

**ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA**

---

**SCUOLA DI INGEGNERIA E ARCHITETTURA**

*DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA ELETTRICA E  
DELL'INFORMAZIONE - DEI*

*CORSO DI LAUREA MAGISTRALE IN INGEGNERIA ELETTRONICA*

**TESI DI LAUREA**

in

ELABORAZIONE DI DATI E SEGNALI BIOMEDICI M

**ROBOTIC TRANSCRANIAL MAGNETIC  
STIMULATION ASSISTANT**

CANDIDATO:  
Riccardo Seganfredo

RELATORE:  
Prof. Lorenzo Chiari

CORRELATORI:  
Ph.D. Melissa Zavaglia  
Dipl. -Ing. Ioannis Xygonakis

Anno accademico 2020/2021

Sessione III



*Heartfelt thanks to everyone who helped me with this thesis.  
Thanks to my supervisors Melissa and Ioannis,  
but also to M.Sc. Samuel Schneider and M.Sc. Viktorija Dimova-Edeleva.  
Thanks to my colleagues and friends Simone, Matilde, Claudia and Federico,  
for their friendship and the moments passed together during these years.  
Thanks to my family, that guided and supported me to achieve this result.*

*To my mum,  
because she was stolen the time to read these lines.*

*Un sentito ringraziamento a coloro i quali mi hanno aiutato con questa tesi.  
Grazie ai miei supervisori Melissa e Ioannis,  
ma anche a M.Sc. Samuel Schneider e M.Sc. Viktorija Dimova-Edeleva.  
Grazie ai miei colleghi ed amici Simone, Matilde, Claudia e Federico,  
per la loro amicizia ed i momenti trascorsi insieme in questi anni.  
Grazie alla mia famiglia, che mi ha guidato e supportato  
fino al raggiungimento di questo traguardo.*

*Per la mia mamma,  
alla quale è stato sottratto il tempo per leggere queste righe.*



# Abstract

The Transcranial Magnetic Stimulation (TMS) is a non-invasive technique to stimulate the brain, with main applications in depression treatment and pre-operative planning (via functional motor mapping and speech mapping). On average, a TMS treatment session lasts for 30 minutes and coil handling/positioning might become a strenuous task for the operator. A robotic arm could be used to replace the human operator during the coil positioning tasks allowing the doctor to focus on the data analysis phase.

In this thesis, a navigated TMS Robotic Assistant is designed, implemented and integrated with a commercial TMS system to automate the TMS sessions. Two different navigation approaches are investigated: i) with fixed head position, ii) with head movement compensation. To assess the performances of the implemented Robotic Assistant, several experimental sessions are carried out; the results satisfy the expectations, with an accuracy error of 3.5 mm for stimulation targets, which decreases below 2 mm with repeated stimulus. Most of the functional requirements are fulfilled, however further investigations are needed to improve the proposed methods and implement new functionalities to obtain an enhanced version of nTMS Robotic Assistant.



# Sommario

La Stimolazione Magnetica Transcranica (*Transcranial Magnetic Stimulation*, o TMS) è una tecnica non invasiva per la stimolazione del tessuto cerebrale, che trova le principali applicazioni nel trattamento di disturbi psichiatrici e neurologici, come la depressione, e nella pianificazione preoperatoria, attraverso la mappatura delle funzionalità motorie e linguistiche (*functional motor mapping* e *speech mapping*). In media, un trattamento tramite TMS dura 30 minuti, rendendo il maneggio e il posizionamento della bobina un compito faticoso per l'operatore. Un braccio robotico può essere utilizzato per sostituire l'operatore umano durante l'attività di posizionamento della bobina, permettendo al medico di concentrarsi sulla fase di elaborazione dei dati raccolti.

In questa tesi si propone il progetto di un Assistente Robotico per nTMS (*navigated TMS*), implementato e realizzato con un sistema TMS commerciale per automatizzare la sessione del trattamento. Sono studiati due approcci di navigazione differenti: i) con posizione fissa della testa, ii) con compensazione dei movimenti della testa. Per valutare le prestazioni dell'Assistente Robotico, sono effettuati diversi esperimenti; i risultati soddisfano le aspettative, con un errore di accuratezza massimo di 3.5 mm per normali target di stimolazione, che scende sotto i 2 mm per target di stimoli ripetuti. La maggior parte dei requisiti di funzionamento sono soddisfatti, tuttavia sono necessari ulteriori studi per raffinare i metodi proposti e implementare nuove funzionalità che permettano di ottenere una versione migliorata dell'Assistente Robotico per nTMS.



# Contents

<b>1</b>	<b>Introduction</b>	<b>11</b>
<b>2</b>	<b>The nervous system</b>	<b>13</b>
<b>3</b>	<b>Transcranial Magnetic Stimulation</b>	<b>17</b>
3.1	The motor mapping . . . . .	18
3.2	The Nexstim NBS System 4 . . . . .	19
3.2.1	Contraindications and warnings . . . . .	19
3.2.2	The Nexstim cart . . . . .	19
3.2.3	TMS workflow with Nexstim NBS Software . . . . .	21
3.2.4	The NBS 4.5 Robotic Interface - Prototype . . . . .	24
<b>4</b>	<b>Franka Emika Panda</b>	<b>27</b>
4.1	Programming interfaces . . . . .	27
4.2	MIOS (Machine Intelligence Operating System) . . . . .	28
<b>5</b>	<b>The TMS Robotic Assistant: State of the Art and design phase</b>	<b>31</b>
5.1	Robotic TMS: State of the Art . . . . .	32
5.2	Design phase . . . . .	32
5.2.1	System Functional Requirements . . . . .	32
5.2.1.1	Feature 1: General . . . . .	33
5.2.1.2	Feature 2: Reach and stimulation of cortical target points . . . . .	33
5.2.1.3	Feature 3: Collision detection . . . . .	34
5.2.1.4	Feature 4: Subject safety mechanism . . . . .	34
5.2.2	System Architecture . . . . .	35
<b>6</b>	<b>The TMS Robotic Assistant: implementation phase</b>	<b>37</b>
6.1	The laboratory set-up . . . . .	37
6.2	The custom coil connector . . . . .	39
6.3	The end-effector parameters set-up . . . . .	41
6.4	The Robotic Assistant client interface . . . . .	43
6.5	The calibration task . . . . .	44
6.6	The Assistant algorithms . . . . .	47
6.6.1	Navigation algorithms . . . . .	47
6.6.1.1	Virtual sphere navigation . . . . .	48
6.6.1.2	Reaching tasks . . . . .	50
6.6.1.3	Area stimulation task . . . . .	54
6.6.2	Panda task functions . . . . .	54
6.6.2.1	<i>MoveToPoseJoint</i> skill . . . . .	54

6.6.2.2	<i>MoveToPoseCart</i> skill . . . . .	55
6.6.3	Safety . . . . .	55
6.6.3.1	Contact detection method . . . . .	55
6.6.3.2	NBS Status watchdog . . . . .	56
6.7	Subsystems integration . . . . .	56
6.7.1	Pre-calibration tasks . . . . .	58
6.7.2	Calibration tasks . . . . .	58
6.7.3	Sphere navigation and positioning task . . . . .	58
6.7.4	Stimulation tasks . . . . .	58
6.7.5	Post-stimulation tasks . . . . .	59
6.8	Navigation with movements compensation . . . . .	59
<b>7</b>	<b>Experiments</b>	<b>61</b>
7.1	Experiments description . . . . .	61
7.1.1	Experiments with movement compensation . . . . .	64
7.2	Experimental data . . . . .	64
7.2.1	Results with movement compensation . . . . .	65
7.3	Interpretation of the results . . . . .	74
7.3.1	Session 1: "Standard" stimulation targets . . . . .	74
7.3.2	Session 2: repeated stimulus targets . . . . .	74
7.3.3	Movement compensation case . . . . .	75
<b>8</b>	<b>Discussion</b>	<b>77</b>
8.1	Experimental data . . . . .	77
8.1.1	Fixed-head approach: first session . . . . .	77
8.1.2	Fixed-head approach: second session . . . . .	78
8.1.3	Movement compensation approach . . . . .	78
8.2	Hardware setup . . . . .	78
8.3	Collision detection . . . . .	79
8.4	Contact detection . . . . .	79
8.5	Functional requirements . . . . .	79
8.6	Future implementations . . . . .	80
<b>9</b>	<b>Conclusions</b>	<b>81</b>

## References

### A List of Abbreviations

# Chapter 1

## Introduction

The Transcranial Magnetic Stimulation (TMS) is a neurophysiologic technique that allows the non-invasive brain stimulation by means of a suitable magnetic field pulse [1]-[2]. The generated magnetic field passes across the skull and induces an electric current that leads to a local stimulation of a certain cortical area. Nowadays TMS, associated with other methods, is used to study intracortical, cortico-cortical and cortico-subcortical interactions, as well as relations between brain activity and behaviour, and various neurological and psychiatric disorders [2]. Among the typical applications of TMS, it can be found the functional motor mapping and the speech mapping. Thanks to the technological progress, a navigation system is now integrated to the TMS technique, which become the so-called nTMS (navigated TMS): the location where the maximum electric field is focused, is determined online and it is usually displayed on anatomical brain images (coming from 3D Magnetic Resonance Imaging, also called 3D MRI) [3]. A camera is used for the real time monitoring of the head and coil trackers positions. However, a typical pre-operative nTMS session, for example, takes at least 30 minutes and the task is quite tedious and repetitive for the operator, which may lead to mistakes. Data produced from this kind of session are used by the surgeon for pre-operative planning, and it is important to be as consistent and accurate as possible.

The project presented in this thesis is intended to be a research application for a navigated TMS system with a robotic assistant. The NBS system used is produced by Nexstim, based in Finland, and it has been equipped specifically for this research project. The adopted robotic arm is Panda by the German company Franka Emika. The project timeline starts from the set-up of the interfaces and architecture required for their communication, moving to the laboratory arrangement and the creation of new custom solutions to reach the compatibility between the robot and the stimulation system. In addition, part of the project is also the evaluation of the Software Robotic Interface developed by Nexstim exclusively for this task. The final system should be able to place the coil at predefined positions and, successively, derive autonomously a functional motor map for the patient. Moreover, delivering various stimulating pulses and monitoring the induced movements via EMG, the autonomous nTMS should be independently able to learn from the relationship brain-muscle extracted from previous mapping steps.

This thesis starts with a brief introduction of the human nervous system, describing what are the main parts and how they are correlated. This constitutes a basis for the description of the transcranial magnetic stimulation technique, explaining why it is used and presenting the hardware and software equipment used for this project. After that, it is given an overview of the adopted robotic arm, introducing the software interfaces for the

taks programming. These first chapters represent a solid base for the construction of the TMS Robotic Assistant.

After a brief overview of the State of the Art, the description of the project development is divided into main parts: i) the design phase, where Assistant features and characteristics are pointed out (Chapter 5), and ii) the implementation phase, where the designed hardware and software solutions are presented (Chapter 6). Next, the experimental set-up used to assess the performances of the implemented Robotic Assistant is described, and the recorded data are analyzed, plotted and interpreted (Chapter 7). Finally, the main issues encountered during the Robotic Assistant development are described, followed by a presentation of possible future improvements for the next implementations of the Robotic Assistant (Chapter 8).

## Chapter 2

# The nervous system

The nervous system is the human body communication network that allows the rapid information transmission between the linked parts [4]. It can be divided into two main subsystems: the central nervous system (CNS) and the peripheral nervous system (PNS). The former consists in the brain and the spinal cord, which are entirely located within the skull and the vertebral canal, and accomplishes the functions related to the thought, language, movement, analysis of sensations and emotions; the latter represents the nerves connections between the CNS and all the rest of the body, extending like wires throughout the tissues. A diagram of the human nervous system is shown in Figure 2.1.

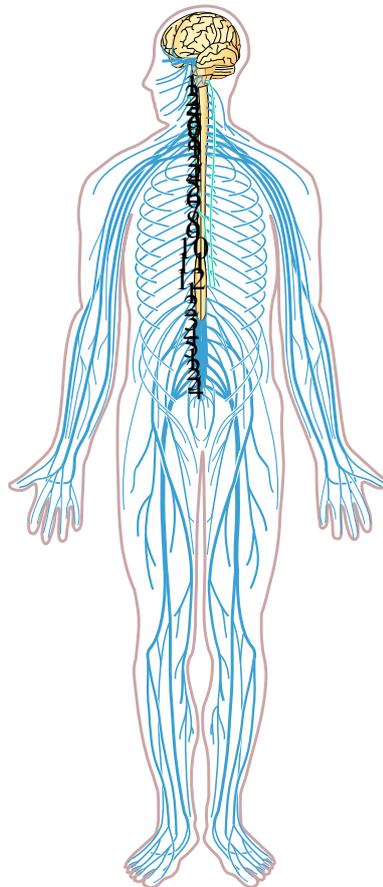


Figure 2.1: A diagram of the human nervous system: the CNS (yellow) and the PNS (blue). From [5].

A nerve cell is also called a neuron, and it is constituted by three main parts: the cell body, the axon and the dendrites (an illustrative diagram is shown in Figure 2.2). The first one contains the nucleus and it is surrounded by the dendrites, extensions of the cell membrane that receive the information and forward it to the cell body. The axon is also a singular long extension of the cell membrane, and its function is to transmit the information from the cell body to the extremities of the cell. The messages between adjacent neurons are transmitted chemically and in a unidirectional way, through specialized structures called synapses. In fact, a narrow gap separates an axon extremity of the transmitting neuron from a dendrite of the receiving one: when the electrical signal reaches the first one, a certain amount of chemical (neurotransmitter) is released by many vesicles through the presynaptic membrane into the gap. It reaches the postsynaptic membrane by concentration gradient and, depending on the nature of the neurotransmitter, it can induce the generation of an action potential in the receiving neuron or a temporary inhibition to following stimulations. The function of a neuron determines the kind of neurotransmitter that is released. The synapses are not simply connecting two different cells, but they enable the interaction of information from several sources on a single neuron, that may similarly influence other cells, forming a complex structure to accomplish sophisticated functions.

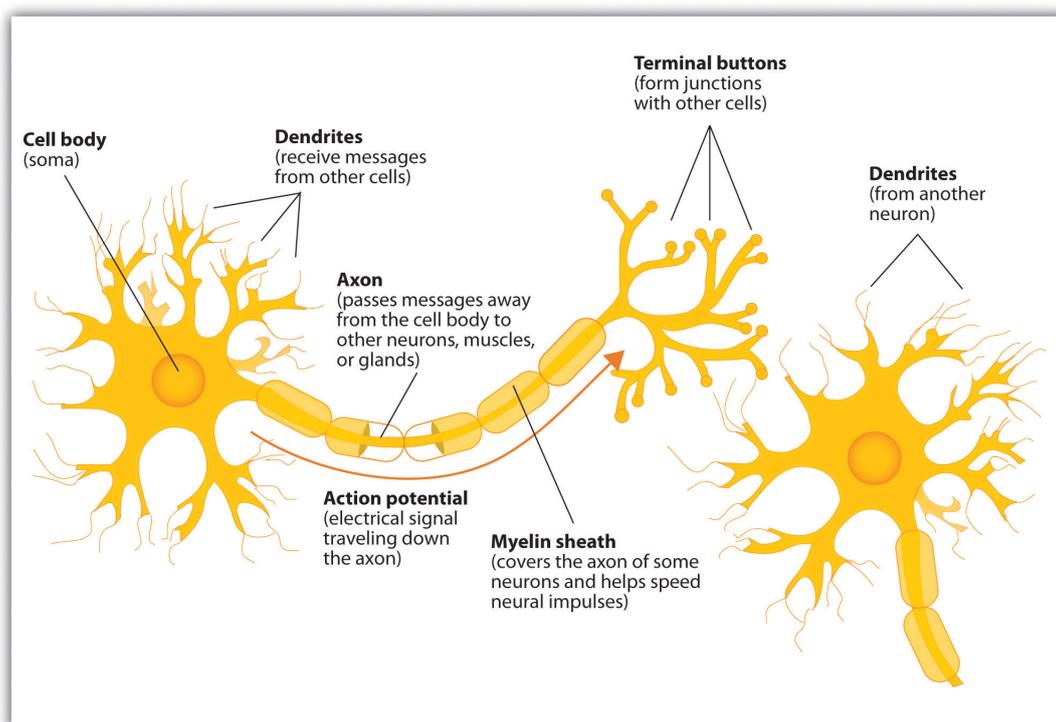


Figure 2.2: Diagram of a neuron and its components. From [6].

Considering the structure of an adult human brain, which is part of the CNS with the spinal cord, three main parts can be identified: the diencephalon, the cerebellum, and the telencephalon (or cerebrum). A simplified diagram is reported in Figure 2.3.

The first lies deep inside and its largest component is the thalamus, whose functions include several motor, sensory and emotional processes. The second is positioned in the posterior cranial fossa, and it carries out different functions, such as the control of posture, repetitive movement and the geometric accuracy of voluntary movements. Finally,

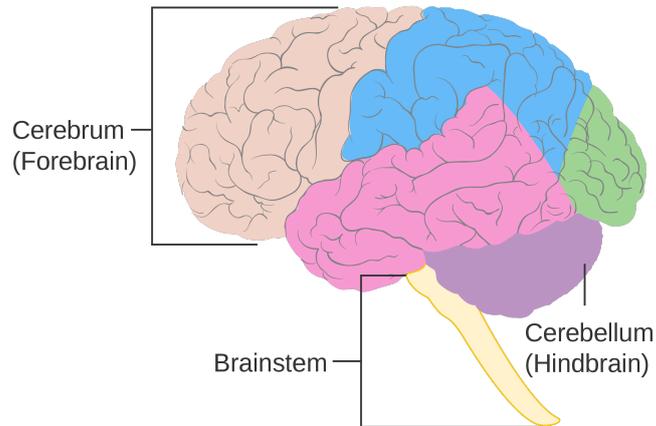


Figure 2.3: Diagram of the main parts of the brain. From [7].

## Motor and Sensory Regions of the Cerebral Cortex

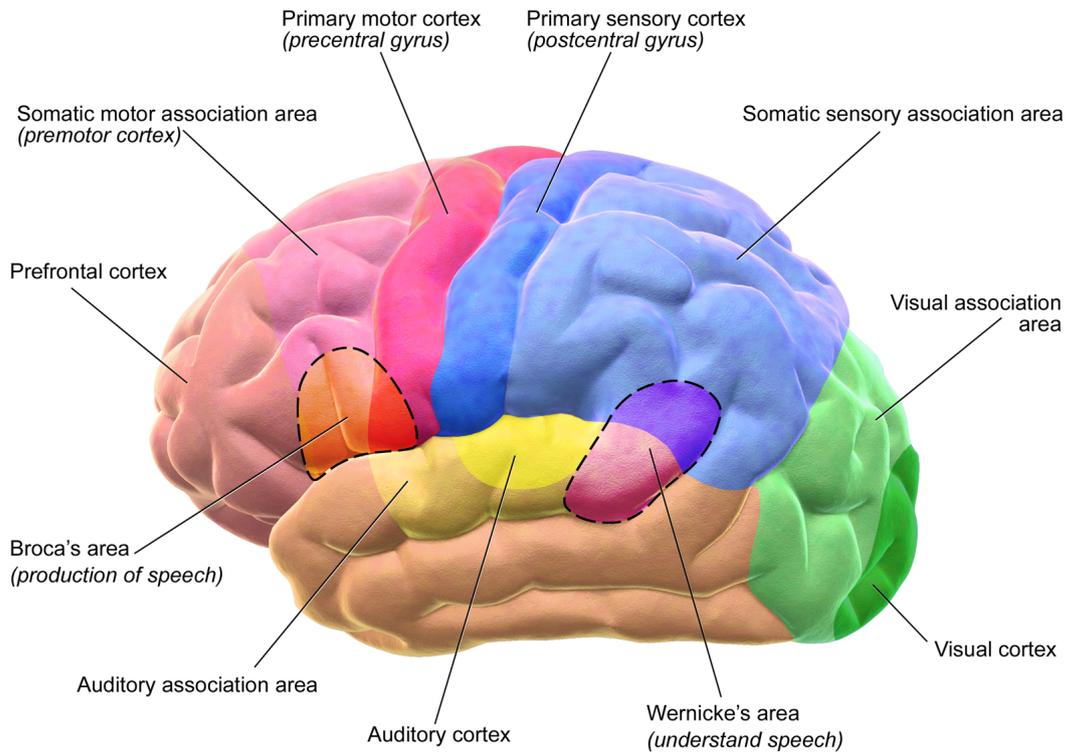


Figure 2.4: Diagram of motor and sensory regions of the cerebral cortex. From [8].

the cerebrum occupies the remainder of the cranial cavity, divided into two halves called cerebral hemispheres. The surface of each of them is composed by layers of grey matter, that takes the name of cerebral cortex, and it is characterized by ridges, called gyri, and depressions, or sulci. Some of them are significant, because they divide the cortex into regions with a specific associated function (as shown in Figure 2.4). For example, the precentral gyrus is the place responsible for the contraction of voluntary muscles, and for this reason it is also referred to as the motor cortex. This brain region and its relationships with muscular movements are extensively investigated during the functional motor mapping technique.

# Chapter 3

## Transcranial Magnetic Stimulation

The Transcranial Magnetic Stimulation, usually abbreviated with TMS, is a technique used for non-invasive brain stimulation [1], consisting in a strong current pulse, driven through a coil placed over the head of the patient. The magnetic field generated penetrates the human skull and, by induction, it leads to a charge movement and a consequent local stimulation; nevertheless, no direct biological damages are caused [3]. In fact, the effect is entirely due to the induced electric current that accumulates a sufficient amount of charge to depolarize or hyperpolarize the cell membranes; this causes synchronous action potentials that propagate through neurons. Nowadays, TMS is an important tool in clinical routine and, moreover, it represents an affordable and reliable technique to study the brain's functionality and connectivity. The magnetic field generated by the coil arises from a high-current pulse (4-20 kA) and lasts for a few milliseconds: the induced current density distribution varies within the cortex, both in direction and magnitude. The latter are influenced by several factors, such as the coil geometry and position above the scalp and the electrical conductivity of the brain tissues. This last variable makes our ability to predict the real current distribution inside the tissue much more complicated, a task that can be carried out only by means of complex models and simulations. Nevertheless, various kinds of neurons react differently to the orientation and location of the induced field, informing us about cell structures and position along the cortex. In fact, deeper brain areas stimulation is not physically possible since the electromagnetic field penetration is limited.

Modern TMS systems can produce two main kinds of stimulations: single or repetitive pulses. The former is indicated for immediate reactions, such as a muscle twitch, while the latter can influence the neuronal behavior. Typically, motor cortex mapping requires single pulse TMS: several EMG channels are available, in order to record the muscle response and map the brain area associated with it. Moreover, single pulse TMS is also applied to study the brain's connectivity and functionality [1]. Generated pulses could be biphasic, monophasic (also said bipolar or monopolar) or polyphasic: the former consist in a single voltage oscillation cycle, generated by taking advantage of the resonance effect between the coil and a parallel capacitor; if the supplied cycles are more than one, then the pulses are said to be polyphasic. If a shunting diode is added in parallel to the coil-capacitor loop described before, the voltage oscillation becomes unidirectional and can be singularly delivered [2]. Two types of coils are frequently used with the mentioned TMS techniques: circular or round coil and figure-of-eight or butterfly coil [2]. The former, with the simplest design, generates a spherical, but not much focal magnetic field, useful for single pulses and peripheral stimulation. The latter is composed of two parallel coils,

placed side by side, in order to get a stronger, focal magnetic field under the contact point between the two. The figure-of-eight coil is preferred in clinical and academic applications of TMS. A third, more complex, coil shape is available: the H-coil, created to stimulate the deeper cortical layers up to 6 cm below the surface.

Navigated TMS technique takes advantage of the individual MRI of the patient, which becomes the navigation map, to make the targeting possible and visible [3]. Several parameters must be taken into account to precisely deliver the stimulation pulses, such as the location, orientation, tilt, size and shape of the coil, as well as the amplitude and waveform of the pulse. Moreover, physical parameters of the subject head are required: MRI scans supply the structural information of the whole head, while its surface, ears and nose are all accessible as landmarks. This information is essential to locate the coil target and to calculate the stimulating intracortical electric fields. To make all this possible, several navigation tools are used, like the coil and head trackers, that allow the system to monitor the precise position and movements of the two parts; a digitizer pen, useful to point out the landmarks over the scalp, and a 3D position sensor that real time locates all the previous reference points, linking them together. A monitor connected to the TMS system shows all the relevant information: the virtual head reconstruction, obtained from the MRI scans, a representation of the coil that perfectly follows the real coil and head position changes, the estimated magnetic field amplitude and direction.

