accurate description of a potential classification of the deformable models that were accessible at the time. They provide a thorough overview summarized in table 2.2 that covers heuristic approaches, which originated from deformable splines [Terzopoulos *et al.* (1987)], and progressed to tensor-mass models, which represent a volumetric expansion of the mass-spring model and take into account the non-linearity of elasticity [Picinbono *et al.* (2000)]. Additionally, the continuum mechanical approaches, which are characterized by high computational weight, are explored such as Fast Finite Elements (FFE) and boundary element methods (BEM).

The Mass-spring method (MSM) is a popular physics-based modeling method that allows the connection of further points in the model through elastic interactions. In the context of the intended application, the MSM can provide an accurate solution, especially when integrated with an IMU for data sharing. This is significant because the algorithm and the model can utilize real-time values to refine the calculations and improve precision. Additionally, the MSM can effectively handle tissue cuts using the state machine developed by Bielser *et al.* (2004).

Free-form deformation methods have been widely employed in surgical simulations, despite their divergence from accurately reproducing physical deformations. Each deformation involves the manipulation of control points on a lattice that virtually contains the modeled organ, which has been subdivided into hyperpatches. These hyperpatches are then moved by the control points to achieve the desired deformation. However, this technique has the drawback of a higher risk of 3D aliasing, which reduces the likelihood of achieving the desired deformation, and due to the global calculation method used, this technique is unsuitable for local deformations [Cover *et al.* (1993)].

When confronted with a challenge such as this, it is imperative to identify the most suitable modeling approach that aligns with the objective requirements.

Table 2.2: Modeling methods, Pros and Cons

Model Class	Modeling Method	Pros	Cons
Heuristic models	**Deformable splines**: lines that adjust themselves to a series of control points.	• Realistic deformations • High control over shapes	• Parameter selection • Compatibility with rendering algorithms • High computational cost
	Spring-mass model: spring meshes distributed over the object's surface connected to mass points.	• Simple mesh structure • High realism • Easy representation of manipulations	• Limited volumetric behaviour • Oscillations probabilities • Limited physical realism • Arbitrary parameters selection
	Tensor-mass model: linear relationships to represent the deformation behavior of the tetrahedral meshes.	• Good volumetric behaviour • Respect incompressibility • Empirical parametrization • Resolution-dependent	• Complex mesh creation and handling • Computational cost • Limitation in physical realism
Continuum-mechanics-based models	**Finite elements method**: solves the equations of a soft model by inverting the stiffness matrix	• Speed • Accurate compliance	• Limited physical realism • Static mesh, fixed nodes • Not applicable to general surgery simulation
	Boundary elements method: numerical techniques used to approximate and solve differential equations by focusing on the boundary of a domain.	• Efficient realistic surface • Reduced dimensions • Robust and quick • Real-time adaptability	• Complex mathematical description • Fully populated matrices • Laborious computation

For the analyzed case, mass-spring models may prove to be a practical solution due to the limitations of the area of interest.

2.3 Synchronization model to physical object

Another essential aspect of the proposed problem is the methodology employed to transmit the acquired data as valuable information to the surgeon through the RMIS procedure. Viewing the healthcare industry as an engineering discipline enables researchers to adapt technologies originally designed for other applications and assess their applicability in the medical domain. This paradigm shift has occurred with robotics, virtual reality, and other related technologies, and is now being witnessed with the utilization of the Digital Twin (DT) tool.

A DT represents a virtual replica of a physical object, generated based on all the available data on its physical counterpart, with the objective of testing strategies and solutions whose outcomes are difficult to predict. Furthermore, the DT serves as an observation instrument that, when coupled with the appropriate set of sensors for the Physical Twin (PT), can exhibit all changes in the PT's status, thereby learning from it and providing accurate predictions while ensuring that the data accurately represents the past and present processes [Erol *et al.* (2020)].

DT are highly intricate yet effective tools. They embody dynamic environments and enable their study without altering the PT, with which the virtual model is synchronized. The incorporation of AI into the DT will expose hidden correlations and infer new conditions for the PT, which can then be manipulated in a more efficient and secure manner. Given this definition, it is evident that this tool is genuinely intriguing and applicable across a broad spectrum of fields, such as manufacturing and aviation. Moreover, the medical domain has also begun leveraging this remarkable tool for hospital management and personalized medicine applications. Regarding the healthcare discipline, the DT model was initially employed for predictive maintenance of medical equipment or to optimize hospital infrastructure and its life cycle, as these applications were comparatively simpler to achieve and implement [Barricelli *et al.* (2019)]. As the DT model necessitates continuous data from the PT, it is challenging to conceive an effective DT for the entire patient's body. Consequently, existing DT models adopted for clinical practice focus solely on individual organs, such as the Living Heart, developed by Dassault Systèmes (2015) and released in 2015. This was one of the first organ DT models to consider all heart functions.

In 2022, the ARAILIS study case by Riedel *et al.* (2022) endeavored to create a DT of the liver to provide surgeons with better guidance during MIS. The study highlighted the difficulties in mapping the organ due to its deformations and distortions during surgery, using only the stereo laparoscopic camera as the sensor and the SLAM algorithm for pattern recognition.

Taking the DT approach, approach into consideration, it would be a worthwhile option to employ this technology as a solution for the problem proposed, which is analogous to the ARAILIS case study. In the ideal scenario of this research, the IMUs would be utilized to detect the deformation and distortion of the organ during surgery, rather than relying on a stereocamera as a sensor. The communication between the inertial sensors and the virtual model, established with the aforementioned techniques, would provide the surgeon with an extensive quantity of data and information to facilitate better orientation during the RMIS with the da Vinci robot.

2.4 Software and frameworks for surgical simulations

In approaching the proposed solution with the digital twin model, selecting the appropriate modeling method and framework to control and host the DT while synchronizing it with the PT becomes crucial. However, this decision is far from easy, given the numerous limitations inherent in each technology. It is often suggested in the literature that starting from scratch to design a framework tailored to the simulation's intent may be more advantageous, as it allows for a better alignment with the surgical training simulation objectives, detached from the real-time communication complexities of the digital twin [Maciel *et al.* (2011)].

While the literature analysis has provided valuable insights into the state of surgical training simulations, there's a noticeable gap in studies focused on real-time communication between the physical and virtual worlds. Given the time limitations of this thesis, the focus narrows to examining two distinct frameworks for their potential adaptability to digital twin requirements: Unreal Engine 5.0[1] (Epic Games, Inc., North Carolina, USA) and the Simulation Open Framework Architecture[2]. By narrowing the scope to these two frameworks, the thesis aims to explore their potential to facilitate real-time communication and synchronization between the Digital Twin and the Physical Twin, contributing to a more effective and efficient simulation.

The literature is still exploring the performance of gameplay engines on this type of project because of the computational weight and the basic need of every gameplay software to be able to process at high speeds and quality transmission. Although UE5 offers impressive graphics and a wide range of tools for creating re-

[1] https://www.unrealengine.com/en-US/unreal-engine-5

[2] https://www.sofa-framework.org/

alistic scenarios, like the new Machine Learning Deformer [Unreal Engine (2023)], yet is not easy to implement it on any type of hardware device, and for simulation of interactions in real-time it is still on development. The interest though is high, and the research is proposing this direction highlighting the integrability of this framework with the surgical simulation[El-Wajeh *et al.* (2022)]

SOFA framework instead, is a certified simulator of these types of scenarios, providing different plugins and basic scenes from which it is possible to start easily a development project. In particular, there is the possibility to interface it through plugins to different scenarios and tools. SOFA is an open-source C++ library but it integrates a plugin to manage Python scripts, giving the possibility to write more dynamic and simple codes and allowing the framework to interface with ROS for various applications like soft robots or haptic feedback robots for simulation training as Geomagic touch. SOFA framework has a detailed and specific structure for scenes and code building and communication, which will be explained further in the dedicated section 3.2.

In conclusion, it was not possible at the time of this research to find in the state of the art any specific comparison between these two tools, deriving from the recent growing interest of UE team toward the medical field, which seems to be driven more by researchers than by the developers of the platform itself.

Chapter 3

Materials and Methods

3.1 Experimental protocol

Before exploiting the practical work, it was essential to develop an experimental protocol to guide the process and ensure that all necessary elements for the experimental procedure were covered. This protocol includes data acquisition, simulation assessment, and the establishment of communication to transform the model into a Digital Twin. To thoroughly analyze the situation, a separate examination of both physical and digital tools was conducted. However, the integration of these elements will only take place during the experimental phase, where they will come together to create the Digital Twin of the organ.

Given the considerations presented in section 2.4, the SOFA framework was selected as tool to construct the Digital Twin. This decision was based on several factors that outweigh the software's disadvantages, the main factor that led to this choice is that SOFA offers an intuitive setup, easy access to programming the model, and the use of a simple, well-known language like Python. In section 3.2, the characteristics and tools utilized in the model will be detailed and explained.

Simultaneously, the research for suitable materials to develop the Physical Twin of the organ commenced. Initially, various types of smart materials with specific characteristics were analyzed. However, after the study, it was decided to utilize a simple sample of polyurethane with specifications detailed in table 3.2. This sample would serve solely to initiate the tuning of the simulation and uncover potential flaws that may emerge in the realistic behavior of the model when applied to real-world situations.

The final consideration pertains to the selection of the appropriate sensor or sensors for the task at hand. During the extensive literature review, the possibility of utilizing one or multiple Inertial Measurement Unit (IMU) sensors was thoroughly explored. The examination of existing successful implementations of various sensors highlighted the opportunity to innovate within the domain.

Consequently, the decision was made to incorporate the MPU-9250 sensor by TDK InvenSense, a choice guided by a desire to introduce a novel technique within the field. Comprehensive details about the specifications and rationale behind the selection of this particular sensor are elucidated in section 3.4.

3.2 SOFA framework

When it comes to the creation of the model that will be used as Digital Twin, the structure of Sofa's software architecture is the main tool that is used, and the one that needs to be understood as deeply as possible to get the results needed.

3.2.1 Software Architecture

The software's structure is intuitively organized in the form of a Directed Acyclic Graph (DAG), a foundational concept within the SOFA framework. This can be thought of as a generalized hierarchy, ensuring that there are no cycles, and every node has a direction pointing to the next node. This hierarchical tree-like structure serves as the cornerstone for real-time simulations, particularly for deformable objects.

The simulation itself is called *scene* and it initializes the graph structure with the main root called *node* which is the entry point of the simulation and organizes the various elements, such as objects, forces, constraints, and visual models. Everything environmental for the scene gets to be referenced to the root node, such as the plugins needed for the scene, the animation loop with the animation time-step, the gravity action, and some of the collision tools needed for all the objects existing in the scene that will appear with the model loading tools.

Every child node will then inherit these characteristics, but when the child is created new rules for that specific object can be applied independently from the other objects. This level of modular flexibility allows for a variety of mechanical and visual models to be plugged together, forming complex simulations. When a child node is defined, the first tools inserted for the description of its behavior will be the *solver*. Firstly the integration scheme is needed to find approximate solutions to the ordinary differential equation in the system, and then a solver will be applied to this set of solutions. Only after defining these ground rules is it possible to apply a mechanical rule to the model imported into the environment, along with the topology distribution. This is where the component structure comes into play, the *component* represents a particular attribute such as physical properties, geometries, or constraints.

The object can have in addition its visualization method and collision handling ruling, ensuring a separation between the mechanical and visual aspects of the simulation. The use of force fields, constraints, and other components further augments the object's characteristics, tailoring its behavior within the scene. In figure 3.1, a usual structure of a graph is shown, while in the next section, the specific graph and elements used for the Digital Twin kidney model tailored for this research will be explained. The dynamic interaction between these components, combined with SOFA's ability to interface with various standard tools and libraries, makes it a versatile and powerful system for simulating complex mechanical systems.

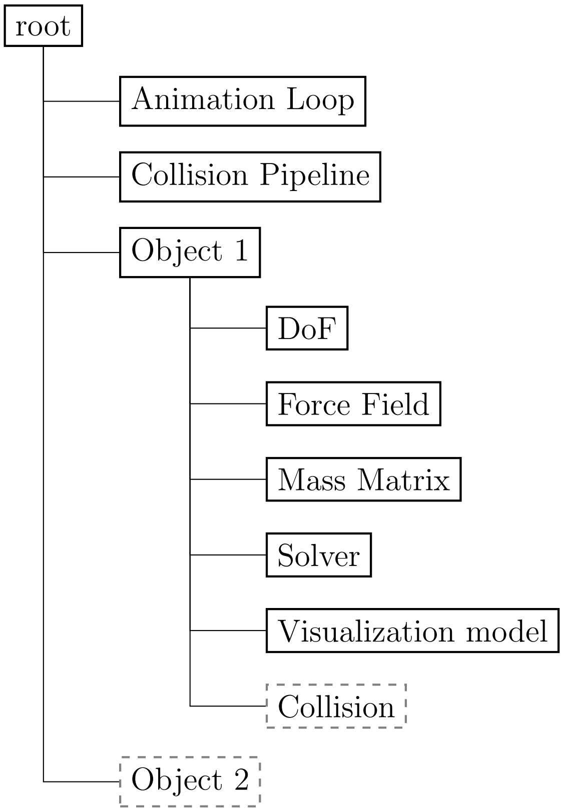


Figure 3.1: General DAG structure for any SOFA simulation. The scene should be constructed following this order and fulfilling all the elements with continuous borders, while the elements with dashed borders are optional, i.e. the Collision for the object is needed only if another Object is present in the scene.

3.2.2 Tools and communication

In this section, the tools employed in the simulation and the outline of their interconnections are detailed. This is presented before diving into the model's description, ensuring that readers or users are familiar with the components before observing their utilization.

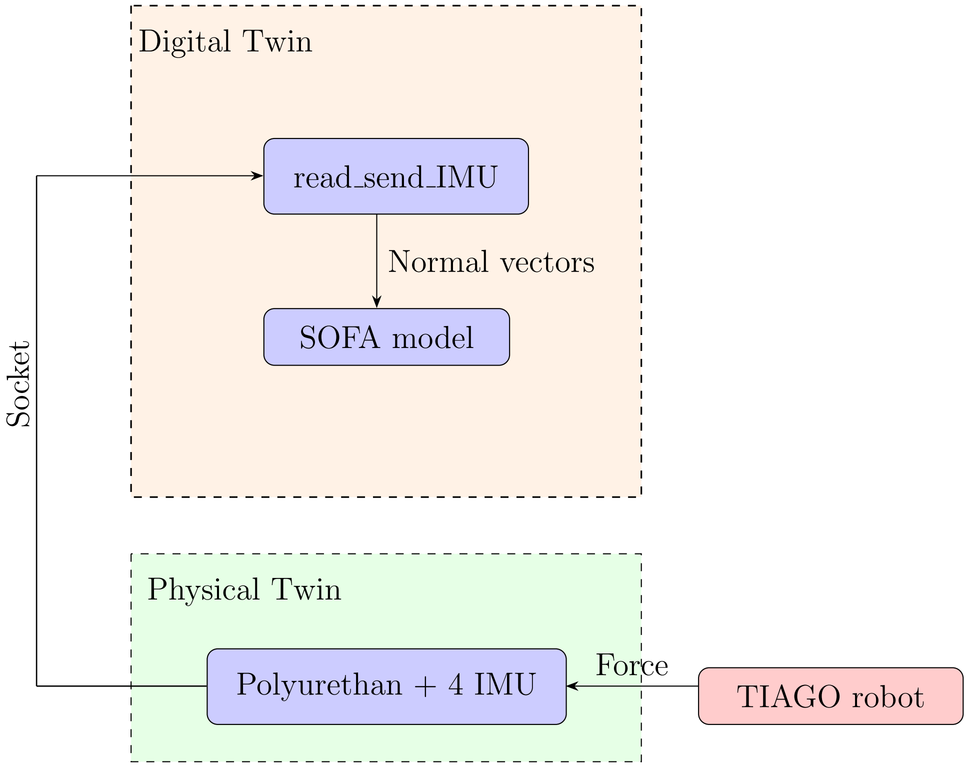


Figure 3.2: Communication between PT and DT. In the scenario, there is a total of three modules: the robot is just a tool to measure the imposed force as an assessment, the PT is a passive object that communicates through the IMUs measurements with the DT which is a software module that unpacks the UDP messages and applies them to the simulation with thanks to SOFA control protocols

The deformations are facilitated by the TIAGo robot PAL Robotics (2018) using a direct programming approach. The robot is equipped with a force sensor, allowing it to serve as a reliable reference for the model. This ensures a quantitative comparison of how closely the simulated movements align with real-world actions. In lieu of an actual kidney, deformations are applied to a polyurethane specimen. Its simpler geometry offers a more precise test environment, supporting the application of assumptions made for the simulation. This approach minimizes discrepancies between the simulated and real-world contexts.

