University of Verona

Master's Degree in

Computer Engeneering for Robotics and Smart Industry

Development of Software for Planning a Mini-Invasive Robot Assisted Vertebroplasty

Supervisor
Doc. Maris Bogdan Mihai

Candidate
Nicola Marchiotto
VR462317

Academic Year 2021/2022

Alla mia famiglia e a Irene,
non avrei potuto chiedere un sostegno più grande.

Abstract

This thesis is the result of an internship and a following collaboration with Tecres, a company which studies and develops biomaterials and medical devices in the fields of orthopedics, traumatology, vertebral consolidation and neurosurgery. The work concerns two main aspects, the refinement of a robot prototype for performing percutaneous vertebroplasty and the development of the software for controlling and planning the surgery. To prove the feasibility of vertebroplasty with the current prototype and test the correctness of the ecosystem, the study exploits computerized tomographies of phantoms to perform image-registration of the robot, plan the needle trajectories and simulate the fluoroscopes, the image-guide supports used by the doctor during a normal vertebroplasty. The results of the research are validated performing trajectories planned on the developed software interface, having as target salient features of the phantoms. Finally, limits of the prototype are reported and possible future directions of development are proposed.

Contents

Chapter 1

Introduction

1.1 Robot-Assisted Medical Procedures

In traditional surgery, the doctor exploits imaging techniques such as magnetic resonance (MRI), ultrasound (US), computerized tomography (CT), to detect lesions, plan the intervention before surgery and to have feedback on the instruments' position in real. The success of robot-assisted surgery heavily relies on the ability to merge pre-operative images, provided by one of the previous cited imaging technique and where the operation is planned, with the intraoperative images, retrieved during the surgery and used for registration purposes.

Medical robots generally fall into three categories: supervisory-controlled, telesurgical, and shared-control. With the first ones, the surgeon is able to design the procedure in its entirety prior to surgery. The robot then performs the procedure as the physician supervise it. Telesurgical robots give the surgeon remote direct control over the robot and its tools during the surgery. Finally shared-control robots give both the surgeon and the robot the ability to control tools and movement at the same time. [4]

Many benefits come from robot assistance in surgery, with many studies proving that robotic surgery have lower rates of post operation complications. Moreover, robotic navigation assistance greatly helps in following long and narrow paths, where the human hand usually struggle. It also reduces hand tremors, allowing a less tiring overall experience for the surgeon. Radiation exposure is also a factor to be considered given the nature of the used imaging systems. Numerous studies have evaluated a reduction of intraoperative radiation time and dosage in robot assisted surgery compared to free-hands operations [4, 8, 15]. Most of these robotic systems allow the surgeon to have a 3-D visualization of the area of interest via dedicated software, while classical image sup-

ports in non-assisted procedures show only a 2-D representation. The history of robotic assistance in surgery begins in the mid-1980s, when the PUMA 560 became the first robot used to perform neurosurgical brain biopsy, achieving better performance than techniques of its time [4].

Depending on the type of surgery, the image acquisition system, whether the operation is percutaneous or open and if the robot must be attached to the body of the patient or on a separate freestanding mobile base, different types of solutions have been adopted in the literature. An excursus of robot assisted applications in the medical field is now provided.

1.1.1 Brain Procedures

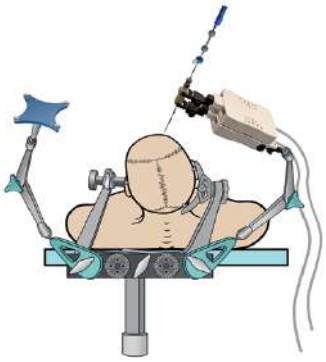
Robotic devices have been employed in stereotactic neurosurgery to address the accuracy and safety issues with frame-based and frame-less methods. The neurosurgical surgeries which have profited more by robot assistance are catheter placement and biopsy for diagnosis confirmation of brain tumor. Despite the great performance of the frame-less systems, frame-based methods, which requires a skull-mounted support, are still the gold-standard [11].

In frame-based systems the patient-to-image data registration is performed by aligning recognizable markers of the rigid frame, with the corresponding points of the intraoperative MRI images, usually retrieved using CT scans. Once registration is completed, the robot is attached to the skull-frame and functions as a guiding system for the surgery.[5]

In frame-less systems, no rigid support is attached to the patient head and point-based registration techniques are used to perform registration. These methods require the identification of characteristic points both on the image and on the physical space, they can be marker-based or not. In the first case clearly identifiable markers are attached to the patient's head before acquiring the MRI, these are then related to the ones retrieved in the intraoperative images to complete the registration. In the latter case instead, the necessary points on the images can be retrieved automatically applying specific image processing operations, or chosen by hand on a dedicated software interface. Points on the physical space are identified by touching them with an ad hoc tracking positioning device. Some example of these points are the tip of the nose, the angulus oculi, the tragus on the skin surface and the nasal spine on the bone surface [5]. Differently from frame-based systems, no fixed points/markers geometry is present, thus resulting in a more challenging task.

Surface based registration is another technique which does not require the attachment

of markers or the identification of points on the patient's skin. This methodology is particularly convenient in case the doctor decides to perform surgery with robot assistance after having the patient already undergone imaging procedures. Performing a second CT scan would mean additional cost and radiations exposure for the patient. Surface based techniques in brain procedures use the geometry of the face for doing image-to-patient registration. Exploiting the information already present in the diagnostic images, the two large set of points of the preoperative and intraoperative images are related. Using surface-matching algorithms, such as the Iterative Closet Point method [2, 3], the transformation which link the two reference frame is estimated. Many studies have evaluated the difference in performance for frame-based and frameless systems, showing no difference in the diagnostic yield and regarding the number of post-operative complications. A shorter mean operation time for the frame-less systems and a greater positioning error for the frame-based ones were reported. [5, 7, 11, 17]



(a) Example of frame-based setup used in neurosurgical procedures

(b) Frame-less SINO robot used for brain biopsy

1.1.2 Robot Biopsy

Medical robots have been widely used to perform robot assisted biopsy to acquire a sample of tissue and confirm the malignancy of the diagnosis. Resulting in 25% of the diagnosed cancer for women, breast cancer is the most prevalent tumor for females worldwide [6, 16, 19]. Palpation and imaging modalities, such as mammography, US scans and MRI scans, are used to detect the malicious lesions.

The radiologist usually performs the procedure under US guidance. Benefits of this technology are its low cost and the fact that it can give accurate feedback in real time of metallic instruments. Moreover it causes relatively little patient discomfort [16]. If the lesion is not identifiable by US, the doctor must rely on MRI. The current procedure

of manual MRI-guided biopsy consists of a fist MRI scan, the patient is then moved out of the scanner and the needle is inserted blindly using a mechanical guide, a MRI scan is then acquired for confirmation. No feedback is possible during the surgery and therefore errors in the needle final position are possible, thus resulting in a new attempt and in more uneasiness for the patient [6]. The solutions for robot-guided breast biopsy can exploit combinations of different imaging technologies. A possible workflow which requires an intraoperative MRI scan for path planning and US acquisition for robot registration is illustrated in Figure 1.2 and explained in the MURAB project [19]. Multi-modality markers attached to the breast, visible on MRI, US and with an optical camera, could also be exploited to compute real-time deformations compared to the pre-operative MRI data. Since MRI scanners contain strong magnetic fields, high image-quality is out of question in the presence of ferromagnetic materials, by which robots are normally composed of. If breast-cancer is only visible using MRI, MRI-safe robots could be exploited. These robots are composed of non-metallic materials, except for the needle used for interacting with the patient. They are small enough to fit inside in the Faraday cage of the MRI scanner and do not necessitate registration procedures, since both the tumor and the robot are visible in the MRI images [6].

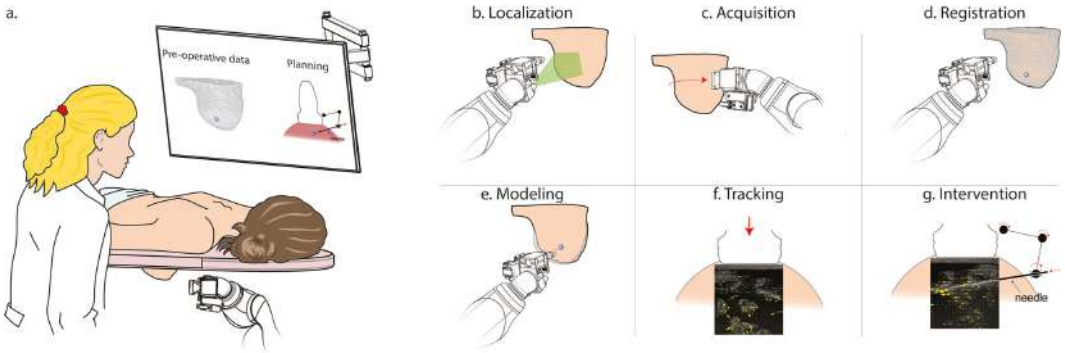


Figure 1.2: Phases of a robotic breast biopsy. a) The radiologist checks the preoperative images and suggested path. b) The robot localizes the patient. c) The robot acquires US data of the site. d) The robot registers the available preoperative data. e) and f) Modelling and tracking are utilized to determine the target location after probe positioning. g) The intervention takes place.

1.1.3 Spine Procedures

It was not until 2004 that robot-assisted spine surgery became relevant and widely used, thanks to the introduction of the US Food and Drug Administration (FDA) certified Mazor SpineAssist. The aim of this robot-assistant was to improve trajectory accuracy of surgical tools during spine procedures [4]. The spine is responsible for important functionalities of our body, most notably it allows us to move and perceive the external world. Damage by surgical dissection or bad placement of spine supports can cause irreversible motor and sensory loss. A standard robotic-assisted spine surgery is usually described by the following steps. A first CT scan of the spine region of interest is performed. On the resulting images, the trajectory is planned either by the doctor or by the provided software of the used system. Once the trajectory is confirmed, the patient is placed on the surgical bed. Depending on the type of surgery, whether the operation is percutaneous or open and if the robot must be attached to the body of the patient or on a separate frame, different methods are used to perform image registration. In open surgery, a frame with so-called fiducials, is usually attached to the bony anatomy of the patient spine. In percutaneous interventions, rather than attaching the frame to the spinous processes, the fiducilas frame can be held up by percutaneously placed guide wires or be positioned on the patient's skin. Using intraoperative CT scans or exploiting fluoroscopies, registration is performed by aligning the detected markers on the pre-operative CT with the ones of the intra-operative images. The robot location in the operative room is another not standard convention and depends on the used device. The Mazor SpineAssist is mounted on the same frame used for image registration, the Mazor X and ROSA spine instead use a freestanding mobile base with a mechanical surgical arm. These latest robotic arms can include an integrated linear optic camera that allows the robot to perform a volumetric assessment of the workspace, detect variations in the position of the markers used for the registration, and behave accordingly to the changed environment. Once it is all set, the robot aligns itself with the planned trajectory and functions as a constraint for the path to follow, a cannulated dilator is placed through the surgical arm followed by a drill guide and guidewire. The doctor then inserts the surgical instrument into the guidewire and, using the feedback of computer assisted navigation, performs the surgery [4, 8, 15].
This thesis focuses on a type of percutaneous intervention called vertebroplasty, a procedure which consists in the injection of a special cement inside a vertebra. This procedure is heavily dependent on the experience of the radiologist, thus the use of robot assistants could potentially improve the accuracy, precision, learning rate and the overall throughput of the intervention.

Figure 1.3: The Mazor X system: A) base station and robotic arm, B) bed and "patient"—mounted robotic arm.

1.2 Percutaneous Vertebroplasty

Percutaneous vertebroplasty is an image-guided surgery in which a cement, a fast-setting polymer, is injected into a pathologic vertebral body. The procedure is used in the setting of painful osteoporotic compression fractures, pathologic fractures from underlying neoplasms, structurally compromised vertebrae and cancerous lesions. The cement is injected in the vertebrae through a needle via a small incision on the back of the patient, exploiting the feedback of digital fluoroscopy. The injected cement solidifies in the damaged vertebral body in usually one hour. This stabilizes the vertebrae, ceasing the pain and reducing the risks of further fractures. The surgery is considered a minimally invasive procedure, is usually performed under local anesthesia and allows a speedy recovery in the range of hours. The concept of minimally invasive surgery was born in 1987, the year of the first-ever laparoscopic cholecystectomy [4]. Surgeries labelled as mini-invasive are characterized by smaller incisions, lower rates of post-operative infections, shorter hospital stays, and shorter convalescence periods for patients after surgery.

During a manual vertebroplasty, the doctor is helped by one or two orthogonal C-Arms to verify the location of the needle while this is being inserted through the back of the patient. C-Arm is a medical imaging device based on X-ray fluoroscopy technology and, differently from CT scans, it allows continuous imaging. Since fluoroscopies produce a 2D representation of a volume which in reality is three-dimensional, errors in the location of the needle's entry point on the back of the patient, and on the needle inclination on the moment of insertion, may occur. Thus, the correct target on the vertebrae may not be reached at the first try. In these occurrences the doctor may decide to repeat the operation, he will extract the needle and redo the procedure until the correct

point is reached [9, 14, 18]. The surgery is not without drawbacks and critical points. For success, the cement must be liquid enough to enable manual injection but to not leak outside the vertebral body. The operation must be carried out in a short operative time window to avoid the abrupt hardening of the cement while the needle is still present in the vertebrae. The continuous X-ray exposure of fluoroscopes to monitor the needle position and cement spread must also be considered [12, 18].

The robotic assistance for vertebroplasty used in the literature, consists in positioning a guide for the needle in the correct entry point on the back of the patient and with the right orientation. Once set, the doctor manually pushes the needle through the guide. The following of the planned trajectory is in this way guaranteed, thus reaching the vertebra at the first try [1, 9, 14, 18].

1.3 C-Arm

C-Arm is a medical machine based on X-ray technology, it is primarily used for fluoroscopy capabilities, although it has radiography capability too. The name C-Arm is due to its C-shaped arm, which is used to connect the x-ray source on one end and the detector on the other.