### 3.1 The motor mapping

In 2009 the FDA approved the navigated TMS as a validated device for presurgical functional mapping of the motor cortex [3]. The TMS is counterposed to other methods, like the functional magnetic resonance imaging (fMRI) or the magnetoencephalography (MEG), that record the cortical areas activity during the execution of a certain task. In the TMS procedure, a map between cortical areas and muscles is generated, by means of the correlation between the coil position and orientation, and the motor evoked potential registered by EMG sensors. This method is also useful to check the integrity of the corticospinal tract (CST) or the presence of a brain tumor and its relation with the cortical anatomy. In this particular case, depending on the position of the tumor, a list of muscles in upper and lower extremities, as well as facial muscles, is usually used as a standard setup for an analysis session. Usually, during a normal TMS session, several points on the scalp are pointed out in correspondence of the cortical motor regions: the motor evoked potentials are recorded and the stimulated targets are often visualized using a color code related to the response amplitude. Moreover, during the off-line analysis, also the latency of each EMG peak is considered for the evaluation.

It is important to remark that the navigated TMS technique is a non-invasive procedure, and it proved its efficiency in many studies where its results have been compared to invasive methods outcomes: the mean distance between the hotspots using the two techniques (invasive and non-invasive) was less than 15 mm. Compared to the fMRI, the nTMS has better spatial and temporal resolution and provides an immediate result [3]; moreover, recent studies (Krieg et al. 2012; Picht et al. 2012, 2013, 2016; Rizzo et al. 2014; [3]) have demonstrated that a nTMS preoperative session has influenced the initial surgical strategy in 25-70% of cases, leading to a quicker surgical operation and better long-term outcomes.

## 3.2 The Nexstim NBS System 4

The Nexstim Navigated Brain Stimulation System 4, hereafter indicated as NBS System, is developed for the primary motor cortex non-invasive mapping, and it is useful to provide information regarding the assessment of the primary motor cortex for pre-operative planning [9]. Moreover, it is also given a description of the hardware set-up of the Nexstim TMS System used for this project.

### 3.2.1 Contraindications and warnings

Since the System is designed for the non-invasive mapping oriented to a pre-procedural planning, it should not be used during a surgical procedure and employed only by trained clinical professionals [9]. It is advised against using the NBS System on people with: any metallic implants in the head (excluding teeth); any kind electrical implants; increased intracranial pressure, with intracardiac lines, intravenous pumps, or dose calculators. A clear benefit or a proved need for clinical reasons are required for a NBS System session on patients with: epilepsy; serious heart disease; lowered seizure threshold due to acute large infarctions, intracranial hemorrhage, or trauma; medication that lowers the seizure threshold; cardiac pacemaker. It is advised against using the NBS System on pregnant women and the System should not be employed by: pregnant operators; operators with any electrical implants or metallic implants in the upper body.

When using the NBS System for a treatment, an operator should always visually monitor the patient to prevent or detect any signs of seizure or muscle twitching and, if necessary, he should immediately terminate the stimulation process and tend the patient.

For safety reasons, the NBS System should not be employed in an Intensive Care Unit, in proximity of sensitive devices (like life maintaining or high frequency surgical equipment) and in presence of flammable gases or liquids. When delivering pulses, there is always the risk to generate induced electrical currents on conductive objects or electrical devices near the System: for this reason, the coil should be used with an appropriate distance from those elements. Moreover, to avoid the risk of getting an electric shock, there should not be evidence of external damage or damp or wet parts. Finally, even if the stimulation intensity may not be sufficient to induce the expected muscle response, which is an individual characteristic, one must consider that this does not automatically mean that the stimulated cortical area is damaged or not functioning properly.

### 3.2.2 The Nexstim cart

In Figure 3.1 is shown the NBS System equipment cart [9], while the numbers listed are used to describe the picture. One of the most important elements of the System is the tracking unit (1), the camera Polaris Vicra by NDI: it allows the real-time tracking of 3D tools positions (up to 15) with a volumetric accuracy of 0.25 mm and 95% confidence interval of 0.5 mm [10]. Data are acquired through a USB interface, with a maximum frame rate of 20 Hz. The camera is firmly attached to a supporting arm fixed to the cart structure. The NBS System software is run on a computer connected to two monitors (2); the GUI of the software allows the operator, among other things, to import the MRI images associated with the patient, to select the characteristics for the stimulation sequence and to activate/disable targets on the cortex. In the main window, a 3D reconstruction of the patient's head and brain is shown thanks to the MRI images analysis; moreover,

the operator has the possibility to enable the visualization of the other tools, like the coil or the digitizing pen (3), with respect to the head position. To measure and record the muscles' electrophysiological responses after each delivered magnetic pulse, an EMG amplifier with 6 channels is equipped to the System (4), with the power source integrated inside the cart. Pre-gelled disposable electrodes are recommended for the measurements with this System. Other important elements are the cooling unit (5) and the cooled coil (6): the former is used to deliver sufficient air flow to the coil, with two flexible pipes for in (red-labeled, hot) and out (blue-labeled, cold) flows. This method allows treatment sessions with long, high frequency pulse trains without overheating. The energy necessary for each pulse is stored in the TMS Stimulator (7), that must be switched on during the entire duration of the stimulation sequence. Finally, three pedals (8) are available and allow the operator to easily interact with the Software, to deliver pulses, to increase or decrease the stimulation intensity and to register the head landmarks.



Figure 3.1: The Nexstim NBS cart: 1) Tracking unit; 2) monitors; 3) digitizing pen; 4) EMG amplifier; 5) cooling unit; 6) cooled coil; 7) TMS Stimulator; 8) pedals.



Figure 3.2: NBS Software; Registration tool window.

### 3.2.3 TMS workflow with Nexstim NBS Software

To start a new stimulation session with a patient, the NBS System should be first started up. The computer equipped on the cart should be switched on and, after that, the NBS Software must be started. During this phase, the Software sets up the tracking unit, that is ready in a few minutes. The cooling unit should be switched on separately.

#### Data import

Once the Software is open, the MRI images of the patient must be imported, that have to include ears and nose that will be used as landmarks. It is recommended to use MRI scans with spatial resolution of the images (voxel size) of 1x1x1 mm. Opening a New Session in the Software allows the operator to select the MRI data and to fill in the details regarding the patient.

#### Registration process

When the data are uploaded, the Registration process can be started (the NBS Software tool is shown in Figure 3.2): this leads to the selection of three anatomical landmarks to link the coordinate system of the virtual head reconstruction to the real-world data acquired by the tracking camera. The landmarks are pointed out on the MRI images through the Software GUI, as shown in Figure 3.3: the crus of helix for each ear and the nasion. After the virtual selection of these points, they must be precisely pinpointed on the patient's head with the assistance of a digitizing pen. When this procedure is completed, the Software asks to register nine additional points on the scalp to increase the precision of the association: if the Registration process is successfully accomplished, the mismatch error between the virtual reconstruction and the real world is below 2mm.

From this moment, the tracking system can precisely localize the position of the head, of the coil and of the digitizing pen thanks to the markers previously applied; moreover, the operator can open new Exam sessions in order to deliver the magnetic pulses. On the NBS Software, an Exam appears like a folder where the data related to the stimulations are stored, such as the intensities, the locations, and the coil orientations for each pulse. To start a new stimulation sequence, an Exam session must be opened. Then, depending on the treatment plan, the operator can choose to pinpoint new targets on the patient's scalp, or to repeat a stimulation profile from a previous session.

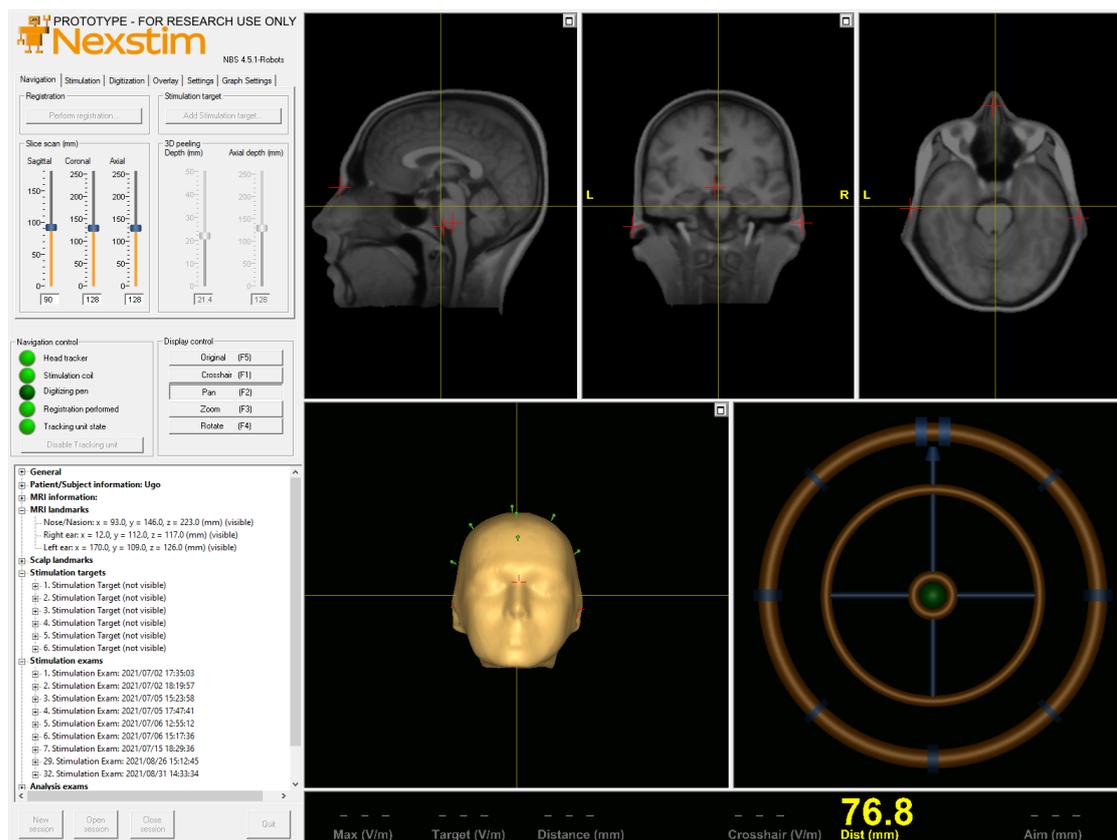


Figure 3.3: NBS Software; main window with landmarks highlighted (red crosses).

### “Standard” stimulation targets and repeated stimulus targets

Two main kinds of targets are available on the Software: the normal target and the repeated stimulus target. The former could be a new point or could be selected on the targets history; once it is set as an active target and it is highlighted on the head’s virtual reconstruction, it localizes a spot through a translation vector with respect to the MRI coordinate system, as shown in Figure 3.4. On the other hand, the latter could be selected only from the history of the previous stimulations: this happens because it returns a complete transformation matrix, made by the concatenation of a rotation matrix and a translation vector. This information enables the operator to position the coil exactly in the same location and with the same orientation of the selected stimulus, leading to the name of repeated stimulus. When this kind of target is set active, the spot is highlighted on the head’s virtual reconstruction and a graphic aiming tool is enabled, helping the operator in the positioning task (Figure 3.5).

Once a target is set active and the coil is correctly positioned, a new Sequence of stimulations can be opened. Here the operator can select the intensity for a single pulse (as shown in Figure 3.6 left) or a train of pulses and, once all is ready, he can start the stimulation. In addition, he can also enable the EMG channels to monitor the muscle response after each pulse, thanks to the disposable electrodes previously attached on the skin area of interest. The EMG channels selection window is shown in Figure 3.6 right.

The data regarding all the pulses, the EMG and the selections made so far are recorded and stored automatically by the NBS Software.

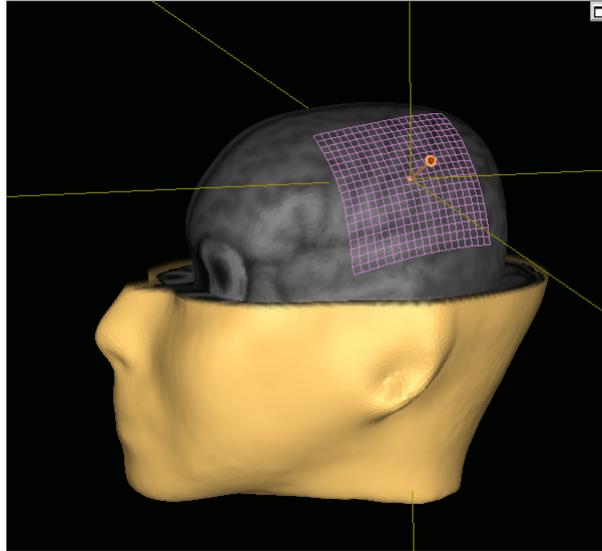


Figure 3.4: NBS Software; "Standard" stimulation target set active.

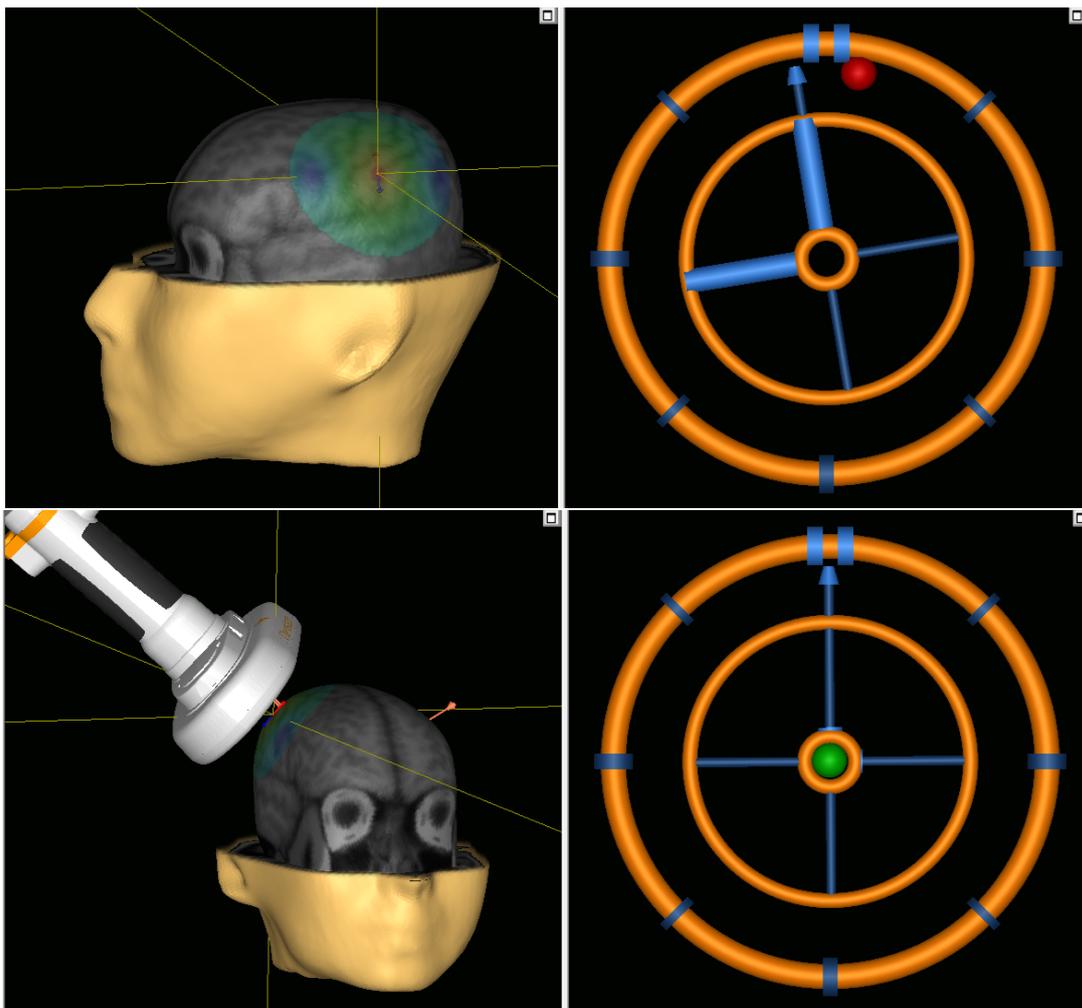


Figure 3.5: NBS Software; Repeated stimulus target set active. The coil is not in position.

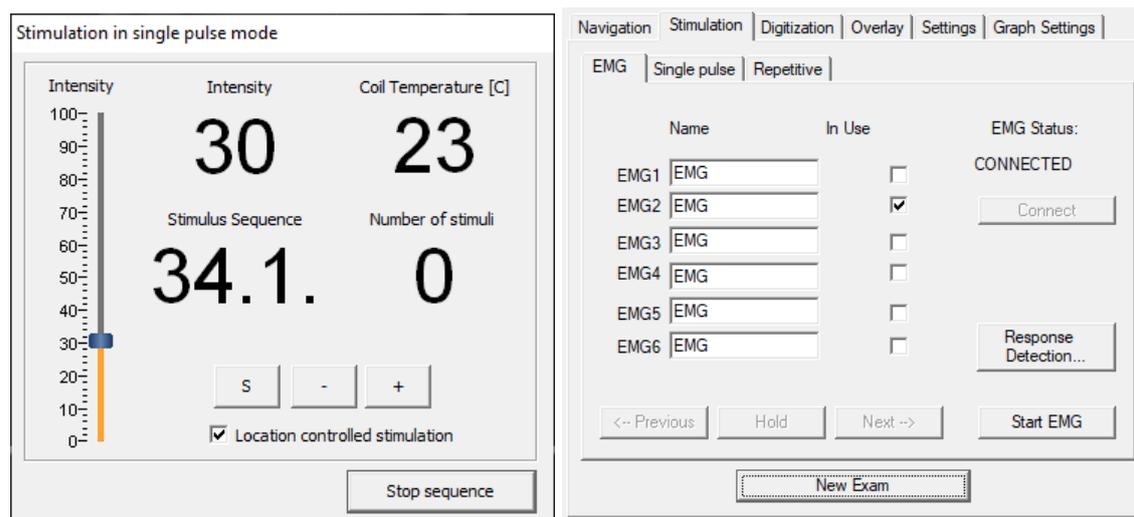


Figure 3.6: NBS Software; Single pulse mode stimulation tool (left). EMG channels selection window (right).

### 3.2.4 The NBS 4.5 Robotic Interface - Prototype

The Nextstim NBS system that was used in this thesis project, was provided by TUM research institute MSRM (from 01.10.2021 to be renamed: MIRMI - Munich Institute of Robotics and Machine Intelligence). This system was uniquely equipped with an interface that works on top of TCP/IP protocol [11]. This interface is intended to be used by the robotic arm controller to position the stimulation coil at the target coordinates specified by the NBS Software. A Client/Server model is used, where the Robotic Assistant acts as the client, while the NBS Software acts as server, listening at port 1001 and accepting only one client per time. To get the correct information from the server, the client can send a request specifying a Function ID, from 1 to 7, where each of them is associated with a specific data. Every request and response message start with a so-called Header, a field made by the Tag, that is the request sequence number, and the Length in bytes of the following data. Moreover, there is a field dedicated to the Function ID and, in case of a response, an additional space for the Error code. The final field of a response packet is assigned to the Data, associated with the Function ID. The Data field always starts with the NBS System Status, a short integer where each bit is a flag for a specific event.

Function ID	Response Data
1	NBS System status
2	Location of MRI landmarks (status + 3 matrices)
3	Coil location (status + 1 matrix)
4	Coil location and repeated stimulus coil location (status + 2 matrices)
5	Coil location and maximum electric field location (status + 2 matrices)
6	Stimulation target location (status + 2 matrices)
7	Delivered electric field pulse information (status + 2 matrices)

Figure 3.7: Table of the Function IDs for the NBS Robotic Interface

The Data field always starts with the NBS System Status, a short integer where each bit is a flag for a specific event. The numerical data exchanged are short integers or floating-point numbers: the firsts are sent as two-byte in little endian order (e.g., 1 is 0x01 0x00); the seconds are codified following the format IEEE 754 in little endian order (e.g., 3.141 is 0x25 0x06 0x49 0x40). A transformation matrix for the location is a 4x4 matrix of floating-point numbers: the upper left 3x3 block is the rotation matrix and represents the orientation of the local coordinate system with respect to the global one; on the other hand, the upper right 3x1 vector is the translation of the local origin with respect to the global one.

$$\begin{bmatrix} r_1 & r_2 & r_3 & p_x \\ r_4 & r_5 & r_6 & p_y \\ r_7 & r_8 & r_9 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Figure 3.8: Structure of a transformation matrix

Bit (LSB=0)	Status (1 is True, 0 is False; - not used)
0	Session open
1	Registration done
2	Exam open
3	Sequence open
4	Head tracker visible
5	Coil visible
6	Repeated stimulus is set active
7	Camera connected
8	Tilt state (when sequence is open, it indicates if the coil is tangential to the scalp surface)
9	Aim state (with repeated stimulus, the coil has reached the correct position and orientation)
10	Stimulation done (after each pulse or sequence)
11	-
12	-
13	Stimulation target is set active
14	-
15	-

Figure 3.9: Table of the Status flags for the NBS Robotic Interface



# Chapter 4

## Franka Emika Panda

Panda is a collaborative robot (cobot, [12]) produced by Franka Emika (Figure 4.1). Cobots are designed to work and collaborate with human operators and other robots, representing an affordable solution to automate processes and support employees. Generally small in size, lightweight and easy to assemble, they can be applied in different environments and carry out various tasks, even in places where there is little room to move. Cobots are specifically made to work safely together with people, as agile and sensitive as a human arm, they do not require to stand in a safety or restricted zone, and the risk of accidents is relatively small. Panda's performance allows processes that require precision, force application and sensitive handling [13]; it can be easily integrated into the existing infrastructure, and it offers a simple programming interface: every technician can be quickly trained, in order to easily develop and deploy tasks to the robot. Panda is a seven axes robot [14], it incorporates the highest mechatronic integration and more than a hundred sensors, such as high-resolution position and high accuracy torque sensors. These allow the robotic arm to sense the surrounding environment, inspired by the human sense touch; moreover, a minimum force of 0.05 N can be applied in delicate tasks, for instance pressing, insertion and screwing. In addition, Panda is equipped with a gravity compensation and friction reduction system, for smooth interactions with humans; several control algorithms allow immediate detection and reaction against accidental collisions and, finally, the flexible torque-controlled joints are useful to adapt to any task or environment.

### 4.1 Programming interfaces

Each Panda robotic arm can be programmed choosing between three different interfaces: the Desk, RIDE or the Franka Control Interface (FCI). The first one is a high-level App-based user interface, developed by the producer Franka Emika, to facilitate and simplify task programming. It is recommended for high-level programs, such as for human-robot interaction studies or quick prototype developments. The second one is a command-based programming environment, to create high performance robot skills that allows the programming of custom Apps and integrating external sensors. The last one, the Franka Control Interface, is a low-level C++ control interface for torque and position, operating at the frequency of 1 kHz, that takes advantage of the available Lagrangian dynamic robot model. Deciding to work with the Desk or RIDE, the user has the possibility to directly connect a computer to the robot via Ethernet cable; on the other hand, selecting the Franka Control Interface, the user connects the local computer to the control unit always through



Figure 4.1: Panda robot by Franka Emika.

Ethernet cable. The control unit is used to bring the power supply and, in the case of the FCI, to carry the control signals for every movement. With FCI, the local computer runs a high-level C++ script that interacts with the interface to command the robotic arm.

## 4.2 MIOS (Machine Intelligence Operating System)

MSRM has developed a C++ Middleware called MIOS (Machine Intelligence Operating System), that operates between the user and the robot, has been developed, among other things, to enable the user to invoke various actions using a specific, so-called, method [15]. MIOS is a complex interface that manages many tasks: for example, it takes care of the robot's administrative credentials; it handles the communication between the robot, its database and the user; it offers a skill-based framework for robot actions, with different control schemes [16]-[17]. The structure of MIOS is composed of several blocks, as shown in Figure 4.2, and among them the most important are the Core, the Interface, the Memory, the Task Engine, and the Skill Engine. The Core acts as a coordinator for the other block, carrying out a surveillance and maintenance job. The communication Interface enables the call to particular functions, called methods, to manage a variety of actions; these calls could be made via RPC (Remote Procedure Call), web socket, TCP, UDP or ROS. A fundamental block for MIOS is the Memory, that is linked to a mongodb database and stores data in json format. The database is a collection of documents referred to two groups: the parameters, and the environment. The Task Engine manages the execution of tasks, substructures that contain the definition of the used skills and parameters for a certain action; the Engine works on a queue, where new tasks may be added, executing each of them after their initialization. Finally, the skills are sets of manipulation primi-

tives, connected in a graph structure like a finite state machine, where each skill could be seen as a state, and the switch between them is determined by the successful completion of the movement primitive. For the development of the TMS Robotic Assistant, MIOS is chosen as the ideal interface to control the robotic arm and the coil.

