POLITECNICO DI MILANO

SCUOLA INTERPOLITECNICA DI DOTTORATO

Doctoral Program in Bioengineering

Final Dissertation

High level control of Robot Behavior in Neurosurgery

Mirko Daniele Comparetti

Supervisors
prof. Giancarlo Ferrigno
Elena De Momi, PhD

Co-ordinator of the Research Doctorate Course
prof. Maria Gabriella Signorini

27th February 2014

MIRKO DANIELE COMPARETTI

HIGH LEVEL CONTROL OF ROBOT BEHAVIOR IN NEUROSURGERY

DOCTORAL PROGRAM IN BIOENGINEERING

Supervisors: Prof. Giancarlo Ferrigno
Elena De Momi, PhD

Politecnico di Milano
Scuola di Ingegneria Industriale e dellInformazione
Dipartimento di Elettronica, Informazione e Bioingegneria
December 2013

Mirko Daniele Comparetti: *High level control of Robot Behavior in Neurosurgery*
Doctoral program in Bioengineering

To my family

When you discover what you will be in your life,
set out to do it as if God Almighty called you
at this particular moment in history to do it.

Don't just set out to do a good job.

Set out to do such a good job that the living,
the dead or the unborn couldn't do it any better.

If it falls your lot to be a street sweeper,
sweep streets like Michelangelo painted pictures,
sweep streets like Beethoven composed music,
sweep streets like Leontyne Price sings
before the Metropolitan Opera.

Sweep streets like Shakespeare wrote poetry.

Sweep streets so well that all the hosts of heaven and earth
will have to pause and say:
Here lived a great street sweeper who swept his job well.

If you can't be a pine at the top of the hill,
be a shrub in the valley.

But be the best little shrub on the side of the hill.

Be a bush if you can't be a tree.
If you can't be a highway, just be a trail.
If you can't be a sun, be a star.

For it isn't by size that you win or fail.

Be the best of whatever you are.

Dr. **Martin Luther King**, Jr.
What Is Your Life's Blueprint? — Philadelphia, 26th October 1967

ABSTRACT

THE technological advancements in the field of biomedical engineering happened in the last decades changed the way in which surgical interventions are performed. Computer Aided Surgery (CAS) techniques were developed to help and assist the surgeon during the surgical intervention, providing him/her with information that can be used to improve the performances of the treatment. The increasing use of robotic systems in the Operating Room (OR) during the last 20 years brought to the development of Computer and Robot Assisted Surgery (CRAS), an extension of the CAS paradigm in which the robot provides an active contribution to the surgeon. The robotic assistant have to act according to the needs of the surgery and the surgeon, as an intelligent transparent operator which ensures an higher accuracy and a better performance with respect to traditional techniques by reducing the fatigue to the human operator and providing a reliable way to verify and improve the accuracy of the procedure.

In this thesis, the aspects of high level control of a robotic device for neurosurgical intervention was studied and an architecture to manage a robotic system during the execution of the workflow of the intervention, in order to change the parameters and control modes in a semi-automatic way according to the current situation in the OR, the step of the intervention and the surgeon's needs. In detail, the work was focused in the definition of a set of Finite State Machines (FSMs) that can manage the transitions between two steps by properly enabling/disabling the control modes and parameters without causing unpredictable movements and glitches of the robot, which is close to the patient.

In this scope, also controllers to move a tool carried by a robot towards a pre-calculated target pose in space were developed and tested; this procedure, called *targeting* was studied with a target 1. that doesn't move in space, and 2. that can change its pose in time.

In the former case the developed algorithm uses an external localizer to measure the accuracy of the position and, based on that, to iteratively correct the pose of the tool until the accuracy requirements are satisfied; studies on the final accuracy and convergence performances of the algorithm were carried out. In the latter case, an algorithm to follow a target that moves in space was implemented in the developed architecture and tests on the performances were carried out to evaluate the accuracy and the delay.

All the developed algorithms were tested in the scope of the EU funded projects for brain surgery ROBOCAST (FP7-ICT-2007-215190)

and ACTIVE (FP7-ICT-2009-6-270460), aimed at developing integrated solutions to assist the surgeons during the intervention.

Those activities are in the direction of the OR of the future, in which the clinical staff, sensors and the robotic assistants share the environment with a context-driven surgical workflow that adapts the behavior of the devices without requiring a massive intervention of the human operator.

ACKNOWLEDGEMENTS

There we are. Another adventure in Politecnico di Milano has reached its end. The last three years were very challenging and, honestly, they ran out very fast. At this point, it's time to thank all the people that contributed in this adventure.

Prima di tutti, la mia famiglia. Questo lavoro è dedicato a loro in quanto posso dire che sono stati la "principale causa" di questo traguardo. La loro fiducia nei miei confronti, il loro condividere le mie idee e soprattutto il supporto che mi hanno dato nei momenti più duri... senza di loro tutto questo non sarebbe potuto succedere.

My supervisors at Politecnico di Milano. I'm grateful to you for having stimulated in me the scientific curiosity, that is fundamental in research. You taught me that it is more important to find new questions to be answered than simply answering one and move on. You gave me the possibility to work in a really international environment, to collaborate with different institutions, to grow professionally and to improve myself: seeing different points of view of the same problem is the key to open our minds.

Mein Betreuer an der Karlsruher Institut für Technologie. Seine Gastfreundschaft hat mich vom ersten Tag im Büro zu Hause fühlen lassen. Seine Unterstützung ist für meine Ideen sehr wichtig gewesen.

All my lab mates. The time we spent together in finding solutions to our research problems was really constructive. Furthermore, all the fun we had in these years was really important to create a comfortable environment in the lab. For that I'm grateful to you all.

Kollegen und Freunde in Deutschland. Die in Deutschland verbrachte Monate waren für mich wirklich interessant. Dank Eurer Freundschaft habe ich eine gute Zeit im Ausland verbracht, eine positive Lebenserfahrung und tolle Erinnerungen. Ich kann doch behaupten, dass ich neue Freunde gefunden habe.

Gli amici. Voi che avete condiviso con me questi anni e mi avete sopportato... la vostra vicinanza è stata davvero molto importante per arrivare fino a qui.

Grazie, Danke schön, Thanks!

Milan, december 2013

Mirko D. Comparetti

CONTENTS

LIST OF ACRONYMS xvii

1 INTRODUCTION 1
- 1.1 Computer Assisted Surgery 1
- 1.2 Robotic surgery 2
- 1.3 Neurosurgery 4
- 1.4 Aim of the thesis and structure of the document 5

2 BEHAVIOR CONTROL 7
- 2.1 Description of the scenario 7
- 2.2 Workflow, states and transitions 8
- 2.3 Control architecture 10
  - 2.3.1 General 10
  - 2.3.2 Detailed description 11
- 2.4 Test scenario and experiments 16
  - 2.4.1 Workflow 16
  - 2.4.2 Experiments 17
- 2.5 Results 18
- 2.6 Discussions 22

3 TARGET IN NEUROSURGERY 25
- 3.1 Target approaching 25
  - 3.1.1 Introduction 25
  - 3.1.2 Accuracy evaluation 28
  - 3.1.3 Convergence evaluation 44
- 3.2 Target following 56
  - 3.2.1 Introduction 56
  - 3.2.2 Methods 58
  - 3.2.3 Results 62
  - 3.2.4 Discussions 73

4 CONCLUSIONS 75

BIBLIOGRAPHY 78

LIST OF FIGURES

Figure 2.1 The setup of the scenario 8
Figure 2.2 High Level Controller UML component diagram 10
Figure 2.3 High Level Controller UML component diagram: the *Robot controller* is showed in detail 12
Figure 2.4 Null-space movement of the LWR 13
Figure 2.5 Animated schema of the possible steps and transitions event 17
Figure 2.6 Example of the recorded data 19
Figure 2.7 Data of the x axis of the OTS measurements with a time window of $0.1\,s$ 20
Figure 2.8 Data of the x axis of the robot pose with a time window of $0.1\,s$ 21
Figure 3.1 Robots calibration schema 30
Figure 3.2 Spatial transformations used during probe targeting 31
Figure 3.3 Block diagram of the target approaching controller 33
Figure 3.4 The OR set-up during the experiments 35
Figure 3.5 Components of the final $\mathbf{R}_j$ 38
Figure 3.6 IC_{GP} and IC_{FP} in the three evaluation scenarios 39
Figure 3.7 Relationship between TC of the $\mathbf{P}_{i^{th}\ robot}$ and TC of the $\mathbf{R}_j$ after the i^{th} robot movement 40
Figure 3.8 Errors between planned trajectory and real one 41
Figure 3.9 The keyhole neurosurgery scenario 45
Figure 3.10 Kinematic chain representation for the targeting convergence analysis 46
Figure 3.11 Maximum rotation component $\mathbf{N}^{\vartheta}_{MAX}$ which allows the targeting convergence 51
Figure 3.12 Number of iterations required for targeting convergence 52
Figure 3.13 Maximum number of targeting iterations . . . 53
Figure 3.14 Target following kinematic chain representation 58
Figure 3.15 Target following algorithm 60
Figure 3.16 Target following trajectories 64
Figure 3.17 RMS error during the $g_1(t)$ movement: x axis. . 65
Figure 3.18 RMS error during the $g_1(t)$ movement: y axis. . 66
Figure 3.19 RMS error during the $g_1(t)$ movement: z axis. . 67
Figure 3.20 RMS error during the $g_2(t)$ movement: x axis. . 68
Figure 3.21 RMS error during the $g_2(t)$ movement: y axis. . 69
Figure 3.22 RMS error during the $g_2(t)$ movement: z axis. . 70

Figure 3.23 RMS error below threshold during the $g_1(t)$ movement 71
Figure 3.24 RMS error below threshold during the $g_2(t)$ movement 72

LIST OF TABLES

Table 3.1 Surgical robots accuracy 27
Table 3.2 Robots characteristics 29
Table 3.3 Experimental protocol for the targeting accuracy tests 36
Table 3.4 Calibration residuals and calibration errors . . 37
Table 3.5 Negative corrections of the iterative algorithm for GP and FP targeting 39
Table 3.6 Testing protocol for the convergence analysis . 49
Table 3.7 Experimental protocol for the target following algorithm 61
Table 3.8 Fitting parameters for the surface threshold boundary 63
Table 3.9 Time delays between target and robot trajectories 73

LIST OF ACRONYMS

ACTIVE	Active Constraints Technologies for Ill-defined or Volatile Environments
CAS	Computer Aided Surgery
CRAS	Computer and Robot Assisted Surgery
CF	Coordinate Frame
CORBA	Common Object Request Broker Architecture
CT	Computed Tomography
DB	DataBase
DBS	Deep Brain Stimulation
DoF	Degree of Freedom
DRF	Dynamic Reference Frame
ECG	Electro-Cardiography
EE	End-Effector
EP	Entry Point
EU	European Union
FOAW	First-Order Adaptive Window
FP	Fine Positioner
FRI	Fast Research Interface
FSM	Finite State Machine
FWHM	Full Width at Half Maximum
GUI	Graphical User Interface
GP	Gross Positioner
IMU	Inertial Measurement Unit
KF	Kalman Filter
LA	Linear Actuator
LAN	Local Area Network

LED	Light Emitting Diode
LP	Low-Pass
LWR	LightWeight Robot
MRI	Magnetic Resonance Image
OR	Operating Room
OROCOS	Open RObot COntrol Software
OTS	Optical Tracking System
PI	Proportional Integral
POI	Point of Interest
PT	Probe Trajectory
PKM	Parallel Kinematic Machine
RC	Rotation Component
RMS	Root Mean Square
RGB	Red Green Blue
ROS	Robot Operating System
ROBOCAST	ROBOt and sensor integration for Computer Assisted Surgery and Therapy
RT	Real-Time
SBS	System Behaviour Supervisor
SEEG	Stereo ElectroEncephalography
TC	Translation Component
TP	Target Point
TCP	Tool Center Point
TRE	Target Registration Error
UI	User Interface
UKF	Unscented Kalman Filter
UML	Unified Modeling Language

1 INTRODUCTION

We've learned from experience that the truth will come out.

Richard Phillips Feynman
Cargo Cult Science – 1974

CONTENTS

1.1 Computer Assisted Surgery 1
1.2 Robotic surgery 2
1.3 Neurosurgery 4
1.4 Aim of the thesis and structure of the document 5

1.1 COMPUTER ASSISTED SURGERY

The surgical activity is a very complex task that is performed by humans in tough conditions: stress and, eventually, reduced working space can make the task to be more complex.

In order to assist the surgeon during the intervention, engineering research in the bio-medical field lead to the development of Computer Aided Surgery (CAS) techniques. This is a set of methods that can assist the surgeon starting from the pre-operative phase to the intra-operative phase, providing tools to check, during the intervention, the correctness of the procedure through visual feedback on the medical images or through augmented reality. This can be done using the diagnostic images, performed prior to the intervention on which the surgeon plans the surgical procedure, after a proper registration with the intra-operative reality.

This kind of CAS procedures were introduced in the beginning of the 1990's, but the basic concepts here used were already introduced in the surgical practice with different devices. One of these was the stereotaxic frame that was introduced in the beginning of 1900 [1]; it is a device that is fixed on the patient's head and allows the insertion of straight instruments (i. e. electrodes or probes for biopsies and localize drug delivery) using a cartesian Coordinate Frame (CF) defined by the frame itself, after properly mounting and aligning it with anatomical landmarks. This is based on the concept that the brain have a constant structure with respect to different subjects, and thus that the frame could be used to identify internal structure based on external landmarks, as if there is an atlas of the internal brain structure that is shared among all people.

The only way to get patient specific information of the brain was to use medical images. In the 1950's, the stereotactic frame was refined [2] and it was possible to take two orthogonal images of the patient wearing the frame with landmarks on the frame itself, in order to get information about the anatomy of the patient with respect to the position of the frame, in order to be able to identify internal points in 3D coordinates. Later, the idea of atlas was refined [3] by assuming that the brain can be reconstructed from a statistical atlas properly adapted using proportional factors measured from anatomical structures, giving the future possibility for rigid and non-rigid registration techniques of a generalized atlas. After that several different stereotactic frames were developed based on the polar coordinate system, giving an isocentric CF.

In 1973 the Computed Tomography (CT) was invented [4] and so it was the direct acquisition of 3D images. This increased the precision in the recognition of internal structure because the information of the third dimension is encompassed by this kind of images and not anymore inferred from two orthogonal images.

In the 1980's, the development of personal computers changed the paradigm of the CAS towards a frameless stereotaxis, in which the current intra-operative reality can be projected in the image space, opposed to the previous paradigm of projection of a point in the images on the intra-operative reality. This became possible through the use of localization systems, as passive arms [5] and triangulation systems such as cameras or, more in general, systems of emitters and receivers using different physical working principles such as electromagnetic and acoustic. The localizers can be used to register the intra-operative reality to the diagnostic images through homologous points [6], thus leading to the possibility of extended reality in which the images are properly displayed on a monitor, highlighting structures and information that are important for the procedure.

1.2 ROBOTIC SURGERY

In the last decades, robotic surgery increased its role in surgery. Computer and Robot Assisted Surgery (CRAS) is an extension of the concept of CAS in which there is the interaction with the environment. The use of robots in surgery is a valid tool to aid the action of the surgeon, providing an active support along with the increased number of information provided by CAS.

The Operating Room (OR) is a tough environment due to the high number of obstacles present, such as other devices and the surgical staff. Moreover, the environment is shared between the robotic manipulators and people, which increases the requirements on the safety of the procedure, in particular from the point of view of the patient.

The robotic device cannot substitute completely the surgeon, but it can be used exploiting its features in order to increase the possibility of the surgery and alleviate difficulties. In fact, a robot can do repetitive tasks accurately, remain in a fixed position and it can also be tele-operated, reducing human contact with the patient lowering the risk of infections. Moreover, it has been proven [7] that the use of a robotic assistant can increase the intervention accuracy and repeatability for instance when positioning surgical tools on target poses defined on medical images during the pre-operative planning.

In surgery, robots can be divided in two classes [8]:

SURGEON EXTENDER as tools that can extend the possibility of the surgeon to perform the intervention through a direct control, making possible the treatment of otherwise untreatable cases, i. e. using standard procedure.

AUXILIARY SURGICAL SUPPORT as tools that act as tool holding devices that work as an assistant, active if controlled by a computer or passive if static.

Using robot that are in contact with the patient can reduce the invasiveness of the intervention, allowing the reduction of the access point but maintaining a good dexterity inside the body of the patient for the intervention [9].

Attention must be paid during the design phase of a surgical robot, because it is sharing the environment with the surgical staff and the patient. In particular, the robot should leave a reasonably easy access to the patient site when in operation and must be easily and fast removable in case of emergency, as long as also the tools connected to it must be easily removable [10]. Purpose specific robots can be designed providing a fail-safe mechanical structure that encompasses some of the safety issues [11], such as constant pose in case of power loss. In some applications standard industrial robot were used in the surgical environment [12], also via proper modification of the internal sensors to increase the safety [13].

For robots and devices in general, in medical applications the safety issues are extremely important in order not to hurt the patient and the surgical staff [14]. Computer assisted tools can give a feedback on the current position of the robot device and notify the surgeon in case the robot is not where it is expected to be. The robot should have redundant systems for checking the safety of the procedure, in particular if they have an active role during the intervention. Surgeons must also be trained in the use of the devices in order to monitor and understand the behavior of the device and react appropriately if needed. The maximum speed of the robot should also be constrained in order to reduce the movement during the reaction time, either automatic or due to the surgeon intervention. It is important for a surgical robot that the eventual fail results in a controlled stop to a safe state in which the robot can be removed and the procedure ended manually, with the only exception of neurosurgery, in which the removal of the surgical tool from the robot can be critical if it is inside the brain of the patient [10].

In order to increase the safety of the procedure, the surgeon has to be asked for confirmation of the actions and movements of the robot prior to them, in order to make him/her aware of the next actions that will happen, and have his/her consensus.

1.3 NEUROSURGERY

Neurosurgery is a field in which the requirement for accuracy is particularly demanding. On the other hand, the landmarks on the brain structure can be difficult to be individuated in minimally invasive procedures and thus it is a field in which CAS systems play an important role, as well as robotic devices in the last decades [15]–[17].

The use of CRAS solutions in this field is of help to the surgeon because those system are faster than humans and also are not susceptible to fatigue and the stress that can arise in human operators during those interventions, which can last several hours.

For this reason, in the 1980's the first passive robot assistants developed in the or where retractable arms that were used to hold tools in place over the head of the patient in a time-reliable [18] way compared to a human operator which suffers of fatigue after a short time.

Surgical robots in neurosurgery can be divided in two categories: 1) passive and 2) semi-active [19]. Passive systems, as the Neuromate (Renishaw ltd., UK) [15], [20], [21], autonomously move to a predefined position defined on the pre-operative data (e. g. close to the entry point on the skull) before locking and powering off, then the probe is manually inserted by the surgeon. In devices that belong to the semi-active category, the surgeon can interact with the robot through a master handling device such as in the NeuroArm, developed for microsurgery and stereotactic brain procedures [22], [23]; in those categories there are also robots in which the probe can also be cooperatively driven (hands-on control), as in the ROSA™ (MedTech, France) system [24].

In this field of surgery, robotic devices are not yet widely used in practice [25], but they are increasing their role, also thanks to the research projects funded by public institutions, as the European Community in Europe, because the use of robotic arm can make the procedure to be more accurate and more reliable, as long as to carry out operations that would result in more complex procedures if performed in a traditional way, i. e. due to the stress for the operator and eventually the reduced working space.

Those projects aim to improve the performances of the procedure in terms of accuracy and effectiveness, creating a wider concept of the surgery with equipped ORs in which the information from navigation systems, medical images and the current reality provide the surgeon with a reliable platform to perform his/her task. In this picture the robot should act as a transparent reliable assistant under the control of the human operator that defines the specifications and the behavior, in terms of parameters and control modes, that the device must apply in the different phases of the procedure, integrating in the traditional procedures without heavily changing the habits of the surgeons; this will bring the surgery to better results, making the overhead, in terms of costs, training and added preparation time, acceptable.

1.4 AIM OF THE THESIS AND STRUCTURE OF THE DOCUMENT

The thesis here presented was aimed at defining a method to generate and handle a surgical workflow for robot-aided neurosurgery in which the device adopts its behavior, in terms of parameters and control modes, according to the phase of the intervention and user needs; also the autonomous movement of the robot was studied by developing and verifying algorithms to bring a tool, held by a robot, on a static target and also to track the movement of the target itself.

This problem was treated through the definition of a Finite State Machine (FSM), which coordinates the workflow of the surgical intervention and also a FSM for each device present in the OR. In the device FSM, the states correspond to the different possible control modes, along with the transition actions, while the global-workflow FSM collects the states of all the devices to accomplish that particular step of the surgical procedure and the custom parameters for each state. This system was implemented in a custom defined Robot Operating System (ROS) and Open RObot COntrol Software (OROCOS) architecture which interacts with a database to get the information about the states and parameters.

In neurosurgical intervention two main types of procedures can be identified: 1) keyhole and 2) open-skull. In the first case, i. e. in keyhole brain biopsy, the research activity was directed towards the convergence and accuracy analysis of an iterative targeting algorithm to place and orient a surgical probe on the skull of the patient using a robotic arm, guided by the information of the target defined by the surgeon on the medical images and the information about the pose of the tool from an optical tracking system that surveys the robotic arm. In the second case, i. e. during open skull surgery for epilepsy treatment with awake patient the brain moves due to breathing and heartbeat, so a robotic actuator which is holding a surgical tool in contact with the brain tissue must follow the target of the surgical procedure in order not to harm healthy tissues; studies on an algorithm for target following were carried out to test the performances of the algorithm and its compliance with respect to the neurosurgical requirements.

The work presented in this thesis is then divided in two chapters. The first, presented in chapter 2 on the facing page, describes the topics of high level control of a surgical procedure as long as the coordination of the robot devices and control modes is presented; the second, presented in chapter 3 on page 25, is about the different strategies used to bring a tool held by a robot on a target, the *targeting* procedure, and to keep it in position within an accuracy that satisfies the requirements of the surgical procedure if the target moves in space. Both these activities were integrated in the EU funded projects ROBOt and sensor integration for Computer Assisted Surgery and Therapy (ROBOCAST) FP7-ICT-2007-215190 [26], [27] and Active Constraints Technologies for Ill-defined or Volatile Environments (ACTIVE) FP7-ICT-2009-6-270460 [12], which both aim at defining an integrated robotic suite for neurosurgical interventions.

The preliminary part of these activities were started during my stay at the Medizingruppe of the Institut für Prozessrechentechnik, Automation und Robotik,[1] Karlsruher Institut für Technologie, Germany, while the proper development along with the experiments were carried out at the NearLab[2] of the Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy.

[1] http://rob.ipr.kit.edu

[2] www.nearlab.polimi.it

2 BEHAVIOR CONTROL

Life is like riding a bicycle. To keep your balance you must keep moving.

Albert Einstein
Letter to his son Eduard – 1930

CONTENTS

2.1 Description of the scenario 7
2.2 Workflow, states and transitions 8
2.3 Control architecture 10
2.3.1 General 10
2.3.2 Detailed description 11
2.4 Test scenario and experiments 16
2.4.1 Workflow 16
2.4.2 Experiments 17
2.5 Results 18
2.6 Discussions 22

The work presented in this chapter[1] describes a control architecture to manage the workflow of a surgical intervention, which changes the control mode and parameters for the devices present in the environment, i. e. input devices, environmental sensors and robots.[2]

The control architecture implemented is described in section 2.3 and a simplified scenario is then introduced in section 2.4, along with the tests performed.

2.1 DESCRIPTION OF THE SCENARIO

The architecture used here, showed in fig. 2.1, encompasses: 1) a robotic arm, 2) a master device and 3) external localization sensors (environmental cameras and tracking systems).

The selected robot is a LightWeight Robot (LWR) 4+ (KUKA Laboratories, Augsburg, DE) [30], which features 7 Degrees of Freedom (DoFs) that can be remotely controlled via a proprietary interface, the Fast Research Interface (FRI), that allows the user to get the internal sensors information and to provide motion commands with control

[1] from the contributions Comparetti *et al.* [28], [29]

[2] video demonstration at www.youtube.com/watch?v=x2Pkx31beJc

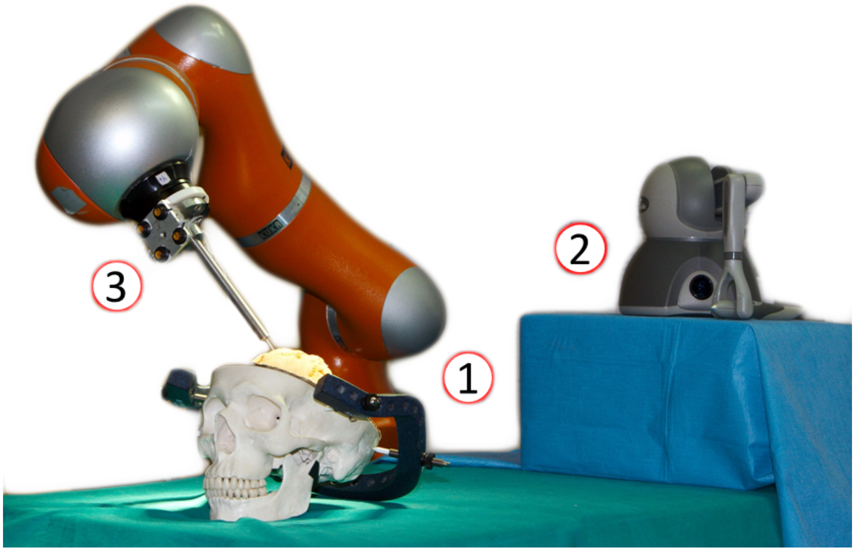


Figure 2.1: The setup of the scenario: 1. is the LWR actuator, 2. is the teleoperation master device and 3. are the markers for the OTS (not shown in the picture) .

frequencies up to 1 kHz, which is the frequency of the internal controller; the LWR also features torque sensors on each joint, which allow to measure the external torque on the joint itself, compensating the weight of the tool attached to the robot flange. The kinematic structure of this robot is anthropomorphic and, like an human arm, it has a spherical joint with three DoFs in the base (shoulder), a planar rotational joint with one DoF in the elbow and a second spherical joint with three DoFs in the flange (wrist).

The chosen master devices is the Geomagic Touch (formerly Sensable Phantom Omni – Geomagic, Morrisville, North Carolina), a small serial robot arm which can measure the motion of the handle in 6 DoFs and can provide force feedback in 3 DoFs (Cartesian translations).

The external localization systems used here are the OptoTrack Certus (NDI, Ontario, Canada), an Optical Tracking System (OTS) used as a supervisor of the system in order to measure and check the performances of the control architecture with a stated accuracy of 0.15 mm in a pyramidal working volume of about 25 m^3, and the Microsoft Kinect for XBOX 360 (Microsoft, Washington, U.S.), to track the human body segments.

2.2 WORKFLOW, STATES AND TRANSITIONS

A surgical procedure is composed of different steps in which the surgical staff executes different tasks. A robot which has to act as a surgeon assistant thus have to change its behavior according to the needs of the current step of the intervention. The possible control modes that can be used can be summarized in the following categories:

- Autonomous for brain electrodes insertion as in Stereo ElectroEncephalography (SEEG) (epilepsy invasive diagnostics);
- Cooperative with active constraints (limiting the operative space for safety purpose) for cortical stimulation or resection (epilepsy surgery, tumor removal);
- Tele-operated via master console for remote resection or disconnection (epilepsy surgery, tumor ablation).

Thus there can be the need to switch from one mode to the other to perform the surgical task. In [31], the robot interacts with the environment in order to get in contact with a target object and thus the robot needs then to switch from an unconstrained motion in space to a constrained motion with force control after the contact. For doing so, sensors information can be used to trigger the instant in time when to perform the transition between the two controllers; in particular the authors used proximity and force sensors to get the information about the contact with the target and thus to trigger the change of control mode.

The problem to be tackled during the switching phase is to guarantee the continuity of the robot trajectory without glitches; this is particularly important in robotic surgery application if the robot is holding a tool which is in contact with the patient.

A surgical workflow is composed by several tasks and the ordered list of those tasks represent the surgical procedure. In each task, a surgical robot that acts as an assistant needs to implement different control modes, according to the task itself. From that, there is the need to define an high level control architecture that is dedicated to supervise and guide the flow of the surgical procedure, defining the control modes that each of the devices should have during the procedure.

The control architecture here presented is a multi-layer architecture in which a FSM, called System Behaviour Supervisor (SBS), is defined to handle the surgical workflow and to coordinate the devices that have a dedicated FSM implemented in their controllers to handle the transition among the different statuses commanded by the high level architecture. In this way, the global FSM takes care of coordinating the events and the respective actions to be performed as long as they appear, while the device-specific FSMs are dedicated in handling the transition among the different control modes.

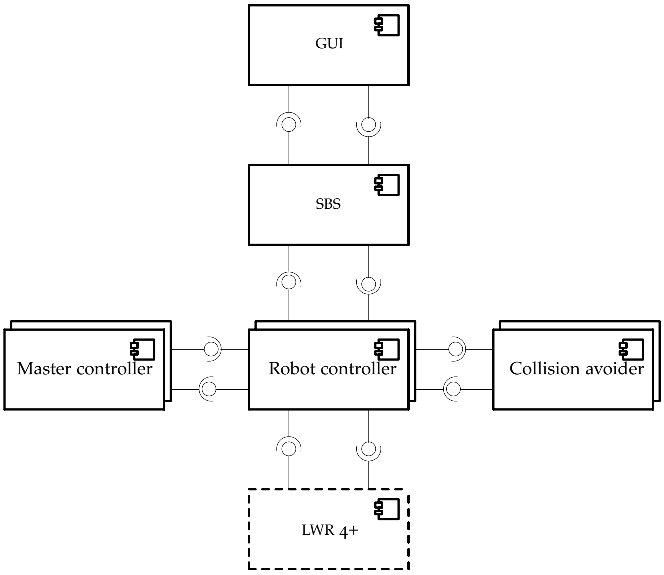


Figure 2.2: High Level Controller UML component diagram: it shows the main components to control the robot along with their connections; the ball represents the provider of information while the socket represents the receiver.