The machine consists of an X-Ray generator, an image intensifier, or flat-panel detector, and a control workstation. The C-shaped arm connecting the generator and detector allows movement horizontally, vertically and around the swivel axis, letting the interested area to be seen from different angles. Doctors can check anatomical details such as bones' fractures or the position of implants and instruments in real-time, all while performing surgery.

Unlike a regular X-ray which records a single image, digital fluoroscopes record a series of images. Once digitized, the area being examined can be seen in real time on a computer monitor. The resulting images show denser bodies with lighter pixels, while lesser dense bodies are represented with darker pixels. C-Arms are commonly used to analyse internal structures such as the intestines, bones, cardiac muscle, or the stomach.

(a) (b)

Figure 1.4: a) Illustration of a manual vertebroplasty, b) C-Arm

1.4 Experimental Workflow

The implemented software was developed using the MeVisLab framework. MeVisLab is a powerful, modular framework for image processing research and development with a special focus on medical imaging. It allows fast integration and testing of algorithms and the development of clinical application prototypes. The developed application exploits CT of phantoms to plan the trajectory to be performed by the robot prototype. To create a doctor-friendly interface for trajectory planning, MeVisLab functionalities were used to simulate fluoroscopies from the CT scans. Upon confirmation, the software will automatically compute the needle's entry point and orientation necessary to reach the desired target. After performing robot-image registration, test trajectories on the phantoms were carried out.

The thesis is organized as follows: Chapter 2 describes the robot prototype and its kinematics, in Chapter 3 the software for planning the surgery and controlling the robot is illustrated, in Chapter 4 the conducted test and results are reported. Finally, in Chapter 5 future directions of development are proposed

Chapter 2

Robot Prototype

2.1 Mechanical Structure

The robot prototype provided by Tecres is a P-P-R-R-P-R robot, where P stands for prismatic joint and R for rotational joint. The Robot is mounted on a Plexiglas frame to ensure stability, joint limits are reported in Table 2.1. Rotational joint 2 and 3 were limited to respectively ±20° and ±30° as safety measure to avoid collision with the structure itself. The last revolute joint accounts for the spinning motion of the needle, in this stage of the study this functionality is not needed and thus omitted.

#	type	inferior	superior
1	P	0 cm	280 cm
2	P	0 cm	168 cm
3	R	-20°	20°
4	R	-30°	30°
5	P	0 cm	84 cm
6	R	-180°	180°

Table 2.1: Robot joint limits.

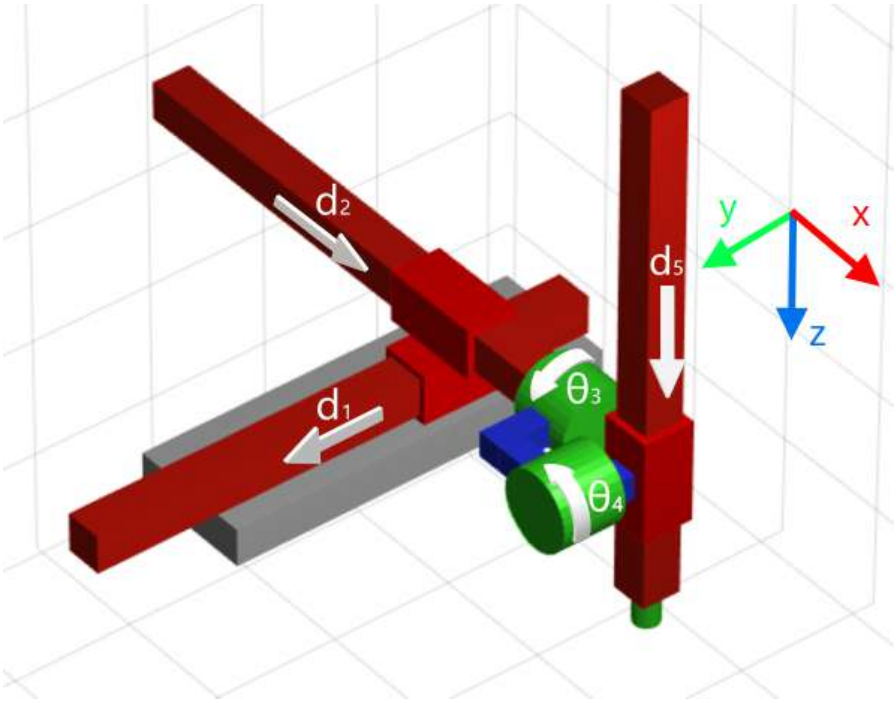


Figure 2.1: Robot structure and reference frame.

Considering the robot reference frame, the first prismatic joint accounts for movement along the y axis, the second for movement along the x axis, the first rotational joint allows rotational motion around the x axis and the second around the y axis. Finally the third prismatic joint is responsible for motion along the z axis. The robot is equipped with stepper motors. A stepper DC motor is a synchronous brushless motor which works in discrete steps by dividing a full rotation into a number of equal intervals. The stepper motor is known for its property of converting a train of input pulses into a precisely defined increment in the shaft's rotational position. Each pulse rotates the shaft of a fixed angle, the so called step. In a stepper motor, a permanent magnet or soft iron is used as the rotor and is surrounded by electromagnetic stators. When voltage is applied at the terminals, the rotor aligns itself with the stator due to the magnetic effect. To allow motion, the stators are energized in sequence. Figure 2.3 illustrate the mechanism of a stepper motor.

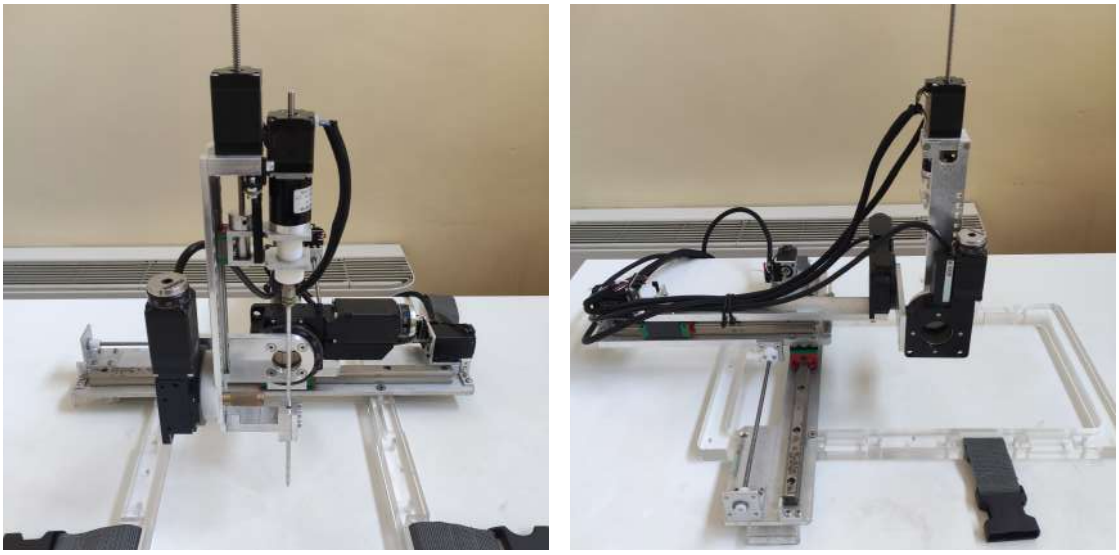


Figure 2.2: Robot prototype on its support frame.

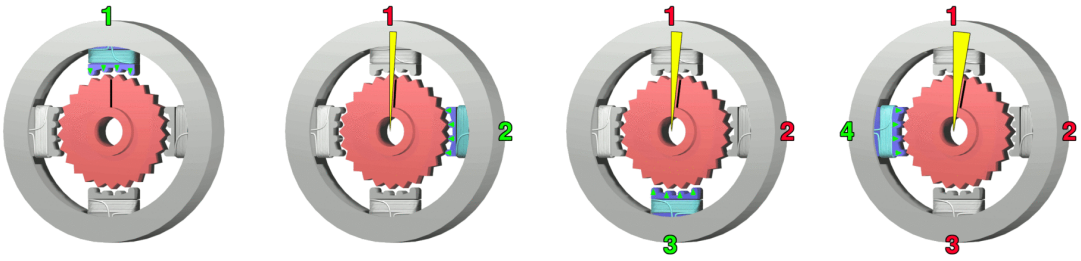


Figure 2.3: Mechanism of a stepper motor.

3D-printed optical limit switches were designed and mounted on the prismatic joints. Revolute joints were not enhanced with limit switches, since no feasible solution which did not constraint the motion was found. On robot power-on, the two revolute joints need to be on their zero value, if not error on the e.e. orientation will occur. The three prismatic joints must instead be calibrated with an implemented homing functionality.

2.2 Forward Kinematics

The forward kinematics is described by the following equation where d_i and θ_i represent the joint variables, respectively for prismatic and revolute joints, the value ee_w indicates the length between the centre of rotation of the wrist and the needle tip when the two rotational and the third prismatic joints are at zero value. The value of $ee_w + d_5$

represents the length of the segment between the centre of rotation of the wrist and the needle tip in any configuration.

$$Forw\_kin: \begin{bmatrix} x \\ y \\ z \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} d_2 \\ d_1 \\ -ee_w \\ \theta_3 \\ \theta_4 \end{bmatrix} + \underbrace{\begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 \\ \sin\theta_3\sin\theta_4 & \cos\theta_3 & -\sin\theta_3\cos\theta_4 \\ -\cos\theta_3\sin\theta_4 & \sin\theta_3 & \cos\theta_3\cos\theta_4 \end{bmatrix}}_{R} \begin{bmatrix} 0 \\ 0 \\ d_5 + ee_w \end{bmatrix}$$

Or in a more compact way

$$Forw\_kin: \begin{bmatrix} x \\ y \\ z \\ \phi \\ \psi \end{bmatrix} = \begin{bmatrix} d_2 \\ d_1 \\ (d_5 + ee_w)\cos\theta_3\theta_4 \\ \theta_3 \\ \theta_4 \end{bmatrix} + \begin{bmatrix} (d_5 + ee_w)\sin\theta_4 \\ -(d_5 + ee_w)\sin\theta_3\cos\theta_4 \\ -ee_w \\ 0 \\ 0 \end{bmatrix}$$

The pose of the E.E. in x and y is described by the values of the first two prismatic joints plus the projection of the segment $ee_w + d_5$ along the respective axis. The z coordinate is the projection of the $ee_w + d_5$ segment minus the initial offset ee_w. The orientation (ϕ, ψ) is instead influenced by only the two revolute joints. Steps for computing the R matrix are provided.

$$R_x(\theta_3) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_3 & -\sin\theta_3 \\ 0 & \sin\theta_3 & \cos\theta_3 \end{bmatrix} \quad R_y(\theta_4) = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 \\ 0 & 1 & 0 \\ -\sin\theta_4 & 0 & \cos\theta_4 \end{bmatrix}$$

$$R = R_x(\theta_3)R_x(\theta_4) = \begin{bmatrix} \cos\theta_4 & 0 & \sin\theta_4 \\ \sin\theta_3\sin\theta_4 & \cos\theta_3 & -\sin\theta_3\cos\theta_4 \\ -\cos\theta_3\sin\theta_4 & \sin\theta_3 & \cos\theta_3\cos\theta_4 \end{bmatrix}$$

2.3 Inverse Kinematics

The inverse kinematics is extracted from the forward kinematics in closed form, steps for extracting d_5 are provided.

$$
\begin{aligned}
&z = (d_5 + ee_w)\cos\theta_3\cos\theta_4 - ee_w \\
&d_5\cos\theta_3\cos\theta_4 = ee_w + z - ee_w\cos\theta_3\cos\theta_4 \\
&d_5 = \frac{z + ee_w}{\cos\theta_3\cos\theta_4} - ee_w
\end{aligned}
\qquad
Inv\_kin : \begin{cases}
d_1 = x + \dfrac{z + ee_w}{\cos\phi\cos\psi}\sin\psi \\
d_2 = y + \dfrac{z + ee_w}{\cos\phi}\sin\phi \\
\theta_3 = \phi \\
\theta_4 = \psi \\
d_5 = \dfrac{z + ee_w}{\cos\phi\cos\psi} - ee_w
\end{cases}
$$

2.4 Robot Control

Communication with the robot was carried out exploiting the TMCM-6212 Trinamic board, a sextuple axis stepper motor controller/driver module for sensorless load dependent current control. The device provides RS485, CAN, USB and 6 encoder interfaces. The 402-CANopen communication protocol was used to interface with the board, computed trajectories were sampled and sent to the board at 1 KHz.

Chapter 3

Developed Software

3.1 Software Architecture

Since the free version of MeVisLab SDK limits the user' python scripts to 2Kb for owned and third-party modules, to avoid the cost of the expensive license, an external server was developed to initialize the communication with the board and control the robot. The server exploit the developed AltairLab library motor-402 [13] for communication with the 402-CANopen® protocol. The server is also responsible for computing the joint trajectories corresponding to the path set on the software interface. APIs implemented on the server are reported in Table 3.1.

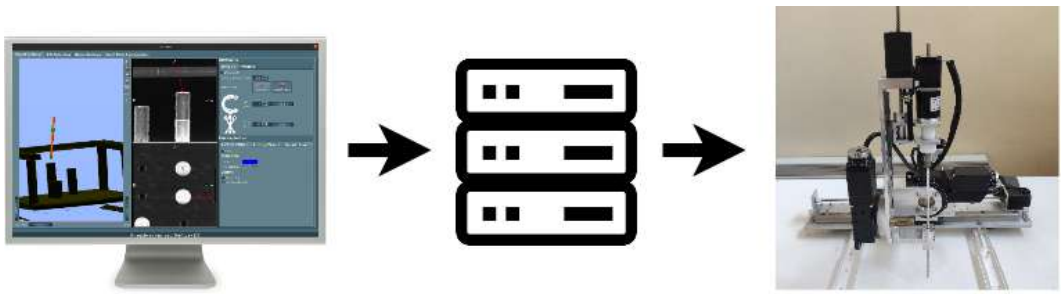


Figure 3.1: Components of the control architecture: a) The user set the needle trajectory on the software interface, the information is sent to the b) server which computes the joint trajectory profiles, c) finally the server writes on the memory registers of the control board to move the robot.