On the polyurethane sample, four IMUs are mounted along the boundary of a specified circumference (dimensions to be detailed later). At its center, the TIAGo robot applies forces of varied magnitudes and angles, producing corresponding deformations. These efforts test the integrity of the model, which initially mirrors the geometry of the polyurethane piece, but later evolves into a specifically designed kidney model.

In Figure 3.2, the most critical component is the *read_send_IMU*. This represents the code crafted to decode the IMU messages transmitted via socket. The resulting message contains the ID marker for each IMU and the quaternion values indicating its orientation. The code then processes these values, interpreting them as normal vectors for the IMU's plane. These vectors are then applied to the constrained regions in the model that correspond to the IMUs, adjusting their orientation to match the IMUs on the PT.

The deformation control would then follow this logic: since the IMUs represent a plane, on the model mesh some planes with the same distances would be chosen and controlled by their normal vector. The deformation would then be approximated by calculating the intersection point of the defined planes.

3.2.3 Model

The experimentation made use of two distinct models: the intricate kidney model and the polyurethane cube model which served as a simplified geometric reference.

Initial project expectations envisioned a linear and structured workflow. However, complexities arose, primarily due to the challenges in sourcing reliable information on constructing mixed material objects. Additionally, the absence of a prototype or reference scenario tailored for this unique application further compounded the difficulties. Owing to these challenges, coupled with time constraints, the final models and the simulation did not fully align with the set project requirements, necessitating their reevaluation in future work. This section offers a detailed analysis of the current state of these models, highlighting identified issues and potential solutions.

The kidney model, as depicted in figures 3.3 and 3.4, showcases two snapshots from the simulation's initial setup, juxtaposing the model against the simulation's global framework. A discerning observation reveals hollow sections within the kidney model. These voids can be attributed to the initial design which incorporated two tumoral masses, along with their associated vascular structures. Given that these elements were superfluous to the project's current focus, they were duly excised from both the scene and the model.

One can also observe the dense meshwork characterizing the model, evident from the figures. This mesh is densely packed with tetrahedral forms that underpin the *VisualObject*. The construction of this model employed the Spring-Mass Model. However, the versatility provided by SOFA ensures that transitioning between this model and a FEM requires only a minimal code modification. But this simplicity in code transition contrasts starkly with the difference in the resulting behaviors. This outcome stems from the intricate process of adapting an existing model to a new context, with challenges arising, especially in the seamless integration of the desired deformation logic.

The mesh file's density posed an initial challenge due to the substantial computational demands associated with its intricacy. To make the model feasible for real-time simulations, it was imperative to reduce this density, at the cost of some precision.

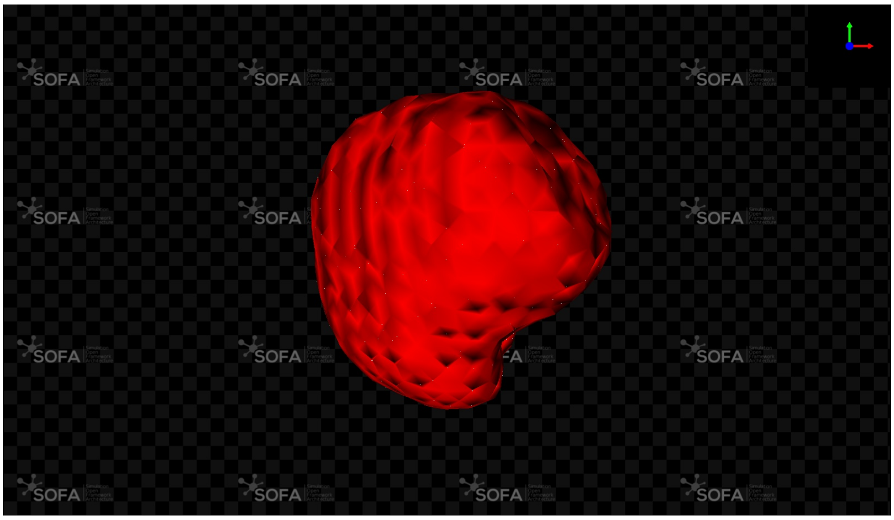
Another complication stemmed from the mesh model's non-uniform surface. The irregularities and non-planar nature of the mesh faces, which often deviated from the defined frame directions, underscored the necessity for a more rudimentary geometric model. This was especially pertinent considering the IMUs required precise plane definitions to operate accurately, as each sensor possesses its unique reference frame, as illustrated in figure 3.11. Retrofitting these sensors onto a non-standard frame was neither straightforward nor efficient.

Despite considerable efforts to adapt the aforementioned intricate model, the focus eventually shifted to a more streamlined geometry. The rationale was rooted in the fact that the surrogate kidney —essentially a pliable material cube— boasted a basic configuration. This informed decision led to the creation of the model depicted in figures 3.5 and 3.6.

Another challenge arose while pinpointing the specific positions of the highlighted points. The mesh framework for the cube model was trimmed to the minimal scale before it would tax the computational capabilities, thus risking the simulation's real-time performance. This scaling adjustment meant that the spacing between the vertices of the planes simulating the IMUs ($1.6cm$) did not mirror the exact dimensions of the real IMUs ($1.5cmx2cm$). The planes are organized in a cross shape and a close inspection of the figures reveals a disparity in the point configurations for the top and bottom planes, which have fewer defining points. If solely four points were allocated for each plane, the consequence would be shared side points between two planes. This overlap engendered a series of computational and logical issues. Disparities in detected movement or displacement between the adjacent planes would confuse the solver, often causing the simulation to fail.

This conundrum was addressed by redesigning the side planes into pentagonal configurations, while the top and bottom planes remained as squares. Such an arrangement facilitated the independent and concurrent operation of all four planes.

The juxtaposition of the two models—namely the kidney and the square ob-



[!h]

Figure 3.3: Kidney at the opening of the scene, on the right-high corner it is possible to see the reference frame: x axis (red), y axis (green), z axis (blue)

.

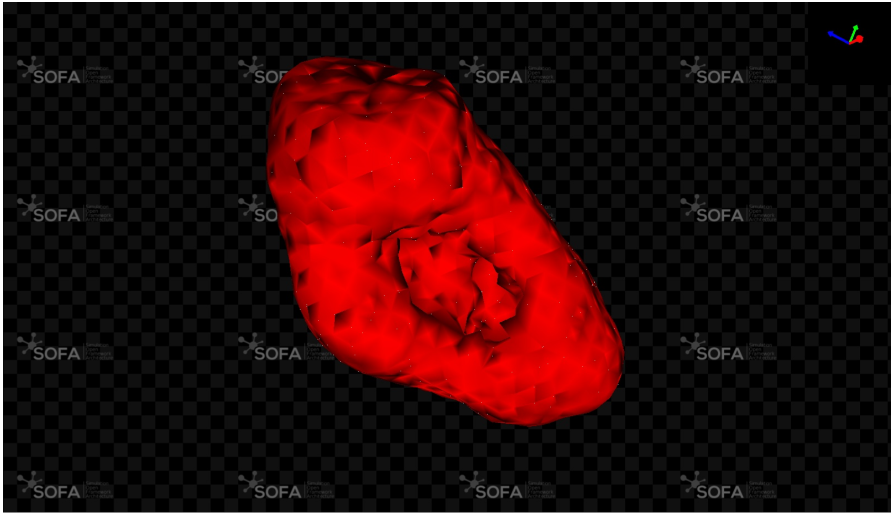


Figure 3.4: Hollow parts of the kidney model due to the removal of tumoral masses and vessel models.

ject—reveals distinct disparities, primarily in the mesh density. In the kidney's visual representation, the force field was deliberately excluded, as its presence would overshadow the intricate surface details, rendering the entire image an indistinct blue hue. This omission was strategic to ensure the viewer's ability to discern the kidney model's unique surface structure and texture.

Conversely, in the square object, the force field is presented prominently in

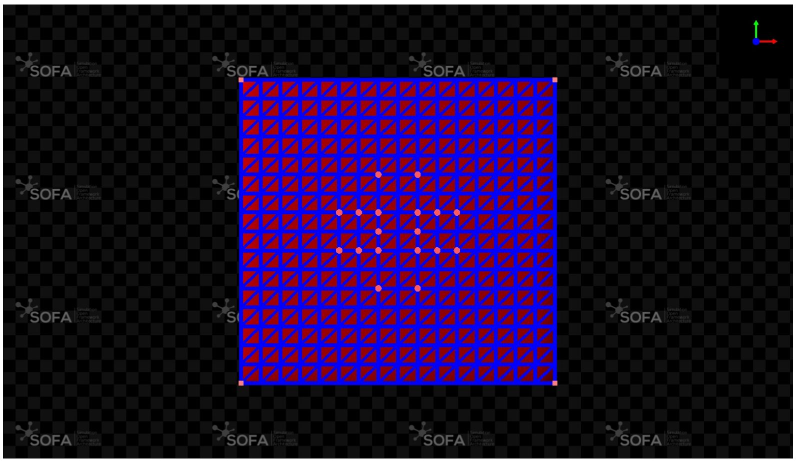


Figure 3.5: Simple geometry representing the polyurethane shape. The blue lines represent the mesh subdivision and the Force field applied, which has the characteristic of being a Spring-Mass force field relationship.

.

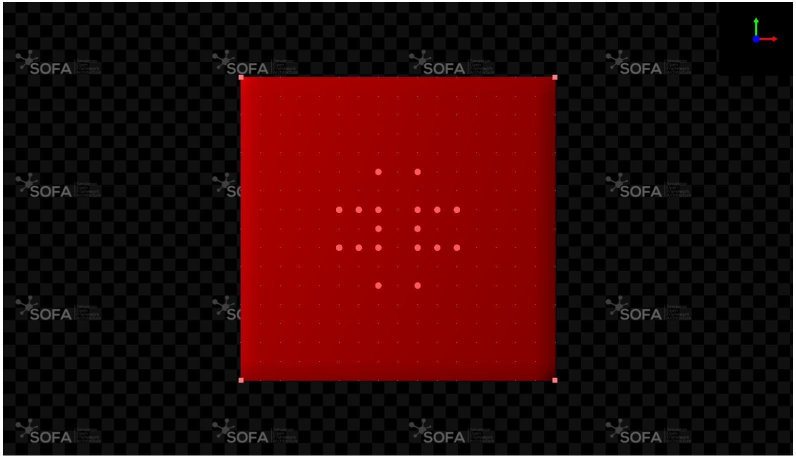


Figure 3.6: Here is a simple picture of the object. The spheres represent the points of the meshes constrained to be the vertexes of the planes that the IMUs will control. The ”square” points at the 4 vertices of the bigger square are fixed points to allow deforming without having the object moving in the environment.

a grid-like formation. However, what might elude an initial observation but becomes evident in figures 3.7 and 3.8, is the contrasting makeup of the mesh files for each model.

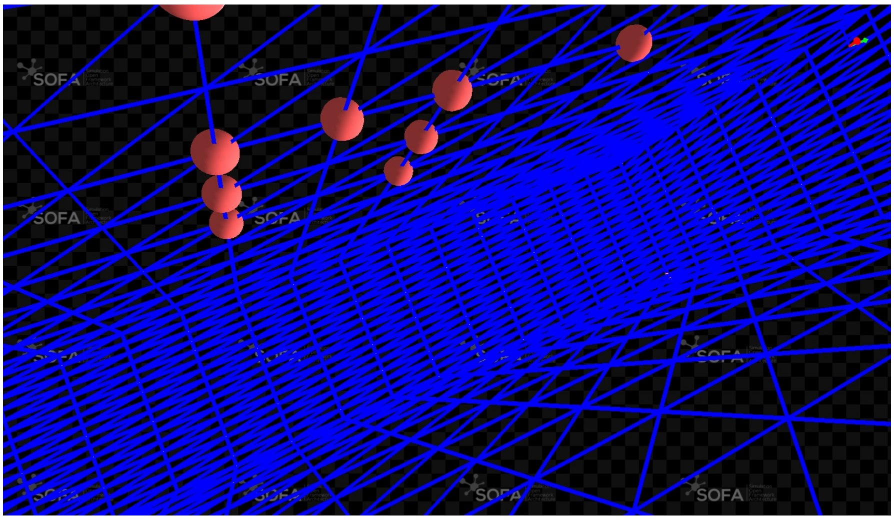


Figure 3.7: The square model mesh is empty from the inside, hence the behavior is different with respect to the kidney full mesh

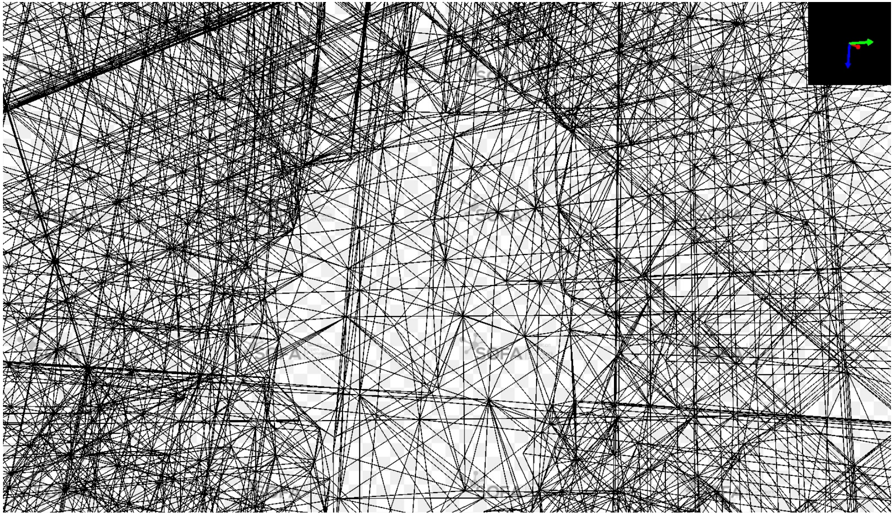


Figure 3.8: Here the kidney mesh is shown highlighting how the body of the mesh is highly connected even between different levels of the mesh.