2.3 CONTROL ARCHITECTURE

2.3.1 General

In fig. 2.2 is presented an UML component diagram of the developed control architecture. It encompasses a Graphical User Interface (GUI) as the main interface with the user and it is aimed to control a robot to perform a surgical procedure (as the one that will be described in section 2.4). In this architecture there are two FSMs: 1) SBS for the high level control of the procedure and 2) Robot controller for the control of the robot device, while the other blocks represent the possible source of data for the robot movement in case of remote control or to avoid obstacles detected by the environment monitoring cameras.

The SBS is the FSM that is deputed to the high level control of the architecture throughout the intervention and it defines the *behavior* for each device, as the control mode and its parameter, given the current situation and step of the intervention. It retrieves the information about the workflow and the behavior in the current step from a proper DataBase (DB)[3] and it forwards the new parameters to the devices' FSM. Moreover, it polls the devices' controller to get information about their status and, in case of problems, it enables the *safe-state* and notifies the user about the potentially dangerous condition. The SBS first forwards the parameters and the new control mode to the devices' FSM and at the end it broadcasts an enabling trigger

[3] document based, *CouchDB* [32]

signal, which is detected by the devices controllers and marks the instant from which the new parameters and control modes must be used. The SBS requires a positive acknowledge from the receiver of its commands, and in case of a negative or absent response, the *safe-state* is commanded and the user notified.

The communication between GUI and SBS and also between SBS and the *Robot controller* is implemented using the Common Object Request Broker Architecture (CORBA) middleware [33]; it is a client-server communication which is used because the data flow on those channels have a limited bandwidth. The devices' FSMs act as servers, providing the methods to enable the different control modes and parameters, while the SBS is a client of each of those; in the client side, the middleware offers the possibility to implement a *timeout* functionality on each call to the server methods, which is used to detect failure in the communication between the modules and to enable the emergency procedure.

During normal operations, when a *switching* event is detected, the SBS is triggered and it loads the new parameters for the devices and it streams them to dedicated device, identified in fig. 2.2 by the sole *Robot controller* component. These events can be the interaction with the GUI or the contact between the robot and the operator, and the correct one is determined by the current status; i. e. the contact between robot and user is, in general, a potentially dangerous collision, while it is correct in case the robot is cooperatively controlled by the user (see section 2.4).

The architecture is implemented using Qt [34], Robot Operating System (ROS) [35], [36] and Open RObot COntrol Software (OROCOS) [37], [38] frameworks. Qt was used to implement the GUI and the SBS, ROS was used as interface with the devices and for streaming the data on the private LAN among the different components and OROCOS was used for the implementation of the controllers in Real-Time (RT) and for the interface with the LWR.

2.3.2 Detailed description

In fig. 2.3, the control schema in fig. 2.2 is presented adding details on the multi-components previously introduced. Here follows a description of all the modules.

2.3.2.1 *Collision avoider*

KINECT This module is the ROS interface with the Microsoft Kinect, in order to get the information about the position of the human body segments in space through the use of the OpenNI libraries [39].

The robot pose in the Kinect reference frame can be identified using a set of three markers made of paper, a circular shape with a diameter of 3 cm, in three different colors. They can be localized us-

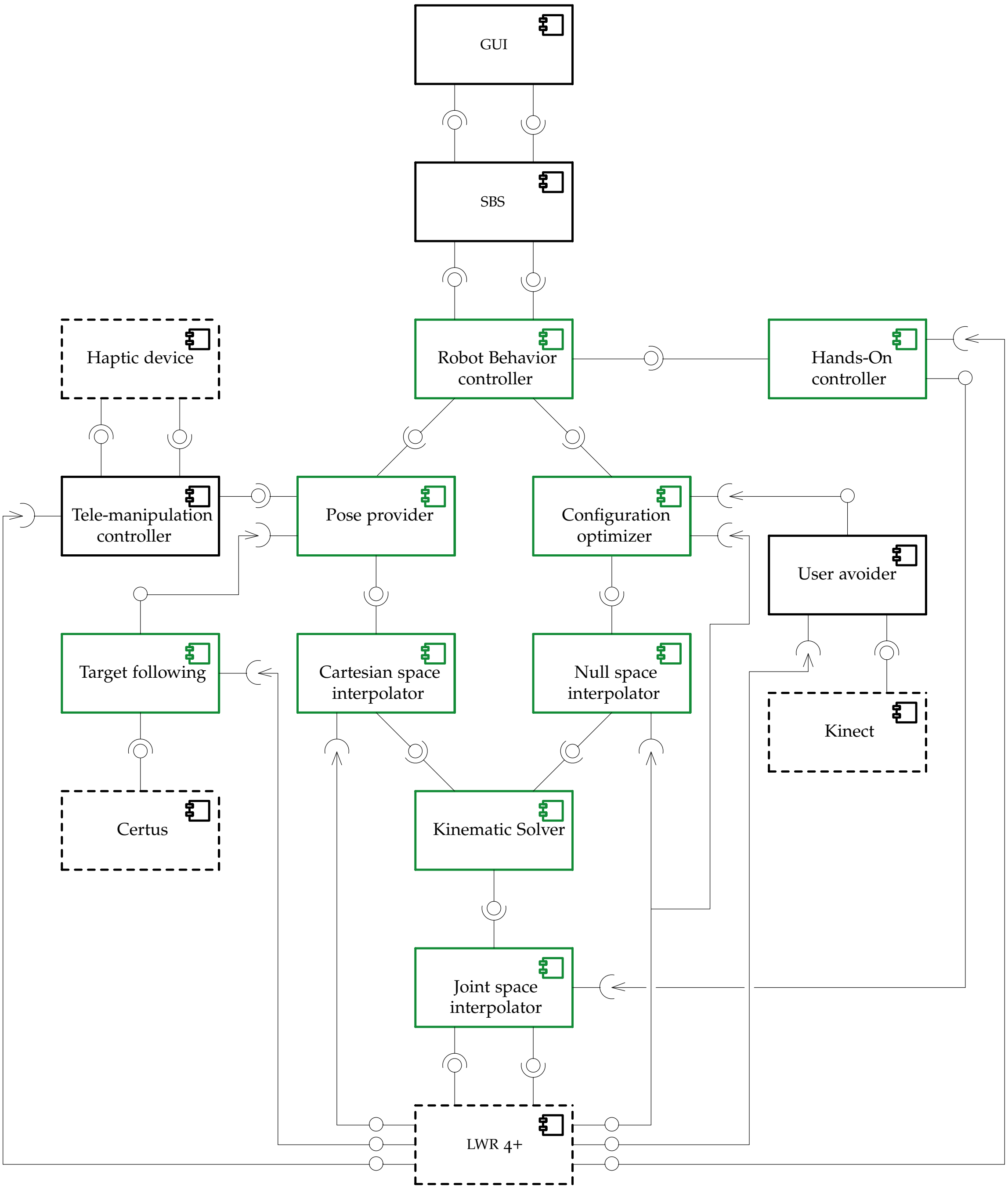


Figure 2.3: High Level Controller UML component diagram: the *Robot controller* is showed in detail, including the different components (highlighted in green) to enable the control modes that the robot can implement. Dashed components represent the devices and their respective controllers.

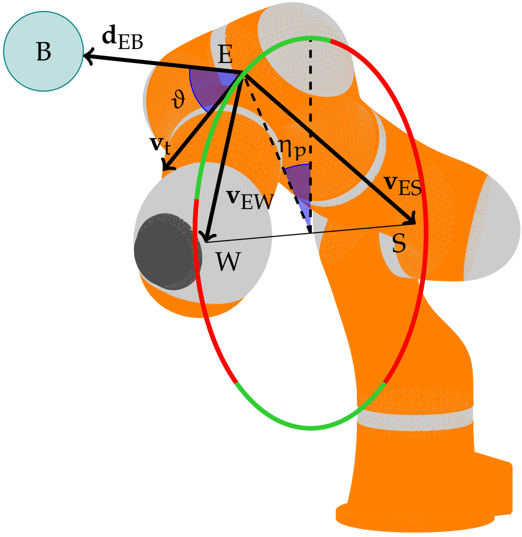


Figure 2.4: Null-space movement of the LWR. The circle represents the null-space along with the allowed and not allowed ranges for η_p highlighted in green and red respectively; B represents the obstacle, while S, E and W represent, respectively, the points of the shoulder, elbow and wrist of the kinematic representation of the robot.

ing the depth and RGB cameras of the device and from that a CF can be defined; by placing a set of markers on both the robot base and flange, using a calibration procedure [40], the robot base pose in the Kinect CF can be estimated.

USER AVOIDER The task for that component is to keep the tool on the target while avoiding the collision with the users present in the environment [41]. This is possible by exploiting the redundancy of the robot, through a movement in the null-space of the robot itself, which is represented by a circle that crosses the elbow and which axis is the vector from the shoulder to the wrist of the robotic arm [42], as shown in fig. 2.4.

For each joint configuration it is possible to define an angle $\eta_p \in [-\pi; \pi]$ which defines the position of the elbow on the circle described earlier, using as a reference the position of the elbow when it points upward. Due to the joint limits, some ranges for η_p may be not allowed in the different Cartesian pose of the work-space of the robot: if we define the i^{th} allowed interval as $\eta_p \in [\alpha_i; \Omega_i]$, moving within this interval doesn't change the pose of the wrist of the robot, while switching to an η_p in a different interval for the same Cartesian pose requires a reconfiguration of the joints with a movement of the wrist.

It is possible to calculate the vector tangent to the null-space circle $\mathbf{v}_t$ by using eq. (2.1)

$$\mathbf{v}_t = \mathbf{v}_{ES} \times \mathbf{v}_{EW} \tag{2.1}$$

where $\mathbf{v}_{ES}$ is the vector from the elbow pose to the shoulder and $\mathbf{v}_{EW}$ is the vector from the elbow pose to the wrist (see fig. 2.4). By knowing the pose of the body segments in the robot CF through the calibration, it is possible to compute the distance from the robot elbow to the human body $\mathbf{d}_{EB}$. If this distance is below a safety threshold d_{safe}, it is possible to calculate a new value for η_p by using eq. (2.2)

$$\eta_p^{t+1} = \begin{cases} \eta_p^t + k \cdot \frac{d_{safe}}{d_{EB}} & \text{if } (-1 < \cos\vartheta < -\bar{c}) \\ \eta_p^t & \text{if } (-\bar{c} < \cos\vartheta < \bar{c}) \\ \eta_p^t - k \cdot \frac{d_{safe}}{d_{EB}} & \text{if } (\bar{c} < \cos\vartheta < 1) \end{cases} \tag{2.2}$$

where $d_{EB} = \|\mathbf{d}_{EB}\|$, k is the proportional gain,[4] ϑ is the angle between $\mathbf{d}_{EB}$ and $\mathbf{v}_t$ and $\bar{c}$ is a threshold;[5] within this threshold, the value of η_p is not updated to avoid instabilities in case $\mathbf{d}_{EB}$ is almost orthogonal to $\mathbf{v}_t$, which causes $\cos\vartheta$ to change frequently its sign due to the noise in the measurement of $\mathbf{d}_{EB}$.

2.3.2.2 *Master controller*

HAPTIC DEVICE This module is the ROS interface with the haptic device, in order to get the information about the current pose of the master device and send the information about the force feedback.

TELE-MANIPULATION CONTROLLER This module is the OROCOS interface that, given the current pose of the robot and the current pose of the master device, computes the target pose for the robot, taking into account proper scaling factors defined by the user to increase the accuracy of the movement [43]. Moreover, it implements the mapping of the CFs from the master side to the robot in order to have the desired correspondence between the movement of the user's hand and the motion replicated by the robot.

2.3.2.3 *Robot controller*

ROBOT BEHAVIOR CONTROLLER This OROCOS module implements the FSM of the robot device. As long as the SBS triggers the new status, the module takes care about the transition phase from the current state to the new one, by properly enabling the different functionalities and setting the initial conditions of the controllers on the other modules.

[4] in the case here presented, k was empirically set to 0.05

[5] in the case here presented, $\bar{c}$ was empirically set to 0.03

HANDS-ON CONTROLLER During the Hands-on mode (cooperative) this OROCOS module takes as input the current pose and external torques measured from the joints' torque sensors, both expressed in joint space, and, as long as the external torque of at least one joint exceed its a joint-specific threshold (meaning that the user is pushing on the robot body) a contact is considered detected; those torques are used by the admittance controller [44] in eq. (2.3)

$$j_i^{t+1} = j_i^t + G_i \cdot \tau_i^t \tag{2.3}$$

where $i \in [1, 7]$ is the joint index, t is the time, τ is the external torque, G_i is the constant gain (joint specific) of the controller, j_i^t is the current joint position from the encoder readings and j_i^{t+1} is the desired value for the joint position.

POSE PROVIDER During autonomous movements and tele-operation, this OROCOS module gives the target in Cartesian space (in the robot base CF). According to the settings from the *Robot Behavior controller*, this module can act as a provider of constant target poses for autonomous movements towards the target itself or it can simply provide the target pose coming from outer modules as the *Master controller* or the *Target following*.

INTERPOLATORS This OROCOS module, given a target in the task space,[6] it calculates the via points from the source to the target with a trapezoidal velocity profile, sampled at the robot cycle frequency; on the rising and falling edge of the velocity profile, the acceleration is kept at the maximum allowed value, until the velocity reaches its maximum. This module is then smoothing input steps, in the task space, by defining the fastest parabolic trajectory that reaches the target with zero final velocity.

KINEMATIC SOLVER This OROCOS module calculates the analytic version of the inverse kinematics for the LWR robot [45], using the null space variable as a constraint for the solution of this redundant robot; this gives the possibility to calculate the new joints in case a new Cartesian pose is provided, as well as the value for η_p, the redundancy parameter.

CONFIGURATION OPTIMIZER This OROCOS module takes care of handling the redundancy of the robot[7] by constraining the possible configurations of the robot itself. Based on the needs of the step which is being executed, there are three possibilities for the null-space variable η_p [42]: 1) keep the current value, or 2) keep the null space in the

[6] either joint, Cartesian or null space

[7] the LWR features 7 DoFs

center of the current allowed interval for η_p, or 3) use the value that is calculated by the *Collision avoider* module to keep the robot away from the users.

CERTUS This ROS module is used as an interface with the OptoTrack Certus, in order to detect the position and orientation of set of LEDs that define a rigid body called Dynamic Reference Frame (DRF) and to make the data to be available on the LAN, to be used by the target following to calculate the motion of the rigid bodies.

TARGET FOLLOWING This OROCOS module, given the current pose of the robot and the desired target measured by the OTS, calculates the robot pose that allows the Tool Center Point (TCP) to be on the target that moves in space, minimizing the tracking error between the desired and current pose. Its behavior is better described in section 3.2.

2.4 TEST SCENARIO AND EXPERIMENTS

2.4.1 Workflow

A simplified scenario was drawn for the current chapter that foresees the following five steps:

1. Move the robot autonomously to the home position (*Homing*);

2. Move the robot autonomously close to the patient (*Autonomous*);

3. Move the robot to the target cooperatively guided by the surgeon (*Hands-on*);

4. Move the robot according to the input from a master device (*Tele-operation*);

5. Keep the robot in the current pose and terminate the procedure (*Steady*).

This represents an example of a possible surgical intervention where the surgeon controls the robot both manually and through a joystick interface.

In the different steps the behavior for the devices in the environment is different, thus it is fundamental to switch from one behavior to another during the execution time without stopping the devices or having unpredictable movements due to a mis-handled transition, i. e. bad settings in the initial parameters of the controller.

The trigger to switch between steps is based on events like pressing buttons on a proper User Interface (UI) or touching the devices and, for doing so, the surgeon is provided with an interface that allows

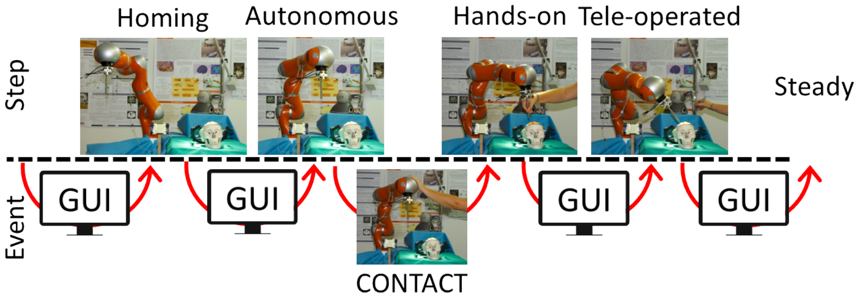


Figure 2.5: Animated schema of the possible steps and transitions event: as long as an event is detected, the step goes from the current to the following one, as indicated by the red arrow.

the control of the devices: as long as the surgeon requires to switch to the following step, (s)he creates an event by pressing a button on the GUI or by touching the robot. fig. 2.5 shows the user interaction for the depicted workflow.

In the GUI is present a button that goes to the next step in the list, and, in case of the event towards the *Hands-on* step, the trigger event can be raised by the surgeon by touching the robot itself: in fact, the user is allowed to touch the robot only during the cooperative mode, while in all the other steps, a contact between user and robot have to be considered as a faulty condition and have to raise a warning event to the user and put the robot in a safe state and, eventually, move it away.

2.4.2 Experiments

In order to test the behavior of the devices during the transition between different states, in the scenario described in section 2.1 the OptoTrack Certus OTS, was used as a supervisor of the system in order to measure and check the performances of the control architecture by measuring the pose of a DRF placed on the robot flange, to detect eventual residual movements that can happen during the switching phase. In the tests, all the devices and modules in the control architecture were sharing the same clock in order to have consistent timestamps among all of them; the components and the robot were running at 100 Hz.

An experiment is divided in six slots with five transitions among the following ordered list of states: Steady, Homing, Autonomous, Hands-on, Tele-operation, Steady. All the transitions are triggered via the GUI, apart from the transition from Autonomous to Hands-on which is triggered by the contact of the user with the robot as in fig. 2.5. This experiment was repeated for 19 times.

During the experiment, the following data were acquired:

- Pose of a DRF fixed on the robot flange, from the OTS;

- Cartesian pose of the LWR, from the forward kinematics of the robot encoders readings.

All those data were analyzed to check the continuity over time of the data, according to this schema:

1. 19 trials were acquired;
2. for each trial, 5 transition events are triggered;
3. for each event, three time-windows of 0.1 s are selected before, during and after the event itself;
4. within each time-window, the recorded signal was averaged to remove the noise;
5. those three values for each event were used to define three populations of 19 samples each.
6. those three populations, were statistically compared using the One Way ANOVA test with Bonferroni adjustment and $\alpha = 0.05$, in order to check if the differences among them are statistically significant;

So globally, five groups of three populations of 19 samples are obtained and compared, to check for statistically significant differences. The same process was also applied on the derivatives of the signals, up to the 3rd order, computed using a First-Order Adaptive Window (FOAW) derivative filter [46].

2.5 RESULTS

In fig. 2.6 an example of the raw data from the OTS is shown, along with the instant in which a transition happened and the contact between human and robot was detected. It can be seen that, after the third transition, the *Hands-on* mode is enforced and the motion of the LWR is following the torques provided by the user.

figs. 2.7 and 2.8 show an example of the three derivatives of the signal recorded from the OTS using a time-window of 0.1 s. In fig. 2.7a it can be noted that the first derivative is close to 0, while the derivatives of higher order present a bigger order of magnitude due to differentiation. Statistical difference was found in the data of the Cartesian pose of the LWR, measured from the robot encoders readings, as shown in fig. 2.8.

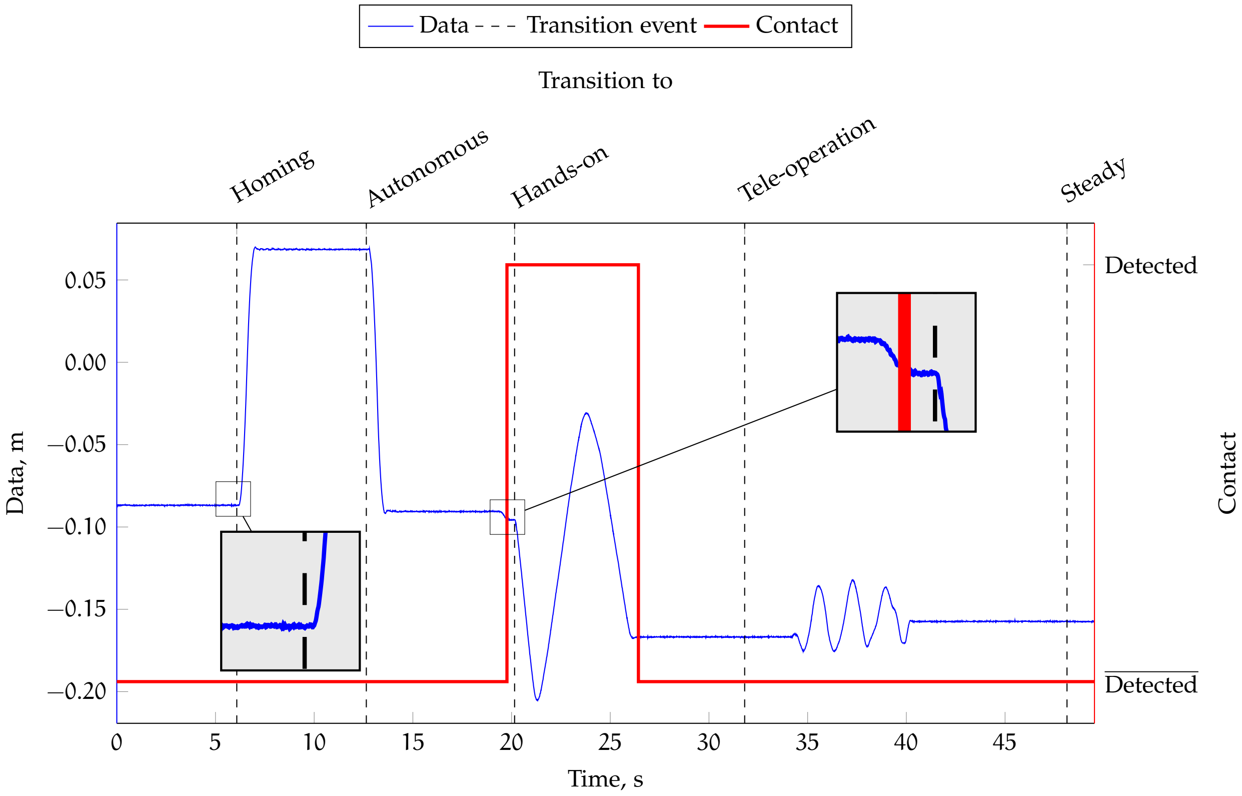
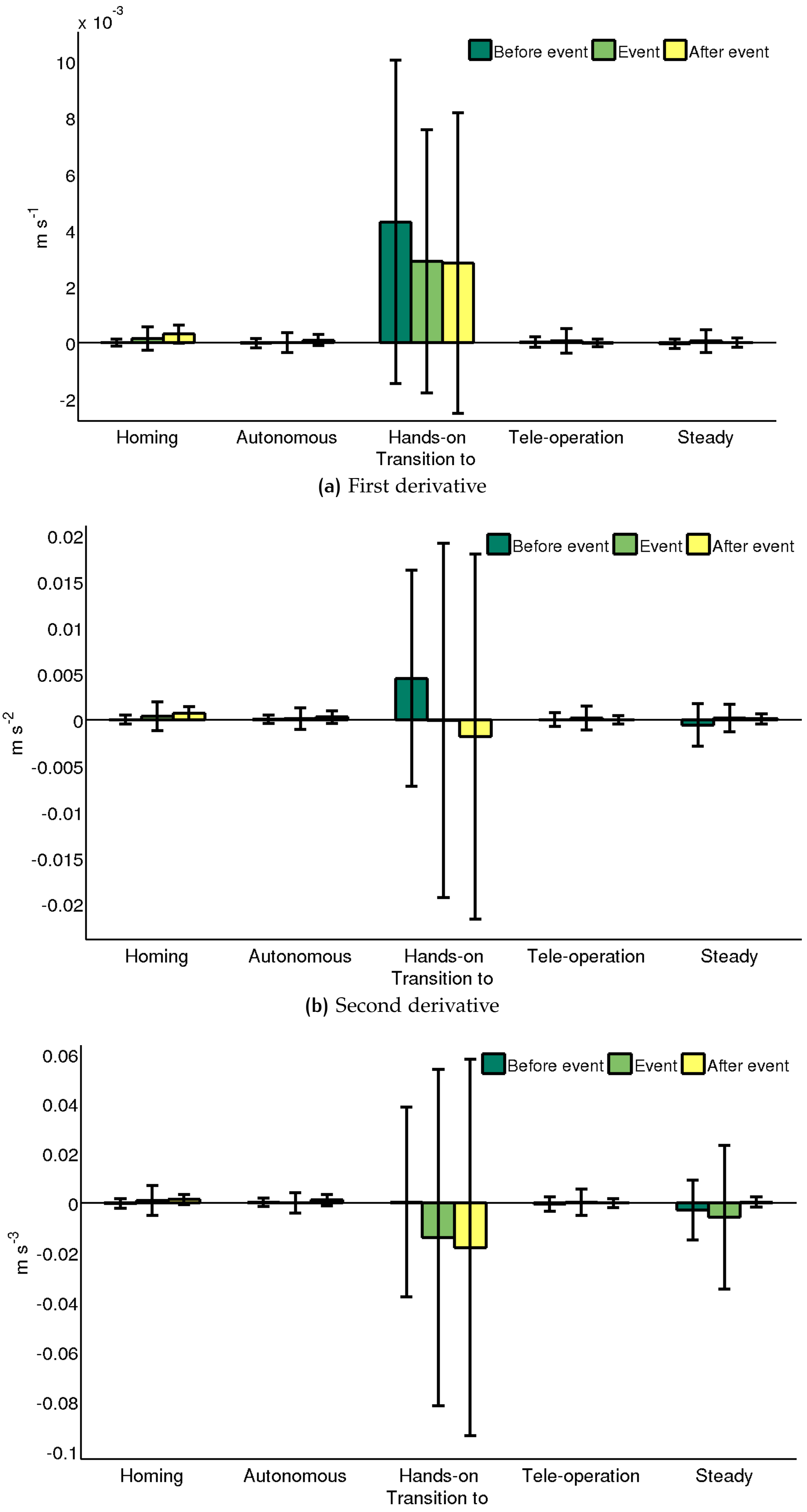


Figure 2.6: Example of the recorded data: the thin solid line is the data of the absolute x axis measured by the OTS, the thick solid line is the detection of the contact between human and robot and the dashed vertical lines are the instant in which the transition occurred; the data are also zoomed-in to highlight the trajectory during the transitions.

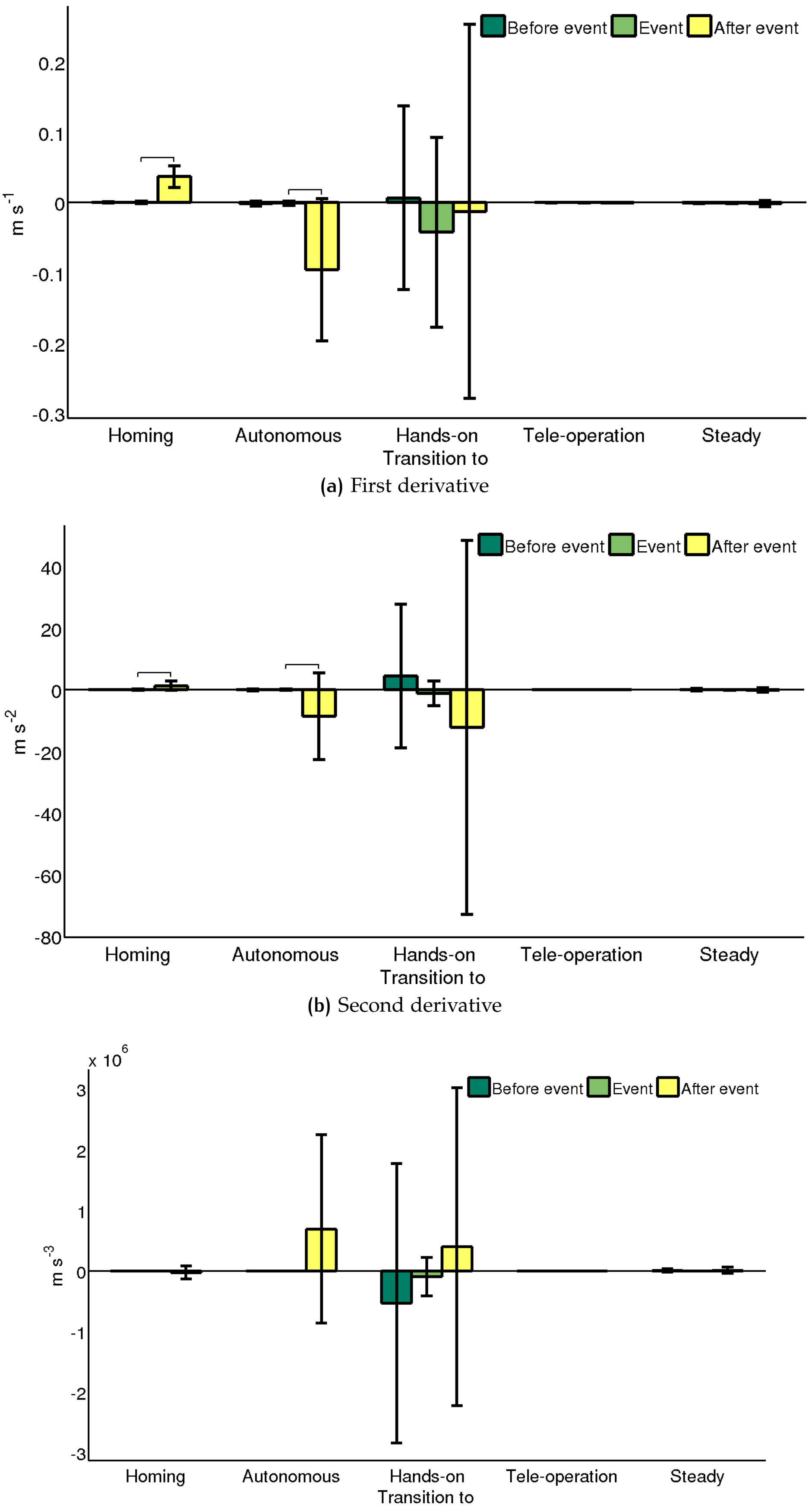


(a) First derivative

(b) Second derivative

(c) Third derivative

Figure 2.7: Data of the x axis of the OTS measurements with a time window of 0.1 s: the mean value and the standard deviation are showed in the interval around the event.



