Technische Universität Braunschweig

Master´s Thesis

Robot guided neuro tissue sensor

Pablo J. Quílez Velilla

Supervisor: Dr.-Ing. Ralf Westphal

21.08.2012

Institut für Robotik und Prozessinformatik

Prof. Dr.-Ing. Friedrich M. Wahl

Declaration in lieu of oath

I hereby declare in lieu of oath that I composed this dissertation without assistance from third parties, and with no additional help other than the literature and means referred to. This thesis has never been made partially available, in this or any other form to any audit board for the purpose of obtaining a degree.

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne Benutzung anderer als der angegebenen Literatur und Hilfsmittel angefertigt habe. Die Arbeit ist in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde zur Erlangung eines akademischen Grades vorgelegt worden.

Pablo Quilez, Braunschweig 21.08.2012

Abstract

Brain surgery requires high precision and complex surgical techniques whose success depends largely on the surgeon's skill. Different technologies for diagnosing such as computer assisted imaging computed tomography (CT), positron emission tomography (PET), magnetic resonance imaging (MRI) or magnetoencephalography (MEG), and for surgery such as microsurgery, minimally invasive endoscopic surgery or stereotactic surgery, have been developed during last decades to help the surgeon task. Neurosurgery is a medical speciality which is in fast and constant evolution and in which robotics will have increasingly special importance because it allows to go beyond the surgeon's abilities and skills due to its precison. Robots in surgery are becoming more and more used in medical fields such as bone reconstruction or cardiology.

When performing the removal of a brain tumor it appears the added difficulty of distinguishing and precise localisation healthy tissue from the tumourous. It is vitally important to completely resect the whole tumourous tissue and at the same time preserve as much healthy brain tissue as possible in the surroundings of the tumour. In order to perform this minimally invasive brain tumour removal we propose in this thesis a robotic assisted method to detect and map tumor tissue from the brain surface, in combination with a previous developed multi-sensor at Leibniz Universität Hannover. In contrast to other current technologies this analysis process is intented to be mostly automatic by only requiring simple human software interaction and supervision.

The main difficulty to complete this task is that soft tissues require a perfect adjustment and control of the force, positions and moves of the robot. Otherwise severe permanent damage or even death of the patient could take place. The main priority of the proposed method is to allow a safe control and move of the robot. Therefore redundant and complex safety measures and algorithms have been elaborated and included to perform displacements and a soft contact with brain tissue surface. Approaching and moving process over the surface has been developed to compensate brain motions due to breathing and heart-beat, and to be perpendicular to the surface while maintaining a defined and constant contact force by using a force sensor integrated in the arm of the robot. Approaching is regulated by a software PID controller whose parameters have been analyzed and calibrated.

This proposed robotic method is intented to advance in the medical robotic field and present new ideas and processes which could be applied to different types of surgery where soft tissues are involved.

Contents

1 Introduction

This section introduces a complete overview of the thesis including a brief discussion of the State of the Art; context, motivation and objectives of the thesis; and a summary of the accomplished results. Firstly it is explained why this thesis has been developed, secondly which components and systems have been used and the desired objectives, then which methods and solutions have been developed and finally a summary of the most important results and conclusions. At the end of this section a description of the document structure is provided to facilitate access to the thesis contents.

1.1 State of the Art

1.1.1 Modern robotic surgery history

It is difficult to establish a starting point for modern robotic surgery history because according to different authors, the importance of the different machines inside robotic surgery history varies [1, 2, 3, 4]. Although there are different opinions, these machines are worldwide acknowledged to revolutionize some aspects of surgery:

- PUMA 560 (Programmable Universal Manipulation Arm) placed in 1985 a needle for a brain biopsy using CT guidance.
- PROBOT in 1988 was introduced to perform prostatic surgery. It is designed to allow a surgeon to specify a volume within the prostate to be cut, and perform automatically the cut without further intervention from the surgeon.
- RoboDoc from Integrated Surgical Systems allowed in 1992 to carry out a hip replacement surgery by coring out the femoral shaft and milling out precise fittings in the femur to accept a hip replacement prosthesis [5].
- ZEUS in 1998 carried out its first tubal re-anastomosis procedure and its first coronary artery bypass surgery (CABG) procedure.
- Da Vinci Surgical System was created in 1999. This robotic surgical system allows the surgeon to control three arms with tools and make precise moves by using a console, while he looks through two eye holes at a 3-D image of the procedure.

1.1.2 Discussion and future

Robotic surgery is a relatively new field inside medicine which is in constant development and evolution. Nowadays, robot-assisted surgery is getting more and more important due to the reduction in the price of surgical robots, technical improvement and innovation, and the beyond human precision it can be achieved by using them. The main advantage of robotic surgery is that it allows to perform minimally invasive surgery and precise moves with pinpoint accuracy exceeding human limitations. It also increases the success rate by reducing human errors or even eliminating them, and improve postoperative recovery because robots can perform less traumatic surgery.

Human loop control has been defined as the best combination between standalone robots and human capacities. Human receives the feedback information from the robotic system and give orders which are checked, corrected and processed by the system. Although robots can run on their own, in complex processes may be dangerous to rely only on software and sensors. The future of robotic surgery seems to reach to a very close relationship between human and machines.

1.2 Context of the thesis

This thesis has been developed in the Institute for Robotics and Process Control of the Technical University of Braunschweig in Germany during Erasmus Intership Programme. The Institute was founded in 1986 and since then it continues investigating new methods and robotic applications for both mobile and industrial robots. The Institute has also collaborated with other Robotic Institutes, companies and hospitals such as Hannover Medical School, in order to complete different lines of investigation.

The multipurpose Staübli RX90 robotic arm was the chosen robot for this thesis due to its versatility, accuracy, precision and, very important for the correct development of the process, due to a perfect placement hanging from a metallic structure allowing it to reach all required positions. It is connected to a PC via ethernet cable. Robot placement can be seen in the button-left part of the figure 1.

Figure 1: Robotics Laboratory of the University of Braunschweig

The thesis assumes that it is difficult for a surgeon when performing a brain tumor resection to difference exactly which part of the brain surface is healthy tissue and which is tomorous. It is very important to resect completely the tumor but it is also equally important to preserve as much

healthy tissue as possible in order to minimize brain damage in the surroundings of the damaged area. To improve surgeon accuracy this thesis has been proposed in combination with a previous tumor electromechanical sensor developed at Leibniz Universität Hannover [6] to elaborate an automatized robotic process which can distinguish between these two different types of tissue. This contact sensor generate an electric field to read different parameters: frequency, amplitude and phase. It measures their change to determine the nature of the tissue. Due to the impossibility of physically using the sensor in the laboratory, it has been simulated with a plastic rod bar with three different possible metallic tips, grabbed by robot hand and software for measuring.

Once tumor sensor is placed on the robot hand, it is necessary to control robot during approaching and moving over the surface. The process a priori doesn't have any information about geometry or exact position of the brain to analyze, so it has to be robust enough. It is prepared to scan any geometry inside an imaginary hemisphere and the only requirement is that the brain must be placed inside. This allows to correctly complete scan process even if the brain is not centered. In order to achieve this task a three-axis force and torque sensor located on the robot hand have been used. The available model for this thesis is JR3 100M40A force sensor [7]. Although its resolution and precision are near the practical limit to use it in the medical field, due to the use of advance methods and techniques, final results and solutions are valid and extendable to a more precise sensor. This sensor is independent to the robot system and connected via PCI-card with the PC so that the developed software must retrieve and process its information while it is controlling the robot.

Human brain is simulated with a brain gel model. It has approximately the same physical properties, size and geometry as a real human brain. The most important properties for this thesis are the mechanical, including elasticity. The model can also be inflated by squeezing a rubber bulb to manually imitate cardiac pulse.

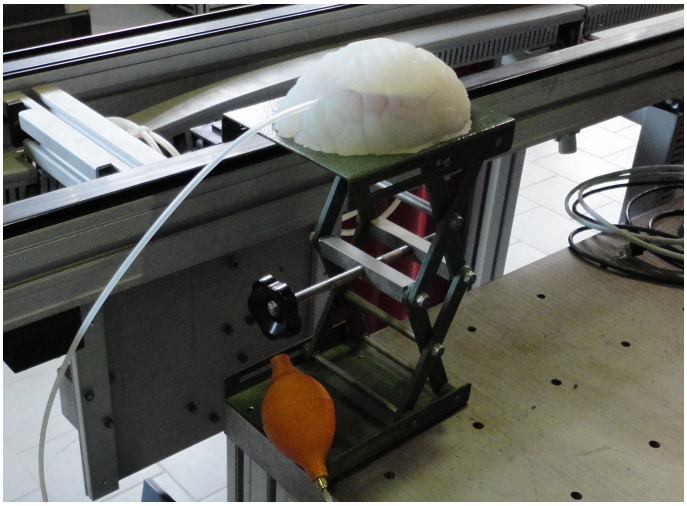


Figure 2: Brain gel model

Software has been developed under Microsoft Visual Studio 2010 environment and C++ was the chosen language in combination whith robot and force sensor libraries.

1.3 Objective of the thesis

The objective of the thesis is to develop methods and techniques which control the robot in order to place safely a tool on a live brain surface while constructing a map of the tumorous tissue. This main objective can be subdivided in small sub-objectives:

1. Move the hand of the robot over the brain surface avoiding collisions, wrong positions and singularities.
2. Softly approach to the brain surface.
3. Move over the brain surface maintaining a smooth and constant force contact while the surface is moving due to cardiac pulse.
4. Place the tumor sensor into a perpendicular position to the surface to prepare it to read a measure.
5. Store measures and create a tumor brain map allowing to visualize it.

There are also some sub-objectives that although they could not be considered as final objectives are required or are very important to accomplish of the final purpose:

1. Create a thread-safe interface for the robot and another one for the sensor including advance capacities such as buffering, statistical and test functions and safety measures.
2. Develop methods to work with spherical coordinate systems.
3. Build a user interface that controls the process and provide graphical information to the user.

1.4 Work plan

At the beginning of the thesis process it was difficult to establish a suitable work plan due to its investigative nature. The existence of a large range of possibilites to complete the main objective and the evolution of the ideas to work with, made Spiral Lifecycle Model a good approach to analyze the problems one by one, design a solution to each problem, implement it and test its results. This developing model allows a fast and constant evolution of the software and methods because it frequently raises new difficulties on each iteration related to the results obtained in the previous one.

Time distribution of the thesis is shown in figure 3. It is divided in five categories. Documentation includes seeking for information and reading books, papers and manuals. Analysis includes experiments design, risk analysis, requirements analysis and brainstorming process. In the implementation category both software design and implementation are included and also code commenting. In test part developed methods and software are probed and examined in order to evaluate their performance and obtain results as well as ellaborate conclusions which will be used in next iteration of the spiral process. It is difficult to distinguish between analysis and test part because their are very related due to the evolutionary development. Writting involves video making, documents writting, graphs and slides.

TOTAL TIME:	800 HOURS
Documentation:	75 hours
Analysis:	125 hours
Implementation:	225 hours
Tests:	225 hours
Writing:	150 hours

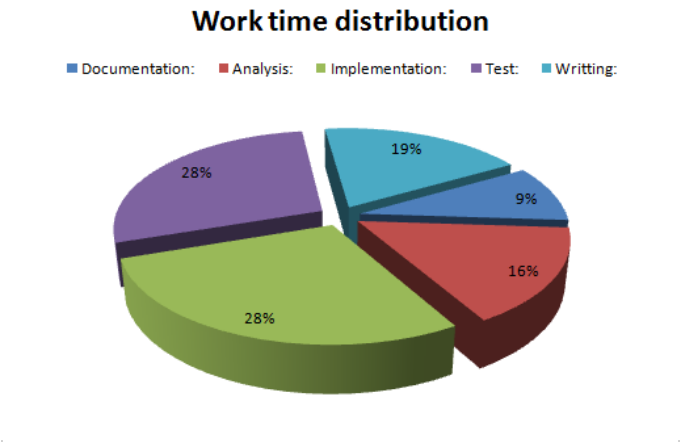


Figure 3: Work time distribution.

1.5 Solutions

Due to the large number of different investigation lines that this thesis covers and the spiral development, proposed solution evolved from a basic design to a complex group of interrelated methods. Initially, solution was divided in four distinct main parts which were developed in parallel:

1. In the first part two different new interfaces for the robot and the sensor were built. These interfaces extend robot and sensor standard capabilities including some stadistic features and buffering among other features.
2. The second part was to safe move robot hand to any desired point over the brain.
3. The third part was to approach to the brain surface, make contact maintaining a constant force and orientate the tool perpendicular to the brain surface.
4. The fourth part consisted on creating a human interface to work with the robot and results displaying and storing.

But in a further step a new part was added:

- Move the tip of the tumor sensor tool on the brain surface while maitaining a constant force.

This part required a combination of methods developed for safe location and sensor approach as well as new investigation. To accomplish this task, a new difficulty appeared which is to actively compensate gravity force while sensor orientation is being changed.

1.5.1 Robot and sensor interfaces

Robot connection and control is performed by a library which is provided by the Braunschweig Institute for Robotics and Process Control. Over this library a new thread-safe interface was developed following singleton design pattern. A unique instance of the class is created during program execution. Main properties of the new interface are:

- It implements a thread which periodically stores the robot position when running. In this thread it is possible to add safety checks such as position check or even combine it with sensor library to stop robot when force reaches a limit.
- It protects access to the robot object by using semaphores. If it is required, the library also provides direct access to the robot and software semaphores.
- It allows to easily transform hand frame when connecting to the robot, to any point in the space in order to have a new reference frame to work with.
- It simplifies robot controlling by grouping different robot functions such as move and speed setting in one. Alter mode is included in the interface. When this mode is active, robot is only moved by small translation and rotationary increments.

Force sensor manufacturer also provides a C++ library. The interface built over is a little bit more complex than robot interface. It includes these functionalities:

- It also implements a thread which periodically buffers force and torque measures.
- It allows to read force and torque directly or from buffer. The read value can also be obtained after applying statistical analysis to obtain a mean value for a given number of measures and after deleting outliers. Statistical indicators such as standard deviation can be also obtained.
- It allows to change sensor filter.
- It can calculate the force module.
- It checks if sensor reset was succesful and repeat the operation if not.
- Test sensor performance to know its precision and accuracy.
- Change sensor measurement center.

1.5.2 Safe robot movement over the brain model

Before approaching to the brain surface it is necessary to move the robot hand over the brain model. It was required to develop a method to control robot moves which avoids collisons and makes it possible to reach every point of the brain surface. It was decided that the brain position should be centered down the robot because it allows to take advantage of axial symmetry. Robot movement can be split in two parts:

1. Move in an arc: tumor tool tip is fixed to an arc over the brain and it is displaced through it.

2. Rotate the arc: by rotating robot first joint.

By combining both parts it is possible to arrive to any desired point in a hemisphere over the brain. This hemisphere is called in this document Displacement Hemisphere. It is generated by all possible rotations of the displacement arc.

Each point in the displacement arc, and thus in the Displacement Hemisphere, is restricted to have an unique associate hand reference frame, with the exception of the rotation center, where tool can have multiple orientations. This restriction allows to simplify robot movement and make it safer. These frames point its Z axis to the center of the arc and X tangent to it, as it is showed in figure 4. Hand reference was previously transformed to be placed on the tip of the tumor sensor in order to make easier to work with spatial translations and rotations.

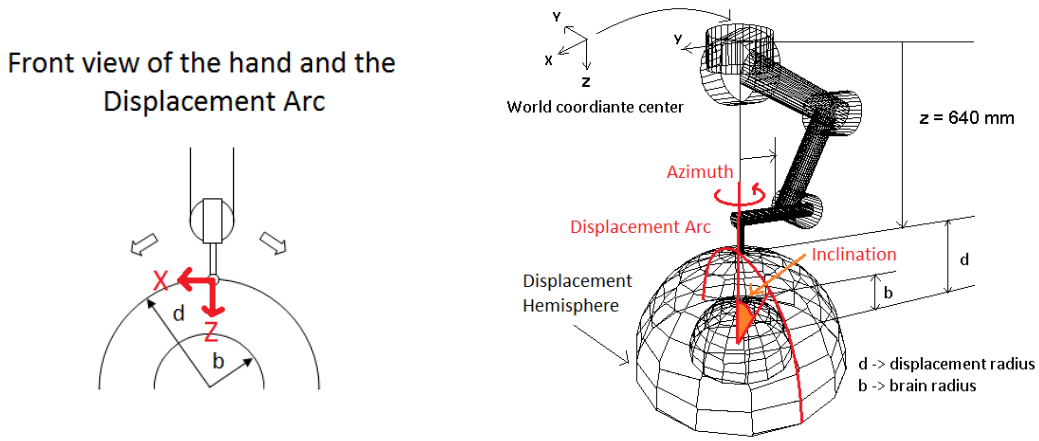


Figure 4: Displacement Arc with radius d and brain with radius b.

To move from one point on the hemisphere to another, a software library was developed which allows to work with spherical coordinate systems using inclination and azimuth. These concepts are represented in figure 5. The path between two points on the hemisphere is divided in small linear steps in which robot hand is moved using frame interpolation (figure 6). Both inclination and azimuth are modified on each step so that it is not necessary to move first to the rotation center. However, due to robot joint 1 rotation limit, some moves require to go first to the rotation center and then to the target point. It is also possible to modify radius parameter but in this movement is always fixed to 15 cm.

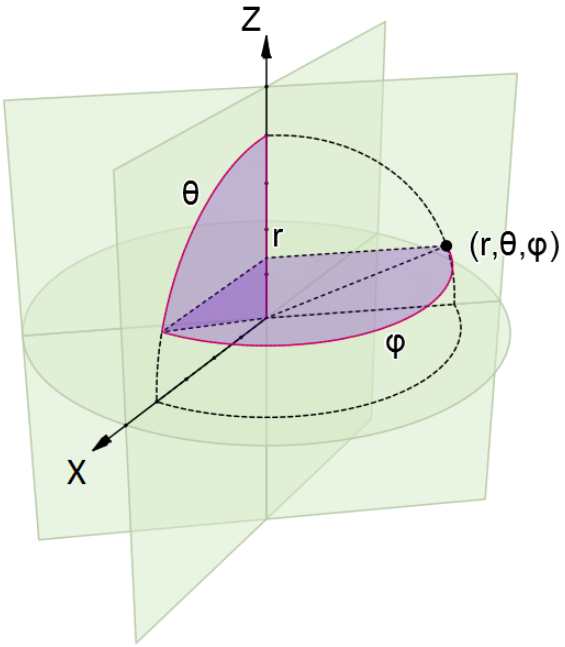


Figure 5: Spherical coordinate system with radius r, inclination θ and azimuth φ.

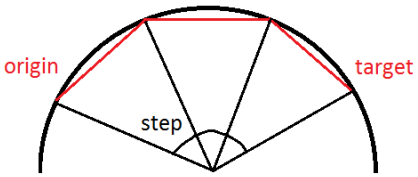


Figure 6: Linearization of the arc with small angular steps.

Movement method limitations could be further improved by using a robot with more degrees of freedom in future investigation because it would allow to surpass rotational limitations.

1.5.3 Approach to the brain surface

Tumor sensor approach is controlled by a proportional-integral-derivative (PID) software controller which was developed for this thesis. PID controller is an independent class. It is highly configurable: target force, number of measures to obtain a mean force value; proportional, integral and derivative constants or the number of iterations to calculate integral value. Approach methods are grouped

in a different class which instanciate a PID object. During approaching, the robot uses alter mode to move the hand according to the force values read by the sensor. This mode translates and rotates the robot hand through small steps. The number of steps and their increment values are also configurable. If a different order is not given, robot executes these small steps each a 16 milliseconds.

Approach is only performed in Z axis, so that force sensor orientation doesn't change. In a preliminar development, sensor was reset on each position before start to approach, so that weight vector is set as offset. In the final solution, gravity is compensated making sensor reset before starting to approach not longer necessary.

Different methods for configure PID values were tested: manual tuning, Ziegler-Nichols method and an automatic battery test. These methods allow to determine a good parameter configuration which makes possible a soft approach but keeping a good response to surface displacements.

1.5.4 Tool orientation

To orientate the tool once contact is successful, two different approaches were developed:

1. Store contact forces at the tip of the tumor tool, calculate force vector and compare with current orientation. Orientation of the tool can now be modified through a normal robot rotation or through alter mode by checking when tool arrives to the desired orientation.
2. Store contact force while it is rotating in alter mode. Hand rotates in the direction of the force vector using a mathematical quaternion until force in X and Y is stabilized.

The first possibility only needs to reset sensor at the beginning, so it was first developed. In the second, it is necessary to compensate gravity deviation. In both cases it is necessary to check for robot singularities when the point of the brain is low in the Displacement Hemisphere.

1.5.5 Moving tool on brain surface

Moving on brain surface was accomplished by combining spherical displacement and approach. Between two points on the surface a path is created as it was described in 1.5.2, but in this case, radius is actively modified from the last position on each step. Displacement in X and Y axis and rotation of the hand reference is done by small fixed increments on each step until desired position and orientation. Approach is active in Z axis during the whole movement allowing to move smoothly with a constant force.

1.5.6 Gravity compensation

Gravity compensation is the process to know how weight vector changes and anticipate to this change. When the force sensor is reset, current weight on each axis of the hand is taken as offset. After changing the spatial orientation of the hand, gravity forces measured also are modified, so that this deviation must be compensated. Different methods were developed and tested for this task. First sensor was reset on each position, but this method disallow to move the tool over the

brain surface. It was necessary to develop advance methods based on a previous calibration step. Measured forces from different orientations are stored in this stage. This data is treated and used in the compensation stage. Several methods were developed: nearest position, spline interpolation, mathematical computation and fitting polynomic curves. They are explained in section 3.4.

1.6 Main conclusions

Main conclusion is that it is possible to use a combination of force sensor and a robotic arm like the used one for this thesis in minimally invasive surgery. A robot with similar characteristics to the Stäubli RX90 can respond fast and smooth enough for this kind of medical applications. Nevertheless, JR3 100M40A force sensor resolution is on the limit to be used for this purpose. Its resolution is ± 0.2 Newton but despite the possibility to increase the prisicion through taking more measures, gravity compensation and joint alignment errors makes diffult to work with forces under 0.5 Newtons. It is also important to note that in this force range, a slighty incorrect joint alignment affects the force measurement as it is explained in section 3.6.

1.7 Document structure

This document presents the thesis development and results in a clear and concise form in order to arrive to the conclussions and discussion of the results. The first part of the document introduces into a complete overview of the whole thesis including its context. In the second part all concepts which are important for the understanding of the work are detailed. In the third part methods which have been developed to perform the brain analysis tasks are described. In the fourth section software architecture and system design is introduced. Fifth part includes the results as well as their discussion. Finally in the last part conclussions are gathered together in order to give a final vision of the work and future lines of investigations. References and appendices are included in the end of the document.

2 Concepts

This section describes the different elements of the system, their functionalities and properties, and the ideas which are necessary to be known in order to understand developed work, methods and implementation.

2.1 Environment and components description

Figure 7 introduces a view of the work environment in the laboratory. The robot hangs upside down from a structure and it has its own dedicated space over a metallic table. The control computer is on a desk next to the table allowing to watch the robot operations. A gel model of a human brain is placed on the table by using an adjustable support to regulate vertical position. This configuration imitates a possible operating theatre distribution where patient is seated directly under the robot. The robot hand attaches a force sensor and grips a plastic rod bar which simulates the tumor sensor. All these elements are sketched in figure 8 and detailed described in subsequent subsections.

Figure 7: Work environment.

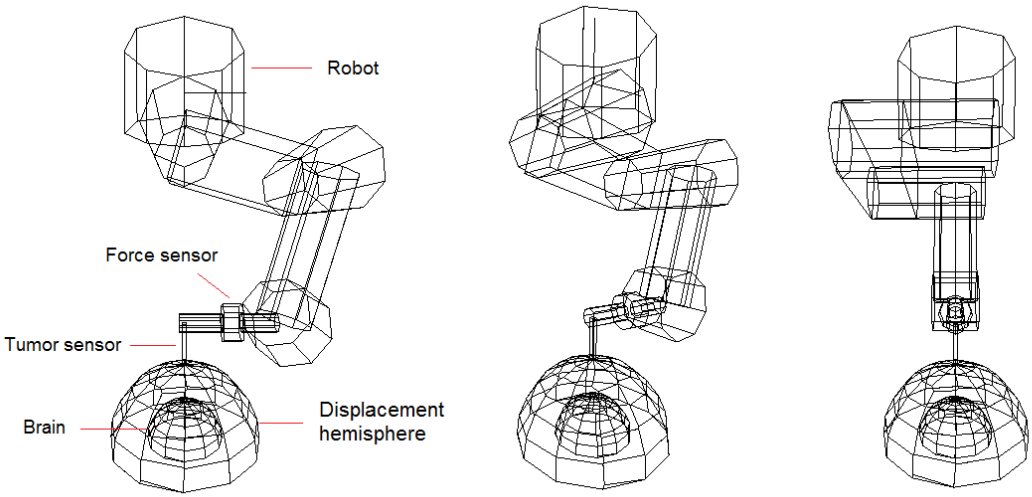


Figure 8: 3D side, oblique and frontal view of the robot and brain positions.

2.1.1 Robot

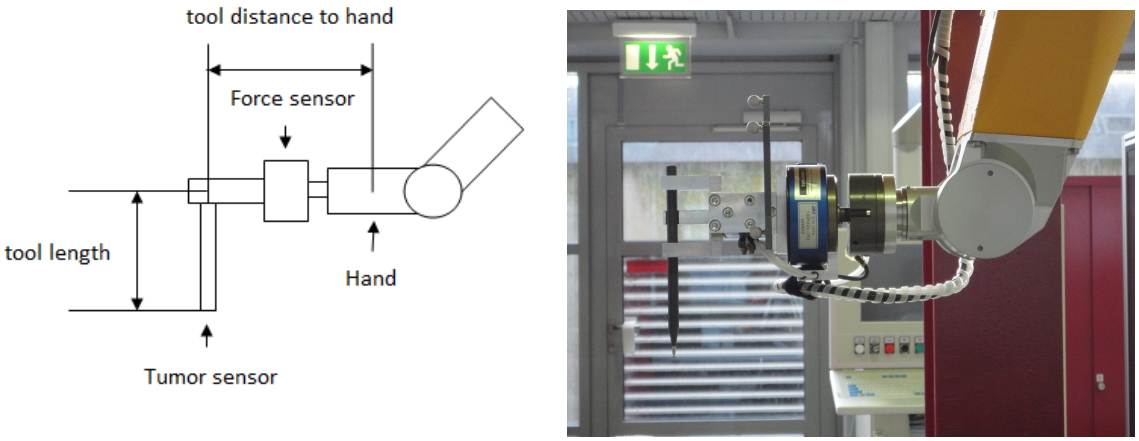
The robot which was used for this thesis is a multipurpose industrial robotic arm model Stäubli RX90. It is powered by a control unit with its own terminal computer. The control unit executes robot moves and checks wrong positions, singularities and other different types of errors. When a problem is detected, control stops the robot and disconnect power. Robot orders can be directly given by a control panel or by a PC to which it is connected via LAN, allowing to execute complex programs.

2.1.2 Robot hand

Robot hand is equipped with a force and torque sensor attached between the end of the last joint and the claw. Claw and force sensor form part of the same system and they are moved together. The claw grips the simulated tumor sensor as it is shown in figure 9. Hand center to tool center is the distance between the center of the original hand coordinate system of the robot and the center of the rod bar wich is used to simulate the tumor sensor. Tool length is the distance between the tip of the rod bar and the center of the claw. These distances are detailed in table 1. Robot hand is provided with a pneumatic safety system to prevent robot of collision damages. The pneumatic system holds the entire hand and in case of a high force release the hand disconnecting robot power as well.

Distance:	mm
From original center to tool center	220
Tool length	140

Table 1: Hand distances



(a) Robot hand diagram.

(b) Photo of the hand.

Figure 9: Side view of the hand of the robot. Force sensor and simulated tumor sensor are also visible.

As it is described in section 2.3.2, original hand frame is transformed when connection between robot and PC is performed and deleted when disconneting. After this transformation the point of the tool becomes the new hand coordinate system center, which makes it easier to implement rotations and a more legible software. It is important to underline that if this transformation is not done at the beginning of the programm execution, the previous transformation is still effective.

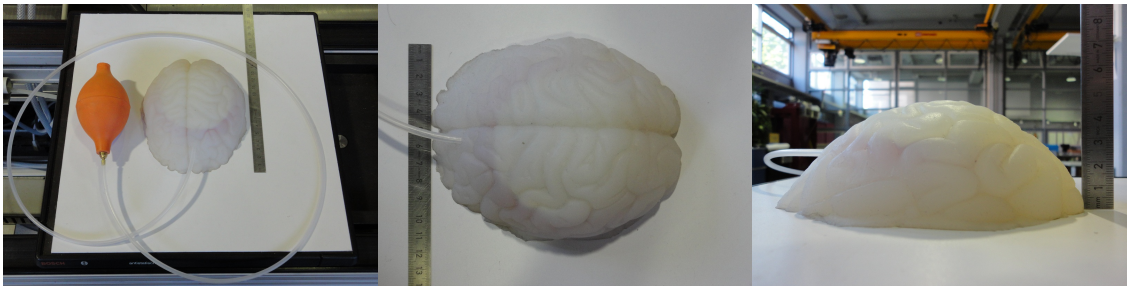
2.1.3 Force sensor

The robot hand attaches a force and torque sensor model JR3 100M40A [7]. It is capable to measure force and torque at a given point in the space in three-axis. Measurement center can be changed to any position through sensor software library modifying read results. Sensor operation details are explained in section 2.2. Figure 9 includes a side view of the sensor.

2.1.4 Brain model

Brain model has the same size and geometry as the top part of a real human brain. It is made of gel which imitates real physical properties such as elasticity. Inside the gel there is a metallic skeleton which protects its integrity in case of high out of range forces. There is also a balloon inside which can be inflated from a rubber bulb. It is deflated automatically by loosing its pressure in a few seconds, depending on the initial pressure. Figure 10 shows three different views of the brain model.

For testing purposes, there is also a smaller gel piece with the same properties as the brain model. Figure 11 includes a photograph of this piece.

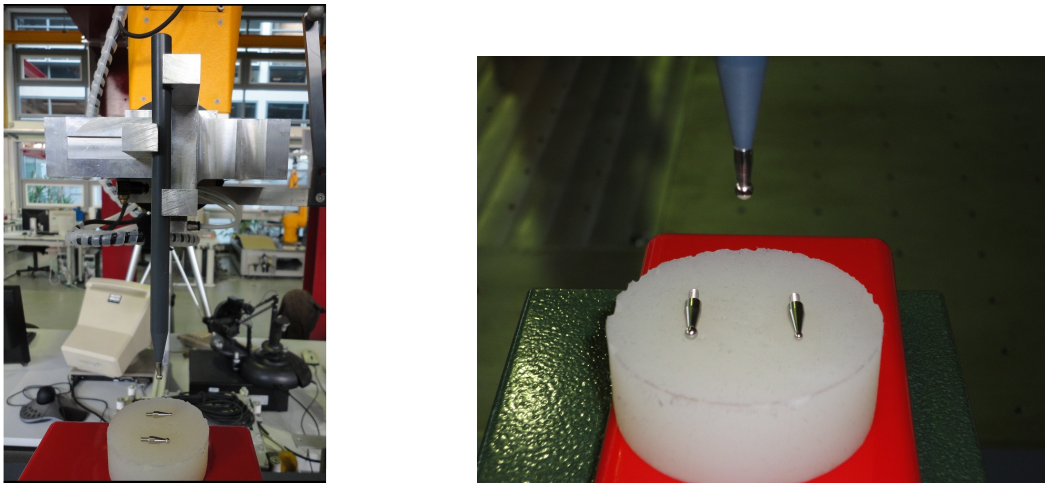


(a) 14 cm long. (b) 11 cm wide. (c) 4.5 cm height

Figure 10: Brain model views and size.

2.1.5 Simulated tumor sensor

Tumor sensor is simulated with a metallic point attached to a plastic rod bar. The bar is gripped by the robot hand so that its point is 14 cm under the hand center. There are 3 different interchangeable spherical metallic points with different diameter: 2 mm, 3 mm and 4 mm.



(a) Tumor sensor. (b) Metallic points on the gel piece.

Figure 11: Simulated tumor sensor in detail.