Name	Description
MOVE_ABS	Move the E.E. in a specified pose
MOVE_REL	Move the E.E. of a given pose increment
PIVOT	Set the orientation of the E.E. while maintaining the x, y, z values
GET_POSE	Returns the Cartesian position of the E.E.
GET_CONF	Returns the current joint configuration
MOVE_NEEDLE	Move the fifth joint of a given increment
HOME	Homing functionality, set prismatic joint value to 0 and revolute joint to startup value

Table 3.1: Functionalities implemented on the server.

3.2 Registration

One of the issues while dealing with medical images is the different coordinate systems in which they are expressed. In imaging application there are three commonly used reference systems: the World, Anatomical and the Image coordinate systems.

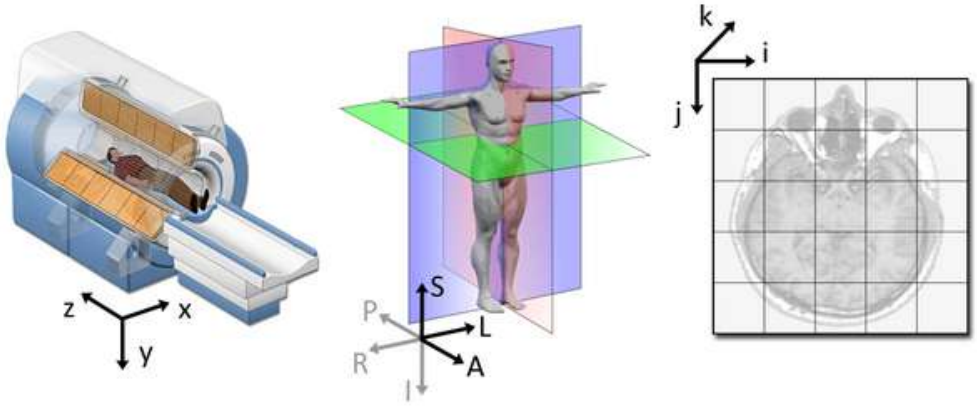


Figure 3.2: Different coordinate systems used in medical images, from left to right the World, Anatomical and Image coordinate systems.

The World coordinate system used in medical images is a Cartesian system and is oriented as shown in Figure 3.2. The Image coordinate system describes how an image was acquired with respect to the anatomy. Medical scanners create regular, rectangular arrays of points and cells which start at the upper left corner of the image. The Anatomical space consists of three planes which describe the standard anatomical position of a

human:

- The axial plane, parallel to the ground and which separates the head (Superior) from the feet (Inferior),
- The coronal plane, perpendicular to the ground and which separates the front from (Anterior) the back (Posterior),
- The sagittal plane, which separates the Left from the Right,

Depending on the medical application, different ways of defining the Anatomical space are used. Possible bases in which the images are defined are LPS and RAS. The one of interest for the study is the LPS base, exploited by the DICOM standard used for CT images.

$$LPS: \begin{cases} \text{From right towards left} \\ \text{From anterior towards posterior} \\ \text{From inferior towards superior} \end{cases} \quad RAS: \begin{cases} \text{From left towards right} \\ \text{From posterior towards anterior} \\ \text{From inferior towards superior} \end{cases}$$

Computerized Tomographies imported in the MeVisLab application are expressed in the MeVisLab coordinate system, and all the framework's modules works with respect to this system. Robot-image registration was necessary to be able to express points on MeVisLab space in the Robot coordinate system. Two different methods to perform registration were implemented.

3.2.1 Registration: Features on Support Frame

The Plexiglas support frame on which the robot is mounted, has a series of cylindrical holes along its perimeter. These holes belong all to the same plane, which is parallel to the xy plane of the Robot reference frame. Since the CT of the phantoms either have an equal sequence of these features, or were acquired with the robot mounting frame already positioned on top of them, these features were extracted from the CT and used for registration purposes. Since no standard is present in the settings of the acquired CT, it was not possible to implement a complete automatic pipeline to detect these holes. The region of interest was identified manually from the whole CT and a procedure based on dilatation $\oplus$ and erosion $\ominus$ morphological operators was used to extract the holes. The erosion of a binary image by a structuring element $I \ominus K$ is an image with ones in all locations (x, y) with value 1 where the structuring element is positioned and completely

fits into the region described by value one, if not the value of (x, y) is set to 0. The dilation of a binary image I by a structuring element K, $I \oplus K$, is in a new binary image with ones in all locations (x, y) where the structuring element is positioned and hits the region of ones of the input image I, if not the value of (x, y) is set to 0. Dilation adds a layer of pixels to both the inner and outer limits of regions, which is the reverse of what erosion does.

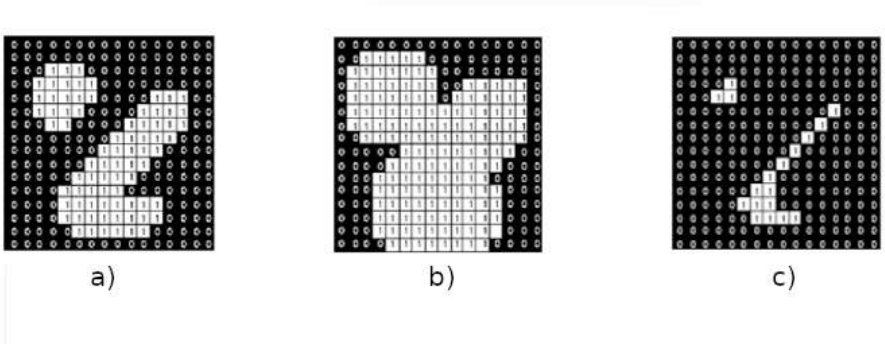
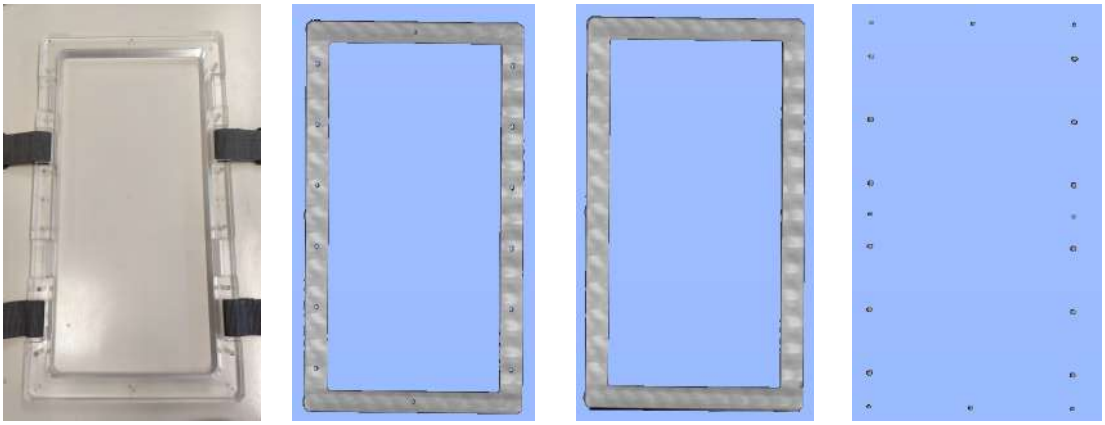


Figure 3.3: Morphological operators: a) is the input image, b) is the dilated image, c) the eroded image, a $3x3$ squared structuring element was used as kernel

$$Hole\_Extraction(Img) = ((Img \oplus K) \ominus K) - Img$$

To extract the volumes relative to the frame holes, the image is first dilatated to fill all the holes, the subsequent erosion removes all the pixels which were previously added to the external layers, thus restoring the region to its original state but with the holes filled. The original image is then subtracted with the image resulting from the dilation-erosion steps. The result is an image with only the cylinders placed in the positions of the holes. The centroid of each cylinder is then set as the feature coordinate to be used to compute the registration matrices.



(a) (b) (c) (d)

Figure 3.4: a) Robot support frame, b) Region manually identified on the TC, c) Region after performing morphological operations, c) Extracted cylindrical features.

SVD for Best Fitting Plane

To compute the plane normal corresponding to the robot z versor, the Singular Value Decomposition SVD was exploited to find the best fitting plane from the given set of points.

The SVD of a matrix $A \in \mathbb{R}^{m \times n}$ is $A = USV^T$, where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, and $S \in \mathbb{R}^{m \times n}$ is a diagonal matrix containing the singular values $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0, \quad r = \min(m, n)$. The plane to be found is the one which produces the smaller orthogonal distances from the set of k points to the plane. The orthogonal distance between a point p_i and the plane can be computed as $(p_i - c)^T n$, where c is a point belonging to the plane and n the plane normal. The plane can then be found by solving the minimization problem

$$\min_{c, \|n\|=1} \sum_{i=1}^{k} ((p_i - c)^T n)$$

Solving for c gives $c = 1/n \sum_{i=1}^{k} p_i$, we now introduce the A matrix $A = [p_1 - c, p_2 - c, ..., p_k - c]$ to formulate the minimization problem as follows

$$\min_{\|n\|=1} \|A^T n\|_2^2$$

We now use the SVD to get to the final equation

$$\|A^T n\|_2^2 = \|VS^T U^T n\|_2^2 = \|S^T U^T n\|_2^2 = (\sigma_1 y_1)^2 + (\sigma_2 y_2)^2 + (\sigma_3 y_3)^2$$

y was set as the unit vector $y = U^T n$, in the equation the matrix V was removed since it is orthogonal and orthogonal matrices do not change the norm of a vector. From the property of the singular values of S, $\sigma_1 \geq \sigma_2 \geq ... \geq \sigma_r \geq 0$, we can deduce that the problem is minimized for $y = [0, 0, 1]^T$, or equivalently for $n = U(:, 3)$. With the described algorithm, it is possible to compute the direction of the plane normal but not its verse. Since the CT were assumed in different modalities, the verse of the normal was set by hand case by case.

RANSAC for Line Fitting

Random Sample Consensus algorithm (RANSAC) is an iterative method to estimate parameters of a mathematical model from a set of observed data that contains outliers, entries which have been wrongly estimated and can't be explained by the model. The algorithm randomly takes the minimum number of points to describe the mathematical model, it then validates the assumed model with the remaining points. A point will be considered an outlier if it does not fit the model within some error threshold defined by the user. At each iteration a new random model will be assumed, if this model will describe better the data than the previous assumed one, it will be considered as the best up to that point. The method terminates when the maximum number of iterations, parameter defined by the user, is reached, returning the model which best describes the data. The x versor of the robot reference frame was computed applying RANSAC to the set of extracted points. The model to estimate was the one of the line $l = c + \lambda d$. A point was considered an outlier if it had a distance to the assumed line greater than $1cm$, the algorithm was run for 100 iterations.

As for the plane normal verse, also the direction of the x versor was adjusted by hand case by case. Finally, the robot y versor was computed as the cross product $y = z \times x$.

Homogeneous Transformation Matrices

Computations to retrieve the transformation matrices to go from the MeVisLab to the Robot reference system ${}^{Rob}T_{Mev}$ and vice versa ${}^{Mev}T_{Rob}$, are now presented. The marker p closest to the robot home position was identified by hand and was used to compute the transformation for going from the so called Marker reference system, oriented as the Robot system but positioned in a different position, to the MeVisLab reference frame. The direction versors x, y, z retrieved with SVD and RANSAC, were used to express the

orientation of the new reference system.

$$ {}^{Mev}T_{Mark} = \begin{bmatrix} x^{3\times1} & y^{3\times1} & z^{3\times1} & p^{3\times1} \\ 0 & 0 & 0 & 1 \end{bmatrix}^{4\times4} $$

Displacement δ between the Marker and Robot reference system was manually computed on the real robot, the matrix ${}^{Mev}T_{Rob}$ was then retrieved.

$$ {}^{Mark}T_{Rob} = \begin{bmatrix} 1 & 0 & 0 & \delta_x \\ 0 & 1 & 0 & \delta_y \\ 0 & 0 & 1 & \delta_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{Mev}T_{Rob} = {}^{Mev}T_{Mark}\,{}^{Mark}T_{Rob} $$

The transformation matrix ${}^{Rob}T_{Mev}$ was computed inverting the matrix ${}^{Mev}T_{Rob}$.

$$ T = \begin{bmatrix} R^{3\times3} & t^{3\times1} \\ 0^{1\times3} & 1 \end{bmatrix} \qquad T^{-1} = \begin{bmatrix} R^{-1} & -R^{-1}t \\ 0 & 1 \end{bmatrix} $$

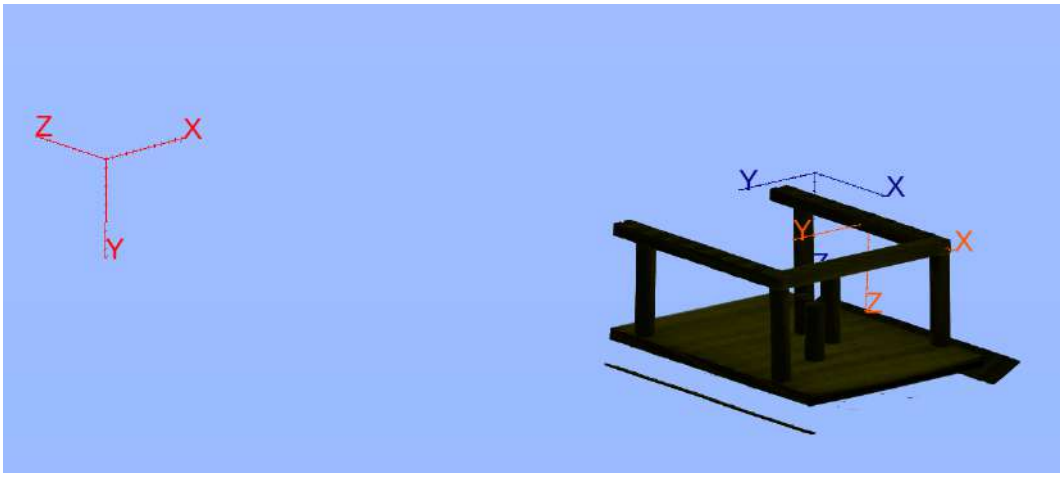


Figure 3.5: CT of the Plexiglas phantom with 3 pillars visualized on the MeVisLab application. In orange the Marker reference frame, in blue the robot reference frame and in red the MeVisLab reference frame.