Last but not least issue found in the model composition, was the possibility of creating a mixed properties model. Since the idea at the base of the model was to have a deformable model with rigid planes that would be controlled and will transmit the movement to the rest of the mesh, the necessity of having a model with different properties on the desired meshes was necessary. Unfortunately, the multi-material and multi-physics feature in SOFA is complex to implement. Firstly, the build version of the application would be needed (while for this experiment the simple binary version was used) and then understand how to adapt the codes coming from the previous version of SOFA, which had more documentation about it. Eventually, due to the time limits and to delay in discovering the possibility of applying the feature, the model produced does not mirror the idea proposed and the result is a not realistic behavior that will be not useful in the evaluation of the results.

This might sound like an endpoint but is a starting line for the future work that this research is opening up to, intending to create a Digital Twin with the open source software, more and more improvements will come considering the vast community interested in these applications.

3.3 TIAGo robot

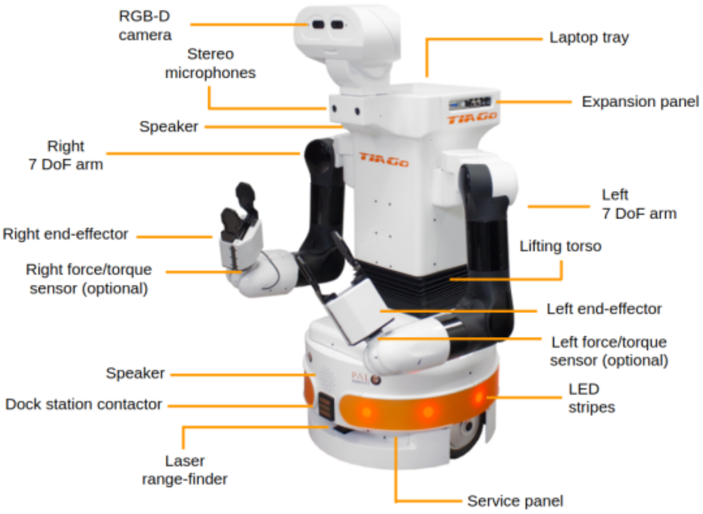


Figure 3.9: TIAGo robot and all its main components

Within the context of sensors associated with the experimental endeavors on the Physical Twin side, an introduction to the TIAGo robot force sensor will be provided. TIAGo++ 3.9, a further development in the PAL Robotics repertoire,

Table 3.1: Torque/Force sensor's specifications

Physical Specs	Weight	Diameter	Height	
	0.0917 Kg	45 mm	15.7 mm	
	F_x, F_y	F_z	T_x, T_y	T_z
Sensing ranges	290 N	580 N	10 Nm	10 Nm
Resolution	1/8 N	1/8 N	1/376 Nm	1/752 Nm

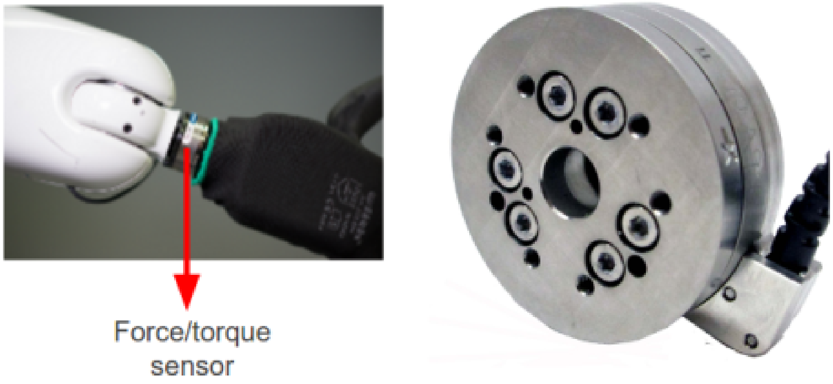


Figure 3.10: TIAGo robot force sensor, used to detect the real force imposed

features a singular bi-manual design, complete with two 7-DoF arms. Built upon the modular philosophy of the original TIAGo robot, PAL Robotics has integrated these exceptionally dexterous arms into the mobile manipulator, thus developing TIAGo++ to address the ongoing demands of the robotics community. This advancement allows TIAGo++ to shine in mobile manipulation tasks, empowering researchers to attain their objectives. It preserves the fundamental characteristics of its predecessor, being ROS-based, fully customizable, augmentable with additional sensors and apparatus, and demonstrably robust. With its bilateral capabilities, TIAGo++ offers an attractive and flexible choice for those seeking a trustworthy platform for their research.

The TIAGo++ robot comes equipped with a Force/Torque sensor 3.10 with the specifications in table 3.1, specifically, an ATI mini45 sensor, strategically positioned at the endpoint of its wrist. This sensor is tasked with delivering feedback on the forces applied to TIAGo++'s end-effector during its operations. Offering six-axis force and torque measurements, it plays a vital role in assuring precision and control in the robot's activities. Due to these particular characteristics, the robot was chosen as an adept tool to impose different types of displacements on the object's surface and also as a reliable ground truth for the inferred force values.

Additionally, to impose the deformation, the prismatic joint at the core of its body will be utilized, providing a reliable value on the actual displacement imposed on the material, in relation to the calculated one.

3.4 IMU sensor

In this paragraph, the specification, communication architecture, and setup of the sensors will be described.

3.4.1 Specifications

The setup involves three primary components working together to ensure data is transmitted seamlessly to the main device for processing.

The first element to be described is the IMU sensor, specifically the MPU-9250 Nine-Axis (Gyro + Accelerometer + Compass) MEMS MotionTracking™ Device[1] (figure 3.11 and 3.12), a product of InveSense under TDK Corporation. This compact device, encased in a 3x3x1 mm unit, stands out for facilitating detailed high-resolution compass data, which simplifies the task of integrating compasses into complex PCBs. The MPU-9250 is essential in providing accurate orientation data, eliminating the need to engage in detailed velocity and acceleration analyses.

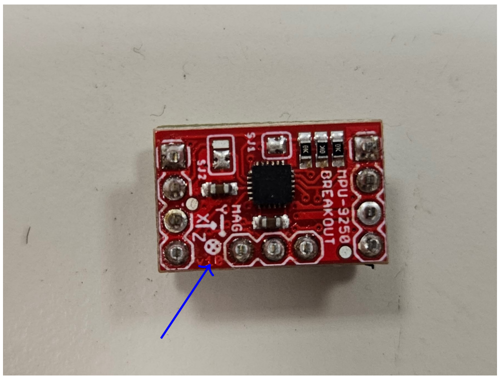


Figure 3.11: IMU sensor MPU-9250. On the board, the reference frame the IMU will use is shown.

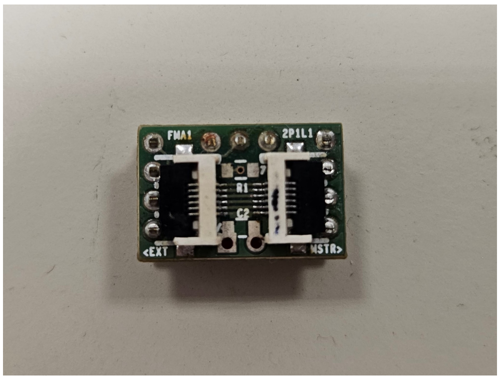


Figure 3.12: IMU communication board, used to connect the single IMU to the bigger Board that will then connect to the microcontroller and the computer

[1] https://invensense.tdk.com/products/motion-tracking/9-axis/mpu-9250/

Communication with the laptop is facilitated through the Firebeetle ESP-32 (figure 3.13), an Arduino-compatible microcontroller. Developed by the DF Robot team[1], this low-power microcontroller is specifically designed for IoT projects and is part of the FireBeetle series. It operates based on the Dual-Core ESP-WROOM-32 module, ensuring MCU and dual-mode communication through Wi-Fi and Bluetooth functionalities.

Lastly, all components are integrated with the ELEGOO UNO R3 BOARD[2] (figure 3.14), a microcontroller board modeled after the Arduino UNO R3, and shares many of its features. The codes that regulate the communication were already developed by the team of the EngineRoom as these components are custom electronic devices, included in the PhD project of Belcamino Valerio. The communication codes are explained and well referenced in the dedicated GitHub repository[3].

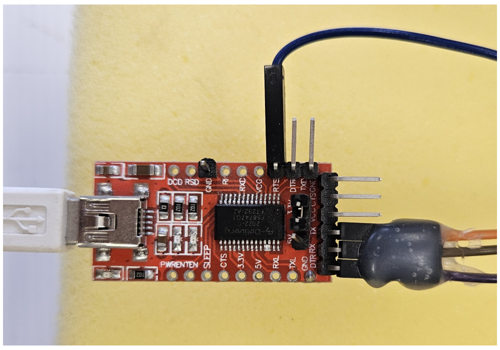


Figure 3.13: FireBeetle processor.

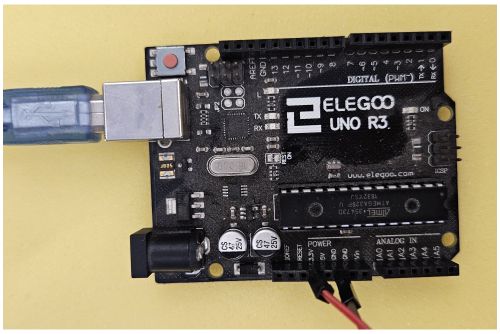


Figure 3.14: Microcontroller ELEGOO UNO R3 to communicate with the laptop.

[1] https://www.dfrobot.com/product-1590.html

[2] https://www.elegoo.com/en-it/collections/uno-r3-starter-kits/products/elegoo-uno-r3-board

[3] https://github.com/TheEngineRoom-UniGe/InDex_Glove

3.4.2 Set up

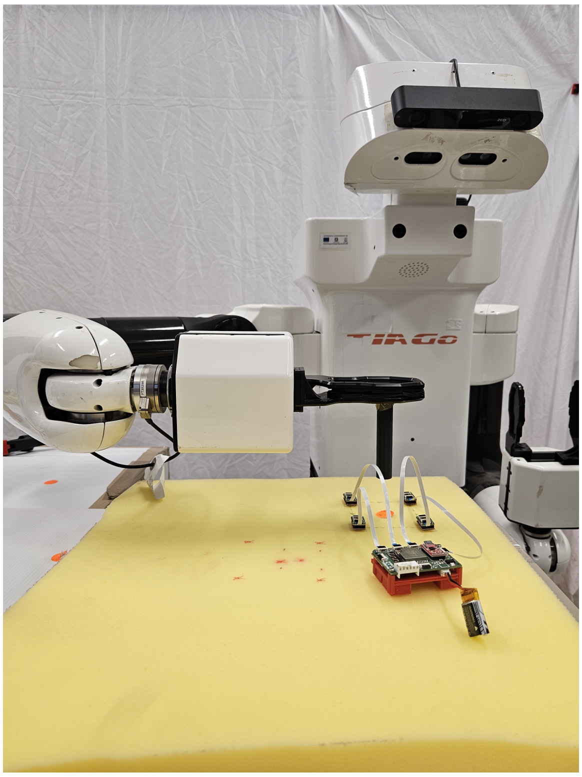


Figure 3.15: Complete set up before the actions start. TIAGo robot in position over the workspace, ready to impose the displacements. The sensors are positioned in the best-designed shape to detect such displacement.

In the preparatory phase of crafting the sensing scenario, a meticulous selection process was undertaken encompassing the choice of a substitute material to represent a kidney, the assessment of the optimal count of IMUs to deploy for ideal deformation detection resolution, rationing it with the relative position of the IMUs between each other; a task steeped in a balance between theoretical rigor and real-world application. This deliberative process necessitated a set of formulated assumptions.

To evaluate the sensing approach, the first step was to identify an appropriate material to use as an experimental sampling phantom. For this duty, a fragment of a polyurethane cube (material characteristics in table 3.2) was designated as the representative physical model. This material, frequently found in mattress manufacturing, was the most readily available and fitting candidate for the experimental objectives. Although its elastic characteristics diverge from the kidney's natural attributes, this difference was assumed irrelevant to the central focus of the project, which remained to assess the efficiency of IMUs in a designated detection setup, to precisely detect deformations and gather useful data to represent it in simulation. The selected prism, with dimensions of 0.14m x 0.14m x 0.05m, maintained a plain surface to ease the initialization process. The IMUs were strategically placed in a large area proximal to the center of the model to maintain a sufficient distance from the borders of the geometry but relocated to another point after a certain number of tests because of the material loss of elasticity and resistance.

The material and the specific details about it were given by the mattress shop Gatti e Materassi[1]

Table 3.2: Material characteristics description, limited to those information that were useful for the experimental setup and for the force calculations.

Characteristic	**Value**
Prism dimensions	0.14m X 0.14m X 0.05m
Action area	$\pi \cdot (0.0075m)^2$
Density	$26\,\mathrm{kg} \cdot \mathrm{m}^3$
Compressive hardness	3.6 KPa
Tensile strength	100 KPa

Further assumptions regarding the most appropriate locations for the IMUs were also made. An imaginary square was envisioned, with IMUs placed at its corners, conceptually contained within a circle, which center indicates the point where the force would be applied. The circle's diameter was based on the size of the pressing tool used in the experiment, a 1.5 cm diameter cylinder, setting the circle's radius at 3 cm. This conceptual framework aimed to facilitate as precise deformation detection as possible, given the flat surface of the polyurethane model and observing that any shape used would produce a circular deformation shape, in figure 3.16 the concept applied. It suggests that different objects enduring this pressure would require different setups, determined by their specific mechanical properties. For a kidney, a detailed study of its mechanical properties to design an optimized detection setup would be a valuable direction for future research in this area.