2.1.6 Displacement Hemisphere

Displacement Hemisphere is an imaginary hemispherical surface which is used as configuration space of the tumor sensor tip for location moves. It is visible in figure 8. This configuration allows to safely move the robot hand avoiding collisions when positioning the tip over the brain. The tip is always moved on this surface, perpendicular to the center and pointing to it. Hemisphere radius is set up to 15 cm and its center must coincide with the brain center as it is explained in section 2.4. All the brain volume must be inside the Hemisphere otherwise the robot may crash during operation.

Displacement hemisphere is generated by all possible rotations of one arc called Displacement Arc as it is described in figure 12. Robot moving algorithm over the surface is explained in detail in section 3.2.

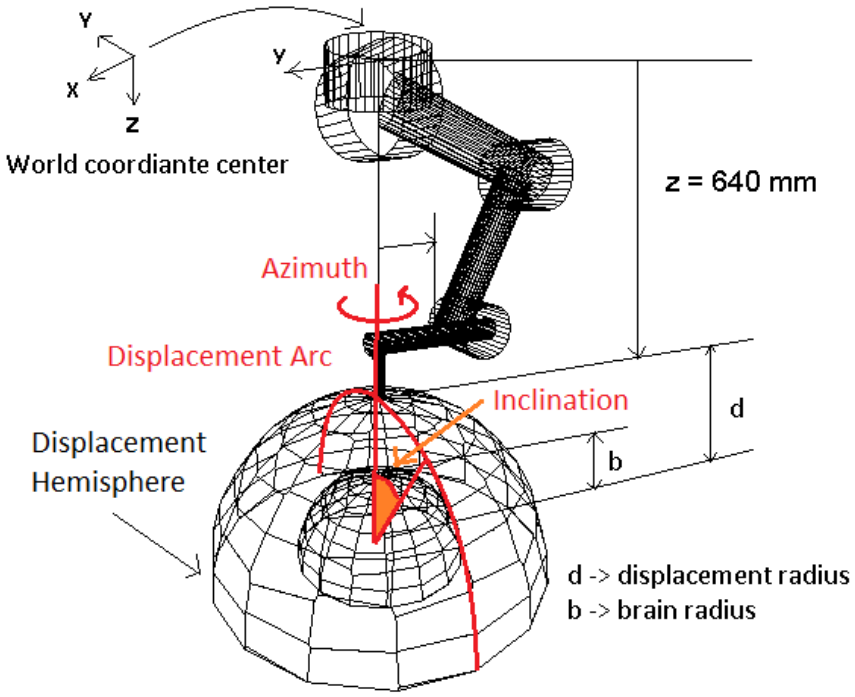


Figure 12: Displacement Hemisphere and one Displacement Arc.

2.2 Force/torque sensor operation details

In this section the most important concepts, operations and functions related to the force/torque sensor are detailed.

2.2.1 Measurement center transformation

Initially, the measurement center is placed inside the force sensor. As it is explained in figure 13, sensor center is originally located on an imaginary line which crosses perpendicular the center of the sensor. It must be rotated and translated from this original position to the point of the tumor sensor. Tool length is previously known but sensor original center to tool distance and first Z rotation had to be confirmed through a method which is explained in section 3.1.1. Results are included in table 2.

Name	Value
First Z rotation	approximately 45 degrees
Sensor to tool	12.5 cm
Tool length	14 cm

Table 2: Values of the distances required to fit sensor measurement center.

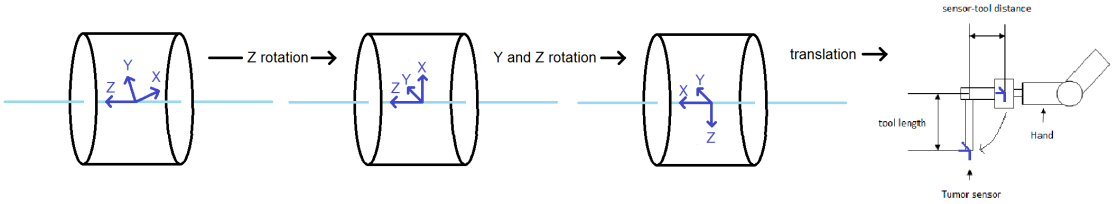


Figure 13: Transformation of the measurement center from original to required pose.

2.2.2 Sensor reset

Before starting to take measures it is required to reset the force sensor. By this operation current force and torque values are stored internally as offset by the sensor software library. From now on, offset values are substracted from new measured values. This operation is transparent to the measurement process. After reseting it is necessary to wait some hundred milliseconds and check if measured forces and torques are null because sometimes reset doesn't succeed making sensor to measure wrong values. Complete reset process last few seconds. To obtain precise values, robot hand during reset process must not contact with any object or be moved.

2.2.3 Sensor filters

Sensor library can work with 7 different filters. Filter 0 indicates that no filter is used. As long as filter number increases, more measures are taken into account to give the final result. This increases precision but entails a loss of speed of response. As it will be concluded, filters higher than 4 cannot be applied for this thesis purpose and recommend is 4.

2.2.4 Force and torque measuring

Force and torque regarding to the transformed sensor coordinate center are measured after applying the selected filter. Forces are measured in Newtons and torques in Newton·meter. It is possible to change units to English metric system. In this thesis only the International Metric System has been used.

2.2.5 Gravity compensation

Force/torque sensor takes as offset the current applied force to the hand of the robot. If nothing is grabbed or contacted, its own weight is taken as offset when resetting the sensor. Gravity compensantion problem appears when changing hand orientation after resetting, because its weight vector projections onto X, Y and Z axis are modified while offset remains constant. In order to compensate this effect, different solutions have been proposed in section 3.4.

2.3 Coordinate systems description

During analysis, design and software implementation, different coordinate systems have been used and adopted, either because they are required by hardware or because they give some advantages in order to simplify and make safer algorithms and processes. The most important coordinate systems are graphically described indicating their center position and orientation in figure 14. A good coordinate system comprehension is important before reading the implemented source code of the application. All different systems are represented from the same point of view.

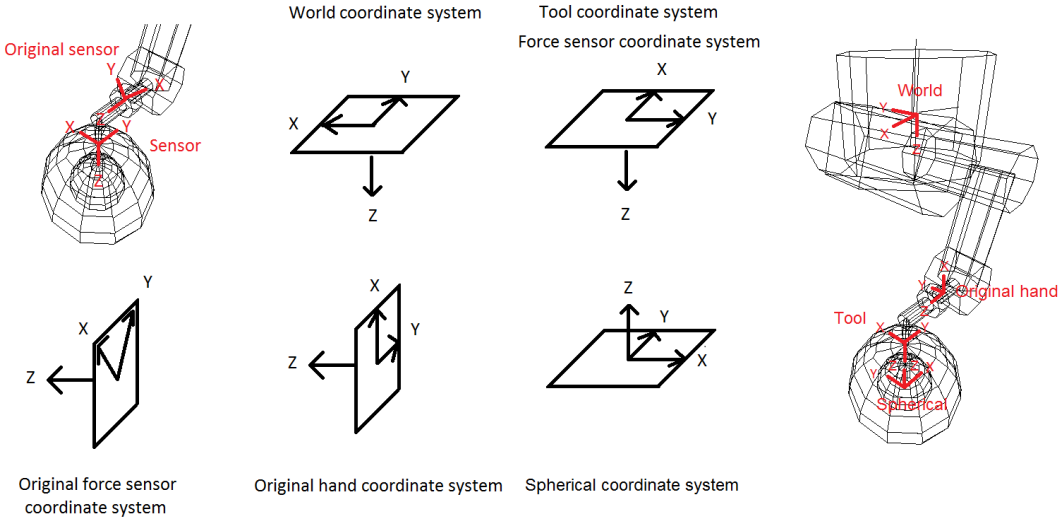


Figure 14: Different coordinate systems viewed from the same point of view.

2.3.1 World Coordinate System

This coordinate system represents the world coordinate system of the robot. It is placed inside the robot base and defined by hardware configuration.

2.3.2 Tool Coordinate Systems

Tool coordinate system can be modified by software in order to get a more appropiate center position and orientation for a given task. In this thesis it is important to find a center position

which simplifies programming rotations and translations of the tumor sensor. It was determined that hand sytem centered at the point of the tumor sensor bar simplifies and make safer rotational and collision detection algorithms. In this thesis two different tool coordinate are referred: original and transformed.

Original system This is the default coordinate system of the robot hand after switching on the robot. It is placed between the last robot joint and the force sensor. Its usage was avoided in this thesis by transforming it into a new coordinate system on the point of the tool.

Transformed system. This is the coordinate system placed at the point of the tumor sensor. Z axis is vertically aligned with the rod bar.

To transform from the original to the new tool system, Denavit-Hartenberg (D-H) conventions were used as it is represented in figure 15 and described below. It is not necessary to follow these conventions because it is simply a homogeneous transformation, but it was considered to be an elegant approach. To apply D-H conventions, tumor rod bar is taken as a joint:

1. Z-axis points along axis of rotation. In this case along the tool length.
2. X-axis is parallel to the common normal ($x_n = z_n \times z_{n-1}$) but translated in Z instead of taking its origin at intersection between old z and new one.
3. Y-axis is constrained to complete a right handed coordinate system.

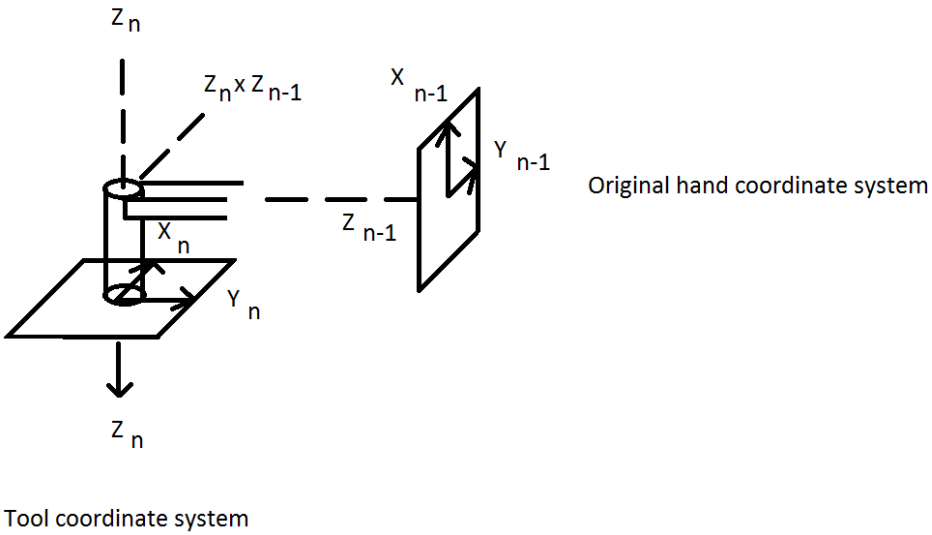


Figure 15: Hand and tool coordinate systems in detail.

Frame transformation through two translations and two rotations:

1. Translate in hand Z axis the distance between hand and tool (220 mm).

2. Translate in hand X axis tool length distance (140 mm).

3. Rotate $\frac{-\pi}{2}$in Y axis.

4. Rotate $\frac{\pi}{2}$in Z axis.

2.3.3 Force sensor coordinate systems

Original force sensor coordinate system is placed inside the sensor and transformed at the point of the tumor sensor in a very similar waz as the transformed hand tool system. The transformation obtaining process from original system is explaiend in section 2.2.1.

2.3.4 Spherical Coordinate System

Spherical coordinate system is used in the brain analysis process to facilitate curve robot moves, algorithm design and increase safety. Its origin coincide with brain center as it is explained in 18. A spherical coordinate system has three parameters: inclination, azimuth and radius. They are represented in figure 16. It is possible to use elevation instead of inclination because they are complementary angles. Displacement Hemisphere is described by using this spherical coordinate system as it is indicated in the figure 2.1.6.

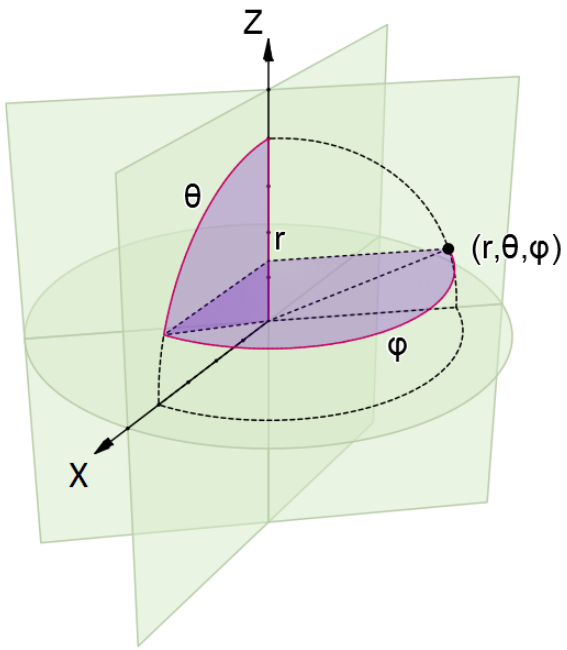


Figure 16: Spherical coordinate system with radius r, inclination θ and azimuth φ.

2.4 Brain positioning and brain center estimation

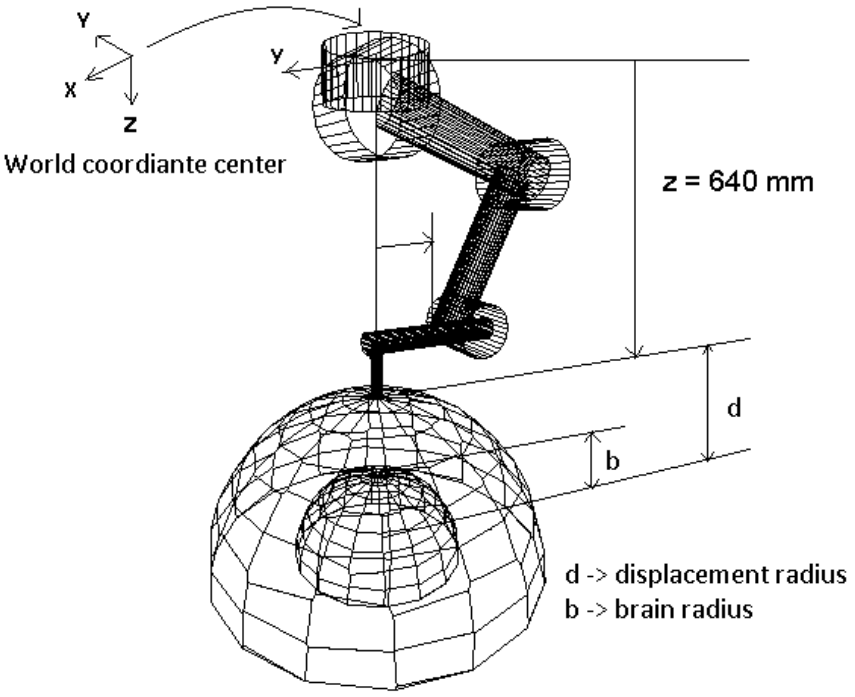


Figure 17: Diagram of the position of the hand and brain in the world coordinate system.

This section describes how the brain model must be placed under the robot. Advisable position of the model is represented in figure 17, but due to the flexible developed scanning algorithm, the only operational requirement is that the model is completely inside the Displacement Hemisphere in order to avoid collisions between brain and the robot hand. Displacement Hemisphere concept is explained in section 2.1.6. This layout allows to reach all parts of the brain surface avoiding singularities and robot wrong positions. Other brain positions and robot move patterns were tested with worse results due to singularities.

When robot is disconnected its arm hangs vertically extended and it is necessary to move it to start position before placing the brain model, and also take it out before moving the robot to end position.

Although it is not required to center the model, it is recommended to align its center on world space. Developed methods and software assume that brain center is located in (0 mm,0 mm,790 mm) world coordinates. It is aligned with world coordinate center but 640 mm plus the Displacement Hemisphere radius (150 mm) down in the world Z direction. Displacement Hemisphere top center is 640 mm under the robot world coordinate center.

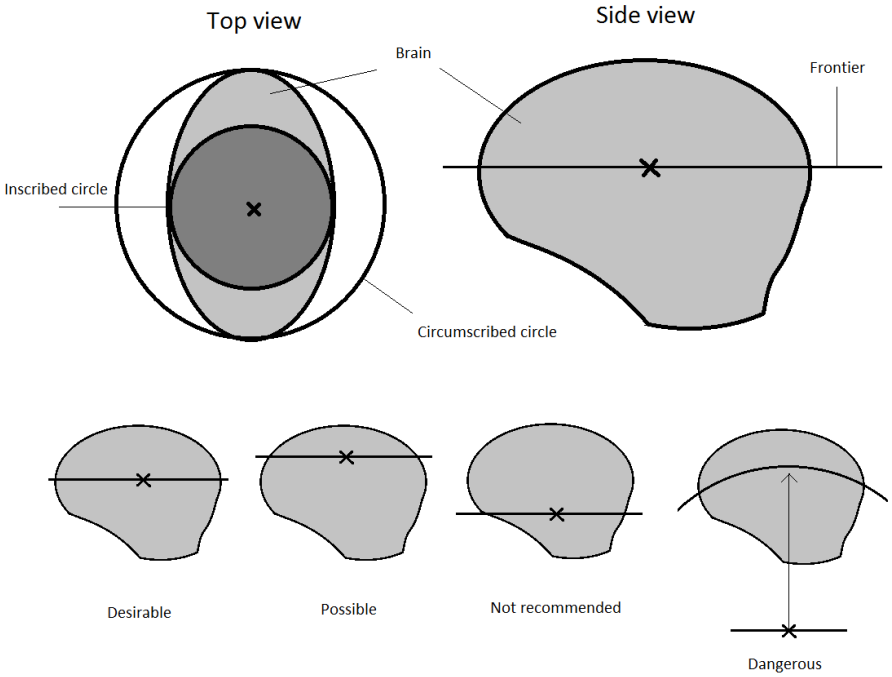


Figure 18: Brain top and side view.

Due to the difficulty to find the real geometric center of the brain, an approximation can be considered. Figure 18 shows how to approximate the geometric center by visual inspection. Brain center coincide with the center of the inscribed and circumscribed circles on X-Y plane supposing an elliptical shape. If brain shape were irregular, a middle point between both centers could be considered. Z-axis center value can be freely chosen depending on the lowest part we want to analyze as long as all the brain is inside the Displacement Hemisphere.

The frontier is the botton part of the Displacement Hemisphere. Nothing under this surface is scanned. Brain surface parts whose perpendicular vector points below this frontier are not guaranteed to be correctly scanned because perpendicular orientation will not be reached by the hand in order to avoid singularities of the robot joints. This limitation is represented in figure 19.

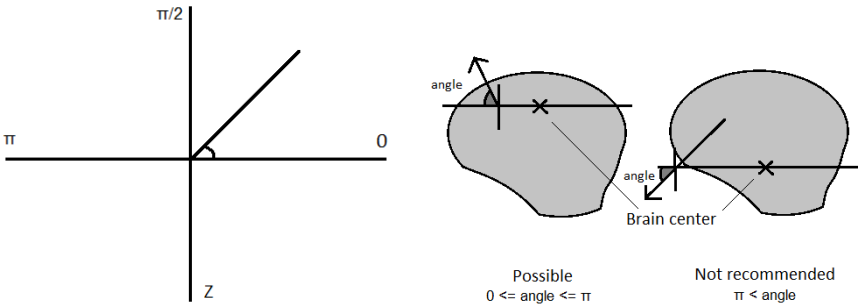


Figure 19: Cartesian coordinate system representing angle limitations.

2.5 Quadrants

World plane X-Y is divided in four quadrants. From a 2D top view, spherical coordinate center placed in the brain center is the same as world coordinate center as it is shown in figure 20. Quadrants facilitate movement algorithm design. Depending on in which quadrant the robot tool is situated, a different orientation of the hand is required due to the limitation of the robot joint 1 turning as it is described in section 3.2.1. Each quadrant corresponds with the following spherical azimuth ranges:

- Quadrant 1: azimuth $[0, \frac{\pi}{2})$
- Quadrant 2: azimuth $[\frac{\pi}{2}, \pi)$
- Quadrant 3: azimuth $[\pi, \frac{3\pi}{2})$
- Quadrant 4: azimuth $[\frac{3\pi}{2}, 2\pi)$

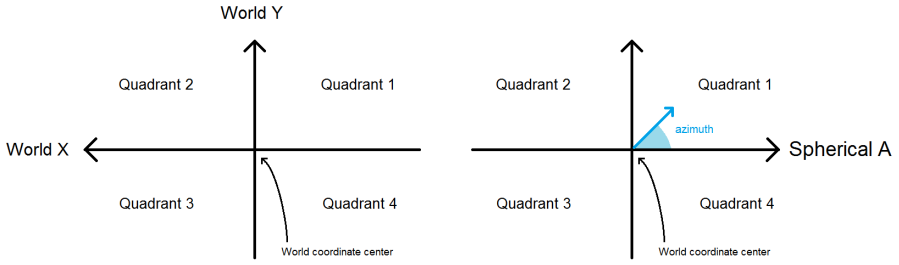


Figure 20: Top view of the quadrant system showing arm starting position and quadrant distribution.

2.6 Robot initialization and stop

Before starting brain analysis process robot must be placed in the start position. Robot will not check if any obstacle is inside the path, so this make this move dangerous. The same happens after finishing the brain analysis task when robot is placed in the rest position. These moves must only be executed if the pacient and any other obstacles are out of the way between the start and stop positions, and if possible take out everything inside the working area. The robot is moved using Place command. Moves are performed following joint interpolation so that the path will be different from a straight line between positions.

Joint	1	2	3	4	5	6
Start (work position) joint values in degrees:	0.0	-170.78	188.80	0.0	71.98	0.0
End (rest position) joint values in degrees:	0.0	-90.0	90.0	0.0	0.0	0.0

3 Methods

This section describes algorithms and processes which have been created or used for this thesis to acomplish the thesis main objectives and related topics.

3.1 Related to sensor

3.1.1 Sensor center positioning values calculation

This method was developed to determine required first Z rotation and distance from sensor original center to tool to obtain a measurement center on the point of the tool as it is explained in section 2.2.1. The method consist of testing different possible rotational and translational values to experimetally obtain which ones follow the predicted behaviour. In this method sensor center is not translated vertically using tool length. It is only translated in the original Z axis of the sensor.

Robot hand is initially placed vertically perpendicular to a piece of gel. For each different parameter to be tested, after transforming robot hand goes down a fixed distance and contacts with the surface. The smallest metallic point was used in order to improve accuracy and the piece of gel was used instead of the brain model due to high forces involved in this process when moving a fixed distance. Then forces and torques are measured and stored. Algorithm pseudocode is described below and sketched in figure 21.

Algorithm 1 Sensor center transform parameters test.

1. For each distance or angle
   (a) Transform sensor center
   (b) Reset sensor
   (c) Move tool down inside a gel piece
   (d) Read and store measures
   (e) Move hand up

Values are calculated in two steps. First Z rotation angle is tested and then the translational value.

Rotation angle calculation In this method, the tool contacts vertically with gel surface. If measurement center is correctly orientated, the highest negative force is measured in Z axis and null in X and Y. If not, X value increases as Z decreases. A correct translation distance is not required in this part to obtain the best fit.

Translation distance calculation To calculate the distance between original force sensor center and tool, translation distance is modified through an iterative process. Testing step and range can be decreased through iterations to find faster the correct distance. If measurement center orientation was not previously calculated, it is necessary to use the least root mean square of the

three torques. If it is orientated with two axes perpendicular to the translation line, it is only necessary to find which distance cause a null torque in axis Y.

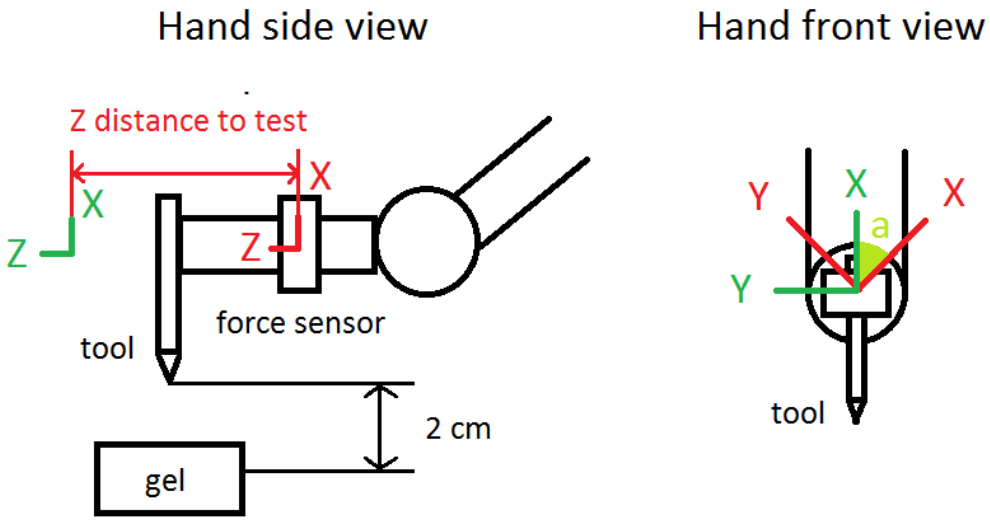


Figure 21: Graphic description of both methods. In left part the initial Z translation distance and in the right part the initial Z rotation angle a are indicated. Original coordinate system is represented in red. The transformated system is showed in red.

3.1.2 Sensor performance determination

There are three different properties of the sensor which are important to determine. Despite resolution of the sensor is known by the manufacturer, precision, accuracy and speed response are unknown.

Sensor precision is calculated in two cases. In the first case the hand keeps a static position, and in the second it is moved during the test. It is important to observe the sensor performance in both cases and also how the different filters modify this sensor performance. In a static position, a large number of measurements are taken to calculate the mean force and the standard deviation for each filter. Dynamic precision is evaluated in the same way as in the static position, but now it is also interesting to visualize changes in the forces through time. In both cases, the robot tool doesn't contact with any object.

Sensor speed response is evaluated by moving the robot tool 1 cm up and down and visualizing force evolution. As it is represented in figure 22, now a piece of gel is placed in order to cause contact.

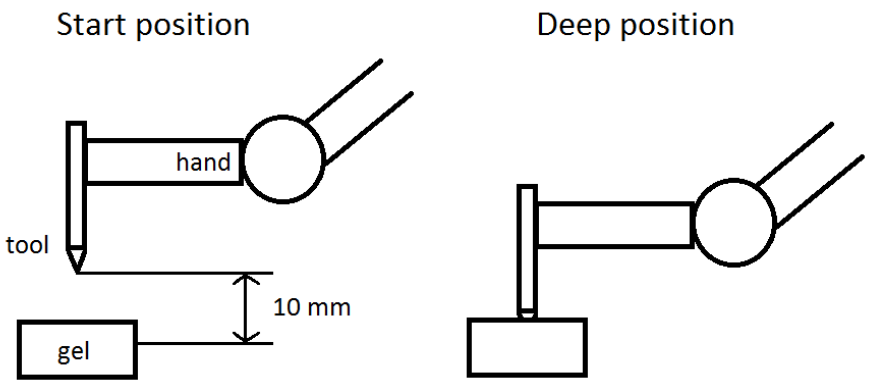


Figure 22: Schema of the start and deep position. Hand moves between these two positions.

It is also interesting to check if sensor accuracy depends on sensor reset. The proposed experiment to check this idea is to reset the sensor a given number of times calculating for each reset the mean value of a large number of measures. The standard deviation calculated from these means indicates the sensor resest accuracy.

3.2 Safe robot movement over the brain model

In this section it is explained in detail how the robot hand is positioned over the brain model. As it is described in section 2.1.6, the robot hand is moved on the Displacement Hemisphere from one point over the brain model to another. Hemisphere radius is 15 cm, nonetheless it is possible to change it. Hemisphere is generated by all possible rotations of an arc called Displacement Arc.

Spherical coordinates were used to represent a point of the Displacement Hemisphere, instead of using a Cartesian system. Spherical coordinates are described with three parameters: inclination, azimuth and radius, as it is explained in 2.3.4. Spherical coordinates facilitate to work with spherical movement. Movement over the brain model is composed by two different parts which can be combine at the same time:

1. Movement in the Displacement Arc: inclination is changed while radius and azimuth are constant.

2. Rotation of the Displacemente Arc: azimuth of the position is changed while radius and inclination are constant.

These parts are represented in figure 23 and detailed in the following sections. It is possible to change azimuth and inclination at the same time to build a path between any desired positions over the Displacement Hemisphere by combining both types of movement.

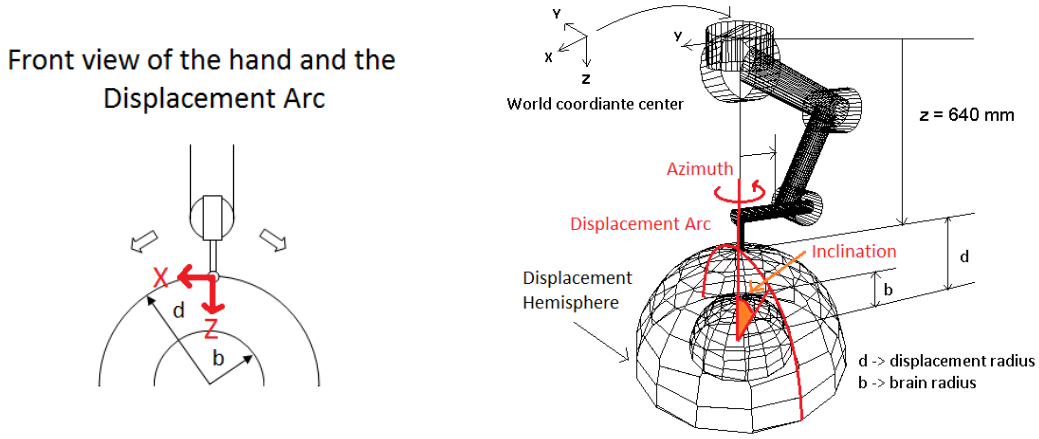


Figure 23: Left image shows a front view of the robot hand and the Displacement Arc with radius *d*. The right image shows how Displacement Arc can be rotated to generate the Displacement Hesmisphere with radius *d*. Brain radius is *b*.

3.2.1 Displacement Arc and robot limits

The Displacement Arc is always perpendicular to the robot hand and the world X-Y plane. To change inclination, the robot hand moves laterally on the Arc pointing to its center, as it is described in figure 24. To perform this displacement, robot joint 1 is rotated less than 45º from the current position to avoid reaching its rotational limits. It is important to emphasize that in low positions, joint 3 is near a singularity because its two attached links get almost parallel. It was tested that all positions of the Displacemente Arc are rechable with this method, but if orientation of the hand is changed in low points, joint 3 must be monitorized in order to stop motions before limit is reached. Otherwise, hardware will disconnect the robot power.

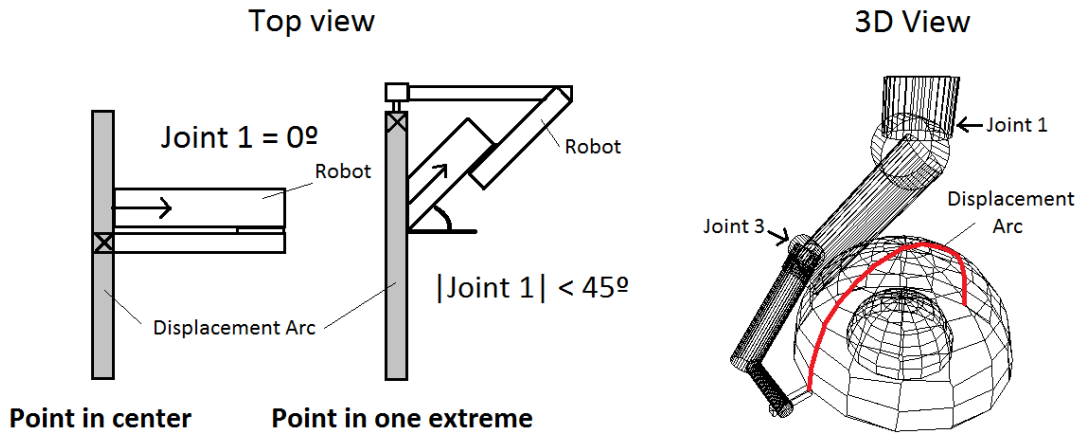


Figure 24: With a limit of 45º on each direction all points of the Displacement Arc are reachable.

