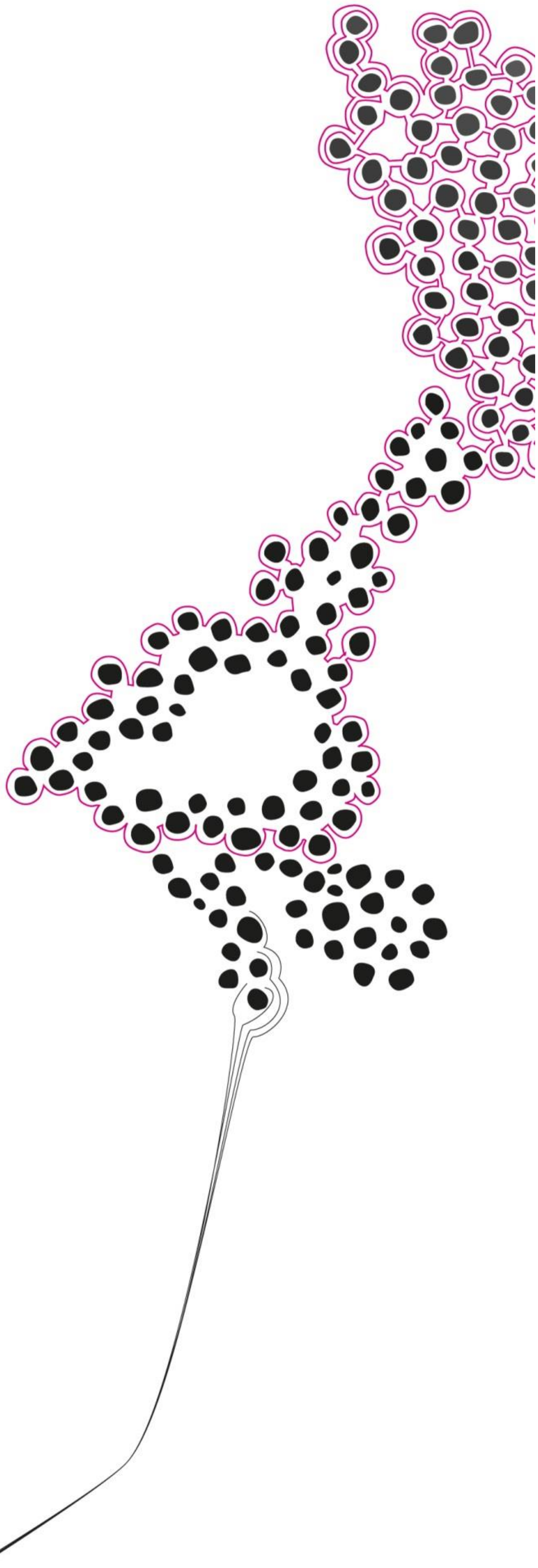
PDENG THESIS



DESIGN OF A ROBOT FOR TMS DURING TREADMILL WALKING

J.J. de Jong

CONSTRUCTION TECHNOLOGY / BIOMEDICAL ENGINEERING
BIOMECHATRONICS AND RENABILITATION

EXAMINATION COMMITTEE
H. van der Kooij
A.H.A. Stienen
M. van de Velde
J. van Dijk

DOCUMENT NUMBER
BW - 1

UNIVERSITY OF TWENTE.

26-05-2015

Design of a robot for TMS during treadmill walking

- *Design and Testing* -

PDEng Thesis

to obtain the Professional Doctorate in Engineering at the University of Twente,

to be publicly defended on Tuesday the 26th of May 2015 at 14:00,

by

Jan Johannes de Jong
Geboren op 14 september 1986 te Bellville, Zuid-Afrika

Table of contents

Chapter 1.General introduction

During the recent years, transcranial magnetic stimulation (TMS) has come under the attention as a non-invasive tool for diagnosis and therapy. It is used for a wide variety of pathologies such as depression, stroke, and, Parkinson's disease. TMS is induced by running large currents though an iron centered coil held against the scalp. These currents will generate a magnetic field inside the brain, inducing small electrical currents. The coil can be placed on the skull to suppress or stimulate certain structures, resulting in locally changed neurological behavior[1]. During most TMS therapies a train of short stimulation pulses, called repetitive TMS (rTMS), is applied to the cortex of interests. The intensity, pulse width, number of pulses, interval time and other parameters can be tuned for optimal stimulation. When the TMS is applied to the motor cortex, involuntary muscle activity can be observed. These are called motor evoked potentials (MEPs). To identify which part of the brain is responsible for the motor pathways associated with a muscle of interest, the activity of the muscle is measured while stimulating over a grid of stimulation points. The areas that result in the most muscle activity are named hot spots. This will give a localization of motor function and is called 'motor mapping'. This method is also used to find the most optimal stimulation site and stimulation parameters before the actual stimulation is performed. Usually, the optimal stimulation parameters are found by increasing the intensity and frequency until 50% of the pulse trains result in a MEP in the in target muscle. The same method can be used to identify the severity and the functional localization of brain damage of stroke survivors or other neurological diseases. The coil can be placed over the affected area to identify which part of the motor pathways are still functioning and which parts are impaired.