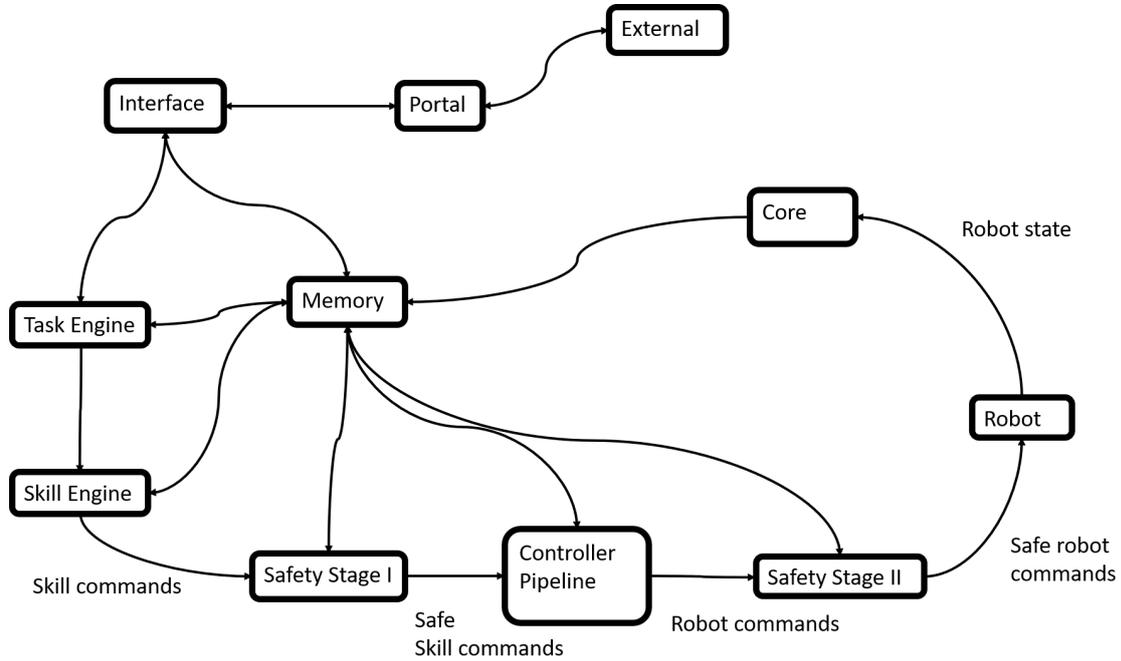


Figure 4.2: An overview of MIOS structure; from [15].



# Chapter 5

## The TMS Robotic Assistant: State of the Art and design phase

Therapy sessions using TMS technique have different duration [18]-[19], however holding the coil for a 30 minute treatment is an exhausting task for a human operator [1]. Rigid or mechanical holder arms could be useful to keep the coil in position during the session, but as long as the patient can slightly move the head, the accuracy of the entire process may be lost. Even if a head resting frame could be used to help the patient maintain a stable position, on the other hand it leads to possible discomfort for the subject, inducing stress and increasing excitability.

The solution to this problem is a robotized navigated TMS system: the coil is no longer held by the operator or by an auxiliary arm, but it is fully controlled by a robotic system. Combining the tracking and navigation system already adopted by the TMS machine, with an artificial mechanism that can autonomously place the coil in the correct position over the patient's scalp, the accuracy of the overall process could be significantly improved: every involuntary movement of the patient, every change of the target location can be compensated and the coil setting automatically adjusted, leading to a positioning error smaller than 2 mm [1]. Nevertheless, it must be considered that, even if the robotic system constantly sends requests to the TMS server to obtain the current target position, due to computation time and communication latency of the feedback loop the corrective actions can't follow instantaneously the head movements, with an overall latency of 200-300 ms [1]. However, this system delay does not compromise the TMS effectiveness: according to studies results, less than 5% of the induced electric field strength is lost after 30 minutes of stimulation [1]. In the next paragraphs, it is first presented the State of the Art for robotized TMS systems and, after that, the TMS Robotic Assistant developed in this project is described following the workflow adopted for its realization, illustrated in Figure 5.1.

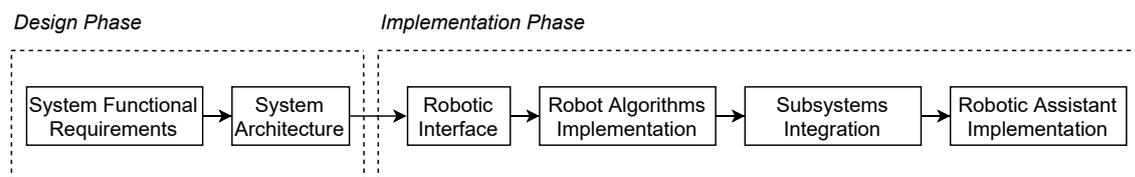


Figure 5.1: Robotic Assistant development phases.

## 5.1 Robotic TMS: State of the Art

Nowadays, only a few solutions for TMS robotic assistants can be found on the market, some of them approved by the main agencies for public health and other research products. Among these, one can cite two products from Axilum Robotics: TMS-Robot and TMS-Cobot [20]. The former has the CE Mark as a class IIa medical device and includes a robotic arm with 7 degrees of freedom and a 2 degree-of-freedom computer-controlled patient seat. It has a hemispherical architecture for the coil positioning areas around a hemisphere, while the head is located inside the robot workspace. It is not piloted by a tracking system. The latter has the CE Mark (EU) as a class IIa medical device, the FDA 510(k) clearance (USA) and it is Registered HAS (Singapore). The TMS-Cobot is compatible with coils produced from several companies, and it can be piloted by the Axilum Robotics own tracking system or via different commercial neuro-navigation systems. It is equipped with a robotic arm mounted to the TMS cart, so the system is lightweight and highly mobile.

Another product intended to be used only for research applications is the Smartmove system by ANT Neuro. It offers a solution that implements the Omron Adept Viper arm and the NDI Spectra camera system, that guarantee high level of accuracy [21]. Smartmove is compatible with any TMS system and any planar coil.

Finally, one can find several research papers on calibration methods and control approaches for robotized TMS systems; in particular, the studies that have been considered as references for this project are the following: Noccaro et al. 2021 [22], Wang et al. 2018 [23] and Yi et al. 2010 [24].

## 5.2 Design phase

The design phase of this project started with the definition of the System Functional Requirements of the final TMS robotic assistant. These requirements are a collection of features that should characterize the final implementation of this system, particularly the standard workflow, the interaction with humans and safety mechanisms.

The second phase of the Assistant design concerns the definition of system architecture, to point out the main parts and understand what kind of protocols are needed for the information exchange among them. Together, the features list and the system architecture map, act also as a reference for the actual development of the Assistant.

### 5.2.1 System Functional Requirements

The functional requirements are grouped into four features/categories, based on their scopes for the Assistant:

1. general,
2. reach and stimulation of cortical target points,
3. collision detection and
4. subject safety mechanism.

The original concept description of these four features is introduced below, and each requirement will be re-evaluated in the discussion chapter, according to the final project implementation

### 5.2.1.1 Feature 1: General

This feature concerns the interaction between the human operator and the TMS robotic assistant. The operator interacts with the system through the NBS Software, that allows him to import the MRI data of the patient, to open a new stimulation session, to point out targets on the MRI and deliver the stimulating pulses. For this category, two are the requirements:

- REQ-01: the TMS robotic assistant starts its function once the registration of the patient's head is accomplished by the operator.
- REQ-02: the pulses can be released only when the operator presses the pedal or clicks on the "Stimulate" button in the stimulation tool on the NBS Software.

### 5.2.1.2 Feature 2: Reach and stimulation of cortical target points

This feature is related to the main task of the TMS robotic assistant: the robot handles the coil and moves it towards the target location on the patient's scalp. When there are no targets active on the NBS Software, the coil is kept in a "ready" position, a spot in proximity of the patient's head, which is convenient to start any movement around it. As soon as the target is set as active on the NBS Software by the operator, the coil leaves the ready position in the direction of the target, in order to start the positioning task. Once the coil reaches the correct location, the Robotic Assistant waits until the end of the stimulation sequence for the target, which is consecutively disabled on the NBS Software. At this point, the coil is moved back to the ready position, and the Robotic Assistant waits until a new target is set as active. For this category, four are the requirements:

- REQ-01: the target position should be predefined via the NBS Software by the operator.
- REQ-02: the TMS robotic assistant should be able to perform the "reach and stimulation" task only if the patient's registration process has been performed.
- REQ-03: the coil should be positioned on the target location and, in case of "repeated stimulus", also with the specified orientation.
- REQ-04: the coil can be considered in the correct target position when:
  - its location and orientation are considered correct by the NBS Software.
  - it is touching the patient's scalp.
- REQ-05: the robot should be able to compensate for the variations of the target location, adjusting the coil position.
- REQ-06: the stimulation can be performed only when REQ-04 is met.
- REQ-07: the TMS robotic assistant should be able to perform more than one stimulation consecutively.

- REQ-08: the robot should return the coil to the default position when the stimulation sequence is finished.

### **5.2.1.3 Feature 3: Collision detection**

This feature concerns the behavior of the robot in case of a contact or an accidental collision. When the robot is enabled, it can freely move inside its workspace: the patient or the operator could be hit accidentally during this window of time. The collision detection algorithm, implemented in the robotic assistant, ensures that in these cases the robot immediately stops. For this category, three are the requirements:

- REQ-01: the robot must stop moving when a collision or a contact is detected.
- REQ-02: the robotic assistant should be able to distinguish a collision from the “move to touch” contact effect of the coil.
- REQ-03: operator intervention is needed to get out of the collision state.

### **5.2.1.4 Feature 4: Subject safety mechanism**

This feature is related to the method used to prevent accidental collision between the robot and the humans in its proximity. It is mandatory that the robot does not accidentally hurt the patient or the doctor while it is moving. As long as the coil and head trackers are seen by the camera, the patient lays on the chair in the correct position and the operator is outside the robot workspace. Considering all this, the robotic assistant must implement an algorithm that checks the feedback from the TMS Software to avoid hitting something if one of the trackers is no longer seen. For this category, six are the requirements:

- REQ-01: when the TMS System is turned off, the robot should return in a safety position.
- REQ-02: the TMS robotic assistant can move from safety to default starting position when the operator gives the permission.
- REQ-03: the TMS robotic assistant can move from default to safety position when the operator gives the permission.
- REQ-04: the robot should stop moving if the coil is not visible by the tracking unit of the TMS System.
- REQ-05: the robot should stop moving if the head tracker is not visible by the tracking unit of the TMS System.
- REQ-06: the robot should stop moving if the digitizing pen is visible by the tracking unit of the TMS System.

### 5.2.2 System Architecture

The second designing step for this project concerns the definition of the System hardware architecture. The two main components that should be interfaced are the TMS System and the Panda robotic arm. The former is constituted by its own cart that carries all the necessary equipment for a normal TMS treatment. The unique interface with it is the prototype NBS Interface, developed by Nexstim for this project, and reachable through an Ethernet cable with traditional RJ45 connector, which is connected to the first slot of the NBS computer network card. The NBS Interface works on top of a client-server model with TCP/IP protocol, where the TMS System acts as the server, listening at port 1001. The main program for the Robotic Assistant is run on an Intel NUC computer: in the final Assistant, the NUC should be connected directly to the NBS computer via Ethernet cable. The Intel NUC computer runs the MIOS Middleware and is connected to the Panda control unit via Ethernet cable, where the FCI is executed. The NUC is also the ideal platform for the execution of the TMS Robotic Assistant software; nevertheless, during the development of the project, the different Assistant versions are tested and debugged on a personal laptop, which is connected to the same local network of the NUC and the TMS System. Finally, a cable connects the control unit to the robotic arm, bringing the power supply and the control signals. A separate industrial type safety switch within reach enables the user to stop any task in execution, locking the robot in its last position. When the user stop is active, the robot can be moved through several control buttons on its pilot interface, on the top of the last junction. The system architecture diagram is shown in 5.2.

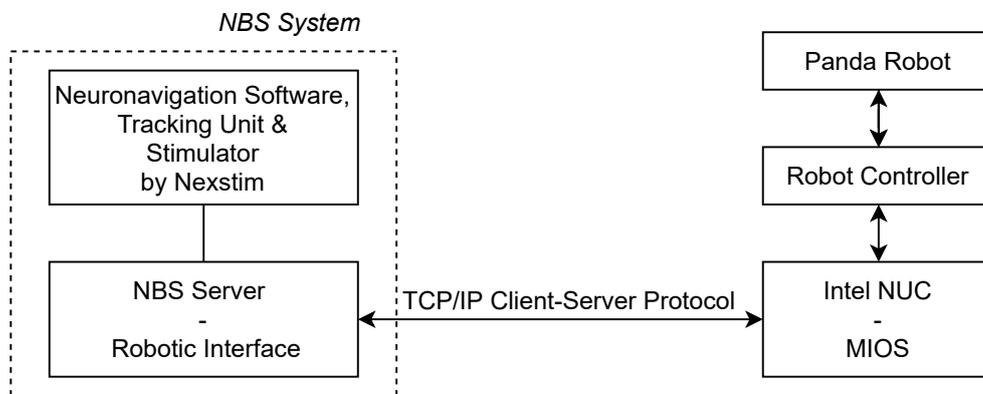


Figure 5.2: Robotic Assistant architecture diagram.



# Chapter 6

## The TMS Robotic Assistant: implementation phase

Once the hardware and software characteristics for the final Assistant are defined, the implementation of the project begins starting from the client-server interface. This is a fundamental step, since it enables the data exchange between the NBS software and the Robotic Assistant, in order to correctly calculate the transformation matrices needed to drive the robotic arm to the targets on the patient's scalp. In particular, the development of the necessary algorithms for the "reach and contact" tasks, as well as the administrative and safety protocols, represents the major activity for the realization of this project. Moreover, the Panda robot has a default-mounted gripper that is not suitable to handle the coil: for this reason, a custom connector is designed (see 6.2) to allow a secure fastening of the coil to the flange of the robotic arm. In addition, an important role is played by the initial calibration task, that is designed to link the robot base frame to the MRI coordinate system, allowing the real fusion of all the subsystems described so far, leading to the concrete realization of the Robotic Assistant. It is constituted by a main script and a collection of libraries all written in Python 3: to communicate with MIOS installed on the NUC, the Assistant uses an additional library, also written in Python, that takes advantage of the WebSocket function call method.

### 6.1 The laboratory set-up

The laboratory environment is set-up in order to take advantage of the already settled equipment and materials. In particular, the robotic arm was already fixed in position on the workstation, with the control unit and the NUC computer connected and ready to work. The Nexstim NBS System has been placed beside the table and close to the robot workspace, thanks to its lockable wheels, while the coil has been fastened to the robot flange through a 3D printed custom connector. The dummy head used for the tests has been fixed in position using a plastic pin as a stand, which keeps the base of the head and a free hole on the table surface aligned; moreover, strips of double-sided tape have been used to stick together the styrofoam head to the table, preventing possible rotations and unwanted movements. This location for the dummy head has been chosen to enable the robot to move the coil around the head without difficulties, with the possibility to reach both the left and right sides of the scalp to simulate the stimulation of the real motor cortex areas. Furthermore, the dummy head has been oriented to facilitate as much as possible the positioning task of the coil, to make both the sides equally accessible,

guaranteeing at the same time that both the head and the coil markers are always visible by the tracking unit, placed in front of the dummy head. Figure 6.1 illustrates the laboratory setup described.

During the Robotic Assistant implementation, a new solution to support the tubes that connect the coil to the Stimulator has been devised. In fact, the TMS System cart is equipped with a supporting arm to which a leather band could be attached using a snap-hook; the band enables the operator to wrap the tubes and support their weight, making the coil easier to move and to position. Even though this set-up was secure, the tubes were long enough to drag on the wooden table, creating friction and conditioning the motion of the robot. To avoid this issue, the leather band supporting the tubes has been replaced with two laces, tied to two different sections along their length. The first one, closer to the NBS System, prevents the pipes from rubbing on the edge of the table, since they are connected to the stimulator placed below the table level; the second lace hangs the tube warding off any kind of drag. This configuration also simplifies the movements of the coil when this reaches the farther points on the scalp, where the pipes are strained and tend to pull the most. The solution described is shown in Figure 6.2.



Figure 6.1: Laboratory setup.



Figure 6.2: Support laces for cables and cooling tubes.

## 6.2 The custom coil connector

Franka Emika's Panda robots are usually equipped with a gripper end effector, that allows it to carry out many tasks where a secure grasp and high precision movements are required (e.g., insert a key in its lock to open it). Nevertheless, the standard Panda's gripper is not suitable to hold and handle the coil, for two main reasons: the coil weight and its connection tubes to the TMS System. In fact, even designing an appropriate 3D printed part that would enable the gripper to grasp the cylindrical handle of the coil, the center of mass of the final end-effector would be too far away from the robot flange; this would lead to a hard system to control, considering that the cooled coil used for this project weighs about 2.2 kg. Moreover, the pipes connected to the rear part of the coil (that bring the electrical wires and the cooling air flows) have not a negligible weight, also presenting a limited flexibility; these aspects negatively influence the movements of the coil, even

in normal conditions when it is handled by a human operator. For all these reasons, it is chosen to design a custom connector that enables attaching the coil directly to the robot flange, without using the gripper and setting suitably the robot parameters, such as the weight compensation.

The custom coil connector was designed in SolidWorks CAD, and the prototypes were manufactured using a 3D printer (Prusa i3 MK3S+). All the connectors realized for this project are based on the attachment system used by the coil positioning holder given with the Nexstim chair equipment. The positioning holder is a metal supporting arm that could be fixed to the patient's reclining chair, and it could be by the operator to keep the coil in position during the TMS session. The holder ends with a shaped pin that is designed to be inserted in a specific hole of the coil handle; to lock the coil in position, a screw on the side of the pin housing is tightened, and the operator can release the grip on the coil. Taking advantage of the same connection mode, the custom connectors have a base that perfectly fit the Panda robot flange, which follows the standard DIN ISO 9409-1-A50; two screws allow a secure fastening of the connector, and so of the coil, to the robotic arm. The first version of the connector is shown in Figure 6.3 left.

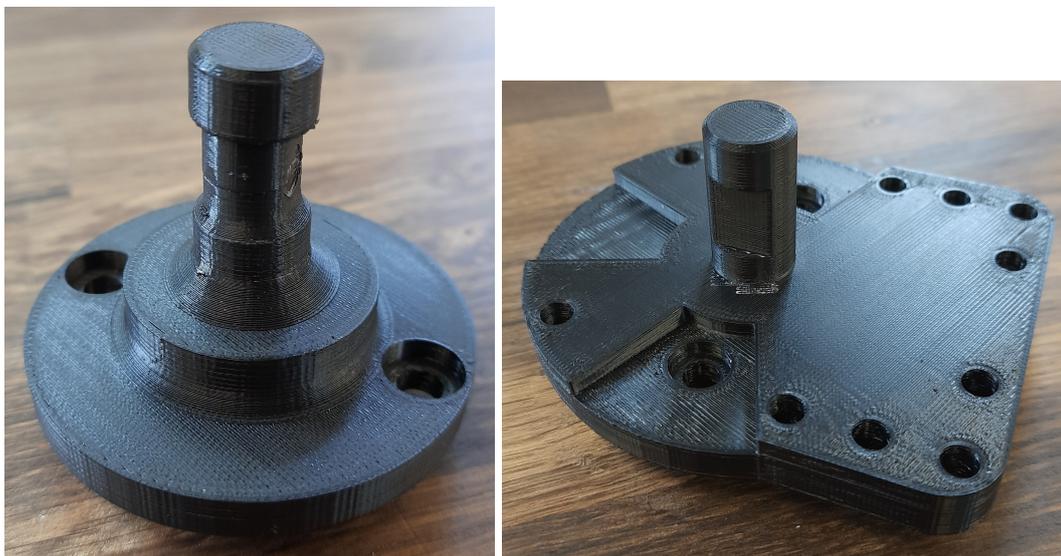


Figure 6.3: Custom connectors: first version (left) and last realization (right).

The last custom connector realized (Figure 6.3 right) and used for the experiments is the result of four previous trial versions, and it solves different issues that appeared during the tests. Indeed, a stronger kind of plastic has been adopted, to reduce the risk of breaks in correspondence with the weakest point of the structure, also making the connector suitable to support the coil weight. In addition, cable ties are adopted to fasten the coil to the connector by wrapping the handle, decreasing the mechanical stress on the pin and preventing the fall of the coil to the table. The implemented solution is shown in Figure 6.4

Nevertheless, during the implementation of the Assistant, it has been noticed that, due to the tubes connected on the rear side, the actual position of the coil was slightly modified by a push-pull action of the pipes themselves; in particular, the coil tends to rotate around the vertical axis passing through the connecting pin. This is also due to a flaw of the plastic housing of the pin, part of the coil handle, in correspondence of the screw hole: in fact, the plastic here is not perfectly shaped and a hole larger than the screw allows

circular movement of the handle around the pin, even if the screw itself is well tighten. To prevent and attenuate this effect, the screw hole of the pin housing of the handle has been partially filled with a malleable material, that allows anyway to fasten the screw without leaving empty spaces; in addition, the plastic ties used to secure the coil to the connector are tighten in such a way to reduce potential circular movements.



Figure 6.4: Panda robot equipped with the last connector version; the coil is fasten with cable ties to the connector base.

### 6.3 The end-effector parameters set-up

In this paragraph, it is given a brief description of the robotic arm end-effector set-up. In fact, through the Desk interface of the robot, accessible via a direct connection to the robot itself with Ethernet cable, the user has the possibility to modify the parameters that characterize a specific end-effector. Figure 6.5 illustrates part of the settings page. In particular, one can set:

- the mass of the load,
- the flange to load center of mass vector,
- the inertia tensor,

- the transformation matrix from flange to end-effector.

The coil weighs around 2.2 kg, but the weight of the tubes connected should also be estimated. After some practical tests, a total load weight estimation of 2.5 kg has given good results in terms of the robot weight compensation during the movements, and for this reason it has been set as the mass parameter. The translation vector of the transformation matrix to the end-effector, that is the center of the coil front surface, has been measured with the aid of a caliber and a measuring tape, while the rotation matrix has been calculated to properly align the coil handle to the robot control board on the flange side. Knowing the transformation matrix, the center of mass vector has been estimated accordingly, trying to also consider the pipes influence. Some tests have been carried out in order to find the best-fitting vector for the mass, leading to a result that enables a good weight compensation during the coil movements. Finally, the inertia tensor has been roughly estimated considering the coil 3D model realized with SolidWorks (Mass Properties Window). Even though the coil is well controlled by the robot, this estimated tensor should be intended as a rough approximation of the real one, where the effect of the pipes should be taken into account.

Mass			
<u>2.5</u>			kg
Flange to Center of Mass of Load Vector			
<u>0.03</u>	<u>0.03</u>	<u>0.1</u>	m
Inertia Tensor			
<u>0.012</u>	<u>0</u>	<u>0</u>	
<u>0</u>	<u>0.012</u>	<u>0</u>	kg x m <sup>2</sup>
<u>0</u>	<u>0</u>	<u>0.012</u>	
Transformation Matrix from Flange to End-Effector			
<u>0.7071</u>	<u>0.7071</u>	<u>0</u>	<u>0.122</u>
<u>-0.7071</u>	<u>0.7071</u>	<u>0</u>	<u>0.122</u>
<u>0</u>	<u>0</u>	<u>1</u>	<u>0.11</u>
<u>0</u>	<u>0</u>	<u>0</u>	<u>1</u>

Figure 6.5: Desk interface; the end-effector parameters setup.

## 6.4 The Robotic Assistant client interface

As previously mentioned in chapter 3.2.4, the communication between the TMS Software and the Robotic Assistant stands on top of a Client/Server model, using the TCP/IP protocol: the former performs the server role, while the latter is the client. The first operation that the client should do to establish a connection with the server is the creation of a socket. A socket is defined as an endpoint, that is an attachment point for a two-way communication link between two devices or two programs running on the network [25]; a socket is characterized by the couple IP address and port number: this information is used by the TCP layer to identify the application that should receive the incoming data. In this case of study, the port number is 1001. After the creation of the socket, the client can establish the connection with the server: in case the procedure is successful, the client can formulate and send the first request packet. This last one is composed by a two bytes request tag, that starts from 0 at the beginning of the communication; a two bytes field for the length of the message (in bytes) from this point forward, that is always 2 for the request packets; finally, a two bytes field for the function ID, that could be a number between 1 and 7. All the short integers in each field are sent following the little endian order. After the request is sent, the server replies with the data correspondent to that specific function ID: the client listens on the socket and receives the data in a 1024 bytes buffer. This information is split into different fields, namely: the tag, corresponding to the request one; the message length from this point forward; the function ID requested; the error code; the TMS System status and finally, if necessary, additional data such as transformation matrices or E-field maximum values. Similarly to the request case, also the response fields are received in little endian order. These data are then analyzed by the Robotic Assistant, according to the function or the task to be carried out at that moment. If no other requests are needed to the client, it can proceed with the socket closing, in order to release the communication channel and the port that return available and ready for a new link. The client flowchart is shown in Figure 6.6 left, while in Figure 6.6 right is reported a screenshot of a typical response from server to a request with Function ID 4.