(a) First derivative

(b) Second derivative

(c) Third derivative

Figure 2.8: Data of the x axis of the robot pose with a time window of 0.1 s: the mean value and the standard deviation are showed in the interval around the event and horiziontal lines indicate a statistical difference among the populations.

2.6 DISCUSSIONS

In this chapter, a flexible, scalar and modular high level architecture to manage the switching among different control modes for a robot manipulator was presented. The need for this kind of architecture is in application in which the user is in close contact with the robot and interacts with it, making the scenario dynamically changing during the execution of tasks.

The chosen approach is the definition of a workflow for a simplified surgical procedure in which the task is to reach a point close to the patient in the target area of the surgery and then to perform the surgical task using the robot as a slave device that replicates the surgeon's movements. For this reason, the user is provided with an interface to the control architecture that, based on GUI and sensors inputs, defines the correct behavior for the devices in the scenario.

In [31] the robot was equipped with external proximity sensors and force sensors to obtain a smooth transition from free motion to contact with the target in a four-stages transition towards the contact; during this transitions, the initial conditions of the following controller are adapted to obtain continuity from the previous state, thus obtaining a reduced impact force with the target.

In a complex and dynamic environment like the operating room, it is important to have the possibility to have a control system that can modify and adapt the behavior of the devices according to user inputs, events and sensor data representing the different steps of the intervention, because during the surgical procedure the tasks for the robot changes and thus the control mode needs to be adapted to cope with the new situation, in order to be used by the surgeon to accomplish his/her task.

The purpose of the work performed in this chapter is to ensure the continuity of the robot trajectory during the transition between states, without unpredictable shaking of the robot flange. Results showed that the robot flange, and then also the tool connected to it, is kept in a steady position during the transitions. This factor is mandatory for a robotic application in which the robot is directly interacting with the patient organs through the contact between the tissues and the tool carried by the robot, in order to prevent damages to the patient due to unpredictable movements. During the transition from Autonomous mode to Hands-on mode, an higher variability is present on all the data due to the fact that, during the first part of the contact the user is acting against the robot, which is commanded to keep its current position but, due to the mechanical compliance in the geometry of this kind of serial robots, this action can cause a small bending in the structure, causing an increased variability of the derivative in this transition (see detail in fig. 2.6); this, however, doesn't affect the performance of the schema because this transition happens far from the patient and it is under the control of the user. When switching to Homing and to Autonomous mode, statistical difference was found in the data from the LWR encoder readings (see fig. 2.8), due to the fact that the robot starts its movement as long as it receives the event (see detail in fig. 2.6), accelerating at the maximum acceleration allowed by the interpolator in Cartesian space, increasing the variability of the data; the same statistical difference was not revealed by the tracking system because the noise on this data is higher. Using a time-window shorter than 0.1 s in the analysis of the data (see section 2.4.2), on the one hand it reduces the effects of the motion of the robot after the transition has been triggered, but on the other hand the contribution of the noise is more relevant.

3 TARGET IN NEUROSURGERY

> Your task is not to foresee the future, but to enable it.
>
> Antoine de Saint Exupéry
> The Wisdom of the Sands – 1948

CONTENTS

3.1 Target approaching 25
3.1.1 Introduction 25
3.1.2 Accuracy evaluation 28
3.1.3 Convergence evaluation 44
3.2 Target following 56
3.2.1 Introduction 56
3.2.2 Methods 58
3.2.3 Results 62
3.2.4 Discussions 73

This chapter presents the works related to *targeting* procedures; those procedures are a set of control strategies that allow a tool carried by a robot to reach a target; in particular this chapter is divided in two sections:

TARGET APPROACHING is the action to put the tool on a target that doesn't move in space[1] (see section 3.1).

TARGET FOLLOWING as the action to keep the tool on a target that moves in space (see section 3.2 on page 56).[2]

3.1 TARGET APPROACHING

3.1.1 Introduction

During keyhole neurosurgical procedures, straight probes are introduced inside the brain, where suitable trajectories are planned on pre-operative medical images, i. e. for Deep Brain Stimulation (DBS), SEEG and biopsies [51].

[1] from the contributions Comparetti *et al.* [47]–[50]

[2] video demonstration at www.youtube.com/watch?v=gpf-D-6z3bI

Keyhole neurosurgery is a highly demanding surgical procedure since functional areas of high importance and, most importantly, blood vessels have to be avoided during probe advancement inside the brain. Robotic systems avoid maneuvers with the stereotactic frame [52] or fixed grid-based trajectories, and have already been used to automatically position and insert such straight probes into the brain [19], [25].

When using a robot, the patient's medical images and the preoperative plan must be registered with the robot reference frame in order to perform the planned probe targeting. This can be achieved using a laser pointer [53] and fiducials placed on the patient's head [13] or an ultrasound localization systems [20], [54]. OTSs can also be used for guiding the surgical robotic arm (e.g. for total hip replacement [55], for neurosurgery [56] and for bone ablation [57]). Integration of the robot and external localization systems requires a calibration procedure in order to establish a common reference frame. Hand eye calibration allows mapping sensor-centered measurements into the robot/world frame [58]. If an external sensor, e. g. an optical localizer, is used, the calibration procedure estimates the geometrical transformation between its CF and the robot CF. Such calibration [59] can be performed with closed form solutions [60] or with iterative optimization approaches [40], [58], [61].

Targeting accuracy requirements depend on the clinical application and it is necessary to distinguish between technical and clinical accuracy achieved. The technical accuracy of frame based stereotactic procedures is about 0.05 mm, while frameless stereotaxy has a technical accuracy worse than 0.5 mm. In both cases, the accuracy reached in clinical practice is far worse than the technical accuracy.

In radiosurgery the clinical targeting error, i. e. residual distance between the beam and the target tissue, is reported to be 1 mm [62] and it is achieved using serial robots. In orthopedic procedures, where the orientation of knee cutting guides influences the final prosthesis components orientation, ±3° of orientation error in the frontal plane guarantee a positive outcome of the procedure [63]. In spine surgery, pedicle screws can be inserted into the spine using a miniaturized Stewart Platform, which allows achieving 0.1 mm of technical targeting accuracy [64] and deviations of 1 mm in clinical trials [65]. In neurosurgical keyhole procedures [51], where access to the brain is about 5 mm in diameter, the clinical positioning accuracy of straight probes at the entry point has to be below 1 mm [66]. The same targeting accuracy is required in ear, nose and throat interventions [67], where robotic drilling guides can be used to perform cochleostomy [68]. In table 3.1 technical and clinical accuracy values of commercially available robotic systems are reported.

OTSs were also used to correct the pose of the robot when deviations between the actual and the planned position were detected, tracking

Table 3.1: Surgical robots accuracy. The values of the accuracy in clinical trials and laboratory tests are reported.

Robot	Clinical accuracy	Technical accuracy
SpineAssist	1 mm [65]	0.1 mm [64]
Neuromate	(0.82 to 2.9) mm [69]	–
Robodoc	1 mm [70]	–
Cyberknife	1.1 mm [71]	–
SABiR	–	0.4 mm [72]
Pathfinder	1 mm [73]	(0.5 to 2.7) mm [13], [69]
MAKOplasty	3° [74]	–
ROSA	–	(0.9 ± 0.3) mm (specs)

the robot base and the tool relative to the patient [75]–[77]. In [75], the tracking system maps the position of the target to the robot reference frame by correcting the position of the robot and updating calibration parameters by means of a Kalman filter. In [77], the control of the robot position and orientation is implemented using Proportional Integral (PI) controllers. Residual targeting errors are approximately twice as big as the localization error.

Tracking systems performance for robot control was also improved by adding inertial measurements units [78] in order to increase the control robustness with respect to marker occlusions, to compensate for delay of the optical system and to reduce noise.

Serial robots generally have a large workspace and their absolute positioning accuracy is low because of modelling errors in the kinematics. On the contrary, parallel robots have greater accuracy, but they suffer from a limited workspace relative to their footprint. Using robots connected in series, having a redundant number of DoFs, allows performing the surgical task with a more flexible robot configuration in the OR environment. A multi-robot approach for neurosurgery applications was already proposed in the Evolution I [79], designed for micro-neurosurgical and micro-endoscopic applications. A Parallel Kinematic Machine (PKM) was coupled with an articulated mobile platform achieving 20 µm as positioning accuracy. Despite the great targeting accuracy reached, the research was dismissed since the developed system was too cumbersome and expensive.

As in the aforementioned papers, the approach here presented combines navigation data via optical sensors and robotic assistance for keyhole neurosurgery using three robots connected as a multiple kinematic chain, part of a robotic suite for keyhole neurosurgery, named ROBOCAST. A biopsy probe is positioned by the modular robotic system at the planned entry point on the patient skull, and it is then inserted in the brain via tele-operation control through a haptic device. The previous work, i. e. the general ROBOCAST system architecture and pre-operative planning algorithms, are described in [27], [80].

In order to perform the targeting, an OTS is used to detect the position of the robots, of the patient and of the surgical tool held by the robot, allowing the procedure navigation, similarly to [75], [77]. The OTS makes it possible to check the accuracy of the task during its execution and for iteratively correcting the pose of the robot when deviations between the actual and the planned pose are detected [49], [75]. Such deviations happen because serial robots have low absolute accuracy due to an inaccurate model of the robot [13], [69] and also because the calibration between the robot space and the measurement system is affected by errors [40]. In the approach here described, the external sensor allows iterative corrections of the pose of the robots in case errors between the desired and the actual poses of the surgical tool are detected., by weighting the correction movement differently in case the error (in terms of position and orientation error) is increasing or decreasing in magnitude over time.

The method proposed can be used with inexpensive, low resolution manipulators, which would help reduce the cost of a medical robotic system, since the residual errors can be reduced by measurement and correction. The main contribution of the work performed in this section is therefore to prove the clinical applicability of a navigated robotic approach for neurosurgical intervention. Only combining the external localization system, the automatic robotic approach can satisfy the application accuracy requirements.

The original concept behind the iterative targeting algorithm was proposed in [48], while here the suitability for clinical applicability has been proved through extensive tests, performed under operating condition settings, since the obtained targeting accuracy satisfies the clinical requirements. The innovative targeting algorithm presented has potential applications for surgical robotic systems already present in the market.

3.1.2 Accuracy evaluation

3.1.2.1 *Materials and Methods*

THE SYSTEM The ROBOCAST system is a robotic chain of three robots, totalling 13 DoFs; the system [27] encompasses (see table 3.2 for details):

- Gross Positioner (GP), (PathFinder, Prosurgics Ldt, UK); a serial 6 DoF arm, which is used to approach the patient head [81];

Table 3.2: Robots characteristics.

Robot	Type	# DoFs	Accuracy	Velocity	Workspace
GP	serial	6	0.5 mm [13]	50 mm/s*	0.75 m × 0.75 m × 0.75 m*
FP	parallel	6	< 0.1 mm	1.3 mm/s, 4.3 °/s	40 mm × 40 mm × 10 mm, 12° × 12° × 12°
LA	linear	1	8 µm	2 mm/s	110 mm

* Fixed by ROBOCAST specifications

- Fine Positioner (FP), (SpineAssist, Mazor, Israel); a parallel 6 DoF miniaturized PKM, used to further correct the targeting. It is rigidly connected to the GP via a custom-built quick release interlock;
- Linear Actuator (LA); a 1 DoF bespoke piezo-actuator, which makes the biopsy linear probe advance through a tele-operated haptic interface (Omega, Force Dimension, Switzerland) [82]. The LA is attached to the FP end effector upper plate.

The Optotrack Certus OTS (NDI, Ontario, Canada), with 0.15 mm stated accuracy, is used for tracking the overall robotic chain. DRFs, rigid bodies composed of four active markers each, are attached to all the bases and the end-effectors of the robots. The tracking system is interfaced with the robot high level controller through a Sensor Manager application [83], which is based on the IGSTK framework[3] (Kitware Inc., Clifton Park, New York, USA).

The desired target in the brain and the entry point on the patient skull are defined during the planning. In the intra-operative scenario, the surgical tool is firstly aligned onto the desired trajectory at the entry point using the GP and the FP and then inserted inside the brain towards the planned target, using the tele-operated LA.

SPATIAL RELATIONS Calibrating the system means to establish a common CF among the robots (GP, FP and LA) and the optical sensor. To compute the spatial transformations (see fig. 3.1) among the forward kinematics of the robot, the joint angles encoders ($\mathbf{B}_{\mathrm{GP}}$, $\mathbf{B}_{\mathrm{FP}}$ and $\mathbf{B}_{\mathrm{LA}}$) and the OTS ($\mathbf{A}_{\mathrm{GP}}$, $\mathbf{A}_{\mathrm{FP}}$ and $\mathbf{A}_{\mathrm{LA}}$), the approach described in [40], [58] was used for GP and FP calibration: $\mathbf{X}_{\mathrm{GP}}$, $\mathbf{Y}_{\mathrm{GP}}$, $\mathbf{X}_{\mathrm{FP}}$ and $\mathbf{Y}_{\mathrm{FP}}$ are the transformations computed during the calibration and represent the transformations between the DRFs in the optical CF and the robots internal CF (see fig. 3.1), in particular:

[3] www.igstk.org

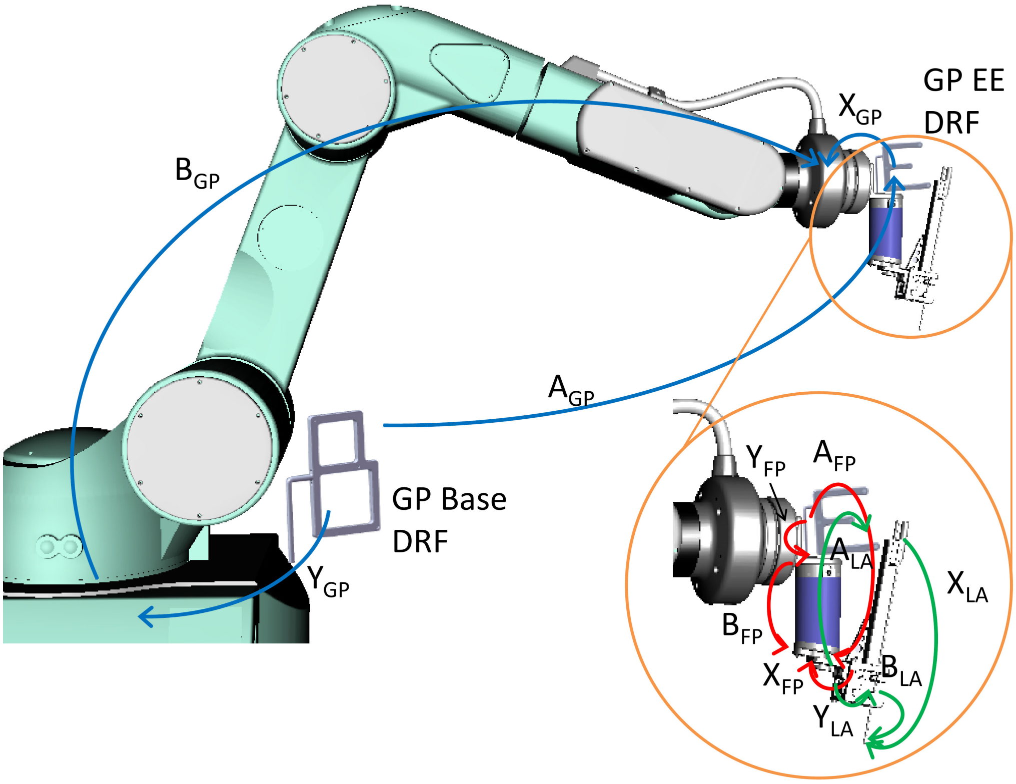


Figure 3.1: Robots calibration schema. The spatial transformations between the robots internal CFs and the DRFs are represented.

- $\mathbf{X}_{GP}$ is the transformation between the GP end-effector internal CF and the origin of the DRF attached to the GP end-effector;
- $\mathbf{Y}_{GP}$ is the transformation between the GP base internal CF and the origin of the DRF attached to the GP base;
- $\mathbf{X}_{FP}$ is the transformation between the FP end-effector internal CF and the origin of the DRF attached to the FP end-effector;
- $\mathbf{Y}_{FP}$ is the transformation between the FP base internal CF and the origin of the DRF attached to the FP base;
- $\mathbf{X}_{LA}$ is the transformation between the LA end-effector internal CF and the origin of the DRF attached to the LA end-effector;
- $\mathbf{Y}_{LA}$ is the transformation between the LA base internal CF and the origin of the DRF attached to the LA base.

In order to perform the calibration, the GP working volume was sampled acquiring 28 poses (in a sphere of 173.21 mm radius). In order to sample the FP working volume as well, 28 poses were acquired (in a sphere of 7.86 mm in radius). Calibration errors were computed on 15 poses not used for calibration.

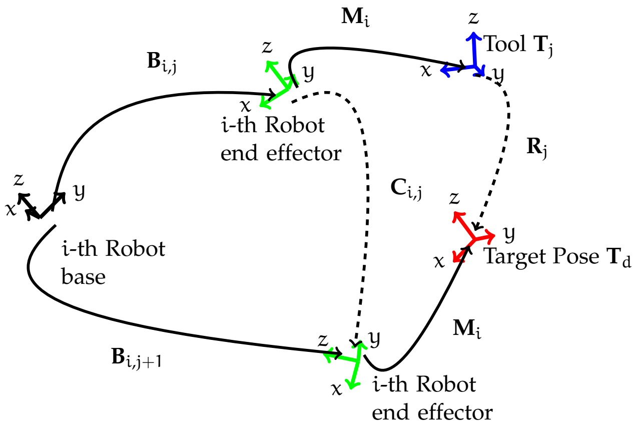


Figure 3.2: Spatial transformations used during probe targeting. $\mathbf{M}_i$ is the spatial transformation between the Tool and the robot end-effector: for the GP robot it is $\mathbf{M}_{GP} = \mathbf{X}_{GP}^{-1} \cdot \mathbf{Y}_{FP} \cdot \mathbf{B}_{FP} \cdot \mathbf{X}_{FP}^{-1} \cdot \mathbf{Y}_{LA} \cdot \mathbf{B}_{LA}$ and for the FP robot $\mathbf{M}_{FP} = \mathbf{X}_{FP}^{-1} \cdot \mathbf{Y}_{LA} \cdot \mathbf{B}_{LA}$.

In order to calibrate the LA, the probe was advanced for 100 mm (backward and forward with 5 mm spacing) and the 3D coordinates of the tip position were sampled using a custom built divot, equipped with a DRF. Principal Component Analysis was applied in order to compute the orientation of the line approximating the probe advancement in the 3D space.

Probe targeting brings the probe (Tool) tip CF $\mathbf{T}_j$ in the desired target pose $\mathbf{T}_d$ (2 cm outside the patient's skull entry point), as planned in the pre-operative phase. From there, the probe is advanced by the surgeon using the haptic interface [27]. In order for the Tool $\mathbf{T}_j$ CF to reach the $\mathbf{T}_d$ CF, the i^{th} robot ($i = GP, FP$) has to change its pose from $\mathbf{B}_{i,j}$ to $\mathbf{B}_{i,j+1}$ (fig. 3.2), where j is the movement iteration index.

If $\mathbf{M}_i$ represents the transformation between $\mathbf{T}_j$ and the i^{th} robot end-effector CF, the transformation to be applied to the i^{th} robot end effector $\mathbf{C}_{i,j}$ is:

$$\mathbf{C}_{i,j} = \mathbf{M}_i \cdot \mathbf{R}_j \cdot \mathbf{M}_i^{-1} \tag{3.1}$$

where $\mathbf{R}_j$ is the transformation between the probe tip CF ($\mathbf{T}_j$) and the target pose ($\mathbf{T}_d$). Therefore, the new robot pose $\mathbf{B}_{i,j+1}$ is:

$$\mathbf{B}_{i,j+1} = \mathbf{B}_{i,j} \cdot \mathbf{C}_{i,j} \tag{3.2}$$

Note that this transformation is independent of the base CF, and depends only on optical differential measurements ($\mathbf{R}_j$), calibration matrices ($\mathbf{X}_i, \mathbf{Y}_i$) and robot initial poses ($\mathbf{B}_{i,j}$):

$$\mathbf{M}_{GP} = \mathbf{X}_{GP}^{-1} \cdot \mathbf{Y}_{FP} \cdot \mathbf{B}_{FP} \cdot \mathbf{X}_{FP}^{-1} \cdot \mathbf{Y}_{LA} \cdot \mathbf{B}_{LA} \tag{3.3}$$

for the GP robot ($i = \text{GP}$) and

$$\mathbf{M}_{\text{FP}} = \mathbf{X}_{\text{FP}}^{-1} \cdot \mathbf{Y}_{\text{LA}} \cdot \mathbf{B}_{\text{LA}} \tag{3.4}$$

for the FP robot ($i = \text{FP}$).

THE CLOSED LOOP TARGETING ALGORITHM The residual transformation $\mathbf{R}_j$ between the desired ($\mathbf{T}_d$) and the actual ($\mathbf{T}_j$) probe tip CF is computed as $\mathbf{R}_j = \mathbf{T}_j^{-1} \cdot \mathbf{T}_d$. The error Translation Component (TC) is the Euclidean distance of the translation part of $\mathbf{R}_j$ [84], while the error Rotation Component (RC) is estimated computing the arctangent of the norm of the vector component of the quaternion associated to $\mathbf{R}_j$, divided by the scalar component.

The targeting algorithm is described in fig. 3.3: first the i^{th} robot approaches the target iteratively until the error $\mathbf{R}_j$ is below a threshold (specifically defined for each robot) or if the maximum number of iterations is reached; the $(i+1)^{th}$ robot is then moved and the control loop continues until success or failure are met. It is worth noting that if $\mathbf{M}_i$ was known perfectly without measurement errors, just one movement would put the end-effector in the correct position: iteration are then required to deal with $\mathbf{M}_i$, $\mathbf{B}_{i,j}$ and optical tracker inaccuracies.

The robot arm pose correction $\mathbf{C}_{i,j}$ (dashed block represented in fig. 3.3) is computed to reduce in magnitude the $\mathbf{R}_j$ transformation (in both TC and RC components), avoiding instability. The $\mathbf{R}_j$ translation component (TC_j) and the $\mathbf{R}_j$ rotation component (RC_j) were scaled multiplying TC_j and RC_j by a scaling factor n_j^s computed as follows:

$$n_j^s\left(\dot{x}_j^s\right) = \frac{1}{1 + e^{\dot{x}_j^s/10}} \tag{3.5}$$

where $\dot{x}$ is the first derivative, computed using finite differences, of the s component (TC_j and RC_j). As an example, the correction n is 0.5 when the error does not change from the previous iteration ($\text{TC}_{j+1} = \text{TC}_j$ or $\text{RC}_{j+1} = \text{RC}_j$); when the error is decreasing, thus $\dot{x} < 0$, the correction n approaches 1, while if the error increases the correction n is reduced towards 0. The exponential parameter $1/10$, the sigmoid amplitude, was empirically set.

EXPERIMENTAL PROTOCOL

The planning A brain phantom was designed and developed using a plastic skull and polyvinyl alcohol as brain mimicking material [85], with two gadolinium markers fixed on the base of the skull, used as targets. Magnetic Resonance Image (MRI) images of the phantom were acquired (T1, $512 \times 512 \times 144$, $0.5\,\text{mm} \times 0.5\,\text{mm} \times 1\,\text{mm}$ slice spacing) to get the 3D coordinates of the targets in the medical images CF.

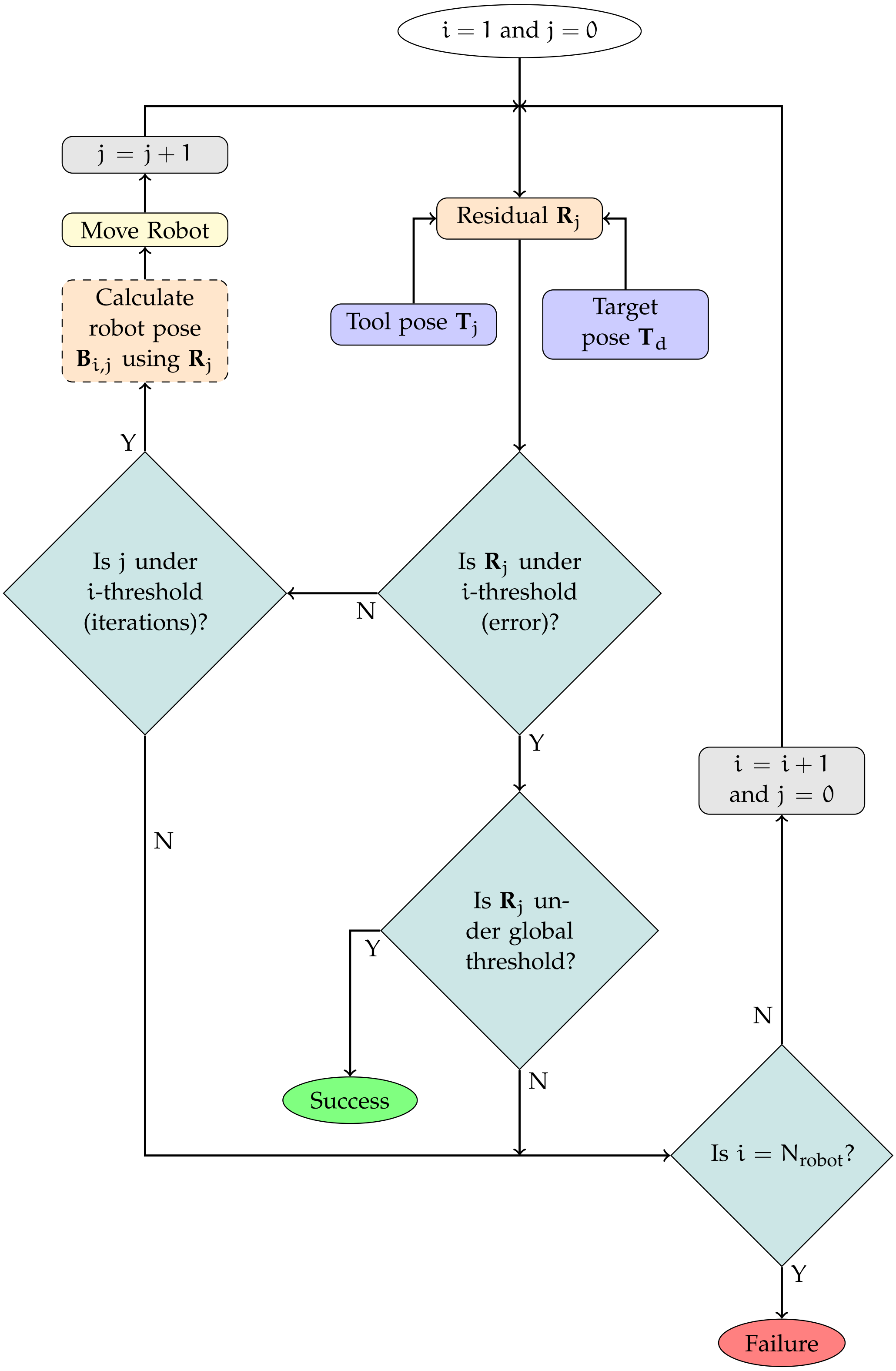

$i = 1$ and $j = 0$
$j = j + 1$
Move Robot
Calculate robot pose $\mathbf{B}_{i,j}$ using $\mathbf{R}_j$
Residual $\mathbf{R}_j$
Tool pose $\mathbf{T}_j$
Target pose $\mathbf{T}_d$
Y
Is j under i-threshold (iterations)?
N
Is $\mathbf{R}_j$ under i-threshold (error)?
Y
$i = i + 1$ and $j = 0$
N
Is $\mathbf{R}_j$ under global threshold?
Y
N
Success
N
Is $i = N_{robot}$?
Y
Failure

Figure 3.3: Block diagram of the target approaching controller. The residual transformation is $\mathbf{R}_j = \mathbf{T}_j^{-1} \cdot \mathbf{T}_d$. First the i^{th} robot approaches the target with an iterative approach until the error $\mathbf{R}_j$ (both the translation (TC) and the rotation (RC) components) is below a threshold (specifically defined for each robot) or if the maximum number of iterations is reached. Then the $(i+1)^{th}$ robot is moved and the control loop continues until success or failure are met.

For each target point, 12 possible entry points were selected on the whole skull surface, therefore 12 probe trajectories were planned for each target point.

The planned trajectories were registered into the operating volume using a point-based registration algorithm [86]. The surgical probe was positioned by the GP and the FP 2 cm above the entry point along the planned trajectory ($\mathbf{T}_d$), then it was inserted in the phantom by the LA towards, following the designed surgical plan.

In order to evaluate system performances, the following data were measured:

- The final probe tip residual transformation matrices ($\mathbf{R}_j$) components (TC_f and RC_f) for both the GP and the FP;
- The actual Entry Point (EP);
- The actual Target Point (TP);
- The actual Probe Trajectory (PT), computed as the line crossing EP and TP.