[1] https://materassigenova.com/

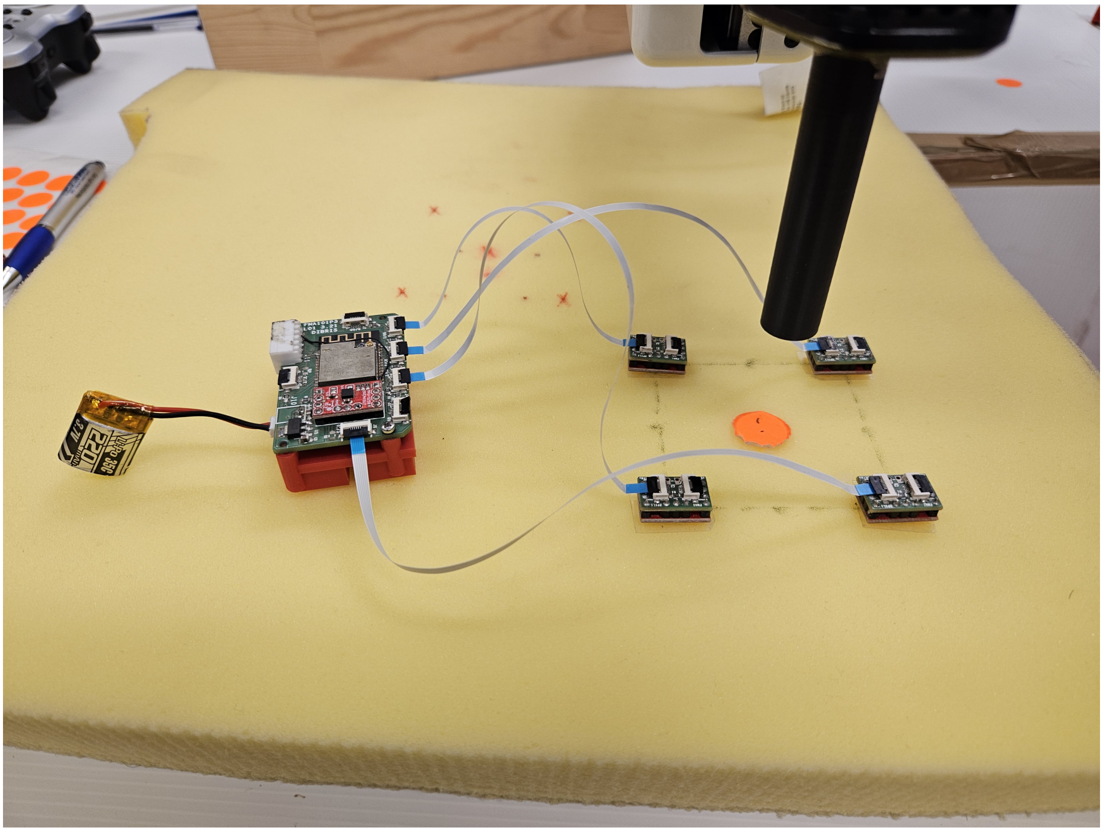


Figure 3.16: IMUs relative positions. The orange dot represents the point on which the force is applied by the black cylinder, while the board to which all the IMUs are connected is what ensures the possibility of connection and is part of the microcontroller interface

Another assumption involved the method of securing the sensors to the object. In this scenario, they were placed on the polyurethane surface using thin double-sided tape to maintain adherence and reduce potential variations during deformation. It is noted that this type of attachment would not be appropriate for a real kidney, especially in the context of live surgery, highlighting another area for future improvements in this project, which serves as a foundational step in a novel detection approach. The same simplification assumption is made on the tool interacting with the object, which is a simple cylinder 3D printed, shown in figures 3.17 and 3.18.

Figure 3.17: Cylinder $8cm$ tall

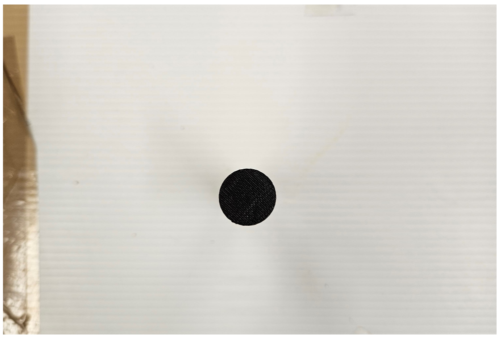


Figure 3.18: Cylinder $1.5cm$ wide

3.4.3 Calculation methods

The code to calculate the strain and the force applied to the material, starting from the IMU orientation is unrelated to the simulation and is connected to it just in the transmission of the real-time orientation of the planes to represent the deformation starting from the normal values of each sensor. This code aims to validate the simulation's accuracy, ensuring that the mathematical logic and results will correspond to the deformation the simulation calculates by simply mirroring the movement. Another step of the validation is to calculate by mathematical means the force that should generate such a strain on the given material and then apply this force to the simulation to see if the characterization of the material is related to reality. For the above reasons, the algorithm of the code will be explained without representing the simulation communication. In diagram 3.19 the flowchart describes the code logic. There is a thread always running, which is the thread reading the orientation values transmitted by the four IMUs. A separate thread, which does not run concurrently with the first, activates whenever there's a plot to process. This design ensures uninterrupted reading of the sensor data, which consistently streams until the code's termination. The readings from the sensors are expressed in quaternions—a mathematical structure used to depict rotations relative to a frame— and represent the orientation of the sensor in space from which is possible to derive the normal vector of the IMUs itself.

The code steps are straightforward: after the UDP messages coming from the IMUs are unpacked, the reference vector gets populated and stored for further use as explained further in this paragraph. Immediately after the population of the references, one for each IMU, the normal vector gets computed by applying the rotation described by the quaternions to the original normal vector, which direction is only along the z axis, perpendicular to the object face in the direction of the ceiling.

Once this information is fully acquired and the calculation is running in real-time, the intersection point is calculated in a function that uses the least square method to minimize the error and acquire a more precise approximation using as an advantage the over-determination of the system.

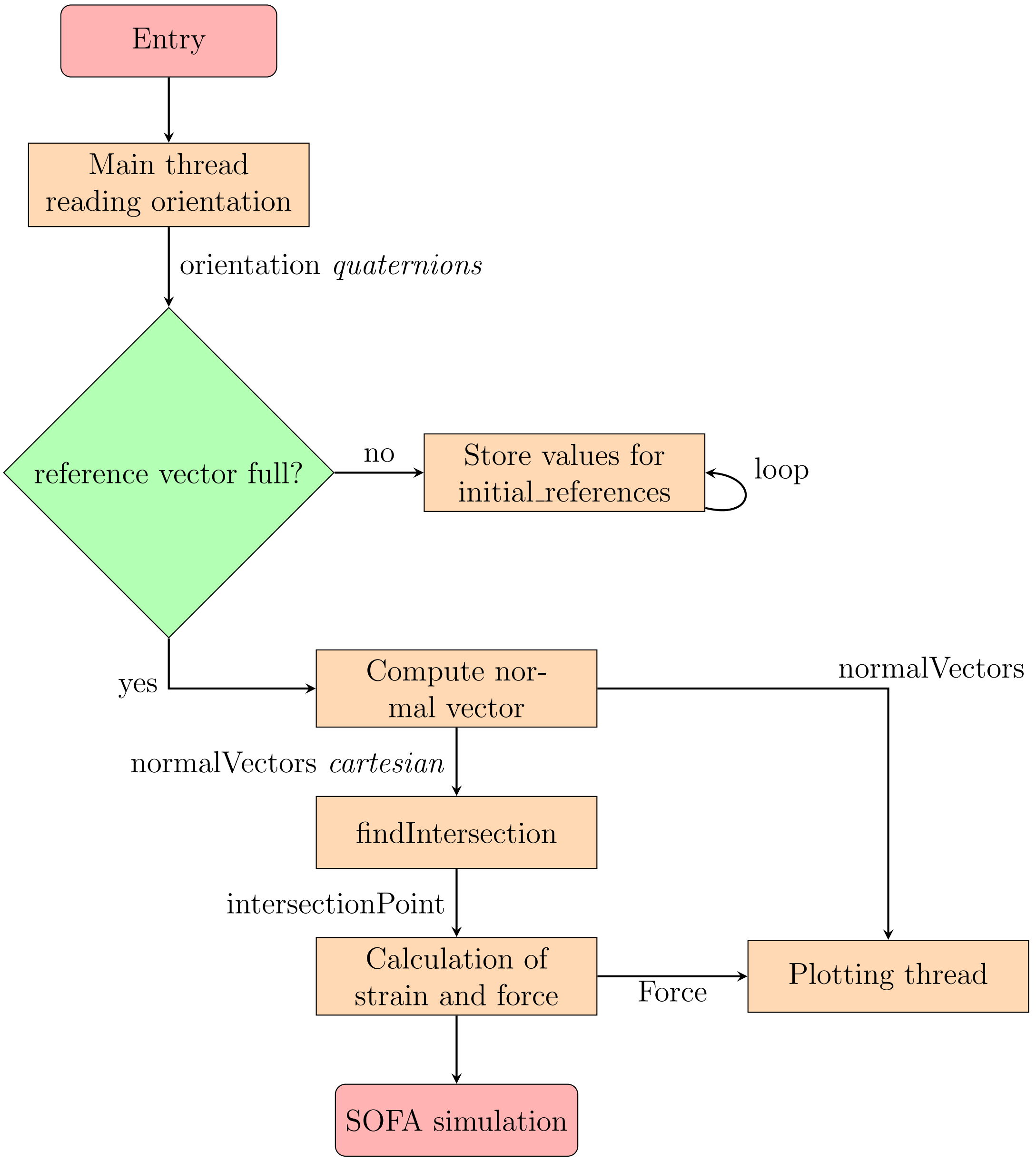

Entry
Main thread reading orientation
orientation *quaternions*
reference vector full?
no
Store values for initial_references
loop
yes
Compute normal vector
normalVectors
normalVectors *cartesian*
findIntersection
intersectionPoint
Calculation of strain and force
Force
Plotting thread
SOFA simulation

Figure 3.19: Flowchart describing the logic of the gathering data code used during the experiments. The main thread and the plotting thread are not concurrent to each other, they run independently ensuring the possibility of reading data while the code is processing the plotting.

The intersection point is the main element that will bring to the inferring of the force that would be needed to apply such deformation.

The SOFA simulation node is represented in the flowchart as the last step, but the reader needs to keep in mind that all these processes are running simultaneously and here the position of the simulation indicates that all these data get to be sent and applied in the SOFA *scene*, especially the normal vectors, which will be passed to the simulation as new normal vectors for the planes representing the IMUs and the force is passed to asses if the force inferred would produce the same displacement also in the simulation environment.

The calculation of the normal was already part of the repository[1], the math is reported in the following steps:

Initially, the reference quaternion from the main IMU must be stored in the vector of the initial_reference, initialized as a vector comprising four null elements:

$$\mathbf{v}_{ref}[0] = \mathbf{q} \tag{3.1}$$

Where:

- $\mathbf{v_{ref}}$ = is the vector of four elements (‘initial_reference‘) of which just the first position is filled
- $\mathbf{q}$ = is the orientation vector of quaternions directly transmitted by the IMUs

where *orientation* signifies the initial message relayed by the reference IMU. Subsequently, the remaining ‘initial_reference‘ entries should be populated, relying on the primary reference frame vector. Starting with the offset vector—which, in this context, is absent but retained for mathematical precision—it’s defined as:

$$offset = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{3.2}$$

Utilizing the above, the relationship is expressed as:

$$\mathbf{v}_{ref}[i] = \mathbf{v}_{ref}[0] \cdot offset \cdot \mathbf{q}_i^{-1} \tag{3.3}$$

Here,

$$\mathbf{q}_i$$

corresponds to the first orientation sourced from the i^{th} IMU, inverted then in the formula

[1] https://github.com/TheEngineRoom-UniGe/InDex_Glove

The aforementioned steps encompass calibration procedures executed singularly. Thus, post the comprehensive population of the ‘initial_reference‘ vector, these values remain unchanged and are requisitioned as required in the real-time computation of the normal vector, which is:

$$\mathbf{n}_{i^{th}} = \mathbf{v}_{ref}[i] \cdot \mathbf{q}_i \cdot \mathbf{n}_\mathbf{z} \tag{3.4}$$

Here, nV represents the normal vector in its idle stance, aligned with the z axis, needed to calculate the normal vector described by the rotation in space given by the q vector.

Once all the steps are completed the normal vector of each specific IMU is calculated.

Here an important specification on the frames acting in the scene is necessary. As mentioned in section 3.4, the IMUs have their own reference frame, which needs to be adjusted to the world frame and to the simulation frame. The previous steps ensure that world and IMU frames are the same, but to adjust it to the simulation the indices of the normal vector need to be shifted as follows:

$$AdjustedNormal = \begin{bmatrix} -y & x & z \end{bmatrix} \tag{3.5}$$

After these steps, the ‘AdjustedNormal‘ vector is obtained for each IMU and can be directly applied to the simulation to control the planes as explained in section 3.2.3.

Once the normal values are computed, the second step of the logic is to calculate strain and resulting force. These values are derived from the plane intersection calculated based on the geometry concept of the plane.

Given four planes in space, each plane can be represented using its normal vector and a point on it. The equation for a plane can be given by:

$$\mathbf{n} \cdot \mathbf{x} = \mathbf{n} \cdot \mathbf{r} \tag{3.6}$$

Where:

- $\mathbf{n}$ = normal vector of the plane
- $\mathbf{x}$ = a general point $[x, y, z]$ in space
- $\mathbf{r}$ = a known/reference point on the plane

For our four IMUs, we have:

IMU North: $\mathbf{n}_N$ and reference point $\mathbf{r}_{01} = [3, -3, 0]$ IMU South: $\mathbf{n}_S$ and reference point $\mathbf{r}_{02} = [-3, 3, 0]$ IMU East: $\mathbf{n}_E$ and reference point $\mathbf{r}_{03} = [-3, -3, 0]$ IMU West: $\mathbf{n}_W$ and reference point $\mathbf{r}_{04} = [3, 3, 0]$

Combining the equations for the four planes, we get the matrix equation:

$$A\mathbf{P_{new}} = \mathbf{b} \tag{3.7}$$

Where:

$$A = \underbrace{\begin{bmatrix} \mathbf{n}_N \\ \mathbf{n}_S \\ \mathbf{n}_E \\ \mathbf{n}_W \end{bmatrix}}_{4\times3} \tag{3.8}$$

and

$$\mathbf{b} = \underbrace{\begin{bmatrix} \mathbf{n}_N \cdot \mathbf{r}_{01} \\ \mathbf{n}_S \cdot \mathbf{r}_{02} \\ \mathbf{n}_E \cdot \mathbf{r}_{03} \\ \mathbf{n}_W \cdot \mathbf{r}_{04} \end{bmatrix}}_{4\times1} \tag{3.9}$$

To solve for the intersection point $\mathbf{P_{new}}$, we use the least squares method:

$$\mathbf{P_{new}}, \text{residuals}, \text{rank}, s = \text{np.linalg.lstsq}(A, \mathbf{b}, \text{rcond=None}) \tag{3.10}$$

This gives us the point $\mathbf{P_{new}}$ in space that minimizes the sum of the squared distances to the four planes. The minimization is achieved with the resolution of the least squares problem, which aims to find an approximate solution to an over-determined system of linear equations. Specifically, given a matrix A and a vector b, the goal is to find a vector x such that $\|Ax - b\|$ is minimized, where $\|\cdot\|$ represents the Euclidean norm. When a system of linear equations has more equations than unknowns, it's unlikely to have a unique solution. The least squares approach provides the best approximation.

The function `np.linalg.lstsq` from the NumPy library solves the linear matrix equation, or system of linear scalar equations, using the least squares method. Its signature is:

```
np.linalg.lstsq(a, b, rcond='warn')
```

Where:

- a is the coefficient matrix.
- b is the dependent value array.
- `rcond` is a parameter used to cut off small singular values. Singular values smaller than `rcond` times the largest singular value is set to zero.

The return results of this function are:

1. x : Least-squares solution. Its shape is identical to b.
2. `residuals` : Sums of residuals; squared Euclidean 2-norm for each column in $b - a \cdot x$. If the rank of a is ¡ N or $M \leq N$, this is an empty array.
3. `rank` : Rank of matrix a.
4. s : Singular values of a.

The process behind the function resolution comes down to the solving of the following equation:

$$x = (A^T A)^{-1} A^T b$$

However, directly computing this can be numerically unstable, especially when $A^T A$ is nearly singular. Instead, `np.linalg.lstsq` employs a more robust method: the Singular Value Decomposition (SVD).

