POLITECNICO DI MILANO
Corso di Laurea Magistrale in Ingegneria Biomedica
Dipartimento di Elettronica, Informazione e Bioingegneria

Towards the Automation of Robotic Ultrasound Scanning: Using Simulation to Tune and Test Optimal Robot Control during Tissue Interaction

Supervisor	Prof. Elena De Momi
Foreign Supervisor	Prof. Pietro Valdastri
Co-Supervisor	Bruno Scaglioni
Academic Tutor	Nils Marahrens
Academic Tutor	Andrea Mariani

Author
Lorenzo Minotti
Matr. 899623

Academic year 2018/2019

UNIVERSITY OF LEEDS

This thesis has been
carried out at the STORM Lab,
University of Leeds (UK)
and it is part of
Ultrasurge research project

Ultrasurge

Surgery Enabled By Ultrasonics

Acknowledgements

Sincere thanks
to my supervisors, who made this adventure feasible,
to my tutors, always on hand for precious advice,
to my family and friends, for their ever present support

Contents

LIST OF TABLES

LIST OF FIGURES

Abstract

In the last few decades, Robotic-Assisted Minimally Invasive Surgery (MIS) has come to the foreground in the surgical field. This thesis is part of a project that aims at automating the manipulation of an intracorporeal ultrasound probe with the da Vinci Surgical System, nowadays the most popular and widespread surgical robot. Ultrasounds allow to see in a non-invasive way beyond organs' surfaces and they are currently employed intra-operatively because of their ability to provide dynamic real-time imaging with an immediate impact on the surgical approach to predominantly oncological solid organ disease. Organs like liver, bile ducts, pancreas, adrenal glands and kidneys are particularly suited for this kind of imaging modality. Nowadays, this technique is utilized in surgical procedures such as nephrectomy and liver resection. Open problems that still prevent a widespread use of intra-operative ultrasound are reported to be the lack of an adequate training and the artefacts caused by deformations induced by the contact force between probe and tissues. In this sense, the automation of ultrasound scanning could offer several benefits: some of them typical of autonomy, such as a reduced surgeon's cognitive workload, others more specific of this technique, like the execution of more precise three-dimensional scans and the independence from operator. Above all, the most significant advantage is the possibility to finely control the force exerted by the ultrasound probe onto human tissues. This allows both to obtain higher quality images

by optimizing the acoustic coupling between US probe and organs, and to minimize harm to the patient during the scan.
The aim of this thesis is the implementation and evaluation within an adequate simulation environment of control algorithms to apply to the robot in such a way that a pre-defined contact force, whose importance has been highlighted above, can be exerted between ultrasound probe and bodies. The employment of simulation is motivated by the fact that it is a remarkable tool to easily carry out experiments in multiple conditions and to conveniently collect the data of interest.

For this reason, the first step in this thesis is the choice of a proper simulation framework. Starting from a comparison among currently available platforms, the choice of the most suitable one for the project fell on the Asynchronous Multi-Body Framework (AMBF). This simulation software offers the possibility to deal with soft bodies, in this way providing a physical scenario very close to the real one, and it furnishes a reliable ROS communication, which represents a convenient mean to control simulated elements from an application external to the framework. AMBF was originally not capable to return the contact force between robot and objects, which is of key importance for this project. Then, the simulation's physics solver was modified to expose this information. Finally, force control algorithms for the Patient Side Manipulator of the da Vinci have been implemented in order to evaluate its interaction with different kinds of rigid and soft bodies in AMBF. In particular, hybrid force control algorithms based on cartesian motion control have been developed and compared starting from a simpler motion control in joint space. In addition, tests have been conducted imposing different contact force goals and varying the physical properties of the interaction bodies (paying particular attention to the case of soft ones).

As result of this thesis, the selected simulation framework (AMBF) has been properly modified and it provides now all the kinematics and dynamics information required for tuning and testing the robot con-

trol during tissue interaction. Specifically, hybrid force-position controls of the da Vinci Patient Side Manipulator have been implemented and tested while interacting with other objects in simulation. These algorithms and the use of such framework, suitably modified, can pave the way towards more complex control strategies such as reinforcement learning-based control, adaptive with respect to tissue's stiffnesses. This can result in a more stable force control, necessary to move a probe around a soft surface with unknown stiffness like in ultrasound scanning.

Summary

Introduction

Minimally Invasive Surgery (MIS) has witnessed a rapid development since the late 80s. The main advancement is the use of very small incisions in operations, big enough just to allow specifically designed surgical instruments to penetrate into the patient's body. In this way, the large wounds required by open surgery are not necessary anymore in a wide range of surgical procedures. This involves several significant advantages for the patient: lower pain, faster recovery, lower scarring and minimization of the post-surgical complication's risk.
Besides the rapid growth of MIS, scientific and technological knowledge increased even faster. As a consequence, new devices, techniques and instruments were introduced in the medical field. New research areas arose, while others had consistent improvements that allowed to overcome previously unsolvable limitations. The introduction of robots in MIS brought to the birth of RAMIS (Robot Assisted MIS).
From the first model employed in MIS, the *Probot*, RAMIS has made great strides by improving and designing new robots over the years, from the *AESOP* to the *da Vinci* Surgical System. This one is, nowadays, the most widely spread surgical robot. The advantages offered by RAMIS regard both the surgeon and the patient. For example, the surgeon's hand-tremor is filtered out, enhanced visualization of the surgical area

is available and wide range of movements can be executed thanks to an improved dexterity.
The introduction of robot autonomy could further simplified the surgeon's work. According to the literature, surgical robot autonomy can be divided in multiple levels depending on the complexity of the task to be executed. Even in this case, there are several benefits, like a reduced workload for the surgeon and a higher precision in finalising the procedure.
This thesis is part of a project that aims at semi-autonomous intracorporeal ultrasound scanning in an unknown soft environment, like the human body. Within this project, the aim of this thesis is the implementation of control algorithms to apply to the robot in simulation in such a way that a pre-defined contact force is exerted between ultrasound probe and tissues. The employment of a simulation framework is motivated by the fact that simulation is a remarkable tool to easily carry out experiments in multiple conditions. In this way, in future steps of the project it will be possible to conveniently obtain the necessary amount of data for the development of modern machine learning algorithms, like reinforcement learning, fundamental for the implementation of autonomy. For this reason, before realizing the aforementioned control algorithms, the choice and analysis of a proper simulation environment is crucial.

State of the Art

The project aims at finding a way to enable a semi-autonomous ultrasound scan using the da Vinci Surgical System, and this thesis in particular employs a model of the *Patient Side Manipulator* (PSM). The *da Vinci* is a tele-operated system with three main components: a surgeon console, a vision cart and a patient-side robot. This system enormously contributed to the growth of RAMIS in the last two decades

and the creation of an open-source mechatronics system, the da Vinci Research Kit, further encouraged research on this platform.
Ultrasound imaging is already employed in laparoscopy. It relies on properties of acoustic physics to create images of body structures underneath patient's skin and offers several advantages with respect to other imaging modalities: for instance, it does not make use of ionizing radiations like CT (Computed Tomography) or it does not employ magnetic field as MRI (Magnetic Resonance Imaging). The use of ultrasounds in laparoscopy provides the surgeon with immediate feedback about changes that occur during operations. Ultrasound probes employed in MIS need to be specifically designed and they have to respect well-definite requirements, for example a small dimension in order to enter through the tiny incisions. While there is a wide range of commercially available ultrasound probes, research is still ongoing especially in the field of microultrasound. Transducers have been already proposed for the *da Vinci* robot, where for instance some works aimed at using elastography to overcome the lack of haptic feedback in RAMIS. This would be helpful in surgical procedures where the surgeon localizes the tumour through palpation.
One of the most important benefits of an autonomous ultrasound scan is the possibility of exerting a known control force. This is important in this kind of imaging to guarantee an optimal organ-transducer acoustic coupling and to limit the deformation of tissues induced by such force. In addition, the use of a robot allows to execute more precise scans, which are traditionally highly human-dependent. In fact, the quality of the scan depends on the user experience and ability, and a specific training is required.
Coming to autonomy, the project in which this thesis is included aims at investigating the usage of modern machine learning algorithms such as Reinforcement Learning to execute a scan in an unknown environment. Reinforcement Learning is a branch of Machine Learning widely used in robotics, and provides a set of tools for the design of hard-to-engineer

behaviours. It is a process of learning from trial and error by exploring the surrounding environment. In order to adopt such an approach, the use of a proper simulation framework is greatly helpful. Experiments can be easily executed in different environmental conditions and the various state-action pairs obtained can be evaluated and used to implement reinforcement learning algorithms. Moreover, simulation offers other benefits such as fast testing and rapid prototyping: for this reason, it is recognized as a valuable tool in research.
The simulation environment that is going to be chosen for this thesis, in which force control algorithms aimed at managing robot-bodies interaction are going to be implemented and tested, should also provide the data necessary to implement machine learning algorithms in future steps of the project. For all these reasons, it is necessary to find and test a simulation framework that fulfills certain requirements, the most relevant of which are the possibility to deal with soft bodies, since the robot interacts with organs and tissues which are non-rigid, and the integration with ROS, which is an intuitive and comfortable communication middleware to control the simulated robot with a code external to the framework.
In this chapter, a comparison between a set of popular simulation frameworks is conducted on the basis of the aforementioned specifications in order to select the most suitable one for the project. As a result of this research, the Asynchronous Multi-Body Framework (AMBF), a promising and recently developed simulation environment, has been identified as the most adequate.

Materials and Methods

The discussion in this Chapter starts with a brief description of the da Vinci Research Kit (dVRK), then moves to the analysis of AMBF

functionalities and utilities. In the end, it covers the types of control applied to the robot.
The dVRK is a telerobotics research platform consisting of electronics, firmware and software; it provides complete access to all levels of control via open-source electronics and software. The Patient Side Manipulator is the component employed in this thesis; for this reason a detailed description of its structure is reported, comprising of degrees of freedom, joint types, and features like the remote centre of motion. Its model is used in AMBF, the simulation environment previously selected and particularly suitable for the project. Nevertheless, as it is explained later in the discussion, the PSM has been modified in order to fit an ultrasound probe-like body onto the *main insertion link* termination. For this reason, forward and inverse kinamatics equations have been computed in order to develop force control algorithms based on position control in cartesian space. Forward kinematics equations have been computed through the Denavit-Hartenberg parameters available in dVRK User Guide and adapted to this particular case, inverse kinematics ones have been found starting from geometric considerations.
Within this framework, the AMBF format was created to overcome the limits of other currently adopted representation formats. It is particularly useful in simulation of surgical robots characterized by closed loop mechanisms and parallel linkages employed to maintain fixed the position of a remote centre of motion, as in the case of the da Vinci.
Moreover, AMBF has a stable, bidirectional ROS communication managed by a Python Client and relies on the Bullet physics engine, which defines the physics of simulation, initializes the world, and allows to treat both rigid and soft bodies. The creation of rigid elements is simplified thanks to a Blender-AMBF add-on that connects these two environments and allows to directly set the desired parameters to the body.
Differently from rigid bodies, soft objects presented problems in the generation through the add-on, in the documentation of their parameters and in the behaviour when imported in the scene, that was often un-

predictable. In a first phase, these problems had to be fixed, then the efforts focused on how to design control algorithms to manage the contact between bodies.
AMBF, however, does not provide any tool to read the force generated when two objects collide, thus making the implementation of such algorithms unfeasible. For this reason, different approaches have been evaluated in order to extract the contact force from simulation. In the end, the force computed in the joint between the *main insertion link* and the *tool roll link* of the PSM, computed in the Bullet Physics solver, has been employed because it represents a good approximation of the contact force exerted. Its value has been transferred onto ROS topics, in order to make it available in an external code.
A new tip for the *main insertion link* of the PSM, characterized by a shape similar to the one of an ultrasound probe, has been designed in Blender, imported in AMBF thanks to the add-on and substituted to the robot's tools. At this point, two kinds of force control have been considered: one based on joint motion control, the other on cartesian motion control. The first one is simple to implement because AMBF already provides functions to set and read robot's joints values. This represents an intermediate step in the development of the second one, more complete and requiring the above described computation of forward and inverse kinematics of the PSM configuration with the new tip.
At the end of this chapter, the test conducted to evaluate the outlined control algorithms applied to the robot in its interaction with different bodies are explained. In particular, three classes of tests are illustrated: force control based on position control in joint space, force control based on position control in cartesian space prescribing different patterns of target force (constant and sinusoidal), and force control based on position control in cartesian space evaluating the effect of a soft body's properties on simulation's outcomes. In each one of these tests, the most significant quantity is the contact force in Z direction of the cartesian reference frame obtained from simulation while a linear trajectory

is prescribed to the robot's end effector in X and Y coordinates.

Results

Experiments have been executed considering mainly three bodies: a rigid cylinder inclined in order to provide a certain steepness to the upper base, a soft sponge and a cloth constrained horizontally. After a first phase consisting in getting in touch with the specific object and reaching a target force, the various tests described at the end of the Materials and Methods Chapter have been conducted.

Test 1: Force Control Based on Position Control in Joint Space

Force read from simulation well tracked the desired one in the case of rigid body, but the absolute force error was much more relevant for soft ones. Such error was definitely too high, nevertheless this kind of control represented in fact just an intermediate step towards the development of a more complete algorithm.

Test 2: Force Control Based on Position Control in Cartesian Space

With respect to the previous one, this offers several advantages, like the possibility to actively control the position of the end effector in the operational space. As explained in the previous chapter, two groups of experiments have been conducted with different target force goals.

Test 2.1: Constant Target Force

It presents better outcomes in terms of absolute force error (between target and read force) with respect to control based on joint motion. The best performances have been obtained with the rigid cylinder, while soft

bodies were still subjected to irregular behaviours in the force read from simulation. In particular, it has been observed that the Z coordinate of the PSM's end effector presented sudden, unexpected variations in its curve influencing the behavior of the force read.

Test 2.2: Sinusoidal Target Force
These tests have been conducted to further investigate robot-bodies interaction in a scenario different from the ones previously described. In fact, X and Y coordinates have been maintained fixed, and the force in Z cartesian direction, instead of remaining constant, was made varying sinusoidally. In this case, a force error increasing with frequency has been obtained without significant differences for each one of the bodies analysed.

Test 3: Force Control Based on Position Control in Cartesian Space: Influence of Soft Bodies Properties on Force Behaviour
It was worth investigating how soft bodies influence simulation's outcomes. For this reason, the following two groups of tests have been conducted.

Test 3.1: Influence of Soft Bodies Parameters
Two parameters in particular, the linear stiffness coefficient and the pose matching coefficient, have been varied considering the soft sponge case. Results show that depending on their value, huge variations can occur in the contact force in Z cartesian direction. This is due to irregularities of body's structure, which unavoidably cause also fluctuations in the Z cartesian coordinate of the robot's end effector.

Test 3.2: Influence of Mesh Characteristics
Considering again the soft sponge case, the number of vertexes composing the mesh has been changed. Tests have been executed with an higher number of vertexes, making mesh's surface smoother. In this case, the

force read from simulation well tracks the desired one with a consequent lower absolute force error.

Conclusions

In this thesis, a framework suitable for the project has been identified: AMBF. First of all, a more thorough analysis than it had previously been done has been carried out. Then, this framework has been adequately modified and it provides now all the kinematics and dynamics information required for tuning and testing the robot control during tissue interaction, such as the contact force between robot and objects that was not previously available. In particular, hybrid force-position control algorithms for the da Vinci Patient Side Manipulator have been developed to evaluate the interaction between robot and bodies. It has been demonstrated how certain properties characterizing simulated bodies, especially soft ones, influence the force exerted by the robot on their surface.
The work carried out can pave the way towards more complex control strategies such as reinforcement learning-based control. In this sense, the next future development could be the implementation of reinforcement learning algorithms for force-position control, as they could enable an adaptive control to tissues characterized by different properties. In fact, reinforcement learning algorithms could tune the parameters of a controller in accordance for example to the stiffness of the environment. In such a way, a more stable force control would be obtained, accounting for the difficulty to control the ultrasound probe around a soft surface with unknown properties. In this case, in order to obtain the necessary amount of data to adopt this approach, various simulations with soft bodies characterized by different properties would be run.
In the future, in order to obtain more realistic simulations, organ-shaped

soft bodies could be generated as well as a comparison between simulated and real physical quantities can be conducted.

Sommario

Introduzione

La chirurgia mini-invasiva (MIS, dall'inglese *Minimally Invasive Surgery*) ha conosciuto uno sviluppo estremamente rapido sin dagli anni '80, quando fu impiegata per la prima volta. Il concetto che vi sta alla base è quello di operare incisioni il più ridotte possibile in modo tale da farvi penetrare strumenti chirurgici appositamente progettati e manipolati dal chirurgo all'esterno del corpo del paziente. In questo modo le più importanti ferite procurate da interventi di *open surgery* non furono più richieste in diverse operazioni chirurgiche; ciò comportò, e comporta tuttora, significativi vantaggi per il paziente. Per citarne alcuni: riduzione del dolore, guarigione più rapida, cicatrici di ridotta entità, minimizzazione del rischio di complicanze post-operatorie.
Mentre la chirurgia mini-invasiva cresceva rapidamente in popolarità, lo sviluppo tecnico-scientifico vedeva un progresso ancor maggiore. In conseguenza di ciò, nuovi dispositivi, tecniche e strumentazioni furono introdotti anche nel campo medicale. Fu così che da una parte venne favorita la nascita di nuove branche di ricerca, dall'altra fu reso possibile l'avanzamento scientifico in diverse aree prima frenate da limitazioni che sembravano insormontabili. L'introduzione di robot in chirurgia mini-invasiva portò alla nascita della RAMIS (*Robot-Assisted Minimally Invasive Surgery).*

Dal primo modello impiegato in MIS, il *Probot*, sono stati compiuti passi da gigante nel corso degli anni grazie all'introduzione di nuovi robot e al miglioramento di altri già presenti sulla scena, dall'*AESOP* al *da Vinci* Surgical System. Oggigiorno quest'ultimo è il robot chirurgico più popolare al mondo. I vantaggi che la RAMIS comporta riguardano entrambi chirurgo e paziente: viene filtrato il tremore della mano di chi opera, viene fornita una miglior visione dell'area in cui si esegue l'operazione, la precisione con cui si eseguono task chirurgici è maggiore rispetto a prima.
Oltre a questi, il lavoro del chirurgo è stato ulteriormente semplificato grazie all'introduzione di una certa autonomia del robot. Tale autonomia, secondo diverse fonti, può essere suddivisa in più livelli a seconda del grado di difficoltà del compito da eseguire. Anche in questo caso i vantaggi sono molteplici, fra cui una riduzione del carico di lavoro per il chirurgo stesso e una maggior precisione nell'eseguire il task chirurgico.
Questa tesi si colloca entro un progetto il cui obiettivo è quello di rendere possibile una scansione a ultrasuoni intra-corporea e semi-autonoma all'interno un ambiente sconosciuto costituito da elementi morbidi, o *soft bodies*, come è appunto il corpo umano. Entro tale progetto, lo scopo di questa tesi è l'implementazione di algoritmi di controllo da applicare al robot in simulazione in modo da mantenere una forza di contatto prestabilita tra sonda a ultrasuoni e tessuti. L'utilizzo di una piattaforma di simulazione è spiegato dal fatto che il suo impiego è universalmente riconosciuto come utile strumento per l'esecuzione di esperimenti in diverse condizioni. In questo modo, in futuri sviluppi del progetto, sarà possibile ottenere velocemente i dati di interesse per lo sviluppo di moderni algoritmi di machine learning come il reinforcement learning, fondamentali per conferire autonomia al robot. Per tutti questi motivi, la scelta di un opportuno framework di simulazione è fondamentale.

Stato dell'Arte

La trattazione in questo capitolo comincia con la descrizione del *da Vinci* Surgical System. Il progetto mira infatti ad implementare una scansione a ultrasuoni semi-automa con questo robot, e questa tesi, nel dettaglio, vede l'utilizzo di un modello del *Patient Side Manipulator* per eseguire gli esperimenti. Il *da Vinci* è un sistema di teleoperazione costituito da tre elementi principali: *surgeon console*, *vision cart* e *patient-side cart*. Tale robot è stato determinante nello sviluppo della chirurgia mini-invasiva e la creazione di un sistema meccatronico open-source, il *da Vinci Research Kit*, ne ha ulteriormente favorito la ricerca.
L'imaging a ultrasuoni è attualmente già impiegato in laparoscopia. Esso si basa sulla fisica del suono per creare immagini di strutture anatomiche e offre diversi vantaggi rispetto ad altre tecniche come la CT (Computed Tomography), che utilizza radiazioni ionizzanti, e la MRI (Magnetic Resonance Imaging), che impiega campi magnetici. L'uso degli ultrasuoni in laparoscopia fornisce un immediato feedback al chirurgo sugli eventuali cambiamenti che si possono verificare durante l'operazione. Le sonde ecografiche utilizzate in MIS devono rispettare ben determinate specifiche come la sterilizzabilità e le dimensioni ridotte in modo da passare per le piccole incisioni operate sul paziente. Attualmente una vasta gamma di sonde è reperibile a livello commerciale, tuttavia la ricerca rimane attiva per trovare nuove soluzioni e miglioramenti, specialmente nel campo dei microultrasuoni. Trasduttori a ultrasuoni sono già stati impiegati con il *da Vinci* in diversi lavori, alcuni di essi orientati ad esempio all'utilizzo dell'elastografia per compensare alla mancanza di un adeguato feedback aptico in RAMIS, fondamentale in operazioni chirurgiche che fanno della palpazione uno strumento essenziale per la localizzazione del tumore.
Uno dei vantaggi maggiormente importanti nell'eseguire una scansione semi-autonoma è dato dalla possibilità di esercitare una forza di contatto nota. Ciò è fondamentale in questo tipo di imaging per garantire un suf-

ficiente accoppiamento acustico tra tessuto e trasduttore e per limitare, ed eventualmente predire, la deformazione indotta da tale forza di contatto. L'uso di robot, inoltre, permette di eseguire scansioni più precise, dato che esse sono tradizionalmente operatore-dipendente. Il risultato di una scansione dipende infatti dall'esperienza ed abilità di chi maneggia la sonda, per cui è necessario un training specifico.
Per quanto riguarda l'autonomia, l'idea è quella di investigare l'impiego di moderni algoritmi di machine learning, come il reinforcement learning, per eseguire la scansione in un ambiente sconosciuto. Il reinforcement learning è un ramo del machine learning vastamente impiegato in robotica che fornisce una serie di strumenti per modellare situazioni complesse da descrivere analiticamente. E' un processo di apprendimento trial and error attraverso l'esplorazione dell'ambiente circostante. Per adottare questa metodologia l'utilizzo di un opportuno framework di simulazione è di grande aiuto, infatti in questo modo gli esperimenti possono essere facilmente eseguiti proponendo diverse condizioni dell'ambiente circostante e le varie coppie stato-azione così ottenute possono essere utilizzate per l'implementazione di algoritmi.
L'ambiente di simulazione scelto per la presente tesi, in cui verranno sviluppati e testati algoritmi di controllo atti a governare l'interazione tra robot e oggetti, deve anche fornire i dati necessari all'implemenazione di algoritmi di machine learning in future evoluzioni del progetto. In conseguenza di ciò, è necessario innanzitutto individuare un ambiente che soddisfi determinate specifiche, le più importanti delle quali sono la possibilità di interagire con corpi morbidi, dal momento che il robot opera tra organi e tessuti che sono non-rigidi, e l'integrazione con ROS (Robot Operating System), che rappresenta un mezzo semplice ed intuitivo per controllare il robot simulato con un codice esterno al framework.
In questo capitolo è riportato un confronto tra diverse piattaforme di simulazione, basandosi sui requisiti sopra menzionati, in modo da individuarne una particolarmente adatta alle esigenze del progetto. Il risultato di questa ricerca ha portato a ritenere che AMBF (*Asynchronous*

Multi-Body Framework), un promettente ambiente di simulazione recentemente sviluppato, fosse la scelta migliore.

Materiali e Metodi

Questo capitolo inizia con una breve descrizione del da Vinci Research Kit (dVRK), verte poi sull'analisi delle funzionalità e delle utilities di AMBF, infine spiega le tipologie di controllo applicate al PSM. Il dVRK è una piattaforma di ricerca telerobotica che comprende firmware ed hardware; fornisce inoltre completo accesso a tutti i livelli di controllo grazie ad elettronica e software open-source. Il Patient Side Manipulator è il componente utilizzato in questa tesi, per questo motivo ne è riportata una descrizione dettagliata riguardante struttura, gradi di libertà e tipi di giunto. Un suo modello è impiegato in AMBF, l'ambiente di simulazione scelto in precedenza e particolarmente adatto per le esigenze del progetto. Tuttavia, come spiegato più innanzi nella discussione, il PSM impiegato è stato modificato inserendo un corpo rigido somigliante ad una sonda a ultrasuoni al posto dei *tools* sulla terminazione del *main insertion link*. Per questo motivo, le equazioni relative alla cinematica diretta ed inversa sono state calcolate al fine di implementare un algoritmo di controllo della forza basato sul controllo della posizione nello spazio cartesiano. Le equazioni relative alla cinematica diretta sono state ottenute tramite l'utilizzo dei parametri di Denavit-Hartenberg disponibili sul manuale per l'utente del dVRK (adattati a questo caso particolare), quelle relative alla cinematica inversa sono state trovate partendo da considerazioni di carattere geometrico sulla struttura del robot. All'interno di tale framework è utilizzato un nuovo formato di rappresentazione, l'AMBF format, al fine di sopperire alle limitazioni di altri già esistenti. Esso è particolarmente indicato per simulazioni di robot chirurgici contraddistinti ad esempio da meccanismi ad anello

chiuso impiegati per mantenere fissa la posizione di elementi quali il *Remote Centre of Motion*, come nel caso del da Vinci.
Inoltre, AMBF fornisce una comunicazione stabile e bidirezionale tramite ROS gestita da un Python Client che ne semplifica l'utilizzo. Tale framework si fonda su Bullet come motore fisico: esso ne inizializza l'ambiente e permette di definire la fisica e il comportamento di corpi sia rigidi che morbidi. La generazione di oggetti rigidi è possibile grazie a un add-on che collega Blender (un software di modellazione 3D) ed AMBF e che permette di definire direttamente i parametri specifici di un corpo all'interno di Blender stesso.
Viceversa, i corpi morbidi presentavano inizialmente dei problemi nell' impiego dell'add-on, nella documentazione relativa ai loro parametri costitutivi e nel comportamento, spesso inatteso, una volta importati nella simulazione. Una prima fase di questo lavoro è stata dunque dedicata alla risoluzione di tali problematiche, successivamente si è proceduto con lo studio di algoritmi da applicare per la gestione del contatto fra corpi. AMBF tuttavia non fornisce alcun mezzo per conoscere la forza prodotta dall'interazione fra due elementi, ciò rende l'implementazione dei suddetti algoritmi irrealizzabile. Per questo motivo sono state valutate una serie di opzioni al fine di ricavare tale grandezza. Dopo un'attenta analisi si è optato per l'utilizzo della forze calcolata nel giunto che collega *main insertion link* e *tool roll link* del PSM, che ben approssima la forza di contatto realmente esercitata. Il suo valore è stato poi trasferito su ROS topics per renderlo disponibile anche in un codice esterno alla piattaforma.
A questo punto, un corpo interamente sviluppato in Blender ed importato in AMBF grazie all'add-on, caratterizzato da una forma simile a quella di una sonda a ultrasuoni, è stato sostituito ai tools del braccio in modo da poter scorrere più facilmente sulle superfici. Successivamente, due tipologie di controllo della forza sono state prese in considerazione: una basata sul controllo del movimento nello spazio dei giunti, l'altra nello spazio cartesiano. La prima è facilmente implementabile poichè

AMBF fornisce delle funzioni che permettono sia di imporre che di leggere i valori dei giunti del robot. Tale controllo costituisce uno step intermedio nello sviluppo della seconda tipologia, più complessa dal momento che richiede il calcolo della cinematica diretta ed inversa della nuova configurazione del PSM.
Nella parte finale di questo capitolo, sono spiegati i test condotti per valutare gli algoritmi implementati precedentemente. In particolare, sono presentate tre tipologie di esperimenti: controllo della forza basato su controllo del moto nello spazio dei giunti, controllo della forza basato su controllo del moto nello spazio cartesiano imponendo diversi andamenti della forza di contatto desiderata (costante e sinuoidale), e algoritmi di controllo della forza basati su controllo del moto nello spazio cartesiano atti a valutare gli effetti delle proprietà dei corpi morbidi sull'output della simulazione. In ognuno di questi test, la quantità fisica più significativa è la forza di contatto letta dalla simulazione in direzione Z nel sistema di riferimento cartesiano mentre viene eseguita una traiettoria lineare assegnando una determinata posizione nelle coordinate X ed Y all'end effector del robot.