3.2.2 Rotation and robot limits

If the joint 1 of the robot is rotated while keeping fixed the other joints, the current Displacement Arc also rotates its azimuth. The physical rotation range of joint 1 is [-160º,160º]. It is possible to generate the Displacement Hemisphere by rotating joint 1 [-135º, 135º). It is required a [90º,90º) rotation to arrive to all the top points of the Hemisphere, but an additional 45º rotation to go to the lower points such as was explained in 3.2.1. Figure 25 shows joint 1 limits.

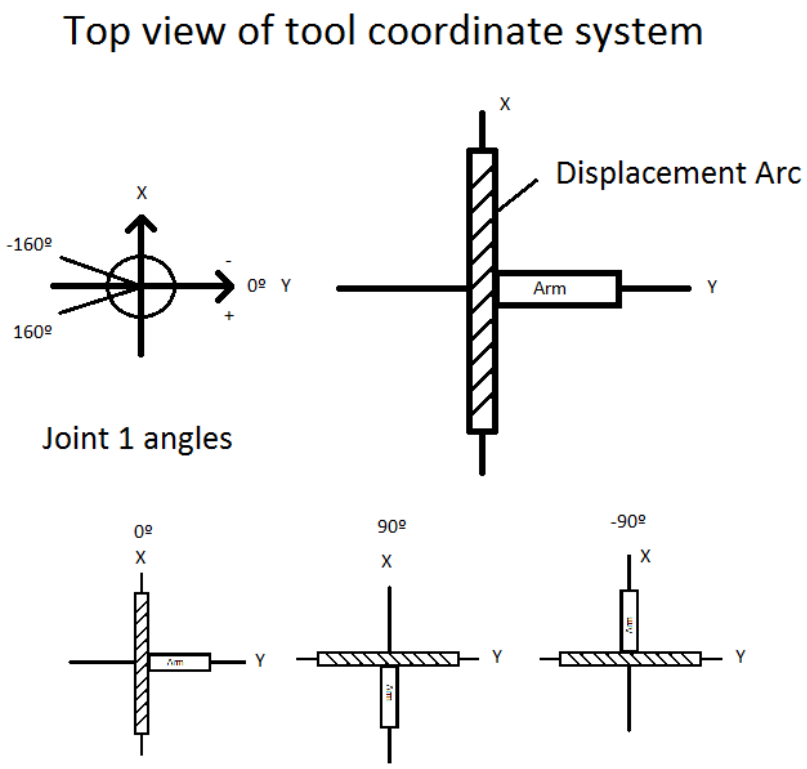


Figure 25: Top view of the Displacement Arc and rotational limits. Coordinate system in these figures is the same as the tool coordinate system in the starting position.

3.2.3 Frames calculation

Each point of the Displacement Hemisphere has an unique associated frame because rotation in Z axis is always fixed, with the exception of points with null inclination. They are located on the top part of the Hemisphere, where multiple orientations are possible for this point.

Before calculating the frame, it is necessary to know in which quadrant desired point will be located through quadrant classification detailed in section 2.5. Depending on the quadrant in which the point is located, robot frame is orientated in one way or in the opposite.

Spherical coordinate center for Displacement Hemisphere is indicated in section 2.3. The proccess to obtain the corresponding robot frame for a given radius, azimuth and inclination; has the following steps:

First X, Y, Z translation components are calculated using basic trigonometry:

- $x = radius \cdot sin(inclination) \cdot cos(azimuth)$
- $y = radius \cdot sin(inclination) \cdot sin(azimuth)$
- $z = radius \cdot cos(inclination)$

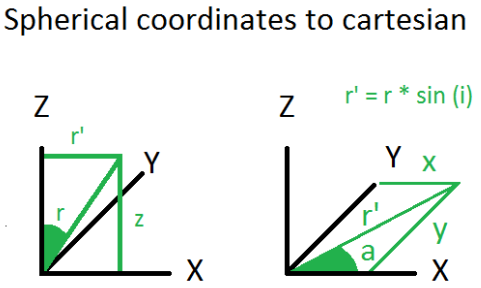


Figure 26: Radius is r, azimuth is a, and inclination is i.

Second, the coordinate system is rotated to a intermediate system which has Z axis pointing perpendicular outside the Hemisphere and Y tangent to the Displacement Arc. If position is in quadrant 1 or 2:

- Rotation in Z: $azimuth - \frac{\pi}{2}$
- Rotation in X: $-inclination$

If position is in quadrant 3 or 4:

- Rotation in Z: $azimuth - \frac{3\pi}{2}$
- Rotation in X: $inclination$

Finally, this intermediate system is transformed to the tool coordinate system by rotating π in X and then $-\frac{\pi}{2}$ in Z. It is necessary to rotate in Z axis in order to prevent the links of the robot to collide with the brain. The whole rotation process is described in figure 27

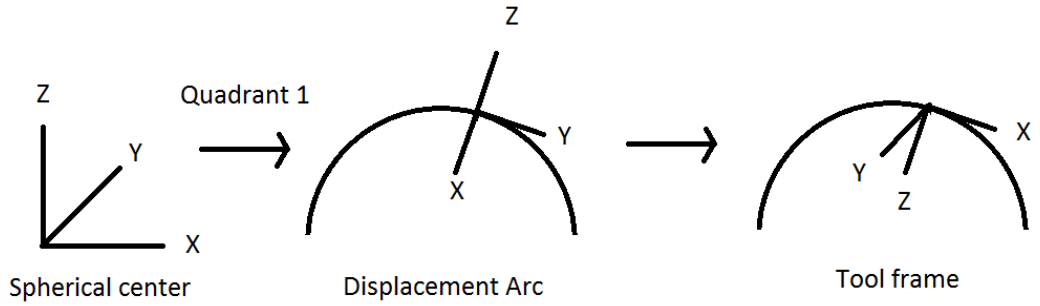


Figure 27: Rotation steps to obtain tool frame from the spherical center.

3.2.4 Path on the Displacement Hemisphere

Once it is possible to know which frame corresponds to a point on the Hemisphere, a spherical path between two points on it can be built. The path is created by using a number of intermediate points, which are calculated by dividing both azimuth and inclination in steps. For each intermediate spherical coordinates a frame is obtained. If target and origin are at the same time in quadrants 1-2 or in 3-4, a direct movement between them can be performed. If not, it is necessary to move first to the Hemisphere top center and then to the target, because orientation of the hand between positions is rotated 180 degrees, which means robot must rotate the arm. The safest point to do this is on the top center. Otherwise the robot arm could crash with the brain model.

Spherical path is built by using a big number of linear steps as it is described in figure 28. On each step, the robot hand moves using joint interpolation movement mode. Linear step distance can be changed but by default is 2 cm. Intermediate frames are calculated by obtaining the mean azimuth and inclination between origin and target frames. Here it is presented the description of the recursive algorithm which calculates the path:

Algorithm 2 Path on the Displacement Hemisphere building.

- If origin and target are both in quadrants 1-2 or in 3-4:
  - If distance(origin, intermediate) $<$ 2 cm: MoveHand(origin, intermediate)
  - If distance(origin, intermediate) $>$ 2 cm:
    - Create intermediate frame between origin and target
    - Path(origin, intermediate)
    - Path(intermediate, target)
- If origin and target are not both in quadrants 1-2 or in 3-4:
  - Path(origin, center)
  - Path(center, target)

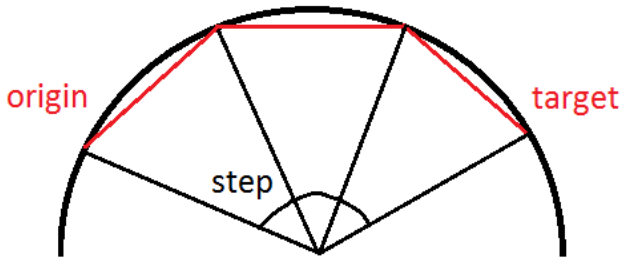


Figure 28: Front view of the Displacement Arc.

3.3 Surface contact

This section describes those methods in which the tumor tool contacts with the brain model surface. Before approaching to the surface, the tumor tool must be placed over the desired contact point, pointing to it, by using methods from section 3.2. Approach process allows to make a soft contact between the tool and the surface by moving the tool in its Z direction. Approach process also keeps a constant force with the surface even if the surface moves due to heart pulse. There are two different approaches to scan the surface: to move back the tool to the Displacement Hemisphere or to displace the tool on the surface while maintaining a constant force contact. Once tumor sensor point is stabilized, it can be orientated to reach perpendicular position to the surface.

3.3.1 Approach to the brain surface

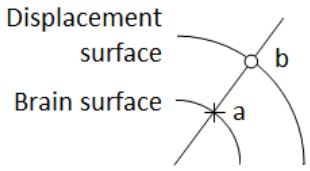


Figure 29: Approach from point b in the Displacement Hemisphere to point a on brain surface.

By this process the tumor sensor tool is moved on its Z axis going down if the desired force have not been reached or up if it is exceeded. The process is controlled by the Proportional-Integral-Derivative (PID) control class, which is detailed in section 3.3.4. To achieve this purpuse, robot works in Alter Mode in order to allow to read force from the sensor while the tool is being moved. This mode was included in the library provided by the Institute. In this mode increments of a small step distance are performed asynchronously. Alter executes each 16 milliseconds this increment value a given number of times if not a different order is received. Almost every parameter is configurable, such as the number of times that alter increment is executed, threshold to consider a force value in range or total time that force must be in range to finish Approach process.

Algorithm 3 This algorithm describes the general idea of the Approach proccess.

- Read force
- Pass force to PID control.
- Alter in Z the distance returned by the PID.
- If force was not in range during last 100 milliseconds, repeat.

3.3.2 Move on the brain surface

Moving on the brain surface while keeping contact with surface is a combination of Approach process described in 3.3.1 and 3.2. Between a start position and orientation and the target position and orientation, several intermediate positions and orientations are created. Robot tool must be moved and orientated from the current position to the next intermediate one. Intermediate positions are calculated in the same way that the path in section 3.2. One intermediate position between two is created if the linear distance between them is higher than a step distance.

To move from one position to next the following algorithm is used:

Algorithm 4 Move between to points on the brain surface.

- Read force
- Pass force to PID control.
- Calculate transformation matrix between current and next.
- Get tool X and Y distance to next position from the matrix
- Get rotation vector of the matrix.
- Alter:
  - Z: the distance returned by the PID.
  - X, Y: small increment in X and Y calculated distance
  - Rotations: small increment in rotation vector direction.
- If X and Y distance is not null or Z axis is not pararell to the line between the desired point and the brain center, repeat.

In this process, tool Z movement is controlled by the PID as it is in Approach process. During movement, changes in force readings due to gravity after changing sensor orientation must be compensated in real time. Section 3.4 includes different methods to accomplish the task.

3.3.3 Orientate the tool

Once the tool is placed on the desired point to scan with the tumor sensor, it is necessary to orientate the tool perpendicular to the surface. Before starting the process forces are read as it is represented in figure 30. When the perpendicular orientation process is completed, lateral forces X and Y disappear. Two different methods have been developed to arrive to this orientation. They are described in this section.

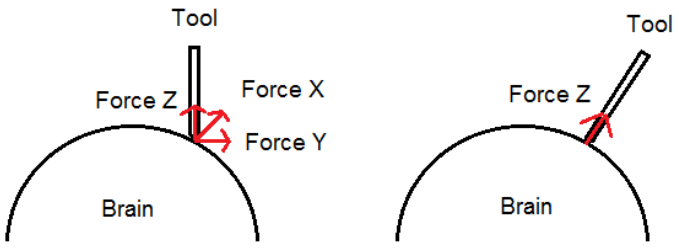


Figure 30: Measured forces change through orientation.

Dynamic orientation

In the dynamic method forces are read and robot tool orientation changed until measured force in X and Y is null. Robot is moved by using Alter mode. Due to the difficulty to totally eliminate errors in the measure through gravity compensation, it is necessary to include rotational limits as well as a minimum required force before changing the orientation. Movement in Z is active while rotating in order to keep soft contact with the brain surface as it was explained in section 3.3.1, but for simplicity this displacement is not included in the following algorithm.

Algorithm 5 Dynamic tool orientation.

- Force = ReadForce()
- If Angle(CurrentFrame,StartFrame) $<$ limit
  - If $|force_x| > minimum$: RotateY(-$K \cdot force_x$)
  - If $|force_y| > minimum$: RotateX($K \cdot force_y$)
  - Repeat while Rotation performed

Static orientation

In the static method, forces are read only once before starting the tool rotation. Due to the gravity compensation error, false forces in X and Y can be reduced by retracting the tool a few centimeters and storing them as offset. After offset compensation, tool is approached again to the surface by using the method explained in section 3.3.1. Forces are read again and then tool orientation is modified by rotating current frame a fixed angular distance in X and Y.

Angular distance is trigonometrically calculated from the force vector, which will be perpendicular to the surface as it is represented in figure 31. It is necessary to calculate the required rotational angle in X and Y. Atan2 is calculated to obtain the rotational angles for the surface vector. It is important to notice that this vector will be in the second or in the third quadrant, so it is necessary to add or substract π depending on if the angle is negative or positive, in order to obtain the angle from the tool coordinate system to the orientation vector which is te opposite to the perpendicular vector. How to calculate this distance is detailed in the following algorithm. Rotation is peformed with the method defined in section 3.3.2 because it allows to keep soft contact while rotating.

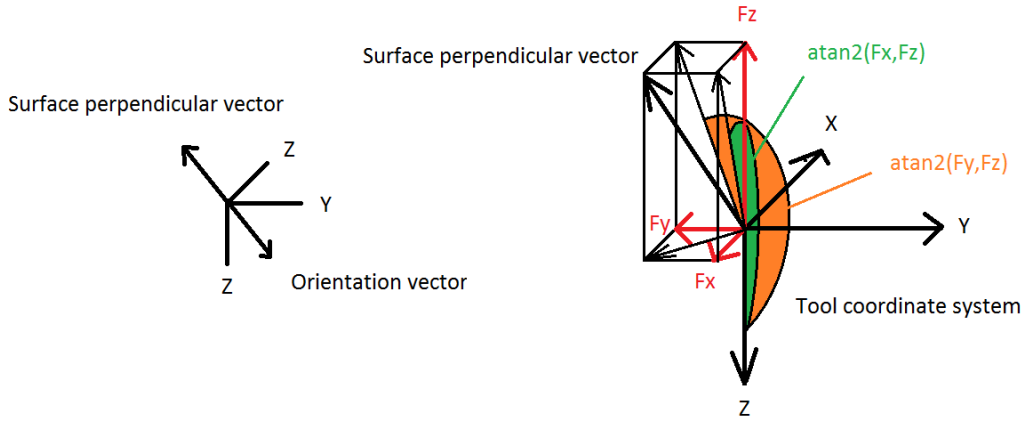


Figure 31: Angle calculation from the perpendicular vector.

Algorithm 6 Dynamic tool orientation.

- MoveZ(-SafeDistance)
- OffsetX = ForceX
- OffsetY = ForceY
- Approach()
- Force = ReadForce()
- temp = $atan2(force_x, force_z)$
- If $(temp \geq 0)$ $rotY = temp - \pi$
- Else $rotY = temp + \pi$
- temp = $atan2(force_y, force_z)$
- If $(temp \geq 0)$ $rotX = -(temp - \pi)$
- Else $rotX = -(temp + \pi)$
- $frame = CurrentFrame \cdot RotX(rotX) \cdot RotY(rotY)$
- ApproachMove($frame$)

3.3.4 Proportional-Integral-Derivative controller

Approach process and force maintenance is controlled by a Proportional-Integral-Derivative controller (PID). This loop feedback controller was implemented as an iterative algorithm by software. The controller on each iteration reads the current force and calculate the difference between its module and the desired force setpoint, and then it adjust the output distance in Z to be displaced by the robot alter mode. The controller has three independent parts. These parts are added to obtain the final output. First, the difference between the desired force and the current is calculated and stored as error. The proportional part is calculated by multiplying the error by constant K_p. The integral part is obtained by adding a given number of previous errors and multiplying them by the integral constant K_i. The derivative part is calculated by multiplying the difference between current and the last error, by the derivative constant K_d.

These parameters, including the number of errors taken into account in the integral part must be set up. Set up process is called PID tuning. First Ziegler-Nichols method was applied to have a general idea about PID parameters, then more detailed battery tests were performed and finally manually modified.

Test results of different parameters involved in the approach process and a general discussion is presented in section 5.5.

Ziegler-Nichols tuning method

Ziegler-Nichols method [8, 9, 10] is a heuristic method of tuning a PID controller. In a first step, integral and derivative gains are set to zero. Proportional gain is increased until the last constant oscillation of the tool tip is reached. Oscillation of the tool is determined by visual inspection. Then this proportional gain is stored as $K_u$and the oscillation period as T_u. Proportional constant can be calculed as $K_p = 0.6 \cdot K_u$, integral as $K_i = 2 \cdot K_p/T_u$ and derivative as $K_p = K_p \cdot T_u/8$. This method is used as starting point for a fine PID tuning.

Battery test

This methods tests different provided proportional, integral and derivative constants as well as other PID variables, Alter mode configuration and force/torque sensor properties. Each parameter is varied by stepping through a given range. The paramters to be evaluated are:

- **Desired force**: target module force in Newtons to be read by the force sensor.
- **Kp**: proportional constant.
- **Ki**: integral constant.
- **Kd:** derivative constant
- **Number of stored errors**: the number of the last errors which are used to calculate the integral part.
- **Force threshold**: allowed distance in Newtons to the desired force to consider a measure as stabilized.

- **Number of measures**: the number of sensor measures which are taken to calculate the mean force value.
- **Compensation**: if gravity compensation is active or forces are read directly.

These parameters were chosen because they allow to control all main aspects of the surface approach process. More extensively, desired force and the constants are required for the basic PID configuration. Number of stored errors is needed to solve the integral windup problem. Force threshold and number of measures allows to stabilize the tool oscillation by ignoring inherent force sensor resolution error. Finally, the possibility to switch on and off gravity compensation is interesting to determine if this compensation diminish the approach process performance. Range of the different parameters is set up before starting the test.

Algorithm 7 PID battery test. For legibility reasons, the 8 loop levels required to iterate a tuple are described in only one line.

- For each different parameters tuple:
  - ApproachTool(20 seconds)
  - Rank results

Due to the large number of parameters which are tested, time to complete all variations is too high. Computational complexity of the algorithm if all parameters are tested is $O(n^8)$ assuming the same number of possibilities for each parameter. This can be reduced by changing only a few parameters to rank the evolution and repeat the process after fixing the previous obtained best parameters.

Each parameter combination is evaluated by vertically approaching the sensor to the brain surface and keeping contact for twenty seconds. Distance to the brain surface is 10 mm so that it is possible to elimite false measured contacts if tool is not deep enough. All the force modules and the corresponding frames during the appraoch are stored in two vectors to be used in the ranking function. This function evaluates the parameters configuration performance. A parameters configuration is ranked by calculating different indicators. They represent different aspects of the approach process such as speed response or stability. Contact range is defined as forces whose difference between the desired force is less than threshold value. The indicators are:

- **Contact percentage**: percentage of the measured forces that are in contact range from the total.
- **Standard deviation of position**: the standard deviation of the tool Z distance once surface is contacted.
- **Overshoot**: maximum force module measured by the sensor minus target force module.
- **Max. depth**: maximum distance in millimeters reached by the tool.
- **Oscillations**: number of complete oscillations after surface is contacted.
- **Amplitude**: mean amplitude of the oscillations.
- **Period**: mean period of the oscillations.

Contact percentage is calculated by dividing the number of positions deeper than 10 mm that correspond to a force module in correct range by the total number of positions. This indicator evaluates the speed response as well as the stability. A higher contact percentage indicates better performance. Standard deviation evaluates the stability of the contact. It should be as low as possible. Overshoot evaluates the damage risk in the brain. Its value should be near the desired force value. Oscillations, amplitude and period give a description of the stability. It is interesting to find those testing parameters which produce low number of oscillations with long period and low amplitude.

To rank each different parameter tuple, they are combined into the following function:

$$rank = \frac{1-\%contact}{4} + \frac{Std.Deviation}{4} + \frac{Overshoot}{4} + \frac{Amplitude}{4}$$

The function is applicable if Amplitude and Period are positive. The lower the value obtained, the better the behaviour. This function is only representative because the speed response is not considered. The best ranked parameters with the lowest rank results must be evaluated by visual inspection.

Manual fine tuning

Finally, once a general idea of which parameters fit best, they are manually modified in order to obtain the best ajustment.

3.4 Gravity compensation

Gravity compensation is required to allow to change the hand orientation while contacting the surface. Due to the sensor resolution limits, it was necessary to develop different methods to increase precision. In this section they are described in order of complexity. The first method doesn't allow to move the tool while it is contacting the surface, in contrast to the rest, which are based on force/torque calibration.

3.4.1 Sensor reset method

Sensor reset on each position is the simplest method to compensate gravity derivation of read force values. It is also the first method which was developed for testing purposes. It is a limited approach because it requires to reset the sensor after changing tool orientation, which disallows to move the tool over the surface while contacting. The algorithm follows these steps:

Algorithm 8 Sensor reset compensation method

1. PLACE THE TOOL OVER THE BRAIN SURFACE.
2. RESET SENSOR.
3. APPROACH WHILE DIRECTLY MEASURING.

3.4.2 Force/torque computation method

This method was provided by the Institute for Robotics and Process Control to be tested in this thesis. It requires a previous calibration in which forces and torques are measured for six different limit positions. Once calibration step is complete, force and torque is trigonometrically calculated by using previous data and Euler II angles from the current position. Figure 32 shows the used coordinate system in the calculation and figure 33 the different calibration positions. Once force measures are stored on each calibration position, it is possible to obtain an offset vector for a fixed coordinate system. After rotating the hand, this offset vector changes its projections. If the rotation angles are known, it is possible to calculate the force to compensate with the given algorithm.

The method was thought to be used with the coordinate system which is described in next figured as Calibration coordinate system, so that it is necessary to transform the current tool system to this system. This include forces and also Euler angles, as it will be described in the algorithm.

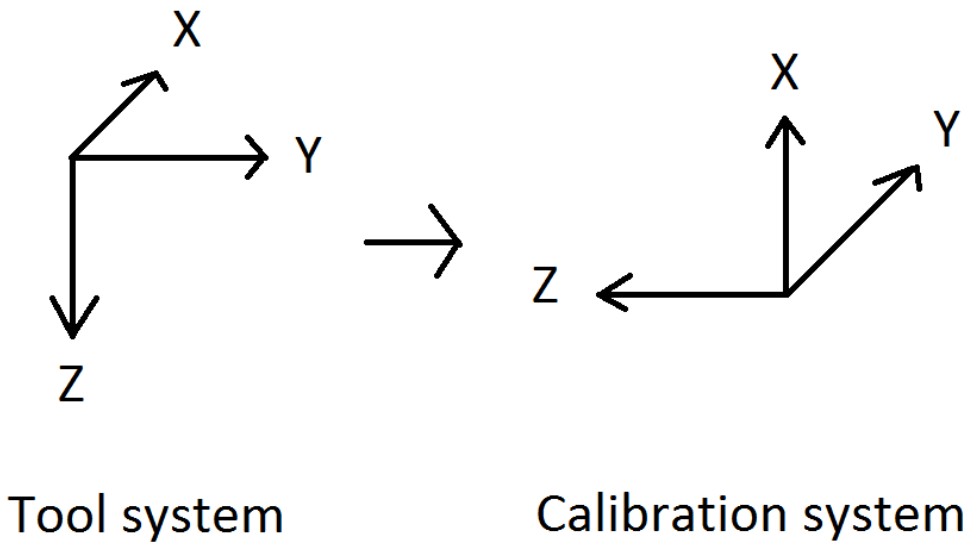


Figure 32: Side view of the six different calibration positions.

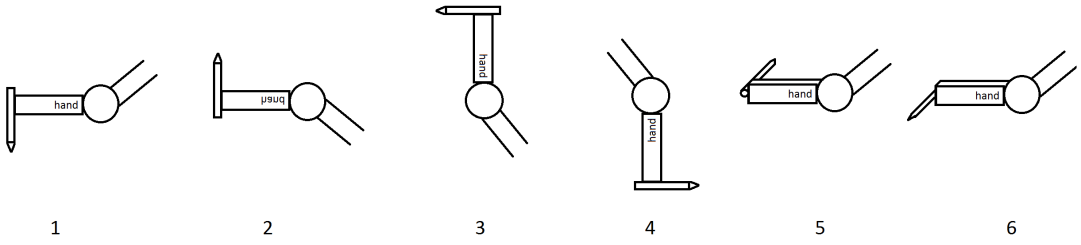


Figure 33: Side view of the six different calibration positions.

Algorithm 9 Force/torque computation method.

CALIBRATION STEP:
FOR EACH CALIBRATION POSITION STORE:

- FORCEX = -FORCEZ
- FORCEY = FORCEX
- FORCEZ = -FORCEY

WORK STEP:
CURRENTFRAME * FROTY($\frac{\pi}{2}$) * FROTX($\frac{\pi}{2}$)
(Z1,Y,Z2) = CALCEULERII(CURRENTFRAME)
OFFS = (POSITION1.FORCEX + POSITION2.FORCEX) * 0.5
GAIN = (POSITION2.FORCEX - POSITION1.FORCEX) * 0.5
OFFSETX = OFFS + SIN(Y) * SIN((Z2-$\frac{\pi}{2}$)) * GAIN;
OFFS = (POSITION5.FORCEY + POSITION6.FORCEY) * 0.5
GAIN = (POSITION6.FORCEY - POSITION5.FORCEY) * 0.5
OFFSETY = OFFS + SIN(Y) * SIN(Z2) * GAIN;
OFFS = (POSITION3.FORCEZ + POSITION4.FORCEZ) * 0.5
GAIN = (POSITION4.FORCEZ - POSITION3.FORCEZ) * 0.5;
OFFSETZ = OFFS + SIN((Y-$\frac{\pi}{2}$)) * GAIN;
FORCEX = FORCEX - OFFSETY
FORCEY = FORCEY + OFFSETZ
FORCEZ = FORCEZ + OFFSETX

3.4.3 Nearest stored measure method

Before applying this method, the previous calibration stage defined in 3.5 must be completed. This is the simplest method based on calibration. In order to compensate a read measure, the nearest force/torque values are found and subtracted from the current values.

In a preliminar stage of the method development, quaternion representation for the orientations were used, but later it was changed to Euler angle representation. The main reason for this decision was to take advantage of the calibration method developed for the fitting curve method which maked redundant to have two different calibration methods. Despite Euler representation presents ambiguities this problem is solved by covering the whole Euler angle space during the calibration. These ideas are detailed explained in section 3.4.5.

The nearest orientation is defined as the stored value which have the smallest root mean square of the difference between its Euler ZYZ angles and current ZYZ angles. Current ZYZ angles are calculated from tool frame as it is described in section 3.5.4. Due to the relative small number of stored measures, for example 512 for a 45 degree step, it is possible to use a simple linear search to find the nearest value or use a logarithmic search after sorting the data. Data is loaded in memory from the file at the initialization.

Algorithm 10 Nearest stored measure compensation method.

- OBTAIN($z1_{current}$,$y_{current}$,$z2_{current}$) FROM ROBOT
- FOR EACH STORED FORCE
  - $Distance_i = \sqrt{(z1_{current} - z1_i)^2 + (y_{current} - y_i)^2 + (z2_{current} - z2_i)^2}$
    - If $Distance_i <$ SMALLESTDISTANCE:
      - $Distance_i =$ SMALLESTDISTANCE
      - STOREDFORCE $= Force_i$
- RETURN CURRENTFORCE-STOREDFORCE

3.4.4 Weighted average method

This method is a improved version of 3.4.3. Instead of only using the nearest stored value for compensation, the four nearest stored orientations are taken to calculate the weighted mean value taking into account the distance between current Euler ZYZ and the four stored. Calculated values are subtracted to the current force and torque to compensate. Division by zero when if null distance is taken into account. It is also possible to take as distance its squared value.

Algorithm 11 Weighted mean value method.

- OBTAIN($z1_{current}$,$y_{current}$,$z2_{current}$) FROM ROBOT
- FOR EACH STORED FORCE
  - $Distance_i = \sqrt{(z1_{current} - z1_i)^2 + (y_{current} - y_i)^2 + (z2_{current} - z2_i)^2}$
    - Store the 4 force/torque with smallest $Distance$
- $StoredForce = (\frac{Force_1}{Distance_1} + \frac{Force_2}{Distance_2} + \frac{Force_3}{Distance_3} + \frac{Force_4}{Distance_4}) \cdot (Distance_1 + Distance_2 + Distance_3 + Distance_4)$
- RETURN $CurrentForce - StoredForce$

3.4.5 Polynomial surface fitting method

This method is the most complex and give the best compensation results. Calibration process is the same as in 3.5, but in this case the result is obtained through a polynomial surface fitting by using all available stored values. The two functions to be fitted are $f(z_1, y, z_2) = force$ and $f(z_1, y, z_2) = torque$. Polynomial surface fitting is accomplished by using an itarative fitting

process in which dimensions are reduced one by one. For readability reasons, force is considered as only one value, but the algorithm is independently applied for each force axis X, Y and Z, and for each torque X, Y and Z.

The main difficulty of this process is to build a multidimensional fitting by using one dimension polynomial curve fittings. The developed algorithm is based on the Divide and Conquer strategy because a multidimension fitting can be divided in fitting the results obtained by reducing one dimension the problem. The algorithm is divided in three different levels. One level for each dimension. In the first level, only the last dimension is considered. It is necessary to calculate as many fitting curves as different tuples of the rest of the dimensions exist. This is illustrated in the next figure by using one example. In this case, z1 and y can take only values 1 and 2. Consequently, it exists 4 different tuples. Fitting curves are obtained from the data which was previously stored in the calibration stage. The desired z2 value is applied to each function to obtain 4 different results. The process is repeated in the next level using the output of the last level as data to calculate the fitting curves. In this case, it exists 2 possibilities which are the number of different values that z1 can take, so that 2 functions are fitted and then the desired y value applied. In the third level the final value is obtained through the same process.

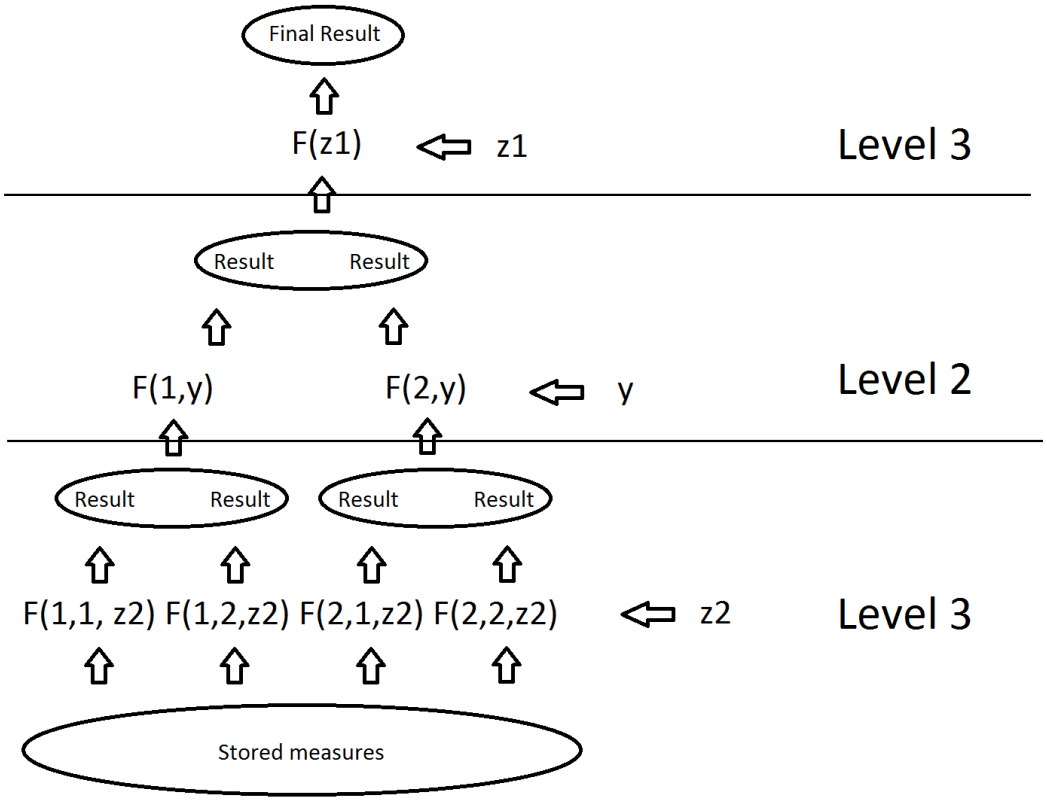


Figure 34: Simple example of the fitting algorithm.