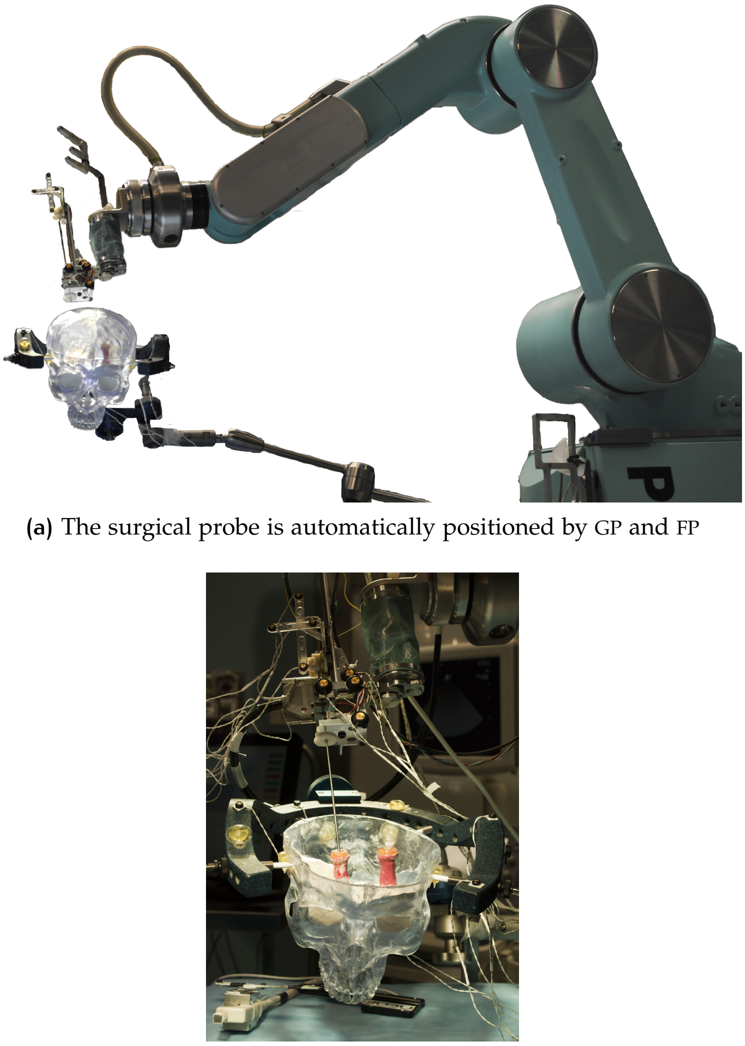
The targeting tests The modular-robotic system was moved in order to align the actual surgical probe trajectory ($\mathbf{T}_j$) with the planned trajectory ($\mathbf{T}_d$). The GP threshold was set to 2 mm for translation (TC) and 0.1 rad for rotation (RC), with a maximum of 10 iterations allowed, after which the FP was moved towards the desired target pose ($\mathbf{T}_d$). The FP threshold was set to 0.08 mm (corresponding to the expected accuracy of DRFs [87]) for translation and 0.01 rad for rotation with maximum 10 iterations allowed.

Targeting tests were performed with the ROBOCAST system in three different conditions:

"CALIBRATED" TARGETING the optical calibration of GP, FP and LA (see section 3.1.2.1 on page 29) was performed immediately before the targeting experiments;

"NON-CALIBRATED" TARGETING the optical calibration of GP, FP and LA was performed the day before the experiments, in order to check the calibration stability;

"PERTURBED" TARGETING in order to test the algorithm robustness, the calibration (see section 3.1.2.1 on page 29) was altered on purpose by multiplying the calibration matrix ($\mathbf{X}_{GP}$) by transformation matrices ($\mathbf{P}_{GP}$) randomly chosen in a population of (0 to 50) mm translation and (0 to 0.09) rad rotation (uniform distribution) and multiplying the calibration matrix ($\mathbf{X}_{FP}$) by transformation matrices ($\mathbf{P}_{FP}$) randomly chosen in a population of (0 to 5) mm range of translation and (0 to 0.03) rad rad of rotation (uniform distribution).



(a) The surgical probe is automatically positioned by GP and FP

(b) Final targeting verification: the probe tip reaches the gadolinium marker center

Figure 3.4: The OR set-up during the experiments.

Table 3.3: Experimental protocol for the targeting accuracy tests.

Number of targets	2	
Number of entry points (for each target)	12	
Number of repetitions in each entry point	3	
Maximum number of iteration per robots	10	
	Translation (mm)	Angle (rad)
GP threshold	2	0.1
FP threshold	0.08	0.01
GP calibration perturbation ($\mathbf{P}_{\mathrm{GP}}$)	50 (max)	0.09 (max)
FP calibration perturbation ($\mathbf{P}_{\mathrm{FP}}$)	5 (max)	0.03 (max)

For each targeting experiment (fig. 3.4), all DRFs poses were acquired at 30 Hz, while the test protocol is reported in table 3.3.

EVALUATION METRICS AND DATA ANALYSIS Targeting performance evaluation was performed computing the residual translation and rotation transformation (TC_j and RC_j components of $\mathbf{R}_j$) for each targeting trial.

Experimental results were evaluated using non-parametric Kruskal-Wallis test with $p < 0.05$ significance (STATISTICA 10, StatSoft).

In order to check whether the iterative targeting always assures that the residual translation transformation (TC_j component of $\mathbf{R}_j$) is decreasing in magnitude, the following data were computed:

- Iterative corrections of the GP ($\mathrm{IC}_{\mathrm{GP}}$), computed for each targeting experiment (k) as:

$$\mathrm{IC}_{\mathrm{GP}k} = \mathrm{TC}_{\mathrm{GP},j-1,k} - \mathrm{TC}_{\mathrm{GP},j,k} \tag{3.6}$$

- Iterative correction of the FP ($\mathrm{IC}_{\mathrm{FP}}$), computed for each targeting experiment (k) as:

$$\mathrm{IC}_{\mathrm{FP}k} = \mathrm{TC}_{\mathrm{FP},j-1,k} - \mathrm{TC}_{\mathrm{FP},j,k} \tag{3.7}$$

The Pearson correlation coefficients among the TC component of the $\mathbf{P}_{\mathrm{GP}}$ and the TC component of the residual $\mathbf{R}_j$ and among the TC component of the $\mathbf{P}_{\mathrm{FP}}$ and the TC component of the residual $\mathbf{R}_j$ were computed.

In order to compare the overall targeting accuracy, the following evaluation metrics were computed:

- Residual Error at the Entry point ($\mathrm{RE}^{\mathrm{EP}}$): Euclidean distance between the actual and the desired EP;

Table 3.4: Calibration residuals and calibration errors (median, 25th and 75th percentile) for the three robots (GP, FP and LA).

(a) Calibration residual

	TC (mm)	RC (rad)
GP	0.2245 (0.1288 to 0.2623)	0.0021 (0.0008 to 0.0306)
FP	0.1548 (0.1260 to 0.2009)	0.0035 (0.0020 to 0.0049)
LA	0.2561 (0.1452 to 0.4279)	0.0014 (0.0009 to 0.0054)

(b) Calibration errors

	TC (mm)	RC (rad)
GP	0.7724 (0.6430 to 0.8303)	0.0008 (0.0006 to 0.0011)
FP	0.2405 (0.2025 to 0.3450)	0.0129 (0.0115 to 0.0140)
LA	0.2863 (0.2639 to 0.3411)	0.0073 (0.0069 to 0.0077)

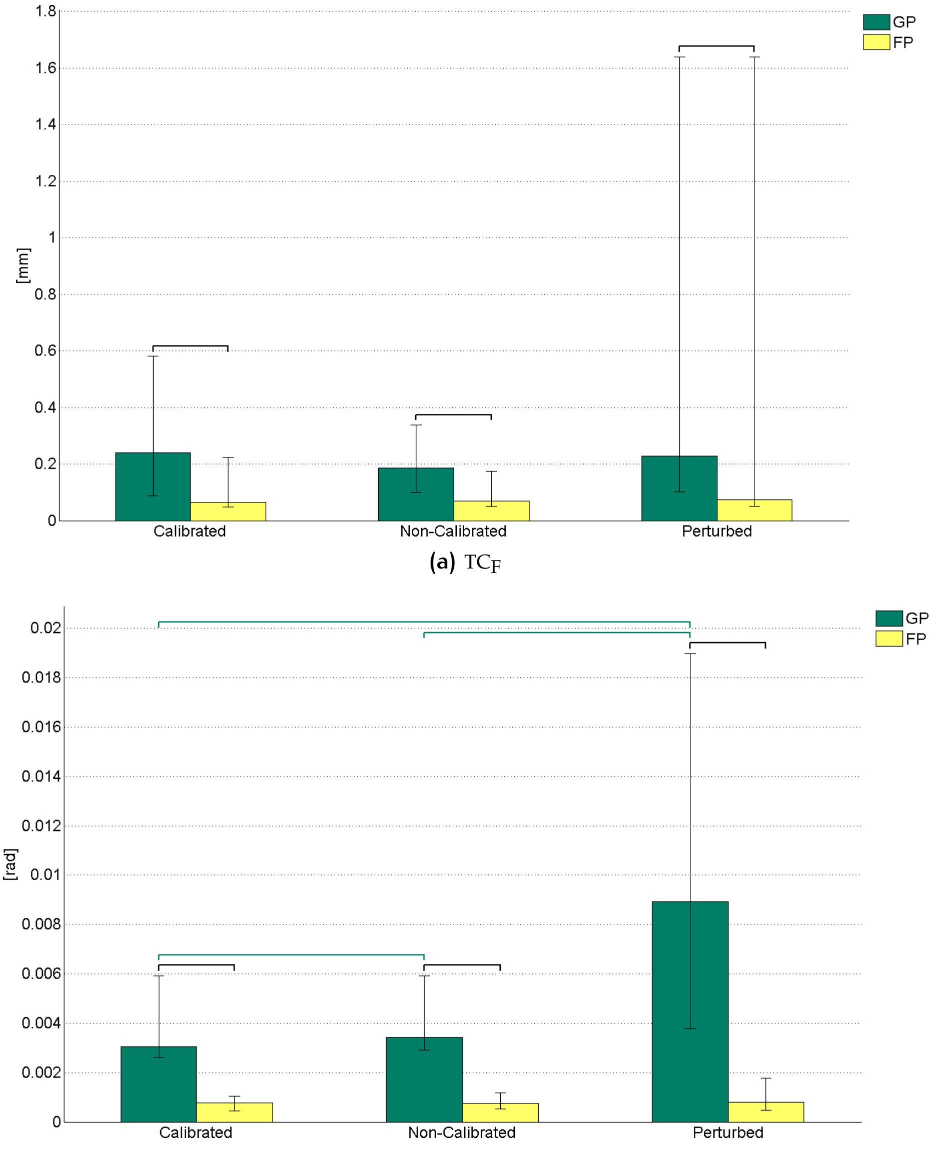
- Residual Error at the Target point ($\mathrm{RE}^{\mathrm{TP}}$): Euclidean distance between the actual and the desired TP;
- Angle between the actual and desired probes trajectories (A^{PT}): angle between the two lines in space.

Different experimental setups were compared using non-parametric Kruskal-Wallis test with $p < 0.05$ significance (STATISTICA 10, StatSoft).

3.1.2.2 *Results*

CALIBRATION Calibration residuals and calibration errors for GP, FP and LA are reported in table 3.4 in both the (TC) and the (RC) components. Calibration residuals are about 0.2 mm for the three robots, which is the accuracy that would be reached without an iterative targeting, if the calibration model would perfectly generalize.

GROSS POSITIONER AND FINE POSITIONER TARGETING Figure 3.5 shows the $\mathrm{TC_F}$ and $\mathrm{RC_F}$ components of the final (either below threshold or after the maximum number of iterations was reached) $\mathbf{R}_j$ transformation matrix for the GP and the FP in the three evaluation scenarios (calibrated, non-calibrated and perturbed). There is no significant difference within the same robot (GP or FP) in the three evaluation scenarios as far as the $\mathrm{TC_F}$ parameter is considered. The angular residual error of the GP ($\mathrm{RC_F}$) worsen when the system is not calibrated, while the $\mathrm{RC_F}$ of the FP is not statistically affected. The FP provides a significantly better final pose of the surgical probe tip with respect to the GP robot.



(a) TC_F

(b) RC_F

Figure 3.5: Components of the final (either below threshold or after the maximum number of iterations was reached) $\mathbf{R}_j$ transformation matrix for the GP and the FP in the three evaluation scenarios (calibrated, non-calibrated and perturbed). Bars represent the median value and the error bars show the interquartile range of the data presented. Horizontal bars indicate significant difference ($p < 0.05$).

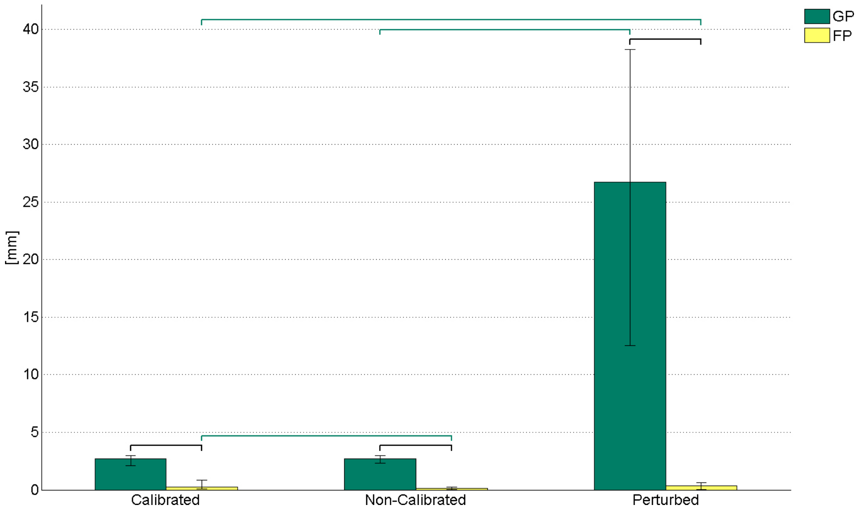


Figure 3.6: IC_{GP} (eq. (3.6)) and IC_{FP} (eq. (3.7)) in the three evaluation scenarios (calibrated, non-calibrated and perturbed). Bars represent the median value and the error bars show the interquartile range of the data presented. Horizontal bars indicate significant difference ($p < 0.05$).

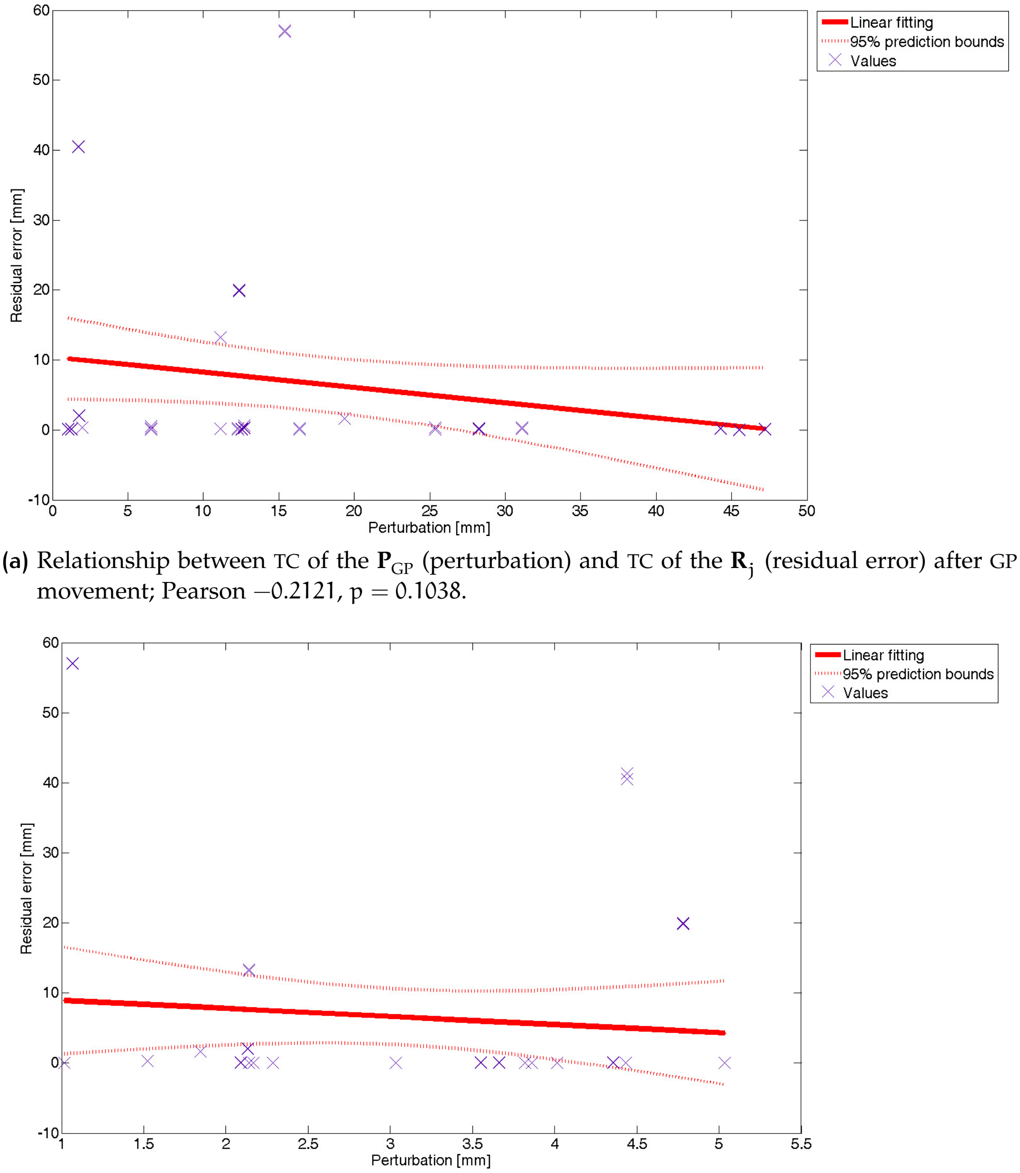
Table 3.5: Negative corrections of the iterative algorithm for GP and FP targeting.

Condition	Value	GP	FP
Calibrated	# (%)	1 (0.49 %)	6 (3 %)
	Largest correction [mm]	−0.1479	−0.2066
Non-calibrated	# (%)	0 (0 %)	10 (4.9 %)
	Largest correction [mm]	0	−0.4011
Perturbed	# (%)	0 (0 %)	14 (0 %)
	Largest correction [mm]	0	−0.8775

fig. 3.6 shows IC_{GP} (see eq. (3.6)) and IC_{FP} (see eq. (3.7)) in the three evaluation scenarios (calibrated, non-calibrated and perturbed). The correction value population appears normally distributed, with few negative outliers (table 3.5).

fig. 3.7 shows the relationship between the TC component of the $\mathbf{P}_{GP}$ and the TC component of the $\mathbf{R}_j$, after the GP targeting movement, and between the TC component of the $\mathbf{P}_{FP}$ and the TC component of the $\mathbf{R}_j$, after the FP targeting movement. As shown, there is no correlation ($p > 0.05$) among the perturbation and the automatic targeting performance, neither for the GP (fig. 3.7a), nor for the FP (fig. 3.7b).

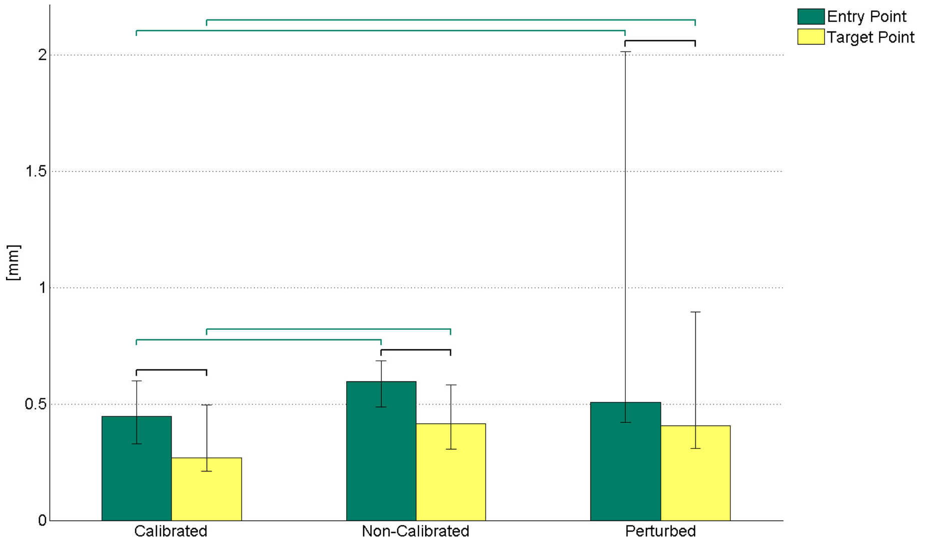
Linear Actuator targeting fig. 3.8 shows the targeting distances at the entry point (RE^{EP}), at the target point (RE^{TP}) and the angular error among the planned and the actual probe trajectory (A^{PT}) in the



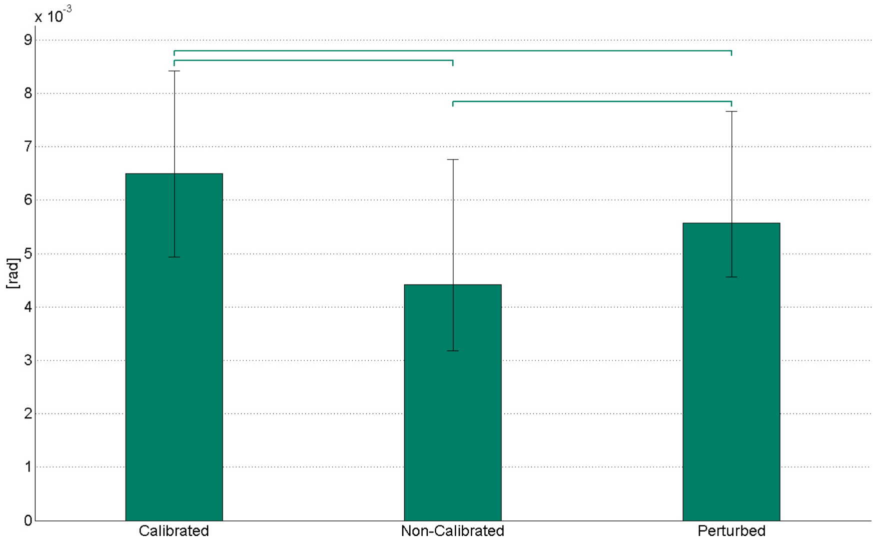
(a) Relationship between TC of the $\mathbf{P}_{GP}$ (perturbation) and TC of the $\mathbf{R}_j$ (residual error) after GP movement; Pearson -0.2121, $p = 0.1038$.

(b) Relationship between TC of the $\mathbf{P}_{GP}$ (perturbation) and TC of the $\mathbf{R}_j$ (residual error) after FP movement; Pearson -0.0948, $p = 0.4712$.

Figure 3.7: Relationship between TC component of the $\mathbf{P}_{i^{th}\ robot}$ and TC of the $\mathbf{R}_j$ after the i^{th} robot movement; Pearson coefficient value and p value are also reported.



(a) Targeting distances at the entry point RE^{EP}, at the target point RE^{TP} in the three evaluation scenarios (calibrated, non-calibrated and perturbed)



(b) Angular error A^{EP} among the planned and the actual probe trajectory in the three evaluation scenarios (calibrated, non-calibrated and perturbed)

Figure 3.8: Errors between planned trajectory and real one. Bars represent the median value and the error bars show the interquartile range of the data presented. Horizontal bars indicate significant difference ($p < 0.05$).

three evaluation scenarios (calibrated, non-calibrated and perturbed). As shown, there is a significant difference between the targeting errors TC at the EP and at the TP, and among the three calibration scenarios. Also, there is a significant difference in the angular trajectory RC errors among the three calibration scenarios.

The location of the entry point did not influence the targeting accuracy, since no statistically significant difference was found in the errors.

3.1.2.3 *Discussions*

In this section we show the experimental evaluation of a novel targeting algorithm designed for a multi-robot system (three robots connected in series) designed for keyhole neurosurgery, based on external sensors. Robotic systems suffer from two main drawbacks: the movement accuracy of a serial robot does not respect surgical requirements and there is a lack of feedback information if the actual robot pose differs from the planned desired one.

In the proposed approach, an optical localization system tracks both the pose, for the registration, of the patient head and the robot end-effector [78], estimating the error between the planned and the current surgical tool pose, allowing further iterative corrections of the pose of the robot end effector. Also, kinematic redundancy (13 DoFs) allows optimizing the robot approach: the miniaturized parallel robot (FP) mounted on the serial robot (GP) provides a more accurate surgical probe pose, due to its greater accuracy and resolution. The iterative targeting algorithm, which moves the second robot only after the GP residual error transformation has reached the pre-defined threshold, assures that the FP is able to perform further corrections within its limited workspace. The LA is then used to make the probe advance along the trajectory reached by the GP and FP.

The system was tested following the planned operatory workflow: first the GP and FP bring the probe's actual pose to the planned one, 2 cm above the entry point (GP, FP targeting). The residual error, is reduced iteratively, first approaching the reduction with the gross positioner (GP), then refining the targeting with the fine positioner (FP).

The targeting approach outperforms other neurosurgical robots targeting accuracies. As examples, the PathFinder was tested on phantoms and the reported targeting accuracy is 0.5 mm [13]. The NeuroMate accuracy was reported to be 1.95 mm in the frameless approach [20] and 2.9 mm using implanted skull fiducials [54].

In literature targeting for keyhole surgery was addressed providing optical tracking based corrections to robots. With respect to existing methodologies, the one proposed is based on iterative corrections and allows reaching accuracy values twice as better.

The optical feedback approach proposed by [76] allowed reaching (2.6 ± 0.8) mm as maximum error at the first iteration and (0.60 ± 0.36) mm as final targeting error, while [77] reached 0.5 mm of accuracy without previous calibration. Using the iterative approach proposed, the GP allowed reaching 0.25 mm as median value in all the three calibration scenarios, while the FP targeting performances were around 0.1 mm as median value. It must be noted that this value is limited only by the registration error and by the accuracy of the localization system.

The iterative approach proposed proved to be independent of the calibration residuals: even if the calibration transformation is inaccurate, the iterative algorithm allows reaching comparable residual errors with the accurate calibrations for both the serial and the parallel robots despite the high initial errors. The feedback control scheme is based on kinematics and it is intrinsically stable because of the low speed allowed. Stability problems were tackled by modulating the amplitude of the correction as a function of current error. Also, the target was considered fixed since the patient head is supposed to be framed in a head-ring.

The correction values of the GP are negative in magnitude (about -0.2 mm which means that the robot is getting further from the target) in one case only when the system is calibrated. The correction values of the FP are negative in magnitude (about -0.3 mm) in approximately 5 % of the cases, independently of the calibration accuracy. Such values are comparable to the OTS accuracy. Stated optical system accuracy is 0.15 mm, which is the worse 3D localization error. When using a four markers DRF, the target error in the center of mass is 0.08 mm [87]. The external sensor accuracy (tracker) affects indeed the results, since the corrections are computed starting from tracker coordinates. Cheap system with low working volume, such as the Vicra system (NDI, Ontario, Canada) can be applied to the target surgical scenario, which is surgery around the head, which is a small anatomical district, allowing to reach 0.13 mm of accuracy in case four markers are used [87].

Considering the LA targeting, the error at the entry point (0.6 mm as median value) is statistically different from the targeting error at the target point (0.4 mm as median value), and both values satisfy requirements for clinical applications (1 mm is the typical stereotactic frame accuracy). Fiducial registration error [87] (which was estimated to be about 0.1 mm as median value) influences the results, but with an amplitude of the same order as the optical localization error.

With regard to targeting performances, it is worth recalling that the ROBOCAST system was specially designed for keyhole approaches, where a straight surgical probe is inserted in the brain tissue and the robot acts as an assistant, providing an accurate holder for guiding the probe insertion. Therefore the aim of this study was to improve and test the accuracy of targeting in terms of position and rotation errors. The novelty of the approach relies on the iterative tracking of the current pose, performed combining the robotic architecture with an external optical tracking device. The control of the position and orientation is implemented applying a correction factor which is scaled depending on the fact that the robot approach is getting closer or further to the target pose.

In conclusion, the objective of this section was reached as the targeting accuracy obtained is better than previous works and than the one required by neurosurgical procedures. The presence of the tracking system is not a further constraint in the OR since it is usually used for neuronavigation, thus, as already introduced, the targeting algorithm proposed can be easily integrated into already commercially available surgical robots suites for keyhole surgery.

3.1.3 Convergence evaluation

The aim of the work presented in this section is to provide simulated and experimental evidence of the iterative targeting convergence robustness, with respect to perturbations in the calibration matrices, robot pose and targeted point for a robot-aided keyhole neurosurgical procedure, such as brain biopsy and electrodes placement.

3.1.3.1 *Theory*

THE TARGETING PROBLEM An iterative algorithm is used for compensating targeting errors, and robustness is assessed as the ability of such an algorithm to converge by starting from perturbed initial condition and by using perturbed data. In fig. 3.9 the experimental scenario is presented.

In a robot manipulator, the base CF ($\mathrm{CF}_{\mathrm{Base}}$) identifies the robot internal CF and the flange CF ($\mathrm{CF}_{\mathrm{Flange}}$) identifies the free extremity CF. External localization sensors (e. g. cameras, fluoroscopic images, localization systems [87]) can be used to track the robot links. The transformation matrices, which relate the position and orientation of the markers attached to the last link DRF with respect to the robot $\mathrm{CF}_{\mathrm{Flange}}$, can be estimated with a "hand-eye" calibration procedure [40], as in section 3.1.2.1, that solves the matrix system eq. (3.8):

$$\mathbf{A} \cdot \mathbf{X} = \mathbf{Y} \cdot \mathbf{B} \tag{3.8}$$

in which the $\mathbf{X}$ and $\mathbf{Y}$ matrices are the calibration matrices, $\mathbf{B}$ is the robot pose (forward kinematics), and $\mathbf{A} = \mathbf{G}^{-1} \cdot \mathbf{T} \cdot \mathbf{R}_j^{-1}$ (fig. 3.10a) is the transformation from the flange DRF and the robot base DRF, where $\mathbf{G}$ and $\mathbf{T}$ are the robot base and robot tool poses in the localizer CF.