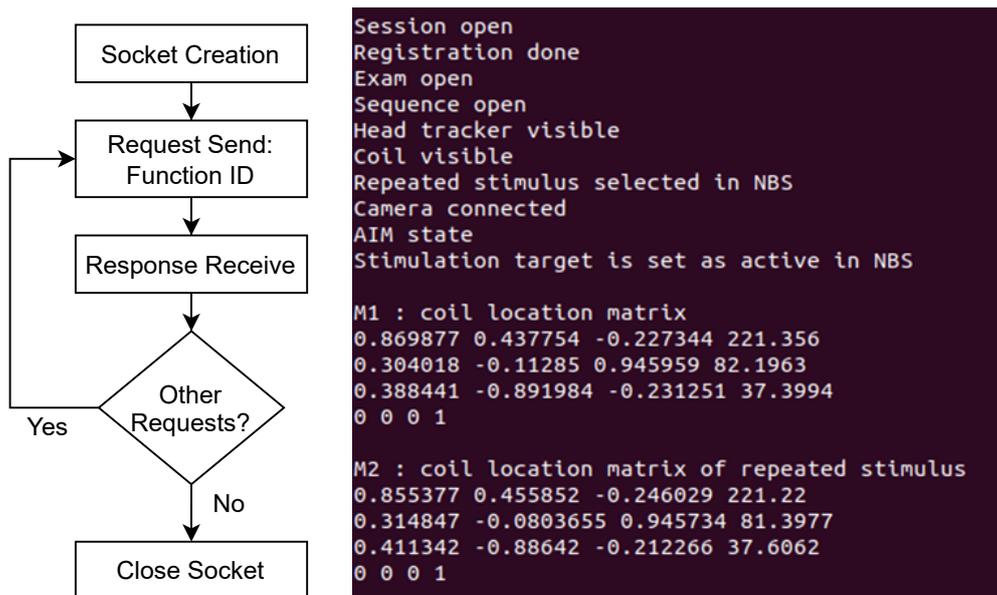


Figure 6.6: TCP client flowchart (left); screenshot from the interface, response to Function ID 4 (right).

## 6.5 The calibration task

The aim of the initial calibration task of the system is the calculation of the transformation matrix from the robot base frame to the MRI coordinate system. In several papers, such as [22] and [23], calibration algorithms like the QR24, QR36, Stochastic Global Optimization and Quaternion Approach are proposed, showing evidence of their good performances. However, they all consider the transformation matrices from flange to end-effector and from robot to camera coordinate systems as unknowns.

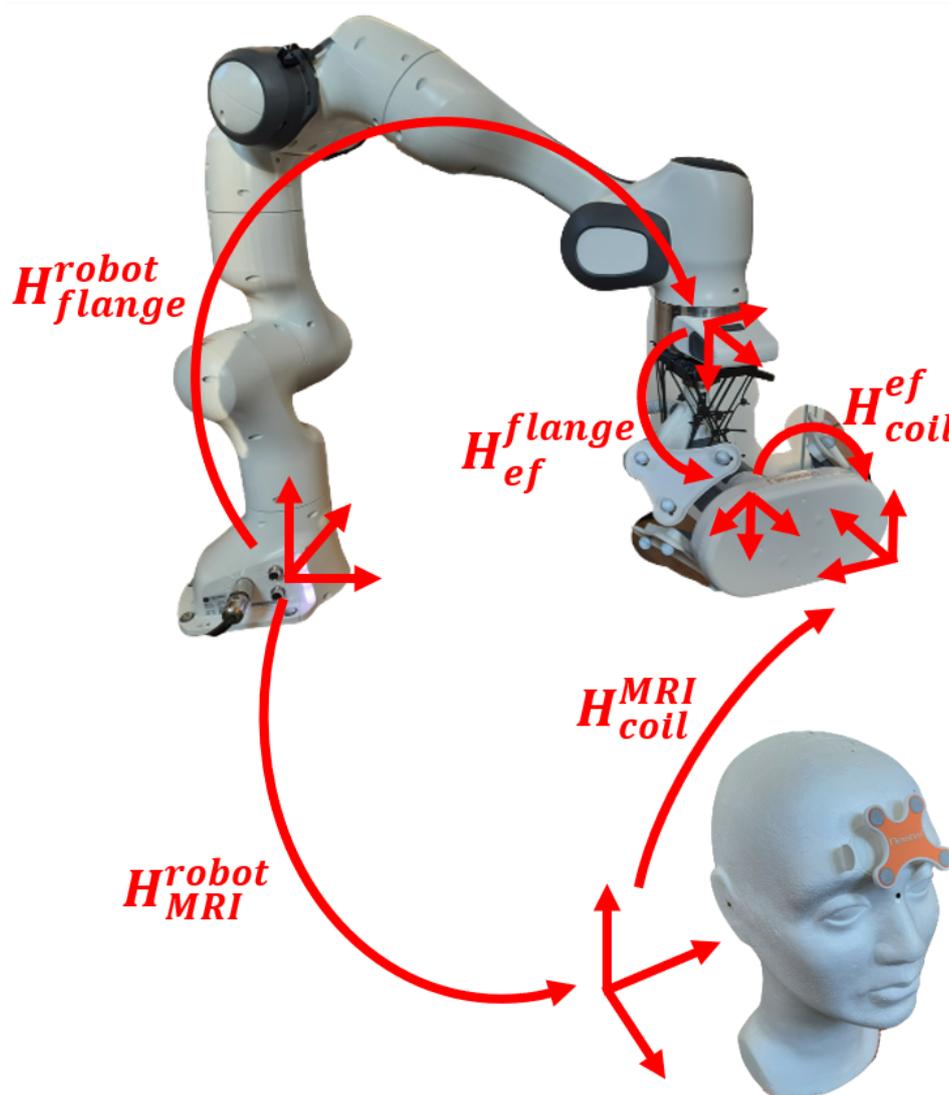


Figure 6.7: The main coordinate systems for the project with the associated transformation matrices.

In this project, as mentioned in 6.3, the transformation matrix from the flange to the end-effector is measured during the set-up phase of the system. This means that the only unknown for this case of study is the transformation from the robot to the MRI frames. For this reason, a different method has been chosen, formally named as Orthogonal Procrustes problem, originally formulated by Schoenemann in 1966. An illustration of the main coordinate systems for this project is reported in Figure 6.7. The Procrustes method consists in a least-squares tool to directly estimate the transformation between two points

sets up to their maximal agreement, avoiding the definition and solution of the classical normal equation systems [26]. When unknown rotation, unknown translation and unknown scale factor are present between the two sets, the problem takes the name of Extended Orthogonal Procrustes, formulated in 1970 by Schoenemann and Carroll. After this publication, similar methods were proposed in computer vision and robotic areas, for example by Arun et al. in 1987 [27], which represents the reference algorithm for the calibration task of this project.

Given two sets of points in the three-dimensional space,  $\{p_i\}$  and  $\{p'_i\}$ , with  $i = 0, \dots, N$  and  $p_i, p'_i$  column vectors, one can write:

$$p'_i = Rp_i + T + e_i, \quad (6.1)$$

where  $R$  is a 3x3 rotation matrix,  $T$  is a column translation vector and  $e_i$  and error vector. The aim of the method is to minimize the square error:

$$E^2 = \sum_{i=1}^N \left\| p'_i - (Rp_i + T) \right\|^2. \quad (6.2)$$

It was demonstrated in [28] that, given  $\hat{R}$  and  $\hat{T}$  the solution to (6.1), then

$$p' = p'', \quad (6.3)$$

where  $p'_i$  and  $p''_i$  are the centroids of  $\{p_i\}$  and  $\{p'_i\}$ , defining:

$$p' = \frac{1}{N} \sum_{i=1}^N p'_i, \quad (6.4)$$

$$p'' = \frac{1}{N} \sum_{i=1}^N p_i = \frac{1}{N} \sum_{i=1}^N (\hat{R}p_i + \hat{T}). \quad (6.5)$$

Imposing

$$p = \frac{1}{N} \sum_{i=1}^N p_i, \quad (6.6)$$

the following can be written:

$$q_i = p_i - p, \quad (6.7)$$

$$q'_i = p'_i - p'. \quad (6.8)$$

The original problem can be reformulated as:

$$E^2 = \sum_{i=1}^N \left\| q'_i - Rq_i \right\|^2 \quad (6.9)$$

and  $\hat{R}$  is found minimizing equation (6.2), while

$$\hat{T} = p' - \hat{R}p \quad (6.10)$$

According to [27], a SVD based algorithm can be implemented in order to find  $\hat{R}$  and  $\hat{T}$  and could be divided in steps as shown below.

Step 1: Calculate the centroids  $p$  and  $p'$  using (6.6) and (6.4) respectively.

Step 2: Translate each set to the origin, using (6.7) and (6.8).

Step 3: Step 3: Calculate the 3x3 matrix

$$H = \sum_{i=1}^N q_i q_i'^T \quad (6.11)$$

Step 4: Apply the SVD method to H, to find

$$H = UDV^T \quad (6.12)$$

Step 5: Calculate the new matrix

$$X = VU^T \quad (6.13)$$

Step 6: Correct X for special reflection case, calculating its determinant as follows:

$$d = \text{sign}(\det(X)) \quad (6.14)$$

$$X = V \text{diag}(1, 1, d) U^T \quad (6.15)$$

Step 7: Compute  $\hat{R}$  as:

$$\hat{R} = X \quad (6.16)$$

and  $\hat{T}$  as in equation (6.10).

For more information on the algorithm the reader can refer to [27].

After testing the calibration algorithm explained above, it has been chosen a number of calibration points equal to 6; moreover, the points were selected so that it is guaranteed that at least two clusters of coil markers are visible by the tracking unit, in order to maximize the localization accuracy of the coil itself. When the robot reaches one of the points, it is identified by two vectors with different frames: one is the current translation vector from the robot base frame to the center of the coil; the other is the translation vector of the coil location with respect to the MRI coordinate system. The points indicated by these two vectors are appended to two different sets of points; naming A and B the two sets, they are defined as follows:

$$A = p_i^r \quad (6.17)$$

$$B = p_i^{MRI} \quad (6.18)$$

where  $p_i^r$  is the translation vector from the robot coordinate system to the coil that identifies the i-th, while  $p_i^{MRI}$  is the translation vector from the MRI frame to the coil, identifying the same point, with  $i = 1, \dots, 6$ .

The transformation matrix  $H_{MRI}^{robot}$ , which is the output of the calibration function, is calculated using the SVD based algorithm on these sets. Last, in order to check the accuracy of the procedure, the calibration function estimates the error associated with the

calculated matrix, as explained below. Let's assume the following matrices:

$$H_{p_i}^{MRI} = \begin{bmatrix} 1 & 0 & 0 & p_{i,x}^{MRI} \\ 0 & 1 & 0 & p_{i,y}^{MRI} \\ 0 & 0 & 1 & p_{i,z}^{MRI} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.19)$$

$$\hat{H}_{p_i}^r = \begin{bmatrix} 1 & 0 & 0 & \hat{p}_{i,x}^r \\ 0 & 1 & 0 & \hat{p}_{i,y}^r \\ 0 & 0 & 1 & \hat{p}_{i,z}^r \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.20)$$

where  $p_{i,x}^{MRI}$ ,  $p_{i,y}^{MRI}$  and  $p_{i,z}^{MRI}$  are the coordinates of  $p_i^{MRI}$ , while  $\hat{p}_{i,x}^r$ ,  $\hat{p}_{i,y}^r$  and  $\hat{p}_{i,z}^r$  are the elements of the estimated vector  $\hat{p}_i^r$ . Multiplying the computed transformation matrix from the robot base frame to the MRI frame by each transformation matrix associated to the coil location with respect to the MRI, it is possible to obtain the estimated transformation matrix from the robot base frame to the coil:

$$\hat{H}_{p_i}^r = H_{MRI}^{robot} H_{p_i}^{MRI} \quad (6.21)$$

The estimated points are then compared to saved correct ones, and the root mean square error between these two sets is calculated:

$$\Delta p_i^r = p_i^r - \hat{p}_i^r \quad (6.22)$$

$$RMSE = \sqrt{\frac{1}{6} \sum_{i=1}^6 (\Delta p_i^r)^2} \quad (6.23)$$

As a side note, during the calibration task, the robotic arm moves sequentially to the defined points that surround the dummy head and there is no risk for the robot to hit it during the movements. For the duration of each movement, a NBS status watchdog monitors that both the coil and the head markers remain visible by the camera, stopping the movement in case one of the two are no more seen.

## 6.6 The Assistant algorithms

In this paragraph, the main algorithms that constitute the Robotic Assistant proposed in this project are described, and they could be associated to three macro-groups:

- I. navigation,
- II. Panda task functions,
- III. safety.

### 6.6.1 Navigation algorithms

To allow the movements of the coil around the patient's head without falling into the risk of accidentally hitting it, a trajectory planning algorithm has been developed, to ensure a safe navigation of the workspace. This algorithm is based on the construction of a virtual sphere that surrounds the head: the sphere surface represents a track where the coil can be moved following simple navigation patterns, in order to reach every point of the scalp.

### 6.6.1.1 Virtual sphere navigation

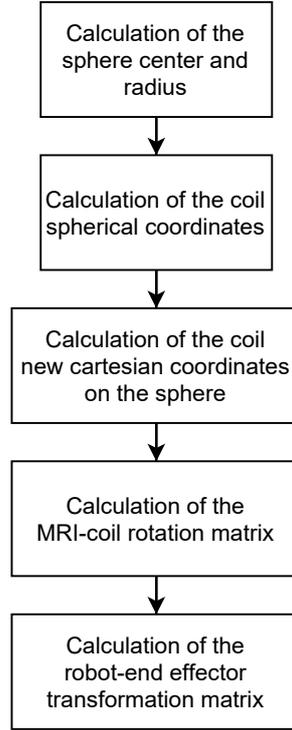


Figure 6.8: The sphere navigation flowchart.

All the location vectors used henceforth are expressed with respect to the MRI coordinate system. The sphere is built starting from the locations of the registered ears landmarks, sending the specific request to the NBS Robotic Interface. The central point vector is then calculated, leading to the definition of the center  $p_C^{MRI}$  and the radius  $r$  of the sphere, as shown below:

$$p_C^{MRI} = \frac{p_1^{MRI} + p_2^{MRI}}{2} + d_{off} \quad (6.24)$$

$$r = \left\| p_2^{MRI} - p_C^{MRI} \right\| + d_{ss} \quad (6.25)$$

where  $p_1^{MRI}$  and  $p_2^{MRI}$  are the two landmarks' locations, while  $d_{ss}$  represents the distance that should be kept between the scalp and the sphere surface (that is also the coil surface). It is noticed that, in order to better fit the head shape with the spherical approximation, the central point is moved by an offset  $d_{off}$ , which has been fixed to  $[0, 0.05, -0.02]$  for the dummy head case.

To point out the current position of the coil on the sphere, as well as the location of a target, a dedicated function has been written to return the spherical coordinates of the searched point. In fact, given for example a target coordinates  $p_t^{MRI}$ , its position vector  $v_t^C$  with respect to the sphere center is calculated, as well as its distance  $d_{ct}$  from the center. After that, the angles  $\varphi_t$  from the z vertical axis, and  $\vartheta_t$  from the horizontal y axis, are calculated from the data previously collected, as follows:

$$v_t^C = p_t^{MRI} - p_C^{MRI} \quad (6.26)$$

$$d_{ct} = \left\| v_{coil}^C \right\| \quad (6.27)$$

$$\varphi_{coil} = \arccos \frac{v_{coil,y}^C}{d_{cc}} \quad (6.28)$$

$$\vartheta_{coil} = \arctan 2 \frac{v_{coil,x}^C}{v_{coil,z}^C} \quad (6.29)$$

The movements of the coil over the sphere surface are performed following a specific pattern: the adopted method for the assumes that the robot moves first on the longitude from the starting point to the destination  $\vartheta$ , keeping the same  $\varphi$  coordinate; then, to reach the destination, the coil is moved along the latitude to the final  $\varphi$ . These two paths from a starting point to the final location are split in shorter segments, thus the coil is moved by steps along the intermediate checkpoints, always keeping an acceptable distance from the patient's head. Another approach is to unify the two phases of the movement (along longitude first, then along latitude), creating a path that goes directly to the destination point, taking into account both  $\vartheta$  and  $\varphi$  steps for each segment. For the experiments with the Robotic Assistant, using the two-phases movements, the longitude steps ( $\Delta\vartheta$ ) are equal to 30 degrees, while the latitude ones ( $\Delta\varphi$ ) are set to 20 degrees. On the other hand, with the direct movement method, the  $\Delta\varphi$  steps have been set to 10 degrees.

The spherical coordinates for each checkpoint are calculated one after the other, each time the coil reaches an intermediate spot; thus, they are converted into the new cartesian coordinates  $v'_x$ ,  $v'_y$  and  $v'_z$ , as shown below.

$$v'_x = r \cdot \sin(\varphi_{coil} + \Delta\varphi) \cdot \sin(\vartheta_{coil} + \Delta\vartheta) \quad (6.30)$$

$$v'_y = r \cdot \cos(\varphi_{coil} + \Delta\varphi) \quad (6.31)$$

$$v'_z = r \cdot \sin(\varphi_{coil} + \Delta\varphi) \cdot \cos(\vartheta_{coil} + \Delta\vartheta) \quad (6.32)$$

Nevertheless, the previous coordinates are not sufficient to adequately control the coil: in fact, a rotation matrix should be calculated for each of them, in order to place the coil surface tangentially to the sphere. To get this result, a preliminary rotation of the MRI coordinate system should be performed, in order to pass to the coil frame; after that, the coordinates  $\varphi_{coil} + \Delta\varphi$  and  $\vartheta_{coil} + \Delta\vartheta$  are used to compute two rotations, about the axes x and z of coil coordinate system. Defining

$$\alpha = (\varphi_{coil} + \Delta\varphi) - \frac{\pi}{2} \quad (6.33)$$

$$\gamma = (\vartheta_{coil} + \Delta\vartheta) \quad (6.34)$$

the following are the rotations needed:

$$R_{coil}^{MRI} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad (6.35)$$

$$R_{x,\gamma} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{bmatrix} \quad (6.36)$$

$$R_{z,\alpha} = \begin{bmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (6.37)$$

where  $R_{coil}^{MRI}$  represents a rotation of 90 degrees about the z axis, followed by a 90 degrees rotation about the x axis, to pass from the MRI frame to the coil one. Thus, the rotation matrix from the MRI frame to the new coil frame, suitable for the latter to be tangential to the sphere is:

$$R_{coil,sphere}^{MRI} = R_{coil}^{MRI} R_{x,\gamma} R_{z,\alpha} \quad (6.38)$$

At this point, all the information needed to compute the transformation matrix to control the robot are available. Following the transformation chain rule, the transformation matrix from the robot base frame to the end-effector frame, to be tangential to the sphere, will be the following:

$$H_{ef,sphere}^{robot} = H_{MRI}^{robot} H_{PC}^{MRI} H_{coil,sphere}^{PC} H_{ef}^{coil} \quad (6.39)$$

The first transformation  $H_{MRI}^{robot}$  is already available. Hence, let's start from the transformation matrix from MRI coordinate system to the sphere center  $p_C^{MRI}$ :

$$H_{PC}^{MRI} = \begin{bmatrix} 1 & 0 & 0 & p_{C,x}^{MRI} \\ 0 & 1 & 0 & p_{C,y}^{MRI} \\ 0 & 0 & 1 & p_{C,z}^{MRI} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.40)$$

where  $p_{C,x}^{MRI}$ ,  $p_{C,y}^{MRI}$  and  $p_{C,z}^{MRI}$  are the three components of  $p_C^{MRI}$ . After that, it follows the definition of the transformation matrix from the center of the sphere  $p_C^{MRI}$  to the new coil frame position:

$$H_{coil,sphere}^{PC} = \begin{bmatrix} R_{coil,sphere}^{MRI} & v' \\ 0 & 1 \end{bmatrix} \quad (6.41)$$

Finally, a new transformation matrix should be defined to pass from the coil coordinate system to the end-effector frame:

$$H_{ef}^{coil} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.42)$$

which represents a 180 degrees rotation about the z axis, followed by 90 degrees about the y axis. In Figure 6.9 are reported the end-effector frame (left) and the coil coordinate system (right).

### 6.6.1.2 Reaching tasks

The following paragraphs explain the algorithms of the “reach and stimulate” tasks for “normal” stimulation targets and repeated stimulus targets. In particular, the former are locations on the cortex pointed out by the operator on the MRI images through the NBS

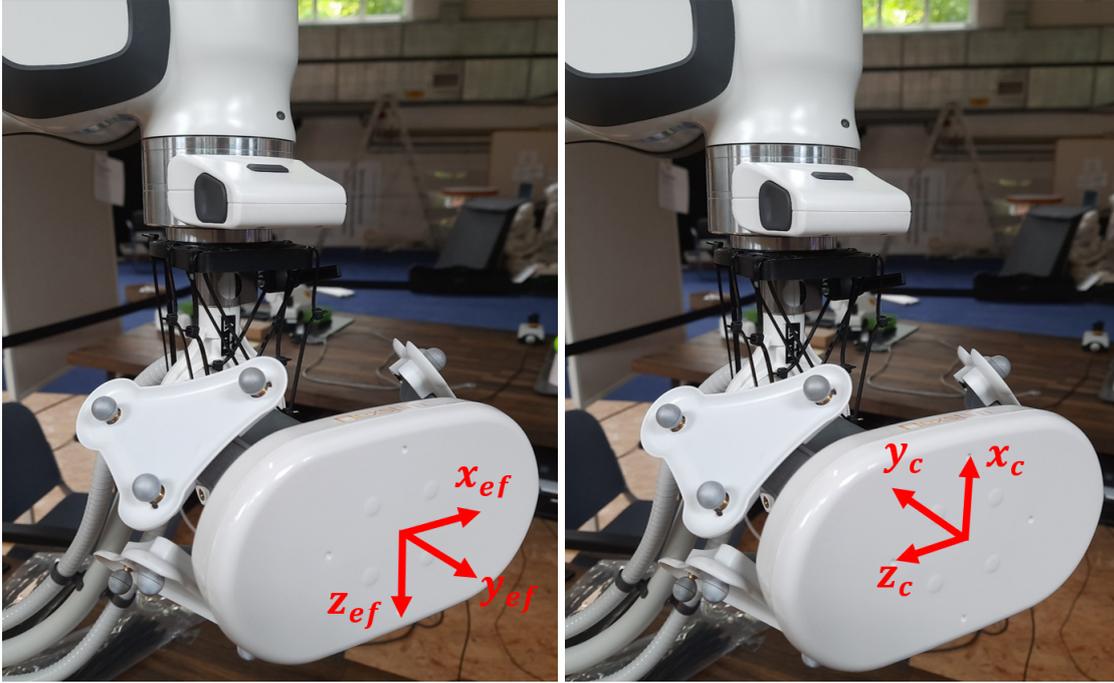


Figure 6.9: The end-effector coordinate system (left); the coil coordinate system (right).

Software. The data related to these kinds of targets consist of a transformation matrix composed by an identity 3x3 rotation matrix and a translation vector  $p_t^{MRI}$  indicating the position in the 3D space, as shown below.

$$H_{std,target}^{MRI} = \begin{bmatrix} 1 & 0 & 0 & p_{t,x}^{MRI} \\ 0 & 1 & 0 & p_{t,y}^{MRI} \\ 0 & 0 & 1 & p_{t,z}^{MRI} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.43)$$

This means that no information is given by the NBS Software for the coil orientation to approach these targets. The repeated stimulus targets are locations on the cortex that have already been stimulated in a previous stimulation sequence and, for this reason, the information related to the coil orientation is available via the Robotic Interface, beyond the translation vector of the target on the scalp:

$$H_{rep,target}^{MRI} = \begin{bmatrix} R_t^{MRI} & p_t^{MRI} \\ 0 & 1 \end{bmatrix} \quad (6.44)$$

#### • “Standard” stimulation targets reaching task

To move the coil to the position of a “normal” stimulation target, the algorithm starts with two requests to the NBS Interface: the first to obtain the transformation matrix of the target on the scalp, the second to get the current position of the coil. After that, the spherical coordinates of the target and of the coil are calculated, in order to move the latter in correspondence to the target projection on the virtual sphere. In this position, the coil is tangent to the virtual sphere itself, and the same rotation matrix (that will be indicated with  $R_{coil,sphere}^{MRI}$  is used also to approach the target, since no information is provided for this operation. Thus, the transformation matrix for the robot is calculated and the first

attempt to teach the target is performed, as shown below.

$$H_{target}^{robot} = H_{MRI}^{robot} \begin{bmatrix} R_{coil,sphere}^{MRI} & p_t^{MRI} \\ 0 & 1 \end{bmatrix} H_{ef}^{coil} \quad (6.45)$$

Once the coil has been positioned, a new request is sent to the Robotic Interface, to get its current position with respect to the MRI. The difference  $d$  between the target nominal location  $p_t^{MRI}$  and the current coil spot  $p_{coil}^{MRI}$  is calculated, and a new transformation matrix for the robot is computed, taking into account the compensation of the difference:

$$d = p_t^{MRI} - p_{coil}^{MRI} \quad (6.46)$$

$$H_{target,comp}^{robot} = H_{MRI}^{robot} \begin{bmatrix} R_{coil,sphere}^{MRI} & p_t^{MRI} + d \\ 0 & 1 \end{bmatrix} H_{ef}^{coil} \quad (6.47)$$

At this point, the coil is moved back to the sphere and the second positioning attempt can be performed, using the last transformation matrix calculated. It may happen that, after the second attempt, the difference between the coil location and the target is higher than the accuracy threshold of 2 mm. Unfortunately, the lack of a precise rotation matrix to approach the target significantly influences the positioning task accuracy for “standard” stimulation targets; to try to reduce the positioning error, if it is too high (a threshold of 4.5 mm has been set for the tests), the procedure is repeated starting from the first attempt.