Given a number of dimensions d and the number of possible values n which each dimension can take, the number of fitting curves which is required to calculate is $\sum_{i=1}^{d} n^{i-1}$. The first level fitting

curves parameters can be precalculated and stored in memory in order to speed up compensation process.

It was decided to use intrinsic Euler II angles because they allow to easily develop a calibration method which guarantee that points are separate a desired angular distance between them on each dimension. It is not trivial to obtain the same effect when using quaternions. This allow to have always the same number of measures on each fitting curve and also reduces the problem to three different dimensions. Ambiguities in the Euler representation don't affect the compensation process. They just only add a bit more time in the calibration stage.

Polynomial curve fitting [11]

Polynomial curve fitting is the process to obtain the function $f(x) = a_0 + a_1 x + \ldots + a_k x^k$ associated to n points (x_i, y_i). k parameter in this method can be freely selected, but accuracy of the results are determined by a good selection.

The function can be written in matrix notation:

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_1^0 & x_1^1 & \cdots & x_1^k \\ \vdots & \vdots & \ddots & \vdots \\ x_n^0 & x_n^1 & \cdots & x_n^k \end{bmatrix} \begin{bmatrix} a_0 \\ \vdots \\ a_k \end{bmatrix} \text{ or } y = X \cdot a$$

Polynomial coefficients $a_0, ..., a_k$ are calculated by solving $a = (X^T X)^{-1} X^T y$

Armadillo C++ library [12] is used to operate with matrices and calculate transposition and inversion.

Step 1: Z2 elimination

In the calibration stage each(Z_1, Y, Z_2) tuple is associated to a read force and torque. One polynomial fitting curve $f(z_2) = force$ is obtained for each different pair of (Z_1, Y) taking Z_2as independent value and read force and torque as dependent. These fitting curves are precalculated and stored in memory when data is loaded in order to speed up computing. Then current Z_2 value obtained from the tool frame is substituted for each fitting curve. $Z_2$is eliminated and results are saved for next step.

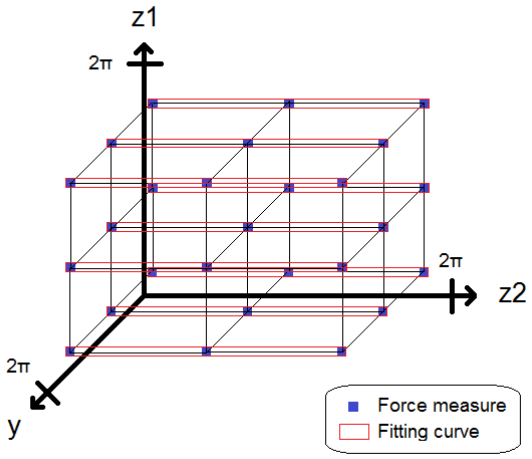


Figure 35: First step of the three-dimensional surface fitting. Each fitting curve is calculated using measured values.

Step 2: Y elimination

Similarly to the previous step, one polynomial fitting curve$f(y) = force$ is obtained for each different Z_1value. Now each Y is taken as independent value and step 1 results as dependent for the fitting process. After solving each fitting curve for the current tool frame Y Euler rotation angle, results are saved.

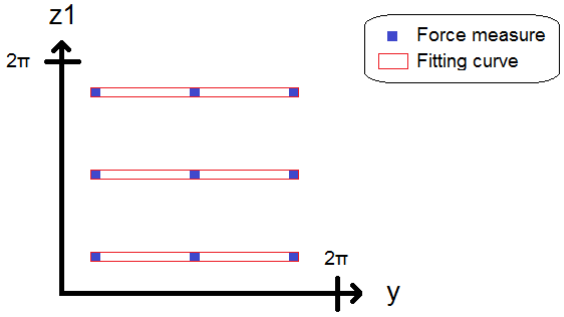


Figure 36: Now each force measure is obtained as result from the previous step.

Step 3: Z2 elimination

In the last step only one fitting curve is calculated. $f(z_1) = force$ is obtained taking all possible Z_1as independent value and the result associated from the previous step as dependent. Final force and torque results are obtained by solving the curve for current tool frame Z_1 rotation.

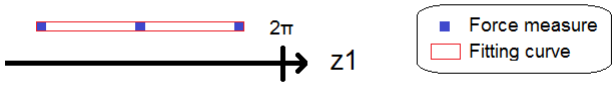


Figure 37

3.5 Gravity calibration

3.5.1 Introduction

Gravity calibration is the process in which force and torque of different tool orientations are stored in order to use this data in a compensation method. Each orientation can be described by using Euler angles. Euler intrinsic ZYZ rotation convention is followed by the process [13]. In order to place the hand on each orientation, robot joint values are directly modified from the start position represented in figure 38. This allow to easily imitate ZYZ rotations by turning robot joint 4, joint 5 and joint 6.

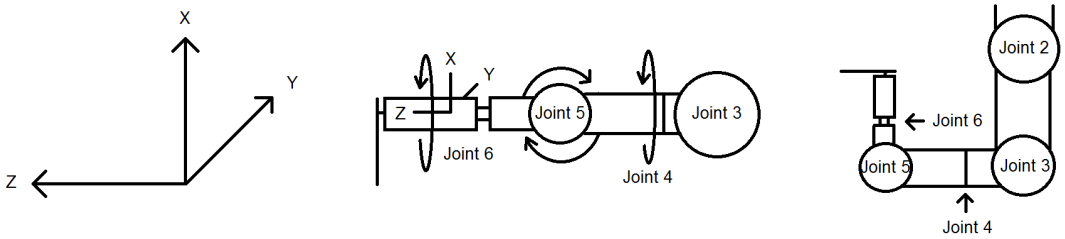


Figure 38: Calibration coordinate system placed on the robot hand and robot joints.

Start joint values described in the following table. Sensor is reset in this position. Joint 3 and 5 needs to be aligned in order to obtain the same exact orientations with frame rotations or by direct joint modifying. Section 3.5.3 introduces this concept.

Joint	1	2	3	4	5	6
Degrees	0.0	-90.0	180.0 + joint3offset	joint4offset	joint5offset	0.0

3.5.2 Calibration process

Calibration process is composed of a three level loop. Each level is used to compute one ZYZ rotation. Due to the countless possible orientations, only a discrete number is visited. If we consider ZYZ rotation as a three dimensional space, each dimension is divided using a fixed step to obtain which positions are required to be visited. The necessary time to complete the calibration process makes step values under 22.5 degrees impracticable, as it is represented in figure 39.

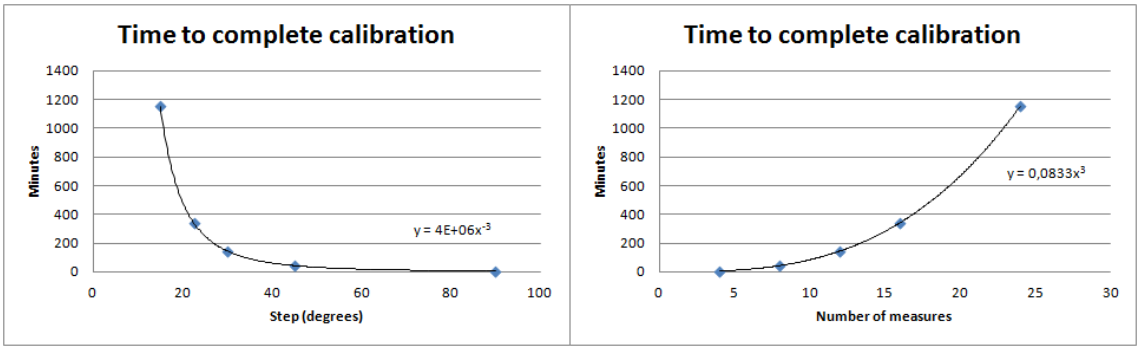


Figure 39: Time to complete calibration with 20 measures on each position.

Algorithm 12 Interpolation and curve fitting calibration process for a given step S in degrees.

- For joint4 = [-180 + joint4offset,180 + joint4offset) step S
  - For joint5 = [-90 + joint5offset,90 + joint5offset) step S
    - For joint6 = [-180,180) step S
      - MoveRobot(joint 4, joint 5, joint6);
      - StoreForces
      - StoreZYZ(z1, y, z2)
  - For joint5 = [-90 + joint5offset,90 + joint5offset) step S
    - For joint6 = [-180,180) step S
      - MoveRobot(-joint 4, joint 5, joint6);
      - StoreForces
      - StoreZYZ(z1, y + 180, z2)

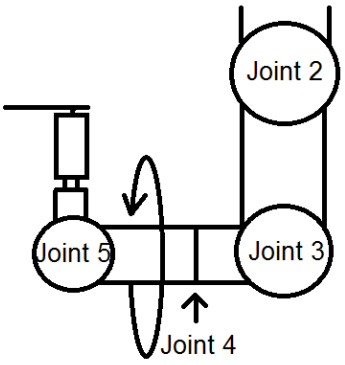


Figure 40: Positive joint 4 rotation.

First loop In the most external loop, joint 4 is rotated 360º to compute first Z rotation through fixed steps as it is indicated in figure 40-

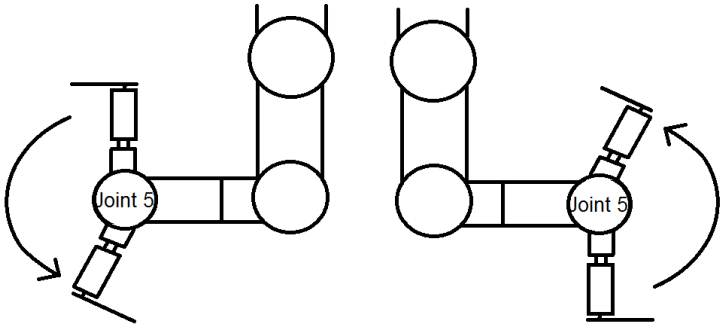


Figure 41: Positive joint 5 rotation.

Second loop In the second loop, joint 5 is rotated twice [-90 + joint3offset, 90 + joint3offset) because of the robot geometry. In the first part it is normally rotated, but in the second, joint 3 is changed to 0.0 (plus offset) and joint 4 to its opposite value in order to preserve current first Z rotation. Figure 41 describes this change.

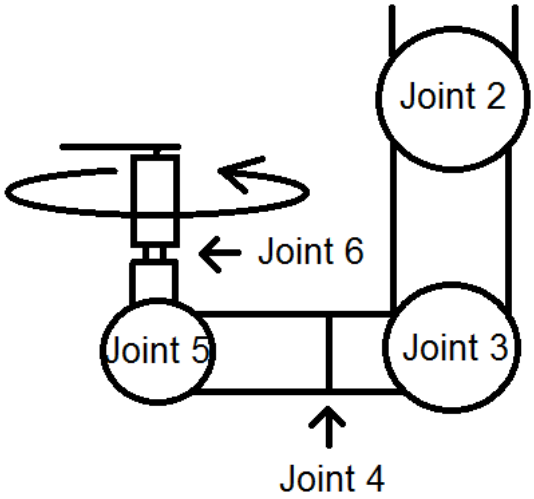


Figure 42: Positive joint 6 rotation.

Third loop Last Z Euler rotation is performed by rotating joint 6 from -180 to 180 degrees. It is visualized in figure 42.

3.5.3 Alignment correction

Alignment correction is divided in two parts. The first one is related to joint 5 alignment and the second one to the world alignment.

Before calibrating it is necessary to assure that joint 5 exactly aligns its both links. Otherwise joint rotations will not coincide with Euler ZYZ rotations due to the swinging. A difference of 0.1 degree can increse force precision error more than 0.1 Newtons, so that a perfect alignment is required. The offset which will be added to joint 5 is represented in figure 43 as the difference between both links. Calculation process is detailed in section 3.6.1.

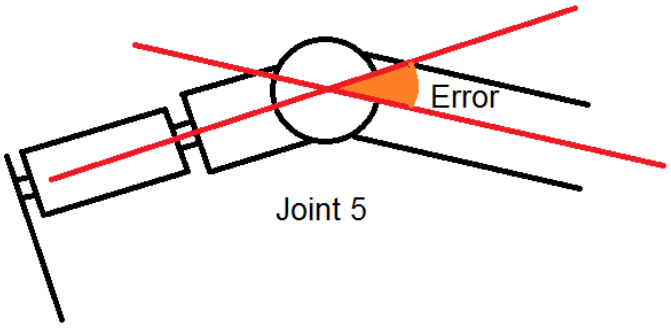


Figure 43: The angle difference between both axis is what it is necessary to correct.

Once joint 5 is aligned, optionally joint 3 and 4 offset can be modified in order to align the robot arm with the real vertical gravity force, but it is not required for compensation process because forces are stored according to the initial reference joints. In both calibration and compensation processes sensor is reset in the same position, so that in both calibration and compensation common positions the same values are read if joint 5 is aligned. Alignment method is detailed in section 3.6.

3.5.4 Intrinsic ZYZ Euler angles calculation

Although in the calibration process ZYZ angles are directly calculated by joint angles, it is necessary to know the ZYZ rotation angles from the calibration coordinate system for any given frame. This is calculated by obtaining the intrinsic ZYZ Euler angles of the transformation frame between the calibration coordinate system and the tool frame. Once the transformation frame is known, the robot library function CalcRotEuler_II is used to obtain ZYZ angles.

Because ZYZ angles are stored respect to the coordinate system described in figure 44 as *Calibration*, it is necessary to obtain ZYZ rotation angles for any given frame using this reference. First a frame is defined with the same orientation as the Calibration reference. This frame could be

placed in any position because its translational component is not used in this method. Then is calculated the transformation frame between it and the current tool frame. ZYZ are obtained from this transformation frame.

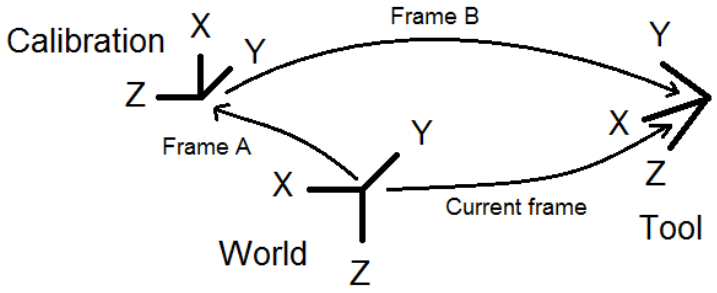


Figure 44: Transformation between calibration frame to tool frame.

Algorithm 13 Intrinsic ZYZ Euler angles calculation.

- Frame A $= (X = (0, 0, -1), Y = (0, 1, 0), Z = (1, 0, 0), P = (0, 0, 0))$
- Frame B $= Inverse(A) \cdot CurrentFrame$
- CalcRotEuler_II(B)

3.6 Joints alignment

There are two error sources which are intented to be corrected through methods defined in this section. The first one is the error in joint 5 angle which cause in gravity calibration method errors in the measured forces. The problem, which was illustrated in section 3.5.3, is solved by using the method defined in subsection 47. The second error source comes from the wrong alignment of the force sensor with the gravity force. Z axis of the sensor must be perfectly aligned with real weight force vector. Changes in joint 3 angle and 4 can correct the alignment. The problem is exemplified in picture 45. This error is independent to the first one, but it requires to solve the first before. Otherwise if sensor is aligned with gravity, but joint 5 is not correct, gravity calibration method will obtain inaccurate results. The order of the methods execution is important because joint 5 modification will change the alignment of the force sensor with the weight vector.

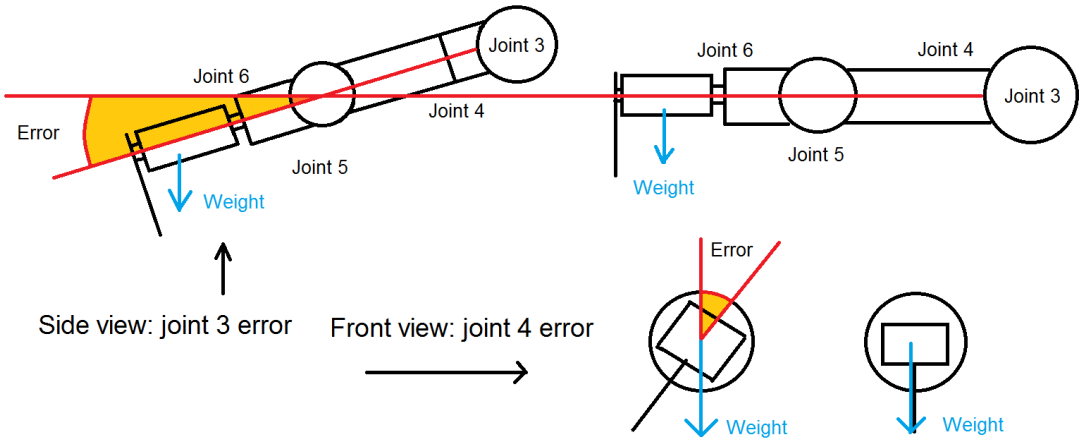


Figure 45: The angle difference which must be added to joint 3 and 4 to aligned the sensor with the real gravity force.

Although robot joints are aligned by visual inspection when the robot is turned on, slight differences between the alignment marks can considerably increase compensation error. So that alignmet methods were developed allowing to reduce joint error less than 0.1 degrees.

Joint 1 alignment can also have impact in the force measurement process if it is not totally perpendicular to the floor. It could be handled with a similar method like the developed for the other joints, but due to the force sensor precision limitations was omitted. If joint 3 is aligned, it is not necessary to align joint 2 because its offset will be zero. Instead of aligning joint 3 is possible to align joint 2 with a very similar method.

3.6.1 Joint 5 alignment

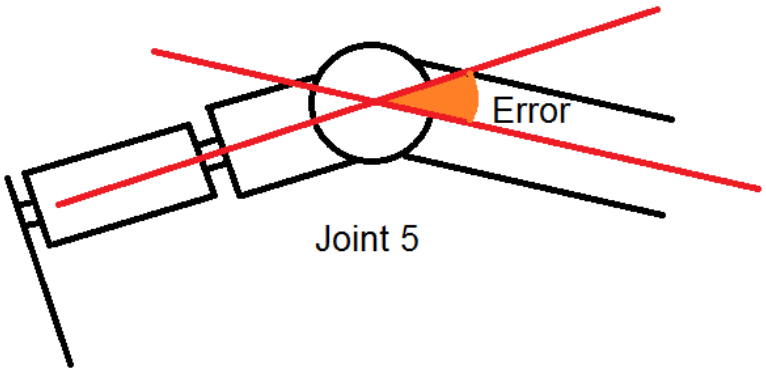


Figure 46: Joint 5 error representation.

The objective of this method is to reduce the angular error between the axes of the two links which are connected by joint 5. Figure 43 describes this error. Error is enlarged in the diagram to facilitate its visualization.

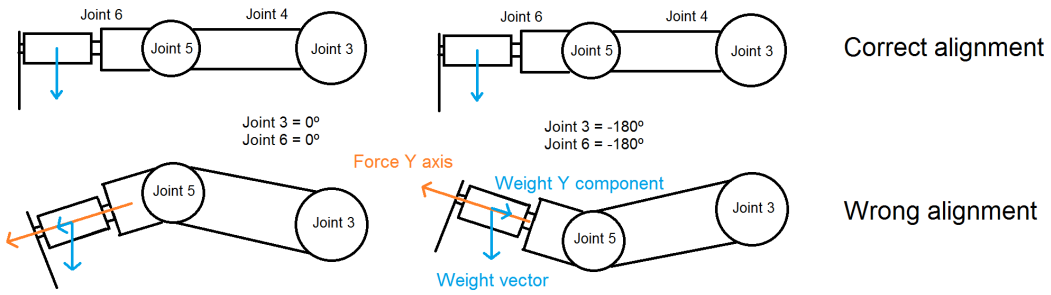


Figure 47: Joint 5 alignment. Start position is drawn in the left part and rotated (opposite) in the right.

The process performs a logarithmic iterative approach to the solution. In this algorithm first a initial joint 5 range is stablished, for example [-10º,10º]. On each iteration, measured force is compared between start and rotated positions for the extremes of the range and the middle. Joint 3 and 6 are rotated at the same time 180º from the start position as it is explained in figure 47. If joint 5 is correctly aligned, tool orientation will be preserved and consequently the same force values will be read, so the difference between these positions is null. If not, force Y weight component will be different on each position. Once the difference in the extremes and in the middle is known, range is divided in two halves. Mathematically, if the difference in Y axis is considered as a function, it has only one minimum, which will be the correct joint 5 offset. The range half in which the minimum is located will not be monotonic and the other part will always be strictly increasing or decreasing, so that extremes of the monotonic half will have less variation than the others. The half where minimum is expected is taken as range in next iteration and the process continues until desired precision.

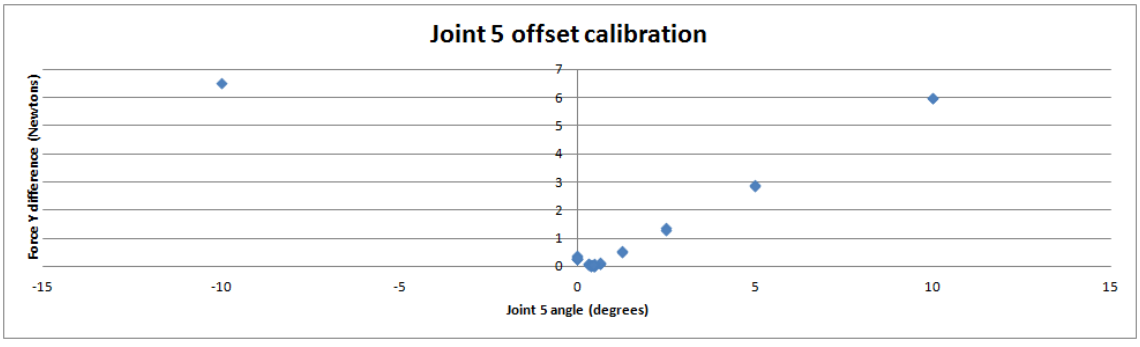


Figure 48: Example of the joint 5 calibration results. Correct joint 5 offset in this case is 0.4º.

Algorithm 14 Joint 5 alignment method.

- FOR i = $limit_{left}$ TO $limit_{right}$ STEP $\frac{limit_{right}-limit_{left}}{2}$
  - SETROBOTJOINTS(0°, -90°, 180°, 0°, i, 0°)
  - $Force_1 = ReadForce()$
  - SETROBOTJOINTS(0°, -90°, 180°, -180°, i, -180°)
  - $Force_2 = ReadForce()$
  - $difference_i = Force_1 - Force_2$
- A = $\left| difference_{limit_{left}} - difference_{limit_{left}+\frac{limit_{right}-limit_{left}}{2}} \right|$
- B = $\left| difference_{limit_{right}} - difference_{limit_{left}+\frac{limit_{right}-limit_{left}}{2}} \right|$
- IF A < B
  - $limit_{left} = limit_{left}$
  - $limit_{right} = limit_{left} + \frac{limit_{right}-limit_{left}}{2}$
- ELSE
  - $limit_{left} = limit_{left} + \frac{limit_{right}-limit_{left}}{2}$
  - $limit_{right} = limit_{right}$
- REPEAT WHILE $limit_{right} - limit_{left} > \epsilon$

3.6.2 Joint 3 and 4 alignment

There are two different approaches to align joint 3 and 4. It is possible to align them sequentially one by one or both at the same time. The first process is very similar to the one for joint 5 described in previous section. Joint 4 is aligned and then joint 5. The second process builds a result matrix by changing both parameters at the same time.

Joint 4

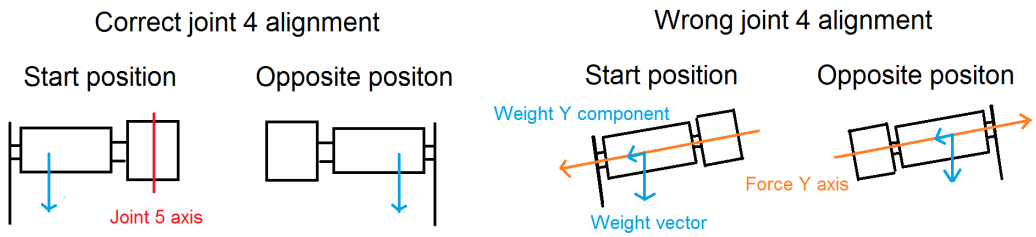


Figure 49: Front view of the robot hand during alignment process.

This process is very similar to joint 5 alignment process. Two different positions are compared to determine how force Y component changes between two opposite positions for each offset. For a given joint 4 offset, if sensor is well aligned with gravity force and joint 5 axis is perpendicular to the floor, rotating 180° in joint 5 won't change measured force. This is illustrated in figure 49. When joint 4 is wrongly aligned, weight component measured in Y force axis changes.

Algorithm 15 Joint 4 alignment method.

- FOR i = $limit_{left}$ TO $limit_{right}$ STEP $\frac{limit_{right}-limit_{left}}{2}$
  - SETROBOTJOINTS(0°, -90°, 180°, 90° + i, -90° + JOINT5OFFSET, -90°)
  - $Force_1 = ReadForce()$
  - SETROBOTJOINTS(0°, -90°, 180°, 90° + i, 90° + JOINT5OFFSET, -90°)
  - $Force_2 = ReadForce()$
  - $difference_i = Force_{Y1} - Force_{Y2}$
- A = $\left|difference_{limit_{left}} - difference_{limit_{left}+\frac{limit_{right}-limit_{left}}{2}}\right|$
- B = $\left|difference_{limit_{right}} - difference_{limit_{left}+\frac{limit_{right}-limit_{left}}{2}}\right|$
- IF A < B
  - $limit_{left} = limit_{left}$
  - $limit_{right} = limit_{left} + \frac{limit_{right}-limit_{left}}{2}$
- ELSE
  - $limit_{left} = limit_{left} + \frac{limit_{right}-limit_{left}}{2}$
  - $limit_{right} = limit_{right}$
- REPEAT WHILE $limit_{right} - limit_{left} > \epsilon$

Joint 3

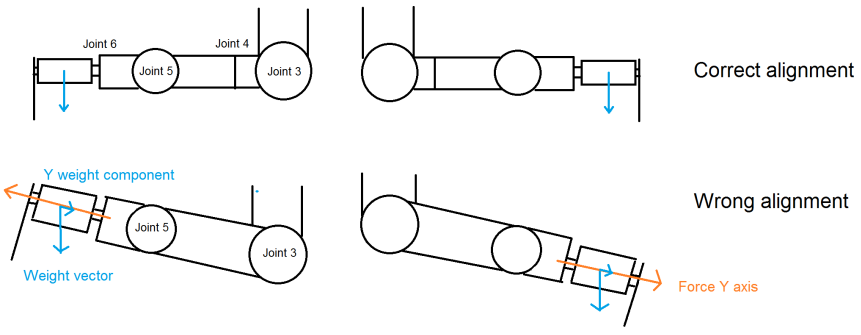


Figure 50: Side view of the robot hand during alignment process.

The same process as for joint 5 and 4 but changing the opposite positions joints.

Algorithm 16 Joint 3 alignment method.

- FOR i = $limit_{left}$ TO $limit_{right}$ STEP $\frac{limit_{right}-limit_{left}}{2}$
  - SETROBOTJOINTS(0°, -90°, 180° + i, 0° + JOINT4OFFSET, 0° + JOINT5OFFSET, 0°)
  - $Force_1 = ReadForce()$
  - SETROBOTJOINTS(0°, -90°, 0° + i, 0° + JOINT4OFFSET, 0° + JOINT5OFFSET, 180°)
  - $Force_2 = ReadForce()$
  - $difference_i = Force_{Y1} - Force_{Y2}$
- A = $\left|difference_{limit_{left}} - difference_{limit_{left}+\frac{limit_{right}-limit_{left}}{2}}\right|$
- B = $\left|difference_{limit_{right}} - difference_{limit_{left}+\frac{limit_{right}-limit_{left}}{2}}\right|$
- IF A < B
  - $limit_{left} = limit_{left}$
  - $limit_{right} = limit_{left} + \frac{limit_{right}-limit_{left}}{2}$
- ELSE
  - $limit_{left} = limit_{left} + \frac{limit_{right}-limit_{left}}{2}$
  - $limit_{right} = limit_{right}$
- REPEAT WHILE $limit_{right} - limit_{left} > \epsilon$

Joint 3 and 4

This method evaluates how forces change for a given joint 3 and 4 offset values. Robot hand is placed so that Y axis is perpendicular to the floor. If sensor Y axis is correctly aligned with weight vector, rotations in joint 6 won't change measured values. If not, it will appear the same deviations which were described in the previous sections. Figure 51 illustrates the position of the robot hand. This method rotates joint6 and stores in a file the difference between positions for each joint 3 and 4 angles to be tested. The lowest stored value will coincide with the most accurate joint position.

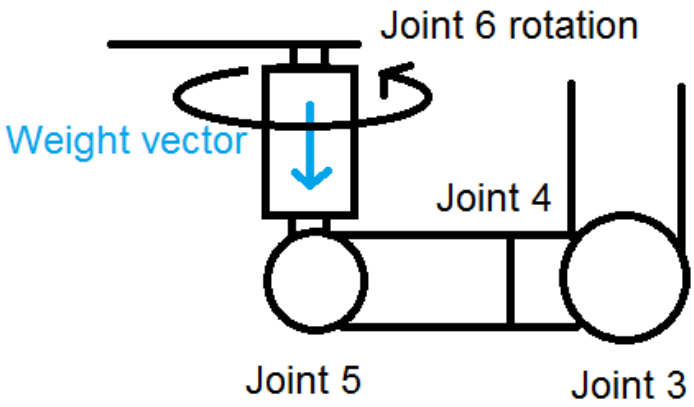


Figure 51: Joints start position.

Algorithm 17 Joint 3 and 4 alignment test.

- For j3 in [min, max] step AngleStep
  - For j4 in [min, max] step AngleStep
    - SetRobotJoints(0°, -90°, 180° + j3, j4, 90.0 + Joint5Offset, 0)
    - Forces1 = ReadForce()
    - SetRobotJoints(0°, -90°, 180° + j3, j4, 90.0 + Joint5Offset, 180)
    - Forces2 = ReadForce()
    - SetRobotJoints(0°, -90°, 180° + j3, j4, 90.0 + Joint5Offset, 90)
    - Forces3 = ReadForce()
    - SetRobotJoints(0°, -90°, 180° + j3, j4, 90.0 + Joint5Offset, -90)
    - Forces4 = ReadForce()
    - Save |Forces1 - Forces2| + |Force3 - Forces4|

4 Implementation

This section focuses on introducing to the system analysis and design. Software has been developed following a Spiral Lifecycle Model [14] in combination with Software Prototyping [15]. It has been implemented in Microsoft Visual Studio 2010 environment using C++ language and some libraries such as the robot library provided by the Robotic Institute, force sensor library, Microsoft MFC [16] and other tool libraries.

4.1 Use case

The main application behaviour is schematized in the diagram 52. Two different roles can be considered: the surgeon and the technical operator. The surgeon role should be independent to the technical operator, although he could be the same person.The surgeon role interacts only with the scan functionality. This functionality performs the brain surface analysis with the robot and produces a three-dimensional map of the results. The technical operator role interacts with the calibration operations and with the different test options.

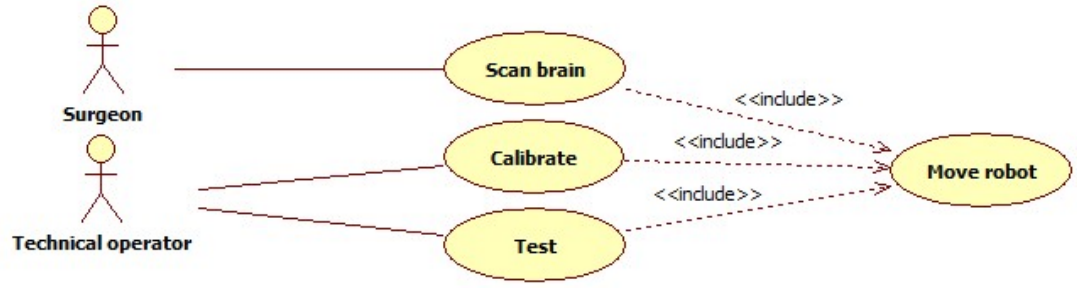


Figure 52: Use case diagram

4.2 Data flow

In this subsection an overwiew of the data transformations is presented. Data flow in the application is divided in three main processes. The user gives the configuration parameters to each process. Calibration is the process which performs the gravity compensation calibration. Analysis involves all the necessary operations to construct the tumor map. Finally it sends the information to the user through the screen or by producing a data file. All the different tests which were developed in the thesis are included in the Tests process.