SVD decomposes A into three matrices:

$$A = U \Sigma V^T$$

Where:

- U and V are orthogonal matrices.
- Σ is a diagonal matrix containing singular values.

Using SVD, the least squares solution x can be computed as:

$$x = V \Sigma^{-1} U^T b$$

Here, the pseudoinverse Σ^{-1} is obtained by taking reciprocals of non-zero singular values in Σ.

In conclusion, `np.linalg.lstsq` is a powerful method in NumPy for solving the least squares problem using the robust Singular Value Decomposition technique. It provides a way to find the best approximate solution to an over-determined system of equations, like the system of the experiment since there are four planes to determine one point of intersection, which is the result of the intersection of three planes. The over-determined condition was necessary to achieve positive results since with a lower number of IMUs the calculations were not satisfactory and not repairable in the short-term period.

The calculation of the planes intersection point allows the approximation of the deformation of the material and gives a general solution not based on the material properties.

Different are the considerations on the estimation of the force used to impress such a deformation. The force calculation depends form the material properties.

When a material is subjected to an external force, it deforms. The amount of deformation with respect to its original dimension is termed as strain. The formula for strain (for small deformations) is:

$$\text{strain} = \frac{\text{change in length}}{\text{original length}} \tag{3.11}$$

in which the original length is equal to zero, so the strain is equal to the point z coordinate. Stress is defined as the force applied per unit area on which the force acts. It can be mathematically expressed as:

$$\text{stress} = \frac{\text{force}}{\text{area}} \tag{3.12}$$

Young's Modulus (often represented as Y) is a measure of the stiffness of a material and it is extracted from table 3.2. It defines the relationship between stress (σ) and strain (ϵ) in a material under axial load. The relationship is linear for many materials until a certain limit (the proportional limit). The formula is:

$$\text{stress} = E \times \text{strain} \tag{3.13}$$

From this definition, stress is the force per unit area. Thus, if you want to find the force applied to a material when you know the stress and the cross-sectional area (S) upon which it acts, the formula is:

$$\text{force} = \text{stress} \times S \tag{3.14}$$

Putting it all together:
Given the relation

$$\text{stress} = E \times \text{strain} \tag{3.15}$$

and substituting this value into the second equation, you get:

$$\text{force} = (E \times \text{strain}) \times S \tag{3.16}$$

After these calculations, the code is concluded and it can either continue to calculate the strain and the force in real-time or it can be interrupted. In the following section, the results of all the methods explained here will be shown and analyzed.

Chapter 4

Results and discussion

The final experimental setup design involves TIAGo robot, equipped with the 3D printed cylinder and the four IMUs MPU-9250 in a cross position with respect to each other, attached to the material as depicted in the setup section 3.4.2, see figure 3.15.

The experimental procedure consisted of distinct steps, as enumerated below:

1. **Scene Setup**: The scene was placed for the experiment.
2. **Positioning the TIAGo Arm**: Leveraging the option to deactivate gravity compensation mode, to position TIAGo's arm, facilitating the arm's free movements.
3. **Initiating the Recording**: With the scene in place, the code was started to log the results from the IMUs, as well as the force perceived by TIAGo via its in-built force sensor.

Upon the successful setup, the detection process was segregated into three pivotal phases, differentiated from the normal vector value derived from the IMUs orientation description and the prismatic joint position of TIAGo robot. Thanks to the robot joint, a pure vertical force was applied to the material. This detail needs to be remembered throughout the full experimental analysis. The duration of the experiment is 1 hour and 30 minutes since all the phases are positioned and maintaining the position for 30 minutes each. The phases are characterized by three types of contact with the material, as precisely described as follows:

- **NoContact Phase**: Characterized by a state of inactivity, this phase was crucial despite its steadiness. Both TIAGo's sensor and the IMUs have inherent offsets when "no movement" is detected, making this phase fundamental for the experiment's execution.

- **Contact Phase**: Here, a modest deformation of $1cm$ or less, is induced on the deformable subject. This phase was enlightening regarding the IMUs' performance: it showcased how frequently accuracy errors manifested. There were instances where the IMUs failed to discern evident displacements, especially when these were less than the $1cm$ threshold.

- **DeepContact Phase**: This is the concluding phase, marked by the robot exerting substantial pressure, making the cylinder delve as deeply as possible into the deformable object. The peak of displacement during this phase was $2cm$. Beyond this measure, the TIAGo robot typically encountered difficulties due to the object's resistance.

Upon the completion of the phases, the plots were generated and are now set to be analyzed.

4.1 Results

The results section will adhere to the organized structure of the experiments, initiating with the NoContact phase. The analysis will then proceed to observe the behavior of the sensors and the material under varying displacements.

4.1.1 NoContact Phase

This represents the initial and stationary state of the scene, which has been previously illustrated in the text. Figure 4.1 is presented as a close-up, depicting the hardware connection methodology of the IMUs with the board and the working space of the robot. The frame on which the IMUs are setup and the force action point are highlighted in the picture.

While this phase is active, a logical gathering component of the code is operational, capturing all the values perceived by the sensor and by TIAGo's force sensor. These are subsequently plotted, as shown in figure 4.2 and 4.3 for the IMUs' calculations, 4.4 illustrating the inferred force, as detailed in the preceding section 3.4.3 and 4.4 reading coming from the embedded torque/force sensor.

Figure 4.2 and 4.3 shows two different plots:

- Plot (a) represents the vertical displacement of the intersection point, focusing solely on the z coordinates, which represent the total strain of the movement. The rationale for the isolation of the z coordinate in a different plot is given by the fact that the force in the experiment is applied purely on the vertical axis. It is possible to notice how the signal is not stable. It has an immediate drop to the value -0.15, which could be considered

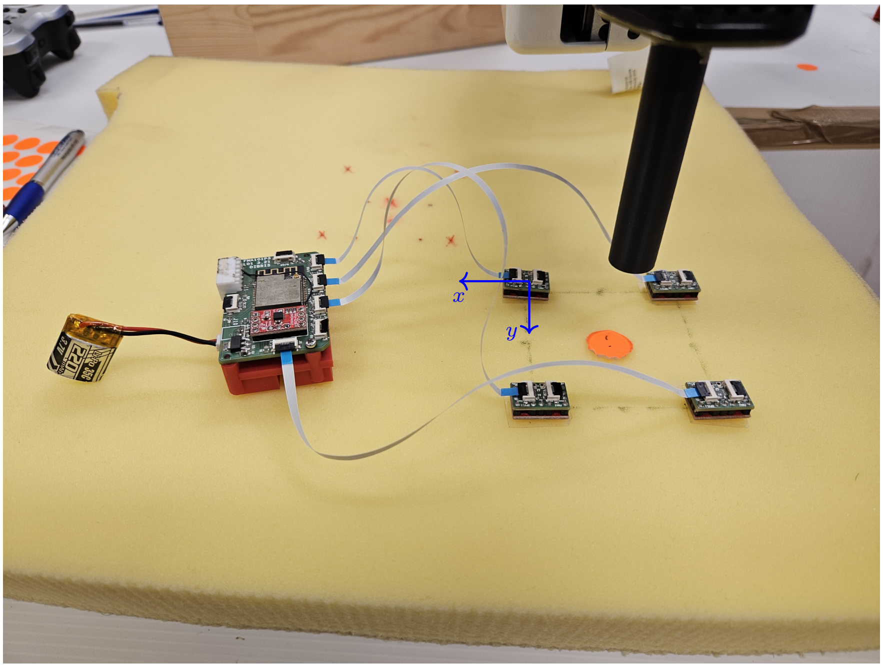


Figure 4.1: Scenario in NoContact phase. The orange dot is the application point of the force, and the reference frame shown refers to the one embedded in the IMUs itself.

as the 0 value for the IMUs in this phase and in this particular execution. After 3 minutes of behavior that could be classified as "stabilization", the value goes back to the initial displacement and presents a small amount of noise after the 20th minute of gathering.

- Plot (b) shows how mathematically, there's a deviation in the x and y coordinates due to the geometric theorem that states co-planar planes will never converge at a single intersection point. There are discrepancies in the computation of the intersection point, which ideally, should be at the 0 coordinates, centrally located between the four IMUs, while the 2D plot clearly shows otherwise.

These plots reveal that even in the most stable NoContact phase, technical anomalies persist and permeate the entire experiment. This is given also by the type of sensor used, IMUs are known for their noise collection problem and should be calibrated and go through a filtering phase.

Hence, taking into account the current observations and previous inferences, the results from these plots appear unstable and characterized by noise, particularly focusing on the z coordinates. Given the acknowledgment that a filtering

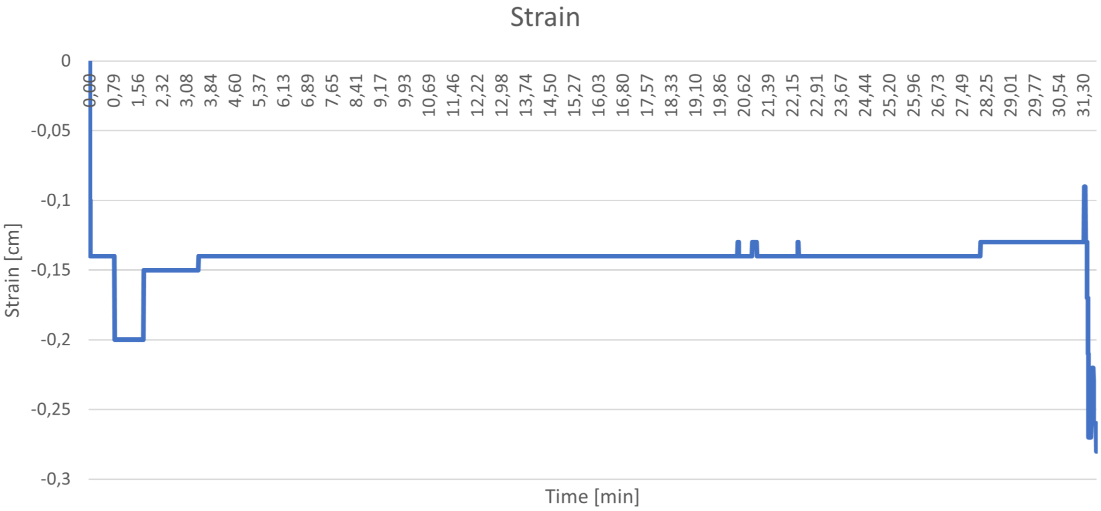


Figure 4.2: z displacement of the intersection point over time. The value is not static and is not adhering to reality. The immediate drop and the pikes during the perception are given by noise.

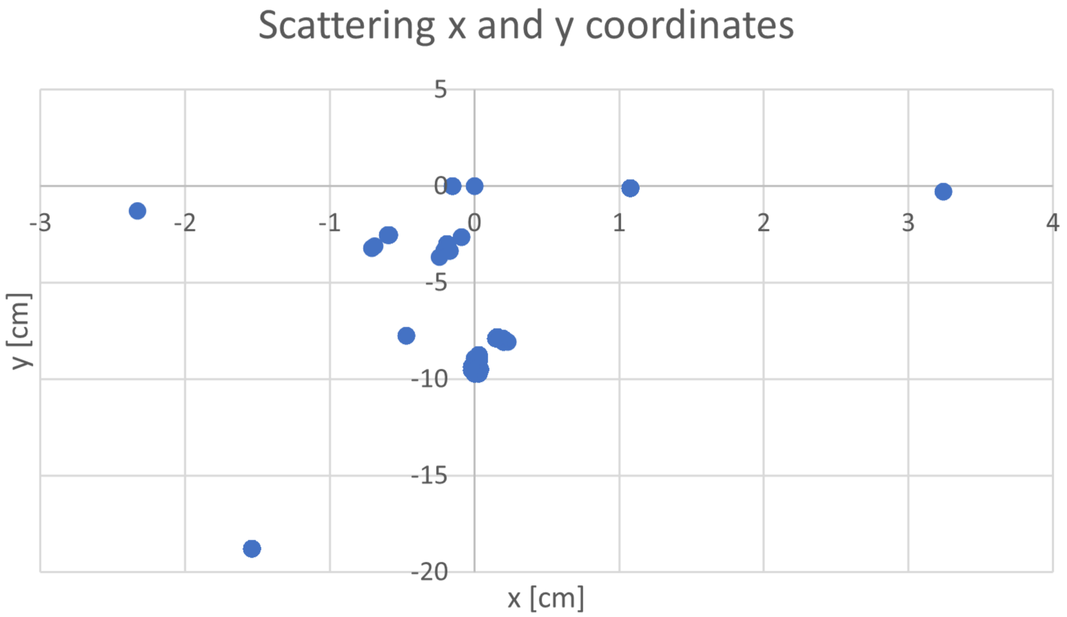


Figure 4.3: The scattering of x and y coordinates given by the difficulty of the algorithm to calculate an intersection point for planes laying on the same plane.

phase would be necessary for future work, as the present raw processing of the signal, the results have anyway a behavior that can be deemed satisfactory.

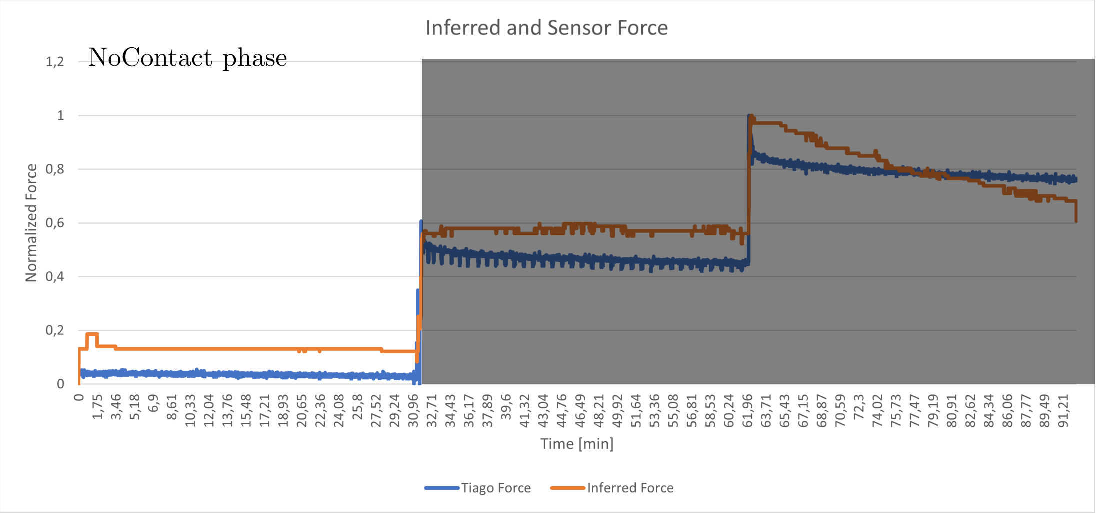


Figure 4.4: The orange path represents the force [N] inferred starting from the intersection point. Given the material characteristics and specification, as explained in the mathematical formula in section 3.4.3. The blue line instead represents the readings from the torque/force sensor of TIAGo robot in NoContact phase. For coherence with the other graphs, only the z component of the force is highlighted and analyzed.

A similar rationale is applied to the force readings, which display analogous deviations in the x and y dimensions. The exclusion of these coordinates from the findings is attributable to the fact that force is derived from the intersection point; consequently, errors propagate equivalently.