Besides the identification of the (pathological) brain, TMS is also used to relieve or even cure some diseases. One of the most used and researched applications of TMS is the treatment of depression. However, the underling neurological mechanism is still unknown and the effectiveness has not yet been proven [1], [2]. This is mainly due to the difficulty to apply 'sham'-based double-blinded tests and the accuracy of manual stimulation.

Another field of interest is that of motor relearning. It is thought that the level of TMS excitability of a part of the brain is a measure for the motor adaption of the subject [3]. In addition, it is hypothesized that TMS can not only identify and quantify the connectivity within the brain; it might also stimulate motor (re)learning when applied before, during or after activities [3]–[6]. For example, in stroke survivors the healthy hemisphere can be inhibited to encourage the patient to use the affected hemisphere in order to develop new motor pathways.

However, for some activities this poses a problem, since the TMS coil must be held at a constant relative position during the natural sway of the head during these activities. For example, during treadmill walking the head can move as much as 10 cm [10]. To move the coil with the head, a novel TMS robot is proposed.

I. Project description

The aim of this project is the design of a robot that can safely and accurately apply TMS during treadmill walking. Such a system enables investigation of neuroadaptation and motor (re)learning during activities via TMS simulation. Additionally, it can be used for TMS treatment and motor mapping.

II. System overview

The final design of the robot is shown in Figure 6. The robotic system supports a TMS coil (A) by a spring system (B). This spring presses the coil against the head to prevent a collision and to ensure a soft human-robot interaction. The spring and the coil are both positioned by a six degree of freedom (6 DOF) robotic manipulator: the Hexa robot (C). To adjust for different tasks and subjects, a support system (D) places and holds the robotic manipulator at different heights and angles. The head position is tracked with a 3D optical tracking system (E) and fed into the control mechanism, which steers the robot. This TMS robot can be used on several base stations such as a simple chair and the LOPES rehabilitation robot[7].

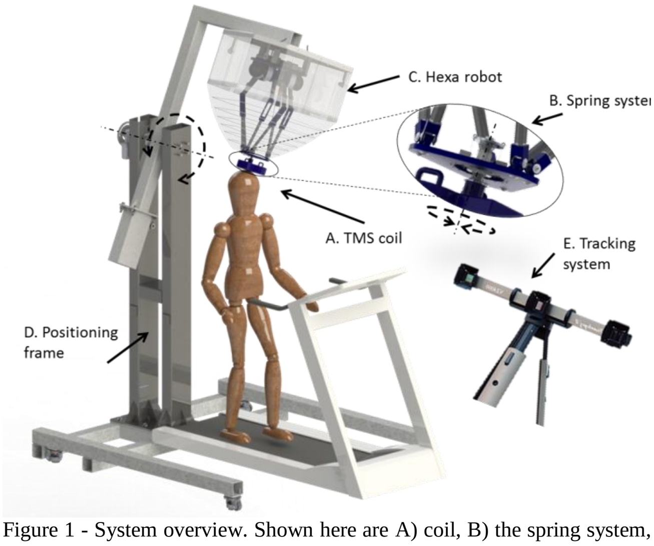


Figure 1 - System overview. Shown here are A) coil, B) the spring system, C) Hexa robot, D) positioning frame and the E) head tracking system.

[1] This introduction is based on the work done during the master thesis, and article [37].

III. Assignment

This report is the result of the work done at the University of Twente - during a Professional Doctorate in Engineering (PDEng) program - on the TMS robot project. Several parts of the assignment where previously done during a master thesis.

The first part of the assignment was the formulation of design requirements of the TMS robot. This includes the analysis of the range of motion of the head during walking. Furthermore, a study into the safety of medical robots has been conducted and the requirements concerning the safety were formulated. These parts were conducted during the master thesis.

The second task is the design and construction of this TMS robot. The general structure was designed during the master period. Several manipulators were compared and a proper geometry was selected. During the PDEng assignment, the TMS robot with all the components was detailed and manufactured. The detailing and construction of the electronic and mechanical parts was done with the help of others. An external safety system was designed and fabricated with the help of an external party. The controller design consists of a position controller to steer the robot, a head tracking part - which calculates the desired robot pose from the measured optical markers, and a force controller which controls the contact force with the subject.