- **Repeated stimulus targets reaching task**

The reaching tasks for repeated targets begin in the same way as the normal targets: the specific request to get the target location is sent to the Robotic Interface, then its spherical coordinates are calculated, and the coil is moved in correspondence of its projection on the virtual sphere surface. When the coil is ready to reach the target, the transformation matrix for the robot is obtained, similarly to the “standard” case explained before; nevertheless, in this case there the rotation matrix to approach the target is available:

$$H_{target}^{robot} = H_{MRI}^{robot} H_{rep,target}^{MRI} H_{ef}^{coil} \quad (6.48)$$

Before the first attempt starts, the coil is rotated so that it acquires the correct orientation to reach the target, maintaining its position on the virtual sphere. The transformation matrix for this movement is the following:

$$H_{ef,sphere,rep}^{robot} = \begin{bmatrix} R_{target}^{robot} & p_{ef,sphere}^{robot} \\ 0 & 1 \end{bmatrix} \quad (6.49)$$

where  $R_{target}^{robot}$  is the rotation part of  $H_{target}^{robot}$ , while  $p_{ef,sphere}^{robot}$  is the translation vector of  $H_{ef,sphere}^{robot}$  when the coil is located on the target projection on the virtual sphere. After this preparatory movement, the first attempt is performed. When the coil reaches the destination position, a request is sent to the Interface in order to get the current location of the coil, then it is moved back to the virtual sphere. Similarly to the previous case, the difference between the nominal target  $p_t^{MRI}$  and the coil spot  $p_{coil}^{MRI}$  is calculated, but it is additionally computed also the discrepancy between the Euler angles as shown below [29]. Indicating with  $r_{coil,i}^{MRI}$  the  $i$ -th element of the coil rotation matrix (with respect to the MRI frame), it is:

$$d = p_t^{MRI} - p_{coil}^{MRI} \quad (6.50)$$

$$s = \sqrt{r_{coil,1}^{MRI\ 2} + r_{coil,4}^{MRI\ 2}} \quad (6.51)$$

If  $s > 10^{-6}$  (non-singular) then:

$$\alpha_{coil}^{MRI} = \arctan 2 \frac{r_{coil,8}^{MRI}}{r_{coil,9}^{MRI}} \quad (6.52)$$

$$\beta_{coil}^{MRI} = \arctan 2 \frac{-r_{coil,7}^{MRI}}{s} \quad (6.53)$$

$$\gamma_{coil}^{MRI} = \arctan 2 \frac{r_{coil,4}^{MRI}}{r_{coil,1}^{MRI}} \quad (6.54)$$

$$(6.55)$$

else:

$$\alpha_{coil}^{MRI} = \arctan 2 \frac{-r_{coil,6}^{MRI}}{r_{coil,5}^{MRI}} \quad (6.56)$$

$$\beta_{coil}^{MRI} = \arctan 2 \frac{-r_{coil,7}^{MRI}}{s} \quad (6.57)$$

$$\gamma_{coil}^{MRI} = 0 \quad (6.58)$$

where  $\alpha_{coil}^{MRI}$ ,  $\beta_{coil}^{MRI}$  and  $\gamma_{coil}^{MRI}$  are the Euler angles respectively about x, y and z axes of MRI coordinate system. Then:

$$\Delta\alpha = \alpha_{target}^{MRI} - \alpha_{coil}^{MRI} \quad (6.59)$$

$$\Delta\beta = \beta_{target}^{MRI} - \beta_{coil}^{MRI} \quad (6.60)$$

$$\Delta\gamma = \gamma_{target}^{MRI} - \gamma_{coil}^{MRI} \quad (6.61)$$

With this information known, it is calculated a compensation for the transformation matrix of the first attempt; to do this, a transformation matrix to correct the rotation of the coil around its x axis is defined as:

$$R_{coil,\Delta\alpha} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Delta\alpha & -\sin\Delta\alpha \\ 0 & \sin\Delta\alpha & \cos\Delta\alpha \end{bmatrix} \quad (6.62)$$

Thus:

$$R_{target,comp}^{robot} = R_{target}^{robot} R_{coil,\Delta\alpha} \quad (6.63)$$

$$H_{target,comp}^{robot} = H_{MRI}^{robot} \begin{bmatrix} R_{target,comp}^{robot} & p_t^{MRI} + d \\ 0 & 1 \end{bmatrix} H_{ef}^{coil} \quad (6.64)$$

With all this known, the orientation of the coil is corrected, and the second attempt can be performed. In case the aiming tool does not give the “aim” status (green ball on the GUI of the NBS Software), which means that the difference between target and coil is still greater than 2 mm, the positioning process is repeated, with the variance that three attempts are performed. The procedure for the first attempt remains unvaried; the second one is performed compensating only the difference of the translation vector; finally, the third attempt compensates for the discrepancy of the Euler angle about the x axis. If, after this sequence, the “aim” status is still not active, the 3-steps procedure is repeated until the difference between the target and coil spots drops under 2 mm.

### 6.6.1.3 Area stimulation task

Finally, to complete the navigation algorithms section, a function has been implemented, to enable the operator to stimulate the area surrounding a “standard” target. Once the stimulation sequence has been completed and closed for the latter, if the target remains active for the following 20 seconds, the robotic arm automatically moves the coil to the first point of a sequence that consists into three circular patterns surrounding the active target. The three patterns have closed circular shapes, and they are composed respectively by 12, 18 and 24 points, leaving 30, 20 and 15 degrees one from each other on the circumferences. To calculate the starting point of each circle, the spherical coordinates  $\vartheta_m$  and  $\varphi_m$  of the coil position after the main target stimulation are computed and saved as the center; the first point of the inner circle is located at  $(\vartheta_m + 2deg; \varphi_m)$ , the first of the central one at  $(\vartheta_m + 4deg; \varphi_m)$  and the external at  $(\vartheta_m + 6deg; \varphi_m)$ . Each pattern is then covered in a counterclockwise direction. An example is shown in Figure 6.10.

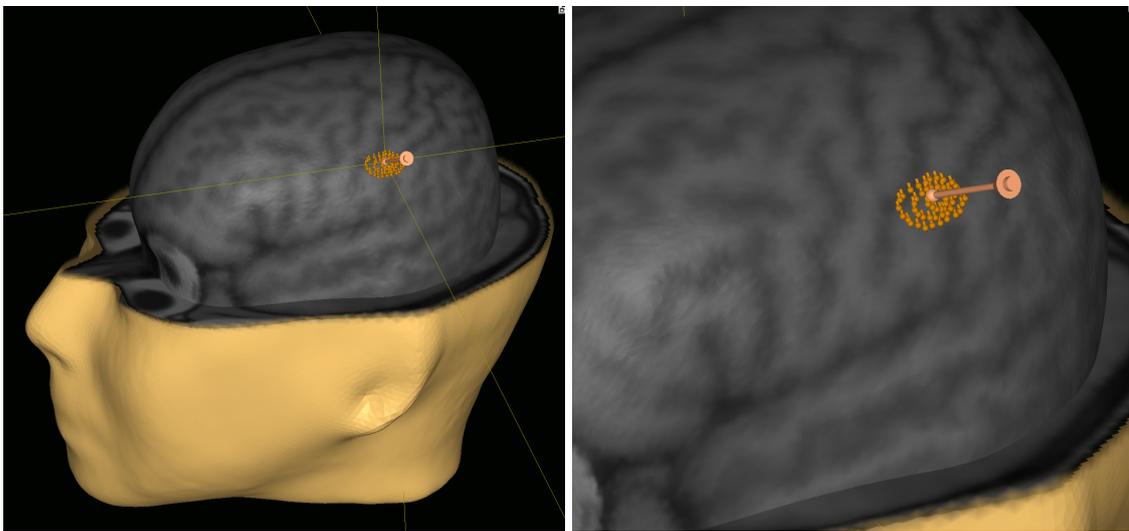


Figure 6.10: A stimulated area surrounding a “standard” stimulation target. Each point in the circular shapes represent a stimulation pulse delivered.

## 6.6.2 Panda task functions

This paragraph is dedicated to the description of the functions that define the MIOS tasks and the skills used to move the robot. In this project, each function allows to start a single task where only one skill and the related parameters are defined. The specific values listed below are obtained thanks to many tries and tests, in order to obtain an Assistant with a good usability that also satisfies the functional requirements.

### 6.6.2.1 *MoveToPoseJoint* skill

The first function called during the code execution uses the *MoveToPoseJoint* skill. It allows the robot to move in a defined position specifying the joints angles vector, 7 elements where the first corresponds to the flange and the last to the base junction. Other parameters for this skill are: the speed, set to 0.1 m/s; the acceleration, set to 0.08  $m/s^2$  and the control mode, set to index 3 which corresponds to the Joint Velocity Control. This skill is used mainly during the calibration task, when no references between the robot and

the patient's head are known yet. In fact, the locations of the six points used for the SVD based method are expressed as joints angles vectors and saved as variables in the code. Finally, the joint pose control is also used to reset the joint orientations when the robot returns to the ready position.

### 6.6.2.2 *MoveToPoseCart* skill

The second function used to move the robot in this version of the Assistant uses the *MoveToPoseCart* skill. Similarly to the previous one, it allows the robotic arm to move to a precise location, specifying the transformation matrix from its base frame to the end-effector coordinate system. It is usually passed as a 16-elements vector, ordered as follows:  $[r_1, r_4, r_7, 0, r_2, r_5, r_8, 0, r_3, r_6, r_9, 0, p_x, p_y, p_z, 1]$ .

For this skill, some of the parameters depend on when the movement function is called. Since it is used to move the coil around the head and to reach the targets, one can distinguish two pairs of speed-acceleration variables: the first, for movements on the virtual sphere, with speed 0.35 m/s and acceleration  $0.05 \text{ m/s}^2$ ; the second, for the movements toward the targets on the scalp, with speed equal to 0.052 m/s and acceleration  $0.003 \text{ m/s}^2$ . The control mode for this skill is index 2, or Cartesian Velocity Control. Finally, for this skill there is also the possibility to declare the joints stiffness as a 6-elements vector, ordered from the robot base to the flange: in particular, for this project, it is set to  $[1500, 1500, 1500, 250, 250, 250]$  N/m.

## 6.6.3 Safety

Based on the defined functional requirements, the assistant should perform the following safety: the contact detection method, when the coil pushes against the patient's head, and the safety task stop if the markers are no longer visible by the tracking unit. Both these routines should be executed in background, while the main control is possessed by the movement task in execution. In case the safety conditions are no longer satisfied, the task is stopped and, if necessary, the operator must intervene.

### 6.6.3.1 Contact detection method

The Panda robot has an internally implemented force measurement method based on the estimation of the external torques, explained in detail in [30]-[31], by means of torque sensors located in the joints and a robot model. The *MoveToPoseCart* offers the possibility to automatically stop a task in execution whenever the external forces sensed are greater than a threshold specified as a parameter. Nevertheless, it has already been mentioned that the hanging pipes connected to the rear part of the coil influence and sometimes limit the coil movements. Pipe forces are sensed by the robot as variations of the torques and forces applied to the junctions. This means that the tubes also influence the external forces sensed, leading to estimations that are in the same order of amplitude of a suitable threshold for a head-coil contact detection; all this makes the contact detection method via a fixed threshold not appropriate. Alternatively, the main idea is to perform a numerical time derivative on the forces data stored, in order to detect a possible contact. To obtain the data related to the external forces estimated by the robotic arm on each joint, a MIOS method is used: subscribing a specific telemetry data to a subscriber list, it is possible to receive the information needed, sampled at 1 kHz, via UDP to the IP and port specified [15]. When a movement towards the head begins, the contact detection thread is started:

the telemetry subscription is performed, and the program stores the collected data into a register sized to hold 40 samples, which means a time window of 40 ms. Only three of the seven elements composing the force vector received from MIOS are kept: the ones related to the flange (the last joint), to the third and the fourth to last joints. The data concerning the second to last junction are discarded, since the experimental data showed that they are the most influenced by the tubes action. The data related to the first 50 ms of the movement are ignored, to avoid the possibility that the forces variations sensed at the beginning of the movement are detected as a contact. Then the real-time numerical derivative is calculated as follows:

$$dF_{ext,i,0} = \frac{F_{ext,i,0} - F_{ext,i,39}}{\Delta t} \quad (6.65)$$

where  $dF_{ext,i,0}$  is the current value of the force derivative,  $F_{ext,i,0}$  is the current value of the force, while  $F_{ext,i,39}$  corresponds to the oldest sample in the register. Finally,  $\Delta t$  is the time gap between the current sample and the oldest one, and in this case, it is equal to 0.04 s. The current derivative is then compared to a threshold value, set to 50 N/s, to detect a contact between the coil and the head. The Robotic Assistant interprets the contact detection as a flag, meaning that the head has been reached: the coil movement is stopped and the program checks its position with respect to the target to assess the accuracy, following the navigation algorithms explained above. As in the previous cases, the programming phase of this method required many tries and parameters adjustment, in order to point out suitable values to correctly carry out the task.

### 6.6.3.2 NBS Status watchdog

This thread is used to satisfy the functional requirements related to markers visibility in the workspace. During the normal operation of the NBS System, both the head and the coil markers should be visible by the tracking unit, especially during the stimulation sequences. When one of the two or both are no longer visible, the possibilities are various: the camera could have been moved and the markers are no more in its visibility range; the patient could be not sitting on the chair; the robot is far from the visibility space; finally, the markers are covered, which means that someone or something is placed between the tasking unit and the markers. For safety reasons, if one of them is no longer visible, then any robot task (i.e., movement) in execution should be stopped and the operator should intervene.

The information related to the markers could be obtained through the NBS Interface: a simple status request is sent, and the received flags related to the markers are checked. In case of a zero-flag for the coil or the head markers, the task is stopped, and the operator should restart the Robotic Assistant. The same measure is taken if, during the task, the Session is closed, the patient's Registration is deleted, or the Exam is closed.

## 6.7 Subsystems integration

In this paragraph the structure of the implemented Robotic Assistant is described, explaining how the different parts work together. The finite state machine diagrams of the Robotic Assistant are represented in Figure 6.11 and 6.12.

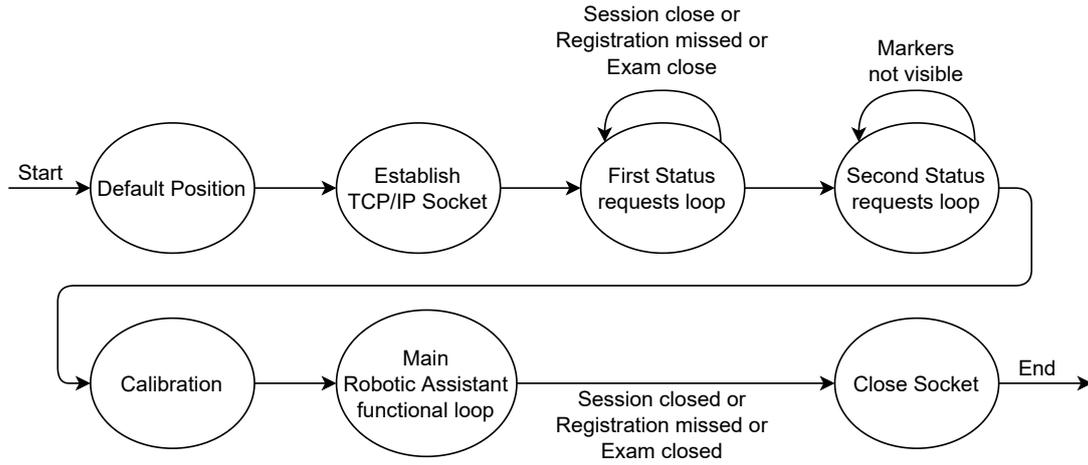


Figure 6.11: The main Robotic Assistant finite state machine diagram.

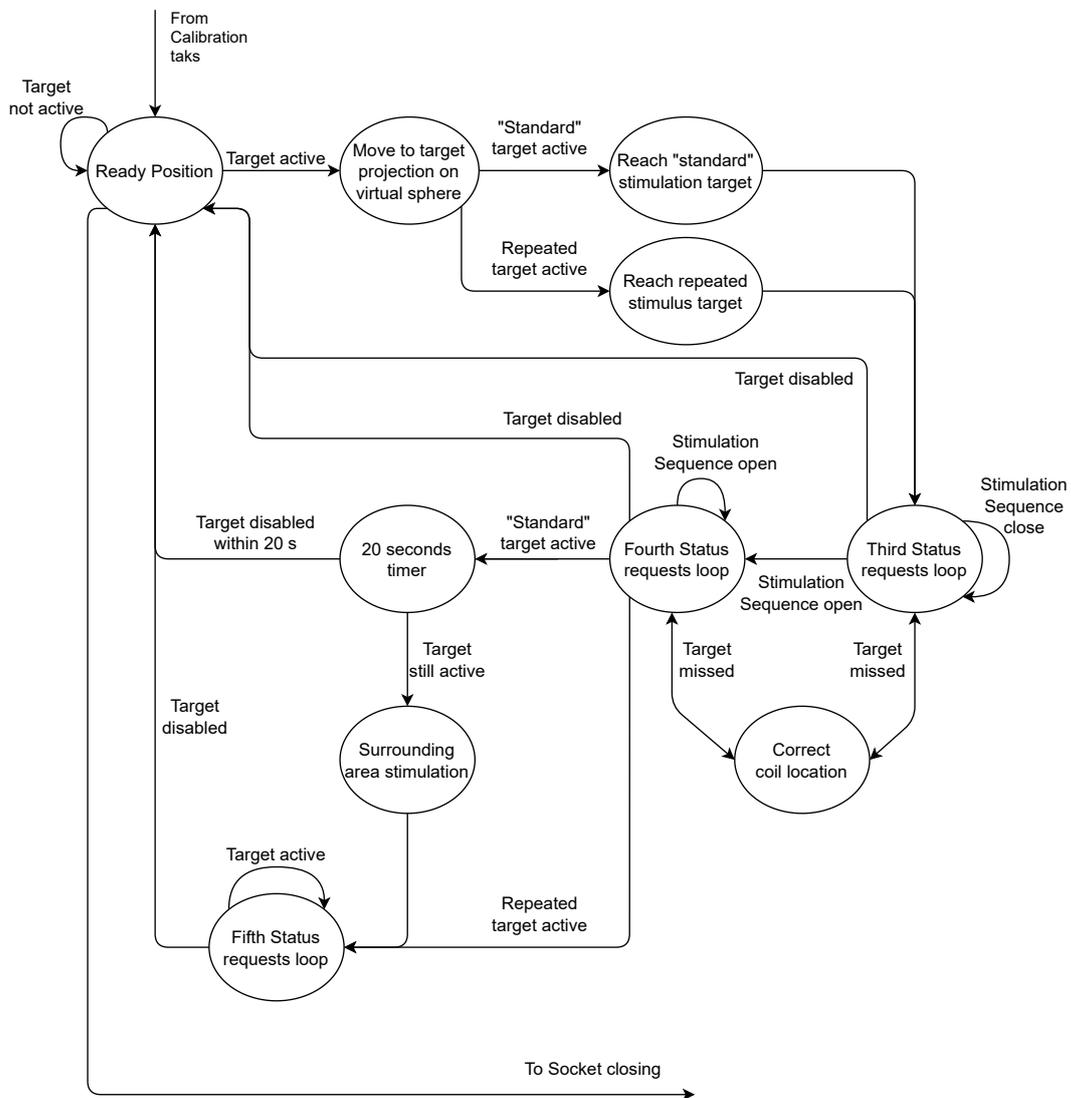


Figure 6.12: The main functional loop finite state machine diagram.

### **6.7.1 Pre-calibration tasks**

Once the Robotic Assistant software is executed, the robotic arm moves to a default position, considered safe for the patient and the operator, using the *MoveToJointPose* skill and the related joints angles vector. Once the robot is in position, the TCP/IP socket is established, to enable the communication with the NBS Interface, and a first request of the NBS System status is performed. At this point, the Robotic Assistant waits until the following operations are completed: the Session is opened; the patient's landmarks are registered, and a new Exam is started. During this waiting time, the robot does not move, and the Assistant only performs status requests to the NBS Interface. When the three status flags are set, the Assistant proceeds to the markers visibility check. The default position of the coil should allow the tracking unit to recognize its markers, while the head one is visible when the patient is correctly sitting on the chair and ready for the TMS treatment. Also in this case, the robot is not moving and the Assistant performs requests until both the markers are visible.

### **6.7.2 Calibration tasks**

At this point, the calibration task begins and the robot moves the coil through the six defined points surrounding the patient's head. During the movements of this phase, the status watchdog thread ensures that the markers remain in the tracking volume. Once the robot reaches the last point, the transformation matrix from the robot base frame to the MRI coordinate system is calculated and saved in a .csv file. Then the coil is moved to a so-called ready position, behind the head, a convenient spot to start all the movements to reach the targets. The location of the ready position is situated on the virtual sphere, calculated after the calibration, and the *MoveToPoseCart* skill is used.

### **6.7.3 Sphere navigation and positioning task**

Now the Robotic Assistant is ready to move the coil to the first target: in fact, it waits until a "standard" stimulation target or a repeated stimulus target is set as active by the operator on the NBS Software. As soon as a target is set, the robotic arm moves the coil toward the target projection on the virtual sphere, following the navigation pattern obtained through the dedicated function. Once arrived at the destination, the Robotic Assistant moves the coil in correspondence with the target location, adopting one of the two algorithms described, depending on the target nature ("standard" or repeated). When the coil is correctly positioned, the Assistant waits until a stimulation Sequence is opened, or the target is deselected. During this time, the status requests are used also to check if the coil is still well-located; in case there are no more sufficient conditions to consider the coil in the correct spot, a repositioning procedure starts, following the same algorithm to reach the targets.

### **6.7.4 Stimulation tasks**

When the coil location is correct and a new stimulation sequence is opened, the Robotic Assistant still monitors that the status flags remain set (i.e., Session open, Registration done, Exam open and markers visible), as well as the active target flag; moreover, in case of necessity, it performs the repositioning of the coil if it is no longer in the correct location. When the stimulation session is closed, the coil remains in the same target

position on the scalp in case a new stimulation Sequence is opened. If this does not happen, there are two possibilities for the coil: the target is deselected and the coil can be moved back to the virtual sphere and then returned to the ready position, behind the head; or it can be moved to perform an area stimulation sequence if the selected target is of the “standard” kind. To identify the latter case, the following rule has been adopted: if the “standard” stimulation target remains set active for more than 20 seconds after the Sequence closing, than the area stimulation procedure begins; on the other hand, if the target is deselected within the 20 seconds time window, the coil is moved back to the virtual sphere and then to the ready position. In case of a repeated stimulus target, it is not possible to stimulate the surrounding area, so the coil is moved directly to the sphere and returned to the ready position as soon as the target is deselected, without time limits. It is useful to remark that all the movements around the head are performed following the sphere navigation patterns.

### 6.7.5 Post-stimulation tasks

Once the coil is returned to the ready position, the joint angles are reset to the initial conditions, after the calibration process. This allows the robot to reach all the targets, avoiding situations where a joint reaches its maximum angle limit; in addition, it also improves the repeatability of the reaching tasks for the same targets, since they are always reached following the same patterns, starting from the same initial conditions. After the joint angles reset, the Robotic Assistant waits until there are no active targets set on the NBS Software: this condition is useful to allow the Assistant to return to the initial state, waiting for a new target. As a final note, during each movement of the robot the status watchdog thread is active, while the contact detection thread is used only during the movements from the virtual sphere toward the scalp. More details about the contact detections and the general collision detections will be reported in chapter 7.