The forces displayed in the graph are normalized between 1 and 0 since there was an important distance of value ranges given by the two different environments the sensors were acting. TIAGo's force sensor, as explained in section 3.3, is influenced by different external factors, bringing it to different perception scales, while the second force is a force derived by calculation that involves the material properties and the intersection point for the specific time.

This assertion is substantiated by how the spike in the z coordinate perception of the intersection point in figure 4.2 mirrors a spike in the force plot 4.4. The force, akin to the intersection point's z coordinate, isn't precisely zero. This variance is evident, albeit inconsistently, in subsequent phases.

Throughout the text, it has been emphasized that the TIAGo's sensor was earmarked as a benchmark to discern the discrepancy between the inferred force and the actual force required to effectuate a specific displacement. As noticeable from the full graph, the robot's force sensor is describing the behavior in a reliable way, it has a noisy behavior but the area contained between the spikes is reliable. Focusing just on the NoContact side, is it possible to see how the values remain in the zero area, highlighting the discrepancy with the inferred values.

The reason why TIAGo's sensor shows noisy behavior is the compensation against the downward gravity plus the arm weight force. Notwithstanding, these graphs retain their utility in the analysis, especially since the offset appears relatively stable here. It suffices to examine the difference between this phase's offset range and succeeding phases to ascertain the force variation in Newtons.

4.1.2 Contact phase

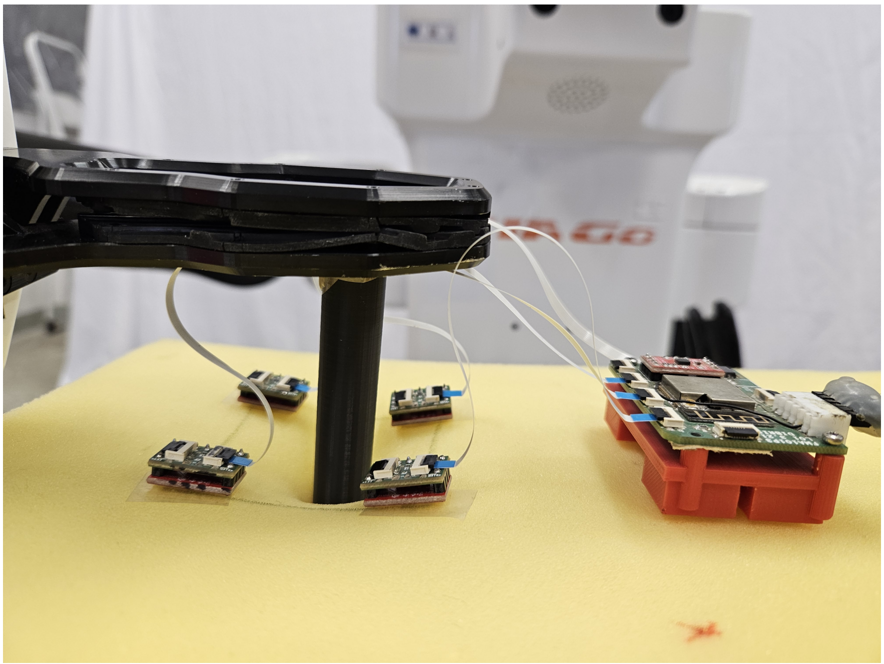


Figure 4.5: Contact phase scene from a side view perspective, to highlight the strain displacement

During the Contact phase, an initial strain, ranging from $1cm$ to $2cm$ at its peak, was applied to the material. The goal was to discern whether the sensors could aptly respond and identify either systematic or accidental anomalies.

As shown in figure 4.6, the z displacement derived from the approximation ranges around the value $-0.6cm$, which is still not equal to the displacement imposed to this phase and in a surgical environment would need tuning and compensations before being used in a reliable way.

Moreover, is possible to notice how the strain values in this phase are less stable and more noisy. Given the positioning of the IMUs it is understandable how such vibrations could come from external factors as from the TIAGo robot arm, which had vibrations to stabilize itself in each phase, inducing involuntary movements in the scene.

Figure 4.7 represents the scattering and here the difference with the previous phase is noticeable in the number of scattered dots and in the ranging area of the scattering. In the previous phase, they were extended to $-10cm$ on the x axis and $\pm 3cm$ on the y, while now the range on the y axis is halved, and the one on the x axis changed direction going mostly on the positive side and concentrating around 0.

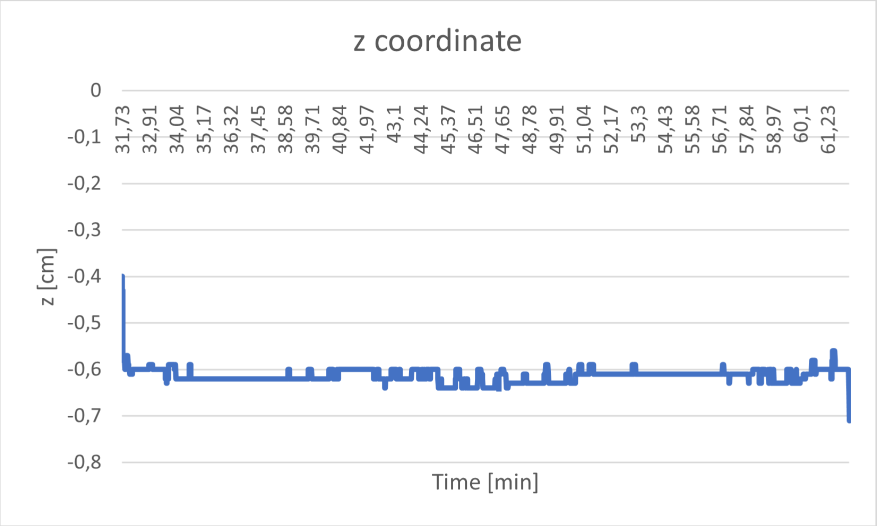


Figure 4.6: z displacement of the intersection point in Contact over time. The behavior is noisier but more stable and linear.

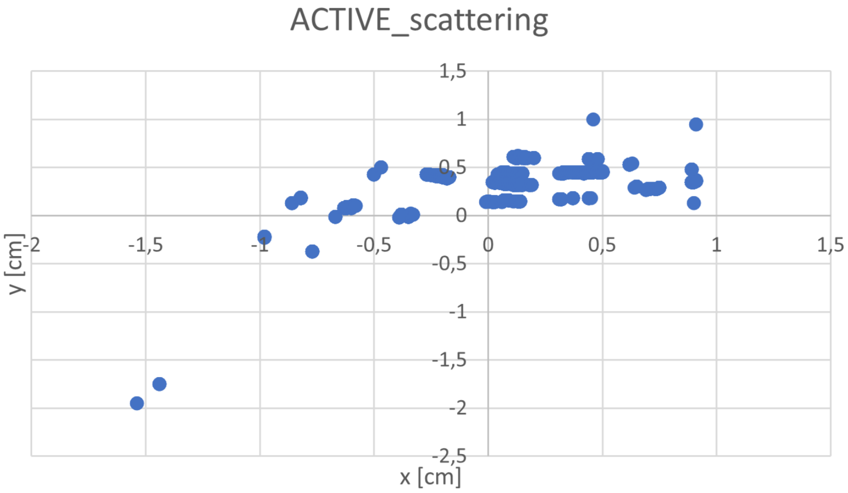


Figure 4.7: Scattering of x and y coordinates in Contact, here is possible to notice how the scattering area is smaller with respect to the previous phase, but the number of different coordinates calculated is higher. The improvement of the scattering area is given by the fact that the planes are not co-planar anymore, while the higher number of different values is given by external vibration and noise as it happened for the strain values..

On the other hand, the force plots 4.8 present different patterns. In the NoContact phase, TIAGo's sensor indicated a negative component, attributed

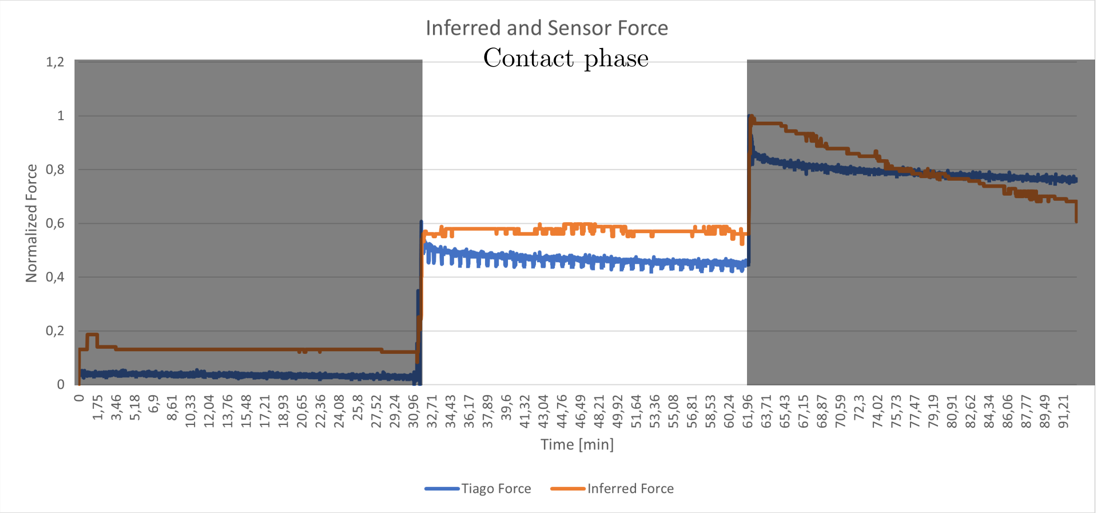


Figure 4.8: The forces here show the same displacement between each other as in the previous phase. They follow a linear direction, except for the TIAGo force that in the beginning has a drop caused by the deactivation of gravity compensation, since the arm now is balancing its weight partially with the material resistance.

to gravity compensation. In this phase, there's a shift as the force significantly reduces. This reduction can likely be explained by the material resistance starting to counterbalance the compensation factor. The force perceived in this phase deviates notably from that in the previous phase. Given the sensor's inconsistent behavior in various tests, it's reasonable to say that the observed behavior is reflective of the situation, but the numeric variations are likely due to noise and calibration errors.

Given these considerations, when the displacement imposed respected the perception ranges creating a change in the normal vector derived from the IMUs, the z displacement approximation is revealed to be precise enough, so it's possible to ensure that in the moment the displacement is recorded, the result will be a valuable approximation of the real case.

4.1.3 DeepContact phase

The last phase of the experiments aimed to test the extreme conditions of a strong force. This type of displacement will rarely appear in a surgical environment, but is important to test every angle of the sensors' capabilities. Here a strain range of $2cm$-$3cm$ is reached and the sensors tilt strong enough to ensure immediate recognition of the displacement.

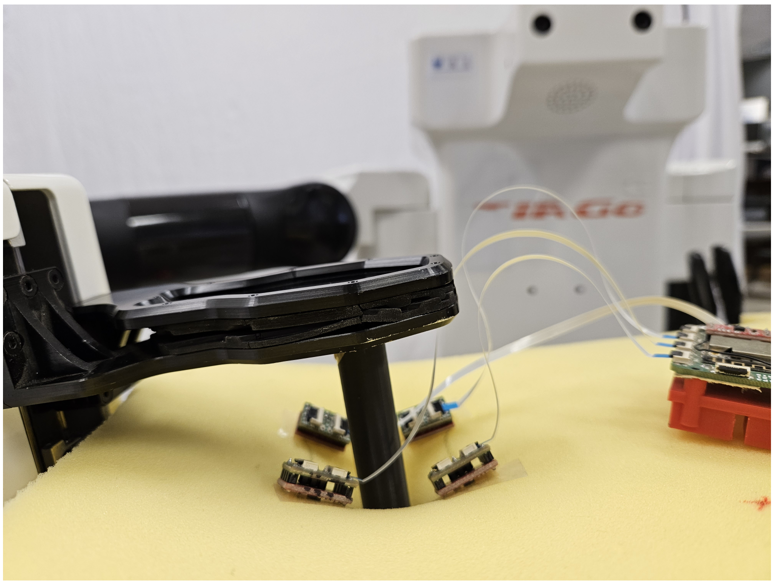


Figure 4.9: DeepContact phase scene during the force acting on the object

In this phase the overall behavior of the algorithm seems to be more precise, starting from the scattering of the x and y coordinates for the intersection point as shown in figure 4.11, in which is shown that the scattering area is now a line and the new positions are nearer to each other.

Nevertheless, the z coordinate appears to have linear crescent behavior, figure 4.10. The noise presence is still strong, but this time there is also a stabilization component that after the first 2 minutes of the collecting data, shows how the z component of the intersection point seems to be higher.

In the real-life scene, there were no changes that could justify this behavior, the sensors were stable and firm, yet the algorithm shows this behavior. As a consequence of error propagation, the minimum value for the z axis reached does not totally correspond to the real situation displacement.

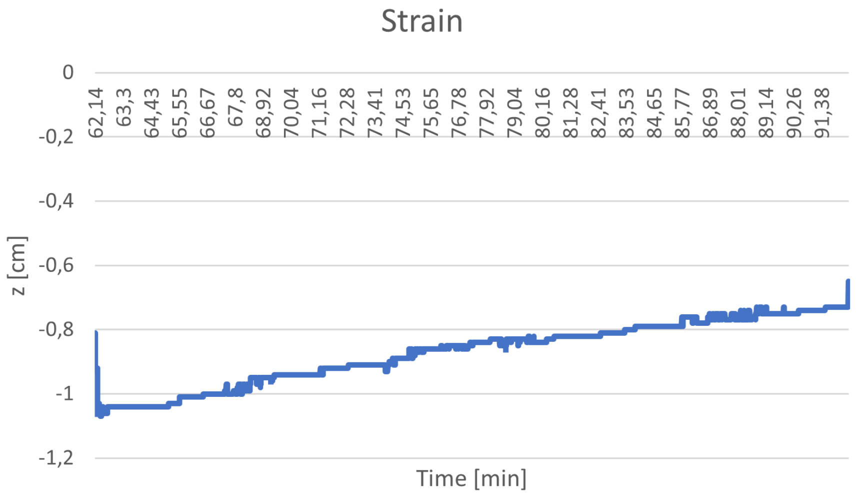


Figure 4.10: z displacement of the intersection point in DeepContact over time.

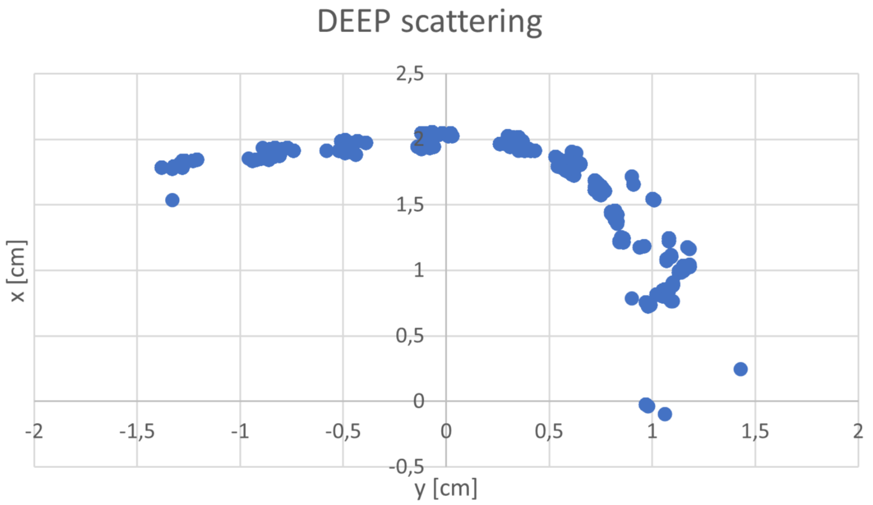


Figure 4.11: The scattering of x and y coordinates, here is possible to notice how the scattering is almost totally deleted.