The surgical tools connected to the robot flange (and actual robot End-Effector (EE)) are also calibrated with respect to the markers on

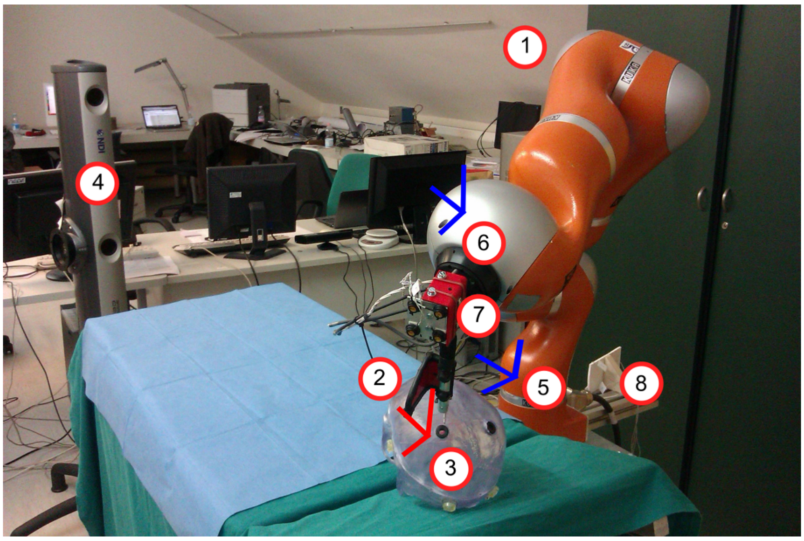
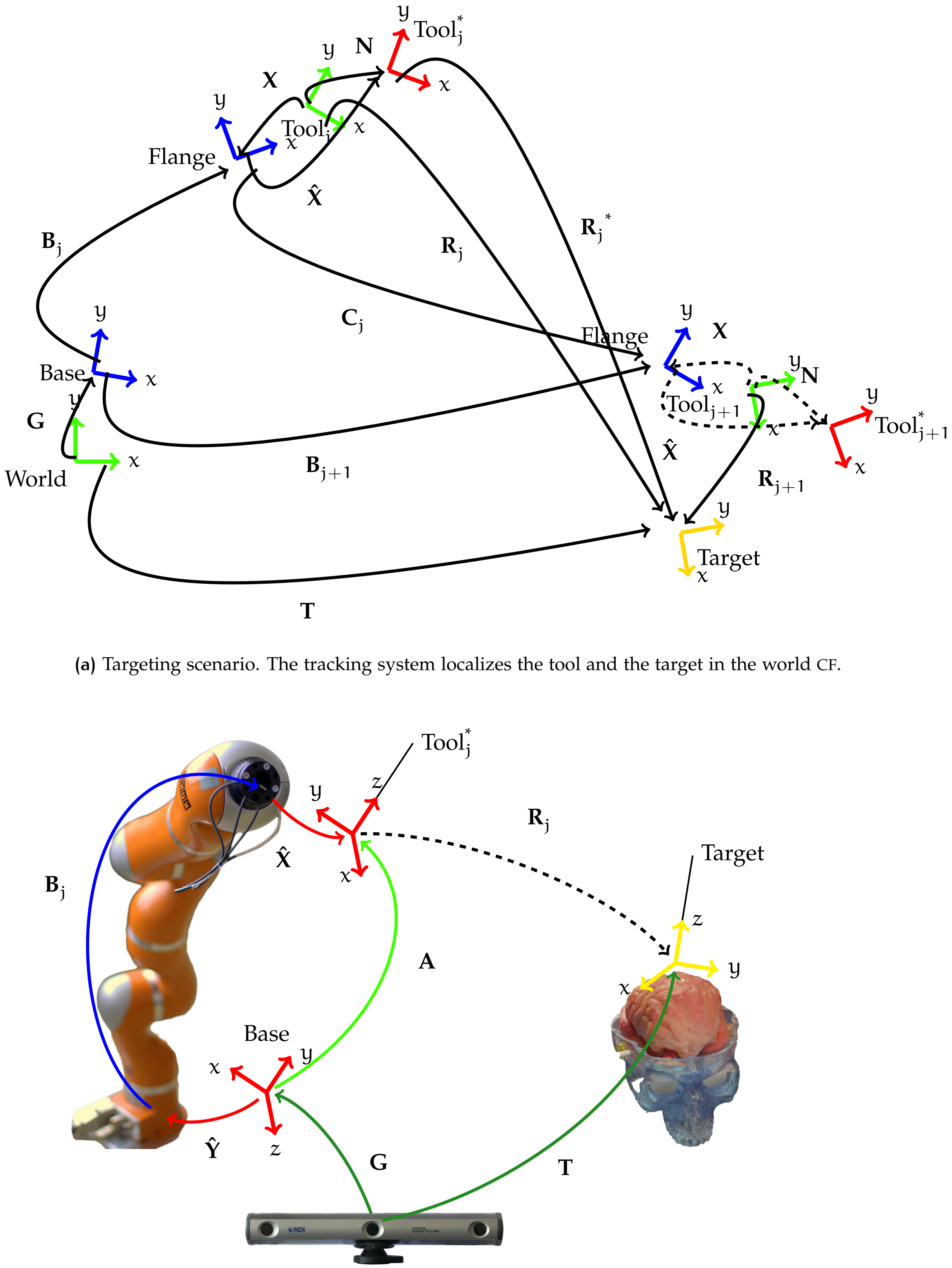


Figure 3.9: The keyhole neurosurgery scenario. The robotic arm (1) automatically aligns the surgical probe (2) to the planned trajectory (3). The external optical sensor (4) information is used to measure the accuracy and evaluate the correction for the robot by measuring the robot base CF (5) and the robot end-effector CF (6) position.

the robot [49]. The desired pose of the surgical tool is generally preoperatively defined in the medical images CF and is known in the intra-operative environment after a registration procedure. Given the tool's actual and desired pose, the robot pose can be consequently estimated. Nonetheless, due to hand-eye calibration inaccuracies [40] and due to serial robot intrinsic inaccuracies [75], further adjustments of the robot pose can be required. In [49], the TCP pose was iteratively corrected until the residual transformation error between the current and the planned surgical tool poses was below a user-defined threshold or until the maximum number of iterations was reached. In the following, we assume that the CF of the surgical tool and the CF of the markers attached to the robot last link are coincident.

SPATIAL TRANSFORMATIONS CFs and geometric transformations are shown in fig. 3.10b, where j indicates the targeting iteration index:

- CF_{World}: the CF of the external sensor or of the laboratory;
- CF_{Base}: the internal CF of the robot base;
- CF_{Flange}: the CF of the robot flange;
- CF_{Tool_j}, $CF_{Tool_{j+1}}$: the actual CF of the tool fixed to the robot flange at iterations j and $j+1$ respectively;
- $CF_{Tool^*_j}$, $CF_{Tool^*_{j+1}}$: the estimated CF of the tool at iterations j and $j+1$ respectively;



(a) Targeting scenario. The tracking system localizes the tool and the target in the world CF.

(b) CFs and transformations: 'World' is the CF of the external sensor, 'Target' is the desired pose to be reached and 'Tool*' is the estimated CF of the tool; the arrows represent the geometrical transformations.

Figure 3.10: Kinematic chain representation for the targeting convergence analysis.

- CF$_{\text{Target}}$: the CF of the tool target pose.

The geometrical transformations among the aforementioned CFs, that are known or that can be measured using external position sensors, are:

- $\mathbf{G}$: pose of the CF$_{\text{Base}}$ in the CF$_{\text{World}}$;
- $\mathbf{T}$: pose of the CF$_{\text{Target}}$ in the CF$_{\text{World}}$;
- $\mathbf{B}_j$, $\mathbf{B}_{j+1}$: pose of the CF$_{\text{Flange}}$ in the CF$_{\text{Base}}$, at iterations j and $j+1$;
- $\mathbf{R}_j$: transformation between the CF$_{\text{Tool}}$ and the CF$_{\text{Target}}$;
- $\mathbf{C}_j$: transformation from $\mathbf{B}_j$ to $\mathbf{B}_{j+1}$;

while the unknown transformations are:

- $\mathbf{R}_j^*$: the transformation between the CF$_{\text{Tool}_j^*}$ and the CF$_{\text{Target}}$;
- $\mathbf{X}$: actual transformation from the CF$_{\text{Tool}}$ to the CF$_{\text{Flange}}$ [40];
- $\hat{\mathbf{X}}$: estimated transformation from the CF$_{\text{Tool}_j^*}$ to the CF$_{\text{Flange}}$ (calibration matrix) [40].

The relationship between $\hat{\mathbf{X}}$ and $\mathbf{X}$ is:

$$\hat{\mathbf{X}} = \mathbf{X}^{-1} \cdot \mathbf{N} \tag{3.9}$$

where $\mathbf{N}$ is the error transformation due to the calibration inaccuracy. In order to align the estimated tool pose (CF$_{\text{Tool}_j^*}$) with the tool target pose (CF$_{\text{Target}}$), the robot pose has to change from $\mathbf{B}_j$ to $\mathbf{B}_{j+1}$ according to the estimated $\mathbf{C}_j$ transformation

$$\mathbf{B}_{j+1} = \mathbf{B}_j \cdot \mathbf{C}_j \tag{3.10}$$

where $\mathbf{C}_j$ is:

$$\mathbf{C}_j = \mathbf{X} \cdot \mathbf{R}_j \cdot \mathbf{X}^{-1} \tag{3.11}$$

that is approximated with $\hat{\mathbf{C}}_j$, computed as:

$$\hat{\mathbf{C}}_j = \hat{\mathbf{X}} \cdot \mathbf{R}_j \cdot \hat{\mathbf{X}}^{-1} \tag{3.12}$$

in which $\mathbf{X}$ is approximated with $\hat{\mathbf{X}}$. From this point on, $\hat{\mathbf{B}}_{j+1} = \mathbf{B}_j \cdot \hat{\mathbf{C}}_j$ (approximating $\mathbf{B}_{j+1}$) is consequently computed as:

$$\begin{aligned} \hat{\mathbf{B}}_{j+1} &= \mathbf{B}_j \cdot \hat{\mathbf{X}} \cdot \mathbf{R}_j \cdot \hat{\mathbf{X}}^{-1} \\ &= \mathbf{B}_j \cdot \mathbf{X}^{-1} \cdot \mathbf{N} \cdot \mathbf{R}_j \cdot \mathbf{N}^{-1} \cdot \mathbf{X} \end{aligned} \tag{3.13}$$

Targeting iterations are therefore needed to make the tool CF approximate with the tool target pose. Iterations are stopped when $\mathbf{R}_j$ translation and rotation components are both below a pre-determined threshold value, chosen according to the application requirements (e. g. as already specified, in case of keyhole neurosurgical application, the targeting accuracy is 1 mm).

3.1.3.2 *Methods*

The targeting algorithm brings the surgical tool into the target pose $\mathbf{T}$, with the robot in the initial pose $\mathbf{B}_0$ (fig. 3.10b).

When the transformation $\mathbf{N}$ and $\mathbf{R}_j$ are pure translations, from eq. (3.13), convergence is reached within one iteration

$$\begin{aligned} \hat{\mathbf{B}}_{j+1} &= \mathbf{B}_j \cdot \mathbf{X}^{-1} \cdot \mathbf{R}_j \cdot \mathbf{X} \\ &= \mathbf{B}_{j+1} \end{aligned} \tag{3.14}$$

while when, instead, the transformation $\mathbf{N}$ is a not-nil rotation, the targeting iterates until the residual error transformation $\mathbf{R}_j$ is below a predefined threshold.

Targeting is iteratively performed correcting the robot EE pose in the Cartesian space by measuring the position reached by the tool after the movement is performed and by calculating the correction from the error with respect to the desired pose. This is performed after the programmed movement of the robot. Time for robot movement, sensing and error correction (quite long) allows neglecting robot dynamics, as discussed in section 3.1.2 on page 28.

The question faced by this section is to find a measurable perturbation in terms of calibration errors that allow defining the robustness of the method with respect to such perturbation.

Numerical simulations and laboratory experiments were performed to assess the targeting convergence with respect to the calibration error, as function of the robot initial and target pose. The calibration matrix $\mathbf{X}$ was multiplied by an error matrix $\mathbf{N}^{\vartheta}$ that represents all the errors in the robot calibration chain which can be due to encoders, external measurement system or calibration estimation inaccuracies.

The rotation component of the transformation matrices ($\mathbf{N}^{\vartheta}$, $\mathbf{B}_0^{\vartheta}$ and $\mathbf{T}^{\vartheta}$) were rotations about the trisectrix of the first octant. The translation component of the transformation matrices, ($\mathbf{B}_0^{d}$ and $\mathbf{T}^{d}$), were varied along a vector obtained by rotating the x axis by 30° around the y axis and rotating the resulting vector by 30° around the x axis. Ranges and step of variations are reported in table 3.6.

The following parameters, which represent the metric of the robustness, were estimated:

1. $\mathbf{N}^{\vartheta}_{MAX}$, which is the maximum angular component of $\mathbf{N}$ which allows convergence as a function of $\mathbf{B}_0^{\vartheta}$, $\mathbf{B}_0^{d}$, $\mathbf{T}^{\vartheta}$ and $\mathbf{T}^{d}$;

Table 3.6: Testing protocol for the convergence analysis: ranges of variation of the variables.

Variable	Range	Step (simulation)	Step (experiments)
$\mathbf{N}^{\vartheta}$	(0 to 60)°	0.25°	10°
$\mathbf{B}_0^{\vartheta}$	(0 to 60)°	1°	6°
$\mathbf{B}_0^{d}$	(0 to 150) mm	1 mm	15 mm
$\mathbf{T}^{\vartheta}$	(0 to 60)°	1°	6°
$\mathbf{T}^{d}$	(0 to 150) mm	1 mm	15 mm

2. # iterations: the number of iterations for targeting, varying $\mathbf{N}^{\vartheta}$, $\mathbf{B}_0^{\vartheta}$, $\mathbf{B}_0^{d}$, $\mathbf{T}^{\vartheta}$ and $\mathbf{T}^{d}$. At $\mathbf{N}^{\vartheta}_{MAX}$, the correlation between the number of iterations and $\mathbf{B}_0$ and $\mathbf{T}$ variations were computed using Pearson correlation coefficient ($\alpha < 0.05$). The relationship between the number of iterations and the value of $\mathbf{N}^{\vartheta}$ component was fitted using an exponential function.

SIMULATIONS In order to analyze the convergence of the targeting over a wide range of the perturbing parameters, numerical simulations were performed (MATLAB® R2012a, The MathWorks, Natick, Massachusetts) since experiments are limited by time. In those tests, $\mathbf{X}$ was assumed to be a 4×4 identity matrix without any loss of generality. This allows rewriting eq. (3.13) as follows:

$$\hat{\mathbf{B}}_{j+1} = \mathbf{B}_j \cdot \mathbf{N} \cdot \mathbf{R}_j \cdot \mathbf{N}^{-1} \tag{3.15}$$

The selected threshold on $\mathbf{R}_j$, as the minimum positioning resolution, is 0.8 mm for the translation component and 0.05° for the rotation component, which allow satisfying neurosurgical applications requirements [48]. The maximum number of allowed iterations was 100 and no measurement noise was added during the simulation.

EXPERIMENTS The convergence performances of the algorithm were also experimentally tested using the 7 DoFs LWR 4+ (KUKA Laboratories, Augsburg, DE) [30], and the OptoTrack Certus (NDI, Ontario, Canada), with 0.15 mm stated accuracy in a pyramidal working volume of about 25 m^3, as external sensor. Two sensors (DRFs), composed of four active markers each, were attached to the robot flange and to the robot base. The robot and the tracking system CFs were calibrated, as reported in [59], and the $\mathbf{X}$ transformation was therefore estimated ($\hat{\mathbf{X}}$).

The robot target positions were varied as reported in table 3.6 and the iterative algorithm convergence was assessed. During each algorithm iteration, the targeting accuracy was computed.

Targeting accuracy of the LWR was assessed estimating the residual translation and rotation components of the transformation connecting the desired and the actual tool target pose. The selected threshold on $\mathbf{R}_j$, as the minimum positioning resolution, is 0.8 mm for the translation component and 0.05° for the rotation component. The maximum number of allowed iterations was 100.

3.1.3.3 *Results*

The experimental calibration procedure (between the OTS and the robot) median residual errors, calculated on the 33 poses used for the calibration matrices estimation, resulted in 1.81 mm (with 1.25 mm and 2.79 mm as first and third quartiles) and 0.14° (with 0.11° and 0.19 ° as first and third quartiles) for translation and rotation components, respectively. The median technical accuracy of the robot in 15 trials while reaching a target fixed in space was 1.18 mm (with 0.91 mm and 1.44 mm as first and third quartiles) and 0.95° (with 0.93° and 0.98° as first and third quartiles) for translation and rotation components, respectively.

fig. 3.11 shows the $\mathbf{N}^{\vartheta}$ values which allow convergence in simulation and experiments, varying $\mathbf{B}_0^{\vartheta}$, $\mathbf{B}_0^{d}$, $\mathbf{T}^{\vartheta}$ and $\mathbf{T}^{d}$ as indicated in table 3.6. The maximum values for the angular component of $\mathbf{N}$ which allows convergence, $\mathbf{N}_{MAX}^{\vartheta}$, resulted to be independent from $\mathbf{B}_0$ and $\mathbf{T}$ variations. In simulations $\mathbf{N}_{MAX}^{\vartheta}$ resulted to be around 55° in case of $\mathbf{B}_0^{\vartheta}$, 54.25° in case of $\mathbf{B}_0^{d}$ variations and around 57.5° in case of $\mathbf{T}$ variations (both angular and translation components). In experimental conditions $\mathbf{N}_{MAX}^{\vartheta}$ resulted 50° in case of $\mathbf{B}_0$ and $\mathbf{T}$ variations. The convergence was reached until $\mathbf{T}^{\vartheta} = 42°$, since for $\mathbf{T}^{\vartheta} \geqslant 42°$ the markers were not visible from the external sensor.

In fig. 3.12 the number of targeting iterations is shown as a function of $\mathbf{B}_0$ and $\mathbf{T}$ variations, with $\mathbf{N}^{\vartheta} = \mathbf{N}_{MAX}^{\vartheta}$, which is higher in simulation results than in the experiments, as shown in fig. 3.11. In simulations, the number of iterations is independent from $\mathbf{B}_0$ and $\mathbf{T}$ variations and is around 65 in case of $\mathbf{B}_0$ variations and 95 in case of $\mathbf{T}$ rotation component. In the set-up described in section 3.1.3.2, when $\mathbf{T}^{d}$ is greater than 40 mm the algorithm does not converge within 100 iterations. In experiments, the number of required iterations is approximately 40 and is independent from $\mathbf{B}_0$ and $\mathbf{T}$ variations, which takes about 80 s to be carried out.

fig. 3.13 shows the maximum number of targeting iterations required as a function of $\mathbf{N}^{\vartheta}$, evaluated for all the possible $\mathbf{B}_0$ and $\mathbf{T}$ variations. $\mathbf{N}^{\vartheta}$ was varied until the experimentally computed $\mathbf{N}_{MAX}^{\vartheta}$ ($\mathbf{N}^{\vartheta} = 50°$). As shown, the number of iterations increases exponentially in both testing conditions. fig. 3.13 further confirms the results

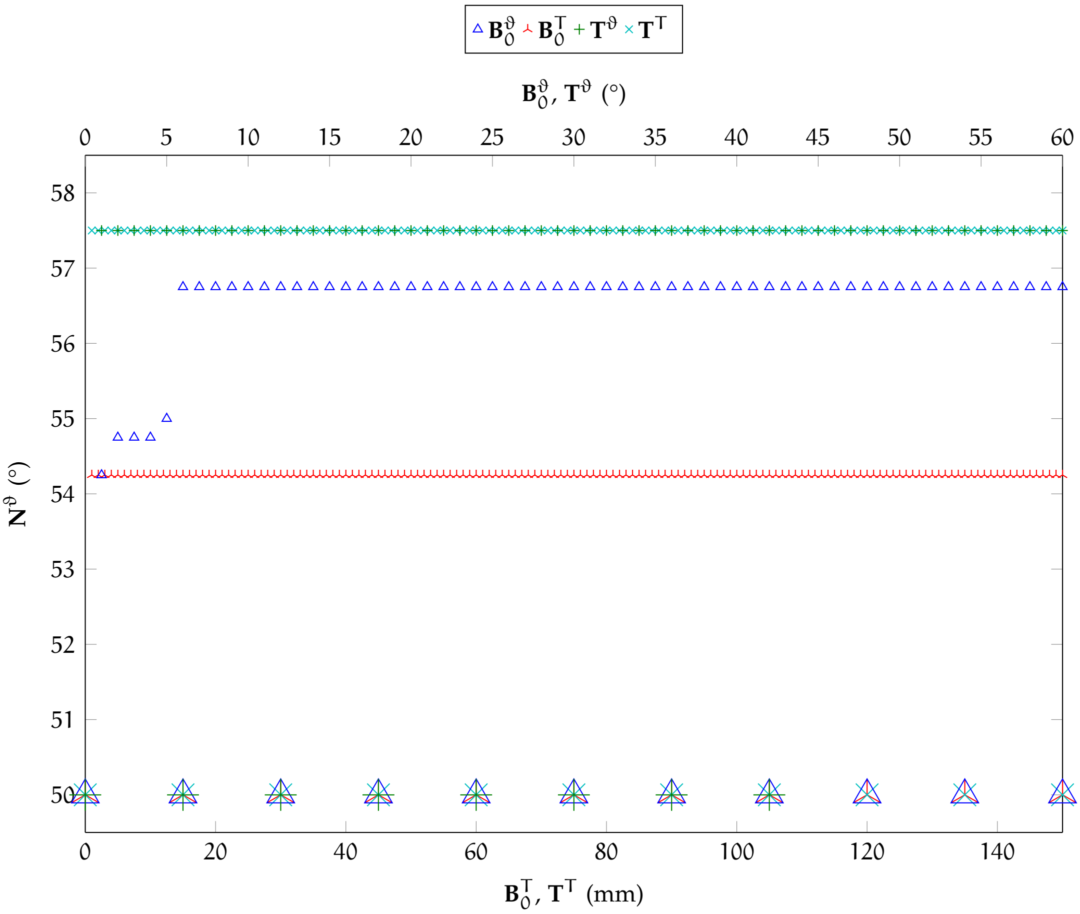


Figure 3.11: Maximum rotation component $\mathbf{N}^\vartheta_\mathrm{MAX}$ which allows the targeting convergence in simulations (small markers) and in experimental (big markers) conditions.

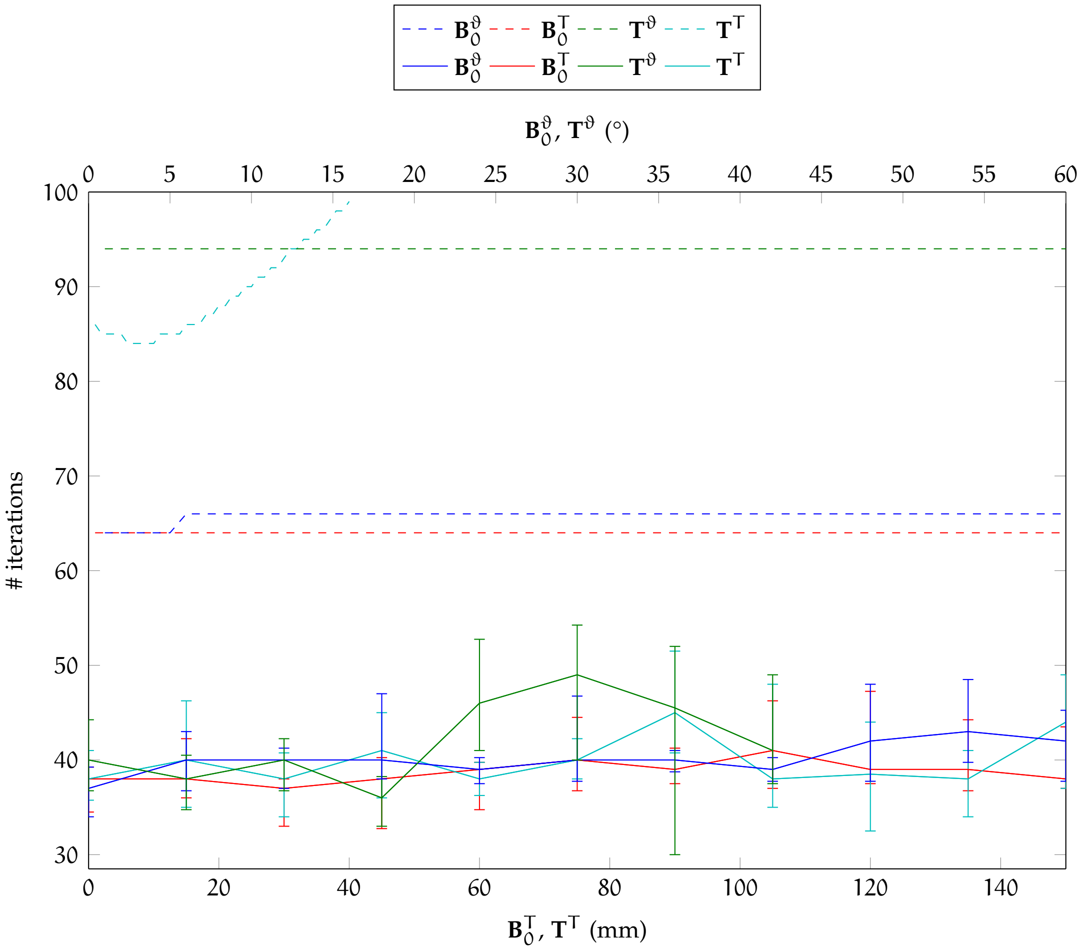


Figure 3.12: Number of iterations required for targeting convergence in simulation (dashed line) with $\mathbf{N}^\vartheta = 57.5°$, and in experimental conditions (solid line, median and percentiles are also reported) with $\mathbf{N}^\vartheta = 50°$.

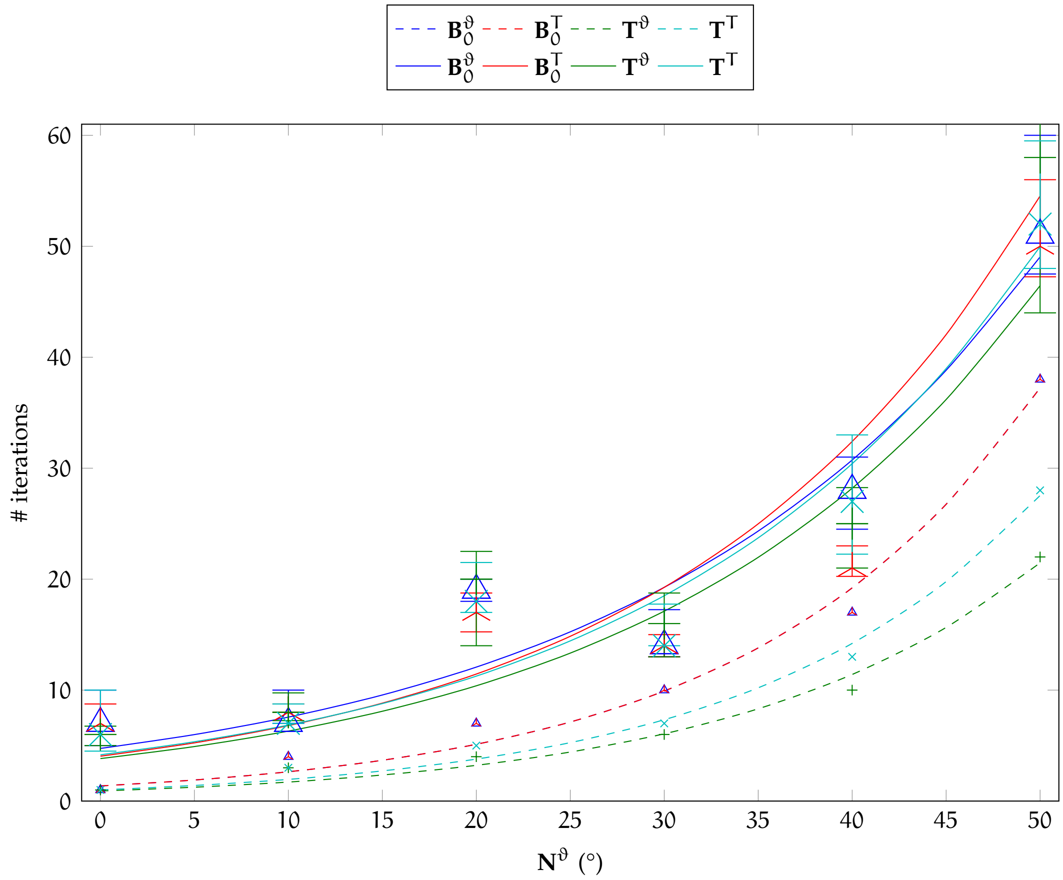


Figure 3.13: Maximum number of targeting iterations required as a function of $\mathbf{N}^\vartheta$, evaluated for all the possible $\mathbf{B}_0$ and $\mathbf{T}$ variations. The lines are the exponential fitting of the data; the dashed lines are simulation results while solid lines are experimental results (with median and quartiles values).

shown in fig. 3.12: convergence was achieved in a finite number of iterations until $\mathbf{N}^{\vartheta}$ was around 55° in simulations and 50° in case of experimental conditions.

3.1.3.4 *Discussions*

In the present section, the convergence of the algorithm presented in section 3.1.2 [48], [49] was verified, numerically and experimentally using a serial robot and an OTS, with respect to modifications of the starting pose of the robot and the target point. Simulations allowed testing convergence extensively exploring (much more than with the experiments) the parameters' space ($\mathbf{B}$, $\mathbf{T}$, $\mathbf{N}$). In the simulation, no noise was added to better understand the behavior of the system in pure conditions.

The convergence of the iterative targeting assures that the obtained accuracy is lower than the selected thresholds (0.8 mm and 0.05°) thus satisfying technical requirements for neurosurgery. Independent from calibration, the targeting accuracy has to be less than 1 mm at the target point. Calibration inaccuracies can be due to noise in the measurements system or robot inaccuracies. Previous works [40], [58] showed that hand-eye calibration errors are in the range of 0.06 mm for translation values and 0.03 rad for rotation values, which are two orders of magnitude less than the $\mathbf{N}^{\vartheta}_{\mathrm{MAX}}$ we computed.

In the tests, only the calibration rotation error was considered, since translations are corrected in a single iteration step. In our setup, the tool to be aligned to the target pose was considered to be coincident with the CF tracked by the external monitoring system. Also, the robots initial pose ($\mathbf{B}_0$) and the tool target pose ($\mathbf{T}$) were varied in range which is compatible with the operating room constraints, both in simulation and in experimental conditions. We showed that targeting convergence was reached independently from $\mathbf{B}_0$ and $\mathbf{T}$ up to errors in the calibration matrix of 60° in simulation, and around 50° in experimental conditions (fig. 3.11).

During keyhole neurosurgery, the robot and target poses could change, e. g. during SEEG procedures, in which up to 18 electrodes are implanted in the brain, generally in a single hemisphere. We showed that, independently from the required task, the robot can reach the target with the required accuracy, provided there is a clear line of sight between the markers attached on the flange and the optical tracker, otherwise the robot is stopped. We showed also that convergence depends only on the amount of angular error in the calibration matrix ($\vartheta < 50°$).