3.2.2 Registration: Point and Click Method

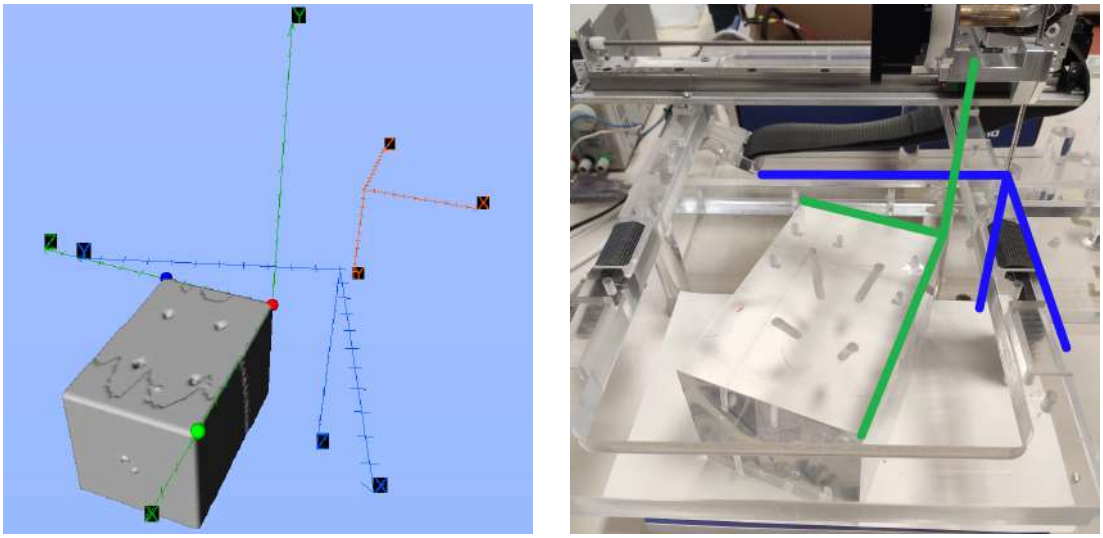
The registration method explained in Section 3.2.1 assumes that the position of the robot frame do not change in the time interval going from the CT acquisition to when the robot will finish operating. If changes in the robot happen in such temporal interval,

correct registration will not be possible since the system will have no way of recognizing that such displacement happened.

To provide an alternative solution in such eventualities, a different registration method is proposed. Three pair of points $({}^{M}p_i, {}^{R}p_i)$ were chosen, ${}^{M}p_i$ being a point expressed in the MeVisLab reference system and ${}^{R}p_i$ the same point but expressed in the Robot reference system. A new support Calibration reference system was positioned on the first point p_1, its orientation was determined by using the other remaining points p_2, p_3. The transformations matrices ${}^{Mev}T_C$ and ${}^{Rob}T_C$ for going from the MeVisLab and Robot reference system to the Calibration reference system were computed using the three points expressed in the relative system. Finally, the ${}^{Mev}T_{Rob}$ was set as ${}^{Mev}T_C \cdot ({}^{Rob}T_C)^{-1}$. Mathematical steps are reported, the superscript $*$ indicates the reference system in which the variables are expressed, Mev or Rob.

$${}^{*}\vec{x} = ({}^{*}p_2 - {}^{*}p_1)/\|({}^{*}p_2 - {}^{*}p_1)\| \quad {}^{*}\vec{y} = getPlaneNormalSVD({}^{*}p_1, {}^{*}p_2, {}^{*}p_3) \quad {}^{*}\vec{z} = {}^{*}\vec{x} \times {}^{*}\vec{y}$$

$${}^{*}T_C = \begin{bmatrix} {}^{*}\vec{x} & {}^{*}\vec{y} & {}^{*}\vec{z} & {}^{*}p_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad {}^{Mev}T_{Rob} = {}^{Mev}T_C({}^{Rob}T_C)^{-1} \qquad {}^{Rob}T_{Mev} = ({}^{Mev}T_{Rob})^{-1}$$



(a) (b)

Figure 3.6: Registration using three pairs of points. a) Reference systems visualized in the software application, in orange the MeVisLab reference frame, in green the Calibration one and in blue the position of the Robot reference system. The showed spheres indicate the used points in the calibration algorithm. b) The position of the object with respect to the robot, calibration and robot reference frames were drawn, as can be seen they coincide with the ones in the MeVisLab application.

3.3 Software Interface

In non robot assisted vertebroplasty, the surgeon use orthogonal C-Arms to have feedback on the position of the needle while operating. Since the fluoroscopy compress the 3D volume in plane representation, a segment on a fluoroscopy does not indicate a line but instead a plane, oriented as the segment and normal to the image. For this reason, the doctor exploits two fluoroscopies acquired from different perspectives and, relating the needle projections on the two image, he is able to determine the linear path of the needle.

The modality for designing the trajectory on the software interface takes inspiration from this methodology. On two fluoroscopies simulated from the CT, the user can draw two segments, each one representing the orientation of a plane normal to the image. The linear trajectory of the robot needle is then set as the intersection of these two planes. Further details of the procedure will be explained in Section 3.3.1.

The application implemented with the MeVisLab SDK is composed of four windows, *Trajectory Planning, ROI Selection, Robot Control* and *Point & Click Registration.*

3.3.1 Trajectory Planning

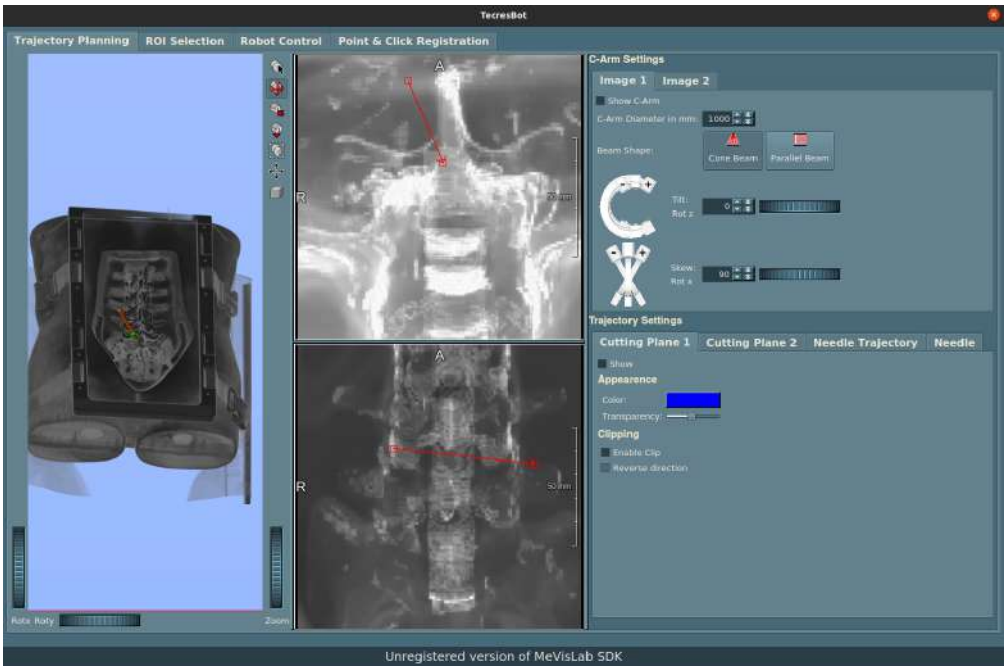


Figure 3.7: Trajectory Planning tab

On the *Trajectory Planning* tab of the application, the user can design the trajectory to be performed. The panel is composed by a main window where the chosen trajectory and the CT of the phantom can be seen, a pair of views with the two simulated fluoroscopies and by the *C-Arm Settings* and *Trajectory Settings* control panels.

C-Arm Settings

By click and dragging on the fluoroscopies, the user can draw two segments corresponding to the planes normal to the images. Interacting with the controls on the panel, the C-Arms relative to the shown fluoroscopies can be visualized on the main window. By changing the orientation of the C-Arms with the dedicated controls, different fluoroscopies can be produced. The line resulting from the intersection of the two drawn planes was computed as follows.

$$d = n_1 \times n_2 \qquad line = c + \lambda d$$

Every point belonging to the line can be chosen as the line centre c. To extract a point on the line, the algorithm illustrated in [10] was implemented. The method computes

the point on a line which is closest to some arbitrary point p_0. The matrix equation for solving the problem is provided, (p_i, n_i) are the point and normal of the plane i. q is the point on the line to be found.

$$\begin{bmatrix} 2 & 0 & 0 & n_{1x} & n_{2x} \\ 0 & 2 & 0 & n_{1y} & n_{2y} \\ 0 & 0 & 2 & n_{1z} & n_{2z} \\ n_{1x} & n_{1y} & n_{1z} & 0 & 0 \\ n_{2x} & n_{2y} & n_{2z} & 0 & 0 \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_z \\ \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} 2p_{0x} \\ 2p_{0y} \\ 2p_{0z} \\ p_1 \cdot n_1 \\ p_2 \cdot n_2 \end{bmatrix}$$

The point q was thus set as the line's centre. The projections of the two plane's centre to the line were then computed. The mean of the two projections was set as the source point s of the linear trajectory. The trajectory target point was computed as $t = s + \lambda d$, with λ having as the default value, the mean of the two drawn segments used to identify the planes on the fluoroscopies. Both s and t position can be shifted along the line direction with dedicated controls.

Trajectory Settings

The *Trajectory Settings* panel is divided into multiple tabs. The *Cutting Planes* tab allows visualizing the planes drawn on the fluoroscopies, the clipping option allows to remove all the part of the CT on one side of the plane to see sections of the image. In the *Needle Trajectory* and *Needle* tab, the user can hide the needle showed on the main window and customize its appearance. Each trajectory designed on the application will be divided in three sub trajectories:

- The robot needle will position itself on the source point, showed on the application as a sphere.
- On the source point, the needle will perform pivoting to reach the desired orientation, identified by the intersection of the two drawn planes.
- Finally, the needle will shift along the trajectory direction to reach the target point, which corresponds to the needle end.

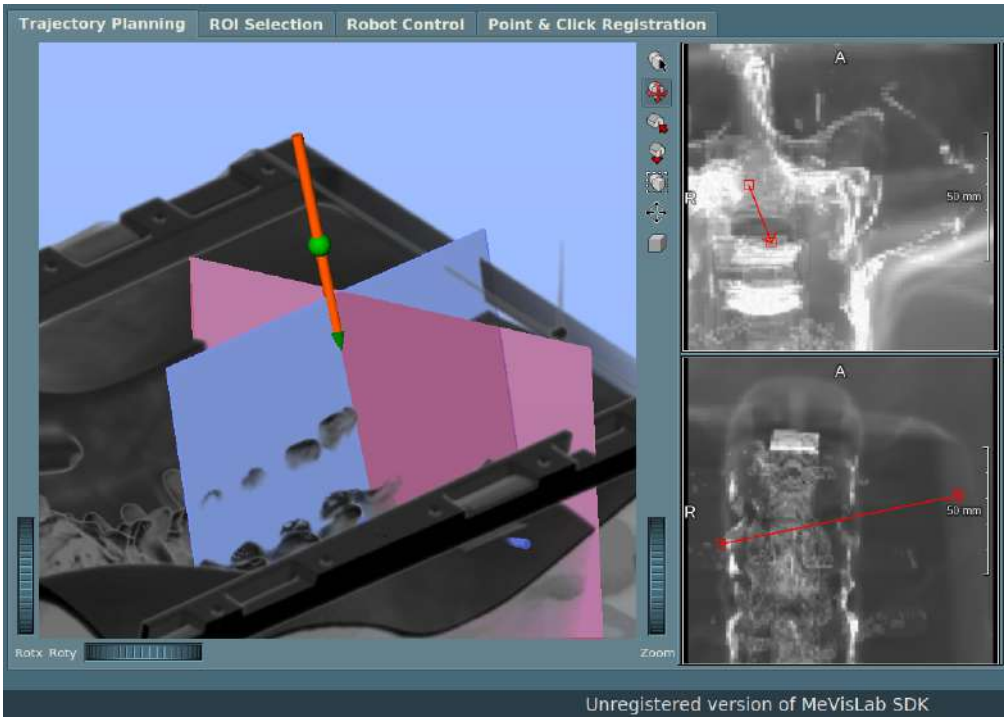


Figure 3.8: Visualization of the planes for designing the needle trajectory, the sphere on the orange needle indicates the source point, the green pointy end indicates the trajectory target point.