The force on field, given by TIAGO's action is displayed in figure 4.12. It is possible to notice how the 2 forces present a totally different behavior. The trend of the inferred force is a direct consequence of the intersection point motion, since the z displacement is adjusting to be higher, the force calculation decreases.

The trend is coherent with the realistic scene. Starting from a lower level of force, since the robot was compensating for the gravity action thanks to the resistance of the object, a stronger push is displayed. To impose the desired displacement, a stronger force is needed, so that the limit behavior can be observed.

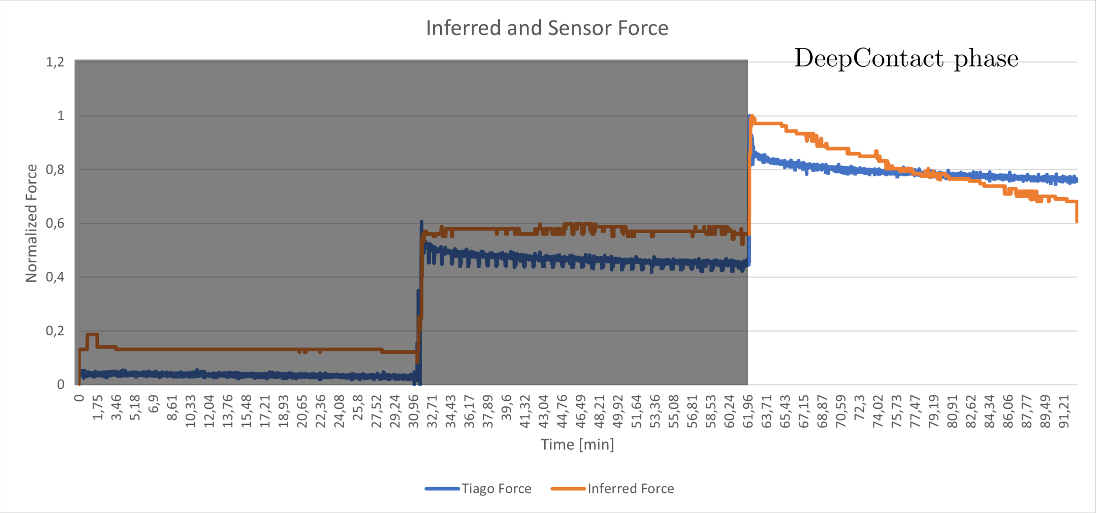


Figure 4.12: Last phase, the higher value of displacement is imposed, and the behavior of the two forces is different highlighting how the two different systems are detached and deserve separate types of noise compensation.

4.1.4 SOFA Simulation

In section 3.2, it is detailed how the simulation was constructed using a simplified cube geometry, adhering to the properties and specifications of polyurethane. Given the complexities faced in materializing the original concept — introducing rigid planes to represent the IMU positions and then adjusting their orientation based on sensor feedback — the simulation's performance is sub-optimal and requires further enhancements. These prospective improvements are elaborated upon in the 5 section.

Despite these shortcomings, the real-time demand is adequately met. The communication streamlines efficiently, courtesy of the SOFA framework, ensuring seamless transmission and instantaneous rendering of alterations.

The NoContact phase isn't the focal point here, as the simulation remains steady and the cube's morphology remains consistent in the absence of external forces. However, the Contact and DeepContact phases unveil interesting outcomes.

To interpret the discussion on results, it's imperative for readers to acquaint themselves with the intricacies of the simulation detailed in the section 3.2, especially the rationale behind the need to have a mixed material property on the simulation and the various adaptations needed to the model meshes.

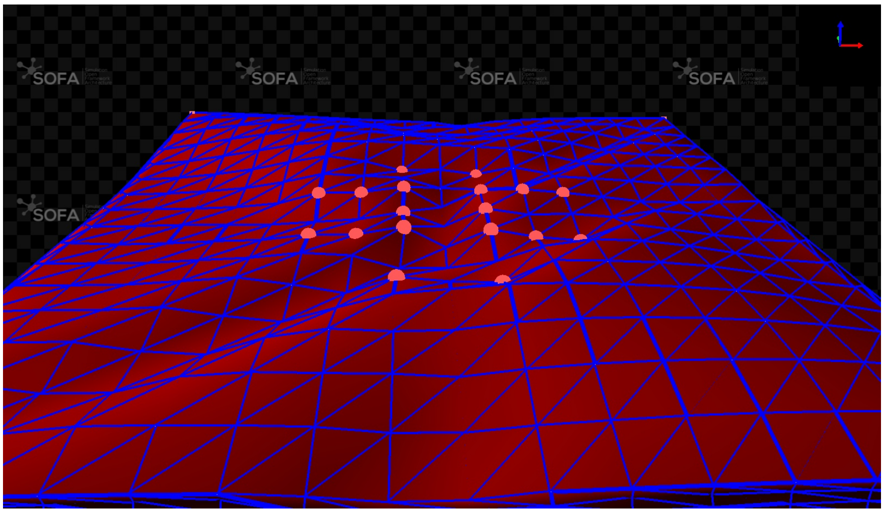


Figure 4.13: Simulation during the Contact phase. The central point shows the desired displacement but is noticeable as the rest of the surface is not homogeneous, and is deforming in a non-realistic way

Figures 4.13 and 4.14 depict the deformations attributable to the Contact phase. One salient observation is the pronounced depression at the cube's center

— a behavior aligning with the anticipated response for that particular area. However, this concordance doesn't extend to the entirety of the shape.

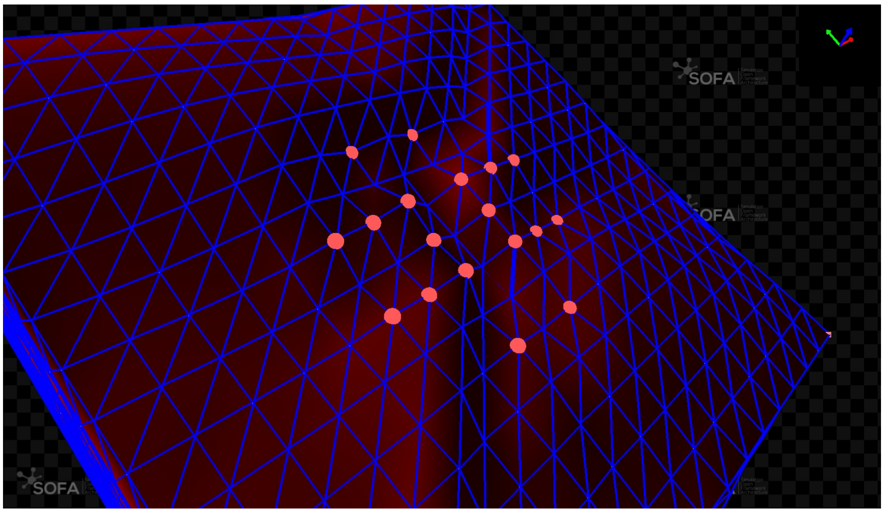


Figure 4.14: Side view of the deformation given in the Contact phase. With this perspective is obvious how the delimited planes do not have a planar relationship anymore. The vertex points are on the same plane but the area that should be constrained to them is not respecting the behavior.

Especially in figure 4.14, while the points symbolizing the plane vertices maintain their planar relationships, the encompassed area doesn't uphold this planar fidelity. Instead, it manifests varied depressions and strains, that deem the simulation to not optimal and not adequate for the desired implementation of the Digital Twin model.

The same argument is valid for the DeepContact phase, with the difference that in this case, since the strain imposed is higher, the planes' deformation are almost disappearing. Is interesting to notice that in figure 4.15 the strain seems to have a component on the x axis (red arrow in the reference frame), so it's possible to declare that if a more precise algorithm can be found, where all the components of the force can be taken into consideration, also the direction of the force can be efficiently displayed to the surgeon.

The main unrealistic characteristic of the simulation, which is noticeable already in the Contact phase, but more obvious in the DeepContact phase side view (figure 4.16), is that the planes, to respect the real behavior of a material when there is a depression in a point, all the areas affected by the deformation, follow it creating slopes. In the simulation instead, is possible to see how the planes create the depression by raising the side nearer to it, instead of following the point downward, the depression is created from the relative position between the center point and the sides of the planes adjacent to it, while the point by itself seems to be in the same position of the initial configuration. This indicates that the method of reconstructing the deformation using solely the orientation of the approximation planes around the moving point is not accurate. A larger area should be controlled to ensure a realistic representation.

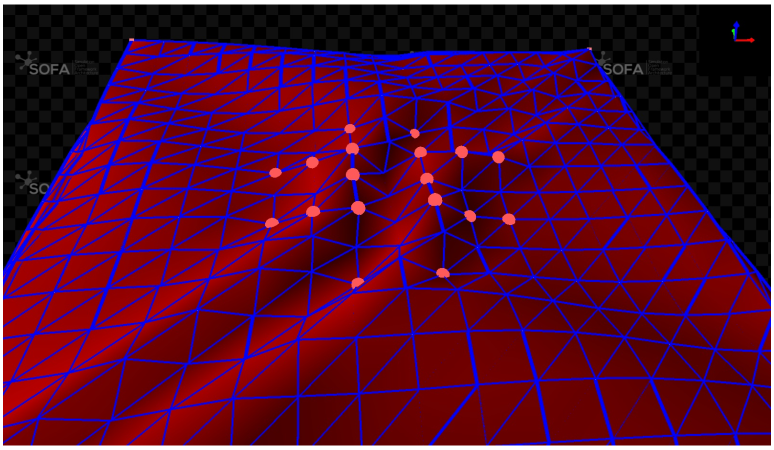


Figure 4.15: Simulation during the DeepContact phase. Noticing the reference frame of the simulation in the right-up corner.

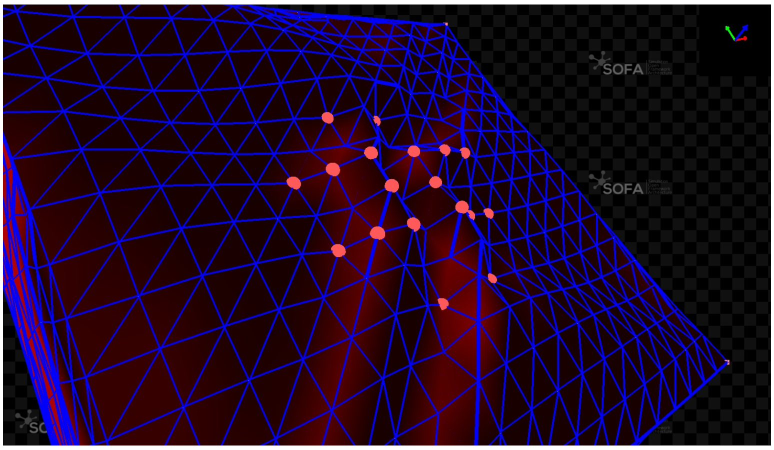


Figure 4.16: Side view of the deformation given in the DeepContact phase. Here the unrealistic behavior of the digital material is highlighted by the fact that the points near the deformation are not going in the negative direction, instead, they are raising to make the point look like is in a lower position.

Chapter 5

Future Work

As the experimentation progresses and evolves, several areas of further inquiry and development are becoming evident, offering exciting potential avenues for advancement in the research on constructing Digital Twins with open-source software.

The enhancements would focus on the simulation side, the model construction, and the algorithm improvements.

One immediate focus would be the refinement of the mesh models, particularly in addressing the challenges posed by mesh density. By leveraging advanced computational techniques or parallel processing, it will be possible to optimize the models without sacrificing accuracy or computational efficiency. Starting from improving the simplified geometry model to create a more accurate mesh connection, deleting the hollow space that impedes the model from behaving like a full object, and arriving at finding the right density of connection for complex models without slowing down the real-time communication.

A pivotal aspect of the proposed model is the integration of deformable elements with rigid planes. Therefore, a significant portion of our future research will be dedicated to the development of models with varying material properties across different meshes. This involves delving deeper into the intricacies of multi-material and multi-physics features in SOFA. The software already offers the possibility of creating a partial rigidification of some areas of the model, the time limit and the late discovery of the opportunity led to modifying the priority of this aspect.

To correctly develop the previous point, transitioning from the binary version of SOFA to its build version will be essential. Not only will this shift offer more flexibility, but it will also open doors to adapting codes from earlier versions of SOFA. Although these versions are better documented, they require careful integration to ensure compatibility and effectiveness. This change in version would open the opportunity to modify the structure of the scene in a deeper way, allowing the modification of standard tools.

The ultimate goal is to produce a model that mirrors real-world behavior as closely as possible. Thus, rectifying the current non-realistic behavior observed in our models will be a top priority. This will involve rigorous testing, feedback loops, and iterative refinements to ensure the model's responses align with real-world expectations, fixing as the first point the deformation logic that, as explained and depicted in the previous section, brings the planes to raise from the base level of the model, instead of pushing those sides down to create the depression required.

Recognizing the vast potential and increasing interest in open-source applications for Digital Twins, efforts will be made to foster a larger, more vibrant community around SOFA. This will include hosting workshops, webinars, and forums to encourage knowledge-sharing, collaboration, and joint problem-solving.

To facilitate the adoption and effective use of SOFA for similar applications, comprehensive documentation, tutorials, and case studies will be developed. This will ensure that new users or researchers can quickly grasp the platform's nuances and potential, accelerating their projects and fostering innovation.

In conclusion, while the current phase of our research has unveiled a series of challenges, it has also underscored the immense potential inherent in this domain. These challenges are not endpoints but rather signposts, guiding our journey towards creating robust, realistic, and effective Digital Twins using open-source platforms. The path forward is rife with opportunities, and with the continued support and collaboration of the broader community, we are optimistic about the groundbreaking advancements on the horizon.

Chapter 6

Conclusions

In conclusion, while the project necessitates further development and enhancements to be applicable in a real-world setting, it nonetheless explores possible solutions to the perception problem underlining limits and breakthroughs.

The aim of this research was to develop a Digital Twin, underpinned by IMU sensors, to bridge the perceptual gap in surgical robotics—particularly for surgical robots like da Vinci, which do not furnish surgeons with haptic feedback.

In facilitating the experimental phase of the objective, two facets were taken into consideration: the simulative Digital Twin (DT) side, and the actual object, which would serve as the Physical Twin. The tools selected for the DT side included:

- SOFA: An open-source framework predominantly known for tackling surgical simulation challenges, which typically offers a range of items and tools to simplify model building for both programmers and users requiring interaction.

- Python coding: This was utilized to manage the unpacking and communication between the microcontroller overseeing the sensors and the simulation, ensuring the scene was represented both on the simulation side and in conventional data gathering.