Risultati

Gli esperimenti sono stati condotti utilizzando tre corpi in particolare: un cilindro rigido lievemente inclinato in modo da conferire una determinata pendenza alla base superiore, un corpo morbido la cui forma ricorda una spugna (*sponge*), e un tessuto (*cloth*) fissato orizzontalmente. Dopo una prima fase in cui l'end effector del robot entra in contatto con il corpo e raggiunge una determinata forza di contatto, gli algoritmi implementati sono valutati eseguendo i test illustrati alla fine del capitolo precedente.

Test 1: Controllo della Forza basato sul Controllo del Moto nello Spazio dei Giunti

La forza ricavata dalla simulazione segue fedelmente l'andamento di quella desiderata nel caso del corpo rigido, tuttavia l'errore tra le due è ben più rilevante nel caso dei corpi morbidi. Tale errore è decisamente troppo elevato, ciononostante questo tipo di controllo rappresenta di fatto solo uno step intermedio nell'implementazione di soluzioni più complesse.

Test 2: Controllo della Forza basato sul Controllo del Moto nello Spazio Cartesiano

Rispetto al caso precedente, questa metodologia offre diversi vantaggi, tra cui la possibilità di controllare attivamente la posizione dell'end effector entro lo spazio operativo. Come spiegato nel capitolo precedente, due classi di esperimenti sono stati eseguiti imponendo diversi andamenti della forza di contatto desiderata.

Test 2.1: Forza Target Costante

Rispetto al controllo del moto nello spazio dei giunti, i risultati in termini di errore tra forza desiderata e letta sono migliori. Questo errore si mantiene più contenuto nel caso del cilindro rigido, mentre i corpi morbidi presentano ancora una volta irregolarità nell'andamento della forza ottenuta dalla simulazione. In particolare si è osservato che la coordinata Z dell'end effector presentava variazioni improvvise nella sua curva che andavano ad influenzare il comportamento della forza.

Test 2.2: Forza Target Sinusoidale

Questi test sono stati eseguiti per indagare uno scenario differente da quelli già presentati. Infatti, le coordinate X ed Y sono state mantenute fisse, mentre la forza in direzione Z del sistema di riferimento della simulazione è stata fatta variare sinusoidalmente. In questo modo, si è ottenuto un errore tra forza assegnata e letta che aumenta incrementando

la frequenza per ognuno dei corpi considerati senza particolari differenze.

Test 3: Controllo della Forza basato sul Controllo del Moto nello Spazio Cartesiano: Influenza dei Parametri dei Corpi Morbidi sull'Andamento della Forza

Si è inoltre indagato come i corpi morbidi influenzassero l'output della simulazione, in particolare alla luce dei risultati precedentemente ottenuti. Per questo motivo sono stati eseguiti i seguenti gruppi di esperimenti.

Test 3.1: Influenza dei Parametri dei Corpi Morbidi

Due parametri in particolare, *linear stiffness coefficient* e *pose matching coefficient* sono stati analizzati considerando la *sponge* come corpo morbido. I risultati dei test hanno mostrato come a seconda del valore dei suddetti parametri possano verificarsi comportamenti radicalmente diversi nell'andamento della forza di contatto caratterizzati da importanti variazioni. Tali variazioni sono dovute ad irregolarità nella struttura del corpo, che inevitabilmente causano fluttuazioni nella coordinata Z dell'end effector.

Test 3.12: Influenza delle Caratteristiche della Mesh

Prendendo in considerazione ancora una volta il caso della *sponge*, si è deciso di cambiare il numero di vertici costituenti la mesh. Sono stati eseguiti esperimenti incrementandone il numero, rendendo in questo modo la superificie del corpo più uniforme. In questo caso, la forza letta (sempre in direzione Z del sistema di riferimento della simulazione) segue in modo fedele quella desiderata, con un conseguente minore errore tra le due.

Conclusioni

In questa tesi è stato individuato un framework adeguato per il progetto: AMBF. Innanzitutto, ne è stata condotta un'analisi approfondita come non ne erano mai state svolte, anche per il fatto che tale piattaforma è di recente sviluppo. Successivamente, tale ambiente è stato adeguatamente modificato ed ora è in grado di fornire tutti i dati cinematici e dinamici necessari per il tuning e il testing di algoritmi di controllo per il robot nella sua interazione con i tessuti: ne è un esempio la forza di contatto tra robot e corpi, inzialmente non disponibile. In particolare sono stati sviluppati algoritmi di controllo ibridi in forza e posizione per il Patient Side Manipulator in modo da valutarne l'interazione con elementi dell'ambiente circostante. E' stato dimostrato inoltre come certe proprietà caratterizzanti i corpi, specialmente quelli morbidi, influenzino la forza esercitata dal robot sulla loro superficie. Ora, il lavoro svolto apre la strada a tipologie di controllo più complesse basate ad esempio sul reinforcement learning. In questo senso, un possibile sviluppo futuro è l'impiego di tale disciplina per l'implementazione di tipologie di controllo adattive rispetto a tessuti con diverse proprietà. In questo caso, algoritmi di reinforcement learning potrebbero essere utilizzati per il tuning dei cofficienti di un controllore in base alle proprietà dell'ambiente. In questo modo si otterrebbe un controllo della forza più stabile che permetterebbe anche di gestire l'interazione della sonda a ultrasuoni con tessuti dalle caratteristiche sconosciute. Per adottare tale approccio, simulazioni con differenti corpi morbidi verrebbero eseguite al fine di ricavare i dati cineamtici e dinamici necessari.
In futuro, al fine di ottenere simulazioni più realistiche, corpi morbidi con una forma simile a quella di organi umani potranno essere generati, così come potrà essere condotto un confronto tra grandezze fisiche simulate e reali.

Chapter 1

Introduction

1.1 Introduction to the Thesis

In the last few decades, Minimally Invasive Surgery (MIS) has replaced traditional open surgery procedures in different kinds of operation. This technique revolutionized the surgical field by bringing several important advantages for the patient and new innovative ways of executing surgical procedures for the surgeon.
The introduction of robots in the medical field further contributed to enhance MIS popularity. In fact, these offered opportunities never seen before in the clinical scenario and their employment both greatly simplified surgeon's work and improved operation's outcomes. Different models of surgical robots have been proposed over the years; nowadays the most widely spread is the da Vinci Surgical System, a master-slave system characterized by a console and four robotic arms, two of which telemanipulated by the surgeon.
One of the most promising research areas is autonomy, that can be divided in different levels depending on the complexity of surgical task to be executed. Autonomy plays a key role in the project in which this thesis is included. The aim is to enable semi-autonomous intra-corporeal

ultrasound scans in an unknown soft environment through the manipulatinon of a robotic probe with the da Vinci.
Regarding ultrasounds, they are already employed in laparoscopy because they represent a valid mean to enhance surgeon's awareness on what is going on during operation thanks to their property to provide a real-time visual feedback.
Concerning autonomy, machine learning techniques like reinforcement learning will be utilized. These would greatly benefit from a tool able to easily produce data for the development of algorithms. To this purpose, an adequate simulation framework represents a valuable option.
In this thesis, the research and analysis of a simulation environment to be employed is carried out. In particular, such framework should satisfy well defined project's requirements. Then, force control algorithms to apply to the Patient Side Manipulator (PSM) of the da Vinci are implemented to evaluate the interaction between the robot and objects in the scene, especially soft ones.

1.2 Thesis Structure

Medical and Technical Background. An introduction on Minimally Invasive Surgery and Robotic-Assisted MIS is conducted. In particular, their basic concepts and the benefits they introduced for both the patient and the surgeon are analysed. Then, robot autonomy for surgery and its advantages and weaknesses are discussed. In the end, an overview on the context in which this thesis has been carried out is delineated, with a detailed explanation of the project in which this thesis is included.

State of the Art. The da Vinci Surgical System is described. Then, the discussion covers intra-operative ultrasound, with special care towards its laparoscopic utilization and the probes employed in MIS. Then, the role of contact force is explained, and a brief introduction on reinforcement learning and simulation is conducted. The thesis workflow is

also reported: identification of a proper simulation framework for the project, analysis of such platform, implementation and evaluation of force control algorithms. Finally, the first step of this work is carried out: a comparison between a series of simulation environments in order to choose the best one.

Materials and Methods. The da Vinci Research Kit is presented. Particular importance is given to the PSM's structure and to the forward and inverse kinematics computations of the modified model employed in this thesis. Then, a detailed analysis of the chosen framework is carried out, with special care towards the importing of soft and rigid bodies and the physics engine characteristics. The importance of the contact force is underlined again. This quantity is used, as final step, to implement force control algorithms to apply to a PSM of the da Vinci's. In the end, the tests conducted to evaluate the developed algorithms are presented.

Results. The results of control algorithms previously implemented and applied to the robot are explained and justified by means of plots representing the most important quantities, such as contact force and end effector's positions. The analysis covers examples of interaction with both rigid and soft bodies.

Conclusions. A recap of the main topics and of the most relevant results is conducted. In the end, a section is dedicated to future developments of this thesis' work.

Chapter 2

Medical and Technical Background

The field in which this thesis has been carried out is robotic surgery, which has witnessed a rapid growth in the last few decades. An overview of the concepts and history of Minimally Invasive Surgery (MIS) and Robotic-Assisted MIS is drawn in the first sections. Then, the analysis focuses on autonomy, which is important to fully understand topics treated in the next chapters. In the final part, the discussion covers a description of the project in which thesis is included, focusing on some of the requirements for this work.

2.1 Robotic-Assisted Minimally Invasive Surgery: an Overview

2.1.1 Minimally Invasive Surgery

Minimally Invasive Surgery (MIS) is a type of surgery whereby the surgical operation is executed through small incisions. It has been rapidly developed since the introduction of laparoscopic cholecystectomy in the late 80s and revolutionized the surgical world. In fact, surgery is invasive by its own nature, but the small incisions required in MIS allow to avoid the larger ones typical of open surgery, that leave significant wounds and take a long time to heal. Figure 2.1 shows the different impact between the two procedures. The basic concept of MIS is to insert surgical instruments into patient's body through small incisions. In such a way the operation can be executed while the surgeon's hands maneuver the surgical instruments from outside the patient's body [20].

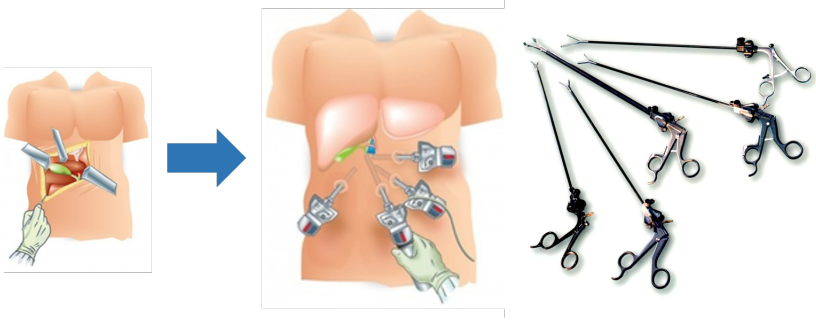


Figure 2.1: Difference between Open Surgery (smaller, on the left) and MIS (right). It is evident how the first one requires larger wounds with respect to the tiny incisions of the second. Besides the MIS example, typical surgical instruments for this kind of procedure

In order to perform these kind of operations, new, specific instruments like the laparascope or the endoscope camera were designed. These

have well definite characteristics: they have to be small enough to enter through the tiny incisions, provide a good handling and be easily sterilizable.
The advantages of MIS for the patient with respect to open surgery are many: less scarring, pain, bleeding and risk of infection thanks to the much smaller wounds, less damages to skin, muscles and tissues, lower risk of post-surgical complications, shorter hospitalization and recovery time. Nowadays, MIS has replaced many traditional procedures in open surgery and represents the standard in different fields such as visceral surgery, thoracic surgery and otolaryngology.
Besides its undoubted advantages, MIS presents however some limitations: the long time required to execute an operation and consequently the fatigue to which the surgeon is subjected, the limited range of movements provided by the surgical instruments, the reduced view provided by the endoscope, the surgeon's hand tremor, the lack of haptic feedback. Another problem regards the counter-intuitive motion of the instruments pivoting about the incision point that requires the use of mirrored motions with respect to that of the operating field, that represent an additional problem for the surgeon.

2.1.2 Robotic-Assisted MIS

The rapid technological and scientific innovation characterizing the last few decades allowed to introduce robots in the surgical field; this permitted to overcome most of MIS weaknesses [7]. In fact, surgical robots have been introduced to assist physicians performing various surgeries (especially in endoscopy and laparoscopy) by providing for instance camera guidance or filtering of the hand tremor; hence the name "Robotic-Assisted MIS" or RAMIS.
Surgical robots are robotic manipulators utilized to support surgeons during operations. Their employment dates back before the wide-spread application of MIS, with the first RAMIS operation performed in 1991

by a robotic system, the "Probot", during a prostatectomy. Then, new robots have been introduced over the years [50, 44].
The AESOP (figure 2.2), acronym of Automated Endoscopy System for Optimal Positioning, was the first robot approved by the FDA in 1990. Designed by Computer Motion, it was an endoscopic robot system for holding cameras in MIS and allowed to reduce problems like fatigue and hand tremor in traditional surgery. The latest generation included 7 degrees of freedom to mimic the human hand and its main applications were abdominal surgical procedures.

Figure 2.2: AESOP, the first surgical robot approved by the FDA in 1990

In history of RAMIS, the ZEUS and the da Vinci Surgical System are the two most famous telemanipulated robots.

1. **ZEUS**: the ZEUS system (figure 2.3), also designed by Computer Motion, had been approved in 2001 by the FDA and for the first time introduced the idea of telerobotics. It was a master-slave robot composed of two parts: a master console and three slave robotic arms (each one created by modifying an AESOP arm) attached directly to the surgical table. Two of them replicated surgeon's arms and had 6 degree of freedom, the third one held a camera that provided 3D images. The master console was where

the surgeon sat and manipulated the ergonomic controls. The main disadvantage of this robot was the large size, which limited space in operating room.

2. **da Vinci**. In 2003, after the merger of Computer Motion and Intuitive Surgical, ZEUS was phased out in favour of the da Vinci Surgical System, that had already been approved by FDA in 2000. This is a master-slave robotic system like Zeus, with a console and four robotic arms: two telemanipulated by the surgeon, two, passive, controlled by human assistants. More on this is explained in detail in the next chapters.

It is worth to point out that the da Vinci surgical system has become today the most widespread MIS robot around the world. Its introduction on the market represented a crucial moment in the evolution and development of robotic-assisted MIS, that was promoted as prevailing surgical practice.

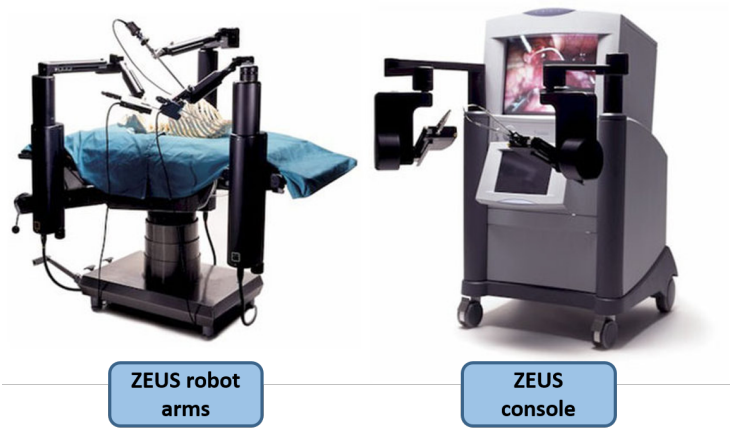


Figure 2.3: The ZEUS system with robot arms and console.

2.2 Robot Autonomy for Surgery

Since robots have been introduced in clinical practice, engineers and researchers tried to fully exploit their possibilities. Hence, another promising topic on which research focused is autonomy. Robot autonomy refers to the automation of surgical tasks, or subtasks, in such a way that they are completed by an intelligent robotic system [57].
A surgical procedure can be divided in multiple subtasks, some of them are relatively simple to be described and executed, and thus easier to automate, others are more complex and way more challenging to be committed to a robot.
Robots operating autonomously have to respect strict regulations, depending in particular on the level of autonomy. In [56], six different classes of autonomy, reported in table 2.1, have been identified. In

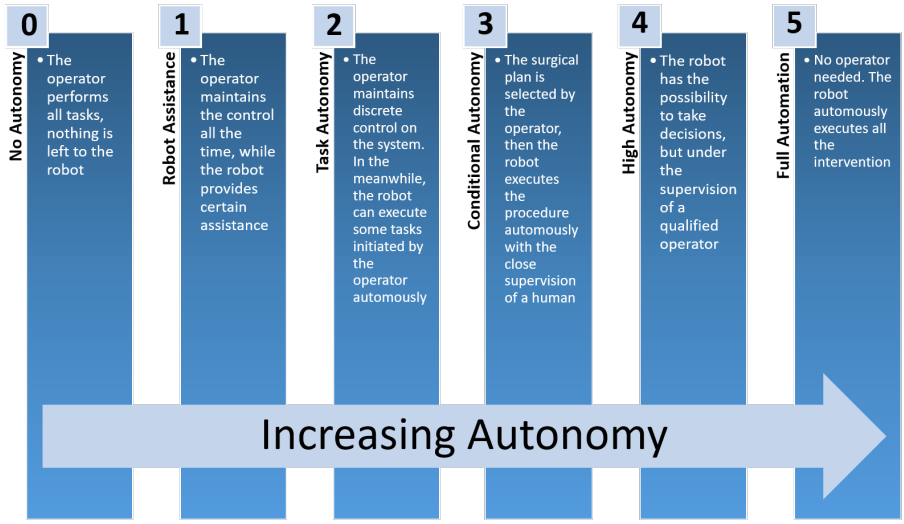


Table 2.1: Different levels of autonomy for surgical robots established in [56]

[12], examples of commercially available robotic systems employed in medicine and classified basing on this categorization are described. The da Vinci robot has just low-level robotic functions extended with capabilities of motion scaling and hand tremor filtering. The da Vinci Xi introduced other features like automated docking, instruments positioning and camera adjustments, nevertheless it remains in the first level of autonomy. The dVRK, vice versa, allows to achieve more complex autonomous tasks (autonomous trajectory planning, automated suturing, etc.) within the levels 2 and 3. Other types of robots, like the ROBODOC, that features autonomous image-to-robot registration and force-driven cutting, belong to higher levels of autonomy (3 in this case). The most advanced autonomous capabilities are integrated in the CyberKnife stereotactic radiosurgery (level 4), which combines image guidance with robot positioning.
Going higher with these levels, robots become more complex and sophisticated, and actually a robot belonging to the last class, level 5, has not been created yet. In this case a robotic system would completely supplant the surgeon and autonomously plan and execute the entire surgical procedure.

	ADVANTAGES	DRAWBACKS
HUMAN	Judgments capabilities	Prone to fatigue, tremors, inattention
	Easily trainable	Limited precision
	Easy communication with humans	Limited quantitative abilities
ROBOT	Good mechanical precision	No qualitative abilities
	Stable, not prone to fatigue	Unable of judgement
	Multi-modal sensory integration	Limited haptic sensation
	High dexterity	Expensive

Table 2.2: Main advantages and drawbacks of humans and robots in MIS

The benefits brought by surgical robots autonomy in MIS are well explained in [57], and the table 2.2 recaps the main advantages and drawbacks of robots with respect to a human in performing a surgical task. Besides the important improvements of reducing fatigue and hand tremors, robots also increase dexterity in MIS, enabling the surgeon to perform more precise and safe movements in a complex environment like patient's body. Nowadays, research is focused on solving issues like the inability to take fast and autonomous decisions in response to an unexpected event, the lack of judgement, the limited haptic feedback (that is problem in procedures that require palpation, for example), and the fact that they are often very expensive.

2.3 Project Description and Requirements for the Thesis

This thesis is part of a project related to the field of surgical robot autonomy. In particular, the aim is to explore and analyze new ways to automatize the manipulation of a robotic ultrasound probe with the da Vinci Surgical System.
The main goal is to enable semi-autonomous intra-operative ultrasound scanning in an unknown soft environment, like a human body, in order to provide the surgeons an enhanced visualization of the area of interest. Various aspects play into the successful realization of the project, for example the real-time processing of the acquired ultrasound data used for the 3D reconstruction of structures and organs, and the development of a customized ultrasound probe to integrate with the *da Vinci*.
This project is promising especially in intra-operative tumor localization: a probe semi-autonomously driven by a surgical robot could both reduce the surgeon's workload and at the same time improve its visualization of the surgical field by performing a 3D reconstruction of the

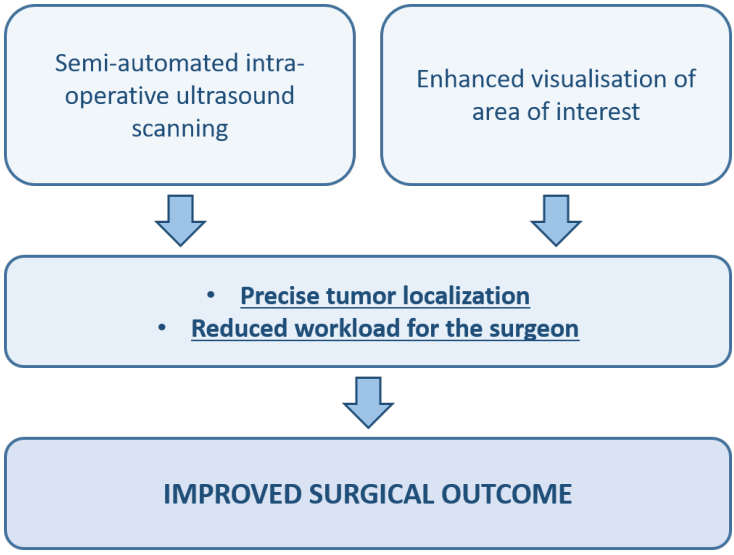


Figure 2.4: The final project goal is to improve the surgical outcome starting from the implementation of semi-automated intra-operative ultrasound scanning that provides an enhanced visualisation of the area of interest

acquired image data (figure 2.4). The precise localization of the tumor strongly affects the outcome of oncological surgical procedures: it is of primary importance to avoid healthy physical tissues and structures like nerves and blood vessels in order to limit the aftermath of the operation. Because in MIS even haptic feedback won't enable the surgeon to palpate like he does during open surgery, only two options are available to localize the tumor: the preoperative images coming from Magnetic Resonance, Computed Tomography or ultrasounds, which however do not provide live feedback; or intra-operative imaging techniques like ultrasounds, which give real-time images of the tumorous structure.
As reported in figure 2.5, research fields involved in this project are many, the most important of which are robot control, image reconstruction, signal processing and machine learning. For this reason there is a great need of multimodal data. For example reinforcement learning,

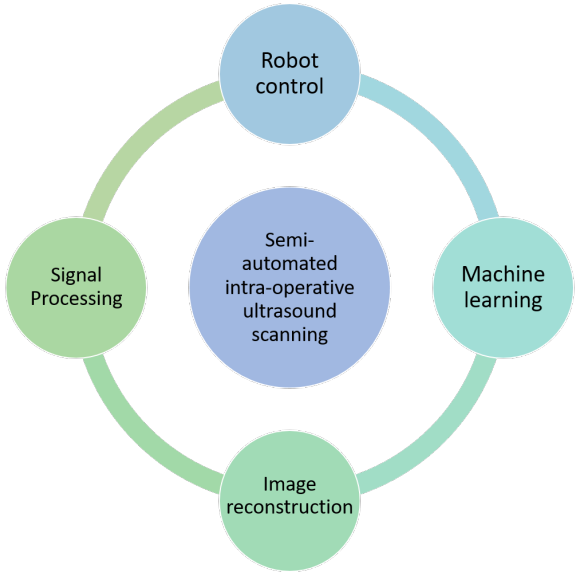


Figure 2.5: Main research fields involved in the project

which is a branch of machine learning, requires many experiments to explore different environmental conditions and obtain in this way the data and state-action pairs necessary to work. From this point of view, simulation serve as valuable tool: this is the reason why it has been used to test the control algorithms implemented in this thesis. A simulation framework suitable for this work should have well definite requirements. First of all, the robot has to explore the surrounding environment, for this reason the platform should support the interaction between the manipulator and both rigid and soft bodies. In order to study such interaction, physical quantities like the contact force, particularly important for the simulation of an ultrasound probe, and the position of the robot's end effector must be known.
The framework should offer a real-time communication thus allowing the user to control objects (and robots) in the simulated scene while receiving at the same time information on their current state. Moreover,

it is important to have the possibility to create and modify bodies, so that various scenarios and environments can be composed and the robot tested in different situations.
In this thesis, force control algorithms aimed at exerting a certain contact force between US probe and soft bodies are implemented and tested in a simulation environment characterized by the aforementioned specifications. In this way, in future steps of the project, various simulations with soft bodies of different stiffness and shape can be run in order to easily collect data for the implementation of reinforcement learning algorithms. These could in turn enable an adaptive force control to tissues characterized by different properties accounting also for the difficulty to control an ultrasound probe around a soft surface with unknown stiffness.