The *Needle Trajectory* tab is equipped with controls for shifting the source and target point along the trajectory direction. Given ${}^{Mev}d, {}^{Mev}s, {}^{Mev}t$, direction vector and source and target points expressed in MeVisLab reference system, the transformation matrix ${}^{Rob}T_{Mev}$ was used to expressed them in the Robot reference system.

$$ {}^{Rob}s = {}^{Rob}T_{Mev} {}^{Mev}s, \quad {}^{Rob}t = {}^{Rob}T_{Mev} {}^{Mev}t, \quad {}^{Rob}d = {}^{Rob}R_{Mev} {}^{Mev}d $$

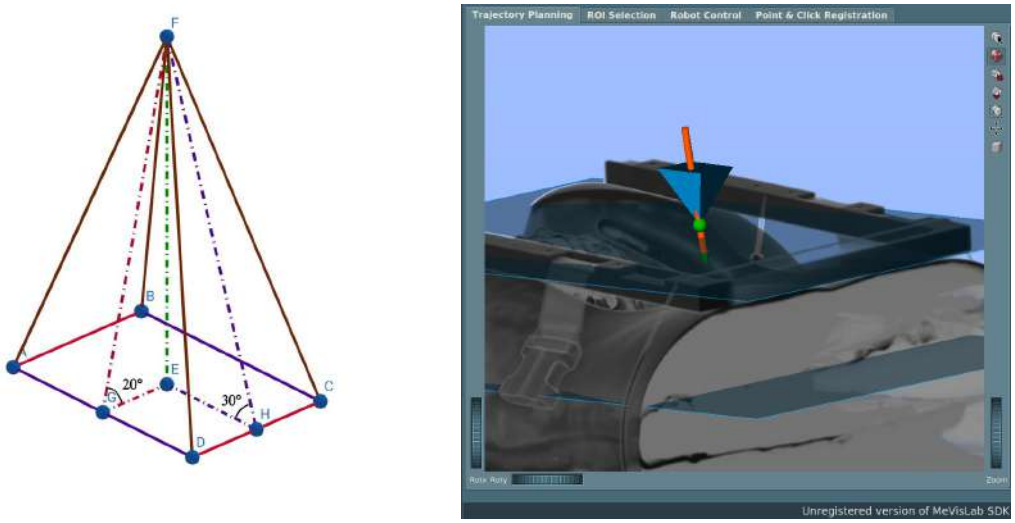
$$ {}^{Rob}\phi = \arctan_2({}^{Rob}d_z, {}^{Rob}d_y), \quad {}^{Rob}\psi = \arctan_2({}^{Rob}d_x, {}^{Rob}d_z) $$

Trajectory Limits Visualization

If the designed trajectory exceeds the joint limits, a message will be prompted in the *Robot Control* tab and the commands for moving the robot will be disabled. To help the user understand how to edit the trajectory and make it feasible, a joint limits visualization option was implemented. The tool consists of a rectangular based pyramid positioned on the trajectory source point, which shows the limit orientations the robot can assume in that point. The 3D polygon dimensions were set so that the cone would span along the x and y directions angles equal to the θ_4 and θ_5 joint limits. To show the minimum and maximum z value reachable by the robot with the given orientation, two planes were added. Considering the Robot reference system, both planes are parallel to the one identified by the xy axes. The first plane has as z coordinate value 0, the z value of the second depends instead on the robot orientation values. The more the two rotational joints (θ_3, θ_4) rotate, the more constrained the robot's workspace along the z axis will be. The z value of the second plane can be obtained using the inverse kinematics by setting d_5 to its joint limit value.

$$ plane_z = (d_5\_lim + ee_w) \cos\theta_3 \cos\theta_4 - ee_w $$

A trajectory is deemed feasible if both the source and target points lie in the region described by the two planes and whether the visualized needle passes through the pyramid base.



(a) (b)

Figure 3.9: a) Rectangular pyramid which spans the joint limits angles. b) Planes and pyramid visible when the trajectory limits visualization option is turned on.

The pivoting primitive changes the orientation of the needle while maintaining the current position. To achieve such goal, motion of the prismatic joints d_1, d_2, d_5 are necessary to balance the changes in position due to the movement of the rotational joints θ_3, θ_4. Thus, given a source pose, not all the orientations within the rotational joint limits are reachable with the pivoting primitive, since not enough range of motion of the prismatic joints could be available. An example is provided to better clarify the concept. Let's assume we want to pivot the robot needle to obtain an orientation equal to $(0, \alpha)$ respectively for (ϕ, ψ). As illustrated in Figure 3.10, enough range of motion $x_{lim} - s_x$ along the x axis is needed so that the right triangle having as vertices the wrist centre of motion, the trajectory source point and the point ${}^{Rob}(x_{lim}, s_y, -ee_w)$, would form an angle Ψ greater or equal than the desired orientation α. To visualize this concept on the software application, four invisible clipping planes were used. These planes are positioned at the robot workspace limits and their orientation describe the maximum orientation values the robot can assume in the selected source point while performing pivoting. This maximum orientation value corresponds to Ψ on Figure 3.10. The intersection of these planes with the rectangular based pyramid, described as the blue region of the figure, corresponds to non reachable orientation values as thus is hidden to the user.

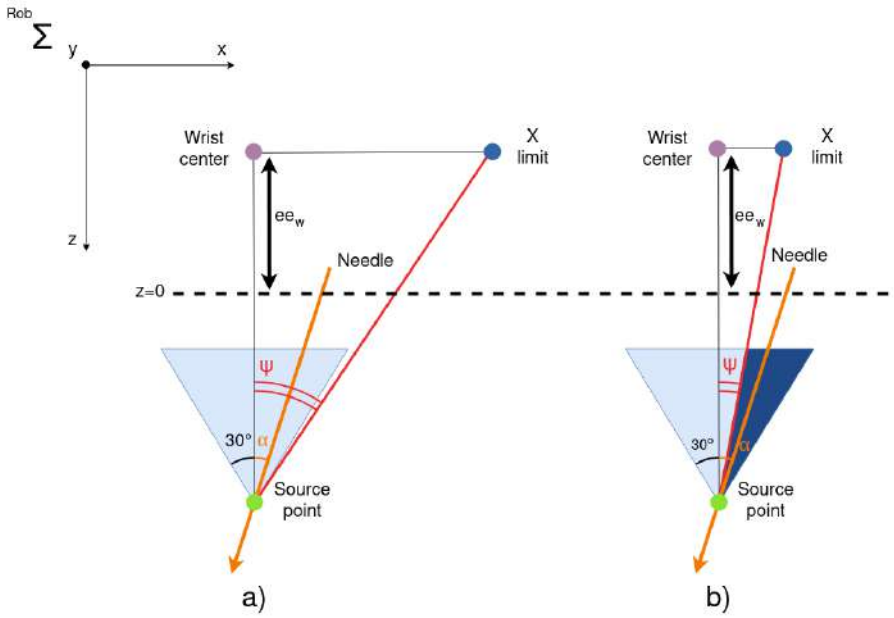
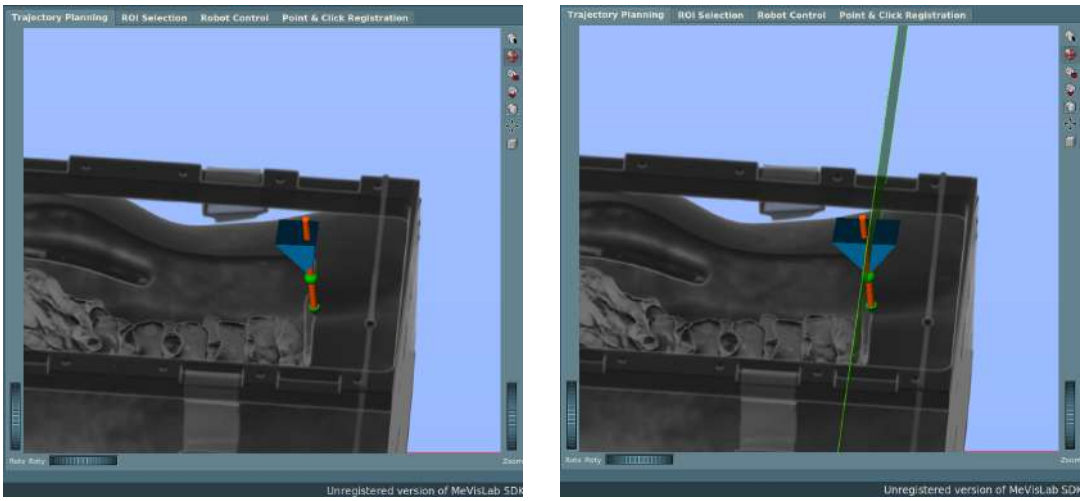


Figure 3.10: Example to illustrate the pivoting primitive. Figure a) shows a feasible pivoting, Figure b) shows a non-possible pivoting even if the desired orientation α is within the joint limit described by the rectangular based pyramid.



(a) (b)

Figure 3.11: a) Clipped pyramid since the designed trajectory is near to the d_2 joint limit. b) Debug visualization which shows the plane responsible for the clipping.

3.3.2 ROI Selection

The *ROI Selection* tab allows selecting the portion of the CT to be used for extracting the fluoroscopies. The selected region of interest will be visible in the main window of the *Visual Settings* tab.

Figure 3.12: ROI Selection tab

3.3.3 Robot Control

Once the trajectory has been set on the *Trajectory Planning* interfaces, robot movement can be instructed on the *Robot Control* tab. Robot homing can be commanded with the *Go Home* button, the *Go To Source* section visualizes the source point s coordinates in Robot Space. By clicking the corresponding button, the robot will move first in $(s_x, s_y, 0)$, then in (s_x, s_y, s_z) and finally it will pivot to reach the desired orientation. The *Shift to Target* section shows the necessary shift to be performed to reach the target position t from the source point s. If the chosen trajectory exceeds the robot joint limits, an error message will be prompted and interaction with the robot will be restricted. On the *Relative Robot Motion* section, the user can jog the robot at will.

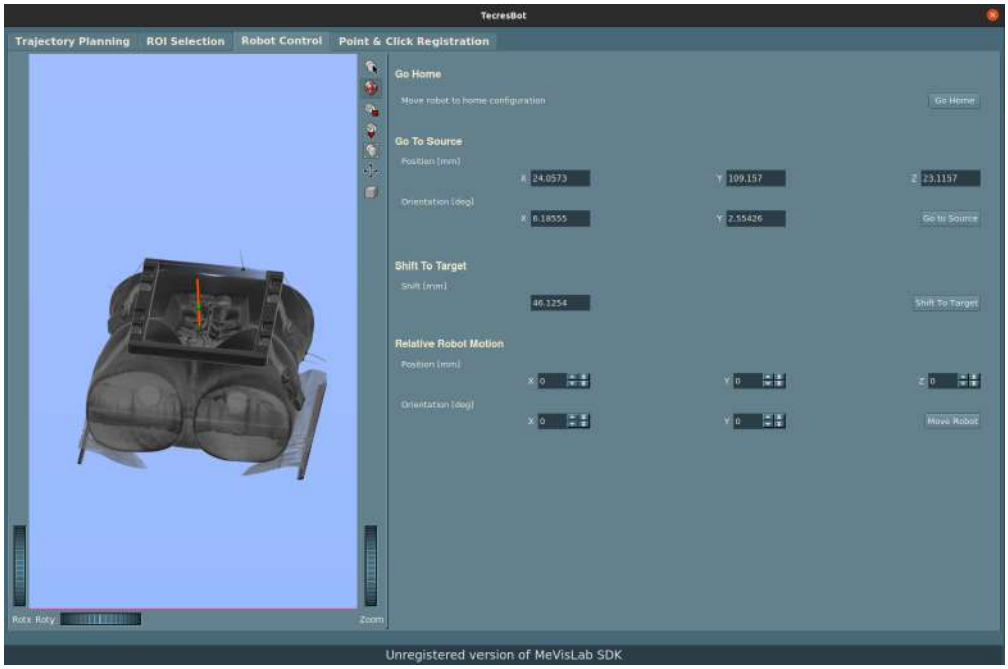


Figure 3.13: Robot Control tab

3.3.4 Point & Click Registration

The *Point & Click Registration* offers an interface to pick the necessary points for the registration method described in Section 3.2.2. The user have to select three points both on the Software and on the Robot space. The points on MeVisLab can be selected by *Alt+Right Clicking* on the object on the main window. A pink sphere will appear indicating the selected point, with the *Confirm on MeVisLab* button the user can confirm the selection. Once clicked, the pink sphere will turn into a different colour, showing that the point has been correctly acquired. To choose the same point on the Robot space, the user can jog the robot with the controls on the *Relative Robot Motion*. Once the robot has reached the desired point, acquisition must be confirmed with the *Confirm on Robot* button. The field *Point Index Selection* goes from 1 to 3 and identify the index of the points necessary for the registration algorithm.

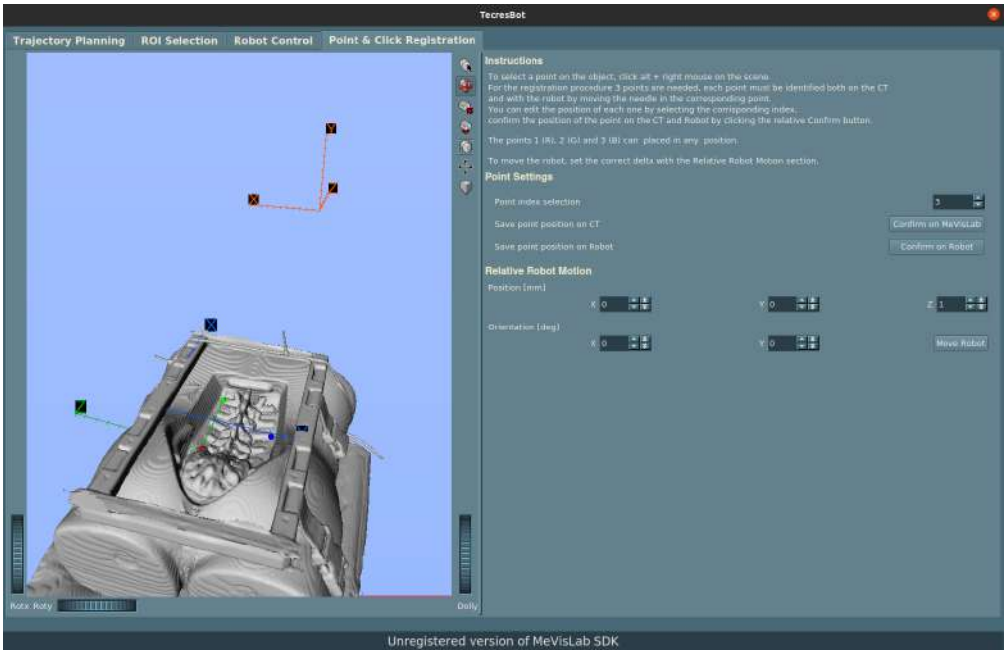


Figure 3.14: Point & Click Registration Tab, in red the MeVisLab reference frame, in green the Calibration reference frame, in blue the Robot Reference frame, the coloured spheres indicate the selected points used for registration.