Using the $\mathbf{N}^{\vartheta}_{\mathrm{MAX}}$ transformation matrix to perturb the calibration matrix, also the number of iterations required to converge showed to be almost independent from $\mathbf{B}_0$ and $\mathbf{T}$ variations in both simulations and experimental conditions (fig. 3.12). In experimental conditions, less iterations are needed to reach the target since the $\mathbf{N}^{\vartheta}_{\mathrm{MAX}}$ value used was slightly inferior (50° instead of 60°).

At last, there is an exponential increase in the number of iterations varying the **N** matrix rotation component, both in simulation and in experimental conditions (fig. 3.13). For calibration errors of the same amplitude of the ones reported in literature, the number of requested iterations is approximately 10, which is compatible with the neurosurgical intervention time; in fact, in each iteration the robot movement takes about 3 s which is a small amount of time for the overall process to reach the target compared to the duration of the intervention which takes several hours.

The targeting accuracy experimentally computed in the same conditions satisfies the intervention requirements. In keyhole surgery, where a straight tool (e. g. a probe) has to be accurately positioned, respecting the pre-operative plan, serial and parallel robots can be used. As an example, neurosurgical interventions, such as deep brain stimulation or SEEG, require a targeting accuracy below 1.1 mm. Unfortunately serial robot's accuracy is limited due to possible manufacturing inaccuracies and to intrinsic system compliance: reported accuracy values of serial robots are in the range of (0.5 to 4.4) mm [13], [69], while parallel robots, which are more accurate, suffer of reduced working space.

External sensors, such as optical localization systems, can be used to register the intra-operative reality on the pre-operative images [56], thus allowing the surgical tools navigation and patient motion compensation in several medical application. In trans-cranial magnetic stimulation, [88] used an OTS and a force sensor and defined an hybrid force and position controller to keep the stimulating coil on site, while [89] proposed an online calibration method which calculates the robot base to optical tracker transformation at the set-up phase based on a previous estimation of the calibration matrices, reporting a final accuracy of 2.21 mm as a mean value. Optical localization systems need clear line-of-sight, but they allow increasing the targeting accuracy, as already reported [48], [49]; the targeting strategy presented here relies on the positioning accuracy of the external measurement system, as long as the selected threshold for the application, i. e. the maximum allowed error, reaches the stated accuracy of the tracking system. In our study, the selected threshold is larger than the one of the tracking system used in the experiments section 3.1.3.2 and it is the same order of magnitude of the most accurate commercially available tracking systems. Moreover, that value was selected because it is of same order of magnitude of stereotactic frames (whose positioning accuracy can be influenced by the manual adjustments of the frame) and can be lowered according to the technical accuracy of the used external sensor. Robotic positioning systems present in literature allowed reaching worse targeting accuracy in clinical trials, e. g. the ROBODOC® system allowed positioning the bone cutting guide with 1 mm error [70] and the Cyberknife® system allowed orienting the beam with 1.1 mm error [71]. The Neuromate® system allowed performing brain biopsies with (0.86 ± 0.32) mm accuracy [54]. The parallel robot presented in [72] was able to achieve 0.4 mm accuracy via the kinematic calibration obtained using an optical localizer and the targeting procedure with a closed-loop controller. For the experimental evaluation, we used a KUKA LWR, whose basic accuracy is not satisfying the surgical requirements. This robot features torque sensors in each joint, which increase the safety of the device in the detection of collisions while making the structure to be more flexible than a classic industrial serial robot. It has a redundant kinematic structure, which enables the possibility to avoid obstacles while keeping the end effector in its pose, increasing the workspace with dexterity of the robot, making it to be more suitable in a crowded environment such as the operating room.

The results presented in this section show that iterative algorithms using external devices can give an improvement in the placement accuracy in neurosurgical procedures, where the accuracy is a key factor and the time required for the positioning of the tool is negligible.

In order to use such a versatile system for neurosurgery, the combined use of an accurate external sensor is therefore helpful because its positional accuracy, i. e. the accuracy of the measurement of position and orientation of rigid bodies in space, is higher than the robot accuracy and then it allows the measurement of discrepancies with respect to the desired pose. In those procedures, the access on the brain is reduced to a small hole, causing a reduced loss of cerebrospinal fluid and a reduced brain shift, minimizing the target displacement.

The present algorithm can also be integrated with other environmental sensors for automatic collision avoidance and with a GUI to simulate the robot movements to check possible collisions before the real movement execution.

3.2 TARGET FOLLOWING

3.2.1 Introduction

In open skull surgery with awake patient, the brain tissue suffers of movements due to breathing and heartbeat [90]; moreover, during epilepsy surgery, the patient can have active movements during the brain mapping procedure, unpredictable epileptic seizures, intentional movements of the patient (i. e. for discomfort) or due to the action of the surgeon [91]; all those movements can result in vibrations of the head of the patient, which is fixed on the surgical table through a proper frame.

In case a robot is holding a surgical tool and it is actively performing the intervention in contact with the brain, these movements rise safety issues if the robot doesn't keep the tool on the target.

The problem of moving target was investigated in particular in quasi-periodic movements as beating heart, in the different surgical applications. In this case the problem can be divided in two main streams: the identification and tracking of the movement and the control of the robot to follow the movement. [92] proposed a robust method to estimate the spatial deformation in 3D using stereoscopic images form endoscopes based on time-varying Fourier series, which considers both the heartbeat and breathing movements. [93] proposed a method to control a robot to cancel breathing motion for intervention in the chest, with a robot that can be tele-manipulated using a master device; the authors kept the breathing cancellation separated from the tele-manipulation controller given the fact that the input from the user is not periodic as breathing is. They developed an adaptive approach of a generalized predictive controller to separate the contribution of the motion in the two components due to heartbeat and breathing, which are recorded using a camera at 500 Hz, and to treat them separately to obtain a prediction of the target Point of Interest (POI), and used that to feed the data to the robot, which was custom made to push the acceleration limits. In [94] they used the biological signal as the Electro-Cardiography (ECG) and blood pressure, combined with the position information of the POIs to predict the motion of the hearth and use it to move the robot to compensate the motion of the target.

In the field of visual servoing and motion control in medical applications, controllers for robotic arms were developed to compensate movements of a target, in particular for periodic and predictable movements like heartbeat and breathing [95]. [96] uses visual servoing techniques with a 500 Hz camera to control two DoFs of a robot to follow the planar movement of a beating heart, through the definition of a proper model of the heart motion; similarly, [97] uses a prediction schema based on feature tracking on the surface as well as the signal of the electrocardiogram and breathing pressure to estimate the movement of the heart. In [98] an analysis of the visual servoing techniques is made referring to the most common prediction algorithms that are generally used in cardiac surgery, as model-based or model-free prediction algorithms.

In the field of neurosurgery, no works are published yet about motion compensation during open skull procedures. This is due to the fact that the robotic assistant for these kind of interventions are mainly used as static tool holders or they are moved cooperatively by the surgeon to reach the desired target. This is part of the EU

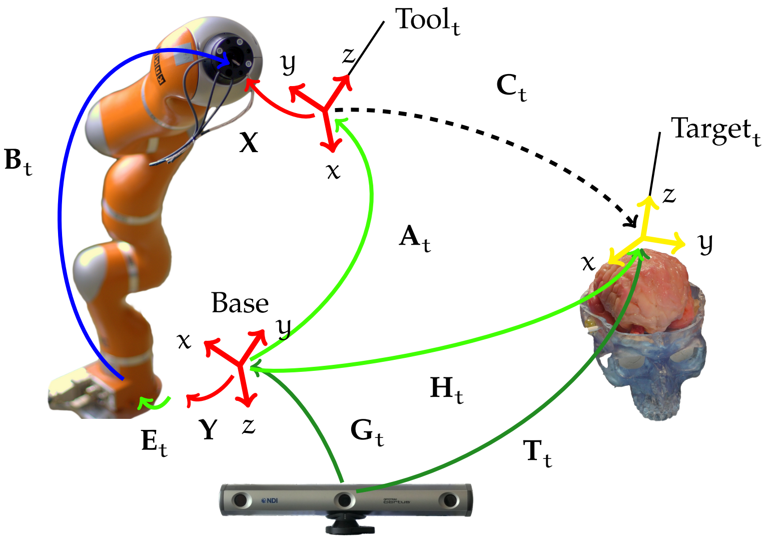


Figure 3.14: Target following kinematic chain representation: the relationship among the robot, the DRFs and the target are represented.

funded project ACTIVE [12], which is pushing the state of the art by developing a multi-robot system for open skull surgery with awake patient. In one of the possible scenarios of this project, the robot is tele-operated by the surgeon which is sitting at a master console; in this console, the user has a monitor, along with the master device, in which (s)he can see the images from a camera pointing on the surgical scene. The information about the motion itself is estimated using a set of 3D cameras that can track features on the brain structure and, based on that data, to reconstruct the motion of the surgical target, providing the information to both the camera-image stabilization and to the robot controller which will automatically compensate the target motion, providing to the surgeon the possibility to work on the master side as if the brain is not moving.

3.2.2 Methods

The control schema here implemented is presented in [78] and it is based on the use of an OTS that can track, as a function of time t, the 3D pose of the robot base $\mathbf{G}_t$, the pose of a tool carried by a robot $\mathbf{A}_t$ and the target pose $\mathbf{T}_t$, as indicated in fig. 3.14. Using the same calibration procedure described in section 3.1.3.1, the matrix $\mathbf{X}$ and $\mathbf{Y}$ can be estimated and they are considered to be constant with respect to time.[4]

[4] within a reasonable limit, which is larger than the time required for the execution of the same session of trials

By closing the kinematic chain on the robot and the OTS, the error $\mathbf{E}_t$ can be calculated with eq. (3.16):

$$\mathbf{E}_t = \mathbf{Y}^{-1} \cdot \mathbf{A}_t \cdot \mathbf{X} \cdot \mathbf{B}_t^{-1} \tag{3.16}$$

where $\mathbf{B}_t$ is the forward kinematics of the robot from the joint encoders readings in the assumption that their errors are negligible.

This error encompasses the contributions due to the measurement noise from the robot encoders and the OTS as long as the contribution due to the approximation from the minimization procedure used to estimate the calibration matrices $\mathbf{X}$ and $\mathbf{Y}$, called *calibration error*. $\mathbf{E}_t$ is a function of the time t because its value changes with the 3D tool pose $\mathbf{A}_t$.[5]

A Low-Pass (LP) filter can be used to reduce the dynamics of the error with respect to the time t, as in eq. (3.17):

$$(\mathbf{E}_t)_f = \text{lowpass}\,(\mathbf{E}_t) \tag{3.17}$$

which is applied on each component of the translation and on each angle of the orientation of the pose.

The new robot pose $\hat{\mathbf{B}}_t$ that causes the tool pose to reach the target pose is the one with $\mathbf{C}_t = \mathbf{I}_4$, the 4D identity matrix, which can be calculated using eq. (3.18):

$$\hat{\mathbf{B}}_t = (\mathbf{E}_t)_f^{-1} \cdot \mathbf{Y}^{-1} \cdot \mathbf{H}_t \cdot \mathbf{X} \tag{3.18}$$

where $\mathbf{H}_t$ is the desired pose of the POI in the CF of the optical DRF of the robot base, calculated as in eq. (3.19):

$$\mathbf{H}_t = \mathbf{G}_t^{-1} \cdot \mathbf{T}_t \tag{3.19}$$

where $\mathbf{G}_t$ is the desired pose of the robot base DRF and $\mathbf{T}_t$ is the target pose, both expressed in the CF of the OTS.

A schematic representation of the controller is shown in fig. 3.15; in this schema, the desired robot pose is fed to the robot controller that, given the initial conditions as current pose and velocity, it calculates the via-points, sampled at the robot controller frequency, that bring the robot on the target pose along the fastest path that respects the maximum velocity and acceleration limits of the robot dynamics.

To test the performances of the target following algorithm, the target pose of the POI in the robot CF, $\mathbf{B}_{t,d}$, was generated using predefined functions. The tool to be kept on the target trajectory is the

[5] the procedure used to estimate the calibration matrices minimize this error on the poses used for the calibration, and thus it is not constant, in particular on a pose which is not one used for the calibration

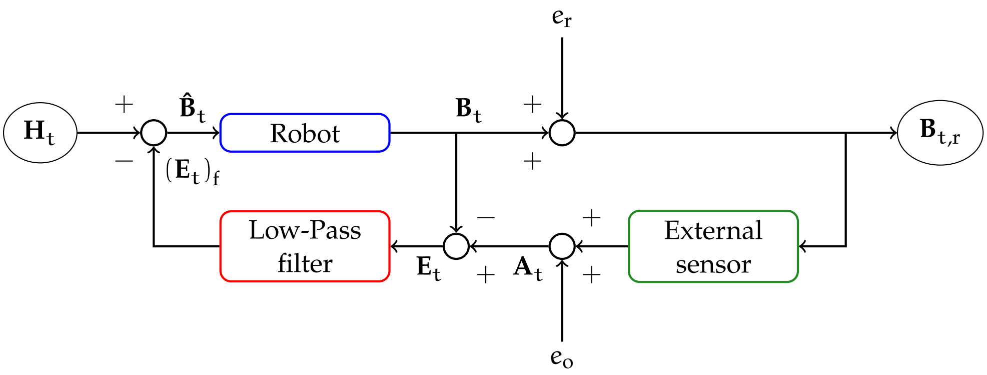


Figure 3.15: Target following algorithm: the schema shows the sources of errors (robot e_r and OTS e_o) and the data flow. $\mathbf{B}_{t,r}$ represents the real pose of the robot, including the positioning error, which can be measured by the OTS. The − sign indicates the inversion of the transformation matrix; the constant contribution of the calibration matrices is omitted in this schema, and the eqs. described in section 3.2.2 have to be used as a reference.

tip of a mock up of a surgical probe, that can be tracked with the OTS through a proper DRF.[6] Instead of using eq. (3.19) to obtain the target pose $\mathbf{H}_t$ in the robot base DRF CF, eq. (3.20) was used:

$$\mathbf{H}_t = \mathbf{Y} \cdot \mathbf{B}_{t,d} \cdot \mathbf{X}^{-1} \tag{3.20}$$

as input for the target following algorithm described in section 3.2.2.

The pre-defined trajectories used for the tests are in eqs. (3.21a) and (3.21b):

$$g_1(t) = \begin{cases} x(t) = x_0 + \frac{d}{2} \cdot \cos(2\pi \cdot f \cdot t) \\ y(t) = y_0 + \frac{d}{2} \cdot \sin(2\pi \cdot f \cdot t) \\ z(t) = z_0 \end{cases} \tag{3.21a}$$

$$g_2(t) = \begin{cases} x(t) = x_0 + vt \\ y(t) = y_0 + at^2 \\ z(t) = z_0 \end{cases} \tag{3.21b}$$

where d, f, v and a are the parameters, respectively diameter and frequency for $g_1(t)$ and velocity and acceleration for $g_2(t)$; the orientation component of the target pose was maintained constant throughout the movement.

[6] a pivoting procedure was performed to directly obtain the information about the tip of the tool from the optical measurements

Table 3.7: Experimental protocol for the target following algorithm: ranges and sampling for the independent variables.

(a) $g_1(t)$

	d	f
Range	[1; 14] mm	[0.5; 4.0] Hz
Sampling	1 mm	0.5 Hz

(b) $g_2(t)$

	v	a
Range	[0.000; 0.010] m/s	[0.01; 0.10] m/s^2
Sampling	0.025 m/s	0.01 m/s^2

In the tests performed, the two parameters were varied according to table 3.7, according to the measurements obtained during real procedures [91]; the trajectories were generated for 20 s in case of $g_1(t)$ and 1.5 s in case of $g_2(t)$ and, in both cases, the TCP was on the trajectory at the beginning of the movement.

The robot and OTS were sampled at 100 Hz, while the LP filter implemented for the error pose of eq. (3.17) is a moving average filter on 50 samples, giving a Full Width at Half Maximum (FWHM) of the frequency response of the filter of 4.8 Hz.

The trajectory of the tool connected to the robot was then recorded through its DRF, converted in the robot base CF using the calibration matrices, and this data was compared with the desired trajectory of the target $\mathbf{B}_{t,d}$, in order to evaluate the tracking error as eq. (3.22)

$$\mathbf{D}_t = \mathbf{B}_{t,d}^{-1} \cdot \mathbf{B}_t \tag{3.22}$$

and the tracking delay on each axis, from the cross-correlation of the translation components of the two signals $\mathbf{B}_{t,d}$ and $\mathbf{B}_t$ on the three axis. The data were processed prior to the extraction of the delay with a LP filter with a cut-off frequency of 0.5 Hz higher than the expected bandwidth content of the generated signal,[7] in order to remove the noise of the optical measurement; after that, the data were fitted in order to reconstruct the functions of eqs. (3.21a) and (3.21b) in order to be able to re-sample the signals at 1000 Hz, to increase the accuracy of the delay extraction.

In order to measure the performances of the tracking, the Root Mean Square (RMS) error of the translation component of $\mathbf{D}_t$ was evaluated on each axis; the tracking is considered good when the

[7] from eqs. (3.21a) and (3.21b)

RMS error is below the threshold of 1.5 mm, which is the Target Registration Error (TRE) [99] of the DRF corresponding to the tip of the tool used in the experiments.[8] This threshold defines the regions of the parameters' space ((d, f) and (a, v) respectively) in which the performance in terms of RMS error are acceptable or not; given the shape of the results (see figs. 3.23 and 3.24 on page 71 and on page 72) these regions are separated by a boundary curve that can be identified throug a fitting. In the case here presented, the selected fitting functions are the hyperbola in eq. (3.23a) for $g_1(t)$ and the first order polynomial function eq. (3.23b) for $g_2(t)$:

$$d \cdot f = k_1 \tag{3.23a}$$

$$a = k_{2_a} \cdot v + k_{2_b} \tag{3.23b}$$

where k_1, k_{2_a} and k_{2_b} are the fitting parameters that describe the curves of the boundaries of the two regions. In particular, the RMS error is acceptable for $d \cdot f \leqslant k_1$ in case of $g_1(t)$ and for $a \leqslant k_{2_a} \cdot v + k_{2_b}$ in case of $g_2(t)$.

The input pose $\mathbf{H}_t$ was used in the control schema using two approaches: 1) by feeding the generated target pose $\mathbf{H}_t$ directly to the controller, or 2) by feeding the controller with a target pose that comes from a prediction one-step in the future $\hat{\mathbf{H}}_{t+\Delta t}$, where Δt is the sampling interval.

To perform the prediction $\hat{\mathbf{H}}_{t+\Delta t}$, a Kalman Filter (KF) approach was chosen. In order to have a general representation of the pose of a rigid body which avoids singularities, the orientation component was described using quaternions and doing so adds non-linearity to the system model that can be handled using extended versions of KF. The Unscented Kalman Filter (UKF) is an extension of the classical KF to non-linear processes and measurement models, which reduces the computational power required and also doesn't require the explicit linearization of the system, leading to better results [100]. The algorithm here implemented is based on a second order model for the translation component of the pose and a first order model for the orientation [101], and takes as input $\mathbf{H}_t$, either generated using eqs. (3.21a) and (3.21b) or measured online by the OTS, and it provides the $\hat{\mathbf{H}}_{t+\Delta t}$.

3.2.3 Results

In fig. 3.16 is shown an example of the data for the two functions in eqs. (3.21a) and (3.21b) as the target trajectory to be followed, along

[8] the tool is a mock-up of a surgical probe, shown in fig. 2.1 on page 8, which main axis is long approximately 20 cm, increasing the TRE of the localization of the DRF

Table 3.8: Fitting parameters for the surface threshold boundary: the parameters are shown with their 95 % confidence bounds and the R^2 of the fitting.

(a) Fitting parameters of eq. (3.23a)

	Data	k_1	Confidence bounds	R^2
No prediction	x	0.0077	(0.0068, 0.0087)	0.9423
	y	0.0075	(0.0067, 0.0083)	0.9645
Prediction	x	0.0075	(0.0067, 0.0084)	0.9583
	y	0.0074	(0.0067, 0.0081)	0.9725

(b) Fitting parameters of eq. (3.23b)

	Data	k_{2_a}	Confidence bounds	k_{2_b}	Confidence bounds	R^2
No prediction	x	0.000	(0.000, 0.000)	0.010	(0.010, 0.010)	—
	y	0.800	(−0.670, 2.270)	0.002	(0.015, 0.033)	0.500
	z	< -0.001	(< -0.001, < 0.001)	0.060	(0.060, 0.060)	—
Prediction	x	0.000	(0.000, 0.000)	0.010	(0.010, 0.010)	—
	y	1.200	(−0.073, 2.473)	0.018	(0.010, 0.026)	0.750
	z	0.800	(−0.670, 2.270)	0.054	(0.045, 0.063)	0.500

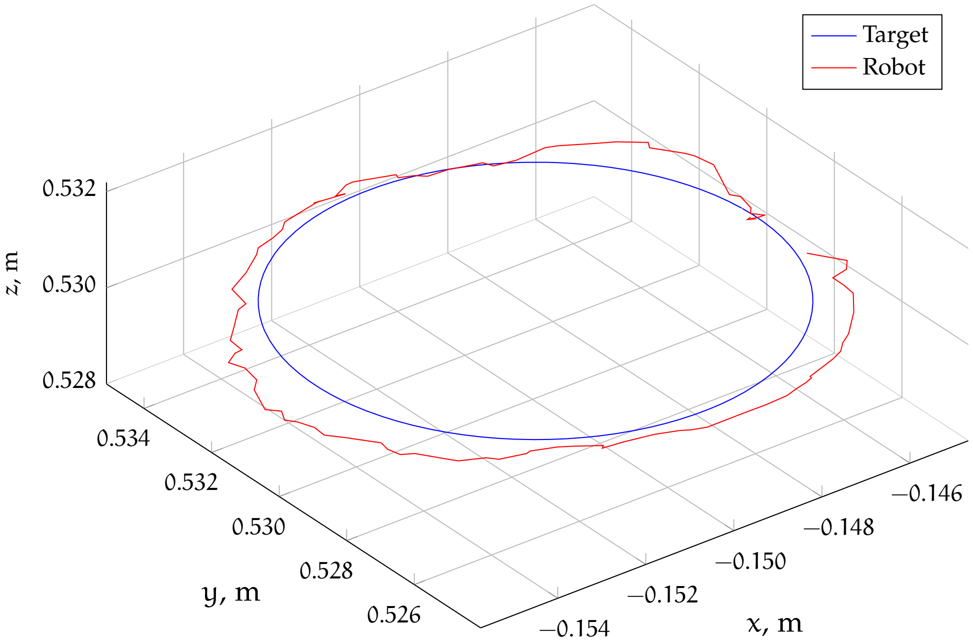
with the real path performed by the robot, as measured by the OTS; both of them are reported in the CF of the robot base.

The RMS errors of the tracking for the three components is presented in figs. 3.17 to 3.19 for the $g_1(t)$ movement, while in figs. 3.20 to 3.22 for $g_2(t)$.

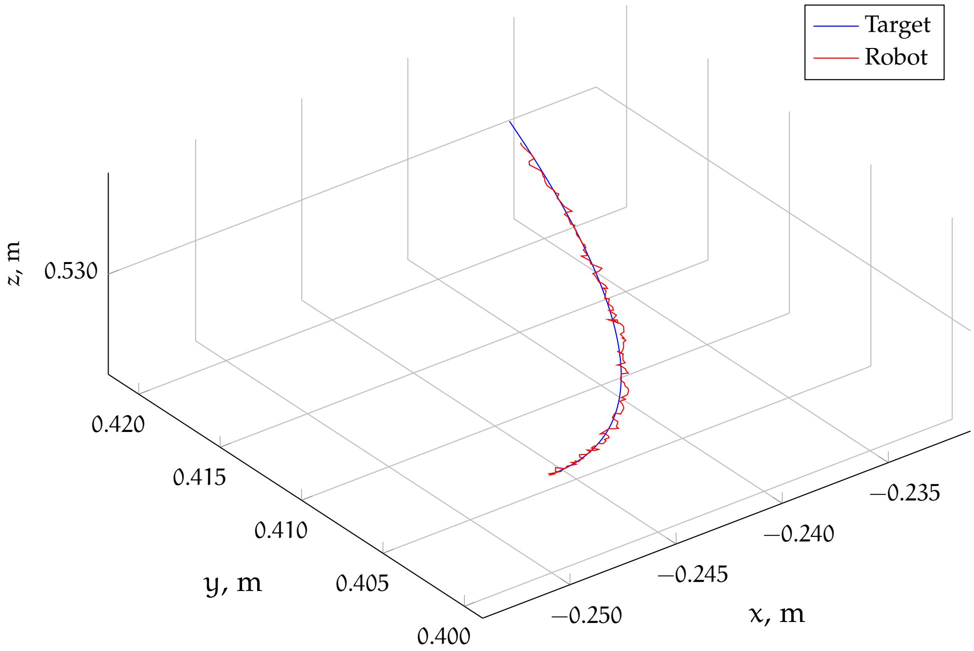
In figs. 3.23 and 3.24, the surfaces in figs. 3.17, 3.18 and 3.20 to 3.22 are presented after a cut with a plane parallel to the parameters' plane at a RMS error value of 1.5 mm, along with the fitting curves approximated with the functions in eqs. (3.23a) and (3.23b), whose results are reported in table 3.8.

As regards the results about $g_1(t)$ in figs. 3.17 and 3.18, the RMS error shows reduced performances for large values of the parameters d and f, with an higher RMS error for $f = 3\,\text{Hz}$. The results about $g_2(t)$ in figs. 3.20 to 3.22, instead show an RMS error with reduced performances for increasing values of the parameters a, while the performances are approximately constant as function of v.

The tracking delays are reported in table 3.9 on the three axis for both $g_1(t)$ and $g_2(t)$; the larger delays are obtained for $g_1(t)$ for $f \geqslant 3$ Hz (data not shown), while no significant cross-correlation was found in the signals for $g_2(t)$ and for the z axis of $g_1(t)$.

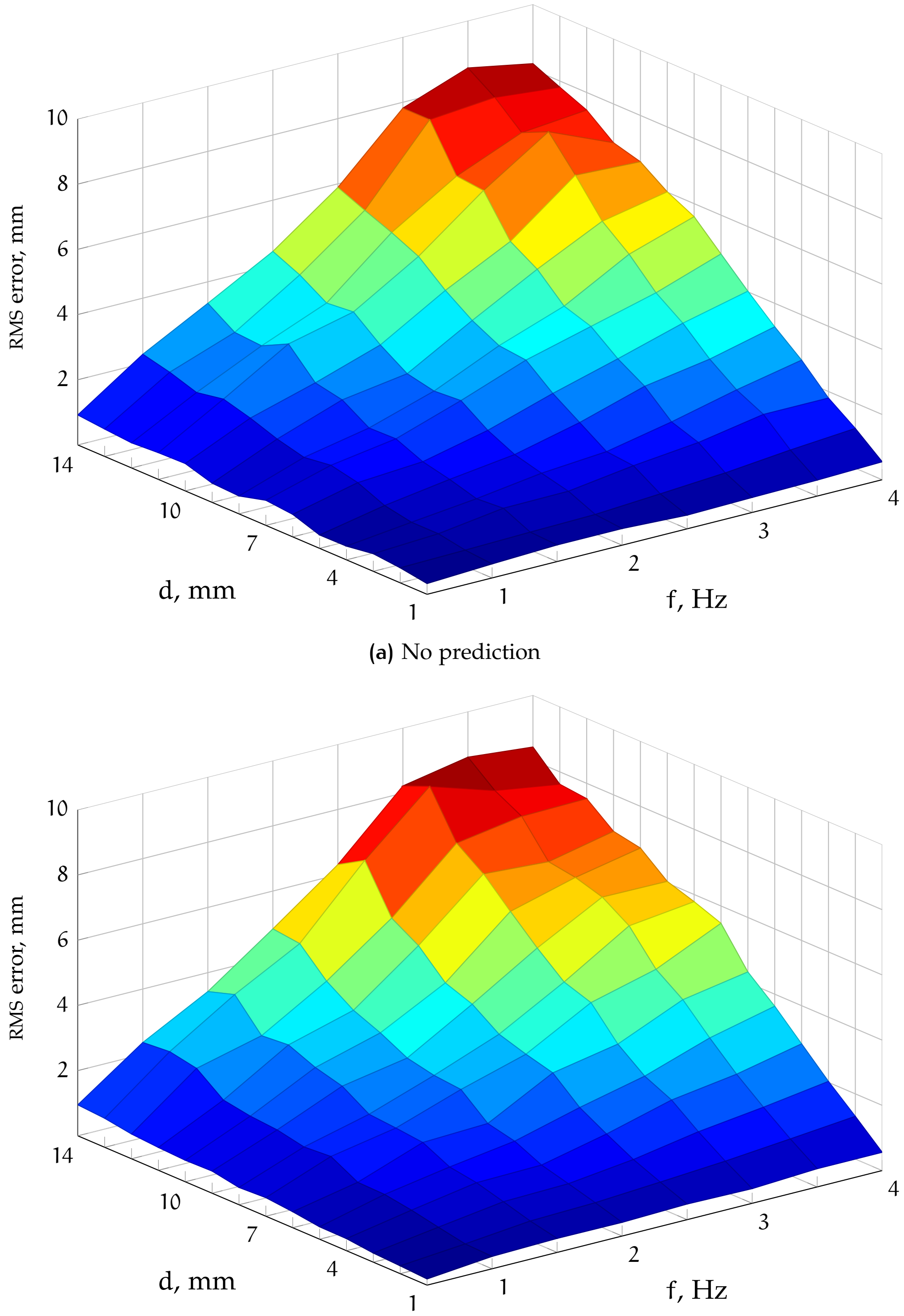


(a) $g_1(t)$: $d = 0.010\,\mathrm{m}$, $f = 1.0\,\mathrm{Hz}$



(b) $g_2(t)$: $v = 0.010\,\mathrm{m/s}$, $a = 0.010\,\mathrm{m/s^2}$

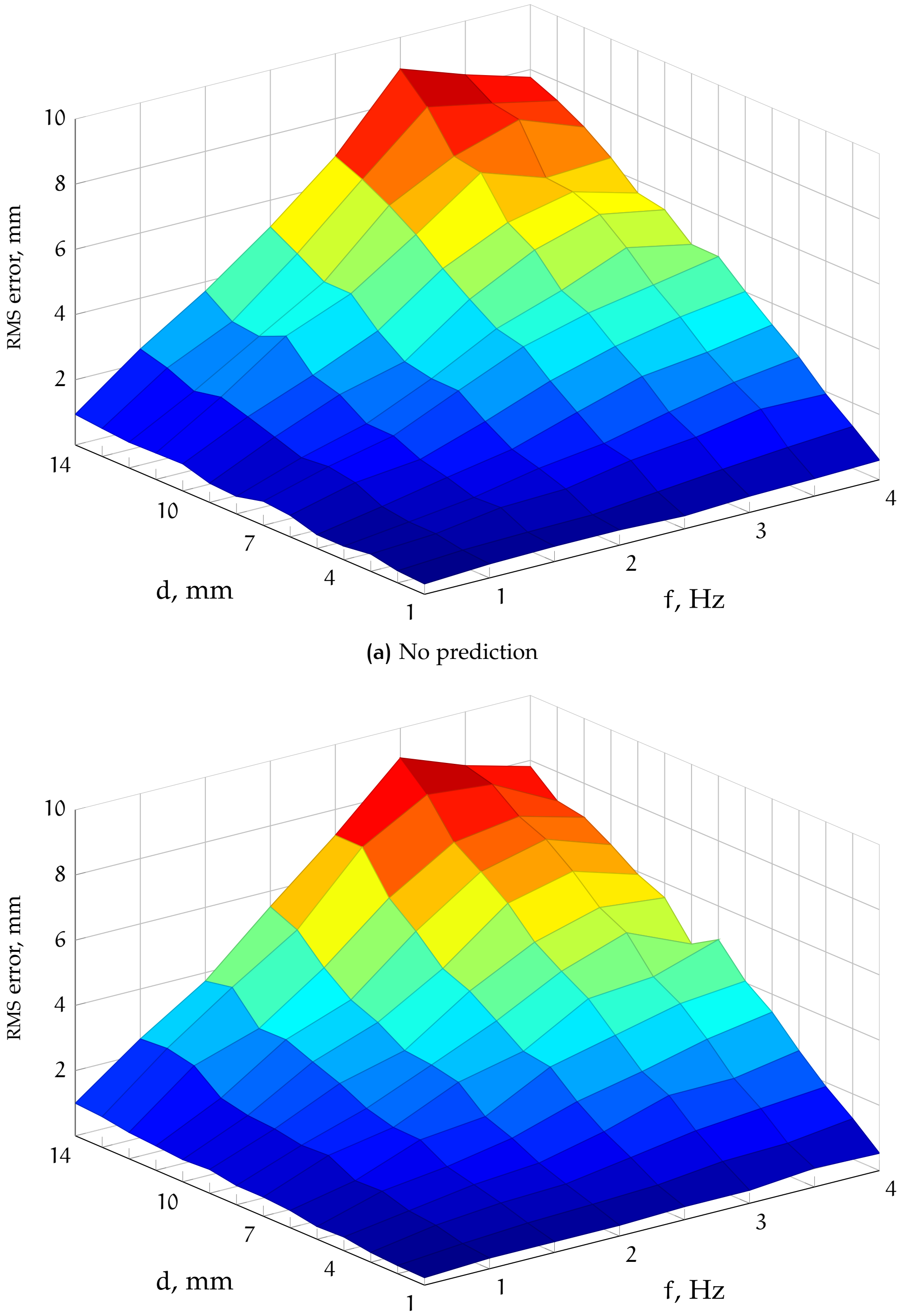
Figure 3.16: Target following trajectories: example of desired and robot actual trajectory.