## 6.8 Navigation with movements compensation

The implementation of the Robotic Assistant described in the previous paragraphs fixes the position of the dummy head in a steady position with respect to the robot base frame. Even though this is a good approximation of the real employment conditions, where the patient lays on a reclining seat and the head is supported by a specific frame, it is an excessively strict requirement. Indeed, the patient should be free to move his head, also during the treatment, to be always in a comfortable position. In order to allow the patient’s head movements, the transformation matrix from the robot base frame to the MRI coordinate system should not be fixed, quite the opposite it should be periodically updated. In this way, the transformation matrix to guide the robot to the target could be corrected with the current position of the destination point. In order to do so, the Assistant workflow has been changed: the initial calibration task is not needed, and the robot-MRI transformation matrices are calculated fusing together two data: the current end-effector position  $H_{ef}^{robot}$ , obtained through MIOS, and the current position of the coil  $H_{MRI}^{coil}$ , requested to the NBS Interface. Indeed, the transformation matrix from the robot base frame to the MRI coordinate system can be computed as follows:

$$H_{MRI}^{robot} = H_{ef}^{robot} H_{coil}^{ef} H_{coil}^{MRI-1} = H_{ef}^{robot} H_{coil}^{ef} H_{MRI}^{coil} \quad (6.66)$$

where the inverse matrix  $H_{MRI}^{coil}$  is calculated as:

$$H_{coil}^{MRI^{-1}} = H_{MRI}^{coil} = \begin{bmatrix} R_{coil}^{MRI^T} & -R_{coil}^{MRI^T} P_{coil}^{MRI} \\ 0 & 1 \end{bmatrix} \quad (6.67)$$

and

$$H_{coil}^{ef} = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6.68)$$

The matrix  $H_{MRI}^{coil}$  is then recomputed every time it has to be used in the code.

In addition, a new functionality has been added to the Assistant: when a new target is set active and the coil is moved to its projection on the virtual sphere, the robot is programmed to follow the location variations of the target itself, before the positioning task begins. Only when the head remains steady for a fixed time window of 3 seconds from the last movement, the coil is enabled to approach the target. However, during the positioning attempts, the tracking algorithm of the target is stopped and the head is supposed to remain steady; when the last attempt is performed and the current position of the coil is too far from the real position of the target, the target tracking is again activated and the coil repositioned.

# Chapter 7

## Experiments

In this chapter the experiments carried out to assess the Robotic Assistant performance are described, and the experimental data recorded during the sessions are presented. The main objective of the experiments is to investigate i) the accuracy of coil positioning and ii) the robustness across different sessions (inter-session performance/variability).

### 7.1 Experiments description

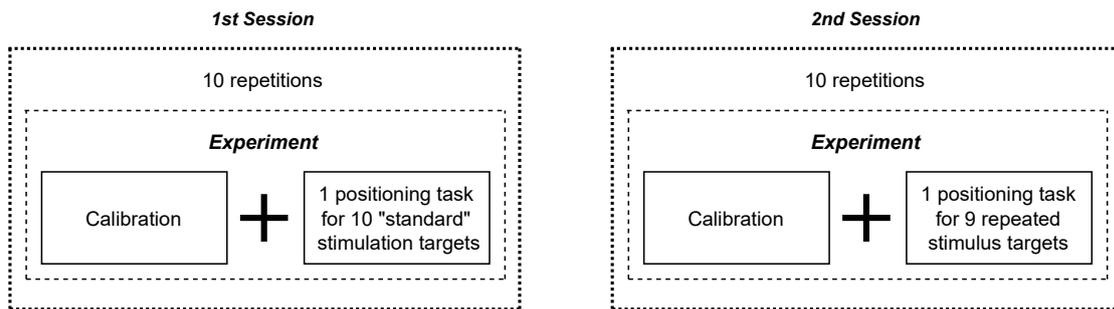


Figure 7.1: The performed experimental sessions diagrams.

The experimental protocol for this project, as shown in Figure 7.1, consists of two main experimental sessions to be carried out: i) the first considers only "standard" stimulation targets, while the ii) second uses only repeated stimulus targets. It is useful to remark that a "standard" stimulation target describes a specified location on the patient's scalp, without any information regarding the coil orientation; on the other hand, a repeated stimulus target provides information concerning the final coil position and orientation to correctly stimulate the target. During the first session, 10 different targets are adopted (Figure 7.2), in order to perform 10 experiments: each of them consists of 10 positioning tasks to simulate the "reach and stimulate" procedure once per target. For both the sessions, the targets are chosen to reproduce the real positions of the investigation areas for the functional motor mapping; in addition, for the first session, two targets are set on the occipital region to further assess the Assistant performances. In Figure 7.3 are illustrated the 9 repeated stimulus targets selected for the second experimental session.

At the beginning of all the experiments the Robotic Assistant is restarted, and a new calibration process is executed; when the robotic arm is in the "ready position", the first target is activated on the NBS Software, and the positioning task begins. The coil is considered in the correct position when the conditions included in the navigation algorithms

are satisfied: the error is less than 4.5 mm for “standard” stimulation targets, or for the repeated stimulus target the error is less than 2 mm and the coil is properly oriented (NBS “Aim” flag is active).

After the coil has reached its final location, the transformation matrix of the coil and the transformation matrix of the target position on the scalp, both with respect to the MRI frame, are recorded. When this information is collected the target is deselected, and the robotic arm returns to the “ready position”. At this point, the second reaching task can be started, activating a new target and the procedure is repeated. Across experiments, the targets are activated always in the same order, based on their enumeration. When the coil has reached all the targets prepared, the experiment ends. The experimental session ends when all intra-session experiments have been completed.

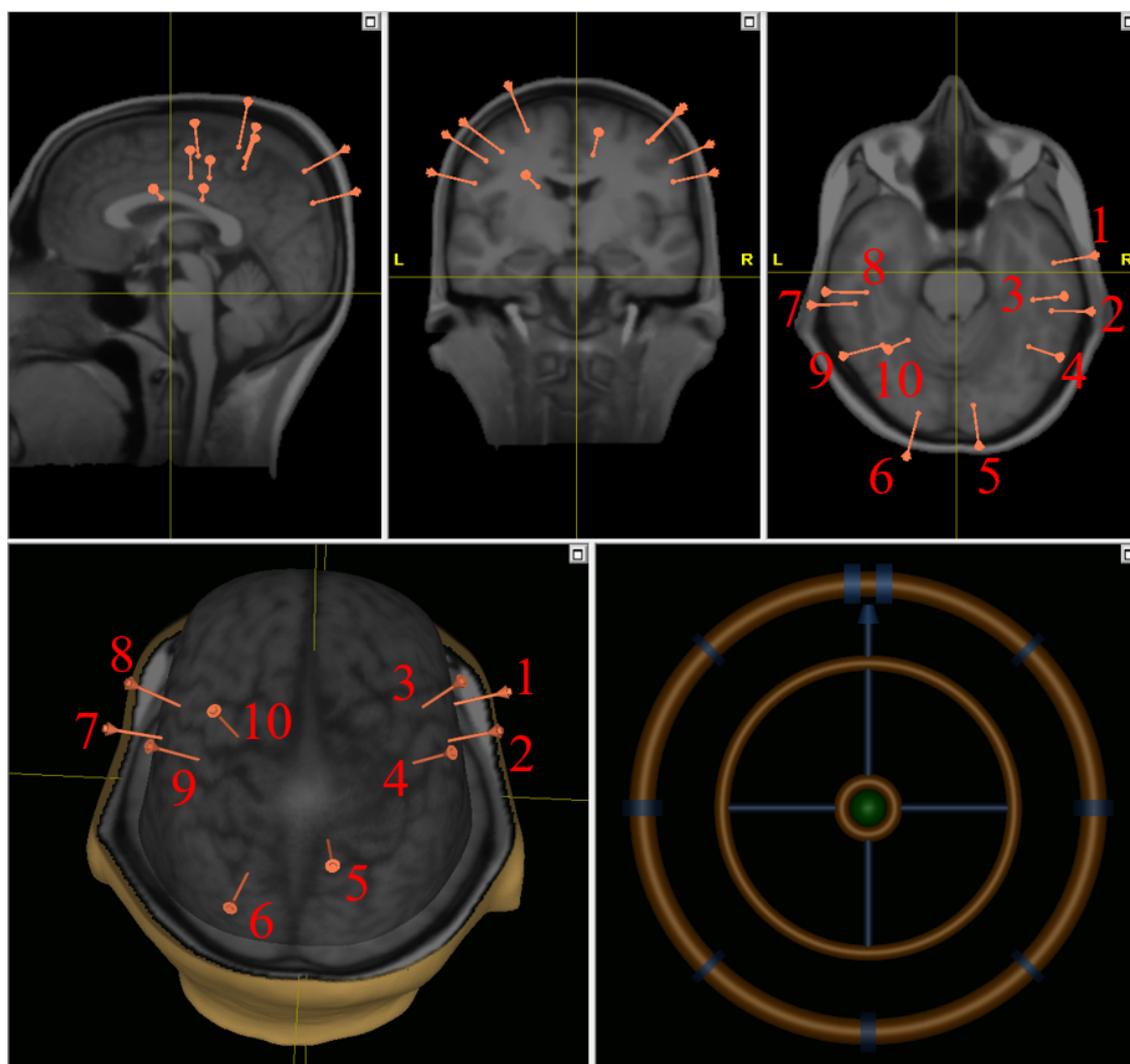


Figure 7.2: First Session; the 10 “standard” stimulation targets selected

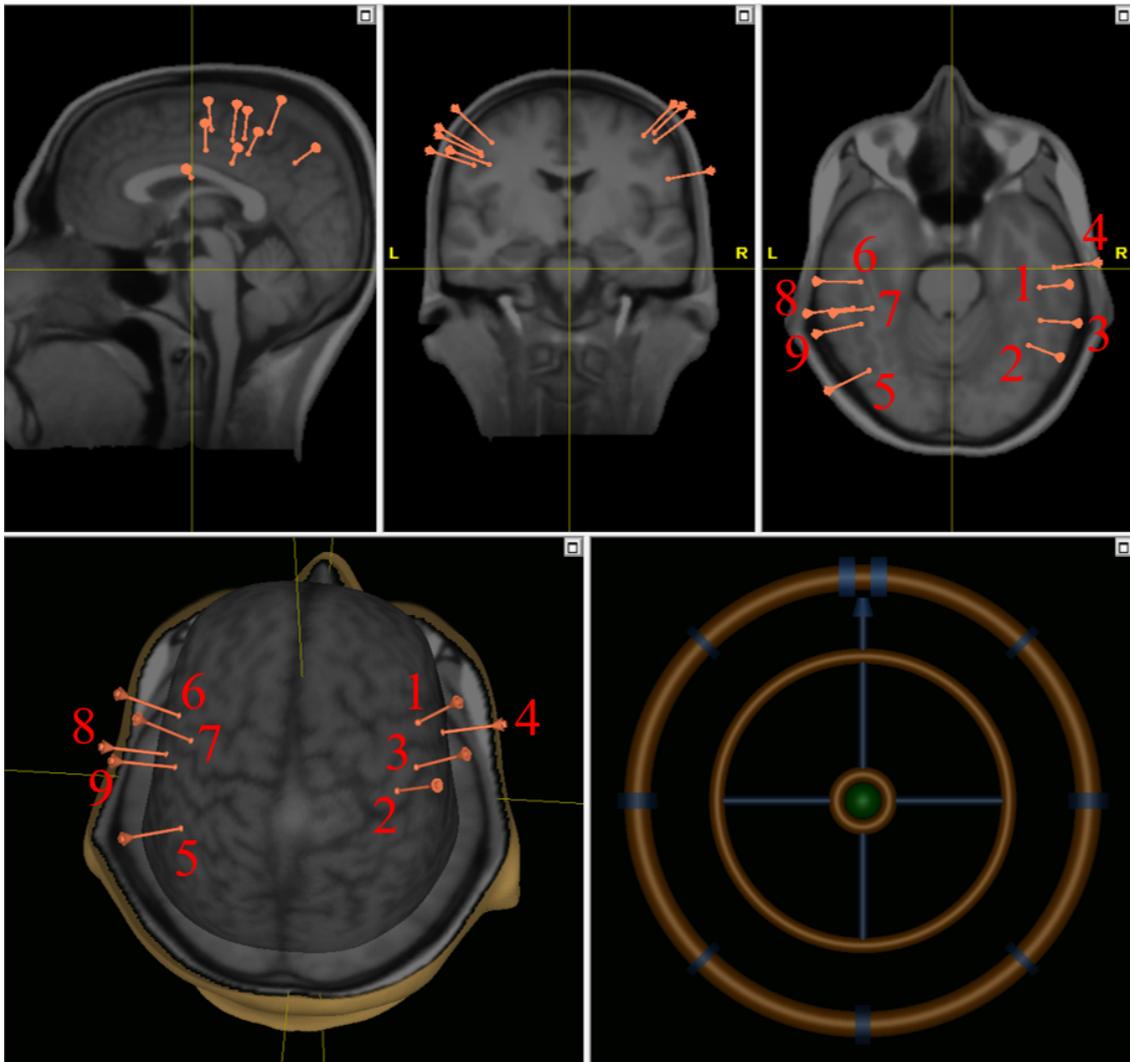


Figure 7.3: Second Session; the 9 repeated stimulus targets selected

### 7.1.1 Experiments with movement compensation

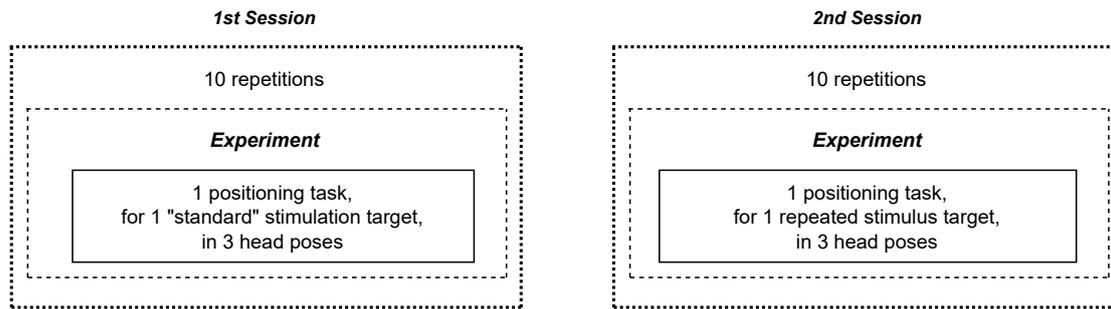


Figure 7.4: The performed experimental sessions diagrams for the movement compensation approach.

To test the second version realized for the Robotic Assistant, where the transformation matrix from the robot base frame to the MRI coordinate system is periodically recalculated, two new experiment sessions are defined (Figure 7.4). To check the accuracy of the positioning tasks and the ability of the robot to compensate for head movement, three different positions of the dummy model are a priori decided. They are obtained through a rotation of the head about its vertical axis, starting from i) an initial pose henceforth indicated with 0 degrees; ii) the second is reached with a counterclockwise rotation of +20 degrees; finally, iii) the last position corresponds to a -10 degrees rotation with respect to the initial one, which is reached from the second through a clockwise rotation of -30 degrees. The three head poses are represented in figure 7.5.

Similarly to the previously described experimental protocol, the first session is dedicated to the “standard” targets, while the second to the repeated stimulus; each session is composed of 10 repetitions of the same experiment. One experiment consists of three positioning tasks for a fixed stimulation target, but with different head positions. When an experiment is started, the coil reaches the target with 0 degrees of head rotation, and the data related to its position are collected; then the head is rotated to its second pose, and the compensation algorithm guides the coil to the new target location. When it reaches the correct position on the scalp, a new data registration is performed and the head is moved to the last pose. The experiment ends when even the final position compensation is completed. The “standard” target chosen for the first session corresponds to number 8 of Figure 7.2, while the repeated stimulus adopted for the second session is number 8 of Figure 7.3.

## 7.2 Experimental data

To begin the introduction of the experimental data, the root mean squared error (RMSEs) between the calibration points and the correspondent coil locations, acquired during the calibration task, are calculated using formula (6.23), as seen in chapter 6.5. They are presented in Figure 7.6 and, computing the norm for each experiment, one can easily obtain a mean calibration error of 1.5 mm for the first session, and 1.3 mm for the second.

The positioning errors for the two sessions are calculated as the Euclidean norm (or 2-norm) of the difference vectors between the final coil position and the “reference” target location. Figures 7.7, 7.8 and 7.9 concern the first experimental session, while Figures



Figure 7.5: The three head poses for the head compensation approach experiments; 0 degrees (left), +20 degrees (middle), -10 degrees (right).

7.10, 7.11 and 7.12 are related to the second one. In Figure 7.7 the positioning errors related to the first attempts are reported in form of box plot; each box is associated with the sequential number of the target and expresses the error related to the 10 positioning tasks (in light blue). The outliers are indicated by the blue crosses. The orange segmented line represents the mean error values calculated on the 10 repetitions for each single target. Figure 7.8 reports the same kind of information as the previous one, but refers to the final position of the coil, and the difference can be significantly noticed from the error scale. Finally, error distribution is plotted in the histogram in Figure 7.9, considering the total error values calculated from the first session.

Regarding the second session, the positioning task accuracy related to the coil-target location errors is presented in Figure 7.10, considering that 9 targets are available for these experiments. To calculate the error in terms of rotations, the Euler angles are obtained from the rotation matrices of the final coil position and of the target, following the convention  $ZYX$  of the `rotm2eul()` Matlab function; then it has been calculated the differences  $\Delta\gamma$ ,  $\Delta\beta$  and  $\Delta\alpha$ , referred respectively to axis  $z$ ,  $y$  and  $x$ , between the targets Euler angles and the coil ones as follows:

$$\Delta\gamma = \gamma_{target}^{MRI} - \gamma_{coil}^{MRI} \quad (7.1)$$

$$\Delta\beta = \beta_{target}^{MRI} - \beta_{coil}^{MRI} \quad (7.2)$$

$$\Delta\alpha = \alpha_{target}^{MRI} - \alpha_{coil}^{MRI} \quad (7.3)$$

The differences are then collected in 3 box plots (Figure 7.11), one per axis, distinguishing the results obtained for each target. Every box represents the data related to the 10 experiments. Finally, similarly to the previous case, the location errors distribution is shown in a histogram plot (Figure 7.12).

### 7.2.1 Results with movement compensation

The same data analysis has been applied to the experimental results from the sessions with the head movements compensation method. The positioning task accuracy for the

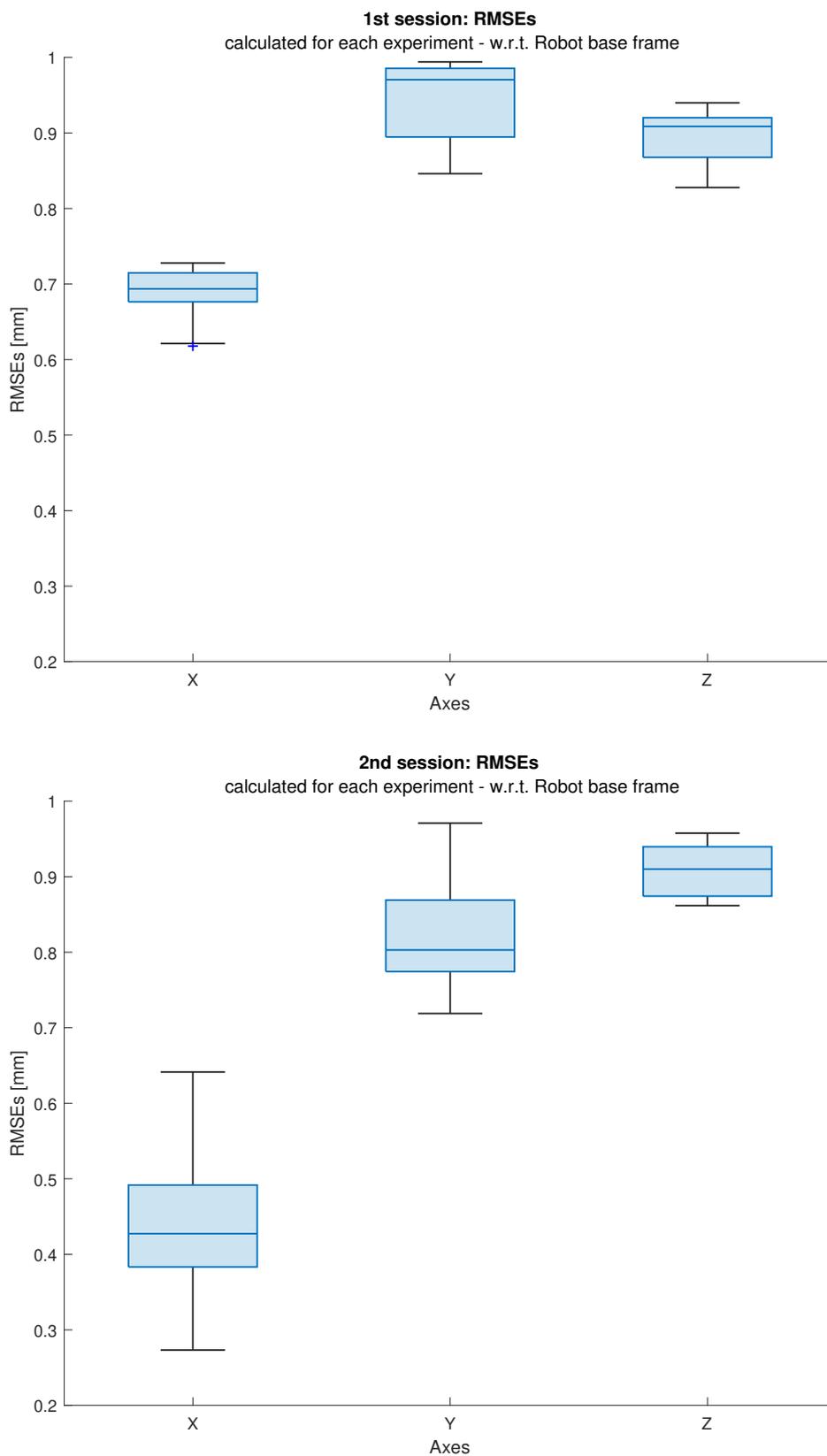


Figure 7.6: Root mean squared errors for First Session (top) and Second Session (bottom).

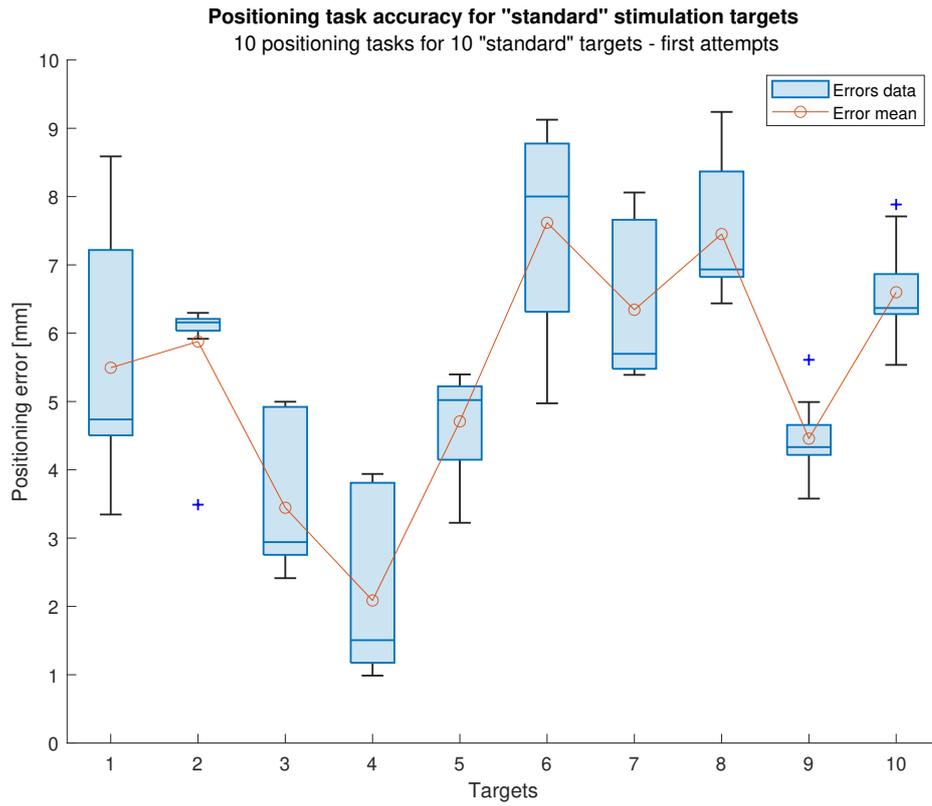


Figure 7.7: First Session; positioning task accuracy for first attempts.