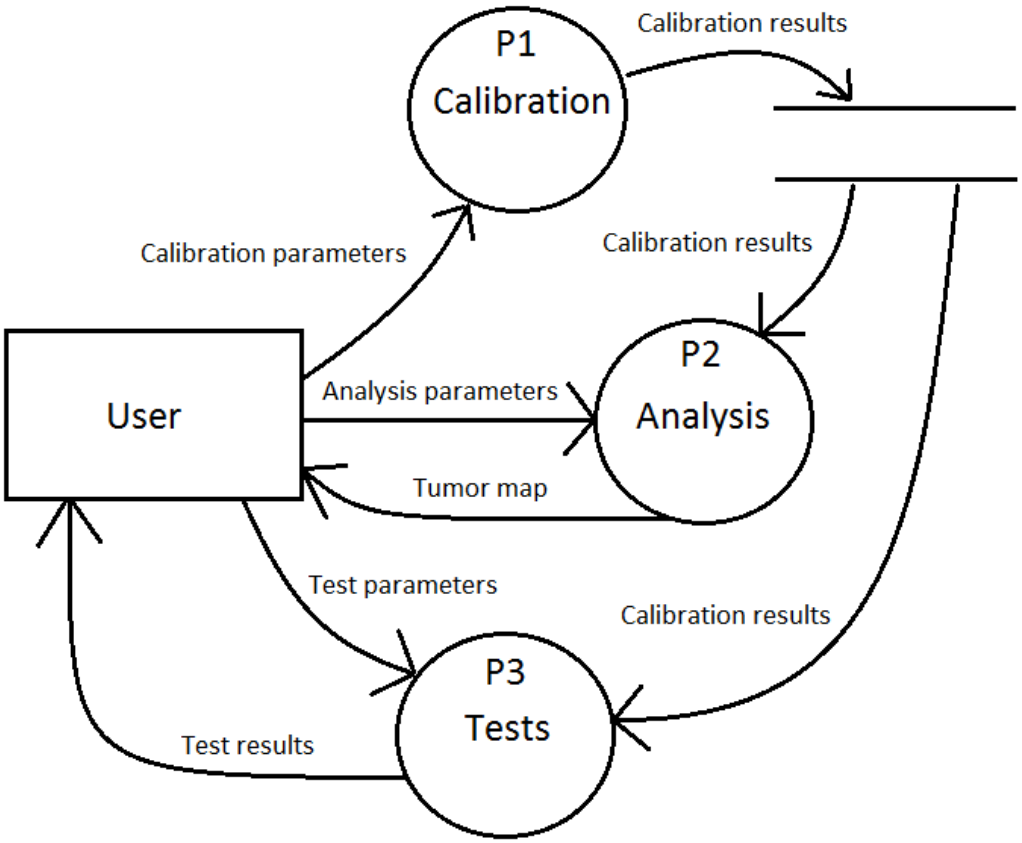


Figure 53: Context level of the Data Flow Diagram.

4.3 System architecture

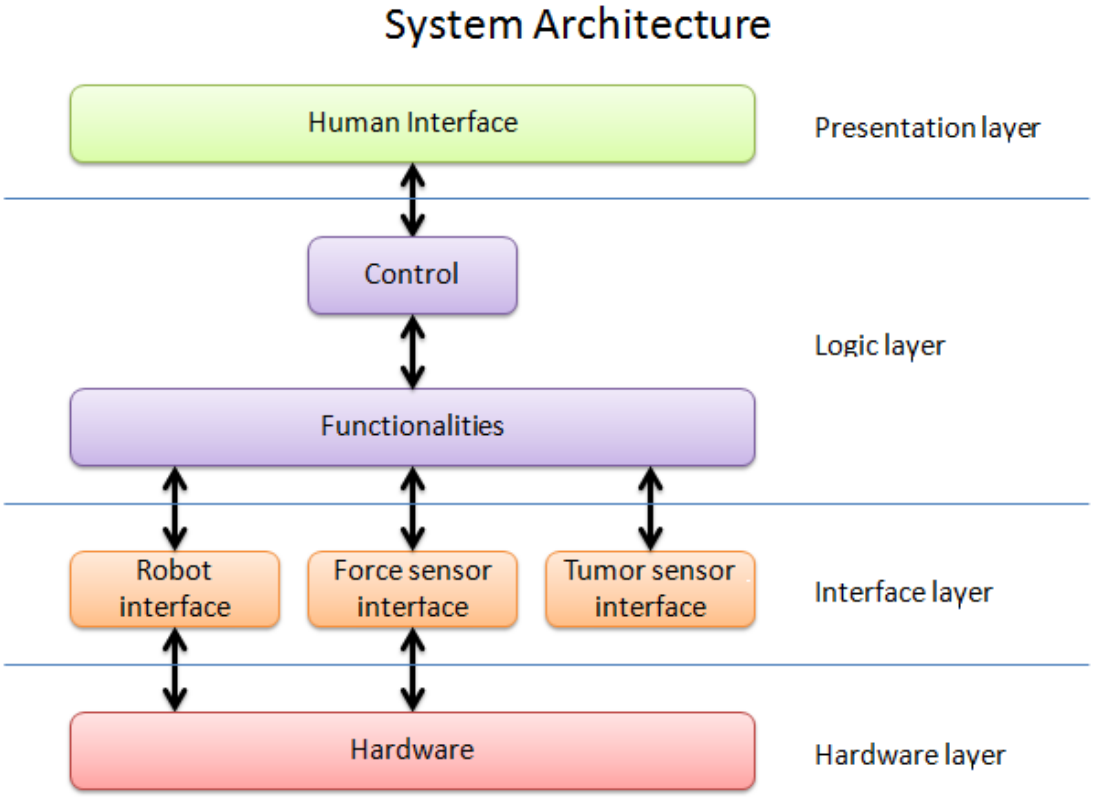


Figure 54: System architecture diagram.

System architecture is divided in four different layers. The hardware layer includes the force/torque sensor and the robot. Developed robot and force/torque sensor interfaces are in the Interface Layer. This part controls the orders which are given to the robot and the information retrieval from the robot and force sensor. Software provides a thread-safe interface and also some improvements to the data acquisition such as speed improvement through buffering. Logic layer controls analysis process, robot positioning, safe moving and brain approaching. All the process functionalities are controlled through a control interface which is the classControl class. The presentation layer displays the information and receives orders from the user. Software provides one esay but complete human interface and results displaying using 2D and 3D graphics.

The software has been implemented following a general Model-View-Controller design pattern, but adapted to the nature of the thesis. Interface layer is considered as the Model, the logic layer as the Controller and the presentation layer as the View. Beside this general pattern, process control, robot library and force sensor were implemented by using Singleton pattern. There is only one unique instance of them during the application execution.

4.4 Class structure

The class diagram in figure 55 describes the software structure. Comments in the code describe in detail each class properties and functionalities.

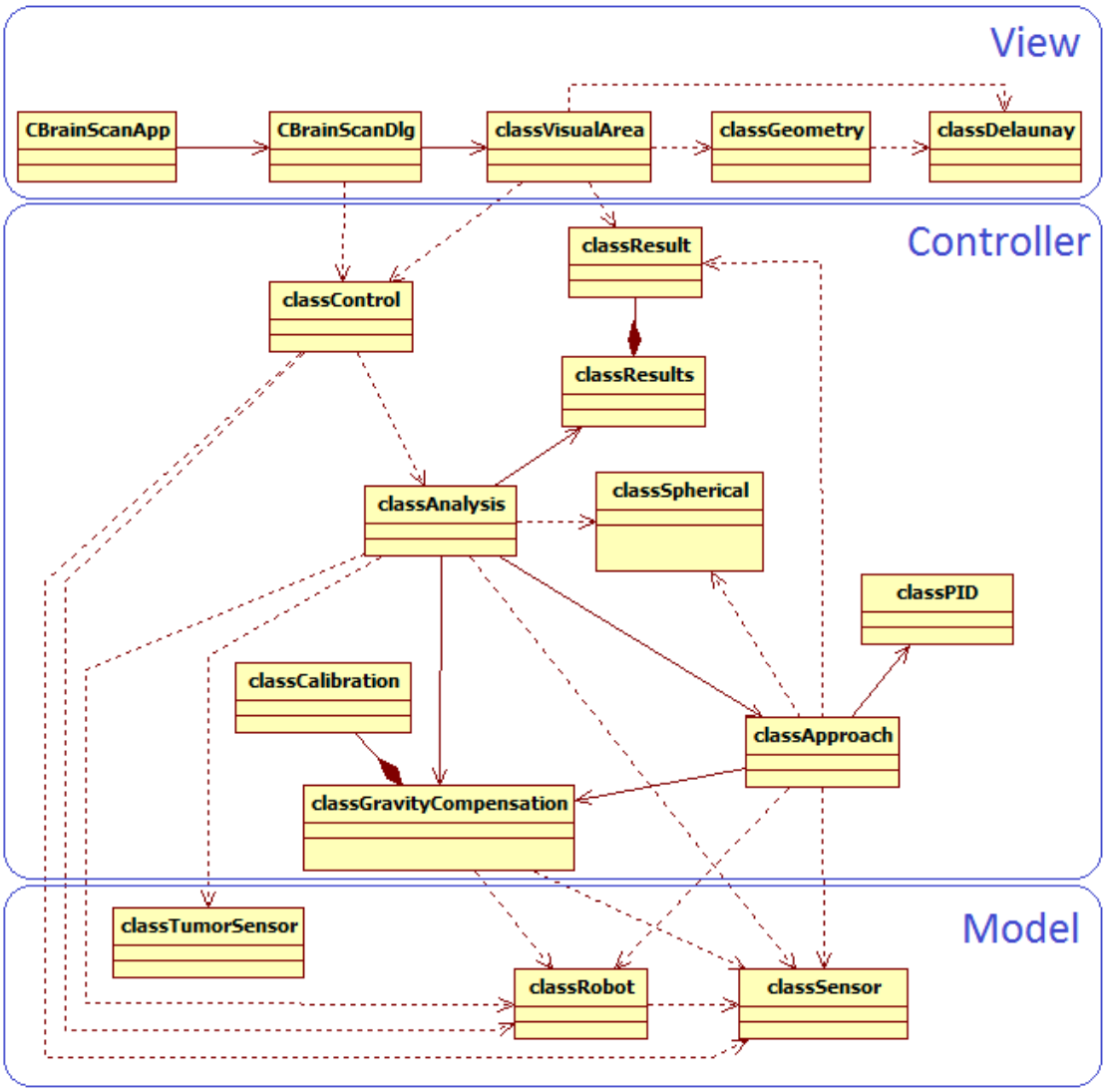


Figure 55: Class diagram including the Model-View-Controller pattern division.

- **CBrainScanApp**: main MFC class.
- **CBrainScanDlg**: main window of the application (GUI)
- **classVisualArea**: component of the main window where 2D and 3D graphics are displayed. It accepts mouse dragging.
- **classGeometry**: loads an .obj graphic file and converts it to a vector of points. It also includes data structures to handle points and triangles.

- **classDelaunay**: calculates a triangulation for a given set of points using Delaunay incremental algorithm. The output information can be displayed by a meshgrid viewer.

- **classControl**: this static class acts as a controller for all the scanning, calibrating and testing processes.

- **classResult**: stores information such as force, position and simulated tumor measure for a scanned brain surface point.

- **classResults**: manages a set of results.

- **classAnalysis**: includes or manages all developed methods and tests. It is implemented as a thread.

- **classSpherical**: contains all necessary elements to work with spherical coordinates.

- **classGravityCompensation**: calibrates the robot and solves the gravity compensation problem.

- **classCalibration**: stores information about a calibration step position.

- **classApproach**: includes all related methods to the approach process.

- **classPID**: software specific implementation of a Proportional-Integral-Derivative controller.

- **classTumorSensor**: static class which simulates the tumor sensor tool.

- **classRobot**: static class which acts as interface for the robot operations.

- **classSensor**: static class which acts as interface for the force/torque sensor operations.

4.5 Sequence structure

The execution sequence for a given functionality such as scan, calibrate or test, follows a similar pattern. The execution order begins in the window dialog after clicking a button. Then the order is passed to the control interface, which configures and launches the given process in the Analysis class. The process will not be launched if another process is currently using the robot. If the robot is available, Analysis class asynchronously starts the thread to allow another compatible operations while robot is working. Running function instanciates required classes to accomplish the task. The sequence is described in figure 56.

Each different process functionality has an unique identifier. A complete and detailed index of the available operations and their identifiers are included in classControl class.

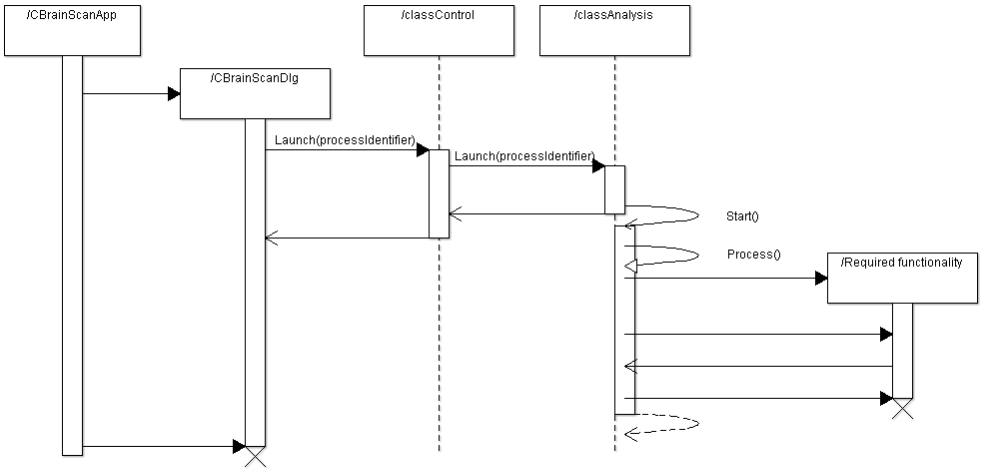


Figure 56: Generic process sequence diagram for a given functionality execution.

Diagram in figure 57 shows how scan parameters are updated before launching scan process. Scan process and calibration requires a previous parameter setting.

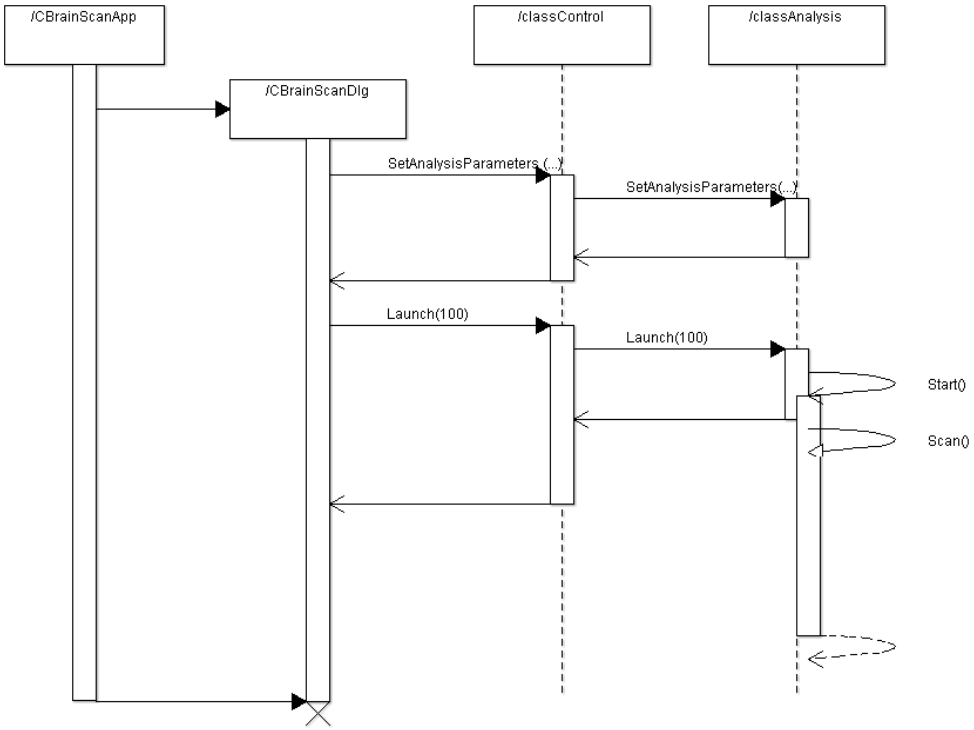


Figure 57: Scan process sequence diagram.

4.6 Graphical user interface functional description

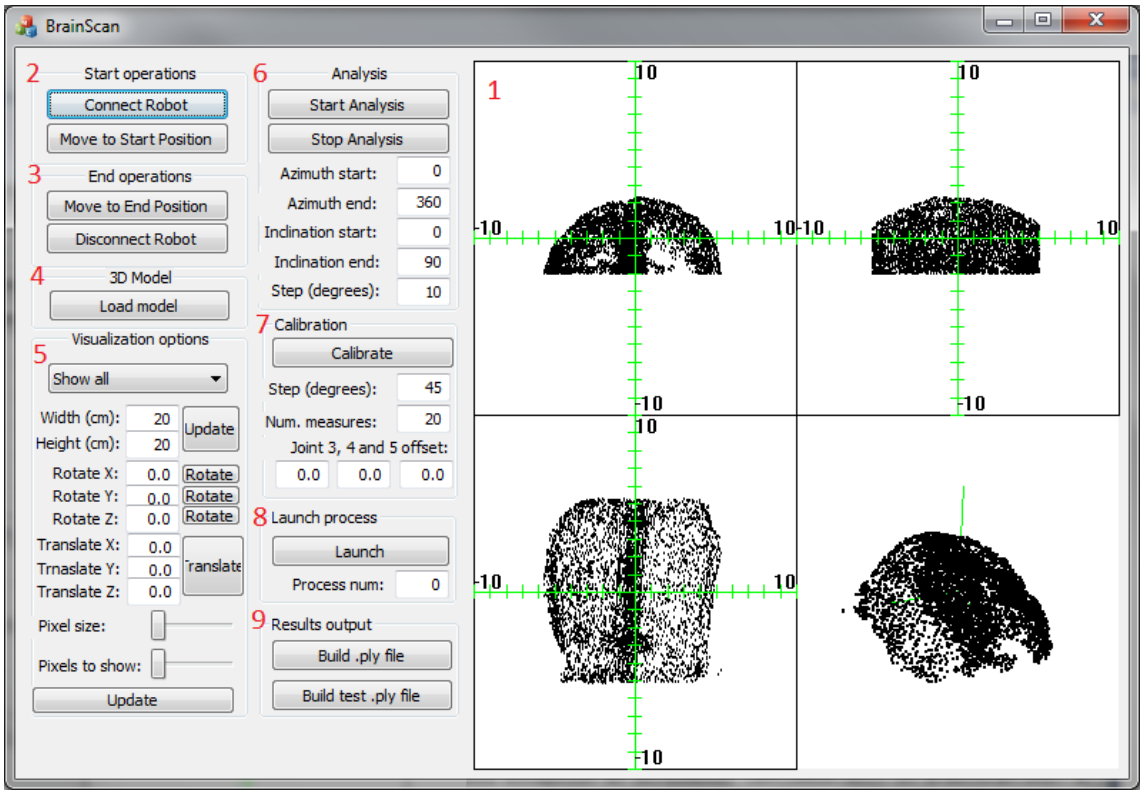


Figure 58: Graphic user interface.

The graphic user interface was build following the Single Document Interface. Everything is placed inside only one window. This window is divided in two parts. In the left part user operations and options are listed and in the right part the visual representation of the data is shown.

1. Visual representation area: it shows the XY projection, XZ, YZ, a isometric 3D view or all of them at the same time. It is possible to rotate the tridimensional view by dragging the mouse.

2. Start operations: first the robot must be connected with Connect button. Then the robot can be moved to the start position.

3. End operations: it allows to place the robot in the shutdown position and also to disconnect it.

4. 3D Model: it loads the file "brain.obj" contained in the application directory. Loading function is included in classGeometry.

5. Visualization options: Model and scanned data points can be rotated and translated in the space. It is also possible to show only some points of the model and configure their size on screen.

6. Analysis: scan process can be launched and configured from here. Azimuth and inclination allows to scan a given area of the brain surface by using spherical coordinates and a step.
7. Calibration: performs a calibration for gravity compensation. Step degrees, the number of measures for each step and joint offsets are configurable.
8. Launch process: it allows to launch a test or a functionality by its code. A code list is provided in the appendix section 7.4.
9. Results output: it builds a .ply file containing the scanned meshgrid information or example information.

4.7 Result tumor map

Result tumor map can be displayed in the application window or stored as a .ply file [17]. To build the .ply file, a meshgrid is created from the scanned points by using Delaunay triangulation incremental algorithm [18, 19]. Each point stores its position in space as well as an RGB value which simulates tumor mapping. This algorithm was implemented in the classDelaunay class. Figure 83 in the last chapter shows the visual representation obtained.

5 Results and discussion

5.1 Sensor positioning values calculation

5.1.1 Initial rotation angle

Through method 3.1.1, it is calculated that correct orientation is obtained with an initial Z rotation of 46 degrees. As it is represented in figure 59 and in detail in 60, for this rotation angle, Z force is at its lowest value and X and Y forces are null. In the graphic it is visible how X and Z are displaced 90 degrees. Y force value is always a little bit under zero due to joint alignment errors presented in section 3.6.

This test also show how center orientation changes if the sensor is slighty rotated by a human. In this case the test is repeated, but due to manipulation now correct initial Z rotation is 45.5 degrees. This problem is difficult to solve by software because fast robot movements could also slighty rotate the force sensor. Changing the fastening system seems to be the best option.

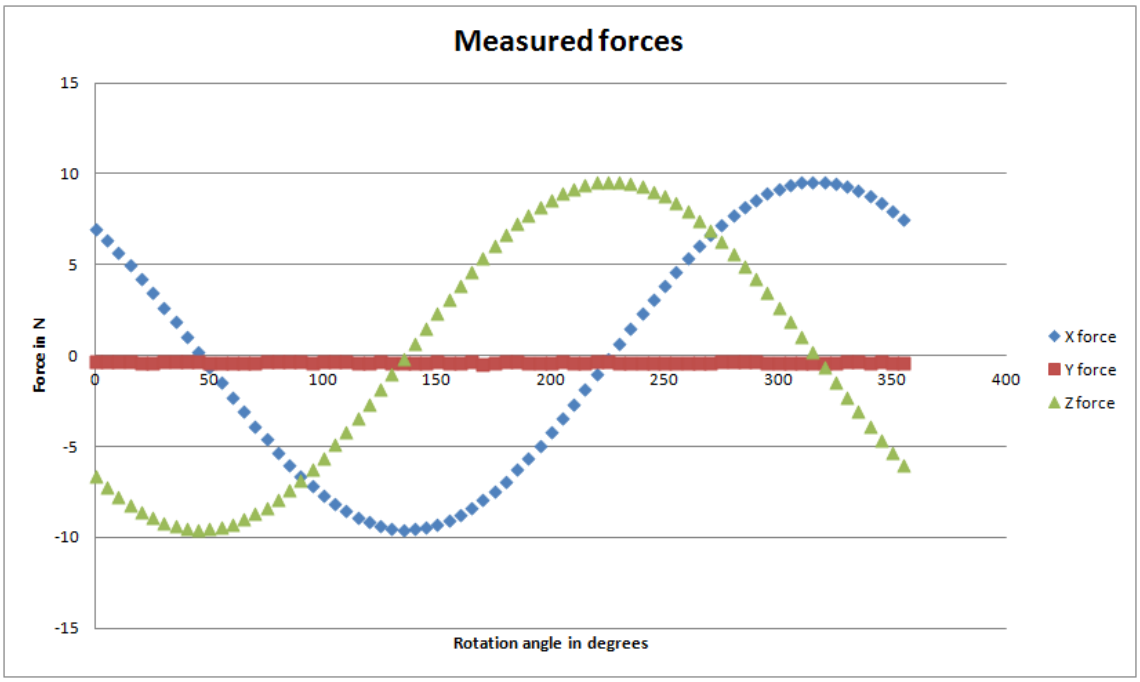


Figure 59: Measured forces evolution through force sensor initial Z rotation.

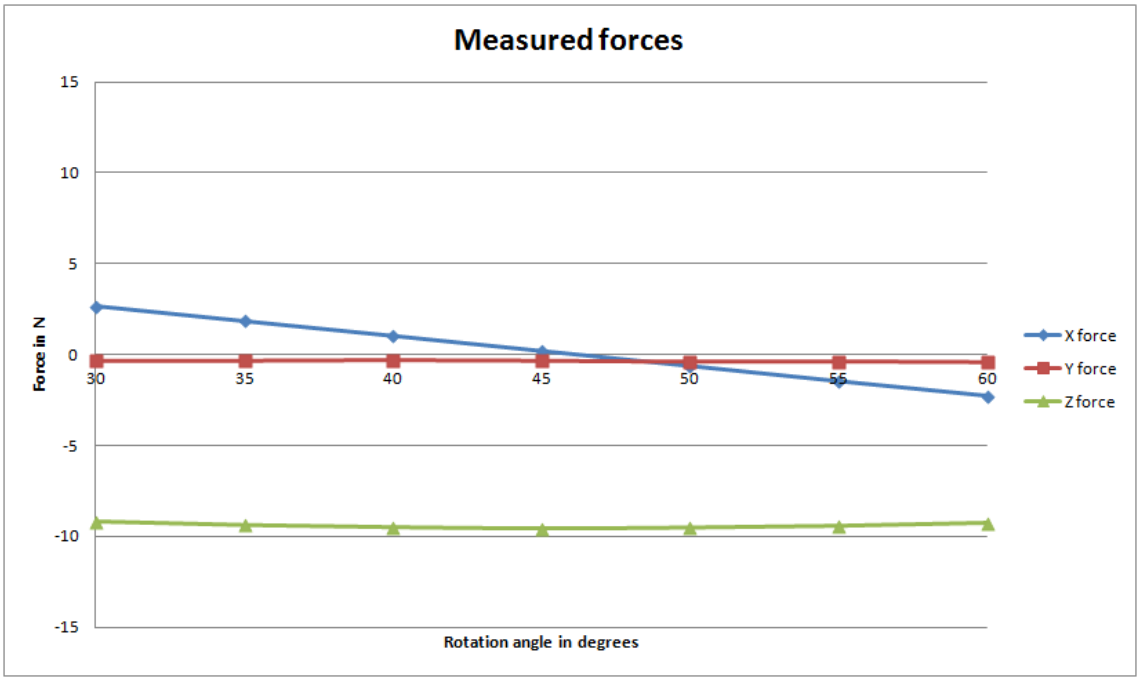


Figure 60: Measured forces in detail. Correct alignment correspond to the rotation angle which makes X force null.

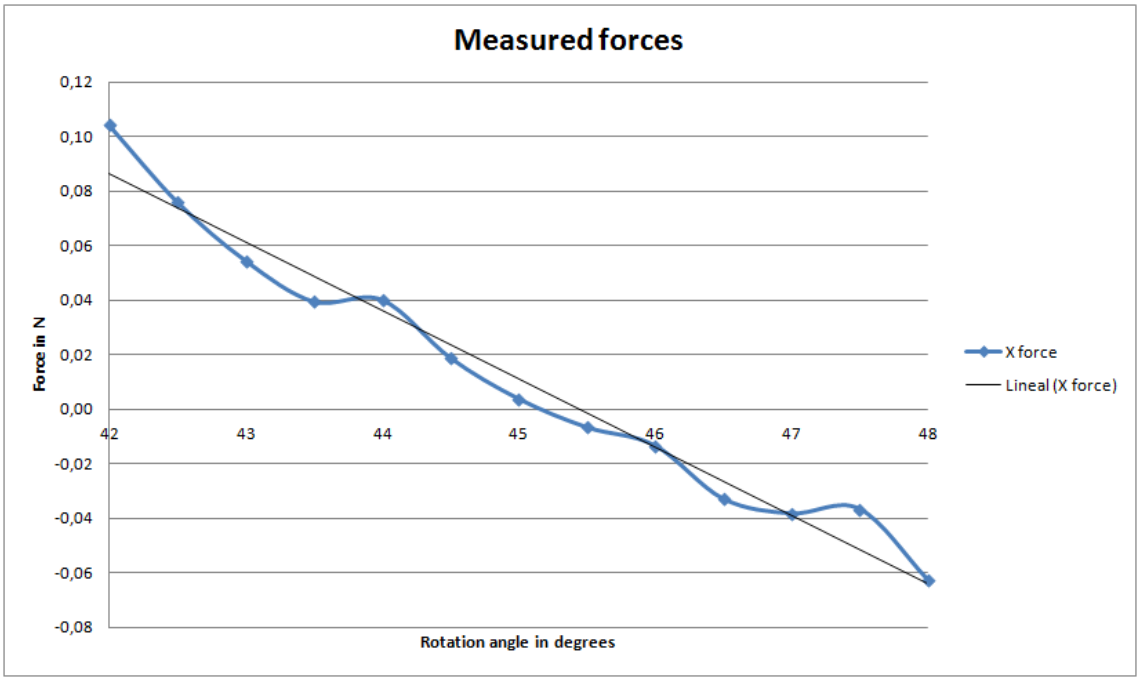


Figure 61: Result of measured forces in detail. This graphic also represents the difficulty to obtain a precise correct rotation value due to sensor resolution limitation. In this case correct rotation value will be 45.5 $\pm$0.2 degrees.

5.1.2 Initial Z translation distance

To calculate the distance between original force sensor center and tool, translation distance is modified through an iterative process. If measurement center orientation were unknown, it would be valid to find the least root mean square of the three torques. If it is orientated with two axes perpendicular to the translation line, it is only necessary to find which distance cause a null torque in one axis. In this case, in the first approximation, measures are taken each 10 mm. In the second, each 5 mm and in the third each 1 mm. Results are detailed in figure 62. With this method it is possible to experimentally obtain an accurate value.

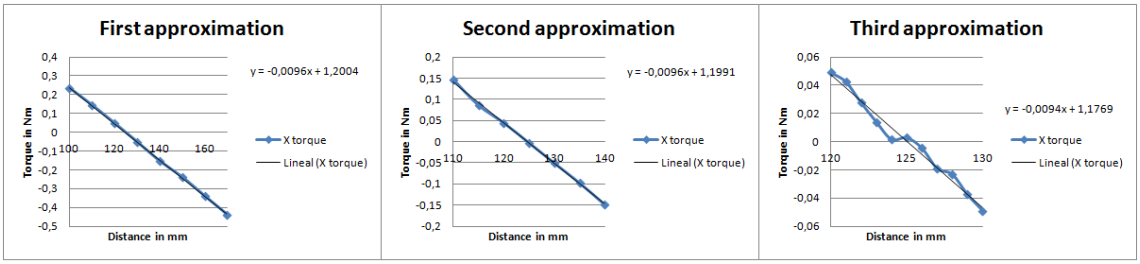


Figure 62: Results of changing distance between sensor center and tool. Through linear regression is possible to determine that correct translation distance is 125 mm in the tree cases.

5.2 Force sensor performance

5.2.1 Sensor precision of static positions

This test was intented to determine which is the precision of the robot hand in a static position. Robot hand didn't move during the test. Sensor precision was obtained by using one thousand measures for each different filter. As filter number increases, standard deviation decreases as it was expected because a higher filter number indicates that more measures are going to be internally taken into account by the sensor software to calculate the measured value. From this test it can be concluded that a high filter increases precision. But it also decreases response time of the sensor as it will be described in next section. This test also shows that results are better than the sensor resolution values provided by the manufacturer [7]. It's important to note that current Y axis corresponds to the original Z sensor axis of the datasheet.

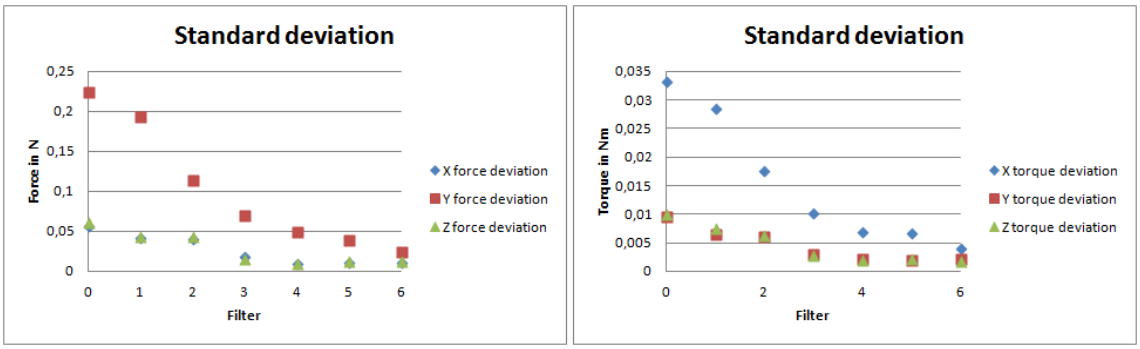


Figure 63: Standard deviation of 1000 measures for each different filter. Sensor precision increases by using a higher filter but as it is described in section 5.2.3, response decreases.