(a) No prediction

(b) Prediction

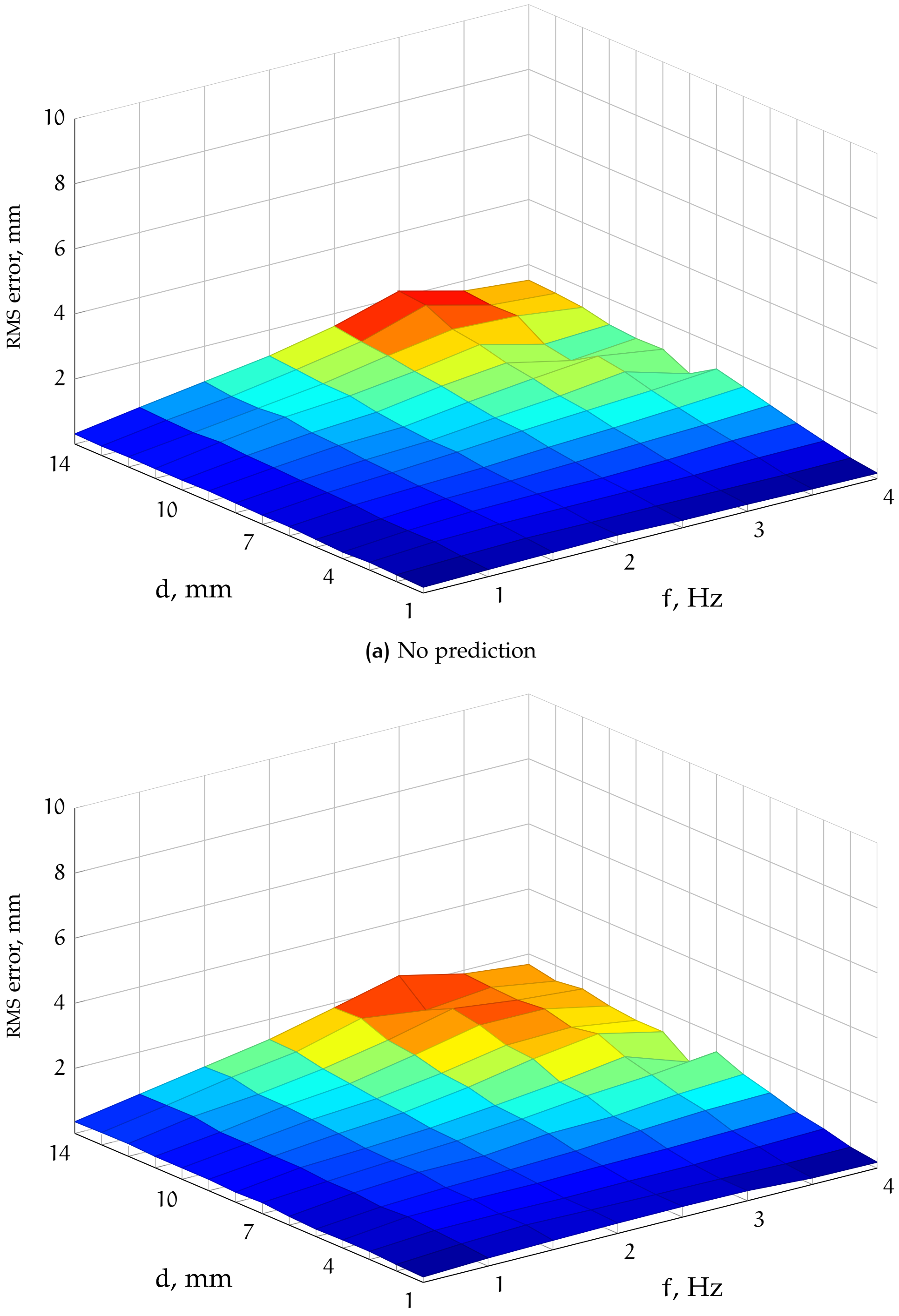
Figure 3.17: RMS error during the $g_1(t)$ movement: x axis.



(a) No prediction

(b) Prediction

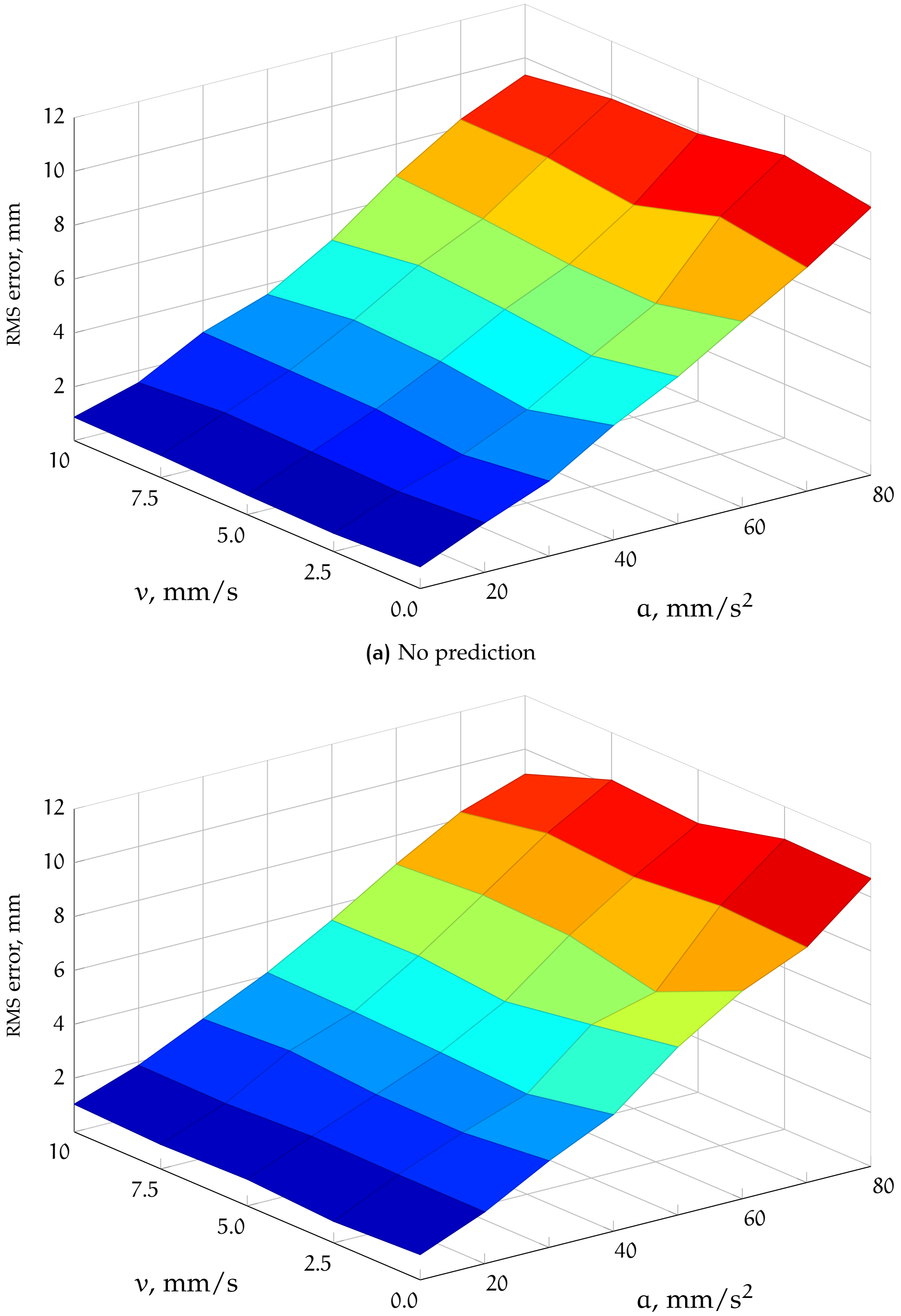
Figure 3.18: RMS error during the $g_1(t)$ movement: y axis.



(a) No prediction

(b) Prediction

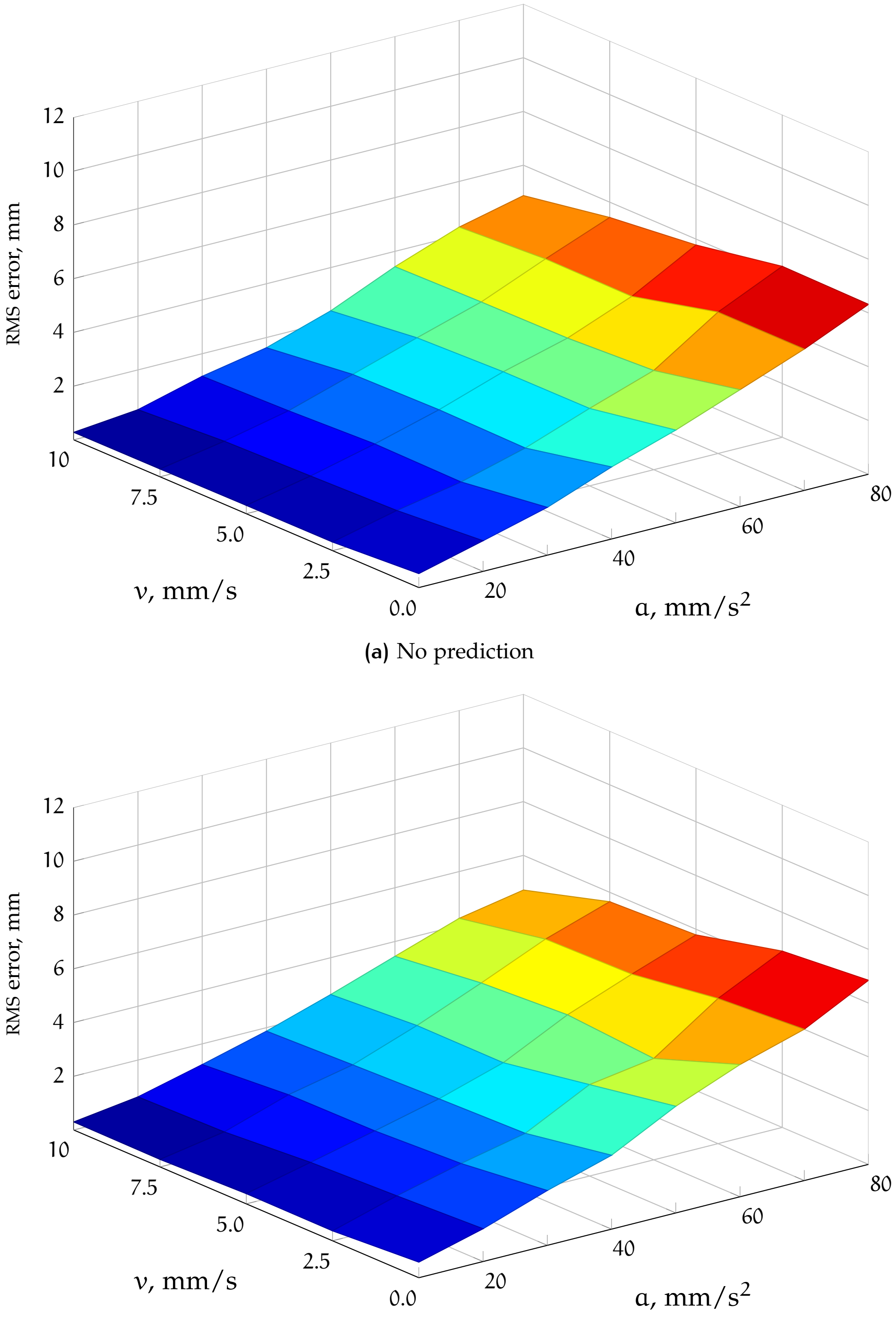
Figure 3.19: RMS error during the $g_1(t)$ movement: z axis.



(a) No prediction

(b) Prediction

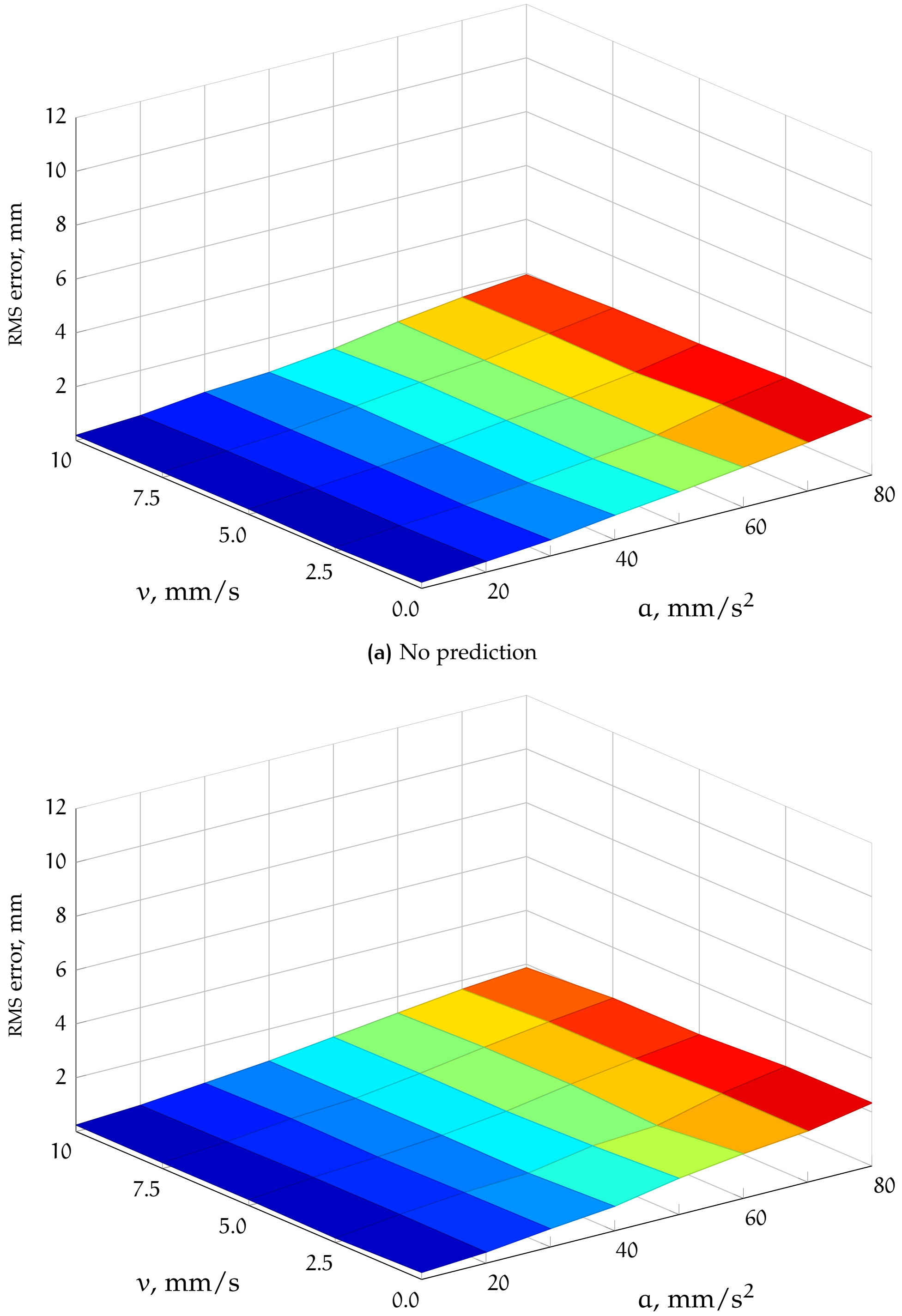
Figure 3.20: RMS error during the $g_2(t)$ movement: x axis.



(a) No prediction

(b) Prediction

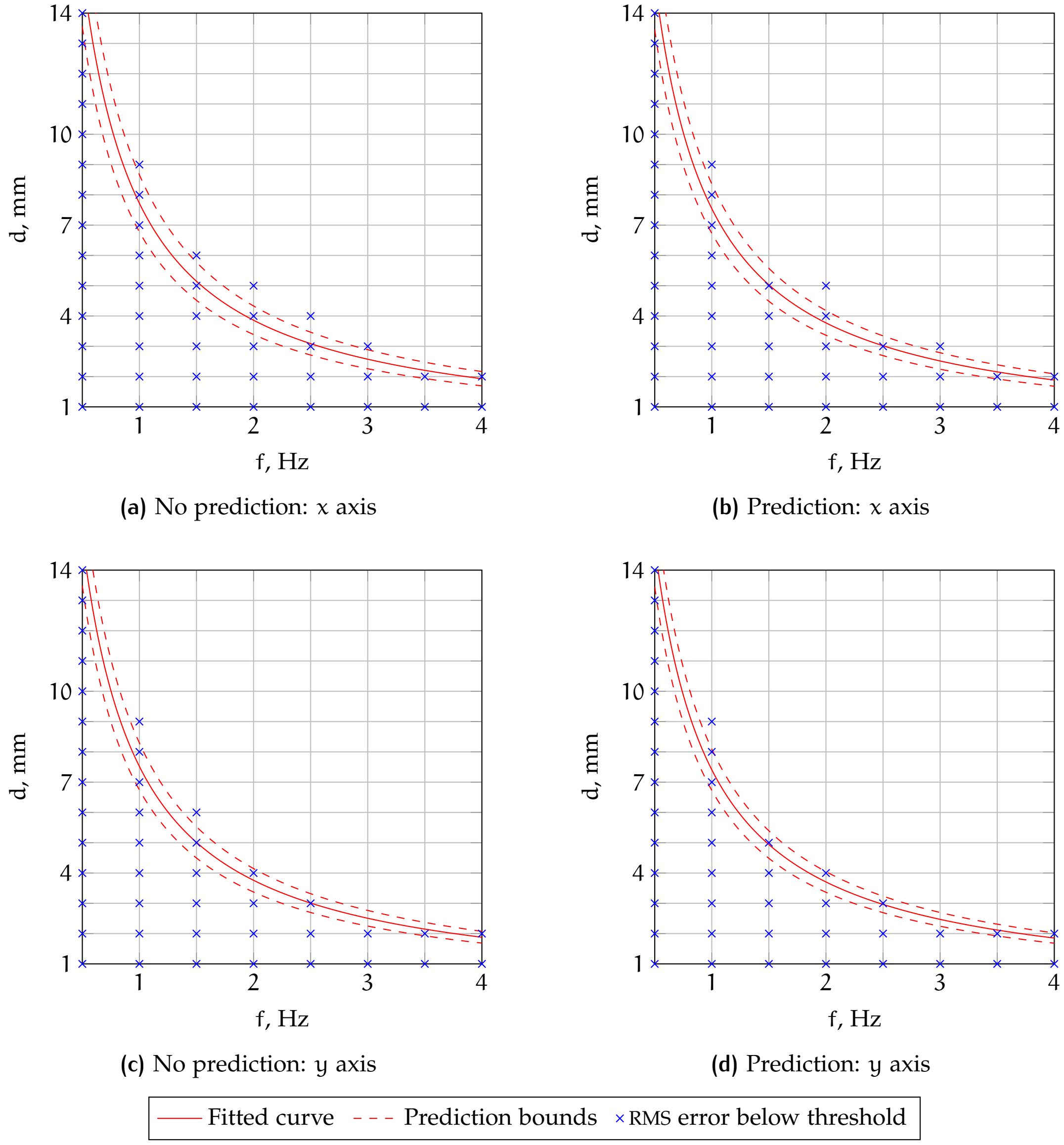
Figure 3.21: RMS error during the $g_2(t)$ movement: y axis.



(a) No prediction

(b) Prediction

Figure 3.22: RMS error during the $g_2(t)$ movement: z axis.



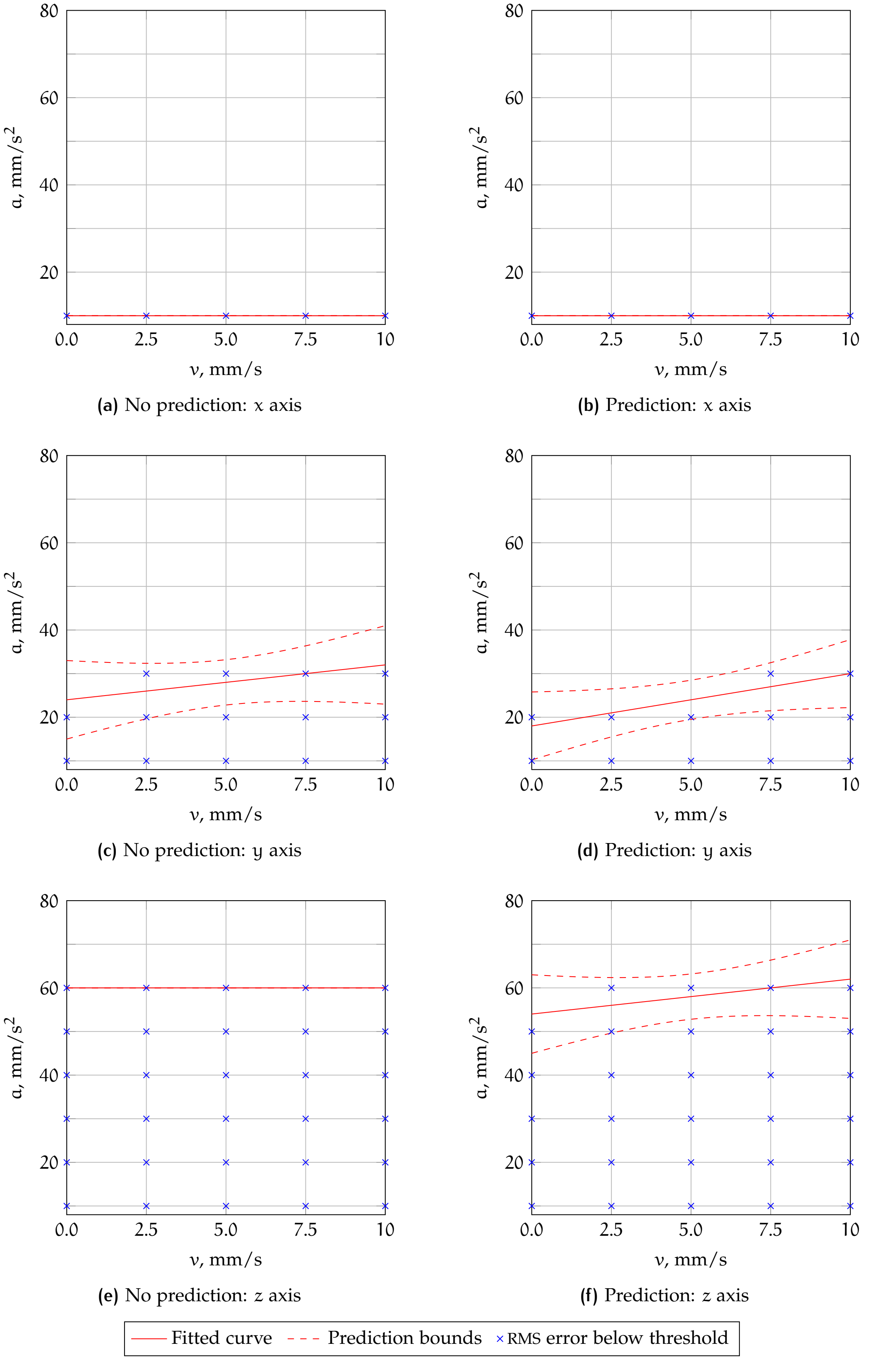
(a) No prediction: x axis

(b) Prediction: x axis

(c) No prediction: y axis

(d) Prediction: y axis

Figure 3.23: RMS error below threshold during the $g_1(t)$ movement: the data and the fitting functions are presented.



(a) No prediction: x axis

(b) Prediction: x axis

(c) No prediction: y axis

(d) Prediction: y axis

(e) No prediction: z axis

(f) Prediction: z axis

Figure 3.24: RMS error below threshold during the $g_2(t)$ movement: the data and the fitting functions are presented.

Table 3.9: Time delays between target and robot trajectories: the median is presented along with the inter-quartile range in brackets (25^{th} and 75^{th} percentiles); for polynomial functions (as z of $g_1(t)$ and for $g_2(t)$), no significant cross-correlation was found, to estimate the delay.

Function	Data	Delay, s	
		No prediction	Prediction
$g_1(t)$	x	0.08 (0.07, 0.18)	0.08 (0.07, 0.20)
	y	0.08 (0.07, 0.19)	0.08 (0.07, 0.09)
	z	—	—
$g_2(t)$	x	—	—
	y	—	—
	z	—	—

3.2.4 Discussions

In this section it was presented how to use the architecture developed in chapter 2, to follow the movement of a target in space. The analysis was conducted by investigating the relationship between frequency and amplitude of the motion and also with respect to accelerations and velocities of the movement.

As regards the relationship between frequency and diameter of the motion studied with eq. (3.21a), the result showed that for movements with a limited bandwidth, the system is able to obtain a tracking error which are of the same order of the TRE of the localization of the DRF defined on the surgical tool, which satisfies the requirements of the depicted application in neurosurgery. Despite of that, high frequencies components of the movement show a decreasing of the performances, in particular if associated with a large range of movement. Worse results are obtained with a frequency of 3 Hz, which is close to the limits of the dynamic ranges of the robot; with higher frequencies the robot is not able to follow the target and then the delays and the tracking errors are increased; the degraded performances for higher frequencies are more evident for larger movements, in which the effects of the delayed and incorrect motion of the robot are more visible. The controller uses trajectory interpolators (see section 2.3.2.3) which are designed to move the robot along the fastest path to the target point given the current velocity and position, under the constraints of the maximum acceleration and velocity that the robot can reach; in case the movement has bandwidth which is larger than what the robotic arm can do, the robot moves along the chords of the circle $g_1(t)$, which are the only possibility to reach the target that is rapidly moving on a circular trajectory. The fitting parameters in table 3.8a can be re-conducted to a limit on the speed along the curvilinear abscissa of the circle drawn by $g_1(t)$; in fact the parameter d is the diameter of the circle, while f represents the number of circles drawn for the unit of time and so an estimate of the velocity is $v_s = \pi \cdot d \cdot f$; the product $d \cdot f$ is the fitting parameter k and thus it is $v_s = \pi k$ which is approximately $0.025\,\mathrm{m/s}$.

As regards the relationship between velocity and accelerations of the movement studied with eq. (3.21b), the acceleration of the movement is the real limit which makes the performance to decrease. This is due to the fact that the delays of the complete architecture make it difficult for the controller to recover after the beginning of the movement and to reduce then the tracking error. In this case, the limit identified by the fitting parameters in table 3.8b can be re-conducted to a line with constant $a = 0.01\,\mathrm{m/s^2}$, which gives a velocity of approximately $0.03\,\mathrm{m/s}$ on the y axis, which is compatible with the limit found for the $g_1(t)$ movement.

This means that for those categories of movements, the limit is the velocity of the movement itself along the trajectory.

In [93] the authors presented a method to estimate the motion of the heart due to heartbeat and the respiratory movement; they developed an algorithm to estimate and predict the heart motion by separating the components due to heartbeat and respiratory movement; the lab experiments results showed an error which is approximately 10 % of the movement and it requires an initial time to learn the characteristics of the disturbance motion to be able to remove it. In [94] they used the ECG signal as a trigger to anticipate the mechanical motion of the heart, with a tracking accuracy of 0.9 mm as RMS error value. The results presented in the current section are obtained with a model-free approach, without any assumption on the motion of the POI, as the movements in neurosurgical environment usually are [91].