Chapter 3

State of the Art

In this chapter, the current state of the art of the methods and technologies of this thesis are discussed. First of all a brief introduction on the da Vinci robot and the dVRK. Subsequently, the discussion covers general features of ultrasound (US) imaging focusing on application of intra-corporeal US imaging in the context of robotic surgery. A basic explanation of reinforcement learning is also presented, necessary to introduce the use of simulation. To conclude, the final part of this section gives an overview on surgical robotics simulation and a comparison between simulation frameworks is exposed.

3.1 Da Vinci Robot: General Specifications

The da Vinci Surgical System is a platform for robot assisted minimally invasive surgery. It uses an advanced computerized control system between the surgeon and the surgical field, in this way enabling complex tissue manipulation while minimizing invasiveness [3]. It offers many advantages like the stabilization of surgeon's movements, the scaling of motion, which is peculiar for precision tasks, and the augmentation of surgeon's senses thanks to navigation assistance. The da Vinci Surgical System includes three different parts represented in figure 3.1: the patient-side cart, composed by three PSMs (Patient Side Manipulators) and an ECM (Endoscopic Camera Manipulator), the surgeon console, which has two MTMs (Master Tool Manipulators) and the vision cart.

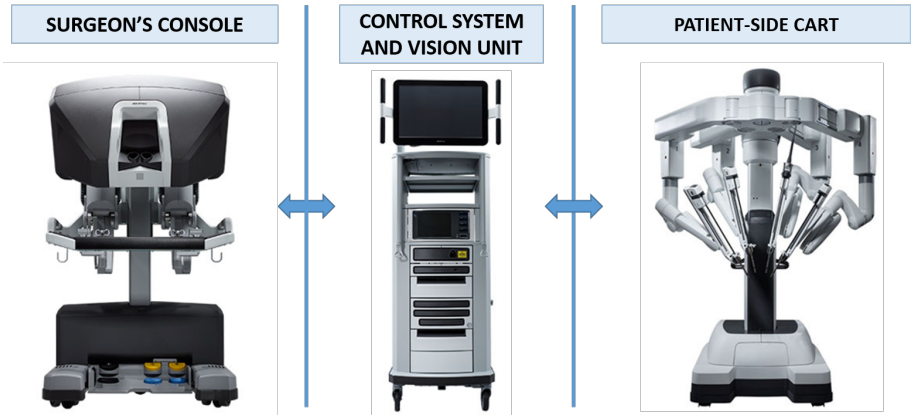


Figure 3.1: Basic structure of da Vinci system with surgeon's console, control system and vision unit, and patient-side cart

This system operates using the principle of teleoperation, where a computerized control system acts as intermediary in a master-slave architecture where the masters are the MTMs and the slaves the PSMs. In

fact, the surgeon is seated at a surgeon console from which they control motion of the surgical instruments at the patient side. The patient side manipulators can support a surgical instrument or a stereo endoscopic camera, and are mounted to the patient cart thanks to a setup structure. While operating, the surgeon remains outside the sterile field in which both the patient and the patient-side cart are located. The computerized control system streams the video-images from the endoscopic camera to the stereo viewer of the console and transmits the surgeon's hands motion from the master to the slave manipulators.
Currently, the da Vinci system is used in urologic surgery, general laparoscopy surgery, gynecologic surgery, trans-oral robotic surgery, thoracic surgery and in some kinds of cardiac surgery.
Moreover, an open-source mechatronics system consisting of electronics, firmware and software has been inroduced: the da Vinci Research Kit (dVRK). This has been introduced in order to encourage research to develop new solutions and tools for this surgical robot. More on this and on the PSMs structure, which is the component actually employed in this thesis, is discussed in detail in the Material and Methods chapter and in Appendix B.

3.2 Intra-operative Ultrasound

3.2.1 Ultrasound Imaging: an Overview

Ultrasound imaging is one of the most common type of imaging modalities used both in diagnostic and intra-operative cases. It is flexible and often provides an additional characterization of tissues, compared with other modalities such as conventional radiography or Computed Tomography (CT) [8].
Ultrasounds rely on properties of acoustic physics to create images of body structures underneath the surface of organ's surface. The transducer probe both produces the sound waves and receives the echoes

back. A voltage difference is applied to the extremities of one or more piezoelectric crystals, which causes a current to flow into the crystals themselves. If the current is alternate, the piezoelectric crystal's shape changes rapidly and continuously producing vibrations at high frequency that generate sound waves. These vibrations generate pulses at a predetermined frequency that propagate in human body at a velocity that depends on the type of soft tissues they go through. As a change of tissue impedance occurs, these waves are reflected back to the piezoelectric crystals of the transducer; higher is the difference in impedance, higher is the reflected ratio. The echoes received generate a signal which is sent back to the ultrasound scanner. After a further processing, the images are available for the operator. It is worth to note that the depth for typical ultrasound images in medicine is in a range from 2 to 16 cm and strictly depends on the transmitting frequency [5].
The reason why ultrasound imaging is so widely spread has to be found in the many advantages they offer with respect to other imaging modalities like CT or Magnetic Resonance Imaging: it does not make use of ionizing radiations, it is relatively low cost and it allows an easy integration with robotic systems [4, 8, 52].
On the other hand, ultrasounds imaging has some drawbacks, the most important of which are the impossibility to evaluate the internal structure of the tissues with high acoustical impedance (e.g. bone, air), the complex readability, the generated image difficult to interpret, the strong dependence on the operator skill level (for this reason an accurate training is necessary).

3.2.2 Laparoscopic Ultrasound

The use of ultrasounds (US) in laparoscopy allows the surgeon to overcome some typical issues of this technique and increases surgical safety. Ultrasounds allow to see beyond the surface of organs and their capability of real-time imaging provides surgeon with immediate feed-

back of changes that occur during operations. For example, they have an immediate impact on the surgical approach to predominantly oncological solid organ diseases [16].
Regarding laparoscopic US, the case of partial nephrectomy has been considered in different works [45, 41]. During this procedure, the surgeon needs to accurately locate the main veins and arteries before starting the resection of the tumor. While many of these structures, tumor included, are located by manual palpation during open surgery, in laparoscopy this cannot be done because the physician is manipulating physically separated surgical tools. As a consequence, haptic feedback is not available. Although this is a major issue in this kind of procedures, new approaches such as US elastography can overcome this problem providing a full image of tissue stiffness and viscosity, the properties that surgeons try to measure during manual palpation [40, 4].

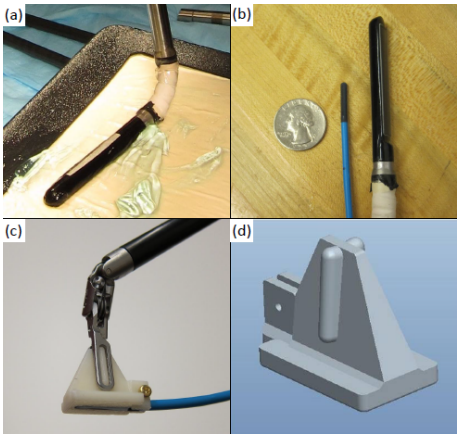


Figure 3.2: US probes used in elastography in [4]. a),b): an actuated US probe prototype from Intuitive Surgical Inc. c),d): micro-array US probe with ProGrasp Forceps and tool-to-probe adaptor for da Vinci

Other issues of traditional laparoscopic procedures have been solved by the introduction of surgical robotics: one above all, the counter-intuitive

motion of the instruments pivoting about the incision point that requires the use of mirrored motions with respect to that of the operating field. Nevertheless, a persisting problem is that during robotic procedures, laparoscopic US probes are usually controlled by the patient bedside surgical assistant instead of the operating surgeon. Additionally, the ultrasound instrument requires a dedicated laparoscopic port, meaning that another instrument must be removed from the surgical field.
To provide the operating surgeon with control of the transducer and to merge intra-operative ultrasound and robotic surgery, some research groups worked on the integration of US with da Vinci Surgical System [21, 39, 41, 4]. This integration allows the surgeon to execute US laparoscopic procedures comfortably from the console, with the high tool dexterity provided by the robot and with the high fidelity 3D visualization given by the system' stereo camera. Figure 3.2 represents examples of US laparoscopic probes, that will be explained in the next section.

3.2.3 Ultrasound Probes for Laparoscopy

Because of the use of US in laparoscopy, specific probes need to be employed. This class of probes should be small enough to enter the abdominal cavity through the trocar, so that the footprint of the probe should be limited to its diameter (around 10mm). The transducer array is placed directly on the target surface for imaging. The natural organ surface moisture is usually sufficient to permit adequate acoustic coupling, otherwise a specific US gel can be applied. The probe is positioned and oriented in the region of interest thanks to the view provided by the video laparoscope. It is of primary importance for the surgeon to have the view of the probe at all times in order to avoid injuries and collision with other instruments. However, the hand–eye coordination can be difficult and the orientation and following interpretation of the images can be complicated because the direction of the probe is seen only through the video laparoscope itself.

Trying to solve the reported issues, different transducers have been developed to be used with a typical da Vinci tool [21, 39]. A better design has been reached in this work [41], where an intraoperative "pick-up" ultrasound transducer offers greater flexibility and integration potential than other transducers. This has been designed to take full advantage of the dexterity of the da Vinci tools and to be easily picked up and manoeuvred by the da Vinci ProGrasp forceps. The transducer should be small enough to be moved inside the abdomen and to fit through a surgical incision, should have a consistent and self-aligning interface with the da Vinci grasper, should not have sharp or breakable components and should have the possibility to be sterilized. Research in US probes is still ongoing and particular interesting for this work could be the use of miniaturized high-frequency US probes, like the ones mentioned in [28]. These probes employ high-frequency (bigger than 20MHz), and therefore high resolution, ultrasounds called microultrasounds, that could represent a further step in the evolution and development of US laparoscopic procedures.

3.3 Autonomous Ultrasound Scanning in Medical Robotics

Different examples of autonomous ultrasound scanning are already present in literature; nevertheless most of them are represented by extra-corporeal applications. In fact, regarding the intra-corporeal case, autonomous robotic ultrasound scanning is a recent area of research. As an example, the work [31], previously cited as integration of an ultrasound probe with the da Vinci, employs this robot to execute a partial nephrectomy procedure by means of autonomous robotic control, guidance through ultrasound imaging and waterjet.
As aforementioned, much more extra-corporeal applications can be found both in research and clinical practice. In [53], a system for autonomous

robotic US acquisitions aimed at the quantitative assessment of patients' vessel diameter for abdominal aortic aneurysms is reported. The work [26] proposes a learning-based controller to enable autonomous US scanning execution through a KUKA Light-Weight Robotic arm; in [25] a robotic system for autonomous liver screening is investigated. All these extra-corporeal applications mainly aim at overcoming human-dependency in US scanning that typically worsens the final quality of images.

3.4 Role of Contact Force

To achieve an optimal acoustic coupling between ultrasound transducer and organs and to obtain good visibility of the target anatomy, it is necessary to apply a certain force on the contact surface [54]. While the induced deformation can be exploited in certain clinical applications to infer mechanical properties of the imaged tissue, in other cases it does not allow to reconstruct the exact localization of anatomies of interest, such as tumor masses. 3D ultrasound, which refers specifically to the volume rendering of US data, is affected by this issue. During 3D ultrasound acquisitions, the induced deformations do not remain constant across the individual 2D images of an US sweep, thus worsening the volume reconstruction. To overcome these problems, different image reconstruction algorithms have been proposed to make the deformation homogeneous along the entire US volume. In addition, in extra-corporeal ultrasound, robotic platforms have been introduced in order to maintain a nearly constant pressure (and so deformation too) on the anatomical region of interest, which simplifies the volume reconstruction. Most of these platforms include force sensing. Recently, a new framework has been introduced to estimate the contact forces even of a da Vinci Surgical based on joint encoders [55].
The quantity that most effectively describes robot-organ interaction is the contact force at the manipulator end effector [1]. Generally, inter-

action control strategies can be grouped in two categories: indirect and direct force control. While the former achieves force control via motion control without explicit closure of a force feedback loop, the latter offers the possibility of controlling the contact force to a desired value, thanks to the closure of a force feedback loop. The realization of a force control scheme can be done with the closure of an outer force regulation feedback loop generating the control input for the motion control scheme.
Indirect force control comprises impedance control and admittance control, which can be found in many robots employed in medicine. Admittance control is used, for instance, in a work reported in [13], where a NeuroMate robot is guided in cooperative control mode for removal of cranial bone on the skull base. Here, readings of the force sensor are coupled back to the control of the robot through an admittance control law; depending on the direction of the force applied on the tool, the robot moves to the desired direction with a velocity proportional to the force.
Another example of admittance control is reported in [51], where the technical principles of a force controller robot designed as generic platform for applications with human interaction (called HapticMaster) are described. Its control diagram is represented in figure 3.3. First of all, the interaction force is measured. Then, a virtual model calculates position, velocity, and acceleration (PVA) which an object touched in space would obtain as a result of this force. The virtual model specifies the space where the object is and the properties of the object itself. Then, a PID servo loop controls the robot to the commanded PVA vector.
In cases where the robot is in contact with a surface, like organs, force control has to be able to control the amount of force the robot exerts. If both the advantages of position and force control want to be exploit, hybrid architectures are employed [33].
In this thesis, force controls characterized by an inner position loop have been implemented following the diagram in figure 3.4. The motion control block has been implemented initially, as an intermediate step, by in

the joint space, then through a more complete control in the cartesian one.

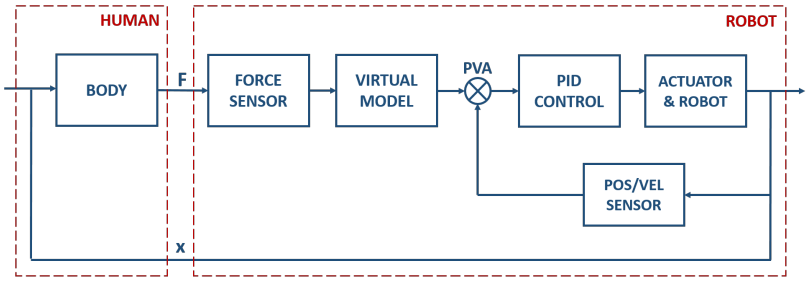


Figure 3.3: Control scheme of an admittance control algorithm as reported in [51]. This is an example of indirect force control.

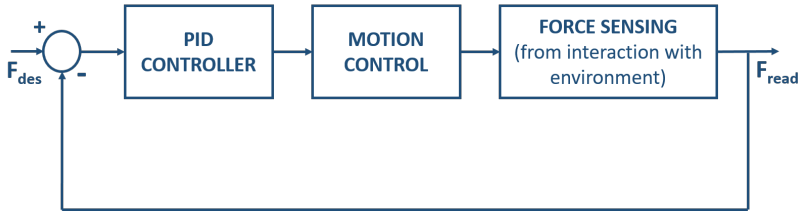


Figure 3.4: General control scheme employed in this thesis. A more detailed analysis on this structure is illustrated in the following chapters.

3.5 Introduction to Reinforcement Learning and Simulation

3.5.1 Reinforcement Learning Overview

As previously stated, one of the main goals of the project in which this thesis is included is to execute an ultrasound scan of the organ or tissue of interest in a semi-autonomous way through one of the da Vinci

arms. In order to achieve this, a promising approach is the employment of reinforcement learning algorithms to control the PSM while executing the scan, for this reason a brief introduction on its concepts and utilization is conducted.
Reinforcement learning is a branch of machine learning widely used in robotics offering a framework for the design of sophisticated and hard-to-engineer behaviors [17]. Its ultimate goal is to give to robots the ability to learn, improve, adapt and reproduce tasks with dynamically changing constraints. Basically, it is the process of learning from trial and error by exploring the surrounding environment. The goal is specified by a reward function which acts as positive reinforcement or negative punishment depending on the performance of the robot with respect to the desired target [19].

Figure 3.5: Reinforcement Learning working principle

The working principle of reinforcement learning is well represented in figure 3.5, in which environment comprises functions that transform an action taken in the current state into the next state and a reward; agent represents functions that transform the new state and reward into the next action. It can be thought in terms of state-action pairs that occur one after the other. This field of study offers the opportunity to teach

new skills to robots like learning new tasks which the human teacher cannot directly program, learning to achieve optimization goals of difficult problems that have no analytic solution, or learning to adapt a skill to a previously unseen version of a task. Reinforcement learning is now gaining more importance in surgical robotics and research is directed at automating subtasks that are part of procedures such as suturing, tumor resection and bone cutting.
Manipulation of tissues is an example of how reinforcement learning is employed in surgery, as illustrated in [43] and [27]. The rapid growth of this discipline brought also to the development of dedicated reinforcement learning environments, for example the dVRL [35] has been specifically designed for the da Vinci Research Kit.
One of the main limitations of this kind of approach, being trial and error, is the huge amount of data needed produced by multiple experiments with different state-action pairs. This is one of the reasons for which simulations has been used in this thesis.

3.6 Simulation

Simulation is a scientific tool well established in robotics that complements more traditional experimental approaches. The possibility to produce training data for offline learning or to speed up interactive learning allows to overcome the scarcity of information that would result from experiments done in real world only [34]. In particular, some learning approaches that make use of the interaction of robot with its environment (like reinforcement learning) require a large number of such interactions before they produce meaningful results. In addition to be time-consuming, experiments in real world could be expensive or dangerous both for the robot and the operator. For all these reasons, the use of a proper simulation framework is recommended. Figure 3.6 shows even other benefits offered by this approach, such as the possibility to discover unforeseen issues, to test different conditions simulating differ-

ent environments, and to rapidly prototyping a new robot.

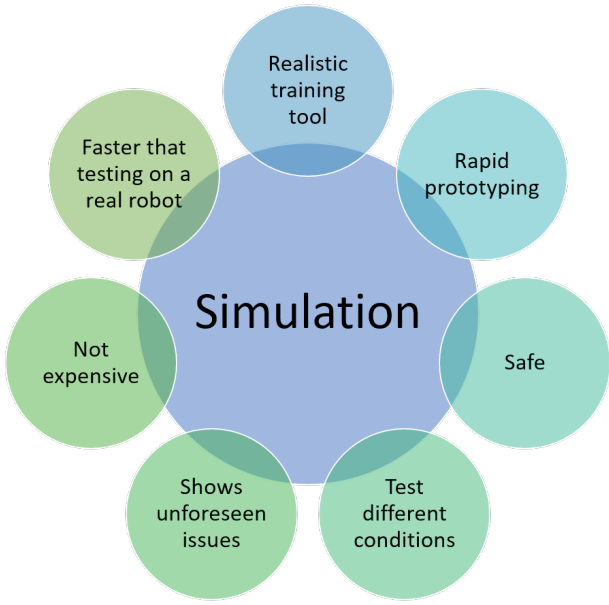


Figure 3.6: Advantages offered by simulation

In medical field, simulation is used especially for surgeons' training in order to allow them to perform increasingly complex minimally invasive procedures while enhancing patient safety. An example of action that requires training is an ultrasound scan: the quality of the resulting image strongly depends on the user experience, so a proper training can improve procedure's outcome.
Real-time simulations methods have been also proposed to create simulated US images starting from Computed Tomography scans or Magnetic Resonance images [42, 2]. On the other hand, common industrial simulators like V-REP and Gazebo are currently used to test and develop control algorithms to be applied to robots in surgical procedures like robotic laparoscopy.

In this thesis, the simulation framework to be chosen has to fulfil the requirements already explained in chapter 2, from which two main conditions have been established: the integration with ROS (Robot Operating System, see Appendix A) and the possibility to deal with soft bodies. The choice of ROS is due to many reasons: it is flexible, there are packages for a huge number of robotics applications, it offers the possibility to control multiple robots, it allows an easy communication between C++ and Python nodes. It also provides real-time data on robot's current state. The second specification refers to the fact that in a surgical scenario the tools obviously interact with soft tissues and organs and not with rigid bodies only. In order to choose the proper environment, a comparison between some of the currently available frameworks has been carried out.

3.7 Thesis: Requested Passages

Before achieving the goal of implementing force control algorithms aimed at managing the interaction between robot and bodies, the passages represented in 3.7 has to be followed. First of all, a simulation environment suitable for the project has to be found. As aforementioned, it should allow to deal with both rigid and soft bodies and provide a stable ROS integration. Once identified the adequate simulation environment, a careful analysis on its properties and functionalities is required in order to exploit at best its possibilities. In particular, special care has to be dedicated towards those quantities useful for the development of control algorithms for the robot, like the contact force. Then, as final goal, force control algorithms for the Patient Side Manipulator of the da Vinci can be implemented and tested in the chosen platform in order to evaluate its interaction with bodies characterized by different properties while exerting a well definite contact force between probe and objects. If results are satisfactory, in the future various simulations with soft bodies characterized by different properties can be run in order to conveniently

collect data for the implementation of reinforcement learning algorithms aimed at enabling an adaptive force control to organs characterized by different or unknown properties.

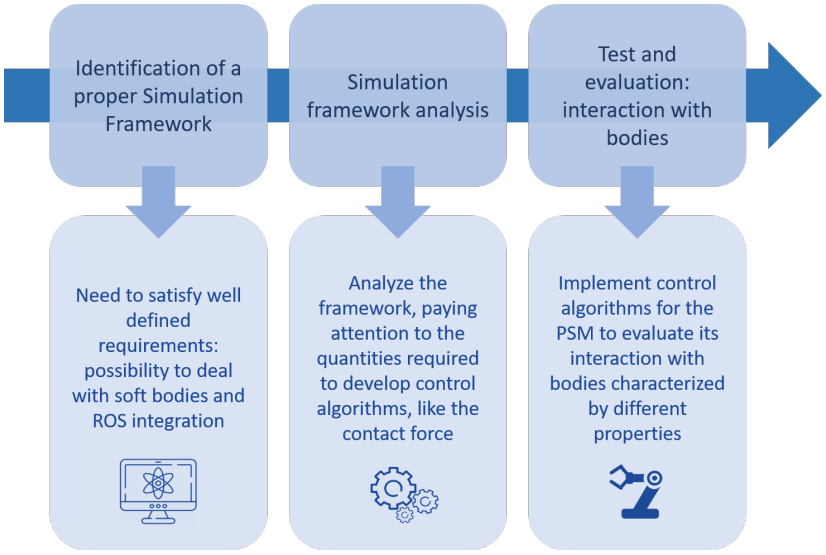


Figure 3.7: Steps required in the thesis, from the identification of a framework suitable for the project to the implementation of control algorithms to evaluate interaction between bodies

3.8 Identification of a Proper Simulation Framework

Choosing a suitable simulator is important, as different simulators environments offer different performances, model detail and built-in features [14]. Other characteristics that vary between frameworks are for example the graphical and physical simulation accuracy that depends on the physics engine employed [36], and the programming languages

support. Some simulators have been chosen among the ones currently available in order to find the one that better satisfies the aforementioned requirements. Basing on these criteria, a first research gave the following candidates: Gazebo, V-REP, ARGoS, Unity, AMBF and Webots.

1. **Webots**: Free and open-source 3D robot simulator. It relies on ODE (Open Dynamics Engine) physics engine, which allows the simulation to deal with rigid bodies only.

2. **V-REP**: Robot simulator with an integrated robot environment [37]. it has a ROS integration and relies on Bullet physics engine (which theoretically features rigid and soft body dynamics). However, on closer inspection, it actually does not allow to treat soft bodies as reported on its website [9], though there are plans to integrate them in the future.

3. **ARGoS**: Mostly used for multi-robot systems, it has an integration with ROS and offers the possibility to use different physics engines (even if characterized by very limited capabilities) to optimize the usage of modern multi-core CPUs [29, 30]. It does not support mesh importing and includes only a small library of robots.

4. **Unity**: Used to create simulations and in virtual and augmented reality. An integration with the dVRK has been found here [32]. ROS comunication is based on C sharp and is unidirectional (it is possible to set the positions of the robot, for example, but is not possible to read them).

5. **AMBF**: Asynchronous Multi-Body Framework. Provides a stable ROS integration and it relies on an extended version of Bullet physics engine, which allow to deal with both rigid and soft bodies.

6. **Gazebo**: It is designed to accurately reproduce the dynamic environments a robot may encounter [18] and it is fully integrated with

ROS. It offers the possibility to use different physics engines such as ODE, Bullet, DART and Simbody. However, it does not deal with soft bodies, even if in literature it is possible to find attempts to extend the Gazebo simulator to surgical robotics [38] or simply to perform a softbody simulation [48].

Webots, V-REP and Gazebo were immediately discarded because they do not allow to deal with soft bodies. ARGoS is recommended for simulations with multi-robot systems or for large swarms of robots of different types, which however do not represent a topic of this thesis. Unity allows to model soft bodies, but in this case the problem is the ROS communication, which is based on C sharp and still work in progress. The only option remained was AMBF. It is a recently developed framework which actually satisfies the two main requirements: soft bodies simulations are possible and a stable ROS communication is provided. Because it has all the desired characteristics, AMBF has been chosen as simulation framework for this thesis. Table 3.1 briefly recaps the main features of the options analysed in this paragraph.

Framework	**Integration with ROS**	**Compatible Physics engines**	**Soft bodies**
V-REP	Available	Vortex, ODE, Bullet, Newton	No
Webots	Available	ODE	No
ARGoS	Available	4 physics engines with limited capabilities	-
Unity	ROS *sharp*, still work in progress	Bullet	Yes
Gazebo	Fully integrated	Bullet, DART, ODE, Simbody	No
AMBF	Available	Bullet	Yes

Table 3.1: Simulation Frameworks: a recap. In the end, the Asynchronous Multi-Body Framework is the only one that features a stable ROS communication and a physics engine that allow to deal with soft bodies too

Chapter 4

Materials and Methods

The materials and methods used for the realization of the current thesis are now analyzed. First of all the da Vinci Research Kit is explained, with a particular focus on the PSM, the main component employed in the experiments executed in the simulation environment. Then, the Asynchronous Multi-Body Framework (AMBF) is described in detail. Specifically, close attention is dedicated to the efforts taken to extract the interaction force out of the simulation and to how this information has been employed in the following steps. Finally, two different kinds of force control algorithms are explained, one based on joint motion control, the other on cartesian motion control, with the first used as starting point for the implementation of the second one.