5.2.2 Sensor precision of dynamic positions

This test reproduces a vertical movement of the robot hand to analyze if precision decreases due to inertia force. The robot hand keeps vertically moving up and down at a constant 10% of the maximum speed while measures are being taken. The last 100 measures are stored and presented in the graphics below. Despite the tool doesn't touch anything during this movement, inertia force introduces a deviation when robot changes its direction. The most important conclusion which is obained from this test is that the error introduced due to the inertia forces is reduced by applying a higher filter. Nonetheless increasing sensor precision will decrease sensor time response, as it is analyzed in section 5.2.3. Standard deviation evolution is represented in figure 64. The conclussion obtained from this test is that a higher filter number reduces the impact of force variations, and a low filter increases too much this impact. It is concluded that the best filter to be used in the approach process must be in the middle between these extremes.

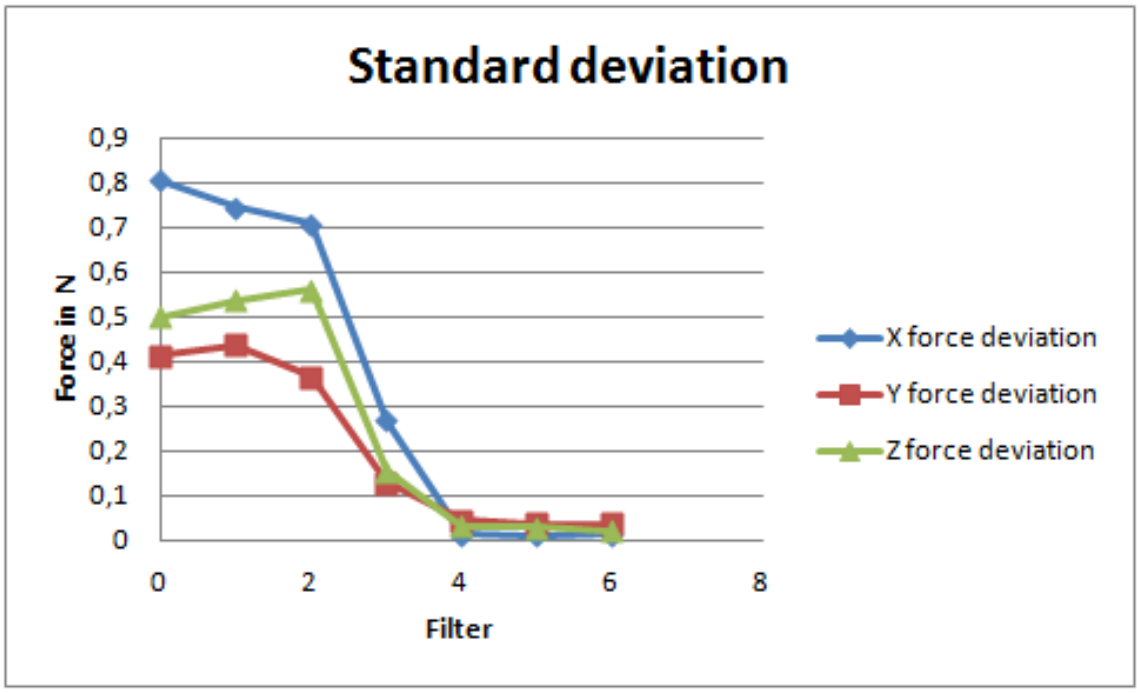


Figure 64: Standard deviation for each filter.

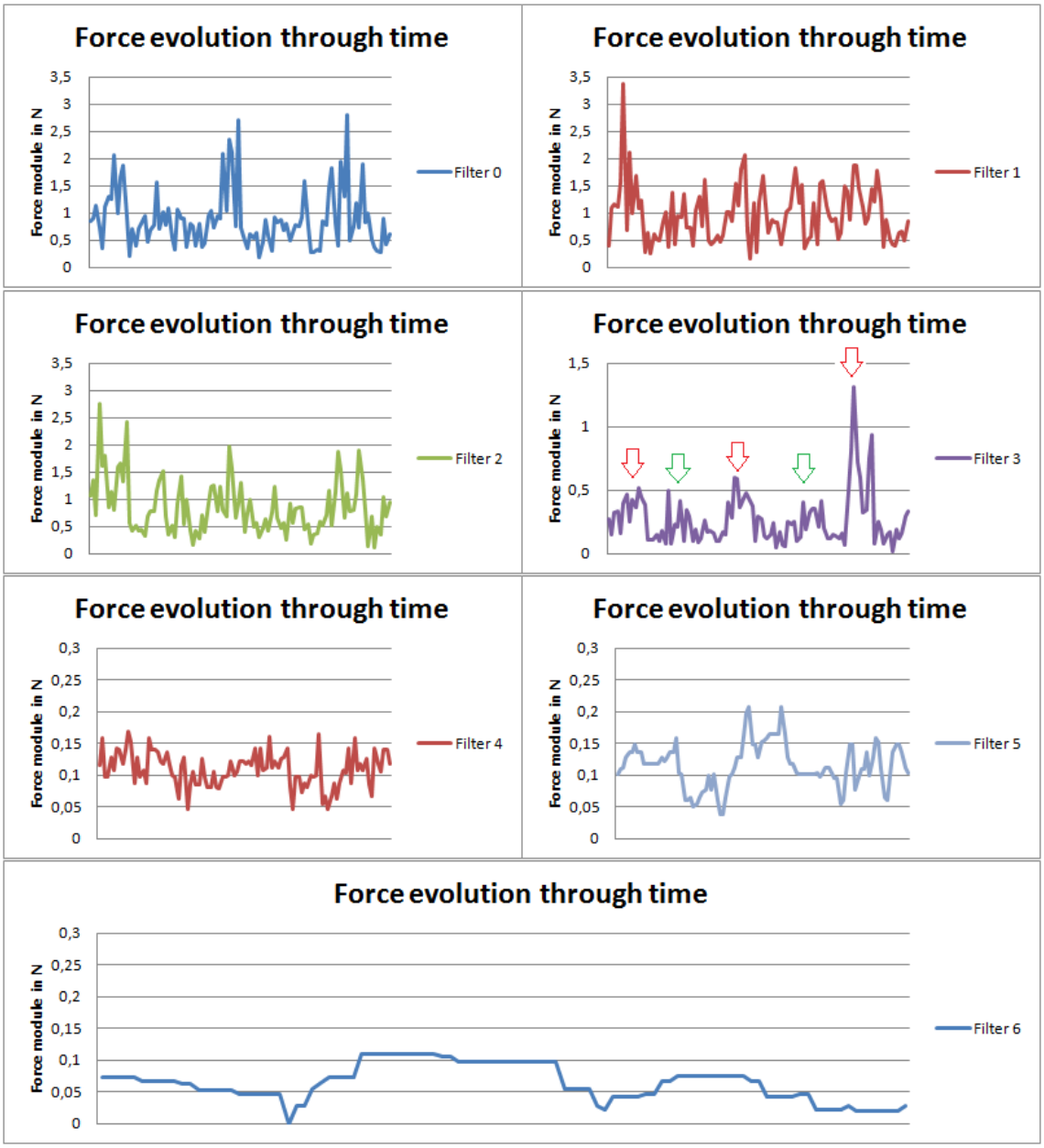


Figure 65: Last 100 buffered force module measures for each filter. Change points of movement sense are indicated for filter 3: down peaks in red and top peaks in green. Filter 4 or higher reduce the impact of errors due to inertia forces.

5.2.3 Sensor response

Depending on the filter selected, sensor response is modified due to filter properties such as change in the number of measures which are internally used by the provided sensor library. It is important to know how filters modify sensor speed response to select the one who give best performance to accomplish thesis objectives.

In this test the tip of the tumor tool was placed over a piece of gel. At the beginning the the tip goes 1 cm down and read several times to obtain a mean force as reference value. Then each filter is tested by teaking measures during a continous fixed 1 cm up and down movement.

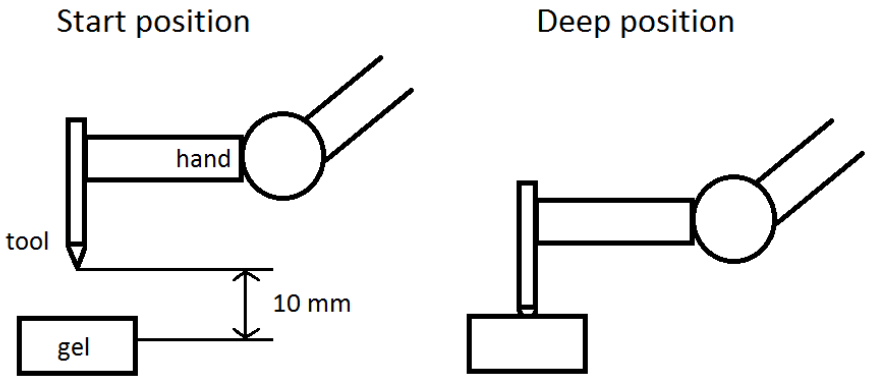


Figure 66: Schema of the start and deep position. Hand moves between these two positions.

Reference force module for figure 67 is 1.4 Newtons. This is the maximum force that sensor should read in the deepest position if inertia is neglected. Graphics show three complete movements. Filters from 0 to 2 are highly affected by inertia and unprecise. Their measures far exceed the maximum limit of 1.4 Newtons and oscillates too much. It was concluded that these filters were not suitable for the application. Filter 3 produces a slighty higher force measure than predicted but responds very fast to changes. Filter 4 doesn't reach 1.4 Newtons and seems not to be affected by inertia. Filter 5 in the graphic has a very similar behaviour to filter 4 . Filter 6 have a very poor and slow behaviour. From this test it can be concluded that filter 3, 4 and 5 are the most interesting options to be used in this thesis. To decide which one has the best behaviour in the approach process, they were tested by visual inspection while performing an approach to the surface. Filter 5 was discarted because it had a delay of apprimately half second. Filter 4 improved the stability of the surface contact but had a slighty slow response than filter 3. Filter 4 was decided to be the best option in this case.

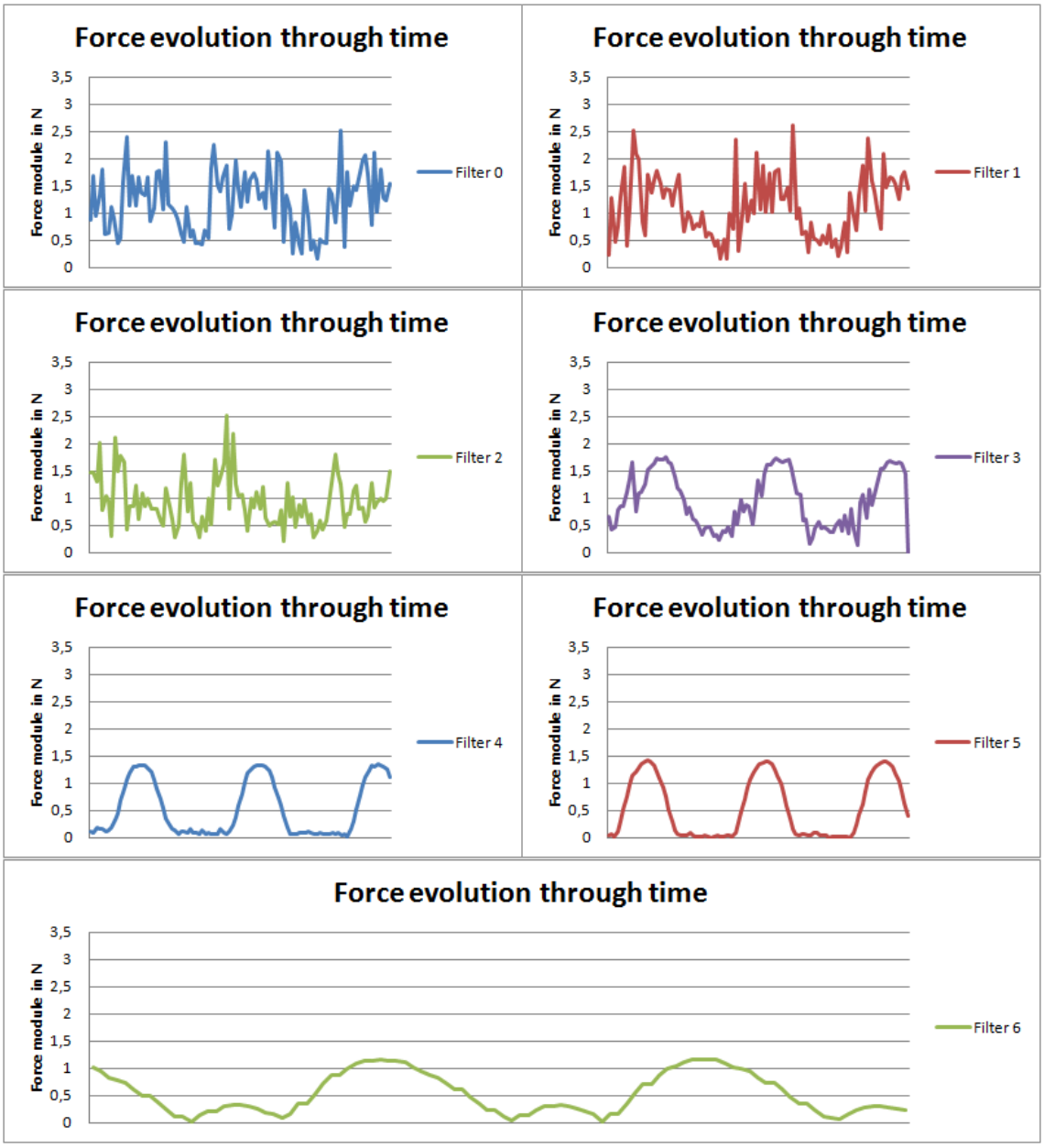


Figure 67: Results of continuously measuring for each filter. Each graphic is independent from the others and may not coincide in the timing so that overlay may conducto to wrong conclusions.

5.2.4 Sensor reset accuracy

This test is intented to determine if resetting the sensor modifies the read value, which without contact should be as near zero as possible. Sensor is reset one hundred times. For each reset five hundred measures are taken to obtain a final mean value.

Empirical results show that sometimes after resetting wrong values are read. Normally these wrong

values are obtained after resetting shortly after switching on the robot. But they are also obtained if two reset orders are near in time, so additional waiting time between resets is recommended.

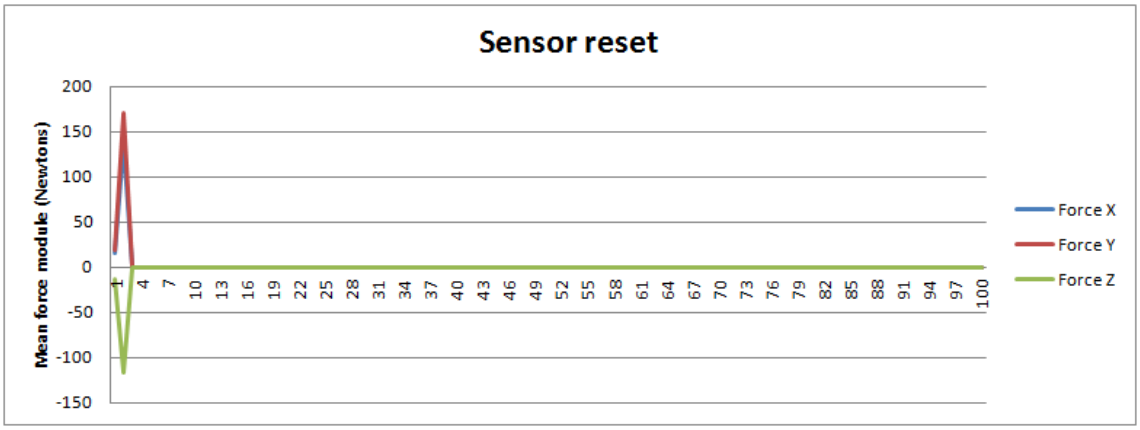


Figure 68: Mean force values read after each reset.

After discarding the first two wrong values, mean force and standard deviation for each axis of one hundred reset repetitions is presented in next table. Sensor standard deviation is less than 0.005 Newtons for five hundred measures, so that results indicate that finding an accurate reset is important to improve sensor performance. Standard deviation in this case can be interpreted as a indicator of the sensor accuracy.

	Force X	Force Y	Force Z
Mean (Newtons)	0.0011	0.0113	-0.0006
Standard deviation (Newtons)	0.0233	0.0661	0.0169

Table 3: Results of the sensor reset test.

The process which have been followed in the developed software to obtain an accurate reset is to repeat the reset until the read values module are below a limit. Mean time required to obtain a correct reset is analyzed through decreasing the force limit. The test was repeated twenty times for each different limit value. Experimentally, it is concluded that the number of resets required increases exponentially with the force limit. Furthermore, it is impossible to obtain an accuracy lower than sensor precision. Due to this it was decided to use a sensor reset limit of 0.04 Newtons, which requires approximately less than half a minute to complete reset process including waiting time.

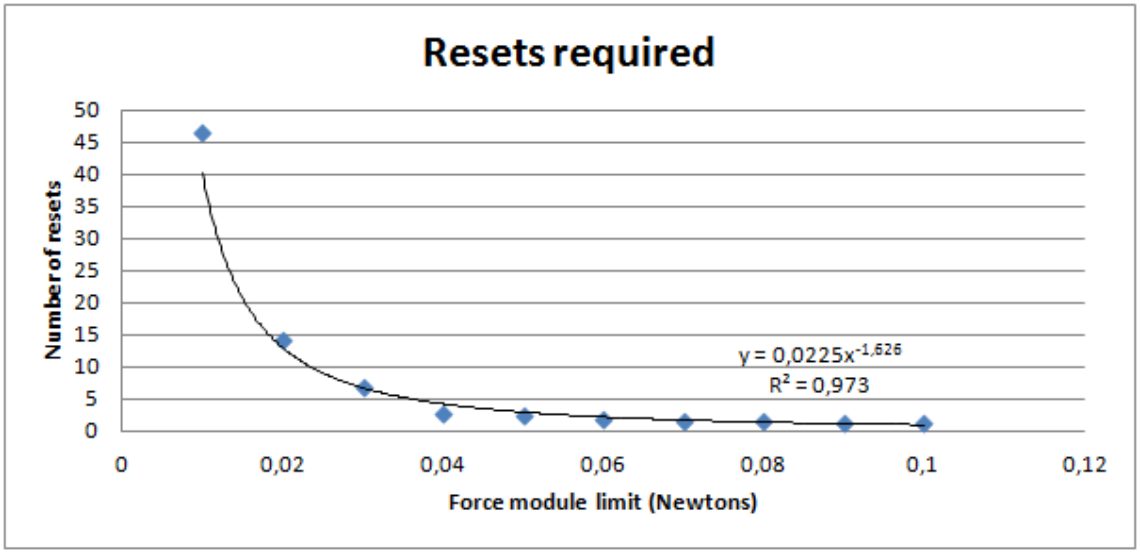


Figure 69: Number of resets required to obtain a force value under the given limit.

5.3 Gravity compensation

To evaluate the accuracy of the different compensation methods, several random tool orientations are tested. Robot hand is moved using Hemispherical Displacement described in section 3.2 to a random point over the brain surface. Then current force and torque are read and compensated by using developed methods. For each different measure it is calculated the root mean square of the X,Y and Z force components to obtain the module of the force. Detailed graphics, mean module force of all measures, standard deviation as well as some conclussions are presented.

5.3.1 Force/torque computation method

Force/torque computation obtains a 0.1 Newtons standard deviation but presents systematic errors, which makes this method difficult to use in the software application. Due to that only 6 different measures are taken in the calibration process, it is vitally important to perfectly align joint axis with the gravity vector, otherwise errors are obtained. In the results which are shown in this section, joints were aligned but systematic errors are still present. In a further investigation, systematic error could be eliminated by using mathematical tools such as adding a offset in the equations. These offsets could be empirically determinated by moving the robot to some positions where force results are known to be null in at least one axis.

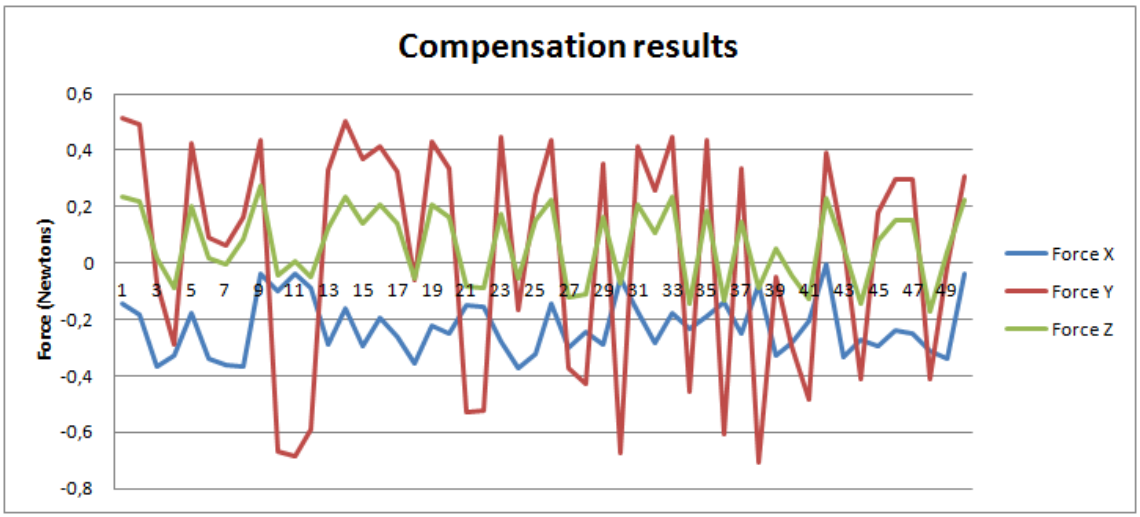


Figure 70: Compensation results for 50 random tool orientations.

Module mean (Newtons)	Standard deviation (Newtons)
0.49	0.10

Table 4: Results for 50 different positions.

5.3.2 Nearest stored measure method

This method is presented just as an example of how calibration data can be used to compensate the gravity because in order to have good performance it requires to take a huge amount of measures in the calibration stage. Time limitation makes impracticable this method. Results of this gravity compensation method shows that performance is very poor.

Calibration step (degrees)	Module mean (Newtons)	Standard deviation (Newtons)
45	4.00	2.39
90	8.43	4.64

Table 5: Results for 50 different positions.

5.3.3 Weighted mean value method

Empirical results of this method showed that this method has a low compensation precision. Due to this reason the method was rejected to be used in the approach process.

Calibration step (degrees)	Measures	Module mean (Newtons)	Standard deviation (Newtons)
45	2	3.59	2.97
45	4	7.28	4.49
45	8	9.79	3.80
90	2	8.20	5.64
90	4	11.13	5.22
90	8	12.56	3.23

Table 6: Results for 50 different positions. Measures indicate how many of the nearest measures are used to calculate the weighted mean value.

Results after modifying the method to consider the quadratic distance instead of just the linear distance. The quadratic distance is calculated by raising it to square.

Calibration step (degrees)	Measures	Module mean (Newtons)	Standard deviation (Newtons)
45	2	2.94	2.09
45	4	5.59	4.25
45	8	5.04	3.42
90	8	8.67	5.44

Table 7: Results for 50 different positions. Measures indicate how many of the nearest measures are used to calculate the weighted mean value.

5.3.4 Polynomial surface fitting method

In this section results by using the polynomial surface ftting method decribed in section 3.4.5 are detailed. To show how incrementing the number of measures in the calibration process increases the method performance, two different steps were taken. In the first case step is taken each 90 degrees, and in the second each 45.

Each 90 degrees

The next graphic shows force module calculated after compensation for 50 different positions. Increasing the polynomial curve order also increases performance until the order is the same that the number of points used on each dimension. In this case, a 90 degree step requires 4 points different points. Incrementing the polynomial order more than this value decreases the performance. Figure 72 represents in detail the mean module force of the 50 different positions for each polynomial order and its standard deviation. This effect is also observed using a 45 degree step.

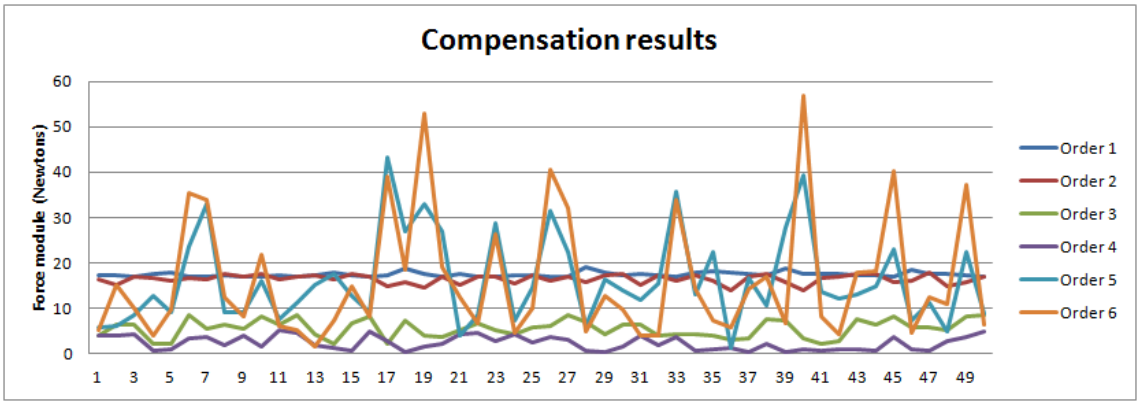


Figure 71: Compensation results for 50 different positions.

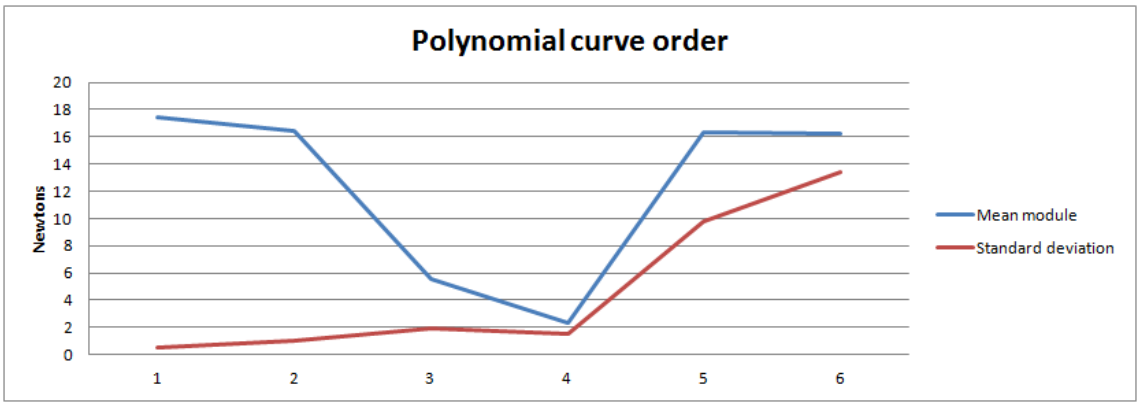


Figure 72: Compensation results for each polynomial order.

Fitting curve k order	Mean force module (Newtons)	Standard deviation (Newtons)
1	17.43	0.47
2	16.44	0.98
3	5.52	1.96
4	2.32	1.52
5	16.32	9.80
6	16.23	13.46

Table 8: Results of 50 random orientations.

The time required to calculate the compensation is less than one millisecond, so it is negligible and can be ignored. Results obtained with a 90 degree step are not good enough to be used in the application.

Each 45 degrees

Results obtained after testing process show that the best fit takes place for $k = 7$.

Fitting curve k order	Mean force module (Newtons)	Standard deviation (Newtons)
1	17.22	1.00
2	14.43	1.57
3	5.09	1.42
4	1.60	0.56
5	0.66	0.23
6	0.26	0.08
7	0.22	0.11
8	0.34	0.26
9	3.11	4.00
10	56.66	79.22

Table 9: Results of 50 different random orientations.

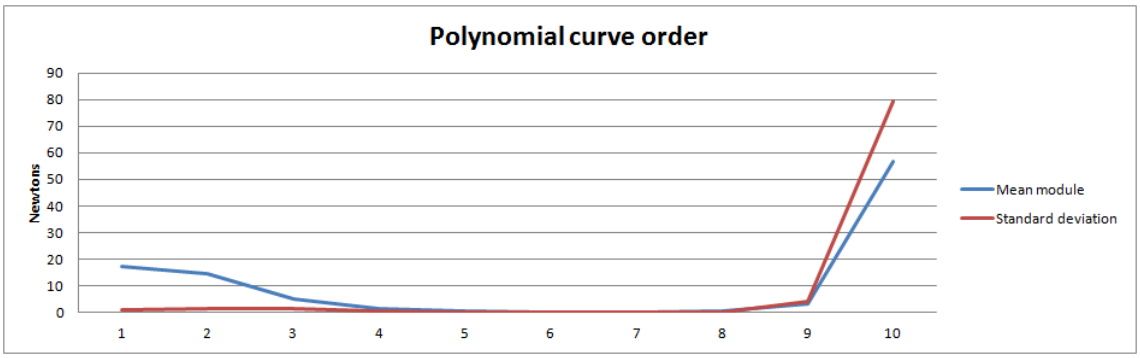


Figure 73: Mean force module and standard deviation evolution.

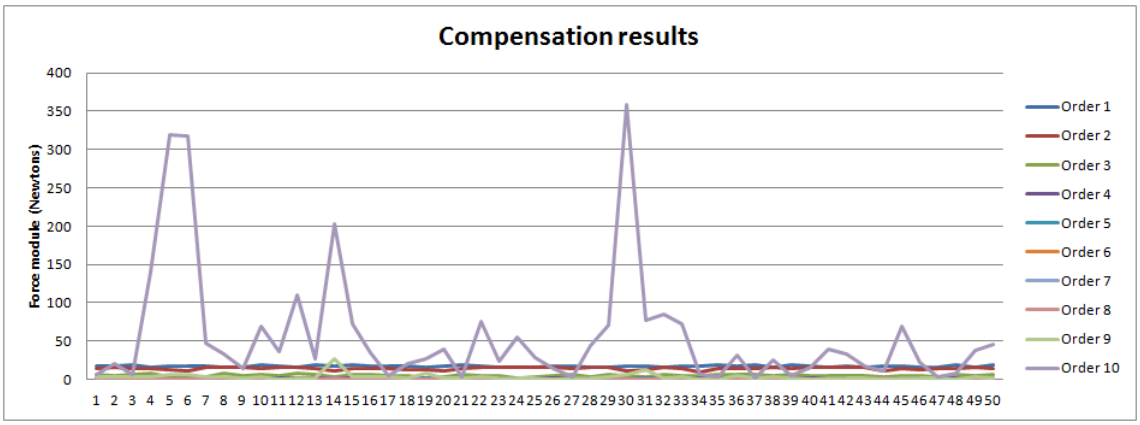


Figure 74: The 50 different positions detailed.

It is also interesting to analyze the required time to calculate the compensation value. Precalculation of Z_2 Euler rotation angle needs between 16 and 32 milliseconds. Precision of the timing function Y and Z_1elimination time is always less than 7 milliseconds, which allows to use this method while the robot tool is being moved.