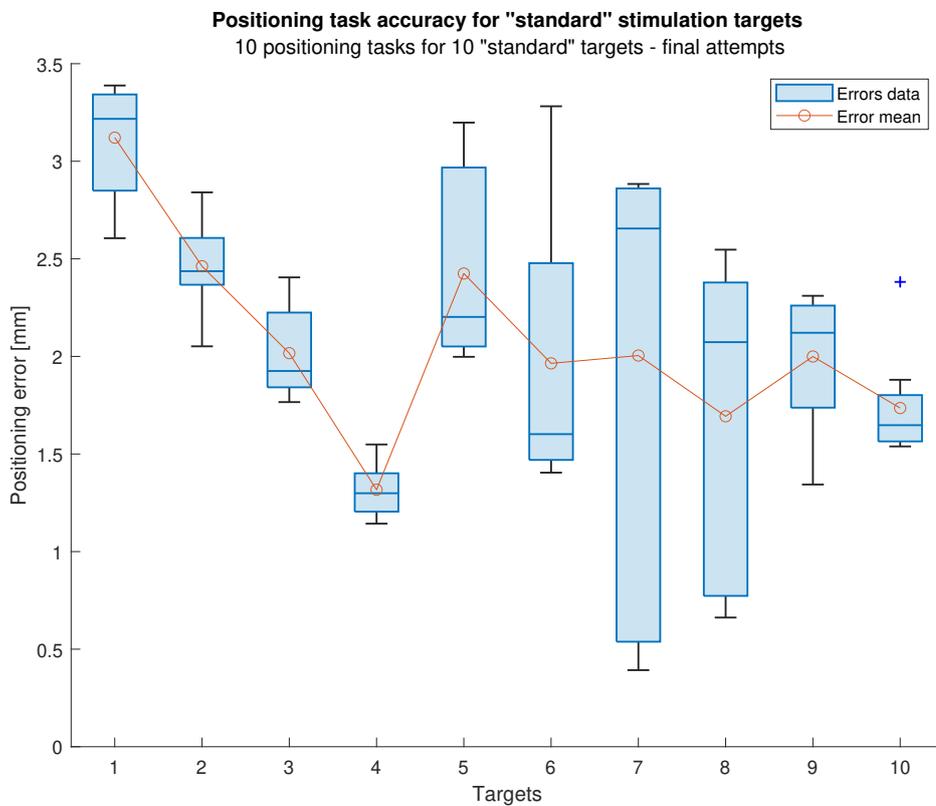


Figure 7.8: First Session; positioning task accuracy for second attempts.

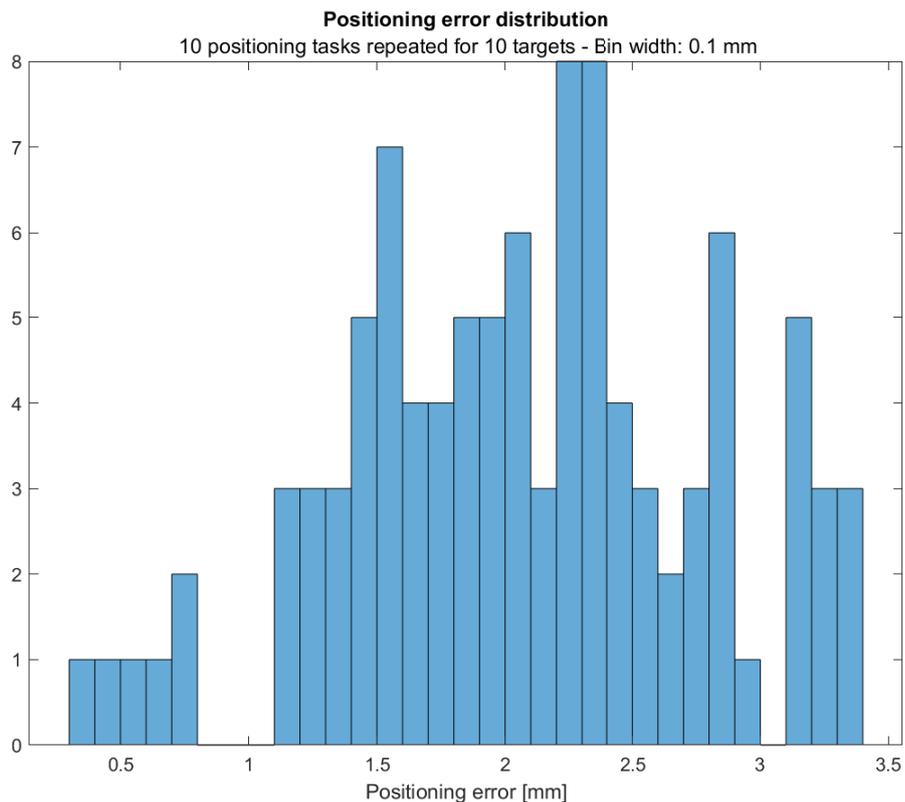


Figure 7.9: First Session; error distribution.

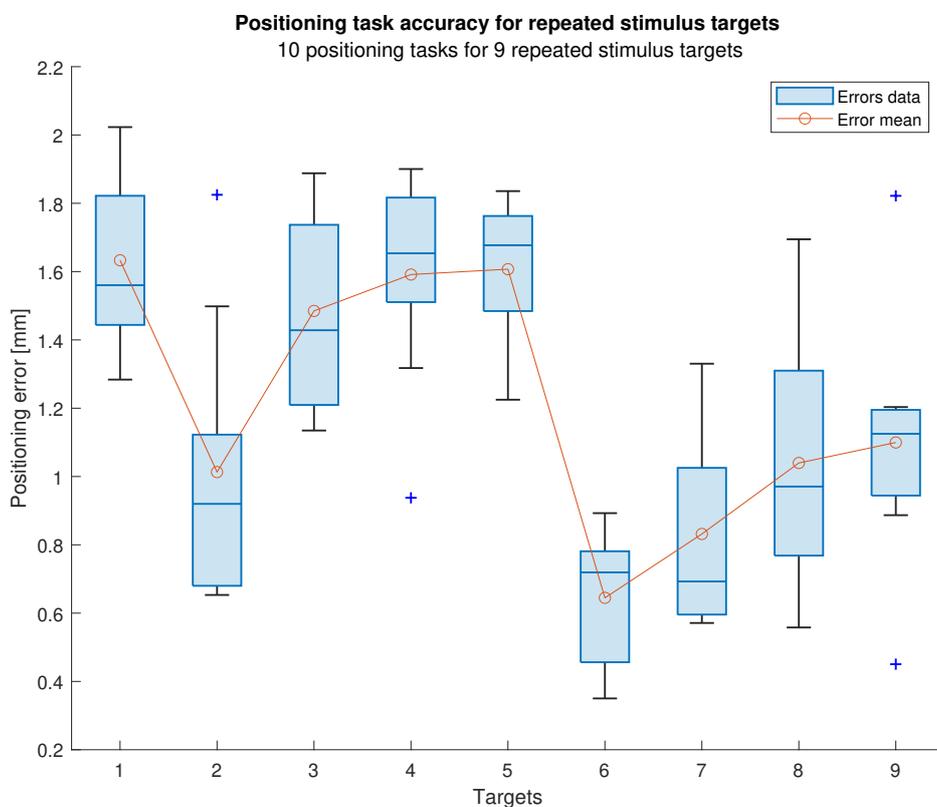


Figure 7.10: Second Session; positioning task accuracy.

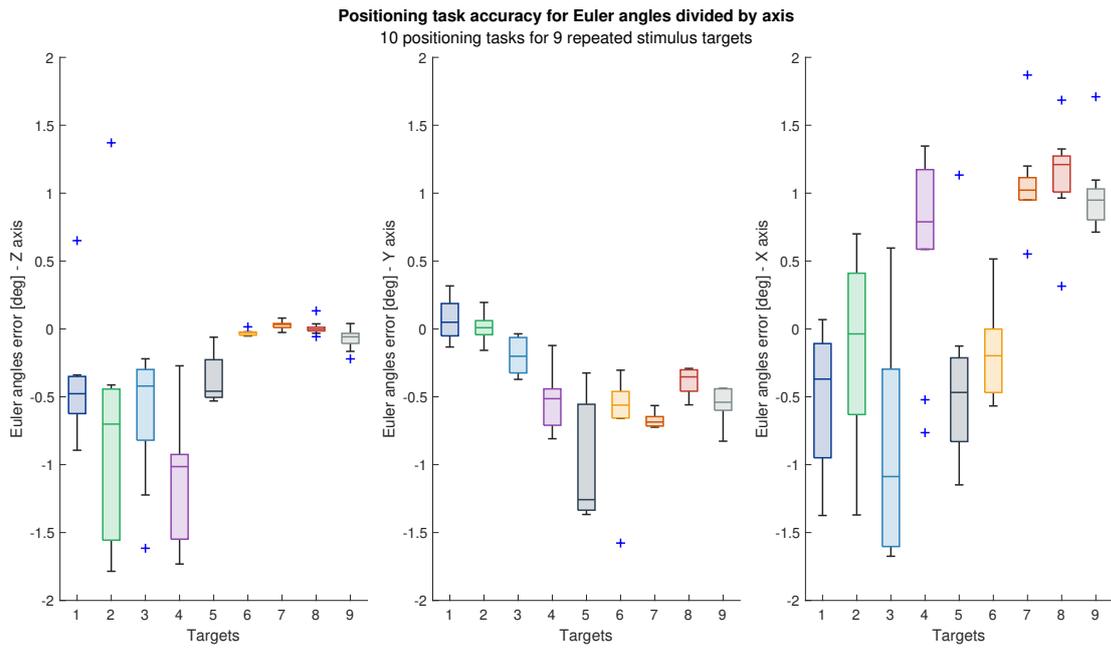


Figure 7.11: Second Session; positioning task accuracy for Euler angles.

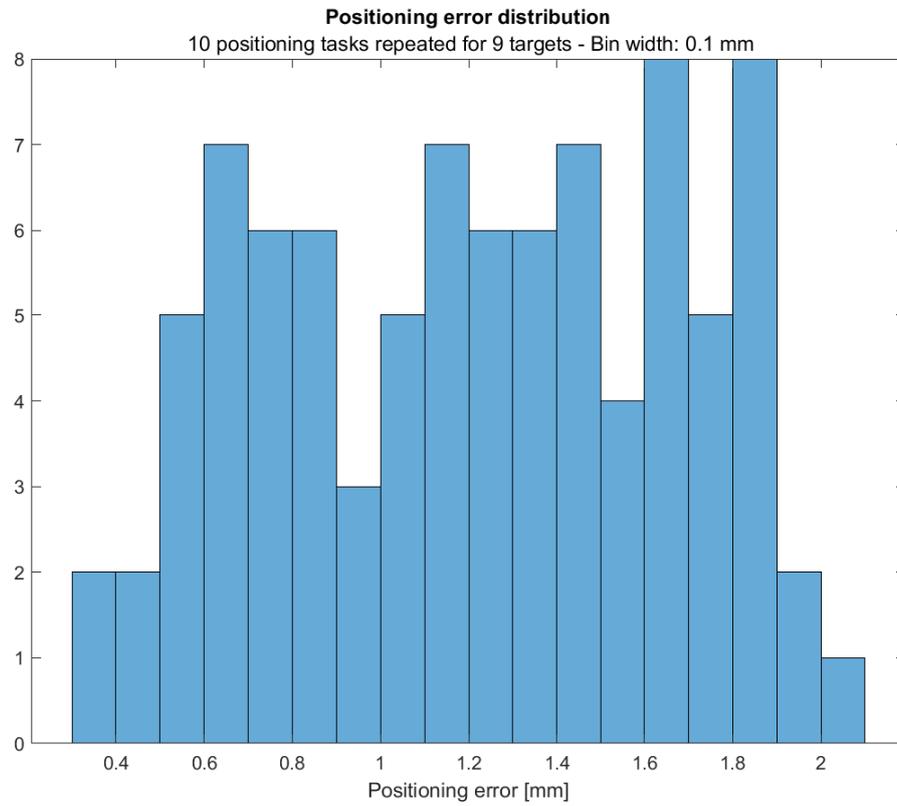


Figure 7.12: Second Session; error distribution.

first session (with “standard” target) is shown in Figure 7.13 and 7.14, where the former is dedicated to the first reaching attempts, and the second to the final coil position after the second attempt. In these plots, each box is referred to a different head pose and collects the information of the 10 repetitions. In Figure 7.15 is shown the error distribution of the 30 repetitions belonging to the first session.

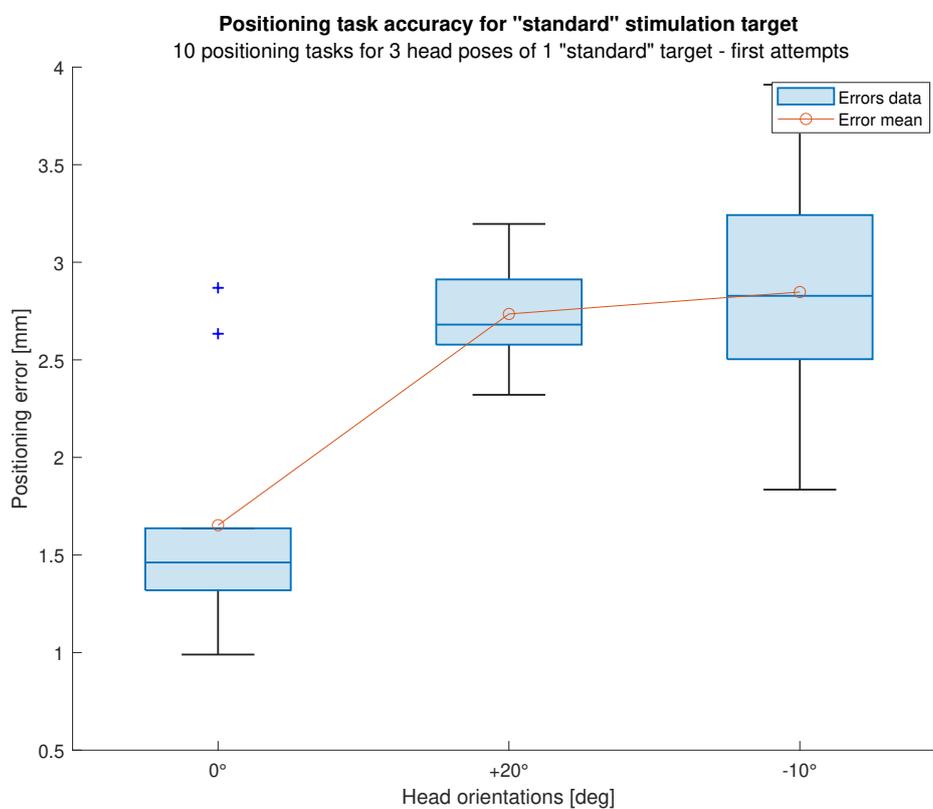


Figure 7.13: First Session; positioning task accuracy for first attempts with movement compensation approach.

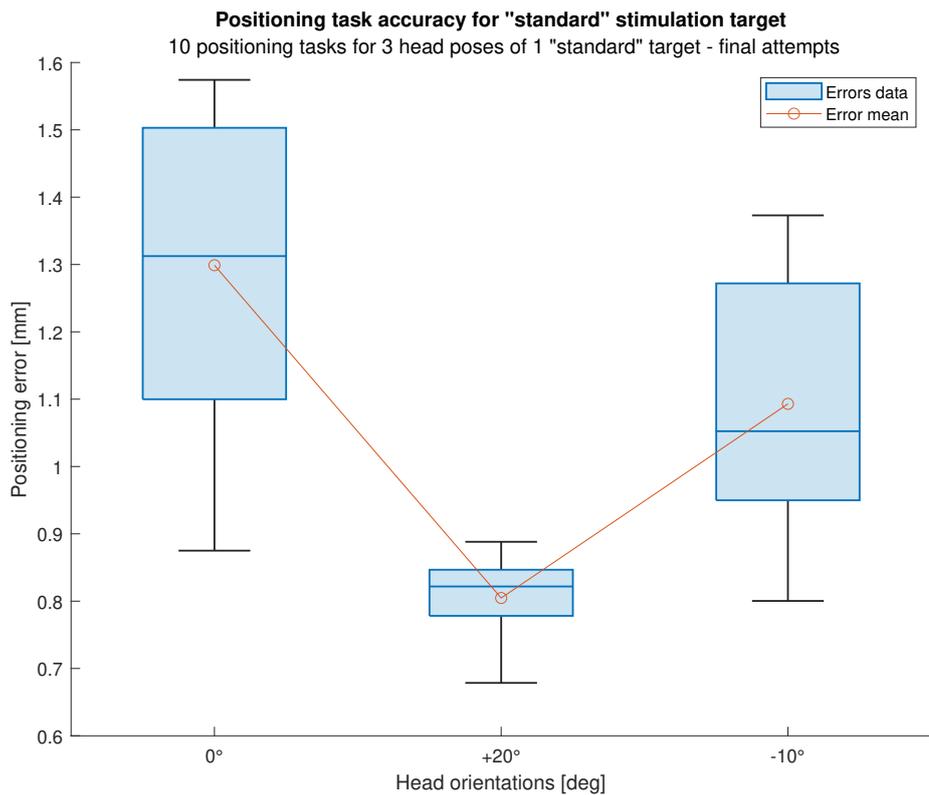


Figure 7.14: First Session; positioning task accuracy for second attempts with movement compensation approach.

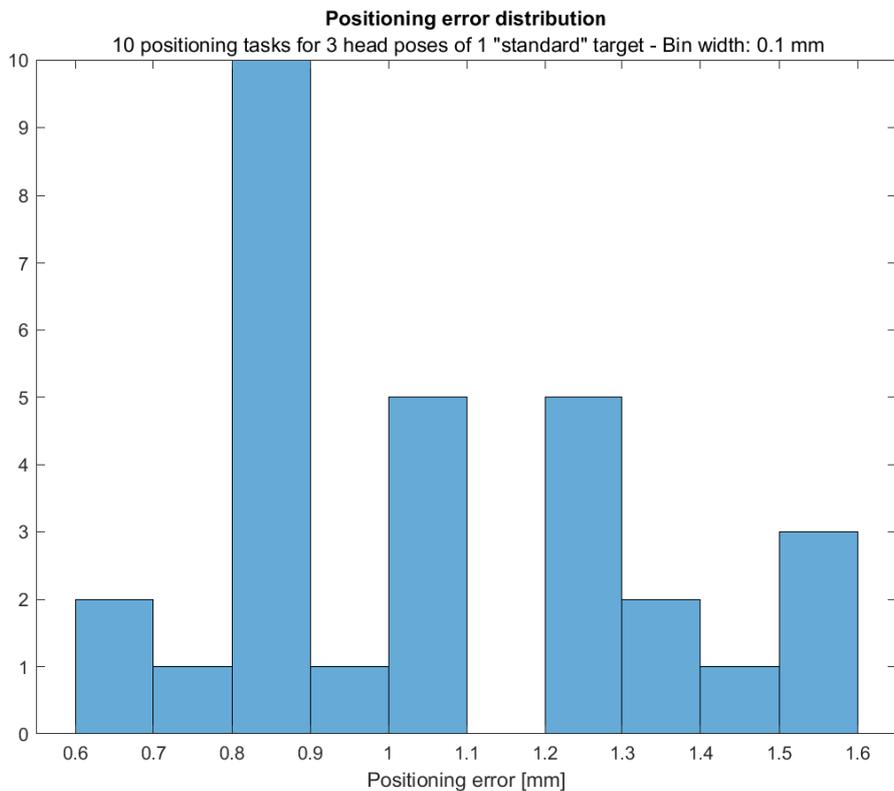


Figure 7.15: First Session; error distribution with movement compensation approach.

Considering the repeated stimulus targets, the positioning task accuracy results are collected in figure 7.16, all referred to the final coil positions. The angle mismatches between the target rotation matrix and the coil one is calculated using equations (7.1), (7.2) and (7.3), as explained in the previous paragraph; indeed, the plots in figure 7.17 are referred to Euler angles mismatch with respect to z, y and x axes respectively. Finally, the positioning error distribution for the second session is plotted in figure 7.18.

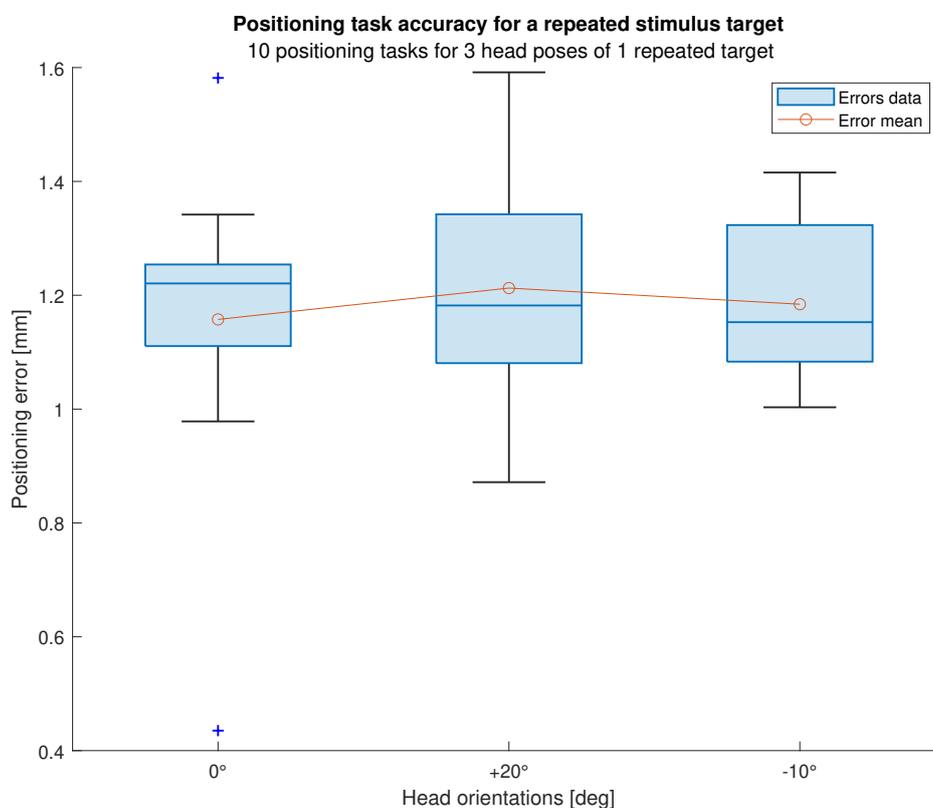


Figure 7.16: Second Session; positioning task accuracy with movement compensation approach.

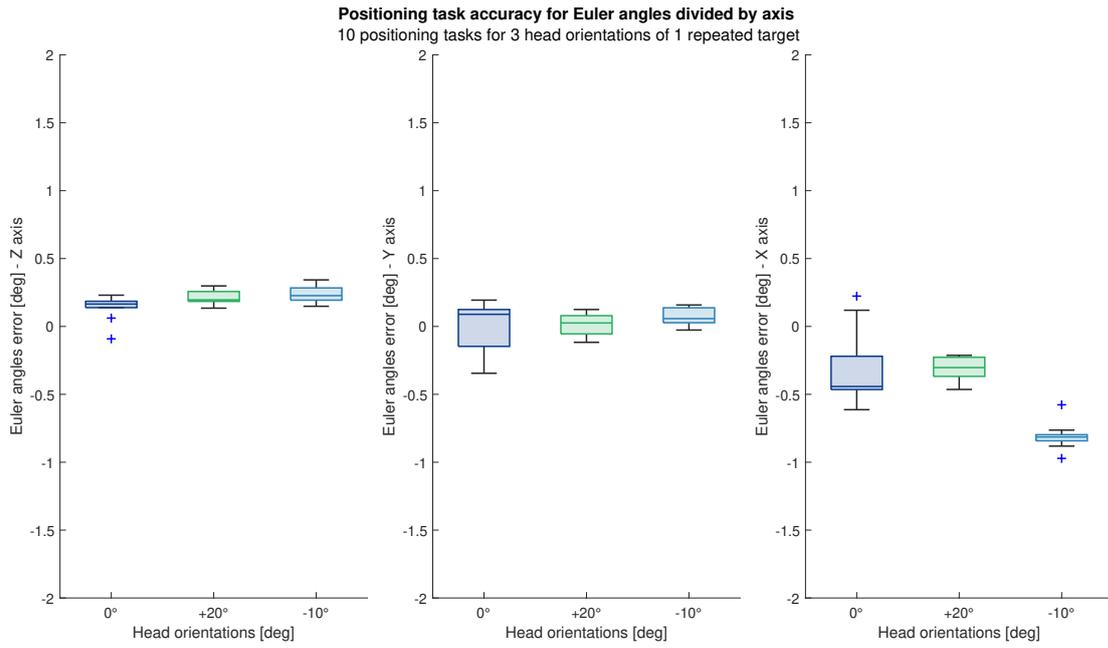


Figure 7.17: Second Session; positioning task accuracy for Euler angles with movement compensation approach.