4.1 Introduction to the dVRK and Description of the Modified PSM Model Employed in this Thesis

4.1.0.1 dVRK

The da Vinci Research Kit is a telerobotics research platform that consists of electronics, firmware and software and provides complete access to all levels of control via open-source electronics and software [15]. It is used, for example, for the design of new control methods, including autonomous or semi-autonomous control. The mechanical hardware is obtained from retired first generation da Vinci robots and consists of the following components: two Master Tool Manipulators, two Patient Side Manipulators, a High Resolution Stereo Viewer (HRSV), a footpedal tray, and documentation (e.g., wiring diagrams, connector pinouts, kinematic parameters); it does not include the passive Setup Joints that support the PSMs, the Endoscopic Camera Manipulator (ECM), the stereo endoscope, control electronics and software.
The development of a common, open-source electronics platform for the research community allowed to overcome problems given by the fact that electronics and software are either proprietary or not included. An FPGA enables a centralized computation and distributed I/O architecture in which all control computations are implemented in a familiar development environment (Linux PC). Moreover, the FPGA provides direct, low-latency, interfaces between the high-speed (400 Mbits/sec) serial network, which is an IEEE-1394a (FireWire), and the I/O hardware. The software is implemented in a component-based C++ framework, with ROS interfaces to facilitate integration with other systems and software packages [15].

4.1.0.2 Patient Side Manipulator

The Patient Side Manipulators (appendix B) are the slaves teleoperated by the Master Tool Manipulators. Each one is a 7 degrees of freedom actuated arm with joint sensors and actuators for control purposes [49]. All joints are of revolute type, except for the In/Out (or Insertion) that is prismatic. In figure 4.1 are represented the robot's joints, while in table 4.1 their name and type are specified.

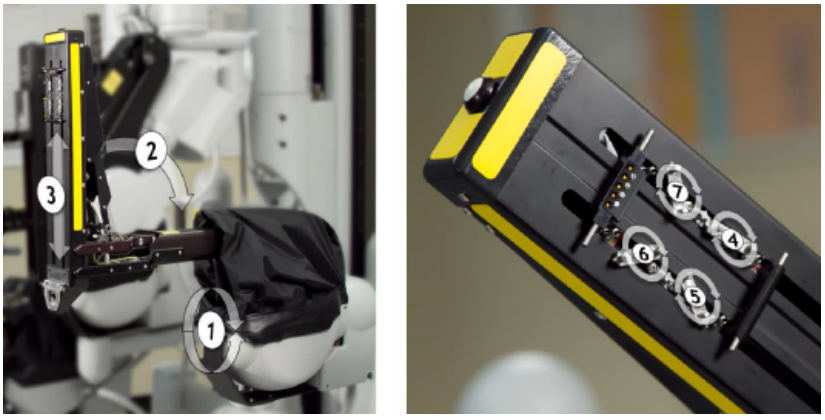


Figure 4.1: The 7 joints of Patient Side Manipulator. Image taken from the dVRK User Guide [49]

Joint number	Type	Name
1	Revolute	Outer Yaw
2	Revolute	Outer Pitch
3	Prismatic	In/out Insertion
4	Revolute	Outer Roll
5	Revolute	Wrist Pitch
6	Revolute	Wrist Yaw
7	Revolute	End effector

Table 4.1: Type and name of the PSM joints as represented in figure 4.1

Its most important characteristic is to move a surgical instrument about a spatially fixed fulcrum invariant with respect to the configuration of the joints: the Remote Centre of Motion (RCM). This represents the insertion point of the surgical instrument in the patient's body. RCM plays an important role in minimally invasive surgery because it allows to avoid additional trauma other than the incision itself. The da Vinci PSMs is based on a double parallelogram mechanism, which is the most widely spread among surgical robots for MIS[46].
A detailed analysis on the PSM structure is available in Appendix B.

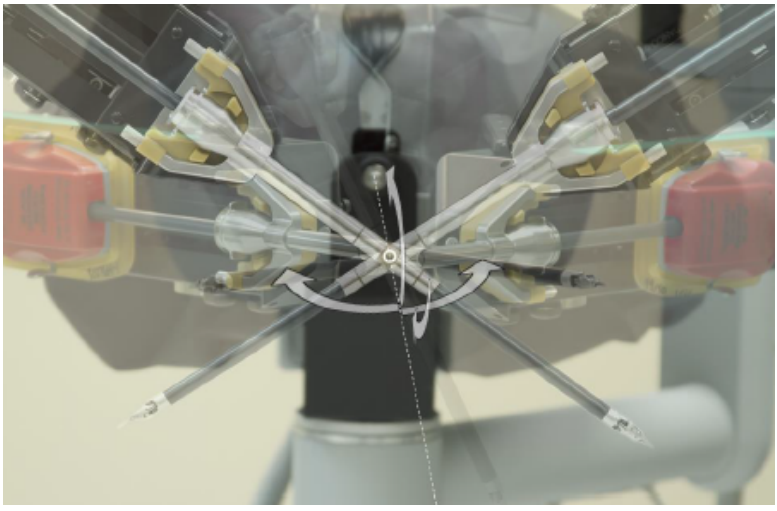


Figure 4.2: Remote Centre of Motion of PSM about which movements are executed. Image taken from [49]

4.1.1 Modified PSM Model Employed in the Thesis: Forward and Inverse Kinematics Computations

The Patient Side Manipulator model employed in this thesis slightly differs from the original one. Indeed, the tools have been substituted with an ultrasound probe-like rigid body attached directly to the *main*

insertion link. This modification is well explained in section 4.2.3.4.
With this new configuration of the robot, the computation of forward and inverse kinematics was required to apply the chosen control algorithms, especially in case of a cartesian motion control.
Inverse kinematics has been computed starting from geometrical considerations, while forward kinematics through the Denavit-Hartenberg parameters of the PSM. These parameters can be found in [49] and are reported in Appendix B. Nevertheless, they are a bit different from the original ones because of the aforementioned differences in the arm's structure. Table 4.2 reports the DH parameters of interest.

joint	a	α	D	θ
1	0	$\frac{\pi}{2}$	0	$q_1 + \frac{\pi}{2}$
2	0	$-\frac{\pi}{2}$	0	$q_2 - \frac{\pi}{2}$
3	0	$\frac{\pi}{2}$	$q_3 - l_{RCC} + l_{TOOL} + \delta l$	0

Table 4.2: DH parameters of the modified PSM

The small δl accounts for the origin of the end effector frame positioned in the centre of the ball. In fact, referring to the figure in Appendix B, this shift, whose value is 4 cm, has to be applied in the Z direction of the O_4 reference frame. In this way

$$l_{TOOLnew} = l_{TOOL} + \delta l$$

For a better understanding of the PSM base reference frame, see the figure B.1 in Appendix B or the figure 4.3. From these considerations it is easy to derive the equations for forward kinematics.

$$P^0 = T_3^0 P^3 \tag{4.1}$$

Once T_3^0 is computed and simplified, by setting the coordinates of P^3

equal to 0, the forward kinematics equations are obtained:

$$X_0 = sin(q_1)cos(q_2)(l_{TOOLnew} - l_{RCC} + q_3) \tag{4.2}$$

$$Y_0 = -sin(q_2)(l_{TOOLnew} - l_{RCC} + q_3) \tag{4.3}$$

$$Z_0 = -cos(q_1)cos(q_2)(l_{TOOLnew} - l_{RCC} + q_3) \tag{4.4}$$

For the inverse kinematics computation, refer to the figure 4.3. The quantities of interest can be computed as:

$$r = \sqrt{x^2 + y^2 + z^2} \tag{4.5}$$

$$sin(q_1) = \frac{X}{\sqrt{X^2 + Z^2}} \tag{4.6}$$

$$sin(-q_2) = \frac{Y}{r} \tag{4.7}$$

So, from 4.6 and 4.7:

$$q_1 = asin(\frac{X}{\sqrt{X^2 + Z^2}}) \tag{4.8}$$

$$q_2 = -asin(\frac{Y}{r}) \tag{4.9}$$

$$q_3 = l_{RCC} - l_{TOOLnew} + r \tag{4.10}$$

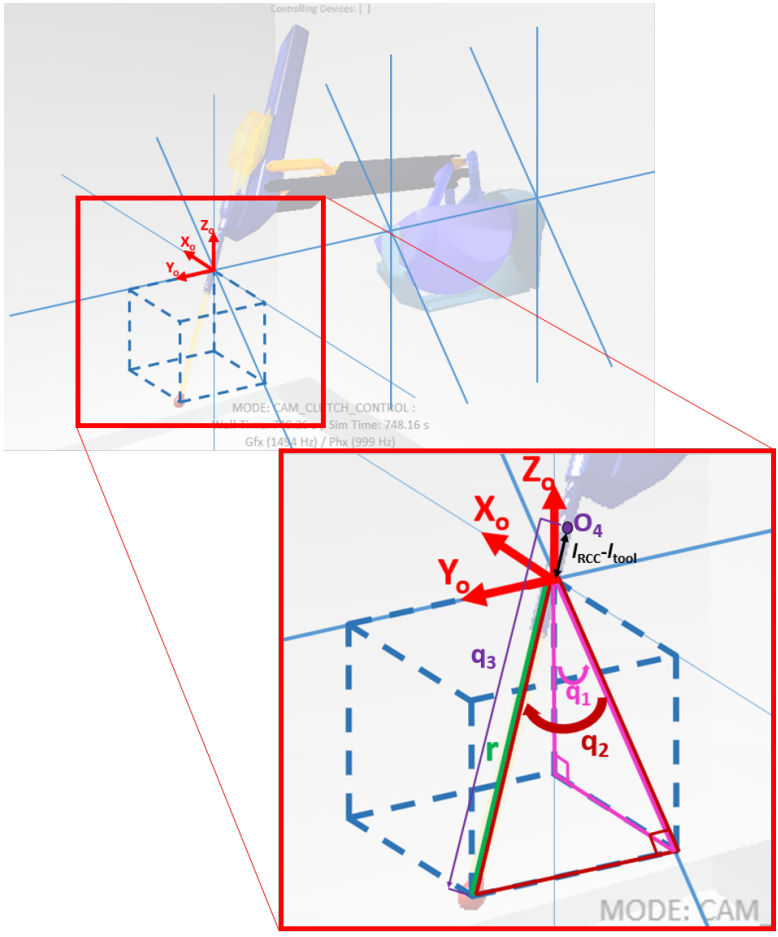


Figure 4.3: Geometric variables employed in the computation of inverse kinematics. It is possible to observe that the PSM tools have been substituted by a single body, changing in this way the general robot's configuration.

4.2 Simulation Framework Analysis: Asynchronous Multi-Body Framework

From the analysis conducted in the State of the Art chapter, the Asynchronous Multi-Body Framework was chosen as simulation environment for this thesis because it offers a reliable ROS communication and the possibility to deal with soft bodies in the simulation environment. In addition to these, it presents several other interesting features that have been used in this work, like the ability to deal with complex closed-loop robots and to execute real-time dynamic simulation of multi-bodies. Furthermore, it provides real-time haptic feedback (with connection of external devices, see appendix E) and the opportunity to take advantage of the Python Client for the training of reinforcement learning agents on real-time data.

4.2.0.1 AMBF File Format

The Asynchronous Multi-Body Framework introduced a new robot description file format to overcome the limitations of most of the currently available ones: the AMBF file format. In fact, the most common representation formats have issues that have not been solved yet. The URDF (Universal Robot Description Format) is one of the most worldwide used format for robots. The idea behind it is that a robot is a spatial tree of bodies where joints are essential parts of links; because of this it perfectly allows to model serial chains, but on the other hand it does not allow to define unconnected, sparsely and densely connected combinations of bodies [24]. This involves the impossibility to reproduce realistic closed-loop constraints, which represents a major issue in the definition of surgical robots employing a remote centre of motion like the da Vinci. Another Format, the SDF (Simulation Description Format), used for example in Gazebo, has the same problem of URDF in defining closed-loop mechanisms. However, it is more flexible than the previous

one thanks to its design studied for a more general purpose use and to support distributed description of robots.
Both the mentioned robot representation formats share the adoption of XML language, traditionally used to store and transmit configuration data and characterized by difficult human readability. Mainly for this reason, other markup languages like JSON (JavaScript Object Notation) and YAML (Yet Another Markup Language) are now growing in popularity because of their readability and possibilities they offer.
AMBF file format overcomes the mentioned limitations of URDF and SDF. Its main characteristics are well explained in [24]:

- *Human Readability*: possibility to intuitively modify, create and test multi-bodies;
- *Distributed definition*: data for a single body, constraint or environment are specified in a relevant definition block so that if a block is modified or removed, other bodies or constraints remain the same;
- *Constraint handling*: a body may have more than one constraint: if one of them is modified, the others are not affected except for the physical implications;
- *Controllability*: ability to apply forces on the body from the running simulation. The connected bodies react passively based on the type of constraint they share;
- *Communicability*: ability to get information from every body independently during simulation;
- *Dynamic loading*: ability to add bodies at run-time and define at the same time new constraints.

4.2.0.2 AMBF Description File and Tree of Bodies

The AMBF description files (ADF) are written based on the AMBF file format. They are used in the AMBF framework to import multi-bodies in simulation. They also offer both the possibility to define multi-bodies in a single description file or to separate their parts in different files. In fact, an AMBF description file can be seen as composed by multiple blocks, the most important of which is the header one located at the beginning that contains global parameters and header lists. The header lists include all the elements of that specific description file. The types available for such elements are: World data, Rigid Body data, Soft Body data, Constraint Data, Lighting data, Camera data and Input Device data. ADF basic structure is represented in figure 4.4.

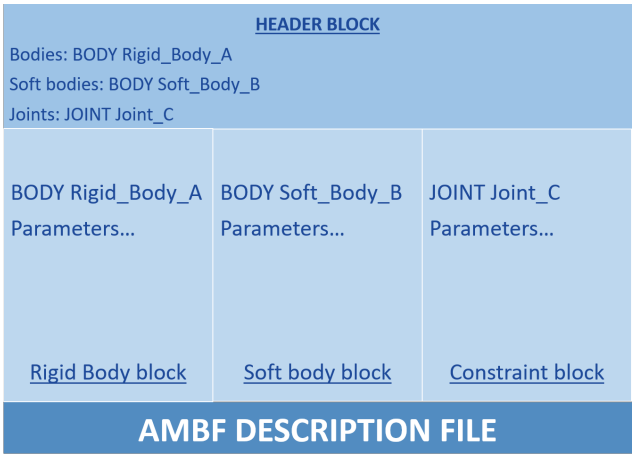


Figure 4.4: Structure of ADF with the header block containing the definition of the elements included in that the specific file

Each element has its own properties and parameters provided by libraries such as Bullet or CHAI-3D. Another important feature is the possibility for every single element to specify a different namespace with respect to

the global one defined at the beginning of the file. Similarly, the resource path of vision and collision meshes can be set in every single block. All of these characteristics reflect the philosophy behind the AMBF file format previously explained, especially the requirements of distributed definition and constraint handling.
This structure allows to overcome the limitations of SDF and URDF in modelling parallel mechanisms by defining a densely interconnected tree structure. Differently from other formats where the parent frame is considered the immediate predecessor body, in AMBF file format all the predecessors of the body are seen as its parents. To achieve this kind of structure, an upward and a downward pass for each added constraint is used. In this way, a single body can have more joints, each one that links it with another body, thus offering the possibility to create parallel and closed-loop mechanisms.
In figure 4.5 the connections between bodies composing a da Vinci PSM are shown.

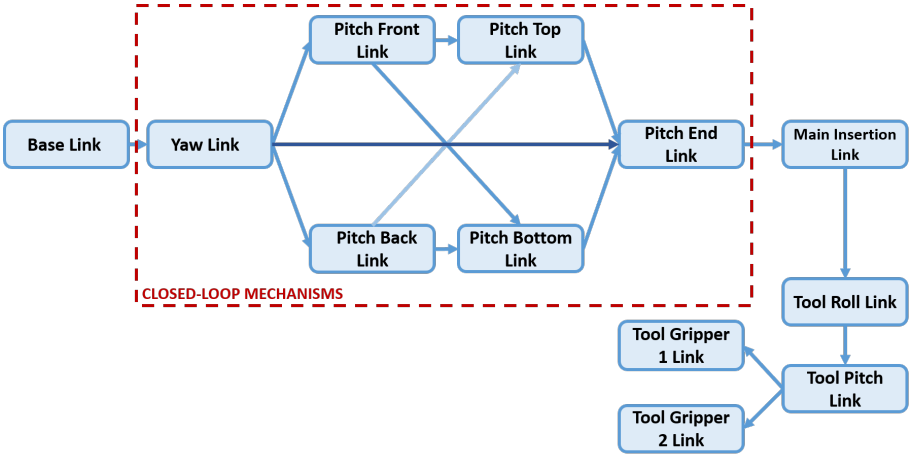


Figure 4.5: Densely connected tree of bodies composing the PSM

From the graph, it is easy to understand for each body what are the parents and what the children. For example, the children of the *Base Link* are all the other bodies, but it has no parents; the parents of the *Tool Gripper1 Link* are all the other bodies except for the *Tool Gripper2*, but it has no children; the parents of *Pitch Front Link* are the *Base Link* and the *Yaw link*, while the children are all the other bodies except for the parents and *Pitch Back Link*.

4.2.0.3 AMBF Python Client

The high-speed Asynchronous Communication between the simulated objects in the framework and the user is implemented via ROS topics (see appendix A), so that an application can be created either with C++ or Phyton. AMBF provides a Python Client that manages the ROS communication. Every body in the simulation initiates a thread using an Inter Process Medium via ROS topics. The communication is bidirectional: this means that it is possible both to set commands to the bodies and at the same time get information about their state. The Python Client manages the interface between the user and the framework by providing a library of Python functions that can be used in a simple code. This functions allow, for example, to set and read bodies and joints positions and orientations or to get the children bodies and the name of the joints. All this information are then put on ROS topics. Figure 4.6 summarizes the communication between the framework and the user. On one hand, commands to the robot are set in a Python code, the Python Client transfers them on the relative ROS topics, and the framework receives the actions to execute and moves the robot. On the other hand, the state of the robot is provided by the AMBF framework itself. Then, it is sent via ROS to the Python Client that in turn manages the topics and provides the desired information to the user.
In a code where one wants to employ the Python Client, the first step

they have to do is to import it from the `ambf_comm` package. Then, an instance of the Client needs to be created and with the method *connect()* the Client itself is connected. This in turn creates callable objects from ROS topics and initiates a shared pool of threads for bidirectional communication. The last step, to act on an element in the simulation or to obtain information about its state, it is to get its handle through the client method *get_obj_handle()*.

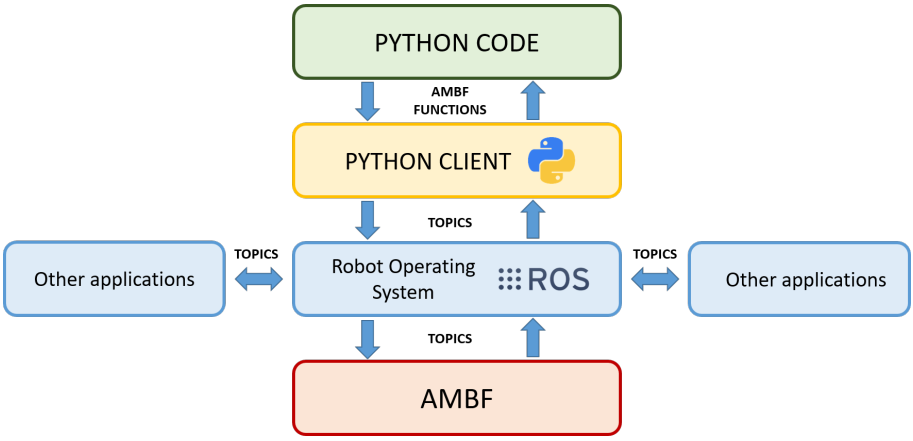


Figure 4.6: Bidirectional ROS communication between a Python application and AMBF managed by the Python Client

4.2.1 Bullet Physics Library

The Asynchronous Multi-Body Framework relies on the Bullet Physics engine. A physics engine is a computer software used to describe physical phenomena and systems. Rigid body dynamics, soft body dynamics, collisions detection are all features owned by a physics engine that need to be resolved in order to provide the updated world transform for all the objects[6]. Bullet includes soft body dynamics for cloth, rope and deformable volumes and AMBF, differently from other frameworks such

as V-REP or Gazebo that consider rigid body dynamics only, fully exploits its features. The main structure of this physics library is the one reported in image 4.7.

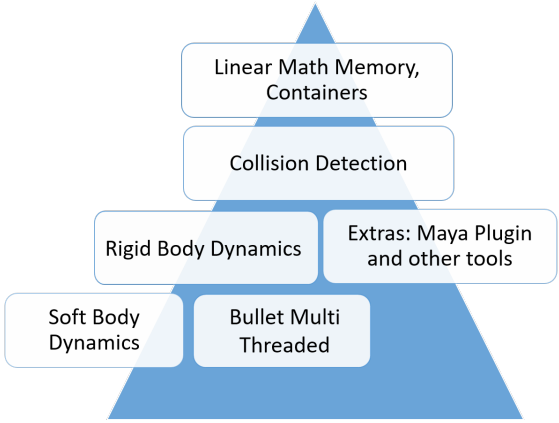


Figure 4.7: Complete Bullet Physics structure as reported in [6]. Soft body dynamics is included in rigid body dynamics, which in turn is a component of the collision detection library

In the AMBF code, the initialization of the Bullet world is done in the `ambf_simulator.cpp` file, where a `btSoftRigidDynamicsWorld` object is created. This, differently from the `btDiscreteDynamicsWorld`, considers rigid body as well as soft body dynamics. These classes are derived from `btDynamicsWorld` and provide a high level interface that manages physical objects and constraints and implements their updates. The Bullet Physics Library computes and updates step by step the physics of the system of bodies in the simulation by calling the `stepSimulation()` function on the dynamics world. In the same file, gravity is also set. There are also 3 loops that are executed every time: the physics loop, the graphics loop and the haptic device update loop. The first one contains the main Bullet simulation loop and runs at a defined physical frequency that can be set from command line while launching the simulation. Its standard value is 1000Hz (see Appendix C). In this loop,

there is also a function `updateDynamics()` that at every iteration of the physics loop iteratively updates the state of every object in the scene. Here, the `stepSimulation` function previously described is called.
The graphics loop simply renders the scene, the haptic device update loop contains the main haptic simulation loop and it is executed only if an haptic device is connected (appendix E).
In order to execute the physics updates, it is important that all the objects in the scene have well established physical properties. These can be specified through the parameters set in the relative description file.

4.2.2 AMBF Simulation: Import Bodies

4.2.2.1 AMBF-Blender addon

A set of robots, environments and objects described in AMBF description files is already available in the Asynchronous Multi-Body Framework. These models include the components of the da Vinci robot (Setup Joints, PSM, MTM, ECM), the KUKA, the Raven, different grippers and tools. However, in some cases the user needs to modify one of these already included models or to create an object ex novo to add to the simulation. This is not trivial. Considering for example the default ROS simulator, RViz, it does not offer the possibility to generate robot models. The same is true for Gazebo, even if this allows to upload URDF robot models that can be created using a 3D CAD software, Solidworks, and a proper converter (Solidworks2URDF). Vice versa, URDF files cannot be uploaded in Solidworks and there is no way to modify existing models.
For the aforementioned reasons, AMBF provides an add-on that overcomes these difficulties using Blender [22]. Blender has a large community support for graphic designers and represents an immediate and intuitive interface for the user to create or modify bodies. The Blender add-on is bidirectional as figure 4.8 testifies. This means that allows both to import objects defined through the AMBF file format and to create

AMBF description files of robots and bodies built in Blender. This component highly simplifies the process of creation of new elements to the user, who can tune rigid bodies, soft bodies and constraints parameters with a real-time visual feedback.
This plugin has been employed in this thesis to generate simple objects to be imported in the simulation in order to study their interaction with the PSM.

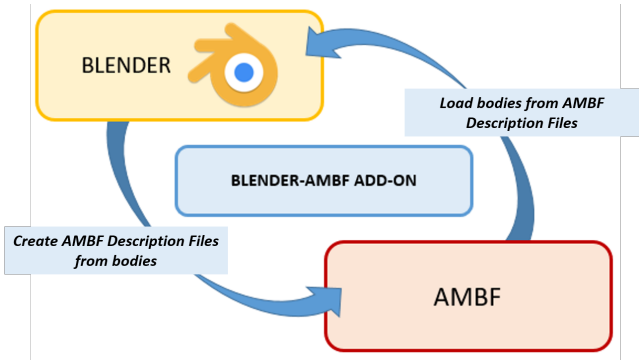


Figure 4.8: AMBF-Blender add-on. It offers both the possibility to load bodies defined in ADF and to generate ADF of objects created in Blender

4.2.2.2 Import Rigid Bodies

Rigid bodies like boxes, cylinders, spheres have been imported in AMBF. All of them have been generated thanks to the AMBF-Blender add-on. The properties set in the configuration file are the same used in the graphic software: in fact, Blender and AMBF share the Bullet Physics library, so the parameters are the same. From Bullet library, the rigid body dynamics is implemented on top of the collision detection module and the main rigid body object is `btRigidBody`. Rigid bodies can be of 3 different types:

1. Dynamic rigid bodies: they have positive mass and at each simulation frame the dynamics is updated;
2. Static rigid bodies: they have zero mass and cannot move, just collide with other elements in the scene
3. Kinematic rigid bodies: they have zero mass and their motion is decided by the user. This means that, unlike conventional rigid bodies, their kinematics is not affected by the force exerted by other objects in the scene.

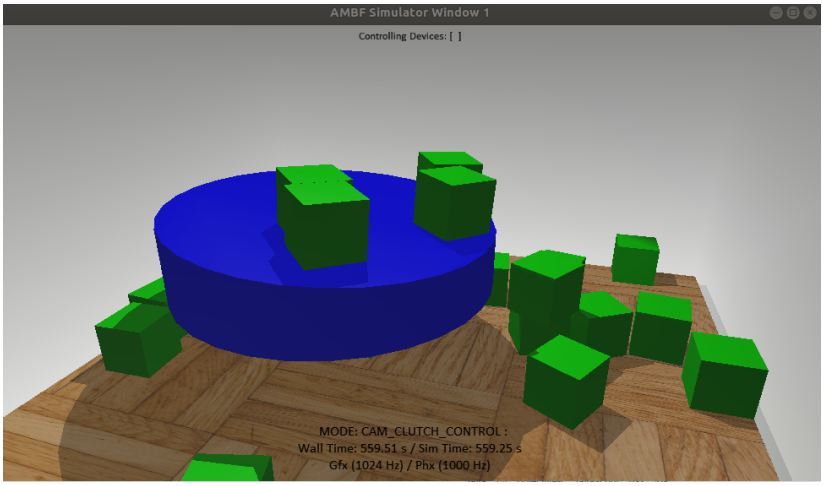


Figure 4.9: Rigid bodies in the framework. In blue, a static rigid cylinder; in green, a series of dynamic rigid boxes

In order to import a rigid multi-body in the simulation, its meshes have to be defined in the AMBF description file. These can be specified in the header or in the relative body block and should contain the path to the file where the mesh is located. There are two fields: the high resolution path and the low resolution path. The first one points to the location of the high resolution visual mesh. The second points to the location

of the low-resolution collision meshes. Then, a set of parameters can be specified: some are collision-related (collision margin, collision shape, collision geometry), some are inherent to the physics of the body (mass, inertia, static friction, rolling friction), some specify the position and orientation of the object. Other parameters, like *"publish joint positions"* or *"publish joint/children names"*, regard ROS communication.
Figure 4.9 shows rigid bodies created with the add-on and imported in the simulation environment.