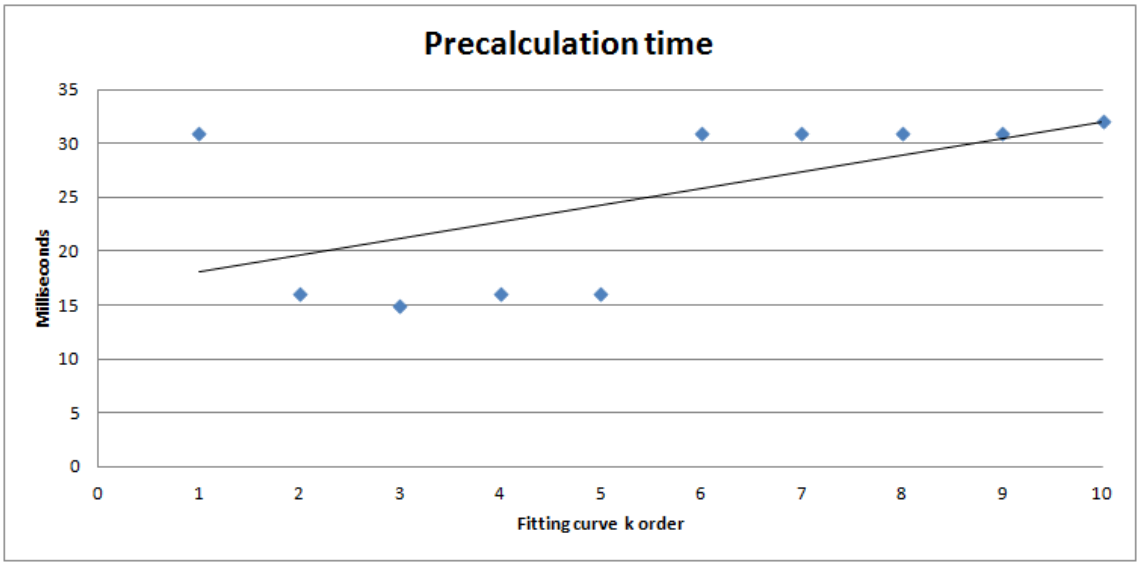


Figure 75: Precalculation time required when stored data is loaded.

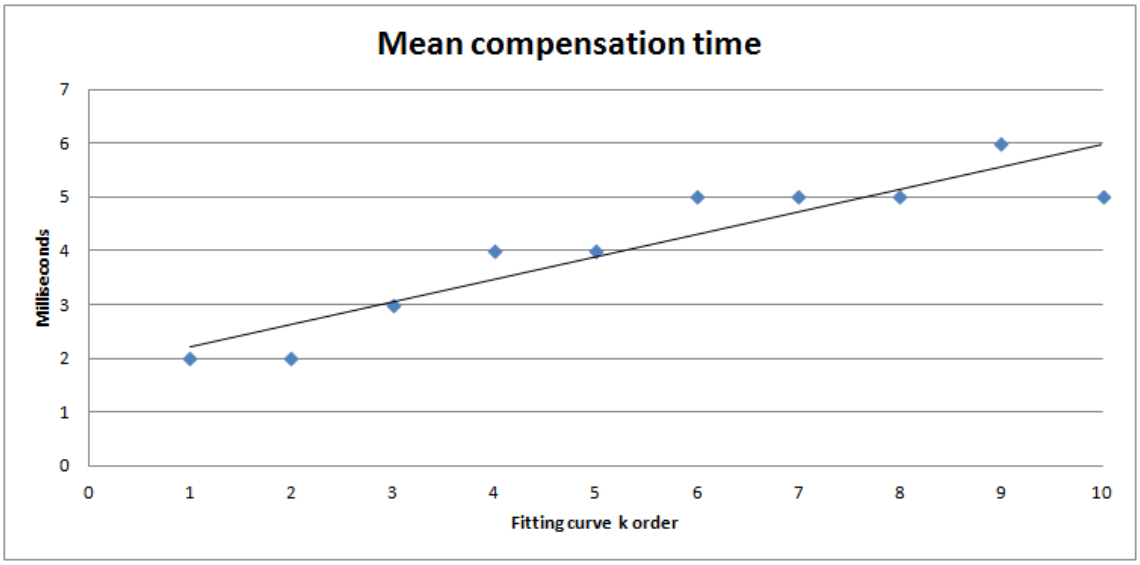


Figure 76: Average computation time for compensating force and torque.

In this case good enough results are obtained. Method is proved to be valid in the software application due to its low error. It is important to clarify that in this test errors are calculated using the module of the force, so that errors on each axis will be always less than this value.

5.3.5 Discussion

Figure 77 graphically shows the difference between the different methods. Theorically, force/torque computation method should arrive to the best results, but due to error in the joints alignment and

sensor precision fitting curve method has a better behaviour. To improve computation method it would be necessary to take into acoount joints alignment errors more deeply. Fitting curve algorithm is more complex but it is also more flexible allowing to handle irregular behaviours such as for example grabbing a glass of water. In this case case the hand losses mass when liquid is spilled.

Figure 77: Comparison between the best obtained results on each gravity compensation method. A lower mean module and standard deviation means a better compensation behaviour.

5.4 Force variation due to joints offset

In this section it is evaluated how small variations in the joints 3 and 4 changes the measured force. Sensor is reset using a null joint 3 and 4 offset, and then they are changed from -0.5 degrees to 0.5. As it is shown, small variations appreciably modify read force.

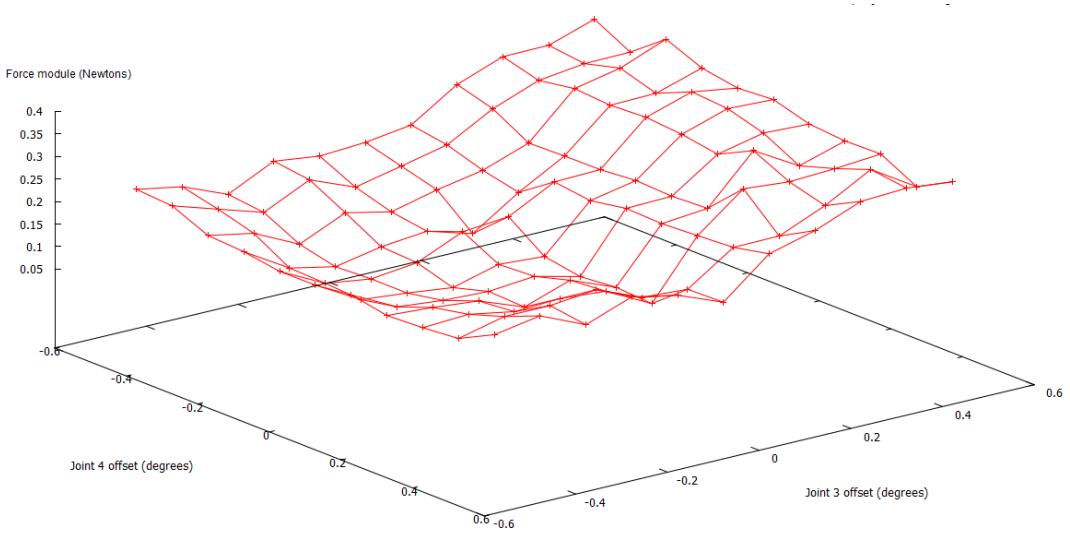


Figure 78: Force module measured.

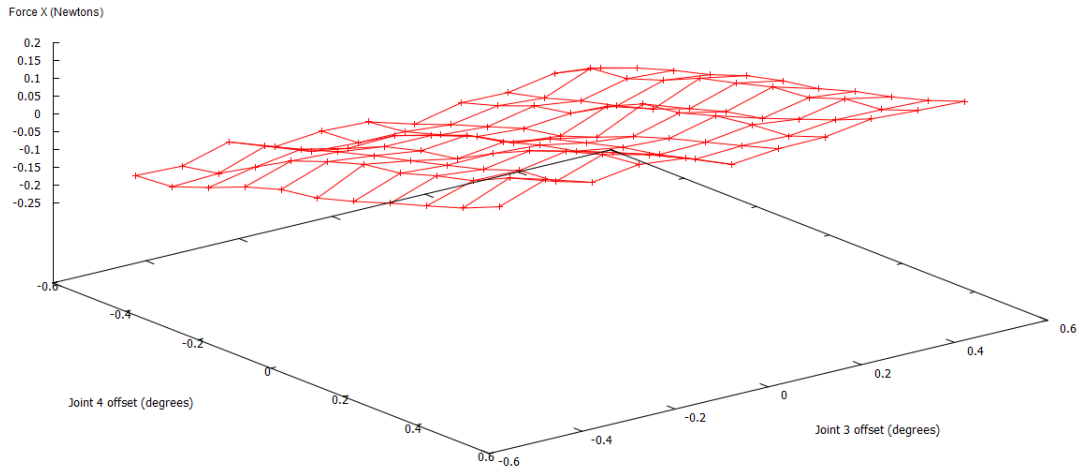


Figure 79: Force X measured.

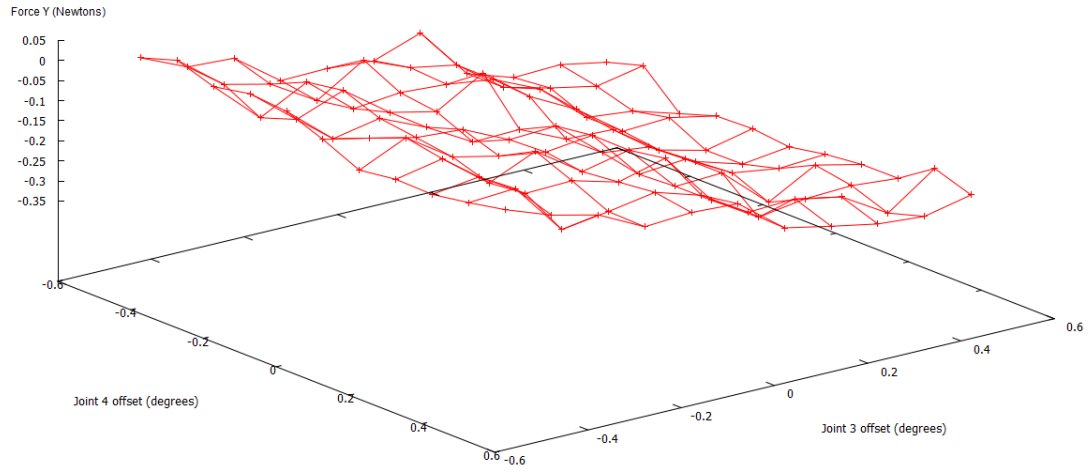


Figure 80: Force Y measured.

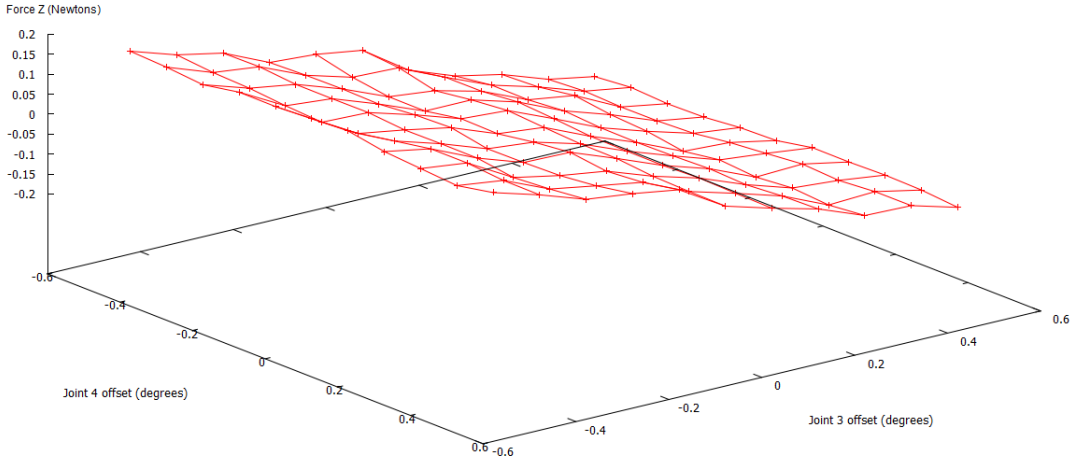


Figure 81: Force Z measured.

5.5 Approach results

5.5.1 PID Ziegler-Nichols tuning method

Before starting Ziegler-Nichols tuning method, target force was set as 0.3 Newtons, threshold was deactivated by setting a null value, number of maximum alter increments were configured to 4 and force module was set to be calculated by taking two measures and obtaining the mean value. Then, beginning with a proportional constant of 0.1 Newtons and null derivative and integral constants,

its value was progressively increased using a 0.1 Newtons step. The results obtained are described in the following table:

Kp	Comments	Period (seconds)
0.4	Constant oscillations	1.6
0.5	Constant oscillations, approximately 1 cm	2.0
0.6	Slighty irregular oscillations	2.5
0.7	Irregular oscillations	2.8

Table 10: Ziegler-Nichols ultimate gain selection.

Next figure shows how oscillation period proportionally increases with Kp:

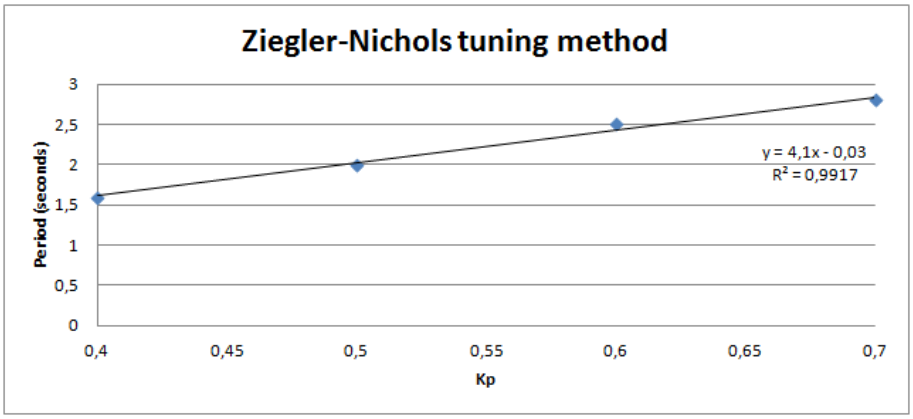


Figure 82: Period of the oscillation for different Kp values.

Considering $K_u = 0.5$ as the last gain with oscillates with constant amplitude, proportional constant are calculed as $K_p = 0.6 \cdot K_u$, integral as $K_i = 2 \cdot K_p/T_u$ and derivative as $K_p = K_p \cdot T_u/8$:

Constant	Value
Kp	0.3
Ki	0.3
Kd	0.075

Table 11: Preliminar Ziegler-Nichols results.

Integral constant value is very high to this application, so it is manually set to zero. This is necessary in order to avoid integral windup and loss of control of the approach process. Kp and Kd results are reasonably suitable and a good starting point for battery test and manual tuning. Only millimetric oscillations can be identified by visual inspection with a static brain model position and speed response is fast enough to follow surface displacements.

5.5.2 PID battery test

PID battery test was repetated for the three different tip diameters: 2 mm, 3 mm and 4 mm. Force was fixed to 0.3 Newtons, threshold to 0.1 and stored measures to be used in the integral component to 10. Gravity compensation during the battery tests was activated. Mean force was obtained by using 2 and 3 measures. Kp varied from 0.2 to 0.6, Kd from 0 to 0.075 and Ki from 0 to 0.01. Complete results are detailed in the charts at the appendix 7.1, 7.2 and 7.3 of the document.

Due to the high number of combinations which can be evaluated and the twenty seconds required to each one, battery test can last some hours if possibilities are not reduced. It is important to focus only in a few different parameters at the same time. Each parameter affects different aspects such as speed response, stability or overshoot. Parameters are also interrelated so that modifying one parameter could require to correct another one. For example using more measures to obtain the mean value will decrease speed response. The conclussions obtained from the battery test and the manual tuning are detailed in next section.

5.5.3 PID conclusions

The main conclussion is that it is complex to perfectly set PID parameters because of different difficulties. The first difficulty is the number of different parameters. They are interrelated, so that a modification of the Alter mode configuration or the force sensor properties changes approach performance. Alter maximum increments, sensor filter, number of measures to obtain the mean are some examples of this kind of parameters. Due to the time requirements it is impossible to test all possible values, so that this ground configuration must be set up at the beginning. The second difficulty is the inherent sensor error, which makes it difficult to stabilize the tool on the surface even if it is not being moved. The third difficulty is that if a better stabilization is desired, then speed response gets worse because basically increasing stabilization involves taking more measures and decreasing Kp.

Increasing Kp also increases speed response, but decreases stabilization. If Kp is bigger than 0.4 for a 0.3 Newton target, process began to fail due to the amplification of measurement errors. Kp = 0.3 is a good average between speed response and stabilization. Increasing Kd allows to respond faster to force variations, but also amplifies measurement error. Kd = 0.025 gives an acceptable behaviour. Ki must be very small to be useful in the approaching process, but in general can be ignored because it doesn't significantly improve performance. Increasing the number of measures which is used to obtain the mean read force value increases stabilization, but decreases speed response. Recommended value is 2 or 3 depending which is more desirable for each specific case. Changing between one tip and other type doesn't modify too much the obtained results so that tecommended parameters are valid for the three tips. If a better response is desired, Kd can be slighty incremented until 0.075.

Constants	Recommended value
Kp	0.3
Ki	0.0
Kd	0.025

Table 12: Recommended PID values.

6 Discussion and conclusions

In this section a discussion about the final conclusion is presented as well as introducing new ideas for further lines of investigation. Here there are also described difficulties which appeared during the thesis development. This section also evaluates the developed work and results in the current medical field context.

6.1 Final discussion

This thesis is included in the robotic surgery field, which is a relatively new topic inside medicine. Furthermore, one of the most interesting and challenging aspects of this thesis is that it introduces a new line of investigation. Due to the large number and diversity of aspects covered by the thesis, parallel investigation in combination with the Spiral Software Development Model was proved to be a good approach. Each iteration of the model facilitates to generate new ideas to investigate. Time limitations of a Master thesis also made this model the best option because it is more important to progress from global perspective rather than delve too deeply into one field while neglecting the others. Despite of this model, it was difficult to completely eliminate bottlenecks, but they were considerably reduced.

Although some obstacles with different degree of difficulty were found, gravity compensation was the most important bottleneck because it masked some error sources such as for example the joints alignment. It also forced to use a robot motion mode limitated by sensor resetting until gravity compensation was fully functional and precise. Force/torque sensor resolution added an additional difficulty.

The main conclusion obtained from this thesis is that it is possible to use an automated guided robotic system in neurosurgical applications and that it is possible to obtain acceptable results even with generic industrial equipment. This thesis proves that a six degrees of freedom robot in combination with force sensor feedback allows to implement a very high precise control. It was also concluded that it is possible to build a fully functional system which can work under an unknown brain geometry. The viability of a safe robotic brain surface analysis is demonstrated by this thesis, as it is shown by the example resulting tumor map of figure 83.

One of the first specific conclusions obtained from the thesis is that the sensor resolution and speed response is critical in order to improve the global performance. Despite robot movement speed is very high and accurate, contact maintaining with a moving brain surface is limited by sensor response. Sensor resolution affects the stability of surface contacts, the gravity compensation and some methods such as obtaining the perpendicular position to the surface . The limitations of the sensor response were partially solved with the use of statistical treatment and curve fitting in the gravity calibration. In spite of response and resolution, a good behaviour and performance was accomplished for contact forces of 0.3 Newtons or higher. One related conclusion is that it is difficult to predict and eliminate error sources when working in a low force range in this kind of applications. The hand of the robot weights several kilograms so that slight errors in joints alignment or in the force sensor physical position have an important impact on the measurments and can produce errors about more than 0.5 Newtons. Another conclusion is that by restricting the possible robot configuration space is possible to increase safety and avoid singularities. Finally it is important to note that it was necessary to ellaborate a large number of small experiments in order to advance in the investigation.

Despite the developed work, due to the system complexity this thesis represents the beginning of a new investigation line. The different aspects which could be investigated are numerous and varied. Some new ideas are introduced in section 6.3.

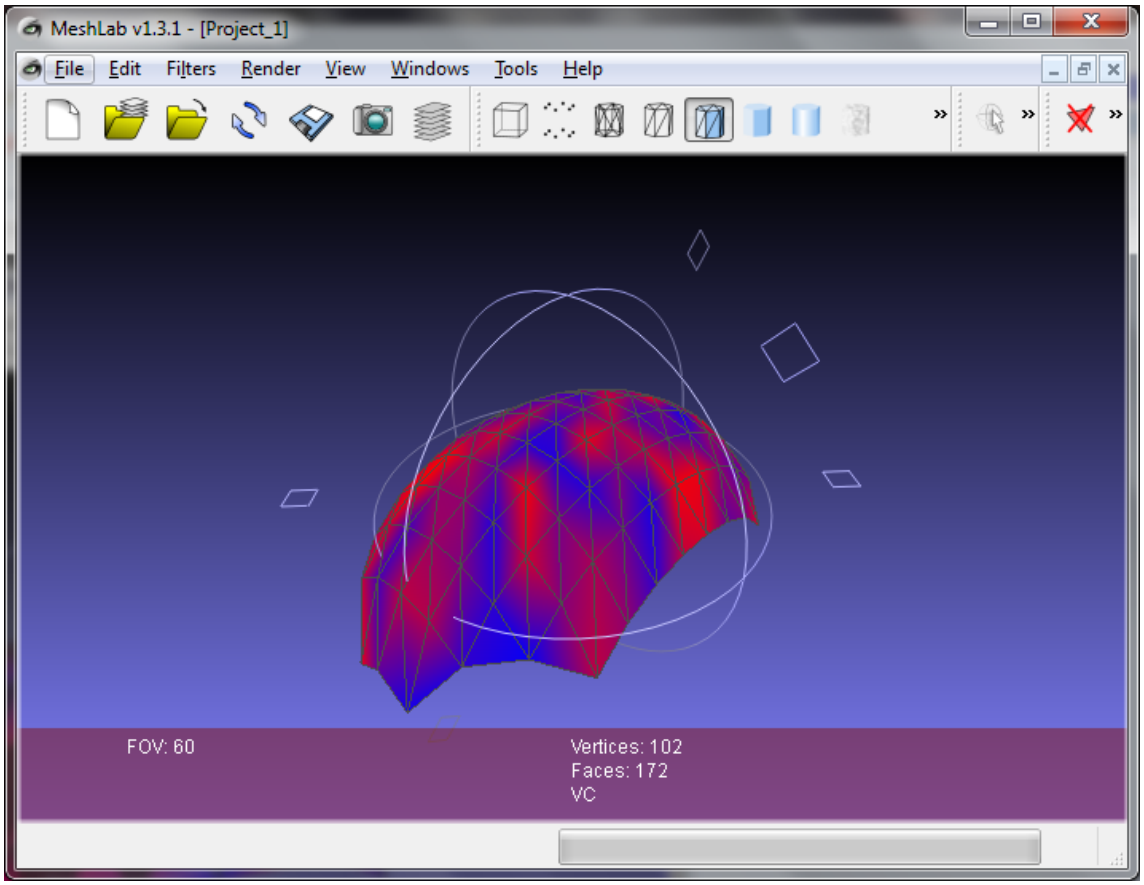


Figure 83: Example three-dimensional representation of the result tumor map displayed with MeshLab [20]. Zones in red correspond to tumorous tissue. Surface in blue is healthy tissue.

6.2 Difficulties found

In this section the most relevant difficulties found through analysis or discovered after are listed in order of appearance. Some difficulties included in this section are related to the nature of some algorithm and were previously analyzed through brainstorming before the implementation phase. Others were discovered through anomalous system behaviour.

6.2.1 Collision free path

Finding a collision free path was the first problem to face. This also involves to find a correct brain placement which makes possible the hand of the robot to arrive to every point of the brain surface. It was necessary to completely understand the hand trajectories and analyse joint turnings to limit

the movement to restricted paths. It was concluded that the best brain position was directly under the robot.

Orientation of the robot hand on each surface point is key in order not to collide with the robot links and also not exceed rotational joint limits. Because of this, orientation is fixed on each point of the Displacement Hemisphere which was described in section 2.1.6. Moving from one position to another may require to totally change robot joints values. This is very dangerous if it is not controlled because it may entail a collision. It was thought that a division of the workspace in four quadrants will allow to solve this problem. Orientation of the hand is independently treated on each quadrant due to the rotational range of the joints.

6.2.2 Spherical coordinates

At the beginning of the thesis, positions over the brain surface were calculated by using cartesian coordinates. Although safe curve movements were accomplished, this forced to develop complex and unclear methods which were difficult to handle and extend. Then it was decided that to use a spherical coordinate system leads to elegant and simplified methods which were easy to understand and maintain.

6.2.3 Uncertainty of the position

When the position of the robot is read from the hardware, it has an inherent error which can be considered as negligible except in that case that the robot hand is being moved in a line between two quadrants. If position of the robot is read from hardware inherent error may cause to erroneously the software to determinate that the hand is in a wrong quadrant causing a hand orientation change and then a collision. This problem is very relevant and dangerous beacause the orientation of the hand is opposite between quadrants 1-2 and 3-4. As it is described in figure 84, areas near axis are called Uncertainty Boxes and they are the zones where is impossible to exactly know where the hand is located.

There are two different solutions two this problem. The first one is to determine in which quadrant the hand is placed by processing the joints values. The second is to assign a fixed quadrant for a spherical coordinate so that it is not necessary to read the position from hardware any more.

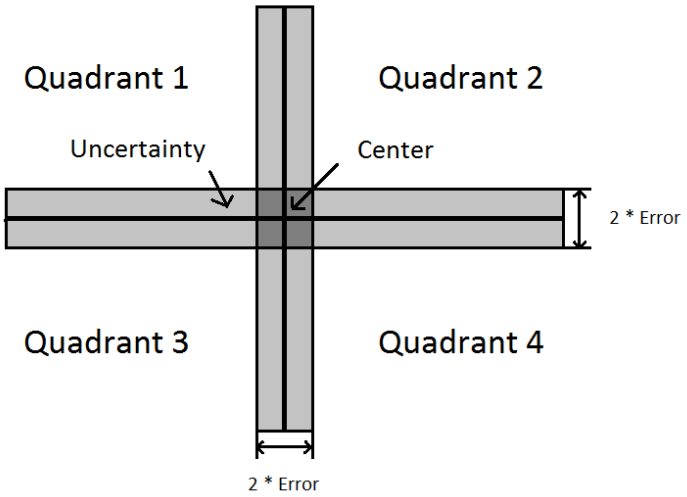


Figure 84: Uncertaint appears near the axis.

6.2.4 Force sensor precision and response

The thesis performance is limited by the force sensor resolution. The manufacturer gives a resolution value of 0.04 Newtons in the original sensor X and Y axes and a 0.2 Newtons in Z. The force range which is required to accomplish the thesis objectives is very near to this physical limit, so that despite of the increment in precision by taking several measures to obtain the mean force, it is impossible to go lower the resolution limits. Empirical results showed in section 5.2.1 how precision calculated as two times the standard deviation is very similar to the resolution limit.

To improve the quality of the measurements and the general performance of the approach methods, it would be necessary to replace the sensor with a more precise and faster one.

6.2.5 Force sensor coordinate system center and orientation

The default force sensor center position and orientation relative to the robot hand were unknown and it was needed to develop the tests proposed in 3.1.1 in order to determine the correct position and orientation. Another related difficulty is that force sensor can be physically slighty rotated by a human or theorically even with fast robot movements. This inconvenient is difficult to handle and decreases calibration precision.

6.2.6 Hand weight

Hand weight vector projections on the three axes change when moving the hand of the robot was a difficult inconvenient to be solved. Because of that sensor is reset in a fixed position, offsets of each axis are stored for the current position, so that they are not valid if orientation changes. It was a challange to develop methods which can compensate weight vector deviation with a good precision.

6.2.7 Joints alignment

Robot joints alignment have two different effects. The first one is related to the calibration stage. In this stage, the robot hand is moved to each calibration point by directly setting joint parameters. In this case, if the links of the joint 5 are not perfectly aligned, robot hand will arrive to slighty wrong positions, which decreases compensation precision. The second effect is that the hand is not aligned with the weight vector. This effect must be perfectly corrected if the mathematical computation method is used, but the fitting curves method will automatically solve this problem.

This was the most difficult error source to identify and also the most difficult to correct. Despite of the developed methods, it is very hard to perfectly align the joints so that errors will always be introduced.

6.3 Further investigation

In this sections new lines of investigation are proposed in order to improve the quality and performance of the process as well as its safety. Some of the new lines involve to replace current equipment and also using new hardware. One of the most important limitation of the current process is that all the approach process depends on the force sensor. If the force sensor fails, the patient brain may be damaged.

The first line of investigation includes developing or buying and testing new specific hardware. A lighter easily dislocable hand will improve safety. A faster force/torque sensor with higher resolution will have impact in the performance.

The second line of investigation is related to the robot movement. It will be interesting to use a different type of robot with more degrees of freedom to evaluate if another method to positionate the tool over the brain surface is possible instead of using the Displacement Hemisphere. This will permit to perform more natural movements with a more permisive rotational range, allowing the surgeon to have enough space to operate.

The third line involves using a vision system or other hardware tools which give more information to the process. For example the process could be combinated with an ultrasonic or laser distance sensor. This tool could be used to improve the approach performance and safety by including distance information in the approach algorithms. This information could be used together with the force measures as feedback for the PID. Safety of the process will also be increased because it would be possible to move back the tumor tool if it is going too deep or in case of a force sensor failure. Another interesting sensor is a 3D mapping system such as the Microsoft Kinect [21]. It could be used for calculating distances and also for collision detection.

The fourth line of investigation consist on developing new gravity compensation algorithms in combination with joints alignment correction. It would be interesting to obtain the gravity compensation values directly from the joints values.

The fifth proposed line of investigation is based on algorithmic improvements of the approach process. The tumor sensor approach performance can be increased by creating a prediction model of the brain surface displacements due to the heart pulse, which could be given in real time as input in combination with a 3D mapping obtained through computed tomography or a vision system.

A last line of investigation is developing new graphics displaying methods and improving the graphical user interface.