3.4 Pivoting Primitive

With the Analytical Jacobian of a robot manipulator, we can relate how changes in joint velocities $\dot{q}$ produce changes in Cartesian velocities $\dot{x}$.

$$J_A \dot{q} = \dot{x}$$

From this equation, we can deduce that the null space of the first three rows of the analytical Jacobian $N(J_A)$, explains how variations in the joint velocities $\dot{q}$ produce no variations in the Cartesian velocities along x, y, z. The joint trajectories for the pivoting primitive were computed as a linear combination of the bases b_1, b_2 which describe the null space of $J_A[0:3,:]$.

$$b_1(\theta_3, \theta_4, d_5) = \begin{bmatrix} \dfrac{(-d_5 - ee_w)\sin\theta_4}{\sin\theta_3} \\ (d_5 + ee_w)\cos\theta_4 \\ \dfrac{\tan\theta_4}{\tan\theta_3} \\ 1 \\ 0 \\ 0 \end{bmatrix} \quad b_2(\theta_3, \theta_4, d_5) = \begin{bmatrix} \dfrac{(d_5 - ee_w)\cos\theta_4}{\cos\theta_3} \\ (d_5 + ee_w)\sin\theta_4\tan\theta_3 \\ 1 \\ 0 \\ (d_5 + ee_w)\tan\theta_3 \\ 0 \end{bmatrix}$$

Given a starting configuration q_i, to perform pivoting and reach the desired orientation described by q_f, polynomials of fifth grade were used to compute the velocity trajectories $\dot{\phi}, \dot{\psi}$ to go from ϕ_i to ϕ_f and from ψ_i to ψ_f. The overall joint profiles were then computed as

$$\dot{q} = \dot{\psi} b_1(q_i) + \dot{\phi} b_2(q_i) \qquad q = \int_0^{t_f} \dot{q}\, dt$$

A linear combination of the two bases is guaranteed to lie within the considered analytical Jacobian kernel, thus the computed trajectory will not produce any changes in (x, y, z) but only in (ϕ, ψ).

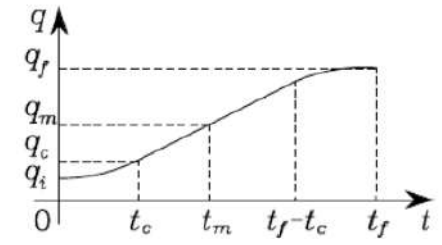
3.5 Joint Trajectory Profile

Except for the pivoting primitive, joint trajectories were designed with a trapezoidal velocity profile. This profile is characterized by a starting and an ending parabolic blend in position, by a trapezoidal profile in velocity and by a squared wave like profile in acceleration. These characteristics allow a constant acceleration in the starting phase, a constant cruise velocity in the middle segment and a constant deceleration in the arrival phase. This type of trajectory is often used in industrial robots given the continuity of the velocity profile.

Definition of the variables and parameters necessary for definition of the trajectory are reported in table 3.2, time laws for computing position, velocity and acceleration time laws are provided.

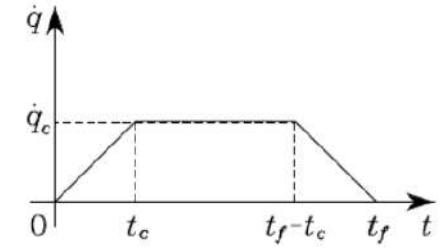
Variables	Value	Definition
q_i, q_f	Parameter	Initial and final joint positions
$\dot{q}_i, \dot{q}_f$	Parameter	Initial and final joint velocities
tc	Parameter	Acceleration and deceleration time interval
σ	$sign(q_f - q_i)$	Sign variable
$\hat{q}_i, \hat{q}_f$	$\sigma q_i, \sigma q_f$	Position for either case $q_f > q_i$ or $q_f < q_i$
$\ddot{q}_c$	$\dfrac{\hat{q}_f - \hat{q}_i}{t_c t_f - t_c^2)}$	Constant cruise acceleration
$\dot{q}_c$	$\ddot{q}_c t_c$	Constant cruise velocity

Table 3.2: Parameters and variables for computing the trapezoidal velocity profile



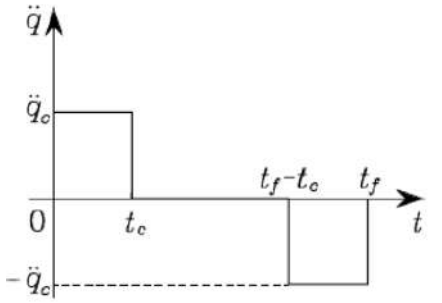
$$\hat{q}(t) = \begin{cases} \hat{q}_i + \dfrac{\dot{q}_c}{2t_c}t^2 & \text{for } 0 \le t \le t_c \\ \hat{q}_i + \dot{q}_c(t - \dfrac{t_c}{2}) & \text{for } t_c < t \le t_f - t_c \\ \hat{q}_f - \dfrac{\dot{q}_c}{2t_c}(t_f - t)^2 & \text{for } t_f - t_c < t \le t_f \end{cases}$$

$$q(t) = \sigma\hat{q}(t)$$



$$\hat{\dot{q}}(t) = \begin{cases} \dfrac{\dot{q}_c}{t_c}t & \text{for } 0 \le t \le t_c \\ \dot{q}_c & \text{for } t_c < t \le t_f - t_c \\ \dfrac{\dot{q}_c}{t_c}(t_f - t) & \text{for } t_f - t_c < t \le t_f \end{cases}$$

$$\dot{q}(t) = \sigma\hat{\dot{q}}(t)$$



$$\hat{\ddot{q}}(t) = \begin{cases} \dfrac{\dot{q}_c}{t_c} & \text{for } 0 \le t \le t_c \\ 0 & \text{for } t_c < t \le t_f - t_c \\ -\dfrac{\dot{q}_c}{t_c} & \text{for } t_f - t_c < t \le t_f \end{cases}$$

$$\ddot{q}(t) = \sigma\hat{\ddot{q}}(t)$$

Table 3.3: Position, velocity and acceleration motion laws of the trapezoidal velocity profile with the relative illustrations

For each joint trajectory, initial and final positions q_i, q_f were computed with the inverse kinematics, initial and final velocities $\dot{q}_i, \dot{q}_f$ were set to 0, m/s or rad/s respectively for prismatic and revolute joints, the overall trajectory time $\Delta t = t_f - t_i$ was set to $5s$ and acceleration and deceleration time t_c were set equally to $1s$.

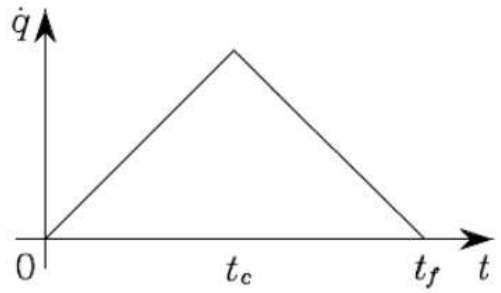


Figure 3.15: Limit case of the trapezoidal velocity profile

The set values for Δt and t_c guarantee the presence of the trapezoidal shaped velocity profile, given that the condition $tc \leq \frac{t_f - t_i}{2}$ is always satisfied. Figure 3.15 shows the limit case with no constant velocity cruise phase and where the velocity assumes a triangular shaped profile.

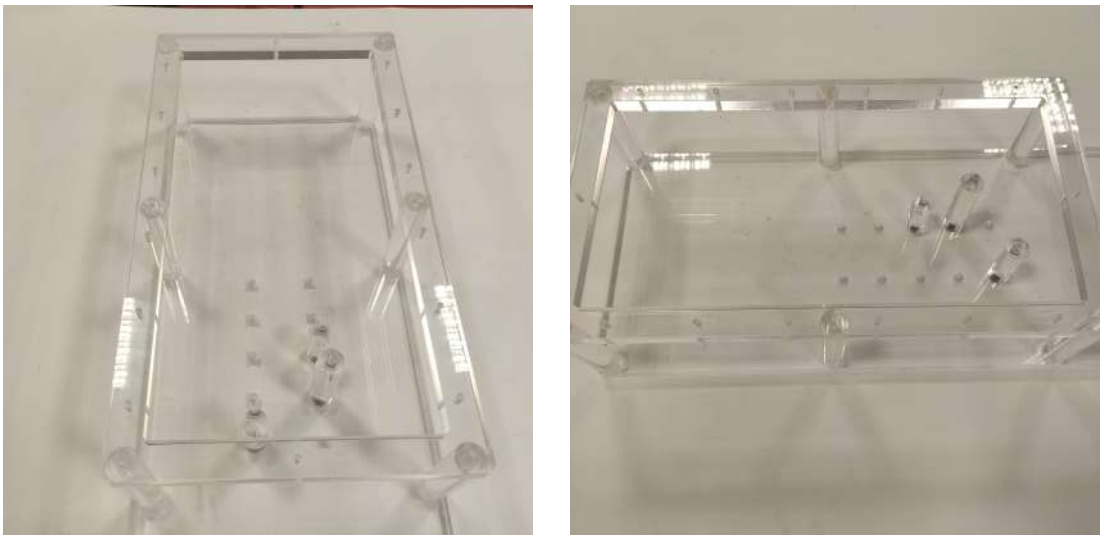
Chapter 4

Experiments and Results

To test the developed system, salient features of three different phantoms were set as the target point of trajectories designed on the software interface.

4.1 Plexiglas Phantom With Pillars

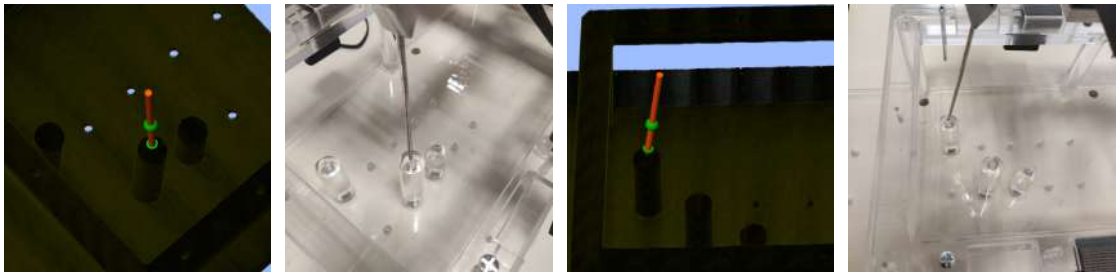
The first phantom used for testing the architecture was the Plexiglas phantom illustrated in Figure 4.1. The phantom



(a) (b)

Figure 4.1: Plexiglas phantom with three pillars.

The object is shaped as a rectangular parallelepiped, it is equipped with three pillars of different heights and it is provided with the same patters of holes of the Robot support frame. This sequence of holes was exploited to guarantee a rigid mounting point for the Robot, fixation was performed with screws passing through both holes of the mounting frame and of the phantom. Method of Section 3.2.1 was chosen to perform image registration. Test trajectories with target point the centre of the circular surface of each pillar were carried out.

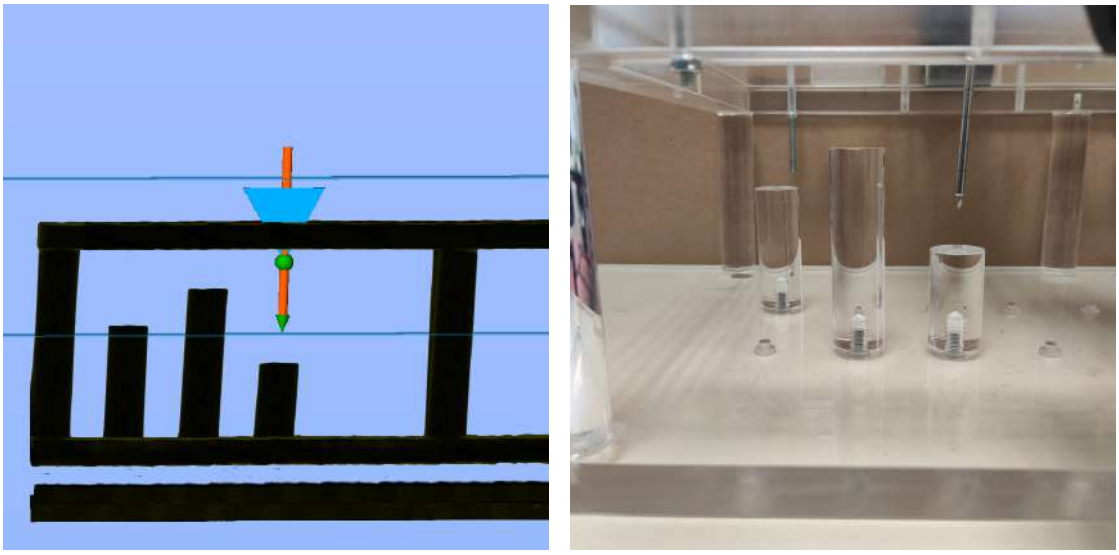


(a) (b) (c) (d)

Figure 4.2: Trajectories for the tallest and medium height pillars, planning on the software and trajectories carried out by the robot.

Two of the three targets were successfully reached, the one corresponding to the lower pillar could not be touched since it lied outside the reachable workspace of the robot. The non feasibility of the trajectory was correctly notified to the user on the software interface as shown in Figure 4.3.

With this phantom we could prove the correctness of the registration method based on the extraction of salient features 3.2.1. In this method, it is mandatory that the relation between the robot position and the location of attachment of the mounting frame is rigid throughout the whole operation. The markers which identify the mounting point on the CT must also not change in the time interval between the X-ray acquisition and the end of the surgery.



(a) (b)

Figure 4.3: Limit depth along the trajectory to reach the shorter pillar.