4.2.2.3 Import Soft Bodies: Issues and Solutions for their Employment in the Framework

AMBF manages soft bodies in a different way from rigid ones.
The first difference consists in how they are created, since the AMBF-Blender add-on does not include soft body generation. A solution to this problem would be to manually write the AMBF configuration file, but some fields like the high and low resolution paths, where the visual and collision meshes are located, are tricky to complete. For this reason, another approach has been adopted. Soft bodies are created in Blender as if they were rigid bodies. In this way, the AMBF-Blender add-on can be used to generate an AMBF description files in which the high and low resolution meshes are automatically defined. Then, in order to switch from rigid to soft bodies, some parameters must be directly modified.
The parameters that can be set come from the Bullet Physics library and are also listed in the `afFramework.cpp` file. There is not a proper documentation about these, but a sufficient knowledge can be achieved by looking at the Bullet forum. The information obtained are reported in appendix C.
In addition, the right flags available in AMBF have to be used when the simulation is launched from the terminal (see appendix D), since different flags correspond to different simulation properties. In particular, special attention has to be paid towards the time-step. There are two

possibilities:

- Variable Time-Step: It is measured how much time the previous frame takes, then this value is fed as delta time for the physics update in the following frame. In this way, the physics update is dynamically adapted to the refresh rate.

- Fixed Time-Step: The physics updates do not accept a "time elapsed" as it was in the variable time-step. On the contrary, with this option it is assumed that each physics update lasts a fixed time period.

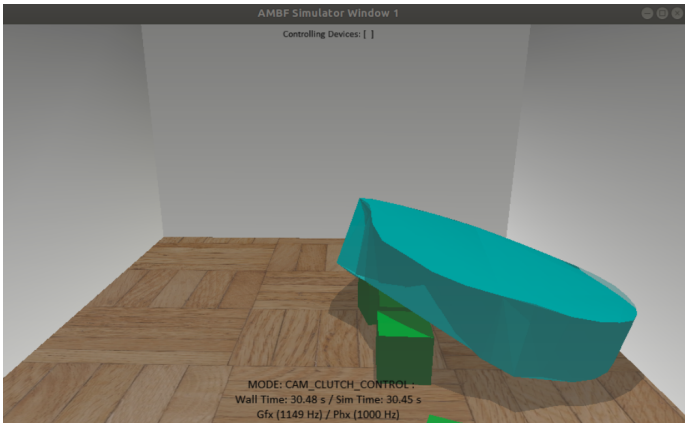


Figure 4.10: Correct interaction between a soft body and rigid boxes (in green)

AMBF sets a variable time-step by default, but this approach has several disadvantages: it is non-deterministic and unpredictable at very large or small steps. In fact, if the time-step varies a lot between consecutive frames, and particular if it is too big, there could be issues with the simulation's integrator. This intrinsically produces errors that are acceptable if the time-step is small enough, otherwise missed collisions and unpredictable behaviors are more likely to occur. Moreover, if a

soft body is present in the scene, the use of variable time-step makes the visual rendering of the simulation definitely too slow with respect to a real world scenario. While these issues can be considered of little relevance in fields like videogames (where this is the preferred option), in simulation where the physical rendering is of great importance like in this case, the use of a variable time-step is not recommended.
On the contrary, a fixed time-step offers a series of advantages: it is predictable, deterministic and adds consistency to the simulation, for this reason it is the preferable choice in physics simulations. It decouples simulation and rendering frame-rates, thus allowing to use a fixed delta time value for the physics update even at different frame-rates [10]. Using a fixed time-step (set through the `-t` flag, that needs to be True), and keeping the default physics update frequency equal to 1000 Hz, the soft body simulation works as expected. In order to avoid inter-penetration issues, it is required to correctly set the parameters of soft bodies in the AMBF description file. In figure 4.10 an example of soft body laying on rigid boxes.

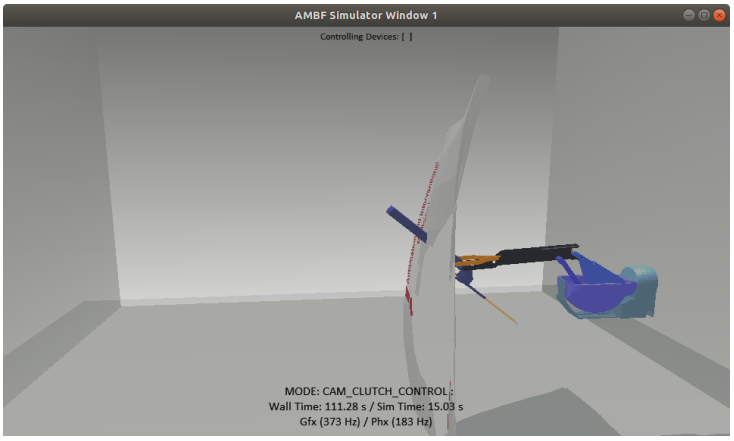


Figure 4.11: Soft body issue: interpenetration between the robot and a cloth. In this case the simulation is too slow and the PSM is not controllable

If the flags for simulation or the body's parameters defined in the configuration file are mistaken, soft bodies present unexpected behaviours when interacting with other objects, both rigid and non-rigid. In a case like this there are problems of inter-penetration, self-collapsing and velocity of simulation. In figure 4.11 the penetration of a cloth by the PSM is shown. In such situation, in addition to the aforementioned issues, the control of the arm by means of the functions provided by AMBF is not possible anymore as the robot, in presence of a soft body in the scene, becomes completely unmanageable. As soon as a command is prescribed, the PSM just starts to execute random, fast movements.
Another reason for which issues of mesh penetration occur is the following. Supposing that a robot is commanded in such a way that it interacts with objects in the scene, deformations of soft bodies are rendered realistically only until a certain point, beyond which penetration takes place. This happens when the contact force becomes too big, and, since AMBF does not provide a tool to read it, these problems are more likely to occur. This represents an obstacle especially in designing control algorithms for the interaction between bodies. Without a force feedback, they would never be able to work in the right way.

4.2.3 Ultrasound Probe-Bodies Contact Force: How AMBF Has Been Modified to Expose this Information

In intra-operative, intra-corporeal ultrasound the contact force exerted by the probe on human tissues is of primary importance. In fact, a proper value of this quantity allows on one hand to have an optimal acoustic coupling, to limit the deformation of tissues and consequently to obtain higher quality images, on another hand it is safer for the patient. For these reasons, contact force is crucial for the implementation of control algorithms aimed at managing interaction between bodies. In the next paragraphs, it is explained how it has been obtained and made available out of the framework by means of the ROS communication.

4.2.3.1 Ultrasound Probe-Bodies Contact Force: Obtain this Information from the Physics Solver

In chapter 3 some of the reasons for which contact force between the ultrasound probe and the patient's body is so important for the project have been explained. In this thesis, the real surgical scenario in which the probe slides on an organ applying a certain force is simplified to a simulation in which the robot exerts a force on a generic soft body. Force information is crucial in simulation not only to provide a realistic modelling of the world, but also from a more practical point of view in order to avoid inter-penetration between meshes. Contact force is also required to implement force control algorithms able to manage the interaction between the da Vinci's PSM and objects present in the scene. Nevertheless, AMBF does not provide any tool to read it.
In order to solve this issue, different approaches have been carefully evaluated, like the possibility to use a function from the Bullet Physics Library, or to exploit haptic devices in order to catch the force providing the haptic sensation.
None of the analysed methods gave the possibility to extract the correct contact force, so it was necessary to look into the Bullet physics solver. This solver is set in the `CBulletWorld.cpp` file, where the Bullet world created in the `ambf_simulator.cpp` is initialized. The solver employed is the `btSequentialImpulseConstraintSolver`, that is an implementation of the Projected Gauss Seidel method. Basically, it treats each contact problem sequentially and then iterates until all contact laws are fulfilled to a specified error or a certain number of iterations is reached [47].
In this solver, forces and torques acting in a robot's joints are computed with respect to the simulation's global reference frame. Consider figure 4.12. Forces computed in joints J represent a good approximation of the contact force exerted by body A on body B if body A has a very small mass and moves slowly enough to consider its inertial effects negligible.

For this reason, and because the direct contact force between two elements is not available in AMBF, force calculated in such a way has been employed in the rest of this thesis.

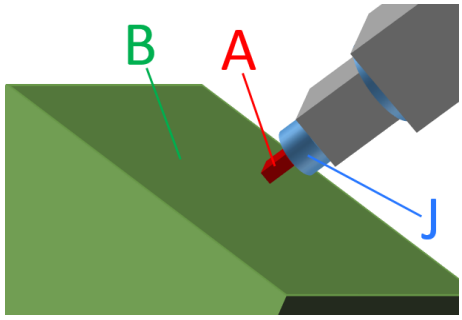


Figure 4.12: Body A belonging to a manipulator enters in contact with body B. Under certain conditions, force in joint J can be considered a good approximation of the real contact force, as explained in the text

4.2.3.2 From the Physics Solver to AMBF

AMBF provides methods to select a body that is part of a specific multi-body, the Bullet Physics Library contains functions to pick a particular joint belonging to a certain body. In this way, starting from the whole PSM model, it has been possible to choose the joint of interest.
Some preliminary experiments have been executed in order to understand what was the optimal option. Maintaining the PSM joints fixed, only the *mainInsertionLink* was moved until it entered in contact with a body: a rigid box or a cloth constrained horizontally.
However, it was not possible to control the tools of the robot through the functions provided by AMBF because of instability issues. In fact, the only possibility to execute the exepriments described was to leave them floating. Considering figure 4.13, take the two joints A (between *tool pitch link* and *tool gripper2 link*) and B (between *main insertion link* and *tool roll link*). Force computed in A is actually useless and does not bring any information about the contact force. Vice versa,

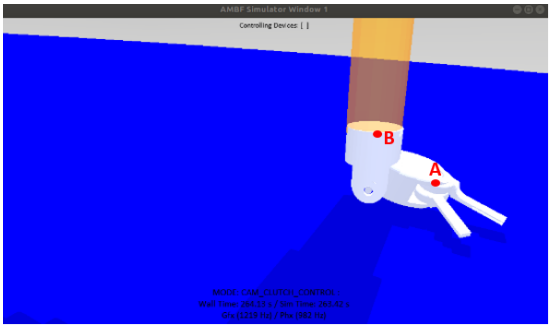


Figure 4.13: Force computed in A was useless since the tools could not be controlled. Force computed in B was much more meaningful. This is one of the reasons that later in this work brought to a new design of the tip of the PSM.

force computed in B is much more meaningful, and these simple tests demonstrated that it well reflects the force of interaction between the two objects.
For this reason, the function *getJointFeedback()*, which returns the forces and torques computed in the `btSequentialImpulseSolver`, has been applied to this second joint.

4.2.3.3 From AMBF to the Application

At this point, the force had to be transferred on a ROS topic in such a way to have it available out of the framework. In the simulation, for every object composing a multi-body, for every element present in the simulated field, including the world itself, the box containing the scene, the cameras and the lights, there are two specific topics: one to set commands, the other to return the state of the object. The topics relative to the state of the objects have different fields and the figure 4.14 well describes their structure; it is worth to note that the field specific for force and torque was already present. In order to transmit the force, first of all this was brought from the `ambf_simulator.cpp`

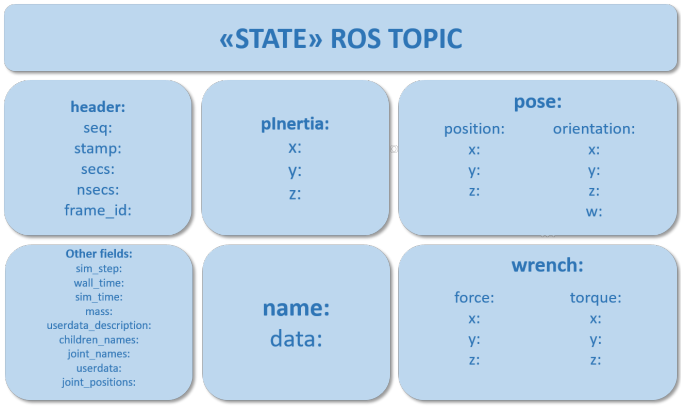


Figure 4.14: Structure of the "State" topic of a body in simulation. It is possible to observe that a *wrench* field was already present, but unused. For this reason, it has been employed to send the force of interaction

to the `afFramework.cpp` file, where the functions implemented for the ROS communication are called. These functions are implemented in the `Object.cpp` file within the `ambf_comm` package, and fill positions, orientations and all the other fields of the "state" topics. Among them, a function built for the force, *cur_force()*, was already defined and the respective field was always set to 0. For this reason, this function has been employed in the `afFramework.cpp` file to send the force over the *"/ambf/env/psm/toolgripper1link/State"* topic.
Finally, the ROS communication is managed by the Python Client. In the file `ambf_object.py` there is a set of functions that can be used in a python code (where the Client has been connected) with the handles of the objects created. These return what is transmitted on the ROS topics, and a simple function has been added in order to conveniently provide the force. In the end, the ROS subscribers defined in the Client have been slightly modified by increasing the `buffer_size` in order to avoid transmission delays. In this way, an easy and effective way to use this force has been implemented.

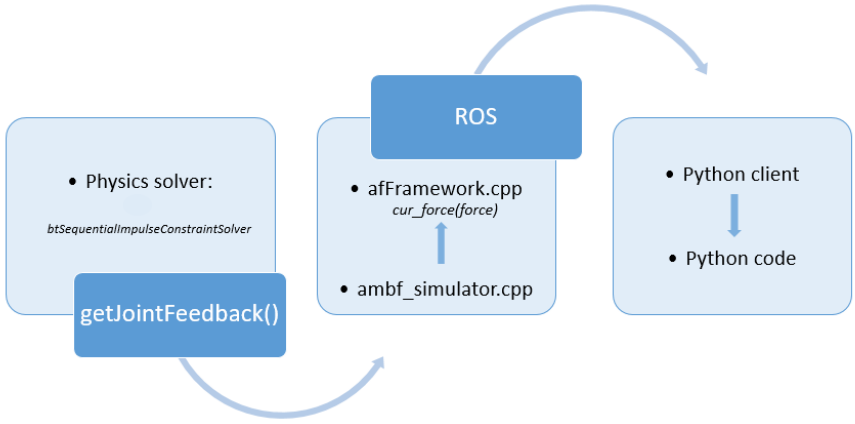


Figure 4.15: Procedure to extract the contact force from the Physics Solver to the application

4.2.3.4 Substitution of the Tools with an Ultrasound Probe-Like Body

In the experiments executed to understand which joint was the optimal one to consider for the force computation, issues of instabilities arose as commands to control the tools were set, therefore they have been left floating. Nevertheless, this solution was not acceptable in the implementation of force control algorithms, for this reason tools have been replaced by a new, small body. The design of a new tip for the PSM had to follow certain specifications. The first requirement was to fit on the PSM *mainInsertionLink*. For this reason a cylindrical base has been chosen, characterized by a diameter equal to the one of the *mainInsertionLink*. The second requirement was to choose a shape that could easily slide on surfaces, hence a sphere has been taken. So, the two shapes have been merged and fixed on the *mainInsertionLink*.
This new tip has been entirely created in Blender, and the generation of the tip's AMBF description file has been realized through the AMBF-Blender add-on. Then, the previous experiments have been repeated

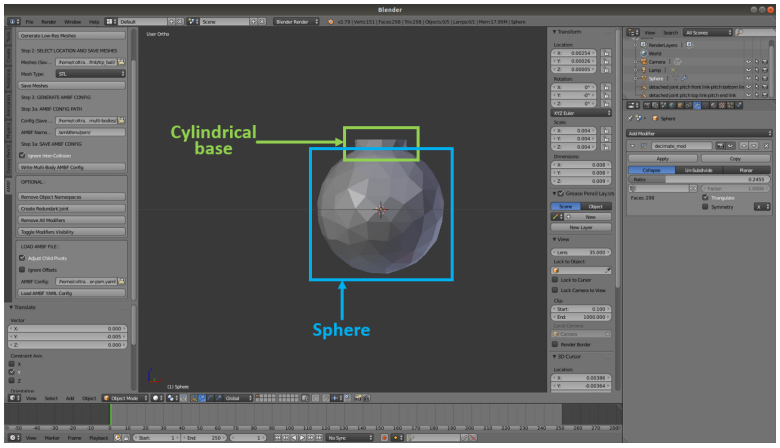


Figure 4.16: Design of the new tip in Blender, composed by merging a cylinder and a sphere

showing better results, as it is explained in the Results chapter. Moreover, the force value has been validated by choosing different values for the new tip's mass while keeping the *mainInsertionLink* vertical, maintaining all the other joints fixed, and without interacting with any other object. With a mass of the tip of 0.001 Kg (then used as default value), a force of -0.0098N has been obtained. By increasing the mass values, the force measured increased proportionally. With a such tiny mass and with the slow movements (and consequently small accelerations) executed in the experiments, inertial forces are negligible.
The created body resembles a real ultrasound probe that could be used with the da Vinci. Ultrasound probes like this, that unavoidably remove some degrees of freedom of the robot, already exist [4]. Nevertheless, a real ultrasound probe to be employed in further steps of the project had not been chosen yet, for this reason the designed, new tip represents in this initial stage a good choice both to test its interaction with other objects and to implement force control algorithms to be applied to the real robot.

4.3 Test and Evaluation

With a force feedback available and a new design of the tip of the PSM, all was settled to implement algorithms in order to evaluate the interaction between the robot and the objects in the simulation. In this section the approaches adopted are illustrated.

4.3.1 Get in Contact With a Body and Reach a Target Force

The simplest case is to control the *main insertion link* only. Thanks to the functions provided by the `AMBF_object` library, it was possible to directly set the desired values of the robot joints in the simulation. In this case, the robot was set in such a way that the *main insertion link* could just slide up and down parallel to the world Z axis, being the joint connecting this body to the *pitch end link* of prismatic type.
The **Algorithm** reported has been used to approach a body and reach a certain force once it is in contact with the robot. Moreover, it has been employed to execute some simple tests in order to observe the response of objects to different values of target force. It also served as first step in all the other experiments carried out before apply the implemented force control algorithms.
First of all, a set of variables is initialized: *length* is the value of the prismatic joint, Δl is the joint value's increase/decrease at each iteration of the main loop, $length_{max}$ is the maximum joint value that can be reached, $force_{target}$ is the target control force. Entering the loop, a moving average filter (N=*10*) is applied to reduce the computational noise affecting the force read. Then, the length of the *main insertion link* is increased or decreased of Δl depending on the value of the force read with respect to the one of the force desired. When this value is reached, the loop is still executed until the force value remains within a certain interval ($force_{target}$+/-ΔF) for a certain number of iterations.

Algorithm: Approach an object and reach a target contact force

```
set Δl, length, length_max, force_target, ΔF, count = 0, count_max;
while force < length_max do
    force = filter(force_read);
    if force < force_target then
        length = length + Δl;
        setJointValue(length);
    else
        length = length − Δl;
        setJointValue(length);
    end if
    if (force < force_target + ΔF)and(force > force_target − ΔF) then
        count = count + 1;
    end if
    if count = count_max then
        break;
    end if
end while
```

Figure 4.17 reports the results of three simulations, each one with a different body: a rigid cylinder, a soft sponge and a cloth constrained horizontally. The **Algorithm** explained previously has been tested by slowly moving the *main insertion link* up and down while keeping the other joints fixed and maintaining the increase in position between each iteration of the main loop equal in each case. a rigid cylinder, a soft sponge and a cloth constrained horizontally.
The target force depends on the object's properties: the stiffer is a body, the higher is the target force that can be applied; vice versa the softer is a body, the lower should be the target force, otherwise issues like mesh penetration are more likely to occur. While the rigid cylinder does not present any problems in the interaction, for the soft sponge and the cloth there is actually a limit in the maximum control force that can be exerted and that should not be overcome. A higher contact force can be reached with the rigid body (in the figure almost 5N). Considering the

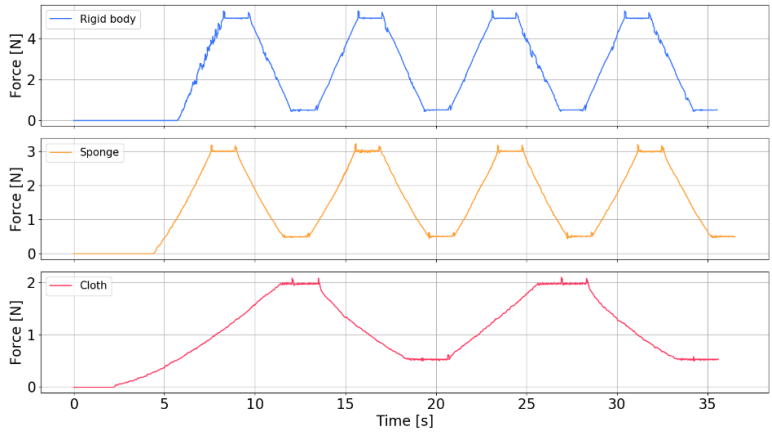


Figure 4.17: From above, **Algorithm** is tested for the rigid cylinder, the soft sponge and the cloth. The prismatic joint, q_3, is increased so that, when the tip enters in contact with the body, force in Z direction of World reference frame (represented in the plots) increases too. When force reaches a certain target value this is maintained for about a couple of seconds, then a new, lower target force value is assigned. This has been repeated for a time of 35 seconds

sponge case, the desired force is reached faster with respect to the cloth. In fact, the second easily deforms because of a lower stiffness, therefore it takes more time to oppose the desired force. In the experiments further executed, a constant force of 2N represents the goal for each one of the bodies considered. This value allows to avoid mesh penetration for each one of the objects examined and represents a realistic value to be exerted on a real human tissue, as the research in [11] testifies.

4.3.1.1 Force Control Based on Position Control in Joint Space

AMBF allows to directly set and read the joint values of the robot. For this reason, the implementation of an hybrid force control in joint space has been implemented as intermediate step prior to develop a more complex algorithm based on cartesian motion control where the computations of forward and inverse kinematics are required.

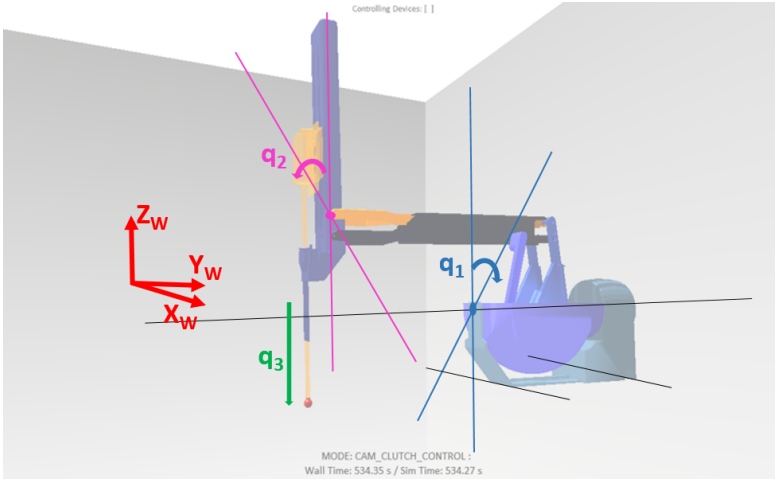


Figure 4.18: Initial configuration of the PSM for joint control. X_W, Y_W, Z_W represent the simulation global reference frame.

Basically, having now available the contact force in the simulation global reference frame (X_W, Y_W, Z_W), it was possible to control the amount of force exerted on the body simply by regulating the value of joint q_3 by means of a PI control. Figure 4.18 represents the initial configuration of the robot. Then, the end effector is made to slide on the object acting on joints q_1 and q_2, nevertheless its position in the cartesian reference frame cannot be accurately known. The control scheme represented in figure 4.19 has been adopted.

The desired joint positions represent the inputs of the motion control. q_3 desired is obtained by adding q_3 read on top of the output of a PI control that relies on the difference between the target and the actual force. The other two inputs, q_1 and q_2, are computed at every iteration by increasing or decreasing by a fixed δq_1 and δq_2 the joint values of the previous iteration, depending on what movement is prescribed to the end effector. As aforementioned, AMBF provides functions to set and

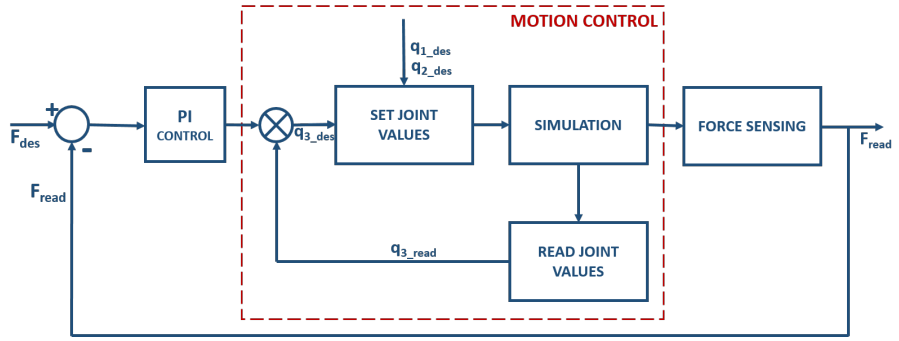


Figure 4.19: Force control adopted with a joint motion control.

read joint's values of the robot: this makes the implementation of such algorithms simple. On the other hand, this approach has strong limitations like the low accuracy in controlling the end effector's position.
Before going on with the analysis of another algorithm, it was checked whether the PID coefficients internally used by the AMBF functions to move the PSM's joints (stored in the PSM description file) were correct, otherwise a modification of their values would have been necessary. In order to verify this, a sinusoidally varying position was separately assigned to each joint, and subsequently read out of the simulation by means of the AMBF functions. The procedure is explained in figure 4.20; with the right PID coefficients, the error between the in and out joint values should have been very low.
The result was that, if the movement between two consecutive iterations of the main loop was small enough, the error was negligible. On the contrary, if the difference in position increased, the error increased as well. Because in the experiments performed only very small movements are executed between consecutive iterations, this error remains very low. This demonstrates that errors generated applying controls based on the AMBF functions to set and read robot's joints cannot be attributed to inaccuracies of the framework's functionalities, but almost entirely to the type of control applied.