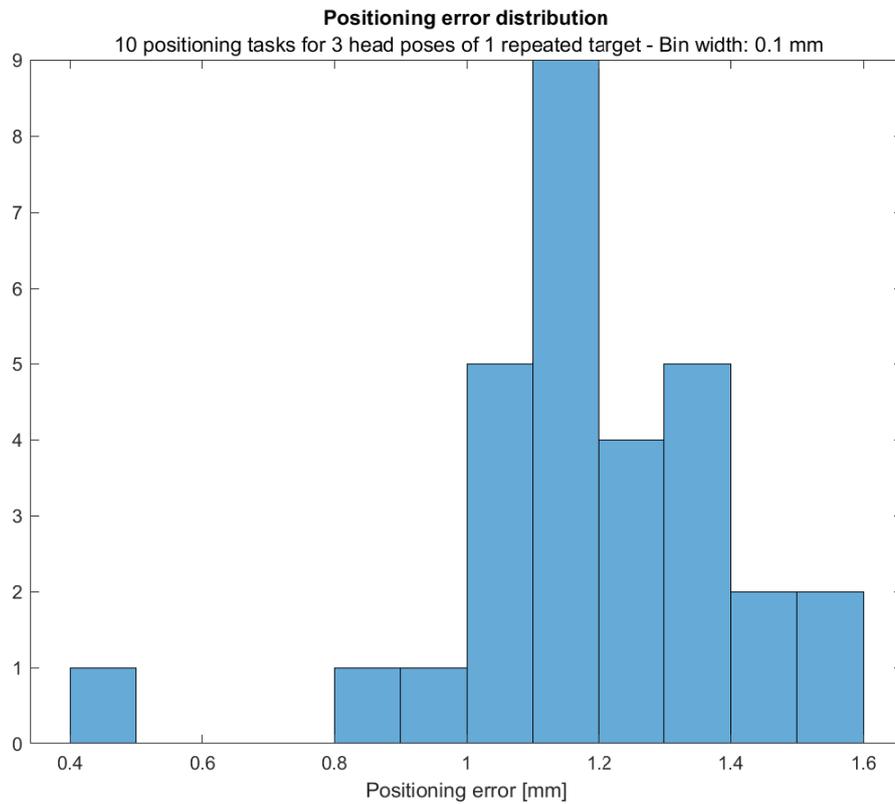


Figure 7.18: Second Session; error distribution with movement compensation approach.

## 7.3 Interpretation of the results

### 7.3.1 Session 1: "Standard" stimulation targets

The accuracy plot in Figure 7.7 shows that the positioning errors for the first attempts, in case of "standard" stimulation targets, vary between 1.1 and 9.2 mm. Significantly better results are presented in Figure 7.8, related to the final position of the coil, where the errors are limited between 0.5 and 3.4 mm. Table in Figure 7.19 shows the root mean squared errors and the standard deviations calculated for every target.

First Session Data (with fixed-head approach)				
Target	Root mean squared error [mm] First attempts	Root mean squared error [mm] Second attempts	Standard deviation [mm] First attempts	Standard deviation [mm] Second attempts
1	5.7	3.1	1.7	0.3
2	5.9	2.5	0.8	0.2
3	3.6	2.0	1.1	0.2
4	2.4	1.3	1.3	0.1
5	4.8	2.5	0.8	0.5
6	7.7	2.1	1.4	0.7
7	6.4	2.3	1.1	1.1
8	7.5	1.8	1.0	0.8
9	4.5	2.0	0.5	0.3
10	6.6	1.8	0.7	0.3
Mean	5.4	2.1	1.0	0.5

Figure 7.19: RMSEs and Standard deviations for First Session with fixed-head approach.

The accuracy errors for the first session ("standard" targets) can be attributed to three main factors: i) calibration errors, ii) disturbance force exerted by coil tubes, and iii) lack of rotation matrices for "standard" stimulation targets. In chapter 8 will follow more details about these three factors.

### 7.3.2 Session 2: repeated stimulus targets

The accuracy plot in Figure 7.10 shows that the positioning error in case of repeated stimulus targets varies between 0.4 and 2 mm. The results presented in Figure 7.11, related to the angle mismatches between the target and the coil orientations, show a mean rotation error of -0.43 degrees about z axis, -0.37 degrees about y and 0.2 about x. This information confirms that the final positions of the coil are sufficiently precise to achieve the "Aim" status and enable the operator to deliver the stimulation pulses. Table in Figure 7.20 shows the root mean squared errors and the standard deviations referred to each target.

Second Session Data (with fixed-head approach)		
Target	Root mean squared error [mm]	Standard deviation [mm]
1	1.6	0.2
2	1.1	0.4
3	1.5	0.3
4	1.6	0.3
5	1.6	0.2
6	0.7	0.2
7	0.9	0.3
8	1.1	0.4
9	1.1	0.3
Mean	1.3	0.3

Figure 7.20: RMSEs and Standard deviations for Second Session with fixed-head approach.

### 7.3.3 Movement compensation case

The plots shown in paragraph 6.2.1 demonstrate that the navigation approach proposed to compensate the target movements is satisfying the expectations. In fact, the robot is able to efficiently reposition the coil as soon as the patient's head is moved and the difference between the coil and the target locations increases over the fixed threshold of 4.5 mm for "standard" targets, or over the NBS System threshold of 2 mm for repeated stimulus targets.

It has to be noted that the targets adopted for the last two sessions correspond to targets number 8 of Figures 7.2 and 7.3 respectively. Comparing the data obtained applying the movements compensation method (Figures 7.13 and 7.14) with the results from the approach with fixed robot-MRI transformation matrix, it can be seen that the accuracy errors for the same "standard" target are substantially reduced. In particular, for the first attempts, the maximum error decreases from 9 mm (fixed-head approach) to 3.9 mm (movement compensation); moreover, for the final attempts, it varies from 2.5 mm (fixed-head approach) to 1.6 mm (movement compensation). Table in Figure 7.21 shows the root mean squared errors and the standard deviations for each head pose, referred to the first experimental session.

<b>First Session Data (with movement compensation)</b>				
Pose	Root mean squared error [mm] First attempts	Root mean squared error [mm] Second attempts	Standard deviation [mm] First attempts	Standard deviation [mm] Second attempts
0°	1.8	1.3	0.6	0.2
+20°	2.7	0.8	0.3	0.1
-10°	2.9	1.1	0.6	0.2
Mean	2.5	1.1	0.5	0.2

Figure 7.21: RMSEs and Standard deviations for First Session with movement compensation.

On the other hand, the positioning error remains of the same order of magnitude in case of repeated stimulus. In terms of angles mismatch, it is evident a higher accuracy using the movement compensation, especially for rotations about the x axis. Table in Figure 7.22 shows, for the second session, the root mean squared errors and the standard deviations.

<b>Second Session Data (with movement compensation)</b>		
Pose	Root mean squared error [mm]	Standard deviation [mm]
0°	1.2	0.3
+20°	1.2	0.2
-10°	1.2	0.1
Mean	1.2	0.2

Figure 7.22: RMSEs and Standard deviations for Second Session with movement compensation.

# Chapter 8

## Discussion

This chapter is dedicated to the discussion of the experimental results and of the issues encountered during the project development, with the proposal of possible solutions or improvements for the future studies.

### 8.1 Experimental data

#### 8.1.1 Fixed-head approach: first session

Considering that “standard” stimulation targets are not associated to an approach rotation matrix and precise information regarding the head surface is missing, the results obtained are satisfying the expectations, with an overall mean error of 2.1 mm for the final positions of the coil. It is useful to remark that the NBS System accuracy threshold for the initial patient’s registration, and the coil positioning task for repeated stimulus targets, is set to 2 mm. As mentioned in paragraph 7.3, here follows a more detailed description of the factors that may contribute to the mismatch errors between the targets and the final coil locations.

i) During the calibration task, the transformation matrix from the robot base frame to the MRI coordinate system is calculated, as reported in the previous paragraph and used in all calculations related to coil positioning. The mean value of the error is 1.5 mm, nevertheless, this error is related to the 6 points used during the calibration task; for other locations in the workspace and considering also different rotations for the coil, the mismatch between the positions registered by the robot and by the NBS tracking unit could be greater. Nevertheless, one should also consider possible measurement errors for the estimation of the transformation matrix from the robot flange to the end-effector.

ii) The second factor is related to the interfering action of the tubes and cables connected on the rear side of the coil. It has already been mentioned that they have a non-negligible weight and a limited flexibility that influences not only the movements of the coil, but also its position. In fact, the custom connector designed to fix the coil to the robotic arm also aims to limit the mechanical force applied by the pipes, but it is not sufficient to suppress it completely. As a result, tubes exert a torque to the coil, leading to an undesired coil rotation about the connector pin axis.

iii) The third and final issue pointed out for the first session with fixed-head approach is the absence of the information regarding the coil rotation to approach the real shape of the scalp surface. Differently from the repeated stimulus targets, the “standard” ones do not include any data about the coil orientation, so they are estimated using the spherical

approximation. However, this approximation is not accurate enough in case the targets are located on the head sides, where the surface is not properly curved but it is extended to reach the temples and the front side. This characteristic impacts on the positioning task performance: if the coil is not tangentially oriented to the head's surface, it may happen that it does not touch the scalp with its center but with a different area, stopping the movement (due to the contact detection) and consequently distancing the final target from the coil central spot.

### **8.1.2 Fixed-head approach: second session**

As already mentioned in the previous chapters, for a repeated stimulus the rotation matrix related to the target is available and it enables an accurate approach of the target itself. Moreover, it also allows the partial compensation for the tubes influence on the coil orientation, leading to small angle mismatches and to an overall error that is less than 2 mm, considered the NBS system maximum error threshold for the Registration process and coil positioning with repeated targets. According to the last consideration, the performances of the Robotic Assistant are fully satisfying the expectations.

### **8.1.3 Movement compensation approach**

The improved result obtained with this approach, with respect to previous one, could be explained considering that the movement compensation method periodically updates the robot-MRI transformation matrix, instead of using a static matrix calculated during calibration, reducing the error. Regarding the repeated stimulus target, the difference between the two approaches is less evident, since both adopt a correction system for the coil frame orientation, in order to reduce the accuracy errors below the threshold of 2 mm. In conclusion, the movement compensation approach offers an efficient way to position the coil when the head is not steady with respect to the robot base frame. Moreover, the results obtained show an accuracy improvement for the same targets considered also for the fixed-head approach; however, since the experiments considered only one target per kind, it is not possible to arrive at the same conclusions also for other targets; this means that there is not the possibility to confirm that the movement compensation method is more accurate than the fixed-head approach, unless of further investigations.

## **8.2 Hardware setup**

With the current hardware setup, described in chapter 6, it has been possible to implement and test the first realization for this Robotic Assistant, with satisfying results. The TMS System is sufficiently close to the robot position, allowing the fastening of the coil to the robotic arm through a custom connector. However, in order to improve the overall behavior of the Robotic Assistant, a first suggestion is to adopt a different arrangement of the hardware setup: in the current state, the dummy head is positioned beside the robot, facing to the tracking unit, while the tubes are coming from a lower level, and they are forced to assume inconvenient shapes to follow the robot movements. A better configuration could be obtained positioning the robot and the head on a further down plane, in order to allow the cables to comfortably reach the coil reducing the induced mechanical stress; moreover, the head could be moved in front robot base, within given limits due to the tracking unit arm, reducing the cases where the robot forces the tubes

to assume a stressful position. This would also improve the contact detection method performance since the robot torque sensors are less affected by the effects of the hanging tubes. In addition, a new custom connector could be designed, to improve the stability of the coil and suppress any other induced movement; the new connector should wrap the coil handle, carrying out other functions such as the weight support, and ensuring a safety mechanism in case of a structural failure. The material of the connector could also be changed, with the aim to increase the structural strength.

### **8.3 Collision detection**

During the experiments, it has been noticed a flaw regarding the contact and collision detection methods. It has already been introduced that external forces estimation is influenced by the additional action of the tubes. A possible implementation for a contact detection method has been presented in the previous paragraphs, but its algorithm is activated only during the movements toward the head. In fact, applying the same algorithm to other movements could lead to accidental interruptions of the tasks caused by the reaching of the force derivative threshold. For these reasons, the implementation of a general collision detection method has been postponed for future investigations, in order to satisfy all the functional requirements pointed out during the design phase.

### **8.4 Contact detection**

Regarding the coil positioning task, the experimental sessions proved that the position control of the robot is sufficiently high to ensure the stimulation conditions; nevertheless, during the experiments it has been noticed that, sporadically, the coil was not always properly touching the head surface, leaving a tiny gap in the order of 1 mm. Even though this could be considered a minor issue, for example because, in a real case, there is usually the hair thickness between the scalp and the coil surface, one would prefer to ensure a secure contact between the two.

This issue can be addressed in several ways: i) forcing the robotic arm to stop the movement toward the head only when a contact is detected, even if this spot does not match the calculated one, that could be affected by an error (not only due to the Robotic Assistant, but also to the NBS System). Moreover, since the contact and the collision detection methods could be improved, this solution could be applicable when these methods are considered precise enough to always distinguish a coil contact with the head. ii) Another concept is to equip the coil with a sensor suitable for this task, such as a pressure sensor or a strain gauge; however, attention must be paid to not interpose any conductive material between the coil and the head, for safety reasons and magnetic pulse interference. To bypass this constriction, a limit switch could be fixed to the coil side, while a thin part made by insulator material, placed on the coil surface, could mechanically act on it.

### **8.5 Functional requirements**

With respect to the functional requirements pointed out in paragraph 5.2.1, the Robotic Assistant fully satisfies features 1, while feature 2 is completely accomplished only if the movement compensation method is used. Regarding feature 3, the contact and collision

detection methods should be improved in order to guarantee the activation of safety procedures; based on the tests and experiments results presented in this thesis, it is not possible to declare that the collision detection method proposed is always able to recognize the contact between the coil and the patient's head.

Moreover, due to the external forces applied to the joints and their highly varying values estimated through the sensors in the robotic arm, it has not been possible to develop a collision detection method that is able to detect and distinguish a collision from a contact. Further investigations are required to fulfill this requirement in future Robotic Assistant implementations.

Finally, considering the last requirement, requirement 01 has not been implemented, because when the NBS Software is closed, the Robotic Assistant first recognizes that the markers are no longer visible. This halts the current task in execution, the robot stops and the operator intervention is required. Moreover, the permissions required from the operator to allow the robot movements (REQ-02 and REQ-03) are a consequence of the operator actions on the NBS Software, that are checked through the NBS status monitoring tasks mentioned in chapter 5. REQ-04 and REQ-05 are also fulfilled via the safety mechanisms explained in paragraph 6.6.3, but since there is no information available about the digitizing pen via the Robotic Interface, REQ-06 could not be satisfied.

## **8.6 Future implementations**

An ideal TMS robotic assistant should not only be able to carry out the coil positioning task, but also "learn" from the functional motor map of the patient, by intelligently using the EMG information associated with the pulses delivered (applying deep learning algorithms on the data collected). The project presented in this thesis should be considered as a first step toward a system that will include functions to analyze and learn from the real-time data acquired from the NBS System; indeed, the current NBS Interface does not allow to request EMG data related to the stimulations, a possibility that will be implemented in the next versions of the Robotic Interface. When this will be possible, the Robotic Assistant will be able to assist the doctors not only during the treatment, but even with the data analysis which leads to the formulation of a diagnosis.

# Chapter 9

## Conclusions

During this thesis, a nTMS Robotic Assistant has been proposed. After a brief introduction of the human nervous system, the two main building blocks of the project are described: i) the transcranial magnetic stimulation technique, with the NBS System by Nexstim, and ii) the Panda robot by Franka Emika. Then the realization phases are introduced: i) the design phase, with the system functional requirements and the system architecture, ii) and the implementation phase, that describes the hardware and the software solutions developed to realize the Robotic Assistant. To assess the performances of the implemented project, several experimental sessions have been performed, in order to investigate the different approaches proposed. The results are satisfying: the coil positioning accuracy depends on the target nature, but for “standard” targets a maximum mismatch error of 3.5 mm can occur between the target and the final coil location; on the other hand, for repeated stimulus targets error of 2 mm was observed (which is the NBS System accuracy threshold). Two different navigated approaches are investigated: the first with a fixed head position, the second with a head movement compensation algorithm. The overall performances of the project fulfill most of the functional requirements pointed out, and further research and investigations for improvements and additional functionalities can enhance the characteristics of the final nTMS Robotic Assistant.



# References

- [1] L. Richter. *Robotized Transcranial Magnetic Stimulation*. Springer Science+ Business Media New York, 2013.
- [2] A. Rotenberg, J. Cooney Horvath, and A. Pascual-Leone. *Transcranial Magnetic Stimulation*. Springer Science+Business Media New York, 2014.
- [3] S. M. Krieg. *Navigated Transcranial Magnetic Stimulation in Neurosurgery*. Springer International Publishing AG, 2017.
- [4] N. Palastanga and R. Soames. *Anatomy and Human Movement, Structure and Function*. 6th ed. Elsevier Ltd., 2012.
- [5] Medium69 and Jmarchn. *A diagram of the human nervous system*. Accessed September 2021. CC BY-SA 4.0 '<https://creativecommons.org/licenses/by-sa/4.0/>', via Wikimedia Commons. 2014. URL: [https://commons.wikimedia.org/wiki/File:Nervous\\_system\\_diagram\\_unlabeled.svg](https://commons.wikimedia.org/wiki/File:Nervous_system_diagram_unlabeled.svg).
- [6] J. Walinga. *Diagram of basic neuron and components*. Accessed September 2021. CC BY-SA 4.0 '<https://creativecommons.org/licenses/by-sa/4.0/>', via Wikimedia Commons. 2014. URL: [https://commons.wikimedia.org/wiki/File:Components\\_of\\_neuron.jpg](https://commons.wikimedia.org/wiki/File:Components_of_neuron.jpg).
- [7] Cancer Research UK. *Diagram showing some of the main areas of the brain*. Accessed September 2021. CC BY-SA 4.0 '<https://creativecommons.org/licenses/by-sa/4.0/>', via Wikimedia Commons. 2014. URL: [https://commons.wikimedia.org/wiki/File:Diagram\\_showing\\_some\\_of\\_the\\_main\\_areas\\_of\\_the\\_brain\\_CRUK\\_188.svg](https://commons.wikimedia.org/wiki/File:Diagram_showing_some_of_the_main_areas_of_the_brain_CRUK_188.svg).
- [8] Blausen.com staff (2014). *Diagram of basic neuron and components*. Accessed September 2021. "Medical gallery of Blausen Medical 2014". WikiJournal of Medicine 1 (2). DOI:10.15347/wjm/ 2014.010. ISSN 2002-4436., CC BY 3.0 '<https://creativecommons.org/licenses/by/3.0/>', via Wikimedia Commons. 2014. URL: [https://commons.wikimedia.org/wiki/File:Blausen\\_0102\\_Brain\\_Motor%5C%26Sensory.png](https://commons.wikimedia.org/wiki/File:Blausen_0102_Brain_Motor%5C%26Sensory.png).
- [9] Nexstim Plc. *Product training, Nexstim NBS System 5, training material*. n.d.
- [10] NDI. *Polaris Vicra*. Accessed September 2021. 2020. URL: <https://www.ndigital.com/products/polaris-vicra/>.
- [11] Nexstim Plc. *Nexstim NBS 4.5 - Robotic interface (prototype) - Supplemental User Manual Sheet*. 2021.
- [12] WiredWorkers. *Robotics at your service*. Accessed September 2021. <https://wiredworkers.io>.

- [13] Franka Emika. *Panda Solution*. Accessed September 2021. 2019. URL: <https://s3-eu-central-1.amazonaws.com/franka-de-uploads/uploads/Panda-solution-EN.pdf>.
  - [14] Franka Emika. *Panda Powertool*. Accessed September 2021. 2019. URL: <https://s3-eu-central-1.amazonaws.com/franka-de-uploads/uploads/Panda-Powertool-EN.pdf>.
  - [15] L. Johansmeier. *Mios*. Lehrstuhl für Robotik und Systemintelligenz (RSI-Wiki). Accessed September 2021. 2019. URL: <https://wiki.tum.de/pages/viewpage.action?spaceKey=RSIWiki&title=MIOS>.
  - [16] L. Johansmeier, M. Gerchow, and S. Haddadin. “A framework for Robot Manipulation: Skill Formalism, Meta Learning and Adaptive Control”. In: *arXiv: 1805.08576 [cs.RO]* (2018).
  - [17] F. Voigt, L. Johansmeier, and S. Haddadin. “Multi-Level Structure vs. End-to-End-Learning in High-Performance Tactile Robotic Manipulation”. In: *Conference on Robot Learning* (2020).
  - [18] Brain Center TMS – Transcranial Magnetic Stimulation. *How Long do the Effects of TMS Treatment Last?* Accessed September 2021. 2021. URL: <https://braincentertms.com/how-long-do-the-effects-of-tms-treatment-last/>.
  - [19] Johns Hopkins Medicine. *Psychiatry and Behavioral Sciences – Frequently Asked Questions About TMS*. Accessed September 2021. URL: [https://www.hopkinsmedicine.org/psychiatry/specialty\\_areas/brain\\_stimulation/tms/faq\\_tms.html](https://www.hopkinsmedicine.org/psychiatry/specialty_areas/brain_stimulation/tms/faq_tms.html).
  - [20] Axilum Robotics. *Robotic Assistant for Transcranial Magnetic Stimulation (TMS)*. Accessed September 2021. URL: <https://www.axilumrobotics.com/en/tms-robot/>.
  - [21] ANT Neuro. *Smartmove – robotized TMS positioning solution*. Accessed September 2021. URL: [https://www.ant-neuro.com/sites/default/files/SLS-SM-0021.01rev012018-01-08\\_smartmove\\_brochure\\_A4.pdf](https://www.ant-neuro.com/sites/default/files/SLS-SM-0021.01rev012018-01-08_smartmove_brochure_A4.pdf).
  - [22] A. Nocco, A. Mioli, M. D’Alonzo, et al. “Development and Validation of a Novel Calibration Methodology and Control Approach for Robot-Aided Transcranial Magnetic Stimulation (TMS)”. In: *IEEE Transactions on Biomedical Engineering* 68.5 (May 2021).
  - [23] H. Wang, J. Jin, X. Wang, et al. “Non-orthogonal one-step calibration method for robotized transcranial magnetic stimulation”. In: *BioMedical Engineering OnLine* (2018).
  - [24] X. Yi and R. Bicker. “Design of a Robotic Transcranial Magnetic Stimulation System”. In: *IEEE Conference on Robotics, Automation and Mechatronics* (2010).
  - [25] Oracle. *The Java Tutorials – What Is a Socket?* Accessed September 2021. URL: <https://docs.oracle.com/javase/tutorial/networking/sockets/definition.html>.
  - [26] D. Akca. *Generalized Procrustes Analysis and its Applications in Photogrammetry*. Internal Technical Report at IGP – ETH, Zurich, June 2003.
-

- [27] K. S. Arun, T. S. Huang, and S. D. Blostein. “Least-Squares Fitting of Two 3-D Point Sets”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9.5 (Sept. 1987).
  - [28] T. S. Huang, S. D. Blostein, and E. A. Margerum. “Least-squares estimation of motion parameters from 3-D point correspondences”. In: *Proc. IEEE Conf. Computer Vision and Pattern Recognition* (June 1986).
  - [29] LearnOpenCV. *Rotation Matrix To Euler Angles*. Accessed September 2021. 2016. URL: <https://learnopencv.com/rotation-matrix-to-euler-angles/>.
  - [30] Sami Haddadin. “Towards Safe Robots, Approaching Asimov’s 1st Law”. Chapter 3.3. Accessed September 2021. PhD thesis. Fakultät für Elektrotechnik und Informationstechnik, Rheinisch-Westfälischen Technischen Hochschule Aachen, 2011. URL: <http://publications.rwth-aachen.de/record/52869/files/3826.pdf>.
  - [31] S. Haddadin, A. De Luca, and A. Albu-Schäffer. “Robot Collisions: A Survey on Detection, isolation, and Identification”. In: *IEEE Transactions on Robotics* 33.6 (Dec. 2017).
-



# Appendix A

## List of Abbreviations

<b>Abbreviation</b>	<b>Meaning</b>
CNS	Central Nervous System
CST	Corticospinal Tract
EMG	Electromyography
FCI	Franka Control Interface
FDA	Food and Drug Administration
fMRI	Functional Magnetic Resonance Imaging
GUI	Graphical User Interface
MEG	Magnetoencephalography
MIOS	Machine Intelligence Operating System
MRI	Magnetic Resonance Imaging
MSRM	Munich School of Robotic and Machine Intelligence
NBS	Navigated Brain Stimulation
nTMS	Navigated Transcranial Magnetic Stimulation
PNS	Peripheral Nervous System
TMS	Transcranial Magnetic Stimulation
USB	Universal Serial Bus
SVD	Singular Value Decomposition