Conversely, the experimental procedure set up in the laboratory involved a disparate set of tools:

- Sensors communication: The sensors demanded a communication management design, constructed by the microcontroller and the programmer in the hardware setup. Utilizing the Arduino board and the sensors themselves enabled the acquisition of essential data to analyze and evaluate successes and challenges in the experiments.

- TIAGo++ robot: Given that the experimental phase required repetition and minimal divergence between each repetition, the assistance of the TIAGo robot was indispensable to consistently apply the same type and magnitude of force throughout the 30-minute data gathering period for each phase. The additional presence of a torque/force sensor allowed for the use of TIAGo's data as a reference for the force value derived from the code, which primarily relied on approximations emerging from the IMU references.

- Polyurethane cube: All experiments should have incorporated an elastic object representing the kidney phantom. Throughout the search for materials that progressively resembled the case scenario, the polyurethane cube was selected as a sufficiently satisfactory material. While it behaves differently from a kidney, it presents analogous challenges.

In the pursuit of suitable materials and methods, several challenges emerged, some of which remained unresolved due to the intricate nature of constructing such a comprehensive system. Consequently, the segment concerning future improvements warranted its own chapter to meticulously delineate the myriad issues awaiting address.

Given the unexpected challenges that culminated in the stringent limitations of the simulation, the primary objective of these experiments underwent a pivot. The revised goal sought to assess the efficacy of IMUs as real-time deformation detectors. While this latter objective was achieved and real-time communication seamlessly interfaced with the simulation, both the data derived from the sensors and the model-building techniques necessitate further refinement and temporal investment to optimally configure them. This optimization would enable their utilization as a foundational structure for other organs or varied pathological scenarios.

A notable point of contention against exclusively employing IMUs as a perception method arises from the detection error observed throughout the experiment. Specifically, a 60% error rate in deformation detection was recorded. Within this context, "error" denotes a comprehensive failure in deformation recognition, exemplified in figure 6.1, where the ACTIVE phase of the experimental procedure reveals a displacement degree unperceived by the IMUs.

The results unveiled a significant limitation of this perception method: its confined accuracy. The technique encounters difficulties in detecting displacements unless they transpire within close proximity to the IMU. Such restrictions could circumscribe the operational realm of robots and surgeons, converting this guidance system into a hindrance due to workspace obstructions. This intimates that an IMU might effectively serve as a displacement detector only when deformations transpire within a narrowly specified spatial vicinity around it.

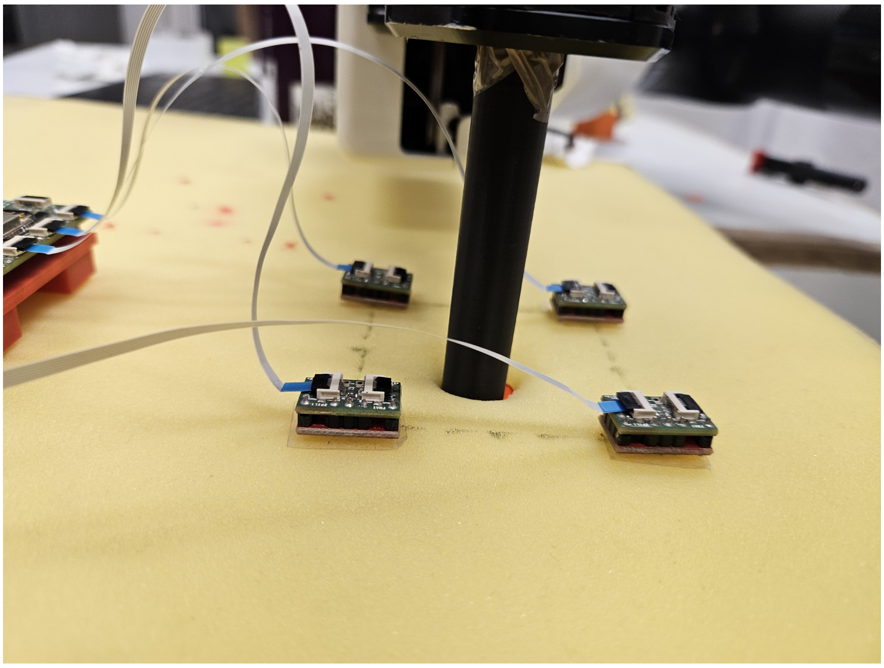


Figure 6.1: ACTIVE phase scene from a side-view perspective, highlighting how a minor displacement evades detection by the sensors.

Conversely, the instances that did yield detectable results were notably promising, and there is rationality to believe that the majority of the perception errors were attributed to a lack of data manipulation. Given the incongruence between available time and requisite effort, data neither underwent a filtering phase nor a calibration step, presenting a significant drawback for sensors like IMUs that are susceptible to storing errors and propagating them throughout the experimental period.

In realistic scenarios, where repositioning sensors affixed to an organ might pose a challenge, this emerges as a tangible concern. The exact dimension of proximity requisite for accurate measurements transcended the focus of this study, thereby providing a potential path for subsequent exploration.

In regards to the simulation side, the observed behavior was influenced by the belated discovery of means within the framework that would facilitate favorable results in the simulation. Moreover, the simulation necessitated a standalone study to comprehend all available tools in modeling and every physical property that might be represented on the scene. For instance, the force field chosen for the mesh model is presented: the mass-spring model was introduced in the state-of-the-art as the most promising modeling tool for the required behavior.

Given the encountered challenges, it's plausible that a Finite Element Method (FEM) model might have been a more apt choice, albeit introducing additional computational challenges. In summary, most problems encountered demand in-depth investigation and experimental processing before all components can be integrated into a functioning algorithm.

In retrospection, the study has shown that IMUs can be valuable for detecting real-time deformation but also highlighted the importance of careful data management, calibration, and choosing the right modeling tools for simulation. The findings provide a solid base for future research, raising new questions and pointing toward areas that need further exploration in the continuous effort to improve and enhance surgical robotics. As we move forward, combining these insights with a steadfast dedication to scientific accuracy and innovation will undoubtedly guide future advancements in this fascinating, multidisciplinary field.

References

Abayazid, M., Kato, T., Silverman, S.G. & Hata, N. (2018). Using needle orientation sensing as surrogate signal for respiratory motion estimation in percutaneous interventions. *International Journal of Computer Assisted Radiology and Surgery*, **13**, 125–133. 10

Amparore, D., Piramide, F., De Cillis, S. & et al. (2022). Robotic partial nephrectomy in 3d virtual reconstructions era: is the paradigm changed? *World J Urol*, **40**, 659–670. 7

Barricelli, B.R., Casiraghi, E. & Fogli, D. (2019). A survey on digital twin: Definitions, characteristics, applications, and design implications. *IEEE Access*, **7**. 13

Benway, B.M., Bhayani, S.B., Rogers, C.G., Dulabon, L.M., Patel, M.N., Lipkin, M., Wang, A.J. & Stifelman, M.D. (2009). Robot assisted partial nephrectomy versus laparoscopic partial nephrectomy for renal tumors: A multi-institutional analysis of perioperative outcomes. *Journal of Urology*, **182**, 866–873. 7

Bernhart, S., Harbour, E., Kranzinger, S., Jensen, U. & Finkenzeller, T. (2023). Wearable chest sensor for stride and respiration detection during running. *Sports Eng*, **26**. 10

Bielser, D., Glardon, P., Teschner, M. & Gross, M. (2004). A state machine for real-time cutting of tetrahedral meshes. *Graphical Models*, **66**, 398–417. 11

Cover, S., Ezquerra, N., O'Brien, J., Rowe, R., Gadacz, T. & Palm, E. (1993). Interactively deformable models for surgery simulation. *IEEE Computer Graphics and Applications*, **13**, 68–75. 11

El-Wajeh, Y.A.M., Hatton, P.V. & Lee, N.J. (2022). Unreal Engine 5 and immersive surgical training: translating advances in gaming technology into extended-reality surgical simulation training programmes. *British Journal of Surgery*, **109**, 470–471. 15

Enayati, N., De Momi, E. & Ferrigno, G. (2016). Haptics in robot-assisted surgery: Challenges and benefits. *IEEE Reviews in Biomedical Engineering*, **9**, 49–65. 3

Erol, T., Mendi, A. & Dogan, D. (2020). The digital twin revolution in healthcare. 1–7. 13

Freschi, C., Ferrari, V., Melfi, F., Ferrari, M., Mosca, F. & Cuschieri, A. (2013). Technical review of the da vinci surgical telemanipulator. *The International Journal of Medical Robotics and Computer Assisted Surgery*, **9**, 396–406. 7

Gettman, M.T., Blute, M.L., Chow, G.K., Neururer, R., Bartsch, G. & Peschel, R. (2004). Robotic-assisted laparoscopic partial nephrectomy: Technique and initial clinical experience with davinci robotic system. *Urology*, **64**, 914–918. 7

Hagn, U., Konietschke, R. & Tobergte, A. (2010). Dlr mirosurge: a versatile system for research in endoscopic telesurgery. *Int J CARS*, **5**, 183–193. 5

Hajtovic, S., Mogilner, A., Ard, J., Gautreaux, J., Britton, H., Fatterpekar, G., Young, M. & Placantonakis, D. (2020). Awake laser ablation for patients with tumors in eloquent brain areas: Operative technique and case series. *Cureus*, **12**. vii, 8

Heizmann, O., Zidowitz, S., Bourquain, H., Potthast, S., Peitgen, H.O., Oertli, D. & Kettelhack, C. (2010). Assessment of intraoperative liver deformation during hepatic resection: Prospective clinical study. *World Journal of Surgery*, **34**, 1887–1893. 11

Hostettler, A., Nicolau, S., Rémond, Y., Marescaux, J. & Soler, L. (2010). A real-time predictive simulation of abdominal viscera positions during quiet free breathing. *Progress in Biophysics and Molecular Biology*, **103**, 169–184. 8

Hughes, T., Rai, B., Madaan, S., Chedgy, E. & Somani, B. (2023). The availability, cost, limitations, learning curve and future of robotic systems in urology and prostate cancer surgery. *Journal of Clinical Medicine*, **12**. 6, 7

Joice, P., Hanna, G. & Cuschieri, A. (1998). Errors enacted during endoscopic surgery—a human reliability analysis. *Applied Ergonomics*, **29**, 409–414. 3

Khalifeh, A., Autorino, R., P. Hillyer, S., Laydner, H., Eyraud, R., Panumatrassamee, K., Long, J.A. & Kaouk, J.H. (2013). Comparative outcomes and assessment of trifecta in 500 robotic and laparoscopic partial nephrectomy cases: A single surgeon experience. *Journal of Urology*, **189**, 1236–1242. 7

Kühnapfel, U., Çakmak, H. & Maass, H. (2000). Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computer and Graphics*, **24**, 671–682. 3

Li, Y., Hannaford, B. & Rosen, J. (2019). Raven: Open surgical robotic platforms. 5

Ma, L. & Wangr, L. (2022). Imu measurement-based bending and deformation identification method for oil and gas pipelines. *CRPASE: Transactions of Mechanical Engineering*, **9**, 1–8. 10

Maciel, A., Sankaranarayanan, G., Halic, T., Arikatla, V.S., Lu, Z. & De., S. (2011). Surgical model-view-controller simulation software framework for local and collaborative applications. *International Journal of Computer Assisted Radiology and Surgery*, **6**, 457–471. 14

Meier, U., López, O., Monserrat, C., Juan, M. & Alcañiz, M. (2005). Real-time deformable models for surgery simulation: a survey. *Computer Methods and Programs in Biomedicine*, **77**, 183–197. 11

Miyasaka, M., Matheson, J., Lewis, A. & Hannaford, B. (2015). Measurement of the cable-pulley coulomb and viscous friction for a cable-driven surgical robotic system. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 804–810. vii, 5

Nakao, M. & Minato, K. (2010). Physics-based interactive volume manipulation for sharing surgical process. *IEEE Transactions on Information Technology in Biomedicine*, **14**, 809–816. 4

Ohuchida, K. (2020). Robotic surgery in gastrointestinal surgery. *Cyborg and Bionic Systems*, **2020**. vii, 10

Okamura, A., Matarić, M. & Christensen, H. (2010). Medical and healthcare robotics. *Robotics & Automation Magazine, IEEE*, **17**, 26 – 37. vii, 4

PAL Robotics (2018). ”tiago robot”. Accessed: September 26, 2023. 19

Pesenti, M., Invernizzi, G., Mazzella, J., Bocciolone, M., Pedrocchi, A. & Gandolla, M. (2023). Imu-based human activity recognition and payload classification for low-back exoskeletons. *Sci Rep*, **13**. 10

Picinbono, G., Delingette, H. & Ayache, N. (2000). Real-time large displacement elasticity for surgery simulation: Non-linear tensor-mass model. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2000*, 643–652, Springer Berlin Heidelberg. 11

Pierorazio, P.M., Patel, H.D., Feng, T., Yohannan, J., Hyams, E.S. & Allaf, M.E. (2011). Robotic-assisted versus traditional laparoscopic partial nephrectomy: Comparison of outcomes and evaluation of learning curve. *Urology*, **78**, 813–819. 7

Riedel, P., Riesner, M., Wendt, K. & Assmann, U. (2022). Data-driven digital twins in surgery utilizing augmented reality and machine learning. In *2022 IEEE International Conference on Communications Workshops (ICC Workshops)*, 580–585. 13

RYU, S., KITAGAWA, T., GOTO, K., OKAMOTO, A., MARUKUCHI, R., HARA, K., ITO, R. & NAKABAYASHI, Y. (2022). Intraoperative holographic guidance using virtual reality and mixed reality technology during laparoscopic colorectal cancer surgery. *Anticancer Research*, **42**, 4849–4856. vii, 9

Strbac, V., Vander Sloten, J. & Famaey, N. (2015). Analyzing the potential of gpgpus for real-time explicit finite element analysis of soft tissue deformation using cuda. *Finite Elements in Analysis and Design*, **105**, 79–89. 8

Systèmes, D. (2015). The living heart project. Accessed on 8 May 2023 from: https://www.3ds.com/products-services/simulia/solutions/life-sciences-healthcare/the-living-heart-project/. 13

Terzopoulos, D., Platt, J., Barr, A. & Fleischer, K. (1987). Elastically deformable models. *SIGGRAPH Comput. Graph.*, **21**, 205–214. 11

Unreal Engine (2023). Machine Learning (ML) Deformer. https://docs.unrealengine.com/5.0/en-US/using-the-machine-learning-deformer-in-unreal-engine/, accessed: August 2, 2023. 15

Vittori, G. (2014). Open versus robotic-assisted partial nephrectomy: a multicenter comparison study of perioperative results and complications. *World J Urol*, **32**, 287–293. 6, 7

Yates, D.R., Vaessen, C. & Roupret, M. (2011). From leonardo to da vinci: the history of robot-assisted surgery in urology. *BJU International*, **108**, 1708–1713. 7

Yip, M. & Das, N. (2017). Robot autonomy for surgery. vii, 6

Zhang, N. (2022). Application of a deformation measurement technology based on multi imu. In *Proceedings of the 2022 2nd International Conference on Control and Intelligent Robotics*, 381–385, Association for Computing Machinery. 10

Zhao, C.C., Shakir, N.A. & Zhao, L.C. (2021). The emerging role of robotics in upper and lower urinary tract reconstruction. *Curr Opin Urol*, **31**, 511–515. 3, 6