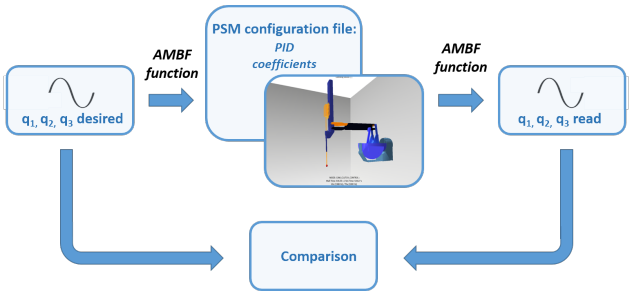


Figure 4.20: Check of PID coefficients in the PSM configuration file. A sinusoidal input is set to the joints through an AMBF function. Another function allows to read the real joint value. Input and output are then compared

4.3.1.2 Force Control Based on Position Control in Cartesian Space

An hybrid force control in cartesian space has been finally implemented. By means of this type of control, it is possible on one hand to accurately control the contact force, on the other hand to specify its desired value. Moreover, it overcomes limitations of the previously developed force control based on joint motion like the impossibility to exactly know the end-effector position in the operational space. This issue, for instance, prevented the user to define a trajectory in the cartesian space to be covered by the robot's end effector. This represented a problem especially in the project considered, since the robot interacts with objects whose position is expressed in the simulation reference frame.
Now, the position of the tip of the PSM is known and expressed with respect to its base reference frame, and can be accurately controlled. In order to implement such control, the forward and inverse kinematics equations of the modified PSM computed at the beginning of this chapter are utilized. In this case, the control diagram represented in figure 4.21 has been adopted.

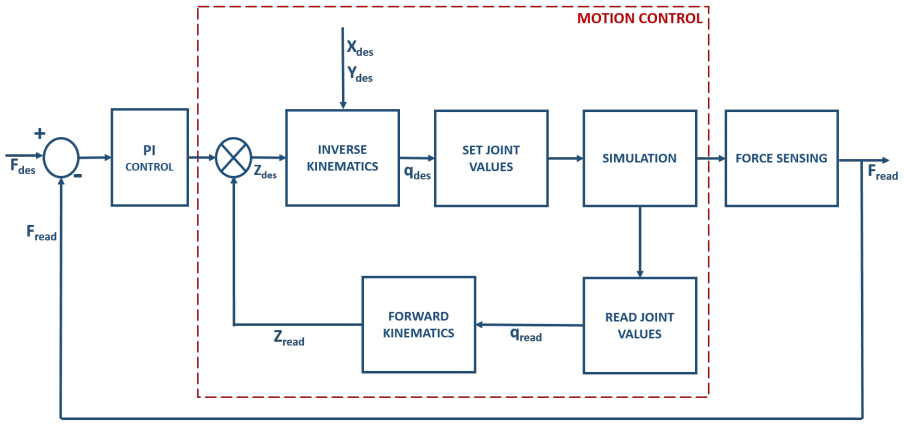


Figure 4.21: Force control adopted with a cartesian motion control

Referring to the scheme, the coordinate X_{des} and Y_{des} are precomputed defining a linear path between two points. The Z_{des} value comes from a PI control that depends on the difference between the target and actual force. These coordinates are expressed with respect to the PSM base reference frame (see X_0, Y_0, Z_0 in figure 4.3) and are given to the inverse kinematics block, which computes the desired joint values to be set to the robot in simulation. Then, joint positions are obtained by means of a function provided by AMBF and enter the forward kinematics block. At this point, the coordinate Z_{read} is added on the top of PI control to calculate its new value in order to track in the best way the target force. In order to test the cartesian motion control at the base of force control, some experiments have been conducted moving the PSM without interacting with any body. Their results are explained in appendix F.

4.3.2 Control Algorithms: Evaluation of the Interaction between Robot and Bodies

At this point, having force control algorithms available, it has been proceeded to the test and evaluation phase, where the interaction be-

tween a Patient Side Manipulator and simulated objects has been analysed. This is the main topic of the next Chapter, where experiments aimed at studying the contact of the robot with rigid and soft objects have been carried out, and their results discussed in detail.
In particular, the controls analysed are those implemented in the previous paragraphs: the algorithm employed to get in contact with a body, the force control based on joints motion, and the force control based on motion control in cartesian space. Especially the third case is studied in depth considering two different patterns of target force (one constant and one sinusoidal) and observing how simulation's outcomes vary changing soft bodies' characteristic parameters.

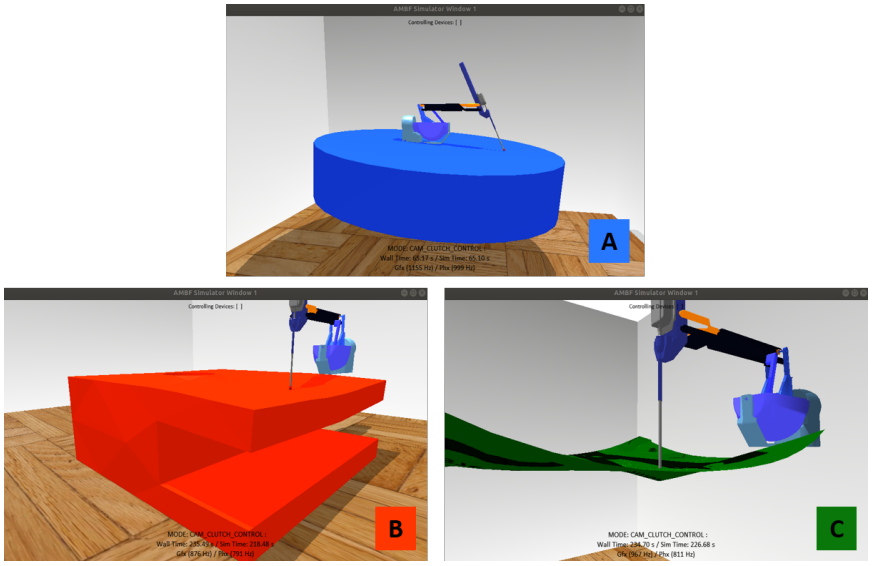


Figure 4.22: The three bodies employed in the experiments while interacting with the PSM: A) Rigid cylinder, B) Soft sponge, C) Soft cloth

The first step in all the experiments conducted is to get in contact with the object, then reach a target force, and finally apply the desired con-

trol. For this reason the first phase in all the experiments consists in applying the **Algorithm** (chapter 4.3.1).
The three bodies used to execute the tests are a cylindrical static rigid body, and two soft bodies: a sponge and a cloth constrained horizontally (figure 4.22). The first one has been slightly rotated in order to provide a certain inclination to the upper base over which the tip of the robot slides. Further information can be found in appendix G. While this rigid body has been entirely generated through to the AMBF-Blender add-on, the meshes of the other two were already present in the AMBF framework, but their physical properties have been modified by adding, removing or changing some parameters.
In the following, experiments aimed at testing the developed control algorithms in robot-bodies interaction are explained.

Test 1: Force Control Based on Position Control in Joint Space

Force control based on joint motion control represents a simple decoupled controller easy to implement by means of the functions already provided by AMBF. In this sense, it can be viewed as an intermediate step in the implementation of a more complete algorithm based on cartesian motion control.
The coefficients of the PI control, k_P and k_I, have been manually tuned, while the k_D coefficient has been omitted because it introduced additional instabilities. Moreover, the force read has been filtered with a moving average ($N = 10$) in order to reduce computational noise.
Resuming what is explained in 4.3.3.1, the contact force exerted in Z direction of the simulation world reference frame onto bodies' surface represents the most important quantity to analyse. In these tests, its desired value is set constant at 2N for the reasons explained in 4.3.1.
A movement is executed increasing joints q_1 and q_2 (refer to figure 4.18) by a fixed amount δq_1 and δq_2 at every iteration of the main control

loop, while joint q_3 is regulated through the action of the PI controller that tries to meet the desired value of target force.

Test 2: Force Control Based on Position Control in Cartesian Space

The previously implemented force control based on joint motion has strong limitations like the impossibility to finely regulate the cartesian position of the robot's end effector. The approach now considered aims at overcoming these issues.
The PI control coefficients have been manually tuned in order to obtain the most accurate results in this case as well. The same experiments executed in Test 1 have been repeated, but this time two scenarios have been analysed in order to better assess the effects of such algorithm into robot-bodies interaction.

Test 2.1: Constant Target Force

A constant desired force of 2N in Z direction of the simulation world reference frame has been established, moving the end effector in X and Y directions directly setting its cartesian coordinates at every iteration of the main control loop. In this way, a comparison with the force control based on position control in joint space can be conducted.

Test 2.2: Sinusoidal Target Force

This test aims at exploring a scenario different from sliding the tip of the PSM over a surface. A sinusoidal desired force in Z direction of the simulation world reference frame has been established. Joint q_1 (refer to figure 4.18) has been fixed at 0 degrees, while the value of joint q_2 was set to -20°in order to give a certain inclination to the *main insertion link*. The X and Y cartesian positions of the tip have been maintained

constant and movement was allowed only along the Z direction of the global reference frame. In particular, four different frequencies have been assigned to the sinusoid: 0.1 Hz, 0.5 Hz, 1 Hz, 5 Hz.

Test 3: Force Control Based on Position Control in Cartesian Space: Influence of Soft Bodies Properties on Force Behaviour

The influence of different soft bodies parameters on simulation's outcomes has been investigated, especially its effect on the force in Z direction of the simulation world reference frame. Considering the sponge, some experiments have been conducted modifying its properties: firstly changing its parameters in the AMBF description file, then analysing the characteristics of its mesh. Force control based on position control in cartesian space has been utilized to slide the end effector onto sponge's surface.

Test 3.1: Influence of Soft Bodies Parameters

From preliminary tests emerged that two parameters in particular influence simulation's outcomes: kLST and kMT. The former is the linear stiffness coefficient and its range is between 0 and 1: if it is closer to 1, the object is more rigid. The latter is the pose matching coefficient, used to enforce the relative vertexes positions. If it tends to 1, vertexes maximally try to maintain their current pose and the object behaves very similarly to a rigid body. In the experiments conducted, three different values have been chosen for both of this parameters, as reported in table 4.3. 10 different movements have been established and effected for each value of both kLST and kMT in order to conduct a statistical analysis (explained later in chapter 4.3.2.1) to assess the effects of these parameters on the error between force desired and read.

Parameter	Value 1	Value 2	Value 3
kLST	0.008	0.08	0.8
kMT	0.0005	0.005	0.05

Table 4.3: Soft bodies parameters of the sponge modified in the experiments to further explore robot-bodies interaction

Test 3.2: Influence of Mesh Characteristics

The effects of soft bodies's mesh characteristics onto the read contact force have been investigated considering the sponge case once again. Its model has been uploaded in Blender thanks to the add-on and the number of vertexes composing the mesh has been increased (figure 4.23). Force control based on position control in cartesian space has been employed to slide the robot's end effector onto the object's surface while maintaining a constant target force of 2N.

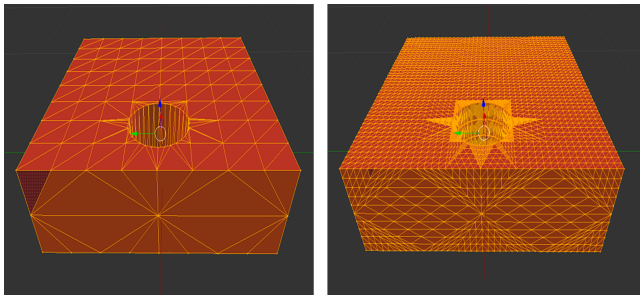


Figure 4.23: Sponge's mesh in Blender environment. Left: low number of vertexes. Right: increased number of vertexes

4.3.2.1 Statistical Analysis

Non-parametric statistical significance tests have been used to assess the effects of the control algorithms and simulation parameters on the

interaction force error (error between the desired contact force and the one read from simulation). In particular, the integral of the force error over time has been used as metric to conduct the statistical analysis. Such metric has been computed as:

$$metric = \int_{t_0}^{t_{end}} force\_error \; dt$$

where t_0 is the time when the experiment begins, t_{end} the time when it ends. The Wilcoxon signed-rank test has been employed with the aforementioned metric as dependent variable, and control strategies or simulation parameters as independent factors. Statistically significant effects have been assessed at $p < 0.05$. The statistical analysis has been performed in MATLAB using the command *signrank()*.

Chapter 5

Results

In order to evaluate the simulation environment and the previously implemented control algorithms, the interaction between the tip of the PSM and different kinds of objects is evaluated. Results of the tests outlined in chapter 4.3.2 are now presented in the following order: force control based on position control in joint space, force control based on position control in cartesian space (both with a constant and a sinusoidal target force), and finally the experiments aimed at analysing the effects of soft bodies properties on simulation's outcomes.

5.1 Test 1: Force Control Based on Position Control in Joint Space

In figure 5.1, the results of the application of this kind of control are represented in order for the cylinder, the sponge and the cloth introduced in chapter 4.3.2. The quantities represented are the force read and the target one, the absolute force error that gives an idea of how well the control force is tracked, the angles of the first two joints which allow to understand when a goal is reached, and, finally, the third joint. In order to examine in a better way the force control applied, force error has been set to 0 until the force read reaches the target value and remains stable for a certain number of iterations. Moreover, referring to table 4.1, q_1 indicates the Outer Yaw joint, q_2 the Outer Pitch joint, q_3 the in/out insertion. For a more explicit representation, refer to figure 4.18. The plots have been obtained by moving the tip through 2 different points, as it is clearly visible looking at the discontinuities in the values of joints q_1 and q_2. More precisely, the movement effected is represented in figure 5.2. Considering the force plots, in case of rigid body the read values wiggle around the desired one maintaining an error below the 0.25N, except for a peak that occurs as a change in the movement occurs. In case of soft bodies, fluctuations are more relevant, with a peak clearly visible in the cloth graph that takes place when the end effector's direction varies.
So, the lower absolute error testifies that the rigid body presents the better results in terms of tracking the target force. Regarding soft bodies' absolute error, except for the spikes, the performances of the sponge are slightly better than the ones of the cloth, probably because of the higher stiffness characterizing the first with respect to the second.
An aspect that needs to be investigated is the unstable behaviour of joint q_3 that emerges in particular with the cloth. This issue concerns soft bodies and it is discussed in the next sections.

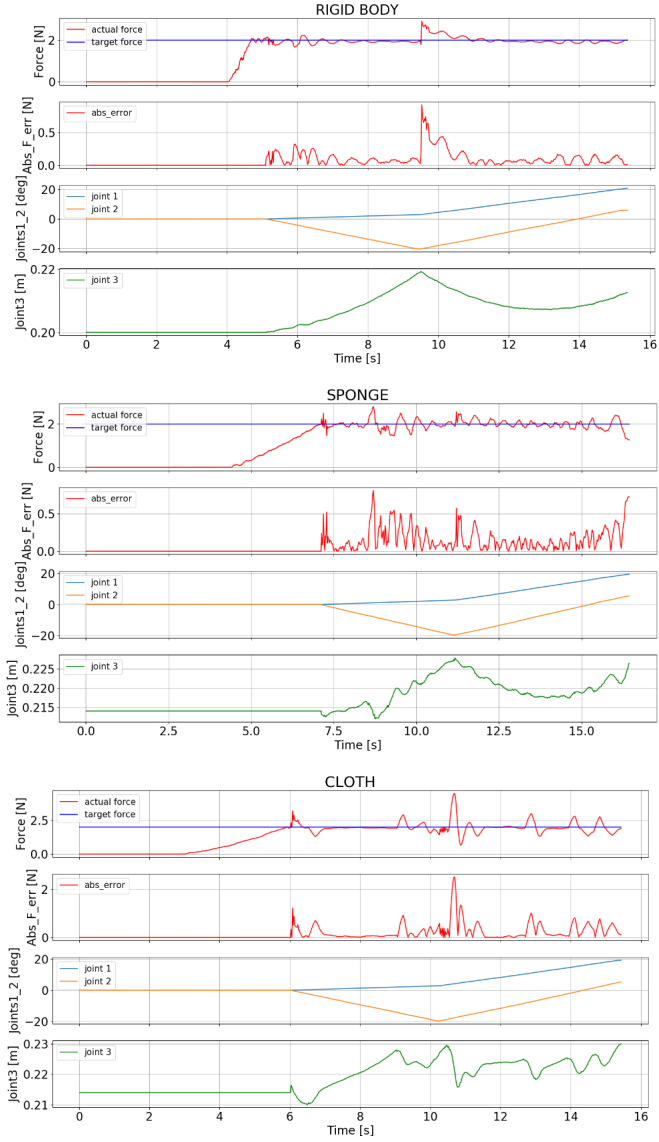


Figure 5.1: From above, joint control applied to rigid body, sponge and cloth executing the movement represented in 5.2. In each graph: force read and desired, absolute force error, q_1 and q_2 positions, read q_3 position. For a clear explanation of q_1, q_2 and q_3 refer to the text or to figure 4.18

5.2 Test 2: Force Control Based on Position Control in Cartesian Space

5.2.1 Test 2.1: Constant Target Force

Simulation results given by the application of this kind of control to the robot, while prescribing a constant target force, are represented in the graphs in figure 5.3. The force read and desired, the absolute force error, and the commanded and recorded positions in X, Y, Z of the end effector expressed with respect to the PSM base reference frame are reported. Each group of plots refers to the interaction respectively with the rigid cylinder, the sponge and cloth. Figure 5.2 shows the movement executed.

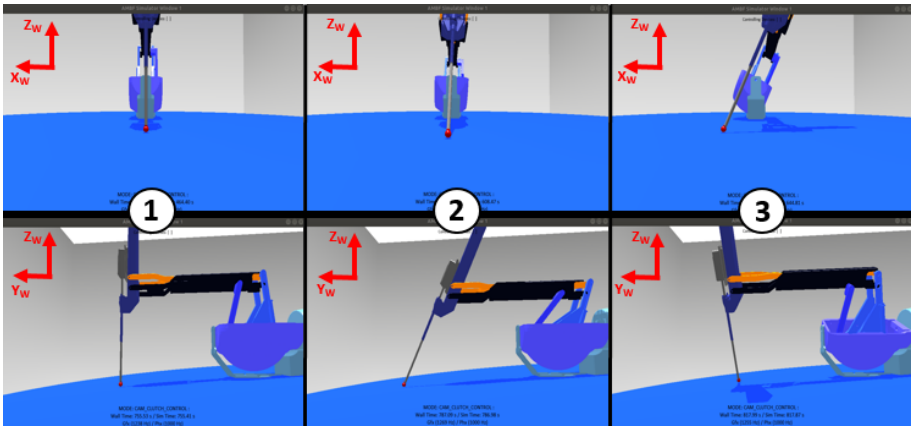


Figure 5.2: Movement executed applying force control based on position control both in joint and cartesian space. Section 1 corresponds to the starting point where the target force of 2N is reached. In figure, the interaction between the slightly inclined rigid cylinder and the robot is represented.

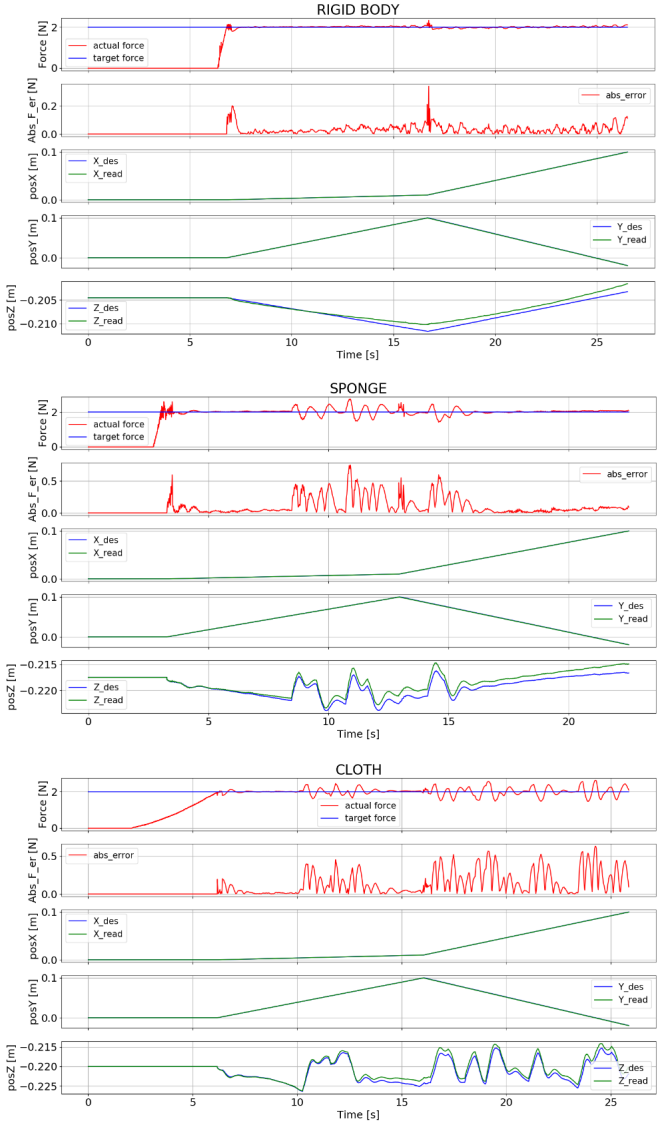


Figure 5.3: From above, cartesian control applied to rigid body, sponge and cloth. In each graph: force read and desired, absolute force error, X desired and read, Y desired and read, Z position desired and read. The movement executed is represented in figure 5.2

Rigid cylinder:

Experiments on this rigid body show that the force read well tracks the desired one. This can be seen looking at the first two plots, especially at the one representing the absolute force error. In fact, it remains below the 0.1N during the whole movement except for the beginning, when the desired force is reached, and when a sudden change in the X and Y commanded position occurs. Position error for both these coordinates is negligible as they are almost perfectly superimposed. The Z position of the tip realistically reflects the inclination of the cylinder; nevertheless, the difference between the Z set and recorded is higher with respect to the other two coordinates but remains contained within a couple of millimeters.

Sponge:

The force read presents irregularities if compared to the desired one. In particular, where these anomalies are present, the absolute force error overcomes the 0.5N; elsewhere, it remains much lower. It is possible to observe force irregularities also at the beginning, when the desired value of the contact force is reached. The X and Y set and read positions appear superimposed, while the Z coordinate slightly differs between its commanded value and the recorded one. Moreover, the Z coordinate presents sudden variations in the order of some millimeters with respect to its expected value.

Cloth:

Irregularities in the values of the force read are present in this case as well. The absolute force error reaches the 0.5N, and it is contained only in an initial phase. The commanded X and Y coordinates are superimposed with the read ones, this means that their error is negligible. The

discrepancy between the Z set and recorded is a bit higher, but remains between a couple of millimeters. The curves of Z position have significant fluctuations.

Experiments on rigid body gave results characterized by the lowest absolute error in terms of tracking the control force. Vice versa, soft bodies present clear irregularities in the force read, with fluctuations that reach a value of about 0.7N in both the cases of sponge and cloth. Considering positions, the error between the commanded and recorded ones is comparable for each one of the objects considered: it is negligible in X and Y direction, very low in Z. The only difference between rigid and soft bodies regards the fluctuation in the Z coordinate that characterizes soft objects, for which a deeper analysis is required.
As explained in 4.3.2, the PI control coefficients have been manually tuned in order to obtain the most accurate results. For each body, slightly different values of k_p and k_i were required (for further details, refer to appendix G). The effect of a wrong tuning of such coefficients is a worsening in the performances, as the plots reported in 5.4 testify.

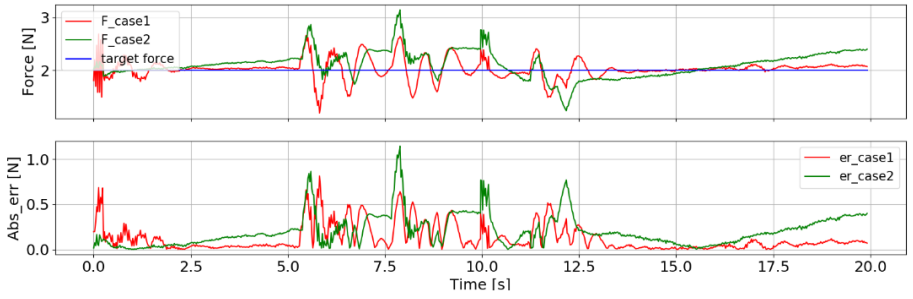


Figure 5.4: Effect on force and force error of different k_p and k_i in the case of the sponge. Case1: $k_p = 1.5 \cdot 10^{-3}$ and $k_i = 0.8 \cdot 10^{-3}$; Case2: $k_p = 0.5 \cdot 10^{-3}$ and $k_i = 0.8 \cdot 10^{-4}$. Where fluctuations take place, there is also a variation in the Z coordinate due to irregularities in soft body's structure (discussed in chapter 5.3). Anyway, a general increase of the error between target and read force is obtained changing the PI coefficients from their optimal values

These graphs consider interaction between robot and sponge and represent two cases: case 1 in which the PI coefficients are the optimal ones manually tuned and employed in figure 5.3; case 2 in which they have been slightly modified. In case 2, the error reaches higher peaks and the baseline of the force read does not follow the desired one as in case 1.
In general, this approach improves results in terms of absolute force error with respect to the previous one based on joint motion control. To quantitatively assess this improvement, 10 different movements have been executed applying joint control, and then repeated employing the cartesian one. The time integral of the force error over each movement has been computed in order to apply statistical analysis (refer to chapter 4.3.2.1 for a more detailed explanation). Boxplots in 5.5 show a force error given by cartesian control statistically significantly lower than the one obtained by means of joint control.

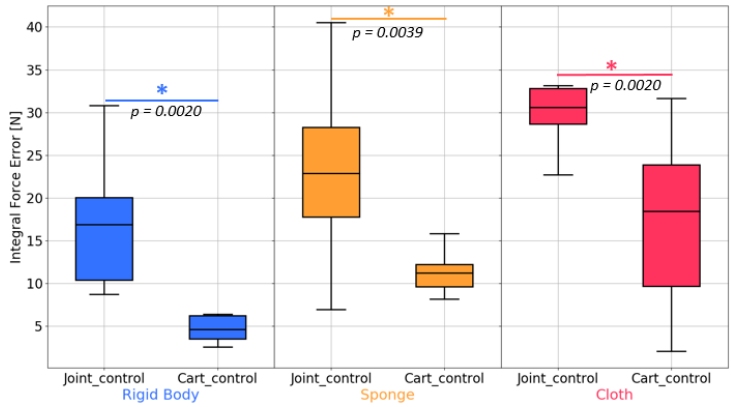


Figure 5.5: Distributions of the absolute force error integral across 10 different movements executed with force control based on motion control in joint and cartesian space. Refer to chapter 4.3.2.1 for further details on metric definition and statistics

This happens because the second type of control (cartesian), despite a

negligible error due to inverse kinematics, allows to have direct control on the cartesian end effector's position, which implies a better control on the force in cartesian Z direction. On the contrary, force control based on position control in joint space does not allow to have such control on the tip of the PSM, whose position is more subjected to variations that in turn translate into higher fluctuations in the contact force in cartesian Z direction.