4.2 Body Phantom

To understand if the current prototype mechanic is suited for perform realistic assistance in vertebroplasty on humans, a body shaped phantom was exploited.

Figure 4.4: Body phantom.

Registration using features on the robot frame

The CT of the body was acquired with the robot mounting frame already positioned on top of the phantom, having taken care of marking the frame position before the image acquisition. Even though precautions were made, correct registration with the method described in Section 3.2.1 was not possible. This was due to involuntary changes in the robot frame position happened between the robot frame fixation and the CT acquisition.

Registration based on point selection

Given the poor performances of the registration method based on the extraction of features, the method described in section 3.2.2 was instead used with this phantom. We emphasize that precision in the selection of the points needed for registration is of most importance for the correctness of the method. The phantom is not provided with small enough and distinctive features to uniquely define these points with the necessary precision, thus small errors in the registration were most probably made.

(a) (b)

Figure 4.5: Trajectory performed on the body phantom.

Trajectories with target the backbone processes were carried out, the points were reached with heterogeneous errors in position along x, y and z measured at most of $2mm$. Errors were measured with a digital calibre, comparing the target position of the software interface with the one reached by the robot on the phantom.

Target in Robot space $[x, y, z, \phi, \psi]$	x error	y error	z error
$[2.73, 80.15, 72.70, 10.71°, -3.10°]$	0.1	0.2	0
$[37.93, 77.18, 75.50, 9.75°, -1.48°]$	0.1	0.1	1.9
$[71.05, 73.01, 74.97, 5.25°, -5.68°]$	0.3	0.6	1.1
$[91.06, 77.09, 76.16, 8.26°, -2.53°]$	0	0.0	1.0
$[121.38, 80.34, 70.73, -16.01°, -6.98°]$	1.3	0.2	2.1
Mean Error	0.36	0.22	1.22

Table 4.1: Errors in target position [mm] of trajectories carried out on the body phantom, registration method based on 3 points.

It was observed that the errors in position increase the more the robot furthest itself from its home position. These can be explained by deformations of the phantom skin caused by the robot weight shifting while it moves. As can be seen in Figure 4.6, the bigger limitation of the robot consists in the length of the fifth joint d_5, which does not allow to reach the vertebral bodies. Given the realistic dimensions of the phantom, we were able to check the compatibility of the robot with the human anatomy. We came to the conclusion that current prototype is not suited for robotic assistance in vertebroplasty.

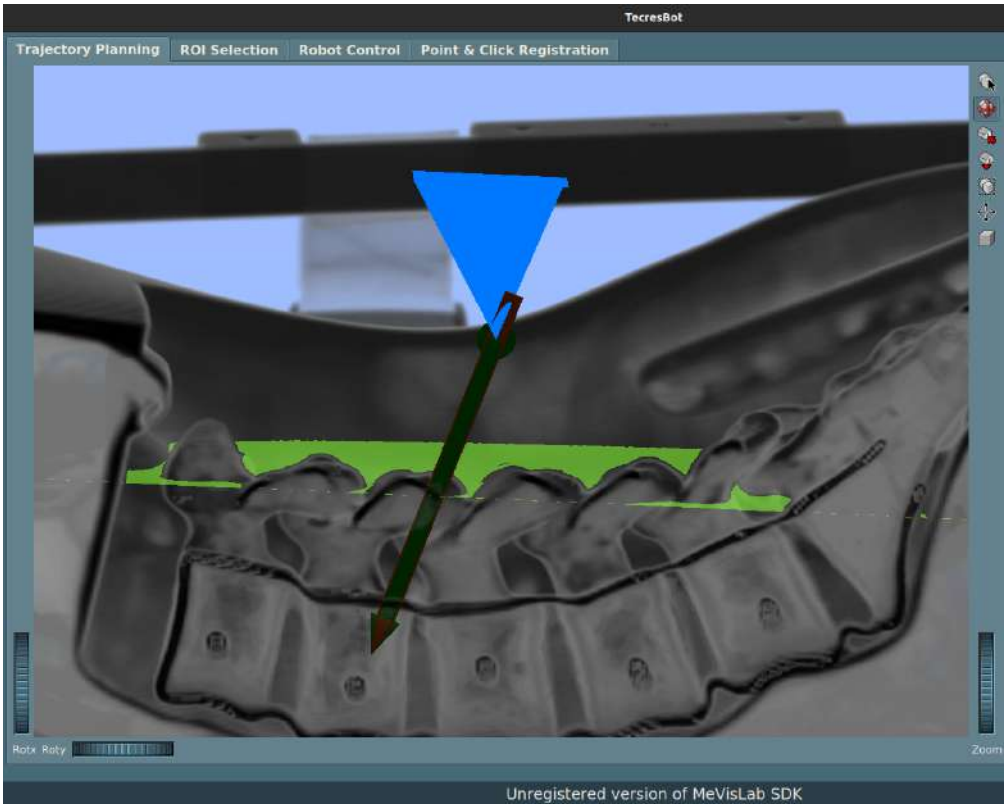
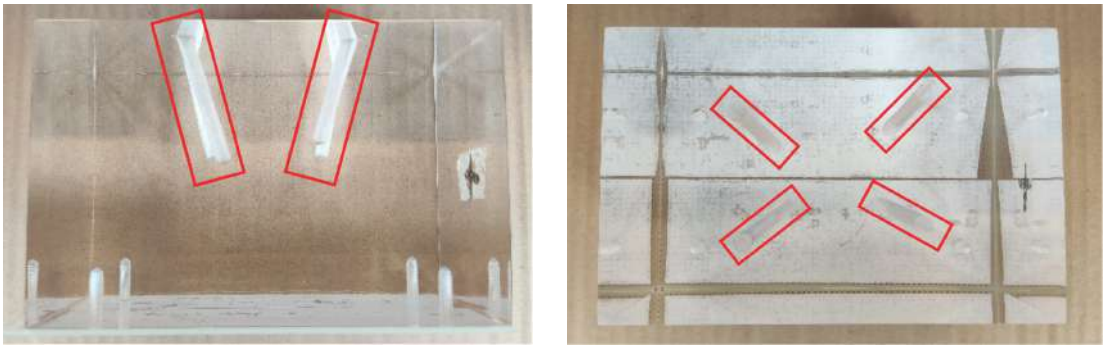


Figure 4.6: Section of the body phantom, the designed trajectory simulates the usual path the needle follows during vertebroplasty. The light green plane indicates the maximum depth reachable by the robot with the showed orientation. Vertebral bodies can't be reached with the current robot.

4.3 Plexiglas Prism Phantom

The last phantom used for the study is a Plexiglas prism on which holes were drilled in an inclined way. The drilled holes have a circular base of $5mm$ in diameter and vary in length, Figure 4.8 shows the dimensions of the longer and more inclined one. Registration based on the method explained in Section 3.2.2 was used. The Robot frame was fixed on a separate rigid structure to avoid problems of deformation of the surface where the robot lied. The prism was not moved throughout the experiments.



(a) (b)

Figure 4.7: Prism phantom seen a) from a lateral perspective and b) from the top.

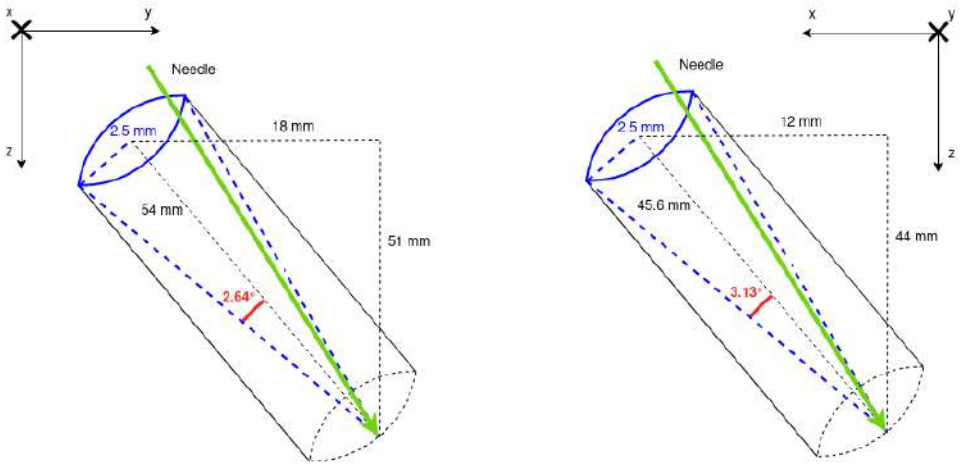
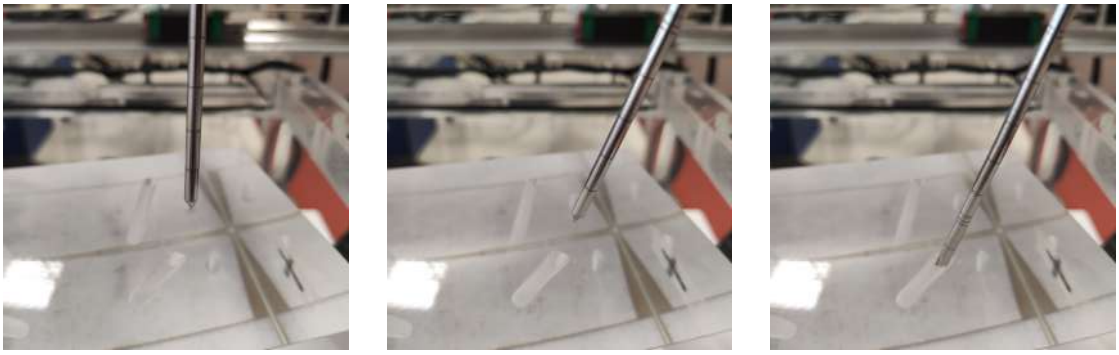


Figure 4.8: Dimensions of the bigger and more inclined hole among the four of the prism. Coordinates expressed in Robot reference system.

4.3.1 Robot Precision in Orientation

Executing trajectories going through the four holes of the prism, we could get an understanding of the robot precision in the orientation around x and y. Trajectories which did not touch the holes boundaries were designed on the software interface and carried out by the robot. Each one of the four trajectories reached the bottom of the hole, with two of them requiring a slight readjust of the trajectory's source position of $1mm$ to achieve no contact with the prism. Holes and needle measure respectively $5mm$ and $3.6mm$ in diameter, thus even small errors in the registration procedure or in the planning on the software interface, will correspond to trajectories which will touch the holes boundaries. Given that only the position of the trajectory needed adjustments, we can safely state that the error in orientation is equal at most to the angles described by the cones with dimensions the base and depth of the holes. The angles in red of Figure 4.8 describe the precision in orientation for ϕ and ψ, respectively $2.64°$ and $3.13°$.



(a) (b) (c)

Figure 4.9: Trajectory going through a hole of the prism. a) The needle is positioned on the source point, b) pivoting is performed to obtain the desired orientation, c) finally the needle is shifted to the target point.

4.3.2 Robot Precision in Position

On the opposite face to where the holes were drilled, a stripboard was glued to the prism. This component offers a grid of small holes of $1mm$ of diameter distanced $2.5mm$ from each other. The stripboard measures $160 \times 100mm \times 1.4$ and is provided with a pattern of 61×38 holes. 3 points identified by these holes were used to register the robot with the CT image. The stripboard was divided in a grid 4×3, with each box measuring

$37.5 \times 30mm$ and composed by 15 holes along the bigger dimension of the stipboard and by 12 holes along the other, division illlustrated in Figure 4.11.

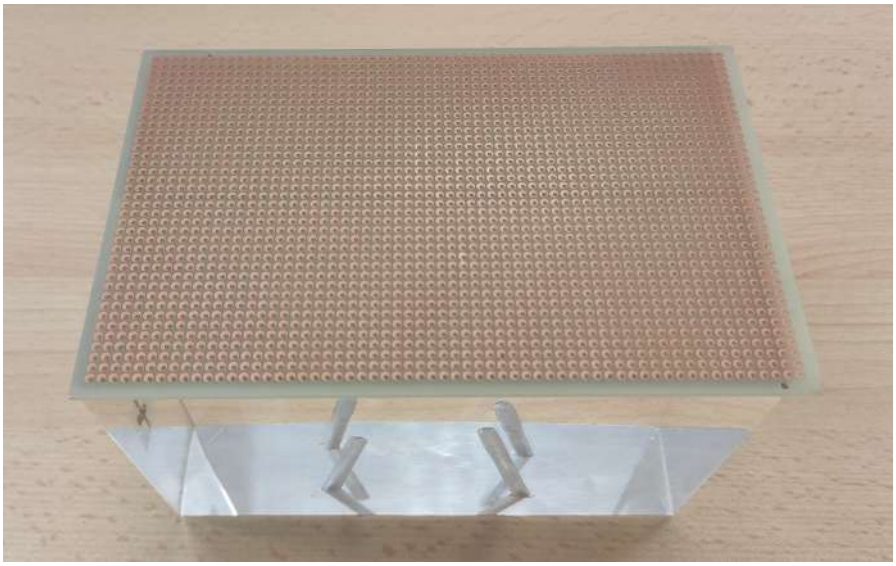


Figure 4.10: Stripboard glued on top of the prism phantom

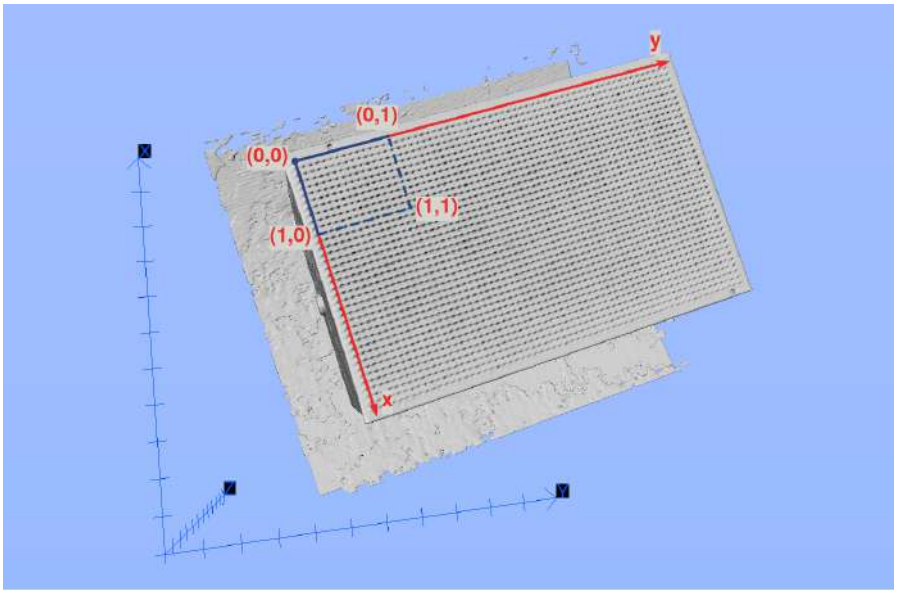


Figure 4.11: TC of the prism with the stripboard glued on top, in red the tabular reference frame used for selecting the target holes for measuring the robot precision, the blue reference frame identifies the robot home position with respect to the prism.