7 Appendix

7.1 PID battery test for the small tip

F Target force in Newtons

Kp PID proportional gain constant

Kd PID derivative gain constant

Ki PID integral gain constant

Int Number of last errors taken into account to calculate integral term

Thr Threshold in Newtons. Valid force range: [target - threshold, target + threshold]

Msr Number of measures taken to obtain the mean force value

C% Contact percentage

Std Standard deviation in Newtons

Osh Overshoot in Newtons

D Maximum depth reached in mm

Osc Number of oscillations

A Mean amplitude of the oscillations in mm

T Mean period of the oscillations in milliseconds

R Ranking value

F	Kp	Kd	Ki	Int	Thr	Msr		C%	Std	Osh	D	Osc	A	T	R
0.3	0.2	0.000	0.00	10	0.1	2		0.35	0.64	0.4	14.0	13.0	0.64	1350	0.59
0.3	0.2	0.000	0.00	10	0.1	3		0.98	0.07	0.2	12.9	3.5	0.18	5129	0.13
0.3	0.2	0.000	0.01	10	0.1	2		0.76	0.28	0.4	13.9	10.5	0.42	1881	0.34
0.3	0.2	0.000	0.01	10	0.1	3		0.53	1.23	0.4	14.0	17.0	0.49	1097	0.65
0.3	0.2	0.025	0.00	10	0.1	2		0.61	0.41	0.4	13.8	20.5	0.20	893	0.35
0.3	0.2	0.025	0.00	10	0.1	3		0.97	0.21	0.2	12.6	3.0	0.47	5800	0.22
0.3	0.2	0.025	0.01	10	0.1	2		0.38	1.05	0.5	14.2	19.0	0.67	1087	0.70
0.3	0.2	0.025	0.01	10	0.1	3		0.44	1.39	0.5	14.6	16.5	0.65	1139	0.79
0.3	0.2	0.050	0.00	10	0.1	2		0.66	0.59	0.5	14.3	16.0	0.37	1150	0.45
0.3	0.2	0.050	0.00	10	0.1	3		0.95	0.35	0.1	12.4	3.0	0.75	6483	0.32
0.3	0.2	0.050	0.01	10	0.1	2		0.36	0.69	0.5	14.5	20.0	0.48	920	0.58
0.3	0.2	0.050	0.01	10	0.1	3		0.42	0.92	0.5	14.1	18.0	0.61	1028	0.64
0.3	0.2	0.075	0.00	10	0.1	2		0.65	0.62	0.4	13.9	12.5	0.28	1224	0.41
0.3	0.2	0.075	0.00	10	0.1	3		0.41	0.76	0.5	14.2	15.5	0.55	1226	0.59
0.3	0.2	0.075	0.01	10	0.1	2		0.66	1.95	0.4	13.8	9.5	0.99	2000	0.92
0.3	0.2	0.075	0.01	10	0.1	3		0.38	0.96	0.6	15.1	16.5	0.92	1215	0.78
0.3	0.3	0.000	0.00	10	0.1	2		0.15	1.73	0.4	13.6	12.0	1.93	1696	1.22
0.3	0.3	0.000	0.00	10	0.1	3		0.41	1.75	0.5	14.3	15.5	1.02	1300	0.96

F	Kp	Kd	Ki	Int	Thr	Msr		C%	Std	Osh	D	Osc	A	T	R
0.3	0.3	0.000	0.01	10	0.1	2		0.49	0.51	0.5	14.2	15.5	0.43	913	0.48
0.3	0.3	0.000	0.01	10	0.1	3		0.36	2.39	0.5	14.4	17.5	0.92	1049	1.11
0.3	0.3	0.025	0.00	10	0.1	2		0.31	2.05	0.4	13.9	22.0	0.94	905	1.02
0.3	0.3	0.025	0.00	10	0.1	3		0.33	1.55	0.5	14.2	19.0	0.85	1024	0.88
0.3	0.3	0.025	0.01	10	0.1	2		0.35	1.22	0.4	14.1	14.0	1.11	1232	0.86
0.3	0.3	0.025	0.01	10	0.1	3		0.31	2.75	0.5	14.2	18.0	1.51	1147	1.36
0.3	0.3	0.050	0.00	10	0.1	2		0.31	2.84	0.5	14.5	19.0	1.06	982	1.28
0.3	0.3	0.050	0.00	10	0.1	3		0.33	1.97	0.4	14.1	16.5	1.08	1158	1.04
0.3	0.3	0.050	0.01	10	0.1	2		0.96	0.19	0.4	13.6	5.0	0.34	4100	0.23
0.3	0.3	0.050	0.01	10	0.1	3		0.49	1.43	0.5	14.2	19.0	0.72	1095	0.78
0.3	0.3	0.075	0.00	10	0.1	2		0.41	1.77	0.6	15.0	22.5	0.84	900	0.95
0.3	0.3	0.075	0.00	10	0.1	3		0.38	0.80	0.5	14.4	22.0	0.55	859	0.62
0.3	0.3	0.075	0.01	10	0.1	2		0.26	1.37	0.5	14.3	18.5	1.22	1070	0.95
0.3	0.3	0.075	0.01	10	0.1	3		0.16	4.70	0.5	14.2	15.0	1.96	1327	1.99
0.3	0.4	0.000	0.00	10	0.1	2		0.35	1.39	0.4	14.0	25.0	0.78	810	0.81
0.3	0.4	0.000	0.00	10	0.1	3		0.12	16.75	0.5	14.5	8.5	4.60	2359	5.69
0.3	0.4	0.000	0.01	10	0.1	2		0.09	10.85	0.5	14.6	10.0	6.44	1770	4.68
0.3	0.4	0.000	0.01	10	0.1	3		0.29	3.77	0.4	14.0	17.5	1.26	1169	1.54
0.3	0.4	0.025	0.00	10	0.1	2		0.37	3.54	0.4	13.8	16.5	1.61	1242	1.54
0.3	0.4	0.025	0.00	10	0.1	3		0.28	2.52	0.5	14.1	21.0	1.17	917	1.22
0.3	0.4	0.025	0.01	10	0.1	2		0.17	6.51	0.5	14.2	13.0	2.40	1458	2.55
0.3	0.4	0.025	0.01	10	0.1	3		0.34	2.98	0.5	14.4	16.0	1.24	1053	1.35
0.3	0.4	0.050	0.00	10	0.1	2		0.10	9.21	0.4	13.6	14.5	2.17	1307	3.16
0.3	0.4	0.050	0.00	10	0.1	3		0.07	7.17	0.5	14.3	13.5	2.78	1396	2.84
0.3	0.4	0.050	0.01	10	0.1	2		0.26	1.81	0.4	13.8	18.5	1.32	1000	1.07
0.3	0.4	0.050	0.01	10	0.1	3		0.70	0.56	0.4	13.8	13.5	0.65	1400	0.48
0.3	0.4	0.075	0.00	10	0.1	2		0.32	5.17	0.5	14.6	18.5	1.55	1038	1.99
0.3	0.4	0.075	0.00	10	0.1	3		0.79	0.48	0.4	13.8	11.5	0.68	1830	0.44
0.3	0.4	0.075	0.01	10	0.1	2		0.19	2.99	0.4	14.0	16.5	1.91	1206	1.53
0.3	0.4	0.075	0.01	10	0.1	3		0.23	2.36	0.5	14.3	16.5	1.88	1264	1.38
0.3	0.5	0.000	0.00	10	0.1	2		0.21	7.23	0.4	13.9	11.0	2.19	1323	2.65
0.3	0.5	0.000	0.00	10	0.1	3		0.25	3.38	0.5	14.4	12.5	1.57	1004	1.55
0.3	0.5	0.000	0.01	10	0.1	2		0.04	16.30	0.2	12.9	6.5	4.96	1308	5.61
0.3	0.5	0.000	0.01	10	0.1	3		0.53	0.59	0.2	12.9	7.0	0.95	936	0.56
0.3	0.5	0.025	0.00	10	0.1	2		0.69	2.82	0.4	13.7	9.5	1.26	2142	1.19
0.3	0.5	0.025	0.00	10	0.1	3		0.02	17.59	0.0	10.2	2.5	4.69	820	5.81
0.3	0.5	0.025	0.01	10	0.1	2		0.17	20.26	0.4	14.1	11.5	4.63	1061	6.54
0.3	0.5	0.025	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.5	0.050	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.5	0.050	0.00	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.5	0.050	0.01	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.5	0.050	0.01	10	0.1	3		0.03	20.24	0.2	12.9	11.5	4.00	1030	6.36
0.3	0.5	0.075	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.5	0.075	0.00	10	0.1	3		0.80	0.99	0.2	12.8	6.0	1.14	3467	0.64
0.3	0.5	0.075	0.01	10	0.1	2		0.13	8.65	0.3	13.5	18.5	2.14	989	3.00
0.3	0.5	0.075	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.000	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.000	0.00	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00

F	Kp	Kd	Ki	Int	Thr	Msr		C%	Std	Osh	D	Osc	A	T	R
0.3	0.6	0.000	0.01	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.000	0.01	10	0.1	3		0.15	18.12	0.3	13.0	4.0	1.07	1000	5.08
0.3	0.6	0.025	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.025	0.00	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.025	0.01	10	0.1	2		0.10	22.64	0.4	13.8	9.0	4.59	1967	7.13
0.3	0.6	0.025	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.050	0.00	10	0.1	2		0.12	18.08	0.3	13.5	7.0	1.37	1064	5.17
0.3	0.6	0.050	0.00	10	0.1	3		0.01	20.75	0.0	10.1	0.0	-1.00	-1	5.18
0.3	0.6	0.050	0.01	10	0.1	2		0.06	22.89	0.4	14.0	7.0	1.65	1393	6.48
0.3	0.6	0.050	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.075	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.075	0.00	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.075	0.01	10	0.1	2		0.01	13.93	0.3	13.4	13.5	5.35	1293	5.15
0.3	0.6	0.075	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00

7.2 PID battery test for the medium tip

F	Target force in Newtons
Kp	PID proportional gain constant
Kd	PID derivative gain constant
Ki	PID integral gain constant
Int	Number of last errors taken into account to calculate integral term
Thr	Threshold in Newtons. Valid force range: [target - threshold, target + threshold]
Msr	Number of measures taken to obtain the mean force value
C%	Contact percentage
Std	Standard deviation in Newtons
Osh	Overshoot in Newtons
D	Maximum depth reached in mm
Osc	Number of oscillations
A	Mean amplitude of the oscillations in mm
T	Mean period of the oscillations in milliseconds
R	Ranking value

F	Kp	Kd	Ki	Int	Thr	Msr		C%	Std.	Osht.	Depth	Osc.	A	T	R
0.3	0.2	0.000	0.00	10	0.1	2		0.96	0.08	0.1	11.9	3.5	0.16	5057	0.10
0.3	0.2	0.000	0.00	10	0.1	3		0.30	0.81	0.4	13.0	18.0	0.47	792	0.58
0.3	0.2	0.000	0.01	10	0.1	2		0.29	0.58	0.4	13.1	16.0	0.68	1134	0.59
0.3	0.2	0.000	0.01	10	0.1	3		0.42	0.49	0.5	13.4	18.5	0.56	1054	0.53
0.3	0.2	0.025	0.00	10	0.1	2		1.00	0.03	0.1	11.7	3.0	0.14	5550	0.06
0.3	0.2	0.025	0.00	10	0.1	3		0.41	0.53	0.4	13.1	18.5	0.49	997	0.50
0.3	0.2	0.025	0.01	10	0.1	2		0.42	0.65	0.5	13.4	17.0	0.75	1106	0.61
0.3	0.2	0.025	0.01	10	0.1	3		0.44	0.48	0.4	12.9	20.5	0.52	971	0.48
0.3	0.2	0.050	0.00	10	0.1	2		0.91	0.38	0.2	12.2	3.0	0.72	5817	0.34
0.3	0.2	0.050	0.00	10	0.1	3		0.44	0.50	0.3	12.8	22.0	0.30	770	0.42
0.3	0.2	0.050	0.01	10	0.1	2		0.71	0.44	0.4	13.0	10.0	0.38	1950	0.37
0.3	0.2	0.050	0.01	10	0.1	3		0.32	0.92	0.4	13.0	15.0	0.96	1247	0.74
0.3	0.2	0.075	0.00	10	0.1	2		0.36	0.57	0.4	13.1	17.5	0.50	1031	0.52
0.3	0.2	0.075	0.00	10	0.1	3		0.33	1.12	0.4	13.1	16.5	0.56	1127	0.69
0.3	0.2	0.075	0.01	10	0.1	2		0.81	0.33	0.4	13.2	8.5	0.40	2306	0.34
0.3	0.2	0.075	0.01	10	0.1	3		0.82	0.62	0.4	13.3	4.5	0.88	3967	0.53
0.3	0.3	0.000	0.00	10	0.1	2		0.27	1.40	0.4	13.1	18.0	1.43	1053	0.99
0.3	0.3	0.000	0.00	10	0.1	3		0.99	0.05	0.2	12.1	3.5	0.26	5800	0.13
0.3	0.3	0.000	0.01	10	0.1	2		0.33	2.43	0.6	14.1	11.5	2.12	1657	1.46
0.3	0.3	0.000	0.01	10	0.1	3		0.13	2.77	0.4	13.3	10.5	2.66	1695	1.69
0.3	0.3	0.025	0.00	10	0.1	2		0.36	0.95	0.5	13.6	24.0	0.77	850	0.72

F	Kp	Kd	Ki	Int	Thr	Msr		C%	Std.	Osht.	Depth	Osc.	A	T	R
0.3	0.3	0.025	0.00	10	0.1	3		0.17	6.02	0.4	13.0	14.0	2.14	1368	2.34
0.3	0.3	0.025	0.01	10	0.1	2		0.30	1.45	0.5	13.4	18.0	1.26	1197	0.97
0.3	0.3	0.025	0.01	10	0.1	3		0.13	3.25	0.4	13.1	15.0	2.15	1330	1.67
0.3	0.3	0.050	0.00	10	0.1	2		0.21	2.64	0.4	13.1	15.5	1.40	1300	1.31
0.3	0.3	0.050	0.00	10	0.1	3		0.19	4.32	0.5	13.5	11.0	2.19	1668	1.95
0.3	0.3	0.050	0.01	10	0.1	2		0.28	1.93	0.4	13.1	14.5	1.27	1186	1.08
0.3	0.3	0.050	0.01	10	0.1	3		0.32	1.61	0.5	13.5	18.0	1.16	1108	0.98
0.3	0.3	0.075	0.00	10	0.1	2		0.23	1.04	0.5	13.4	14.0	1.25	1307	0.88
0.3	0.3	0.075	0.00	10	0.1	3		0.24	1.12	0.4	13.3	17.5	1.11	1114	0.86
0.3	0.3	0.075	0.01	10	0.1	2		0.18	2.77	0.5	13.7	16.5	1.85	1276	1.49
0.3	0.3	0.075	0.01	10	0.1	3		0.31	1.27	0.5	13.5	17.5	1.32	1174	0.94
0.3	0.4	0.000	0.00	10	0.1	2		0.21	2.45	0.5	13.7	17.5	1.60	1146	1.34
0.3	0.4	0.000	0.00	10	0.1	3		0.17	1.90	0.5	13.6	19.0	1.64	1029	1.22
0.3	0.4	0.000	0.01	10	0.1	2		0.19	2.56	0.6	13.9	15.0	2.23	1343	1.54
0.3	0.4	0.000	0.01	10	0.1	3		0.14	8.61	0.5	13.4	15.0	3.35	1427	3.32
0.3	0.4	0.025	0.00	10	0.1	2		0.22	1.53	0.4	13.3	15.5	1.55	1339	1.08
0.3	0.4	0.025	0.00	10	0.1	3		0.33	4.10	0.5	13.4	19.0	1.37	1097	1.65
0.3	0.4	0.025	0.01	10	0.1	2		0.17	2.81	0.6	14.2	16.5	2.17	1267	1.62
0.3	0.4	0.025	0.01	10	0.1	3		0.22	3.60	0.5	13.4	14.0	2.52	1468	1.84
0.3	0.4	0.050	0.00	10	0.1	2		0.24	15.74	0.4	13.2	18.5	2.97	1000	4.97
0.3	0.4	0.050	0.00	10	0.1	3		0.41	2.88	0.4	13.0	11.5	1.82	1852	1.42
0.3	0.4	0.050	0.01	10	0.1	2		0.08	4.24	0.6	13.8	9.0	3.79	1544	2.38
0.3	0.4	0.050	0.01	10	0.1	3		0.11	4.94	0.4	13.2	16.0	2.78	1222	2.26
0.3	0.4	0.075	0.00	10	0.1	2		0.57	0.56	0.5	13.5	7.5	1.00	1340	0.62
0.3	0.4	0.075	0.00	10	0.1	3		0.06	19.06	0.5	13.4	11.0	3.37	1877	5.96
0.3	0.4	0.075	0.01	10	0.1	2		0.07	6.49	0.5	13.6	9.5	4.64	2226	3.14
0.3	0.4	0.075	0.01	10	0.1	3		0.11	3.06	0.3	12.8	19.5	1.43	908	1.43
0.3	0.5	0.000	0.00	10	0.1	2		0.00	6.10	0.4	13.2	12.5	4.14	1608	2.91
0.3	0.5	0.000	0.00	10	0.1	3		0.04	6.94	0.3	12.8	16.0	3.41	1250	2.91
0.3	0.5	0.000	0.01	10	0.1	2		0.10	6.58	0.5	13.4	16.5	2.82	1264	2.69
0.3	0.5	0.000	0.01	10	0.1	3		0.07	6.91	0.4	13.0	15.5	2.15	994	2.59
0.3	0.5	0.025	0.00	10	0.1	2		0.02	16.89	0.5	13.6	13.5	3.36	1226	5.44
0.3	0.5	0.025	0.00	10	0.1	3		0.20	4.88	0.4	13.2	14.0	2.12	1236	2.05
0.3	0.5	0.025	0.01	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.5	0.025	0.01	10	0.1	3		0.02	17.20	0.5	13.7	10.5	8.04	1924	6.69
0.3	0.5	0.050	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.5	0.050	0.00	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.5	0.050	0.01	10	0.1	2		0.02	15.27	0.5	13.7	8.5	6.59	1565	5.85
0.3	0.5	0.050	0.01	10	0.1	3		0.05	10.71	0.4	13.3	7.5	7.16	2220	4.82
0.3	0.5	0.075	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.5	0.075	0.00	10	0.1	3		0.32	4.04	0.3	12.8	10.0	0.70	795	1.44
0.3	0.5	0.075	0.01	10	0.1	2		0.06	22.03	0.4	12.9	2.5	10.78	1460	8.53
0.3	0.5	0.075	0.01	10	0.1	3		0.07	19.38	0.5	13.5	14.0	0.94	996	5.44
0.3	0.6	0.000	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.000	0.00	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.000	0.01	10	0.1	2		0.12	21.95	0.4	13.3	2.0	1.69	800	6.24
0.3	0.6	0.000	0.01	10	0.1	3		0.12	22.05	0.1	12.0	7.0	1.49	1086	6.14
0.3	0.6	0.025	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00

F	Kp	Kd	Ki	Int	Thr	Msr		C%	Std.	Osht.	Depth	Osc.	A	T	R
0.3	0.6	0.025	0.00	10	0.1	3		0.23	2.75	0.4	13.0	5.5	1.97	1200	1.47
0.3	0.6	0.025	0.01	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.025	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.050	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.050	0.00	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.050	0.01	10	0.1	2		0.08	13.90	0.3	12.6	16.0	2.17	1106	4.32
0.3	0.6	0.050	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.075	0.00	10	0.1	2		0.04	20.17	0.0	10.4	1.5	1.72	200	5.71
0.3	0.6	0.075	0.00	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.075	0.01	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.075	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00

7.3 PID battery test for the big tip

F	Target force in Newtons
Kp	PID proportional gain constant
Kd	PID derivative gain constant
Ki	PID integral gain constant
Int	Number of last errors taken into account to calculate integral term
Thr	Threshold in Newtons. Valid force range: [target - threshold, target + threshold]
Msr	Number of measures taken to obtain the mean force value
C%	Contact percentage
Std	Standard deviation in Newtons
Osh	Overshoot in Newtons
D	Maximum depth reached in mm
Osc	Number of oscillations
A	Mean amplitude of the oscillations in mm
T	Mean period of the oscillations in milliseconds
R	Ranking value

F	Kp	Kd	Ki	Int	Thr	Msr		C%	Std.	Osht.	Depth	Osc.	A	T	R
0.3	0.2	0.000	0.00	10	0.1	2		0.35	0.39	0.5	12.6	17.0	0.41	1035	0.49
0.3	0.2	0.000	0.00	10	0.1	3		0.36	0.59	0.6	12.9	19.5	0.40	972	0.56
0.3	0.2	0.000	0.01	10	0.1	2		0.46	0.69	0.6	12.8	16.0	0.75	1281	0.64
0.3	0.2	0.000	0.01	10	0.1	3		0.23	1.19	0.7	13.3	10.5	1.93	1838	1.15
0.3	0.2	0.025	0.00	10	0.1	2		0.39	0.45	0.6	12.8	16.0	0.46	1088	0.52
0.3	0.2	0.025	0.00	10	0.1	3		0.92	0.17	0.5	12.5	8.5	0.19	2182	0.23
0.3	0.2	0.025	0.01	10	0.1	2		0.72	0.35	0.6	12.9	13.0	0.43	1485	0.42
0.3	0.2	0.025	0.01	10	0.1	3		0.80	0.37	0.4	12.4	6.5	0.73	3115	0.43
0.3	0.2	0.050	0.00	10	0.1	2		0.38	0.42	0.5	12.7	21.0	0.34	893	0.47
0.3	0.2	0.050	0.00	10	0.1	3		0.33	1.14	0.5	12.6	24.5	0.36	769	0.67
0.3	0.2	0.050	0.01	10	0.1	2		0.40	1.42	0.7	13.3	15.0	0.73	1193	0.87
0.3	0.2	0.050	0.01	10	0.1	3		0.43	0.62	0.6	12.9	15.5	0.47	1171	0.56
0.3	0.2	0.075	0.00	10	0.1	2		0.47	0.35	0.6	12.8	11.0	0.43	1218	0.47
0.3	0.2	0.075	0.00	10	0.1	3		0.86	0.31	0.5	12.4	3.5	0.23	5129	0.29
0.3	0.2	0.075	0.01	10	0.1	2		0.22	1.28	0.7	13.2	15.0	1.20	1263	0.99
0.3	0.2	0.075	0.01	10	0.1	3		0.32	1.33	0.6	12.9	15.5	0.84	1094	0.86
0.3	0.3	0.000	0.00	10	0.1	2		0.42	1.41	0.6	12.8	17.0	0.75	1076	0.83
0.3	0.3	0.000	0.00	10	0.1	3		0.17	3.50	0.5	12.6	8.5	1.67	1253	1.62
0.3	0.3	0.000	0.01	10	0.1	2		0.13	2.48	0.6	13.0	16.5	1.90	1236	1.47
0.3	0.3	0.000	0.01	10	0.1	3		0.27	2.64	0.7	13.2	14.0	1.28	1136	1.34
0.3	0.3	0.025	0.00	10	0.1	2		0.26	1.01	0.6	12.9	18.0	1.11	1094	0.86

F	Kp	Kd	Ki	Int	Thr	Msr		C%	Std.	Osht.	Depth	Osc.	A	T	R
0.3	0.3	0.025	0.00	10	0.1	3		0.67	0.62	0.6	12.8	14.5	0.74	1407	0.56
0.3	0.3	0.025	0.01	10	0.1	2		0.11	2.55	0.8	13.7	12.0	2.84	1763	1.78
0.3	0.3	0.025	0.01	10	0.1	3		0.21	4.22	0.7	13.2	16.5	1.62	1252	1.82
0.3	0.3	0.050	0.00	10	0.1	2		0.34	0.69	0.6	12.9	19.5	0.64	933	0.65
0.3	0.3	0.050	0.00	10	0.1	3		0.25	1.26	0.6	13.0	18.0	1.11	1014	0.94
0.3	0.3	0.050	0.01	10	0.1	2		0.13	2.00	0.9	13.9	16.5	1.95	1209	1.43
0.3	0.3	0.050	0.01	10	0.1	3		0.15	4.78	0.7	13.2	10.5	2.67	1800	2.25
0.3	0.3	0.075	0.00	10	0.1	2		0.21	1.64	0.5	12.6	18.0	1.24	1053	1.04
0.3	0.3	0.075	0.00	10	0.1	3		0.23	3.71	0.6	13.1	14.5	1.61	1300	1.68
0.3	0.3	0.075	0.01	10	0.1	2		0.21	2.48	0.9	13.8	13.5	2.03	1459	1.55
0.3	0.3	0.075	0.01	10	0.1	3		0.31	1.00	0.6	13.0	22.5	0.86	880	0.80
0.3	0.4	0.000	0.00	10	0.1	2		0.15	5.21	0.7	13.3	16.5	1.64	1170	2.10
0.3	0.4	0.000	0.00	10	0.1	3		0.23	1.64	0.9	13.7	15.5	1.90	1284	1.29
0.3	0.4	0.000	0.01	10	0.1	2		0.17	8.66	0.6	13.0	8.5	4.75	1618	3.72
0.3	0.4	0.000	0.01	10	0.1	3		0.29	2.14	0.8	13.5	12.5	2.64	1584	1.57
0.3	0.4	0.025	0.00	10	0.1	2		0.10	13.55	0.5	12.8	8.0	6.20	2475	5.30
0.3	0.4	0.025	0.00	10	0.1	3		0.09	8.37	0.5	12.7	8.0	5.07	2363	3.72
0.3	0.4	0.025	0.01	10	0.1	2		0.16	3.62	0.8	13.6	16.5	2.40	1218	1.92
0.3	0.4	0.025	0.01	10	0.1	3		0.58	1.20	0.6	12.8	10.0	1.75	2095	0.98
0.3	0.4	0.050	0.00	10	0.1	2		0.51	0.88	0.8	13.5	15.0	0.86	1333	0.75
0.3	0.4	0.050	0.00	10	0.1	3		0.35	1.68	0.7	13.1	10.5	1.74	1471	1.18
0.3	0.4	0.050	0.01	10	0.1	2		0.14	3.86	0.6	12.9	12.0	3.63	1779	2.24
0.3	0.4	0.050	0.01	10	0.1	3		0.36	6.19	0.7	13.1	11.5	2.68	1796	2.54
0.3	0.4	0.075	0.00	10	0.1	2		0.28	1.80	0.7	13.2	19.5	1.20	1033	1.10
0.3	0.4	0.075	0.00	10	0.1	3		0.24	3.23	0.7	13.2	20.5	1.17	976	1.46
0.3	0.4	0.075	0.01	10	0.1	2		0.06	17.34	0.6	13.1	8.0	2.06	1494	5.25
0.3	0.4	0.075	0.01	10	0.1	3		0.15	3.02	0.7	13.4	9.0	3.05	1478	1.91
0.3	0.5	0.000	0.00	10	0.1	2		0.07	9.93	0.6	13.1	12.0	5.41	1600	4.23
0.3	0.5	0.000	0.00	10	0.1	3		0.14	6.54	0.5	12.6	7.0	3.32	1464	2.81
0.3	0.5	0.000	0.01	10	0.1	2		0.14	4.49	0.7	13.3	16.0	3.10	1244	2.29
0.3	0.5	0.000	0.01	10	0.1	3		0.05	16.95	0.7	13.2	11.5	4.06	1261	5.67
0.3	0.5	0.025	0.00	10	0.1	2		0.36	6.81	0.6	12.9	12.0	1.75	1400	2.44
0.3	0.5	0.025	0.00	10	0.1	3		0.11	13.27	0.7	13.1	13.0	2.74	1212	4.39
0.3	0.5	0.025	0.01	10	0.1	2		0.04	6.47	0.8	13.7	17.5	3.62	1146	2.97
0.3	0.5	0.025	0.01	10	0.1	3		0.16	10.37	0.8	13.5	2.0	18.11	1600	7.52
0.3	0.5	0.050	0.00	10	0.1	2		0.16	14.02	0.6	13.1	14.0	1.35	911	4.21
0.3	0.5	0.050	0.00	10	0.1	3		0.19	2.57	0.6	13.0	26.0	1.27	846	1.32
0.3	0.5	0.050	0.01	10	0.1	2		0.04	17.15	1.0	14.3	6.5	10.40	2777	7.38
0.3	0.5	0.050	0.01	10	0.1	3		0.12	15.56	0.8	13.5	7.5	2.17	960	4.85
0.3	0.5	0.075	0.00	10	0.1	2		0.12	10.82	0.5	12.6	9.5	4.94	1716	4.29
0.3	0.5	0.075	0.00	10	0.1	3		0.20	8.08	0.6	13.0	4.0	1.54	975	2.76
0.3	0.5	0.075	0.01	10	0.1	2		0.09	5.07	0.6	13.1	9.5	3.54	1268	2.54
0.3	0.5	0.075	0.01	10	0.1	3		0.09	5.75	0.5	12.7	6.5	4.12	1408	2.82
0.3	0.6	0.000	0.00	10	0.1	2		0.08	11.87	0.5	12.7	11.0	6.17	1895	4.88
0.3	0.6	0.000	0.00	10	0.1	3		0.05	15.66	0.8	13.5	10.5	7.31	1976	6.18
0.3	0.6	0.000	0.01	10	0.1	2		0.01	16.99	0.0	10.7	9.5	7.92	1774	6.48
0.3	0.6	0.000	0.01	10	0.1	3		0.06	17.74	0.8	13.4	6.5	11.65	2469	7.77
0.3	0.6	0.025	0.00	10	0.1	2		0.03	18.76	1.0	14.1	3.0	0.98	1117	5.41

F	Kp	Kd	Ki	Int	Thr	Msr		C%	Std.	Osht.	Depth	Osc.	A	T	R
0.3	0.6	0.025	0.00	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.025	0.01	10	0.1	2		0.07	17.45	0.5	12.6	4.0	1.27	1038	5.04
0.3	0.6	0.025	0.01	10	0.1	3		0.01	12.52	0.5	12.5	7.5	5.36	1180	4.83
0.3	0.6	0.050	0.00	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.050	0.00	10	0.1	3		0.04	16.06	0.3	11.8	2.5	12.14	1140	7.35
0.3	0.6	0.050	0.01	10	0.1	2		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.050	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00
0.3	0.6	0.075	0.00	10	0.1	2		0.05	20.59	0.8	13.7	1.5	5.10	533	6.87
0.3	0.6	0.075	0.00	10	0.1	3		0.05	17.73	0.3	12.1	15.0	3.42	1360	5.61
0.3	0.6	0.075	0.01	10	0.1	2		0.09	14.94	0.1	11.4	7.0	2.19	1057	4.55
0.3	0.6	0.075	0.01	10	0.1	3		0.00	0.00	0.0	0.0	0.0	-1.00	-1	0.00

7.4 Launch codes

In this section the launch codes associated to each functionality are listed. This list is also detailed and extended in the OnRun() function of the classAnalysis class, including some other functionalities which are not accesible directly from the GUI due to safety reasons. Codes from 100 to 199 are reserved for the scan process, from 200 to 299 for calibration and from 300 for tests.

100	Brain surface scan process.
200	Gravity compensation calibration for polynomial fitting.
201	Force sensor initial distance between original center and tumor tool determination.
202	Force sensor original rotation determination.
203	Joint 5 alignment calibration.
204	Joint 4 alignment calibration.
205	Joint 3 alignment calibration.
206	Joint 3, 4, 5 alignment calibration at the same time.
250	Battery test for the Approach process and the PID controller.
300	Force sensor test. Mean, deviation and time requirements are stored.
301	Force sensor reset time test. It stores the number of resets required to read a mean force under different thresholds.
302	Force sensor accuracy test.
303	Force sensor static precision test.
304	Force sensor dynamic precision test.
305	Force sensor age test. It checks the time required to take a measure.
310	Spherical move test. It moves the hand to different positions on the Displacement Hemisphere.
320	Alter behaviour test. It tests the behaviour of the Alter Mode by performing some translations and rotations.
330	Gravity compensation: force computation method test.
331	Gravity compensation: nearest method test.
332	Gravity compensation: weighted mean method test.
333	Gravity compensation: polynomial fitting method test.
340	Deep test. It stores the measured force according to a known distance into the model.
350	Perpendicular position test. It performs an approach to the surface and then it orientates the hand.

References

[1] Russell A. Faust (2007): Robotics in Surgery: History, Current, and Future Applications.

[2] Anthony R. Lanfranco, Andres E. Castellanos, Jaydev P. Desai, William C. Meyers (2004): Robotic Surgery: A Current Perspective, Ann Surg. 2004 January; 239(1): 14–21.

[3] J. Lucena et al. (2007): Historia, evolución, estado actual y futuro de la cirugía robótica, Revista de la Facultad de Medicina, Volumen 30 - Número 2, 2007 (109-114),http://www.scielo.org.ve/pdf/rfm/v30n2/art02.pdf (last visit on August 21, 2012)

[4] Etienne Dombre (2005): Introduction to Surgical Robotics, LIRMM Montpellier, http://www.lirmm.fr/~w3rob/UEE05/presentations/lecturers/Dombre.pdf (last visit on August 21, 2012)

[5] NBCNews (2011): Robo-Doc Knee Surgery, http://www.youtube.com/watch?v=fhjvxi7X1rw (last visit on August 21, 2012)

[6] Ralf Stroop (2007): Taktile Sensorik in der intraoperativen Robotik.

[7] JR3 Inc.: JR3 100M40A3 Force Sensor Data Sheet, http://www.jr3.com/documents/datasheets/MSeries/100M40A3.pdf (last visit on August 21, 2012)

[8] Tomas B. Co (2004): Ziegler-Nichols Closed Loop Tuning, Michigan Technological University.

[9] J. G. Ziegler, N. B. Nichols (1942): Optimum settings for automatic controllers, Transactions of the ASME. 64. pp. 759–768

[10] Microstar Laboratories (2009): Ziegler-Nichols Tuning Rules for PID, http://www.mstarlabs.com/control/znrule.html (last visit on August 21, 2012)

[11] Eric W. Weisstein (2012): Least Squares Fitting-Polynomial, MathWorld - A Wolfram Web Resource, http://mathworld.wolfram.com/LeastSquaresFittingPolynomial.html (last visit on August 21, 2012)

[12] Armadillo C++ linear algebra library, http://arma.sourceforge.net/ (last visit on August 21, 2012)

[13] K. S. Fu, R. C. Gonzalez, C. S. G. Lee (1987): Robotics, p. 23.

[14] Barry W. Boehm (1988): A spiral model of software development and enhancement.

[15] Sharam Hekmatpour, Darrel C. Ince (1988): Software Prototyping, Formal Methods, and VDM, Addison-Wesley, ISBN:020117572X.

[16] Prosise, Jeff (1999): Programming Windows with MFC, Microsoft Press.

[17] Paul Bourke: PLY - Polygon File Format, http://paulbourke.net/dataformats/ply/ (last visit on August 21, 2012)

[18] Mark de Berg, Otfried Cheong, Marc van Kreveld, Mark Overmars (2000): Computational Geometry: Algorithms and Applications, Springer-Verlag, ISBN 9783-540-77973-5.

[19] E. A. Rodríguez Tello (April 13 2010): Triangulaciones de Delaunay (construcción incremental), Cinvestav Tamaulipas.

[20] MeshLab, http://meshlab.sourceforge.net/ (last visit on August 21, 2012)

[21] Microsoft Corporation (2012): KINECT for Windows http://www.microsoft.com/en-us/kinectforwindows/ (last visit on August 21, 2012)