5.2.2 Test 2.2: Sinusoidal Target Force

Simulation's outcomes of this type of control (in which a sinusoidal target force is prescribed) are presented in figure 5.7. Results are comparable for each one of the three bodies, with an error that increases as the frequency augments. In fact, while the lowest frequency allows to track the target force most accurately with an absolute error stably below 0.1N, the higher one produces an error comprised between the 0.2N and 0.3N.
In these experiments soft bodies do not present significant sudden variations between the force read from the simulation and the target one as it happened in the previous tests. The average absolute force errors are represented for each case considered (adding a frequency of 5Hz) in figure 5.6: higher frequencies involve higher errors.

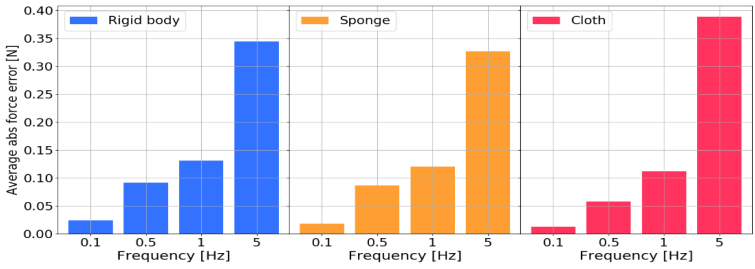


Figure 5.6: Average absolute force error for rigid body, sponge and cloth at frequencies of 0.1Hz, 0.5Hz, 1Hz, 5Hz. Errors increase if frequency augments.

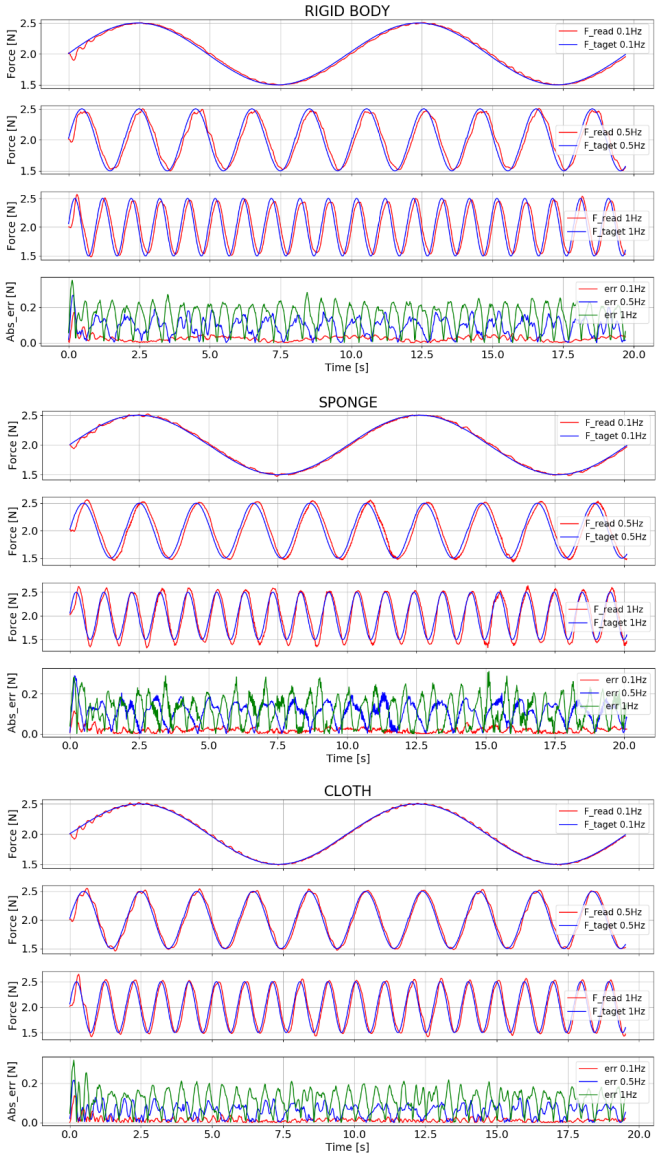


Figure 5.7: From above, force control with cartesian motion control applied to rigid body, sponge and cloth with a sinusoidal control force

5.3 Test 3: Force Control Based on Position Control in Cartesian Space: Influence of Soft Bodies Properties on Force Behaviour

Looking at the plots of the sponge and the cloth in 5.3, force and Z coordinate variations seem to be related. Figure 5.8 (in which the sponge case is illustrated), reports above the force, read and desired, below the Z coordinate set and recorded. When a fluctuation in the force occurs, there is also a variation in the Z coordinate that presents a similar pattern every time it takes place. This issue is characteristic of soft bodies and an explanation on why it occurs is given in the next paragraphs.

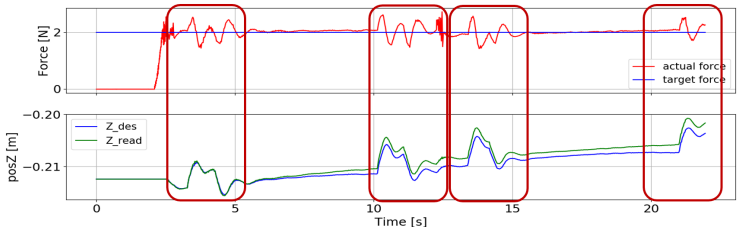


Figure 5.8: Force and Z coordinates fluctuations in the case of the sponge

5.3.1 Test 3.1: Influence of Soft Bodies Parameters

As explained in 4.3.2, two parameters are now considered: kLST and kMT. The effect of the first one is represented in figure 5.9. In the boxplots on the right, the influence of kLST is verified starting from the force error distributions for three values of kLST. 10 different movements have been executed and repeated in each case, then, in order to assess the effects of kLST on the interaction force error, a statistical analysis has been carried out. The integral of the contact force error over time has been used as metric (for further information, refer to chapter 4.3.2

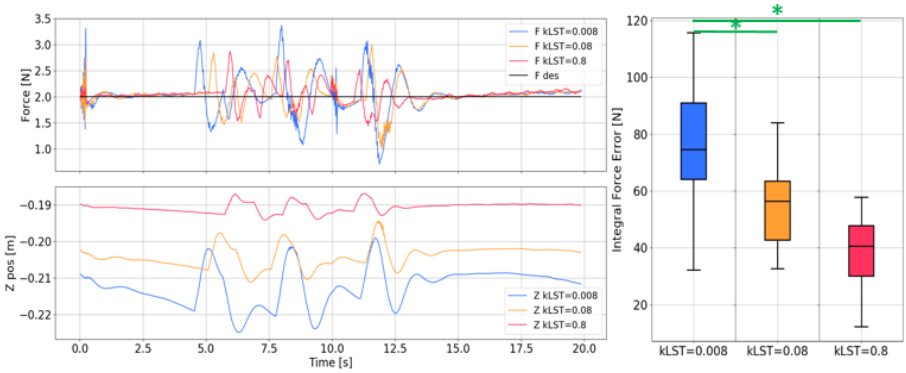


Figure 5.9: Cartesian control applied to a sponge, each time with a different kLST. On the left, force behaviour and Z coordinate in a single movement. On the right, distributions of the absolute force error time integral obtained by executing 10 different movements for each value of stiffness. Information on p values can be found in Appendix G

and 4.3.2.1). Decreasing kLST value, whose range is between 0 and 1, the body exerts a weaker opposition to the deformation induced by the robot, for this reason the tip of the PSM penetrates more in depth and variations in Z coordinate become more relevant. Once again, Z coordinate fluctuations reflect the ones of the force read, whose absolute error lowers as kLST is closer to 1.

Figure 5.10 shows the influence of kMT on simulation's output. If kMT is low, the object tends to deform very easily, the tip penetrates more in depth and the variation in Z, then reflected on the force read, is higher. Even in this case, the desired force is better tracked if this parameter's value is high. As for kLST, this has been verified starting from the force error distributions for three values of kMT, executing 10 different movements for each case, using the integral of the contact force error over time as metric, and carrying out the statistical analysis reported in chapter 4.3.2.1. Statistical significant difference has been obtained for

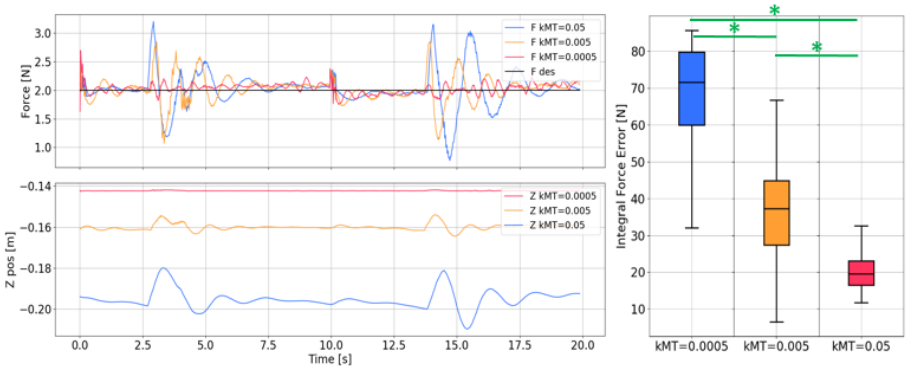


Figure 5.10: Cartesian control applied to a sponge, each time with a different kMT. On the left, force behaviour and Z coordinate in a single movement. On the right, distributions of the absolute force error time integral in each case obtained by executing 10 different movements for each value of kMT. Information on p values can be found in Appendix G

each pair of distributions.

5.3.2 Test 3.2: Influence of Mesh Characteristics

None of the parameters settable in the AMBF description file cancelled the sudden fluctuations in force and in Z coordinate reported in figure 5.8, hence an explanation had to be found in the intrinsic properties of the mesh. As explained in chapter 4.3.2, the number of vertexes constituting the mesh of the sponge has been increased. Force control based on position control in cartesian space has been applied once again and results are shown in figure 5.11. In this case no relevant variations in the read Z coordinate are observed, and the force recorded tracks the desired one more accurately than before with an absolute error between the two contained below the 0.25N. Spikes in the force are present only at the beginning and when there is a change in the direction of move-

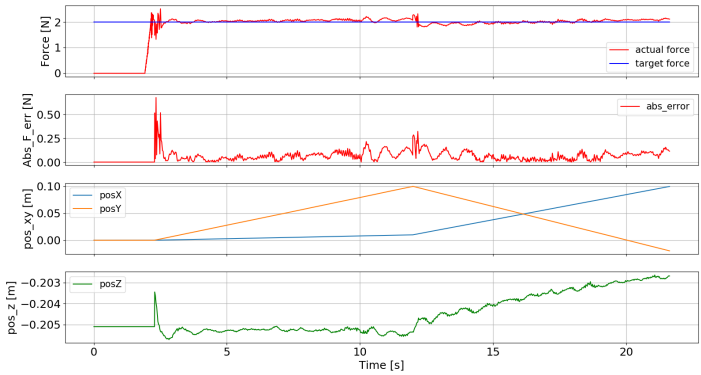


Figure 5.11: Results obtained increasing the number of vertexes composing the sponge's mesh. From above, the control force and the one read, the absolute force error, the commanded X and Y position and the Z position obtained.

ment. In fact, with a low number of vertexes the mesh is subdivided in a limited number of surfaces. If a deformation occurs, they change their relative position in such a way that two close surfaces may have a consistently different inclination, thus making their contiguous border sharper. In this way, when the tip of the PSM slides on the deformed mesh and meets a line connecting two vertexes, there is sudden variation in the mesh surface (that explains the change in the Z coordinate) which automatically causes a change also in the force read. With an higher number of vertexes, the number of surfaces composing the mesh increases and it appears smoother. When it deforms, the difference in inclination between contiguous surfaces is less relevant. This is the reason why sliding on a mesh like this, the Z position of the PSM does not present the aforementioned issues.

Chapter 6

Conclusions

This thesis is part of a project that aims at automatizing the manipulation of a robotic ultrasound probe with the da Vinci Surgical System, enabling semi-autonomous intra-operative ultrasound scanning into a patient's body. In particular, the goal of this thesis was to implement and evaluate in a proper simulation environment control algorithms for the Patient Side Manipulator of the da Vinci aimed at exerting a well-definite contact force between robot and simulated objects. The Asynchronous Multi-Body Framework (AMBF) has been chosen as a simulation platform suitable for the project. This framework has been recently developed and in this thesis a thorough analysis on its functionalities

has been carried out.
The contact force generated in the interaction between the robot and other objects, both soft and rigid, is particularly important for the implementation of force control algorithms. Initially, AMBF did not provide this feature, therefore the platform has been modified in order to make it available. Employing this quantity and starting from the analysis of simpler cases, a force control algorithm based on cartesian motion control has been implemented to evaluate the interaction between robot and objects.
Experiments have been conducted to test the developed algorithms evaluating various scenarios by changing the desired target contact force (constant and sinusoidal) and setting different properties to the interaction bodies, particularly to soft ones. It has been found out that principally soft bodies's characteristics consistently influence simulation's outcomes, especially the contact force read in the Z cartesian direction and the Z coordinate of the robot's end effector.
In conclusion, the selected simulation framework (AMBF) has been properly modified and it provides now all the kinematics and dynamics information required for tuning and testing the robot control during tissue interaction. More in detail, hybrid force-position controls for the da Vinci Patient Side Manipulator have been implemented and tested while interacting with other objects in simulation. In addition, it is worth pointing out that the Asynchronous Multi-Body Framework already offers the possibility to take advantage of the Python Client to train neural networks and reinforcement learning agents on real-time data with simulation in the loop. This represents a valuable characteristic, which is not common among other simulation frameworks, that will further simplify the training of machine learning algorithms in future steps of the project. In this sense, the work carried out represents a valid starting point for the development of more complex control strategies such as reinforcement learning-based controls, which are discussed in the next paragraph relative to future works.

6.1 Future Work

Starting from the work carried out in this thesis, adaptive force controls with respect to different tissue's properties could be implemented. In this case, various simulation in AMBF would be run with different bodies in order to collect the data necessary to implement modern machine learning algorithms, like reinforcement learning, able to tune the parameters of a controller in accordance to the stiffness of the environment, thus resulting in a more stable force control and also accounting for the difficulty to control the ultrasound probe around a soft surface with unknown stiffness.
AMBF has been recently developed: for this reason many features are still work in progress. New tools are being implemented in order to provide the user new instruments to control the elements in the scene, for example methods for the forward and inverse kinematics computation of the da Vinci PSM. In this sense, these features will be useful to extend this work to the real robot.
In future developments, there are many points to work on. For example, it could be evaluated how much simulation effectively reflects the real world. This could be achieved through a comparison between real data obtained from sensors integrated to the robot (such as force sensors) and simulated ones. Moreover, soft body properties could be further investigated with the goal of reproducing the physical characteristics of real organs. In this sense, simulations could be made more realistic even by shaping organs-like soft objects.
Another aspect on which concentrate future works is the ultrasound probe. In this thesis, the da Vinci's tools have just been substituted with a spherical body in order to easily slide on surfaces and with a shape similar to the one of a probe, but more precise designs can undoubtedly be achieved. For this reason, other shapes could be evaluated in order to make simulation more realistic.

Bibliography

[1] "(Advanced Textbooks in Control and Signal Processing) Bruno Siciliano, Lorenzo Sciavicco, Luigi Villani, Giuseppe Oriolo - Robotics - Modelling, Planning and Control-Springer (2008)". In: ().

[2] "2015_Salehi et al._Patient-specific 3D Ultrasound Simulation Based on Convolutional Ray-tracing and Appea[3770]". In: ().

[3] Mahdi Azizian et al. *Chapter 1 THE DA VINCI SURGICAL SYSTEM*. Tech. rep. 2018, p. 39. URL: www.worldscientific.com.

[4] Seth Billings et al. "System for robot-assisted real-time laparoscopic ultrasound elastography". In: *Medical Imaging 2012: Image-Guided Procedures, Robotic Interventions, and Modeling*. Vol. 8316. SPIE, Feb. 2012, 83161W. ISBN: 9780819489654. DOI: 10.1117/12.911086.

[5] by Caitlin Marie Schneider. *Ultrasound Elastography for Intra-Operative Use and Renal Tissue Imaging*. Tech. rep. 2017.

[6] Erwin Coumans. *Bullet 2.80 Physics SDK Manual*. Tech. rep. 2012. URL: http://bulletphysics.org.

[7] B Davies. *A review of robotics in surgery*. Tech. rep. 2000.

[8] Dr Daniel J Bell and Dr Matt A. Morgan et al. *No Title*. URL: https://radiopaedia.org/articles/ultrasound-introduction?lang=us.

[9] *Dynamics*. URL: https://www.coppeliarobotics.com/helpFiles/en/dynamicsModule.htm.

[10] Glenn Fiedler. *Fix Your Timestep!* 2004. URL: https://gafferongames.com/post/fix_your_timestep/.

[11] Matthew W. Gilbertson and Brian W. Anthony. "Ergonomic control strategies for a handheld force-controlled ultrasound probe". In: *IEEE International Conference on Intelligent Robots and Systems*. 2012, pp. 1284–1291. ISBN: 9781467317375. DOI: 10.1109/IROS.2012.6385996.

[12] Tamas Haidegger. "Autonomy for Surgical Robots: Concepts and Paradigms". In: *IEEE Transactions on Medical Robotics and Bionics* 1.2 (Apr. 2019), pp. 65–76. DOI: 10.1109/tmrb.2019.2913282.

[13] Tamás Haidegger et al. *Force Sensing and Force Control for Surgical Robots*. Tech. rep. 2009.

[14] Adam Harris and James M. Conrad. "Survey of popular robotics simulators, frameworks, and toolkits". In: *Conference Proceedings - IEEE SOUTHEASTCON*. 2011, pp. 243–249. ISBN: 9781612847399. DOI: 10.1109/SECON.2011.5752942.

[15] Peter Kazanzides et al. "An Open-Source Research Kit for the da Vinci Surgical System". In: ().

[16] Steven Kennish and Jane A. Smith. "Intraoperative ultrasound". In: ().

[17] Jens Kober, J. Andrew Bagnell, and Jan Peters. "Reinforcement learning in robotics: A survey". In: *International Journal of Robotics Research* 32.11 (Sept. 2013), pp. 1238–1274. ISSN: 02783649. DOI: 10.1177/0278364913495721.

[18] Nathan Koenig and Andrew Howard. *Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator*. Tech. rep. URL: http://playerstage.sourceforge.net/gazebo/.

[19] Petar Kormushev, Sylvain Calinon, and Darwin G. Caldwell. "Reinforcement learning in robotics: Applications and real-world challenges". In: *Robotics* 2.3 (Sept. 2013), pp. 122–148. ISSN: 22186581. DOI: 10.3390/robotics2030122.

[20] Chin-Hsing Kuo and Jian S Dai. "Robotics for Minimally Invasive Surgery: A Historical Review from the Perspective of Kinematics". In: *International Symposium on History of Machines and Mechanisms*. Springer Netherlands, Jan. 2009, pp. 337–354. DOI: 10.1007/978-1-4020-9485-9{_}24.

[21] Joshua Leven et al. *LNCS 3749 - DaVinci Canvas: A Telerobotic Surgical System with Integrated, Robot-Assisted, Laparoscopic Ultrasound Capability*. Tech. rep. 2005, pp. 811–818.

[22] Adnan Munawar. *Blender AMBF Add-on*. URL: https://github.com/adnanmunawar/ambf_addon/tree/378082a557d278c85309be3affc58708

[23] Adnan Munawar and Melody Su. "Asynchronous Multi-Body Framework (AMBF)". In: (). URL: https://github.com/WPI-AIM/ambf.

[24] Adnan Munawar et al. *A Real-Time Dynamic Simulator and an Associated Front-End Representation Format for Simulating Complex Robots and Environments*. ISBN: 9781728140049. URL: www.json.org/.

[25] Ammar Safwan Bin Mustafa et al. "Development of robotic system for autonomous liver screening using ultrasound scanning device". In: *2013 IEEE International Conference on Robotics and Biomimetics, ROBIO 2013*. IEEE Computer Society, 2013, pp. 804–809. DOI: 10.1109/ROBIO.2013.6739561.

[26] George P. Mylonas et al. "Autonomous eFAST ultrasound scanning by a robotic manipulator using learning from demonstrations". In: *IEEE International Conference on Intelligent Robots*

and Systems. 2013, pp. 3251–3256. ISBN: 9781467363587. DOI: 10.1109/IROS.2013.6696818.

[27] Ngoc Duy Nguyen et al. "Manipulating Soft Tissues by Deep Reinforcement Learning for Autonomous Robotic Surgery". In: (Feb. 2019). URL: http://arxiv.org/abs/1902.05183.

[28] Joseph C. Norton et al. "Intelligent magnetic manipulation for gastrointestinal ultrasound". In: (). URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6677276/.

[29] Carlo Pinciroli et al. "ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems". In: *Swarm Intelligence* 6.4 (Dec. 2012), pp. 271–295. ISSN: 19353812. DOI: 10.1007/s11721-012-0072-5.

[30] Lenka Pitonakova et al. *Feature and performance comparison of the V-REP, Gazebo and ARGoS robot simulators*. Tech. rep.

[31] Philip Pratt et al. *Autonomous Ultrasound-Guided Tissue Dissection*. Tech. rep. 2015, pp. 1–8.

[32] Long Qian, Anton Deguet, and Peter Kazanzides. *dVRK-XR: Mixed Reality Extension for da Vinci Research Kit*. Tech. rep. URL: https://unity.com.

[33] M H Raibert and J J Craig. *Hybrid Position/Force Control of Manipulators 1*. Tech. rep. 1981. URL: http://asme.org/terms.

[34] Michael Reckhaus et al. *An Overview about Simulation and Emulation in Robotics*. ISBN: 9783000328633.

[35] Florian Richter, Ryan K. Orosco, and Michael C. Yip. "Open-Sourced Reinforcement Learning Environments for Surgical Robotics". In: (Mar. 2019). URL: http://arxiv.org/abs/1903.02090.

[36] Andrew. Rohl, ACM Digital Library., and ACM Special Interest Group on Computer Graphics and Interactive Techniques. *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia.* ACM, 2007, p. 335. ISBN: 9781595939128.

[37] Eric Rohmer, Surya P. N. Singh, and Marc Freese. "V-REP: a Versatile and Scalable Robot Simulation Framework". In: (2013).

[38] Artur Sagitov et al. "Extending Gazebo simulator for surgical robotics: tissue and suture modeling". In: *Proceedings of International Conference on Artificial Life and Robotics* 24 (Jan. 2019), pp. 124–127. DOI: 10.5954/icarob.2019.os4-5.

[39] Caitlin M Schneider et al. *Robot-Assisted Laparoscopic Ultrasound.* Tech. rep. 2010, pp. 67–80.

[40] Caitlin Schneider et al. *LNCS 7510 - Remote Ultrasound Palpation for Robotic Interventions Using Absolute Elastography.* Tech. rep. 2012.

[41] Caitlin Schneider et al. "Tracked "pick-Up" ultrasound for robot-assisted minimally invasive surgery". In: *IEEE Transactions on Biomedical Engineering* 63.2 (Feb. 2016), pp. 260–268. ISSN: 15582531. DOI: 10.1109/TBME.2015.2453173.

[42] Ramtin Shams, Richard Hartley, and Nassir Navab. *LNCS 5242 - Real-Time Simulation of Medical Ultrasound from CT Images.* Tech. rep. 2008.

[43] Changyeob Shin et al. "Autonomous Tissue Manipulation via Surgical Robot Using Learning Based Model Predictive Control". In: (Feb. 2019). DOI: 10.1109/ICRA.2019.8794159. URL: http://arxiv.org/abs/1902.01459%20http://dx.doi.org/10.1109/ICRA.2019.8794159.

[44] Giuseppe Spinoglio et al. "History of Robotic Surgery". In: 2015, pp. 1–12. DOI: 10.1007/978-88-470-5714-2{_}1.

[45] Agostino Stilli et al. “Pneumatically Attachable Flexible Rails for Track-Guided Ultrasound Scanning in Robotic-Assisted Partial Nephrectomy - A Preliminary Design Study”. In: *IEEE Robotics and Automation Letters* 4.2 (Apr. 2019), pp. 1208–1215. ISSN: 23773766. DOI: 10.1109/LRA.2019.2894499.

[46] Jingyuan Sun, Zhiyuan Yan, and Zhijiang Du. “Optimal design of a novel remote center-of-motion mechanism for minimally invasive surgical robot”. In: *IOP Conference Series: Earth and Environmental Science*. Vol. 69. 1. Institute of Physics Publishing, June 2017. DOI: 10.1088/1755-1315/69/1/012097.

[47] Johan Sundberg. *Parallel projected Gauss-Seidel solver for large-scale granular matter Examining the physics of the parallel solver and development of a multigrid solver*. Tech. rep. 2014.

[48] Hudson Thomas and Simon Leonard. *Softbody Simulation in Gazebo for Satellite Servicing*. Tech. rep.

[49] *User Guide Project: The da Vinci Research Kit*. Tech. rep.

[50] R. Valero et al. “Robotic surgery: History and teaching impact”. In: *Actas Urológicas Españolas (English Edition)* 35.9 (Oct. 2011), pp. 540–545. ISSN: 21735786. DOI: 10.1016/j.acuroe.2011.12.004.

[51] R. Q. Van Der Linde and P. Lammertse. “HapticMaster - A generic force controlled robot for human interaction”. In: *Industrial Robot* 30.6 (2003), pp. 515–524. ISSN: 0143991X. DOI: 10.1108/01439910310506783.

[52] Cecilie Våpenstad et al. “Laparoscopic ultrasound: A survey of its current and future use, requirements, and integration with navigation technology”. In: *Surgical Endoscopy* 24.12 (2010), pp. 2944–2953. ISSN: 14322218. DOI: 10.1007/s00464-010-1135-6.

[53] Salvatore Virga et al. “Automatic force-compliant robotic Ultrasound screening of abdominal aortic aneurysms”. In: *IEEE International Conference on Intelligent Robots and Systems*. Vol. 2016-November. Institute of Electrical and Electronics Engineers Inc., Nov. 2016, pp. 508–513. ISBN: 9781509037629. DOI: 10.1109/IROS.2016.7759101.