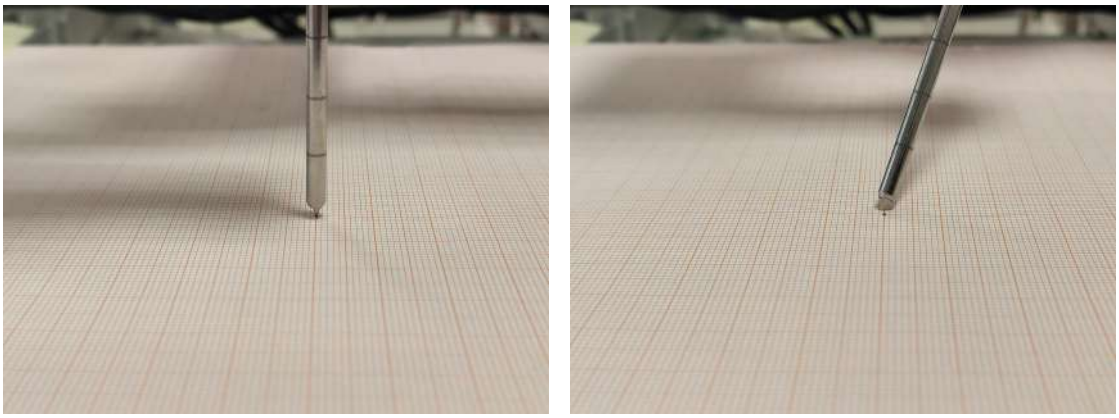
To measure the robot precision in position, test trajectories with target the holes describing the corners of the grid's boxes were carried out. The needle measures $3.5mm$ in diameter, its pointy end measures $0.5mm$ in diameter. The error of a trajectory which centred the hole and did not touch the stripboard was considered null. If contact instead happen, errors in the expected target were measured with a digital calibre. The error along the z robot axis could not be measured reliably with this method. The last column of table 4.2 indicates if the stripboard was touched or not, the trajectories required a maximum needle shift of $-2mm$ to detach from the stipboard.

Grid Index	Target $[x, y, z, \phi, \psi]$	x error	y error	hit
$[3, 0]$	$[30.52, 59.89, 50.92, -0.07°, -0.14°]$	–	–	–
$[2, 0]$	$[57.93, 54.66, 50.63, -0.33°, -1.45838°]$	–	–	–
$[1, 0]$	$[85.41, 49.35, 50.60, -0.22°, -0.87°]$	–	$+0.5$	*hit*
$[0, 0]$	$[112.86, 44.04, 50.73, 0.91°, 4.94°]$	$+0.1$	$+0.8$	*hit*
$[3, 1]$	$[37.40, 94.87, 51.26, -0.03°, 0.11°]$	$+0.5$	–	*hit*
$[2, 1]$	$[64.78, 89.53, 50.76, -0.02°, 0.17°]$	$+0.5$	$+0.5$	*hit*
$[1, 1]$	$[94.66, 83.66, 50.955, 0.18°, 1.18°]$	$+0.5$	$+0.5$	*hit*
$[0, 1]$	$[124.48, 77.78, 51.10, 0.18°, 1.15°]$	$+0.5$	$+0.5$	*hit*
$[3, 2]$	$[44.70, 132.03, 51.78, 0.34°, 1.97°]$	–	$+0.5$	*hit*
$[2, 2]$	$[74.45, 126.39, 51.30, -0.89°, -4.06°]$	$+0.4$	$+0.5$	*hit*
$[1, 2]$	$[104.37, 120.46, 51.19, -0.30°, -2.45°]$	$+0.5$	$+0.5$	*hit*
$[0, 2]$	$[131.92, 115.46, 51.51, 0.21°, 1.34°]$	$+0.2$	$+0.7$	*hit*
$[3, 3]$	$[51.90, 169.78, 51.95, 0.14°, 0.95°]$	–	$+0.5$	*hit*
$[2, 3]$	$[81.77, 163.60, 51.27, -0.06°, -0.014°]$	–	$+0.5$	*hit*
$[1, 3]$	$[111.78, 158.04, 51.23, 0.07°, 0.61°]$	$+0.2$	$+0.7$	*hit*
$[0, 3]$	$[139.11, 152.56, 51.39, 0.20°, 1.27°]$	–	$+0.6$	*hit*
$[3, 4]$	$[59.35, 206.88, 51.49, -0.01°, 0.20°]$	–	$+0.4$	*hit*
$[2, 4]$	$[86.82, 201.48, 51.32, -0.31°, -1.33°]$	$+0.1$	$+0.5$	*hit*
$[1, 4]$	$[116.70, 195.77, 51.00, 0.15°, 1.04°]$	$+0.3$	-0.5	*hit*
$[0, 4]$	$[139.11, 152.56, 51.39, 0.20°, 1.27°]$	$+0.3$	-0.3	*hit*
	Mean Error	0.205	0.595	18/20

Table 4.2: Errors in target position [mm] of trajectories carried out on the prism phantom, registration method based on 3 points.

4.3.3 Pivoting Error

The performances of the pivoting primitive were measured using a sheet of graph paper. The robot needle was positioned on the sheet surface, and displacement were measured after pivoting the robot of the maximum orientations. Since the pivoting depends also on the value of the fifth joint d_5, the procedure was performed for different values of Robot z.



(a) (b)

Figure 4.12: Error of the pivoting primitive measured on a sheet of graph paper.

z value	(ϕ, ψ)	δ_x	δ_y
2	$(20°, 0)$	-0.5	$+0.0$
	$(20°, 0°)$	-1.0	$+0.9$
	$(0°, 30°)$	-0.5	$+0.9$
19	$(20°, 0)$	-1.0	-1.0
	$(20°, 0°)$	-1.4	$+0.0$
	$(0°, 30°)$	-0.4	$+1.0$
49	$(20°, 0)$	-1.0	-1.0
	$(20°, 0°)$	-1.1	$+0.0$
	$(0°, 30°)$	-0.1	$+1.0$

Table 4.3: Errors on the E.E. position after pivoting $[mm]$. Measurements recorded for different values of Robot z.

Chapter 5

Discussion and Conclusions

Exploiting the various phantoms, we were able to understand strengths and weakness of the developed system. The rigid and regular geometry of the Plexiglas phantom with three pillars helped us understand the limitations of the prototype's mechanic. Given the similarity of the phantom surface with the robot support frame, we were able to test the registration method based on the features extraction explained in Section 3.2.1. The method requires that the relation between the robot and the phantom to be rigid and known in advance, posing a strong constraint on the usability of the system in real medical applications. On the other hand, registration based on 3 points explained in Section 3.2.2 showed similar performances while offering fewer constraints. The method can estimate the phantom position with respect to the robot at run time, however it demands that the selection of the required points to be extremely precise. Tests performed on the human-shaped phantom highlighted the disadvantages of positioning the robot directly on the patient skin. The human body is not a rigid volume and is subject to deformations if a load is applied on its surface. Moreover in our application, the weight is not constrained on a single point but it shifts as the robot moves. All these points translate in a change of the robot position upon movement, thus producing errors in the carried out trajectories. To obtain good performances with this setup, the robot attachment position must also be maintained fixed throughout the whole procedure. State of the art solutions exploit fixation directly to the bones to compensate deformations due to the robot weight, ensuring that the robot's position do not change while performing surgery. A further direction of development could be adopting more modern registration setups which corresponds to a less invasive experience for the patient. A possible solution would exploit an external stereo camera to estimate the markers' position. The advantage of this approach is that even if the robot or the patient locations change during surgery, the

camera would be able to detect these changes and update the image-robot registration at run-time, transparently and without the human intervention. Another alternative path could be using the C-Arms to directly register the robot with the fluoroscopic images. This solution would guarantee the best precision since the robot markers, needle and patient's body could all be detected directly by the machine. On the contrary, the time of radiation exposure would increase. Access to the fluoroscopies is not open source, and the integration of these images in the developed application would require a collaboration with the manufacturing company of the C-Arm. The carried out experiments also emphasized the necessity of improving the Robot mechanic. The major constraint is the range of motion of the fifth joint d_5, which limits the movement along the Robot z axis. As shown in Section 4.2, this limitation does not allow to reach the vertebrae, target of the vertebroplasty. A new version of the robot should also address the overall rigidity of the structure. The first areas to improve should be the ones regarding the portions of attachment of the first and second joints d_1, d_2. These structures are responsible for carrying the overall weight of the robot and are the ones subjected to more flex upon motion. The Plexiglas prism was the phantom from which we could extract the more significant data. As shown in Section 4.3, maintaining the E.E. orientation close to $(0°, 0°)$, we were able to estimate an error in position along x and y of $\pm 0.205mm$ and $\pm 0.595mm$ respectively. The error in the robot orientation was estimated to be $2.64°$ and $3.13°$ respectively for ϕ and ψ. The error in the pivoting primitive was estimated and reported in Section 4.3.3. Considering the trajectory described in 3.3.1, composed of pivoting motion and needle shifting, the current overall Robot error in position was estimated to be less than $2mm$ along x and y. The error along z could not be recorded reliably with current setup.

Bibliography

[1] Akira Bekku et al. "A body-mounted surgical assistance robot for minimally invasive spinal puncture surgery". In: *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics* (Sept. 2014), pp. 19–23.

[2] P.J. Besl and Neil D. McKay. "A method for registration of 3-D shapes". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14.2 (1992), pp. 239–256.

[3] Yang Chen and Gérard Medioni. "Object modelling by registration of multiple range images". In: *Image and Vision Computing* 10.3 (1992). Range Image Understanding, pp. 145–155.

[4] Marissa D'Souza et al. "Robotic-Assisted Spine Surgery: History, Efficacy, Cost, And Future Trends [Corrigendum]". In: *Robotic Surgery: Research and Reviews* Volume 6 (Dec. 2019), pp. 25–26.

[5] G. Eggers, J. Mühling, and R. Marmulla. "Image-to-patient registration techniques in head surgery". In: *International Journal of Oral and Maxillofacial Surgery* 35.12 (2006), pp. 1081–1095.

[6] Vincent Groenhuis et al. "Stormram 3: A Magnetic Resonance Imaging-Compatible Robotic System for Breast Biopsy". In: *IEEE Robotics & Automation Magazine* 24.2 (2017), pp. 34–41.

[7] Yue Hu et al. "A Comparation Between Frame-Based and Robot-Assisted in Stereotactic Biopsy". In: *Frontiers in Neurology* 13 (July 2022).

[8] Meng Huang et al. "The current state of navigation in robotic spine surgery". In: *Annals of Translational Medicine* 9 (Jan. 2021), pp. 86–86.

[9] Hao Ju et al. "A Robot-Assisted System for Minimally Invasive Spine Surgery of Percutaneous Vertebroplasty Based on CT Images". In: *2008 IEEE Conference on Robotics, Automation and Mechatronics*. 2008, pp. 290–295.

[10] John Krumm. *Intersection of Two Planes*. https://www.microsoft.com/en-us/research/publication/intersection-of-two-planes/. May 2000.

[11] Georgi Minchev et al. "A novel miniature robotic guidance device for stereotactic neurosurgical interventions: Preliminary experience with the iSYS1 robot". In: *Journal of neurosurgery* 126 (Apr. 2016), pp. 1–12.

[12] N. Neumann et al. "Robot-Assisted Bone Cement Injection". In: *IEEE Transactions on Biomedical Engineering* 69.1 (2022), pp. 138–147.

[13] Matteo Meneghetti Noè Murr. *An implementation of a CANopen device profile for drives and motion control according to CiA 402®*. https://gitlab.com/altairLab/elasticteam/can/motor402.

[14] Shinya Onogi et al. "Development of the Needle Insertion Robot for Percutaneous Vertebroplasty." In: Jan. 2005, pp. 105–113.

[15] Rogers-LaVanne MP Perfetti DC Kisinde S. "Robotic Spine Surgery: Past, Present, and Future surgery". In: *Spine* 47 (July 2022), pp. 909–921.

[16] Françoise Siepel et al. "Needle and Biopsy Robots: a Review". In: *Current Robotics Reports* 2 (Mar. 2021), pp. 1–12.

[17] Martin Vychopen et al. "Patient Safety Comparison of Frameless and Frame-Based Stereotactic Navigation for Brain Biopsy—A Single Center Cohort Study". In: *Brain Sciences* 12.9 (Sept. 2022), p. 1178.

[18] Biao Wang et al. "Pediculoplasty combined with vertebroplasty for the treatment of Kummell's disease without neurological impairment: robot-assisted and fluoroscopy-guided". In: *American journal of translational research* 12 (Dec. 2020), pp. 8019–8029.

[19] M. Welleweerd et al. "Design of an end-effector for robot-assisted ultrasound-guided breast biopsies". In: *International Journal of Computer Assisted Radiology and Surgery* 15 (Feb. 2020).