The prediction algorithm here implemented is used with constant orientation, but the implementation works also in the general case in which both translations and orientations are changing over time. The use of predicting algorithms doesn't reduce the tracking delays, which are mostly due to computational and communication delays in the LAN. To reduce those problems, a better analysis of the source of the delays in the complete system have to be performed, new solutions with dedicated hardware for the more power-demanding calculations have to be investigated, along with the development of predicting algorithms that can predict the pose at more than one sample ahead; for doing so, higher bandwidth in the control loops have to be used, as long as an higher sampling frequency of the measurement of the motion from the OTS, also including sensor fusion algorithms with Inertial Measurement Units (IMUs).

4 CONCLUSIONS

> Experience is what you get when you didn't get what you wanted.
>
> Randolph Frederick Pausch
> The Last Lecture – 2007

In this thesis, an architecture for robot control for neurosurgical robotic operations was described. In order to help the surgeon during the intervention, the user may need to change the behavior of the device at high level to cope with the different needs of the phases of the intervention. For doing so, an architecture with an UI was implemented to automatically handle the switching phase among behaviors keeping the safety of the system.

The architecture presented in chapter 2[1] is modular and scalable, which means that new functionalities can be added to the schema by simply connecting the new components and adding them properly to the FSMs, with also the possibility to improve the current functionality by substituting the current components with the new one, without the need to change and reprogram the architecture of the controller. ROS and OROCOS proved to be a flexible and hardware independent tool to implement both high and low level controllers for robotic applications, with the possibility to distribute the computational power over a dedicated LAN, using machines that can withstand the computational power required by the application. In order to keep the time synchronization of all the machines included in the architecture, all the computers share the same clock over the network ensuring a timing accuracy of 0.5 ms with the reference machine.

The control architecture handles the different situations that can happen during a surgical procedure, ensuring that the transition do not cause unpredictable behavior of the robot arm, which is the machine that can effectively harm the patient with the surgical tool attached to it. The robot arm used in the architecture, the LWR 4+, is made to share its workspace with the operators, and for doing so, internal controllers are implemented to effectively cope with situations that can be potentially dangerous for the operators, i. e. by using the information coming from the torque sensors present on each joint. These sensors can also be used to detect contacts with obstacles or operators and, based on this information, to eventually enable the

[1] from the contributions Comparetti *et al.* [28], [29]

emergency procedures to maintain the safety, or to manually guide the robot towards the target.

In chapter 3[2] algorithms to position and orient tools carried by a robot were presented. The discussion was conducted in two different conditions: 1) with a static target in section 3.1, and 2) with a moving target in section 3.2.

In section 3.1.2 an accuracy evaluation of a targeting algorithm was presented. The iterative approach described proved to ensure a positioning accuracy which is compatible with the neurosurgical application, while in section 3.1.3 a convergence evaluation of the algorithm was performed in order to test the robustness of the algorithm with respect to errors in the estimation of the pose of the tool on the target. This algorithm can be applied to already existing commercial solutions which encompass a robot and an optical localizer, without the need to change the controllers of the devices, but simply using their data to improve the positioning accuracy. The final achieved targeting accuracy is the one of the external sensor used, which is the one used for the registration of the pre-operative images with the intra-operative reality, making it consistent with the accuracy level of the procedure. Future works are directed towards increasing the safety of the system so that the chosen path towards the target is clear of obstacles. Performance analysis in a real operating room for electrodes positioning during SEEG will be also investigated.

In section 3.2 the developed architecture was used to track a target that moves in space, as can happen during a neurosurgical intervention with awake patient. The results presented show that the tracking error is compatible with the surgical procedure for slow motions, while for faster movements both the delays and tracking errors are not suitable for the application; however, those movements are outside the ranges of movement registered in the clinical practice [91] and are close to the limits of the robot device. Possible solutions to improve the performances of the tracking, the measurement of the desired pose have to be acquired at higher frequencies; sensor fusion algorithms with OTSs and IMUs can also be used to extend the bandwidth of the information about the position of the target, as long as prediction algorithms that can estimate the future pose with at a time that is as long as the delay of the movement of the robot can improve the results.

In neurosurgical applications each intervention is prepared in the pre-operative phase with the acquisition of medical images on which the surgeon defines the target region(s) for the surgery; during the intra-operative phase, the pre-operative images are registered with the current reality to obtain the information in the same CF and to enable the possibility of intra-operative navigation. In this process, the accuracy of the reconstruction of the target for the surgical in-

[2] from the contributions Comparetti *et al.* [47]–[50]

tervention in the intra-operative phase is determined by the whole processing of the information, which defines the requirements of accuracy for the robotic actuators that perform the surgery, resulting in an approximate estimation of this requirement of 1 mm [102], which was used in the present thesis as a reference value to verify the performances of the algorithms developed.

BIBLIOGRAPHY

[1] V. Horseley, R. Clark, A. Raabe, and V. Seifert, "The structure and functions of the cerebellum examined by a new method.", *Brain*, vol. 31, pp. 45–124, 1908.

[2] E. Spiegel, H. Wycis, M. Marks, and A. Lee, "Stereotaxic apparatus for operations on the human brain", *Science*, vol. 106, no. 2754, pp. 349–350, 1947.

[3] J. Talairach, *Atlas d'anatomie stéréotaxique: repérage radiologique indirect des noyaux gris centraux des régions mésencéphalo-sous-optique et hypothalamique de l'homme*. Masson, 1957.

[4] G. N. Hounsfield, "Computerized transverse axial scanning (tomography): part 1. description of system", *British Journal of Radiology*, vol. 46, no. 552, pp. 1016–1022, 1973.

[5] K. Cleary, V. Watson, D. Lindisch, A. Patriciu, D. Mazilu, and D. Stoianovici, "Robotically assisted interventions: clinical trial for spinal blocks", in *Medical Image Computing and Computer-Assisted Intervention-MICCAI 2003*, Springer, 2003, pp. 963–964.

[6] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions", *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[7] A. Lanfranco, A. Castellanos, J. Desai, and W. Meyers, "Robotic surgery - a current perspective", *Annals of Surgery*, vol. 239, no. 1, pp. 14–21, Jan. 2004.

[8] R. H. Taylor and D. Stoianovici, "Medical robotics in computer-integrated surgery", *IEEE Transactions on Robotics and Automation*, vol. 19, no. 5, pp. 765–781, Oct. 2003.

[9] I. K. McLeod and P. C. Melder, "Da vinci robot-assisted excision of a vallecular cyst: a case report", *Ear, nose & throat journal*, vol. 84, pp. 170–172, 2005.

[10] R. H. Taylor, S. Lavallé, G. C. Burdea, and R. Mösges, *Computer-Integrated Surgery, Technology and Clinical Applications*. MIT Press, 1995.

[11] P. Kazanzides, G. Fichtinger, G. D. Hager, A. M. Okamura, L. L. Whitcomb, and R. H. Taylor, "Surgical and interventional robotics - core concepts, technology, and design", *IEEE Robotics & Automation Magazine*, vol. 15, no. 2, pp. 122–130, Jun. 2008.

[12] Politecnico di Milano. (Nov. 16, 2013). Active Constraints Technologies for Ill-defined or Volatile Environments – ACTIVE, [Online]. Available: http://www.active-fp7.eu.

[13] M. Eljamel, “Validation of the PathFinder™ neurosurgical robot using a phantom”, *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 3, no. 4, pp. 372–377, 2007.

[14] S. Lavallee, “A new system for computer assisted neurosurgery”, in *Engineering in Medicine and Biology Society. Images of the Twenty-First Century., Proceedings of the Annual International Conference of the IEEE Engineering in*, IEEE, 1989, pp. 926–927.

[15] A. Benabid, P. Cinquin, S. Lavalle, J. Lebas, J. Demongeot, and J. Derougemont, “Computer-driven robot for stereotactic surgery connected to ct scan and magnetic-resonance imaging - techological design and preliminary-results”, *Applied Neurophysiology*, vol. 50, no. 1–6, pp. 153–154, 1987.

[16] P. B. McBeth, D. F. Louw, P. R. Rizun, and G. R. Sutherland, “Robotics in neurosurgery”, *American Journal of Surgery*, vol. 188, no. 4A, Suppl. S, 68S–75S, Oct. 2004.

[17] C. Karas and E. Chiocca, “Neurosurgical robotics: a review of brain and spine applications”, *Journal of Robotic Surgery*, vol. 1, no. 1, pp. 39–43, 2007.

[18] T. Fukushima and K. Sano, “Simple retractor holder for the mayfield skull clamp.”, *Surgical neurology*, vol. 13, no. 4, p. 320, 1980.

[19] G. Dogangil, B. L. Davies, and F. Rodriguez y Baena, “A review of medical robotics for minimally invasive soft tissue surgery”, *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 224, no. H5, Sp. Iss. SI, pp. 653–679, 2010.

[20] T. R. K. Varma and P. Eldridge, “Use of the Neuromate stereotactic robot in a frameless mode for functional neurosurgery”, *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 2, no. 2, pp. 107–113, 2006.

[21] M. Cossu, G. Lo Russo, S. Francione, R. Mai, L. Nobili, I. Sartori, L. Tassi, A. Citterio, N. Colombo, M. Bramerio, C. Galli, L. Castana, and F. Cardinale, “Epilepsy surgery in children: results and predictors of outcome on seizures”, *Epilepsia*, vol. 49, no. 1, pp. 65–72, Jan. 2008.

[22] G. Sutherland, I. Latour, and A. Greer, “Integrating an image-guided robot with intraoperative MRI: a review of the design and construction of NeuroArm.”, *IEEE engineering in medicine and biology magazine: the quarterly magazine of the Engineering in Medicine & Biology Society*, vol. 27, no. 3, p. 59, 2008.

[23] P. Kazanzides, T. Xia, C. Baird, G. Jallo, K. Hayes, N. Nakajima, and N. Hata, “A cooperatively-controlled image guided robot system for skull base surgery.”, *Studies in health technology and informatics*, vol. 132, p. 198, 2008.

[24] S. Ferrand-Sorbets, D. Taussig, M. Fohlen, C. Bulteau, G. Dorfmuller, and O. Delalande, "Frameless stereotactic robot-guided placement of depth electrodes for stereo-electroencephalography in the presurgical evaluation of children with drug-resistant focal epilepsy", in *CNS Annual Meeting*, 2010.

[25] T. Haidegger, L. Kovacs, G. Fordos, Z. Benyo, and P. Kazanzides, "Future trends in robotic neurosurgery", in *Proceedings of the 14th Nordic-Baltic Conference on Biomedical Engineering and Medical Physics*, ser. IFMBE Proceedings, vol. 20, Springer Berlin Heidelberg, 2008, pp. 229–233.

[26] Politecnico di Milano. (Dec. 27, 2013). ROBOt and sensor integration for Computer Assisted Surgery and Therapy – ROBOCAST, [Online]. Available: http://www.robocast.eu.

[27] E. De Momi and G. Ferrigno, "Robotic and artificial intelligence for keyhole neurosurgery: the ROBOCAST project, a multi-modal autonomous path planner", *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 224, no. H5, Sp. Iss. SI, pp. 715–727, 2010.

[28] M. D. Comparetti, E. De Momi, D. De Lorenzo, T. Beyl, J. Raczkowsky, and G. Ferrigno, "Safe surgical robotic system and workflow design in the active project for awake neurosurgery", in *Proceedings of the 2012 IEEE International Conference on Intelligent Robots and Systems (IROS, Vilamoura), Workshop on Safety in Human-Robot Coexistence & Interaction: How Can Standardization and Research benefit from each other?*, Oct. 2012.

[29] M. D. Comparetti, E. De Momi, T. Beyl, M. Kunze, J. Raczkowsky, and G. Ferrigno, "Event-based device-behavior switching in surgical human-robot interaction", in *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA, Hong Kong)*, Accepted, May 2014.

[30] R. Bischoff, J. Kurth, G. Schreiber, R. Koeppe, A. Albu-Schäffer, A. Beyer, O. Eiberger, S. Haddadin, A. Stemmer, G. Grunwald, *et al.*, "The KUKA-DLR lightweight robot arm. a new reference platform for robotics research and manufacturing", in *41^{st} International Symposium on Robotics (ISR) and 6^{th} German Conference on Robotics (ROBOTIK)*, 2010, pp. 1–8.

[31] Y. F. Li, "A sensor-based robot transition control strategy", *The International journal of robotics research*, vol. 15, no. 2, pp. 128–136, 1996.

[32] Apache. (Aug. 16, 2012). CouchDB, [Online]. Available: http://couchdb.apache.org/.

[33] Object Management Group. (Aug. 23, 2010). Common Object Request Broker Architecture (CORBA), [Online]. Available: http://www.corba.org/.

[34] Digia. (Oct. 27, 2013). Qt, [Online]. Available: http://qt.digia.com/.

[35] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source Robot Operating System", in *IEEE International Conference on Robotics and Automation, Workshop on Open Source Software*, vol. 3, 2009.

[36] Robot Operating System (ROS). (Oct. 27, 2013). Open source platform for robotics, [Online]. Available: http://www.ros.org.

[37] H. Bruyninckx, "OROCOS: design and implementation of a robot control software framework", in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) - Tutorial*, 2002.

[38] Open RObot COntrol Software (OROCOS). (Oct. 27, 2013). Realtime environment for robot control, [Online]. Available: http://www.orocos.org.

[39] OpenNI, Open Natural Interaction. (Sep. 10, 2012). Open source platform for natural interaction, [Online]. Available: http://www.openni.org.

[40] E. De Momi, P. Cerveri, E. Gambaretto, M. Marchente, O. Effretti, S. Barbariga, G. Gini, and G. Ferrigno, "Robotic alignment of femoral cutting mask during total knee arthroplasty", *International Journal of Computer Assisted Radiology and Surgery*, vol. 3, no. 5, pp. 413–419, 2008.

[41] D. De Lorenzo, S. Calò, A. Ciullo, M. D. Comparetti, M. Kunze, E. De Momi, and G. Ferrigno, "Redundancy management of a LWR4+ for safe, collision free, robotic surgical applications, using environmental sensors", in *Proceedings of the Third Joint Workshop on New Technologies for Computer/Robot Assisted Surgery*, vol. 1, Sep. 2013, pp. 61–64.

[42] H. Moradi and S. Lee, "Joint limit analysis and elbow movement minimization for redundant manipulators using closed form method", in *Advances in Intelligent Computing*, Springer, 2005, pp. 423–432.

[43] V. Penza, "Dynamic tool compensation in a bilateral telemanipulation system for neurosurgery", Master's thesis, Politecnico di Milano, Oct. 2013.

[44] G. Zeng and A. Hemami, "An overview of robot force control", *Robotica*, vol. 15, no. 5, pp. 473–482, 1997.

[45] M. Shimizu, H. Kakuya, W.-K. Yoon, K. Kitagaki, and K. Kosuge, "Analytical inverse kinematic computation for 7-dof redundant manipulators with joint limits and its application to redundancy resolution", *Robotics, IEEE Transactions on*, vol. 24, no. 5, pp. 1131–1142, 2008.

[46] F. Janabi-Sharifi, V. Hayward, and C.-S. Chen, "Discrete-time adaptive windowing for velocity estimation", *IEEE Transactions on Control Systems Technology,* vol. 8, no. 6, pp. 1003–1009, 2000.

[47] M. D. Comparetti, M. Riechmann, A. Cappelletti, E. De Momi, A. Vaccarella, H. Wörn, and G. Ferrigno, "Pose correction algorithm for a robot in a neurosurgical application", in *Proceedings of the 2010 International Congress of Computer Assisted Radiology and Surgery (CARS, Geneva),* vol. 5, Jun. 2010, pp. 304–306.

[48] M. D. Comparetti, E. De Momi, A. Vaccarella, M. Riechmann, and G. Ferrigno, "Optically tracked multi-robot system for keyhole neurosurgery", in *Proceedings of the 2011 IEEE International Conference on Robotics and Automation (ICRA, Shangai),* May 2011, pp. 661–666.

[49] M. D. Comparetti, A. Vaccarella, I. Dyagilev, M. Shoham, G. Ferrigno, and E. De Momi, "Accurate multi-robot targeting for keyhole neurosurgery based on external sensors monitoring", *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine,* vol. 226, no. 5, pp. 347–359, May 2012.

[50] M. D. Comparetti, E. De Momi, T. Beyl, M. Kunze, J. Raczkowsky, and G. Ferrigno, "Convergence analysis of iterative targeting method for image guided robotic surgery", *International Journal of Advanced Robotic Systems,* 2013, Accepted.

[51] L. Joskowicz, R. Shamir, M. Freiman, M. Shoham, E. Zehavi, F. Umansky, and Y. Shoshan, "Image-guided system with miniature robot for precise positioning and targeting in keyhole neurosurgery", *Computer Aided Surgery,* vol. 11, no. 4, pp. 181–193, 2006.

[52] M. J. Lee and E. L. Lin, "The use of the three-pronged mayfield head clamp resulting in an intracranial epidural hematoma in an adult patient", *European Spine Journal,* vol. 19, pp. 187–189, 2010.

[53] R. Shamir, M. Freiman, L. Joskowicz, M. Shoham, E. Zehavi, and Y. Shoshan, "Robot-assisted image-guided targeting for minimally invasive neurosurgery: planning, registration, and in-vitro experiment", *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2005,* pp. 131–138, 2005.

[54] Q. Li, L. Zamorano, A. Pandya, R. Perez, J. Gong, and F. Diaz, "The application accuracy of the Neuromate robot - a quantitative comparison with frameless and frame-based surgical localization systems", *Computer Aided Surgery,* vol. 7, no. 2, pp. 90–98, 2002.

[55] R. A. Castillo Cruces, H. Christian Schneider, and J. Wahrburg, "Medical robotics", in. I-Tech Education and Publishing, 2008, ch. Cooperative Robotic System to Support Surgical Interventions.

[56] T. Xia, C. Baird, G. Jallo, K. Hayes, N. Nakajima, N. Hata, and P. Kazanzides, "An integrated system for planning, navigation and robotic assistance for skull base surgery", *International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 4, no. 4, pp. 321–330, Dec. 2008.

[57] J. Burgner, J. Raczkowsky, and H. Wörn, "End-effector calibration and registration procedure for robot assisted laser material processing: tailored to the particular needs of short pulsed CO_2 laser bone ablation", in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*, IEEE Press, 2009, pp. 2136–2141.

[58] K. H. Strobl and G. Hirzinger, "Optimal hand-eye calibration", in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS, Beijing)*, 2006, pp. 4647–4653.

[59] R. Horaud and F. Dornaika, "Hand-eye calibration", *The international journal of robotics research*, vol. 14, no. 3, pp. 195–2, 1995.

[60] K. Daniilidis, "Hand-eye calibration using dual quaternions", *The International Journal of Robotics Research*, vol. 18, no. 3, pp. 286–298, 1999.

[61] I. Fassi and G. Legnani, "Hand to sensor calibration: a geometrical interpretation of the matrix equation $A \cdot X = X \cdot B$", *Journal of Robotic Systems*, vol. 22, no. 9, pp. 497–506, 2005.

[62] J. Jin, S. Ryu, J. Rock, K. Faber, M. Gates, S. Li, and B. Movsas, "Image-guided target localization for stereotactic radiosurgery: accuracy of 6D versus 3D image fusion", *Radiosurgery-Basel*, vol. 6, p. 50, 2006.

[63] H. M. J. van der Linden-van der Zwaag, J. Bos, H. J. L. van der Heide, and R. G. H. H. Nelissen, "A computed tomography based study on rotational alignment accuracy of the femoral component in total knee arthroplasty using computer-assisted orthopaedic surgery", *International Orthopaedics*, vol. 35, no. 6, pp. 845–850, 2011.

[64] M. Shoham, M. Burman, L. Joskowicz, E. Batkilin, and Y. Kunicher, "Bone-mounted miniature robot for surgical procedures: concept and clinical applications", *IEEE Transaction on Robotics and Automation*, vol. 19, no. 5, pp. 893–901, Oct. 2003.

[65] W. Sukovich, S. Brink-Danan, and M. Hardenbrook, “Miniature robotic guidance for pedicle screw placement in posterior spinal fusion: early clinical experience with the SpineAssist®”, *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 2, no. 2, pp. 114–122, 2006.

[66] F. Cardinale, M. Cossu, L. Castana, M. Schiariti, A. Miserocchi, G. Casaceli, C. Caborni, A. Moscato, and G. Lo Russo, “Five hundreds stereoelectroencephalography (SEEG) procedures for epilepsy surgery: a retrospective analysis of clinical safety and in vivo application accuracy”, in *International Journal of Computer Assisted Radiology and Surgery*, In press, Jun. 2012.

[67] C. M. Oliveira, H. T. Nguyen, A. R. Ferraz, K. Watters, B. Rosman, and R. Rahbar, “Robotic surgery in otolaryngology and head and neck surgery: a review”, *Minimally Invasive Surgery*, vol. 2012, pp. 1–11, 2012.

[68] C. J. Coulson, A. P. Reid, D. W. Proops, and P. N. Brett, “ENT challenges at the small scale”, *The International Journal of Medical Robotics and Computer Assisted Surgery*, vol. 3, no. 2, pp. 91–96, 2007.

[69] P. Morgan, T. Carter, S. Davis, A. Sepehri, J. Punt, P. Byrne, A. Moody, and P. Finlay, “The application accuracy of the pathfinder neurosurgical robot”, in *International Journal of Computer Assisted Radiology and Surgery*, H. U. Lemke, K. Inamura, K. Doi, M. W. Vannier, A. G. Farman, and J. H. C. Reiber, Eds., 17th International Congress and Exhibition of Computer Assisted Radiology and Surgery, London, England, June 25-28, vol. 1256, Berlin, Jun. 2003, pp. 561–567.

[70] S. Nishihara, N. Sugano, T. Nishii, H. Tanaka, N. Nakamura, H. Yoshikawa, and T. Ochi, “Clinical accuracy evaluation of femoral canal preparation using the ROBODOC system”, *Journal of Orthopaedic Science*, vol. 9, no. 5, pp. 452–461, 2004.

[71] S. D. Chang, W. Main, D. P. Martin, I. Gibbs, and M. P. Heilbrun, “An analysis of the accuracy of the CyberKnife: a robotic frameless stereotactic radiosurgical system”, *Neurosurgery*, vol. 52, no. 1, pp. 140–147, 2003.

[72] O. Bebek, M. Hwang, and M. C. Cavusoglu, “Design of a parallel robot for needle-based interventions on small animals”, *IEEE/ASME Transactions on Mechatronics*, vol. 18, no. 1, pp. 1–12, 2011.

[73] M. S. Eljamel, “Robotic neurological surgery applications: accuracy and consistency or pure fantasy?”, *Stereotactic and Functional Neurosurgery*, vol. 87, no. 2, pp. 88–93, 2009.

[74] R. Tarwala and L. D. Dorr, "Robotic assisted total hip arthroplasty using the MAKO platform", *Current reviews in musculoskeletal medicine,* vol. 4, no. 3, pp. 151–156, 2011.

[75] J. Cornellà, O. J. Elle, W. Ali, and E. Samset, "Improving cartesian position accuracy of a telesurgical robot", in *IEEE International Symposium on Industrial Electronics, 2008. ISIE 2008,* 2008, pp. 1261–1266.

[76] ——, "Intraoperative navigation of an optically tracked surgical robot", in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2008, PT II, proceedings,* D. Metaxas, L. Axel, G. Fichtinger, and G. Szekely, Eds., ser. Lecture notes in Computer Science, vol. 5242, 2008, pp. 587–594.

[77] S. Baron, H. Eilers, B. Munske, J. L. Toennies, R. Balachandran, R. F. Labadie, T. Ortmaier, and R. J. Webster III, "Percutaneous inner-ear access via an image-guided industrial robot system", *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine,* vol. 224, no. H5, Sp. Iss. SI, pp. 632–648, 2010.

[78] A. Tobergte, F. A. Fröhlich, M. Pomarlan, and G. Hirzinger, "Towards accurate motion compensation in surgical robotics", in *Proceedings of the 2010 IEEE International Conference on Robotics and Automation (ICRA, Anchorage),* IEEE, 2010, pp. 4566–4572.

[79] M. Zimmermann, R. Krishnan, A. Raabe, and V. Seifert, "Robot-assisted navigated endoscopic ventriculostomy: implementation of a new technology and first clinical results", *Acta Neurochirurgica,* vol. 146, no. 7, pp. 697–704, 2004.

[80] M. D. Comparetti, A. Vaccarella, D. De Lorenzo, G. Ferrigno, and E. De Momi, "Multi-robotic approach for keyhole neurosurgery: the ROBOCAST project", in *Proceedings of the 2011 Joint Workshop on New technologies for Computer/Robot Assisted Surgery, Graz, Austria,* Jul. 2011, pp. 1–4.

[81] G. Deacon, A. Harwood, J. Holdback, D. Maiwand, M. Pearce, I. Reid, M. Street, and J. Taylor, "The Pathfinder image-guided surgical robot", *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine,* vol. 224, no. H5, Sp. Iss. SI, pp. 691–713, 2010.

[82] D. De Lorenzo, R. Manganelli, I. Dyagilev, A. Formaglio, E. De Momi, D. Prattichizzo, M. Shoham, and G. Ferrigno, "Miniaturized rigid probe driver with haptic loop control for neurosurgical interventions", in *Proceedings of the 2010 3rd IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics, Tokyo, Japan,* 2010, pp. 522–527.

[83] A. Vaccarella, P. Cerveri, E. De Momi, and G. Ferrigno, "A new IGSTK-based architecture for the integration of multimodal sensors and robots in neurosurgical robotics applications", in *Proceedings of the 24th International Congress and Exhibition,* vol. 5, Jun. 2010, pp. 308–309.

[84] S. Kumar, P. Premkumar, A. Dutta, and L. Behera, "Visual motor control of a 7DoF redundant manipulator using redundancy preserving learning network", *Robotica,* vol. 28, pp. 1–16, 2009.

[85] S. J.-S. Chen, P. Hellier, J.-Y. Gauvrit, M. Marchal, X. Morandi, and D. L. Collins, "An anthropomorphic polyvinyl alcohol triple-modality brain phantom based on colin27", *IEEE Transactions on Medical Imaging,* vol. 13, no. 2, pp. 92–100, 2010.

[86] P. Besl and N. McKay, "A method for registration of 3D shapes", *IEEE Transactions on pattern analysis and machine intelligence,* vol. 14, no. 2, pp. 239–256, 1992.

[87] J. B. West and C. Maurer, "Designing optically tracked instruments for image-guided surgery", *IEEE Transactions on Medical Imaging,* vol. 23, no. 5, pp. 533–545, May 2004.

[88] X. Yi and R. Bicker, "Design of a robotic transcranial magnetic stimulation system", PhD thesis, University of Newcastle upon Tyne, 2012.

[89] L. Richter, F. Ernst, A. Schlaefer, and A. Schweikard, "Robust real-time robot–world calibration for robotized transcranial magnetic stimulation", *The International Journal of Medical Robotics and Computer Assisted Surgery,* vol. 7, no. 4, pp. 414–422, 2011.

[90] M. E. Wagshul, P. K. Eide, and J. R. Madsen, "The pulsating brain: a review of experimental and clinical studies of intracranial pulsatility", *Fluids Barriers CNS,* vol. 8, no. 1, p. 5, 2011.

[91] D. De Lorenzo, E. De Momi, L. Conti, E. Votta, M. Riva, E. Fava, L. Bello, and G. Ferrigno, "Intra-operative forces and moments analysis on patient head clamp during awake brain surgery", *Medical & Biological Engineering & Computing,* 2012, submitted.

[92] R. Richa, A. P. L. Bó, and P. Poignet, "Towards robust 3D visual tracking for motion compensation in beating heart surgery", *Medical Image Analysis,* vol. 15, no. 3, pp. 302–315, 2011.

[93] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M. M. A. Sanchez, and J. Marescaux, "Active filtering of physiological motion in robotized surgery using predictive control", *IEEE Transactions on Robotics,* vol. 21, no. 1, pp. 67–79, 2005.

[94] O. Bebek and M. C. Cavusoglu, "Intelligent control algorithms for robotic-assisted beating heart surgery", *IEEE Transactions on Robotics,* vol. 23, no. 3, pp. 468–480, 2007.

[95] C. N. Riviere, J. Gangloff, and M. de Mathelin, “Robotic compensation of biological motion to enhance surgical accuracy”, *Proceedings of the IEEE,* vol. 94, no. 9, pp. 1705–1716, Sep. 2006.

[96] R. Ginhoux, J. Gangloff, M. de Mathelin, L. Soler, M. M. A. Sanchez, and J. Marescaux, “Beating heart tracking in robotic surgery using 500 hz visual servoing, model predictive control and an adaptive observer”, in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA),* IEEE, vol. 1, 2004, pp. 274–279.

[97] T. Ortmaier, M. Groger, D. H. Boehm, V. Falk, and G. Hirzinger, “Motion estimation in beating heart surgery”, *IEEE Transactions on Biomedical Engineering,* vol. 52, no. 10, pp. 1729–1740, Oct. 2005.

[98] W. Bachta, P. Renaud, L. Cuvillon, E. Laroche, A. Forgione, and J. Gangloff, “Motion prediction for computer-assisted beating heart surgery”, *IEEE Transactions on Biomedical Engineering,* vol. 56, no. 11, pp. 2551–2563, 2009.

[99] J. West and J. Fitzpatrick, “The distribution of target registration error in rigid-body, point-based registration”, in *Information Processing in Medical Imaging,* Springer, 1999, pp. 460–465.

[100] S. J. Julier and J. K. Uhlmann, “Unscented filtering and nonlinear estimation”, *Proceedings of the IEEE,* vol. 92, no. 3, pp. 401–422, 2004.

[101] N. Enayati, “Quaternion based unscented kalman filter for robust motion tracking in neurosurgery”, Master’s thesis, Politecnico di Milano, Oct. 2013.

[102] R. Shamir, L. Joskowicz, S. Spektor, and Y. Shoshan, “Localization and registration accuracy in image guided neurosurgery: a clinical study”, *International Journal of Computer Assisted Radiology and Surgery,* vol. 4, pp. 45–52, 1 2009.

COLOPHON

This work was written in LaTeX using the ClassicThesis style from André Miede, inspired by *The Elements of Typographic Style* from Robert Bringhurst.

December 2013.