[54] Salvatore Virga et al. “Use the force: deformation correction in robotic 3D ultrasound”. In: *International Journal of Computer Assisted Radiology and Surgery* 13.5 (May 2018), pp. 619–627. ISSN: 18616429. DOI: 10.1007/s11548-018-1716-8.

[55] Yan Wang et al. “A Convex Optimization-Based Dynamic Model Identification Package for the da Vinci Research Kit”. In: *IEEE Robotics and Automation Letters* 4.4 (Oct. 2019), pp. 3657–3664. ISSN: 23773766. DOI: 10.1109/LRA.2019.2927947.

[56] Guang Zhong Yang et al. *Medical robotics-Regulatory, ethical, and legal considerations for increasing levels of autonomy*. Mar. 2017. DOI: 10.1126/scirobotics.aam8638.

[57] Michael Yip and Nikhil Das. “Robot Autonomy for Surgery”. In: (July 2017). URL: http://arxiv.org/abs/1707.03080.

Appendix A

Robot Operating System

The Robot Operating System (ROS), is used by AMBF to provide the communication between the simulation environment and a code external to the framework. ROS is a middleware whose main supported operating system is Ubuntu, responsible for handling the communication between programs in a distributed system.
ROS is composed of two elements: a core with communication tools and a set of plugins and libraries.
ROS core manages communication between nodes, which are separate codes contained into different packages. ROS provides three communication tools:

1. **Topics**. They are buses over which nodes exchange messages. They are intended for unidirectional, streaming communication.

2. **Services**. They allow to create simple synchronous client/server communication between nodes.

3. **Actions**. They are not as commonly employed as the previous two. They provide an asynchronous client/server architecture, where the client sends a request that takes a quite long time. The client can asynchronously monitor the state of the server and cancel the request anytime.

In order to employ these tools, proper libraries have to be used in the code and specific message types have to be defined. Moreover, being open-source, ROS provides lots of packages for a huge number of applications. This characteristic made it one of the most commonly used robotic tool worldwide. Figure A.1 shows the general structure of ROS, with a master and two nodes with a topic between them.

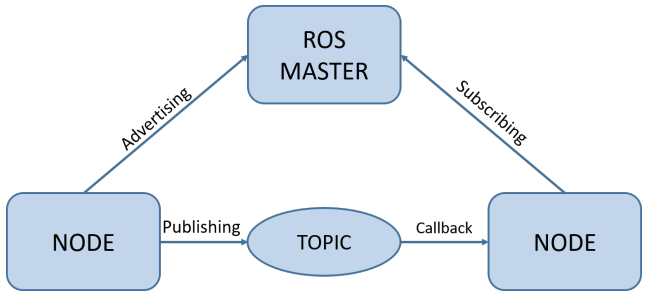


Figure A.1: ROS basic structure

Appendix B

PSMs Structure

This appendix reports more detailed information about the Patient Side Manipulators directly extracted from the user guide of the dVRK [49]. The following elements are reported: a drawing representing the reference frames used to compute the forward kinematics, a table with all the Denavit-Hartenberg parameters of the arm and a table with a complete description for each joint (from the base up to the grippers).

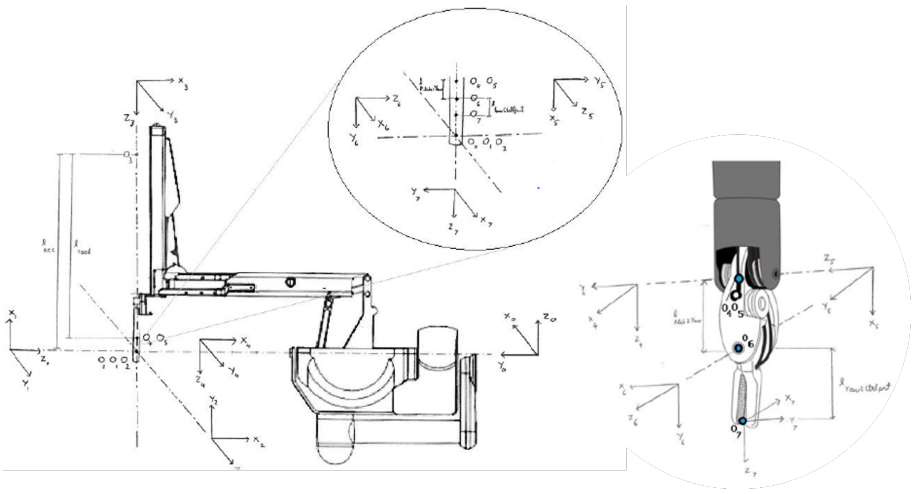


Figure B.1: Reference frames for kinematics description through DH parameters

Frame	Joint Name	Joint type	a	α	D	θ
1	Outer Yaw	1	0	$\frac{\pi}{2}$	0	$q_1 + \frac{\pi}{2}$
2	Outer Pitch	1	0	$-\frac{\pi}{2}$	0	$q_2 - \frac{\pi}{2}$
3	In/out or Insertion	2	0	$\frac{\pi}{2}$	$q_3 - l_{RCC}$	0
4	Outer Roll	1	0	0	l_{tool}	q_4
5	Wrist Pitch	1	0	$-\frac{\pi}{2}$	0	$q_5 - \frac{\pi}{2}$
6	Wrist Yaw	1	$l_{Pitch2Yaw}$	$-\frac{\pi}{2}$	0	$q_6 - \frac{\pi}{2}$
7	End Effector	0	0	$-\frac{\pi}{2}$	$l_{Yaw2Ctrlpnt}$	0

$l_{RCC} = 0.4318\ m$
$l_{tool} = 0.4162\ m$
$l_{Pitch2Yaw} = 0.0091\ m$
$l_{Yaw2CtrlPnt} = 0.0102\ m$
q_1 *to* q_6 *are the joint variables*

Table B.2: PSM DH parameters reported on [49]

Name	Description
Outer Yaw	This is the only joint that moves the entire PSM with respect to its mounting base. It pivots the instrument in a yaw motion about the remote center. Home position (zero joint-angle) is center range of motion, which makes the insertion axis perpendicular to the PSM mounting plate.
Outer Pitch	This joint pivots the instrument in a pitching motion about the remote center. Home position (zero joint-angle) is chosen to make the insertion axis perpendicular to the PSM mounting plate, which it turns out is not quite center range of motion.
In/out out Insertion	This axis moves the instrument along the axis of its shaft into or out of the patient. Home position (zero joint angle) is fully retracted, with the instrument's control point located at the remote center.
Outer Roll	This axis rolls the instrument shaft. Home position (zero joint-angle) is center range of motion.
Wrist Pitch	This axis is the first (proximal) axis on the wrist mechanism (for standard 8mm instruments). Anthropomorphic to a human wrist knocking on a door. da Vinci does not home with instruments installed, so home is not defined in motor space. However, the zero joint-angle corresponds to a straight wrist.
Wrist Yaw	This axis is the second (more distal) axis on the wrist mechanism (for standard 8mm instruments). Anthropomorphic to a human wrist wiping a surface. It is a coordinated motion of two mechanical joints representing the two grippers. da Vinci does not home with instruments installed, so home is not defined for instruments. However, the zero joint-angle corresponds to a straight wrist.
End effector	This joint is controlled in combination with Wrist Yaw 1 to effect wrist yaw and jaw open and close actuation.

Table B.1: Description of PSM's joints

Appendix C

Soft Bodies Parameters

In this appendix is reported a set of parameters for soft bodies that can be set in the relative AMBF description file. These parameters come from Bullet Physics Library, however a clear and exhaustive description was not found online. In the following table, the information retrieved on the web (in particular from the Bullet forum) and from the AMBF code itself are reported and summed up in order to provide a good overview on the soft body properties that can be specified or modified.

Name	Extended Name	Description
m_kLST	Linear stiffness coef.	Self explanatory
m_kAST	Angular stiffness coef.	Self explanatory
m_kVST	Volume stiffness coef.	Self explanatory
m_flags	-	-
kVCF	Velocity correction factor	Amount of correction per time step for drift solver
kDP	Damping coefficient	Damping
kDG	Drag coefficient	Drag or resistance of an object in a fluid environment
kLF	Lift coefficient	Component of force exerted by a fluid flowing around an object perpendicular to the oncoming flow direction
kPR	Pressure coefficient	Pressure used to conserve volume
kVC	Volume conservation coef.	Magnitude of the force used to conserve volume
kDF	Dynamic friction coef.	As rigid body friction. If equals to 0 slides, if to 1 sticks
kMT	Pose matching coefficient	Used for pose matching: enforces relative vertex positions
kCHR	Rigid contacts hardness	How "soft" contact with rigid bodies are
kKHR	Kinetic contacts hardness	How "soft" contact with kinetic/static bodies are
kSHR	Soft contacts hardness	How "soft" contact with other soft bodies are
kAHR	Anchors hardness	How "soft" anchor constraints (joints) are
kSRHR_CL	Soft vs rigid hardness	For clusters only, similar to kCHR.
kSKHR_CL	Soft vs kinetic hardness	For clusters only, similar to kKHR
kSSHR_CL	Soft vs soft hardness	For clusters only, similar to kSHR
kSR_SPLT_CL	Soft vs rigid impulse split	For clusters only: proportion to split impulse with a rigid body after collision
kSK_SPLT_CL	Soft vs kinetic impulse split	For clusters only: proportion to split impulse with a kinetic/static body after collision
kSS_SPLT_CL	Soft vs soft impulse split	For clusters only: proportion to split impulse with another soft body after collision
maxvolume	Maximum volume ratio for pose	-
timescale	Time scale	To speed up or slow down the simulation of a specific soft body
viterations	Velocities solver iterations	Number of iterations for velocities solvers
piterations	Positions solver iterations	Number of iterations for position solvers
diterations	Drift solver iterations	Number of iterations for drift solvers
citerations	Cluster solver iterations	Number of iterations for cluster solvers
collisions	Collisions flags	Go to btSoftBody::fCollision
m_vsequence	Velocity solvers sequence	Order and type of solvers to apply for velocities
m_psequence	Position solvers sequence	Order and type of solvers to apply for positions
m_dsequence	Drift solvers sequence	Order and type of solvers to apply for drift

Table C.1: Soft bodies parameters: description

Name	Type	Default value	Range
m_kLST	btScalar	1	[0, 1]
m_kAST	btScalar	1	[0, 1]
m_kVST	btScalar	1	[0, 1]
m_flags	int	-	btSoftBody::fMaterial
kVCF	btScalar	1	-
kDP	btScalar	0	[0, 1]
kDG	btScalar	0	[0, +∞]
kLF	btScalar	0	[0, +∞]
kPR	btScalar	0	[-∞, +∞
kVC	btScalar	0	[0, +∞]
kDF	btScalar	0.2	[0, 1]
kMT	btScalar	0	[0, 1]
kCHR	btScalar	1	[0, 1]
kKHR	btScalar	0.1	[0, 1]
kSHR	btScalar	1	[0, 1]
kAHR	btScalar	0.7	[0, 1]
kSRHR_CL	btScalar	0.1	[0, 1]
kSKHR_CL	btScalar	1	[0, 1]
kSSHR_CL	btScalar	0.5	[0, 1]
kSR_SPLT_CL	btScalar	0.5	[0, 1]
kSK_SPLT_CL	btScalar	0.5	[0, 1]
kSS_SPLT_CL	btScalar	0.5	[0, 1]
maxvolume	btScalar	1	[0, +∞]
timescale	btScalar	1	[0, +∞]
viterations	int	0	[0, +∞]
piterations	int	1	[0, +∞]
diterations	int	0	[0, +∞]
citerations	int	4	[0, +∞]
collisions	int	fCollision::Default	btSoftBody::fCollision
m_vsequence	tVSolverArray	-	n/a
m_psequence	tPSolverArray	-	n/a
m_dsequence	tPSolverArray	-	n/a

Table C.2: Soft bodies parameters: technical properties

Appendix D

AMBF: Options for Simulation

In this appendix some basic instructions on how to use AMBF are reported. Most of these information can be found in the wiki at the GitHub page [23]. In addition, if it is true that much about AMBF structure has already been explained in the chapters of this thesis, especially in *Materials and Methods*, on the other hand other indications, such as technical details or the location of useful files, were not actually discussed. These topics could be helpful for anyone interested in using this framework and for those who want to further advance in the project.

Launching the simulator

To launch the simulator, the first step is to go to proper directory, specifically the input to run from command line is
`cd /ambf/bin/<os>`
From here, it is enough to run `./ambf_simulator` with the desired flags. An explanation of the flags available can be found using the `-h` flag. The options are the following:

Load objects in simulation

There are three ways to import objects in simulation.

- With the `-l` flag. In this case, the name of the multibody's configuration file has to be specified in the yaml file located in *ambf/ambf_models/descriptions/launch.yaml*, where contains an ordered list of configuration files of robots, objects, haptic devices and environments constituted by multiple objects. The argument is the index at which the desired multibody is reported. More multibodies can be launched at the same time. An example to launch bodies whose configuration file is reported respectively at indexes 1,6,10 in the `launch.yaml` is
  `./ambf_simulator -l 1,6,10`

- With the `- a` flag. This time the argument is the directory in which the configuration file is located. Even in this case, multiple directories of different configuration files can be specified to load more than one body. An example is
  `./ambf_simulator -a /users/object/tests/body1.yaml`

Flag	Brief description
`-n [--ndevs] arg`	number of haptic devices to load
`-i [--load_devices] arg`	index number of devices to load specified in input_device.yaml
`-e [--enableforces] arg`	enable force feedback on haptic devices
`-p [--phx_frequency] arg`	physics update frequency (default=1000Hz)
`-d [--htx_frequency] arg`	haptics update frequency (default=1000Hz)
`-t [--fixed_phx_timestep] arg`	use fixed time-step for physics (default=False)
`-f [--fixed_htx_timestep] arg`	use fixed time-step for haptics (default=False)
`-a [--load_multibody_files] arg`	description filenames of multi-body(ies) to launch
`-l [--load_multibodies] arg`	index of multi-body(ies) to launch
`--launch_file arg`	launch file path to load
`-m [--margin] arg`	soft cloth collision margin
`-s [--show_patch] arg`	show soft cloth patch
`-g [--show_GUI] arg`	show GUI

Table D.1: AMBF flags and relative description

- The easiest way is to directly select with the configuration file in the folder where is located and drag it directly into the framework already launched. From command line only the environment has to be launched.

It is worth to say that entire environments composed by multiple bodies can be can be loaded, which could be useful to recreate certain conditions for an experiment. The configuration files are located in *ambf/ambf_models/descriptions*. Here there are four folders. One con-

tains the colors available in AMBF, the other three contain the configuration files respectively of a variety of multibodies, of the input devices and of the world. Into the folder containing the bodies there are other folders with environments, puzzles of multibodies, tools, grippers, robots or examples of soft bodies.

Appendix E

Connection of External Devices to AMBF

In this thesis an Omega.7 device available in the laboratory where this thesis has been carried out has been connected to the simulation framework. It has been useful in a first phase of the work (before a force feedback was available) to verify how soft bodies reacted to the application of a certain force.
The omega.7, represented in figure E.1 is one of the world's most advanced desktop 7-dof haptic interface. It introduces high precision active grasping capabilities with orientation sensing. Moreover, the force-

feedback grippers are finely tuned to display perfect gravity compensation and offer extraordinary haptic capabilities.

Nevertheless, this device was not included in the list of supported haptic

Figure E.1: Omega.7 device

devices of AMBF. These are specified in the the `input_devices.yaml` file contained in the *ambf/ambf_models/descriptions* folder. For every element in the list some properties have to be specified, like `workspace scaling, location, root link, simulated multi-body`. The last is particularly important because represents how the device is viewed in the virtual environment. To set it, the path relative to the desired multi-body needs to be reported. Commonly, grippers are used with input devices. The section in the `input_devices.yaml` file in which the Omega.7 properties are set in AMBF is reported in E.2. The communication between the device and AMBF is made possible thanks to a library, CHAI-3D. Its functions are used in the `afInputDevices.cpp`, into the `ambf_framework`. Here, the forces computed in simulation (actually not real forces, but rather "virtual" forces used to provide the haptic feedback) are sent to the device through a specific function of CHAI-3D, `setForceAndTorqueAndGripperForce()`. In order to avoid instabilities when using the device, force has been filtered before using

```
Omega7:
    hardware name: omega.7
    haptic gain: {linear: 0.03, angular: 0.0}
    deadband: 0.01
    workspace scaling: 5
    enable joint control: true
    simulated multibody: "../multi-bodies/grippers/pr2 gripper.yaml"
    location: {
     orientation: {p: 0.001, r: -1.571, y: 1.569},
     position: {x: 0.499, y: 0.514, z: -0.693}}
    max force: 12
```

Figure E.2: Section relative to the Omega7 in the `input_devices.yaml` file

such function.
In general, once a device is connected to AMBF, and a multi-body is set in the `input_devices.yaml`, the framework creates two ROS topics for the simulated device: one to set commands, one to read the state. Even if the simulated device is represented by a multi-body, differently from what happens with common multi-bodies, a ROS topic is not initialized for every body composing it, but only for the entire multi-body.
Anyway, to launch a simulation with an input device, it is necessary to specify the `-i, -e, -d, -f` flags, explained in table D.1. For example, to launch a simulation with one single input device with index 6 in the `input_device.yaml` file, with force feedback, the command should be:

```
./ambf_simulator -n 1 -i 6 -e 1
```

Appendix F

Motion Control in Cartesian Space: Experiments without Interaction between Robot and Bodies

In order to test the cartesian motion control at the base of force control, some experiments have been conducted moving the PSM without

interacting with any body.

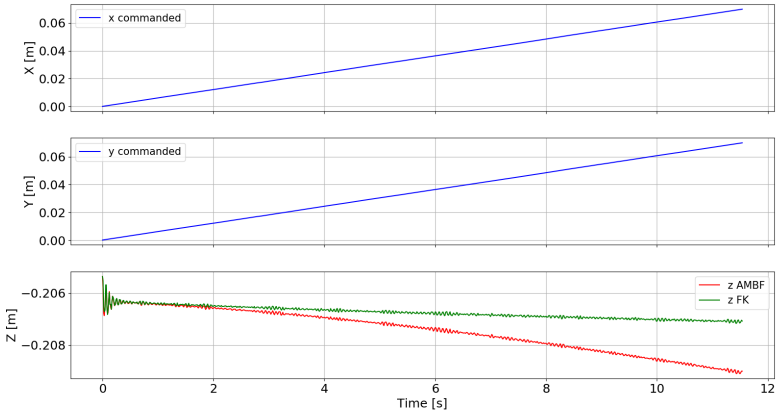


Figure F.1: Graph explaining the issue in Z direction. Increasing the commanded X and Y commanded positions, the discrepancy between the Z coordinates obtained by means of the AMBF function and the forward kinematics increases

A simple path between a starting point expressed in the PSM's base reference frame and one end point was defined. To verify the correctness of forward and inverse kinematics computations two positions have been compared: one obtained through the forward kinematics of the simulated joint values, the other given by a function provided by AMBF that directly returns the tip's position in the simulation reference frame. An issue in the Z coordinate has been identified.
The graphs reported in F.1 show the difference between the two Z coordinates. The error increases as the position in X and Y increases, but remains contained within few millimeters. Nevertheless, the Z coordinate is of key importance in the control employed because it is the variable that directly influences the force exerted on the body. For this reason, it is important to understand which between these two options

actually represents the real position of the tip. To understand this, a simple experiment has been carried out by sliding the tip of the PSM on a flat rigid body with surface parallel to the XY plane.

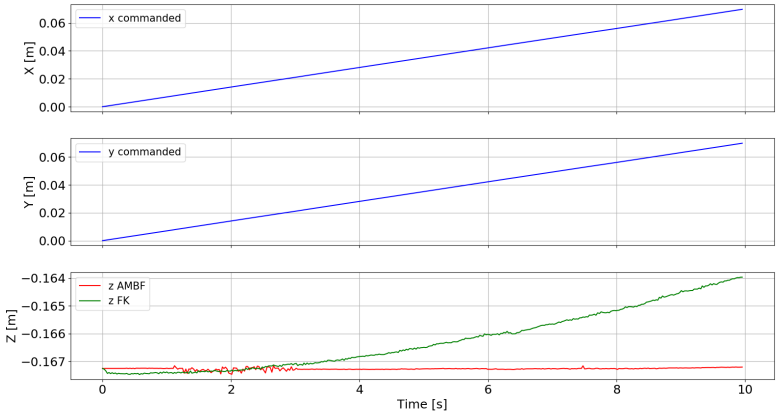


Figure F.2: In this case the movement along Z is constrained in a XY plane thanks to a flat rigid body, over which the PSM tip slides. The AMBF function provides a better result in terms of Z coordinate.

Results are represented in figure F.2. While the Z coordinate provided by the AMBF function remains constant as expected, the Z coordinate given by the forward kinematics slightly varies depending on the distance in X and Y directions. However, this one is not the behaviour observed in simulation, where the tip maintains the contact with the surface for the whole movement. Because the Z coordinate provided by the AMBF function behaves more realistically, it has been chosen for the control employed.

Appendix G

Experiments: technical information

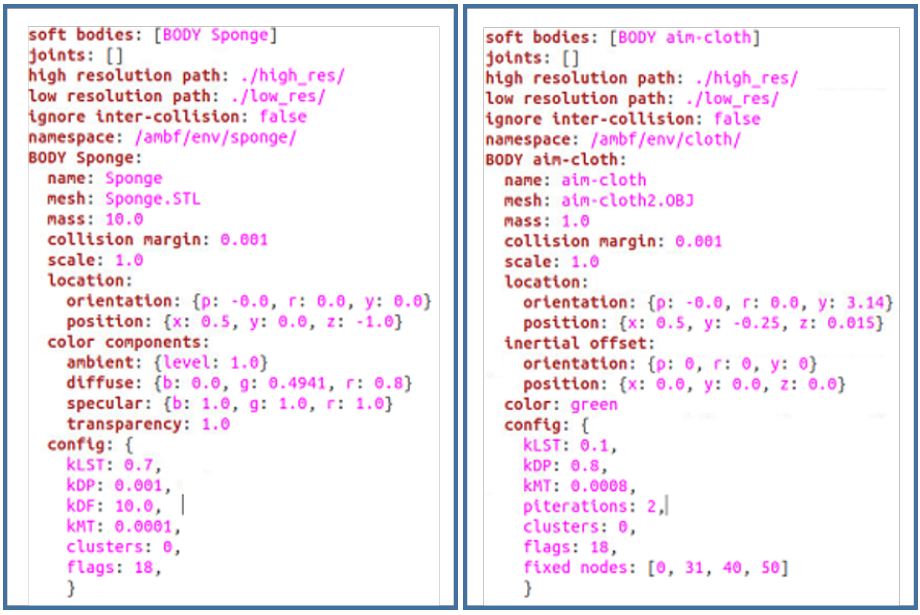
In the following, firstly are reported the PI coefficients employed in the experiments aimed at testing the interaction between the PSM and the three bodies considered (rigid cylinder, sponge, cloth) applying a force control based on cartesian motion control to the robot. Then, the parameters set in the configurations file of each one of the three objects are reported.

Bodies	k_p $[\frac{m}{N}]$	k_i $[\frac{m}{N}]$
Cylinder	$1.5 \cdot 10^{-3}$	$0.8 \cdot 10^{-3}$
Sponge	$1.5 \cdot 10^{-3}$	$0.4 \cdot 10^{-3}$
Cloth	$1 \cdot 10^{-3}$	$0.8 \cdot 10^{-3}$

Table G.1: PI coefficients employed in the force control based on cartesian motion control

```
bodies: [BODY Cyl02]
joints: []
high resolution path: ../../meshes/blender_afmb/shapes/high_res/
low resolution path: ../../meshes/blender_afmb/shapes/low_res/
ignore inter-collision: false
namespace: /ambf/env/
BODY Cyl02:
  name: Cyl02
  mesh: Cyl02.STL
  mass: 0.0
  collision margin: 0.001
  scale: 1.0
  location:
    orientation: {p: -0.0, r: 0.0, y: 4.71}
    position: {x: 0.5, y: 0.0, z: -0.5}
  color: blue
  publish joint names: true
  publish joint positions: true
  friction: {rolling: 0.01, static: 0.5}
  damping: {angular: 0.1, linear: 0.04}
  restitution: 0
```

Figure G.1: AMBF description file of the rigid cylinder



```
soft bodies: [BODY Sponge]
joints: []
high resolution path: ./high_res/
low resolution path: ./low_res/
ignore inter-collision: false
namespace: /ambf/env/sponge/
BODY Sponge:
  name: Sponge
  mesh: Sponge.STL
  mass: 10.0
  collision margin: 0.001
  scale: 1.0
  location:
    orientation: {p: -0.0, r: 0.0, y: 0.0}
    position: {x: 0.5, y: 0.0, z: -1.0}
  color components:
    ambient: {level: 1.0}
    diffuse: {b: 0.0, g: 0.4941, r: 0.8}
    specular: {b: 1.0, g: 1.0, r: 1.0}
    transparency: 1.0
  config: {
    kLST: 0.7,
    kDP: 0.001,
    kDF: 10.0,
    kMT: 0.0001,
    clusters: 0,
    flags: 18,
    }
```

(a) Sponge

```
soft bodies: [BODY aim-cloth]
joints: []
high resolution path: ./high_res/
low resolution path: ./low_res/
ignore inter-collision: false
namespace: /ambf/env/cloth/
BODY aim-cloth:
  name: aim-cloth
  mesh: aim-cloth2.OBJ
  mass: 1.0
  collision margin: 0.001
  scale: 1.0
  location:
    orientation: {p: -0.0, r: 0.0, y: 3.14}
    position: {x: 0.5, y: -0.25, z: 0.015}
  inertial offset:
    orientation: {p: 0, r: 0, y: 0}
    position: {x: 0.0, y: 0.0, z: 0.0}
  color: green
  config: {
    kLST: 0.1,
    kDP: 0.8,
    kMT: 0.0008,
    piterations: 2,
    clusters: 0,
    flags: 18,
    fixed nodes: [0, 31, 40, 50]
    }
```

(b) Cloth

Figure G.2: AMBF description files of sponge (a) and cloth (b)

Soft bodies statistical analysis: p values

KLST	*p values*
KLST = 0.008 vs KLST = 0.08	*0.0039*
KLST = 0.08 vs KLST = 0.8	*0.0840*
KLST = 0.008 vs KLST = 0.8	*0.0098*

Table G.2: p values in kLST analysis

KMT	*p values*
KMT = 0.0005 vs KMT = 0.005	*0.0137*
KMT = 0.005 vs KMT = 0.05	*0.0020*
KMT = 0.0005 vs KMT = 0.05	*0.0039*

Table G.3: p values in kMT analysis