The third task is the evaluation of the robot design. First, the robot should be able to move according to the specified requirements. This involves calibration and evaluation of the kinematics of the robot. It also includes dynamic testing of the robot and its controller. Secondly, the robot should be able to follow motion of the head during treadmill walking. Finally, the TMS stimulation should be applied to see the usability and efficacy of the system. This last part could not be conducted during this assignment.

IV. Assignment framework

This assignment was conducted in collaboration with the company ANT Neuro. ANT Neuro is an Enschede-based company that is specialized in neuroscience and neuronavigation. Their products include various EEG measuring devices, neuronavigation systems, and a TMS robot. Their interest lies in the design of a novel, more versatile TMS robot, and in the commercial application in the future.

The design and evaluation of a novel TMS robot is supported with grand PIDON082046 of the Overijssel government. The research was conducted during the Master and PDEng assignments. The PDEng program constitutes of one year of courses and a one year design assignment. During the Master assignment, the requirements were specified and system design was conducted. During the PDEng program, the design has been detailed and constructed. The control and evaluation of the robot was performed during the PDEng phase.

The research was conducted at the Laboratory for Biomedical Engineering at the University of Twente. Here, several novel revalidation devices and robots have been designed. Notably, the robot treadmill trainer LOPES [7] - which is currently in the clinical evaluation phase - has been developed and tested here. This group does research on the usability of neurological stimulation methods such as transcranial direct current stimulation (TDCS) and TMS as a tool to identify and promote neurorehabilition. In this research framework, the TMS robot is to be used to understand the influence of TMS on motor relearning and to use it as a quantification tool for other stimulation strategies.

V. Background information

In the next section, several essential aspects of the design considerations of the TMS robot will be explained.

A. Safety design

For the design of any medical robot - and a TMS robot specifically - the safety is the most important design criteria. The first safety measure for industrial robots often is to mechanically or electronically fence them off from humans to prevent contact at all times. However, medical robots are designed to work in close interaction with humans in order to aid them. Completely fencing the robot off is often not an option to achieve complete safety.

In Chapter 4, a literature review is done on how medical robots are designed for safety. From literature, it can be concluded that sensor failure is amongst the biggest risks, as it can result in a run-away robot [8], [9]. Additionally, a risk exists when the robot controller has an internal malfunction. One can think of a lock-up or a broken output. This can also result in a run-away robot. As countermeasures to these failures redundant sensors, redundant controllers and watchdog are proposed [10]. A watchdog is an external timer, which requires active communication from the controller to prevent it from executing an emergency stop. This can be used to detect computer lock-up. Mechanical risks include impact with the subject and pinching of limbs, such as fingers. This can be prevented by shielding small orifices of the robot by a protective cover. More failure modes have been identified in a Failure Mode Effect Analysis (FMEA), which results are described in Chapter 4.

Medical safety is designed on four levels. The first level is to prevent malfunctions in the first place. This requires the sensors and components to be built with the highest quality. A maintenance protocol has to be implemented to detect wear and prevent possible malfunctions. The misuse of the machine is to be prevented for example with intuitive interface and user manual. The second level includes a timely detection of malfunctions or otherwise dangerous situations. This can be achieved by using redundant sensors and controllers to check and guarantee the function of the primary ones. The third level requires the robot to perform a timely emergency stop whenever a possible hazard is detected. This requires the robot to quickly and safely dissipate its kinetic and electrical energy. It also requires the stop conditions to pose no danger for the subjects, i.e. the robot does not collapse in case of power break. The fourth level is to minimize the effect of malfunction on the subject. This can be achieved by designing a robot which has low potential, kinetic, electric energy and by isolating the subject as much as possible from the robot, for example with elastic elements.

B. Current positioning devices for TMS

To reduce the strain on the physician during long stimulation sessions and to increase the accuracy of stimulation, several stereotactic neuro navigations systems have been developed [11]. These systems rely on pre-recorded fMRI images and optical head tracking to show the

Figure 2 – NeuroStar's static TMS positioning device (Source: http://www.neurostar.com Visited:23/04/2015)

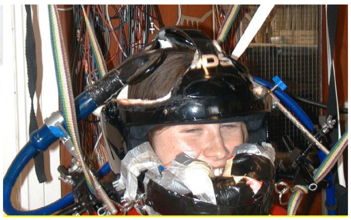


Figure 3 – TMS stimulation fixed to the skull using a modified motor cycle helmet. (Source: http://www.pdn.cam.ac.uk/staff/edgley/index.shtml, Visited: 23/04/2015)

Figure 4 – Commercial available TMS robot using an industrial type robot. Produced by ANT –neuro (Source: https://www.ant-neuro.com/products/smartmove Visited:23/04/2015)

Figure 5 – Commercial available TMS robot produced by Axilum robotics (Source: http://www.axilumrobotics.com Visited:19/03/2015)

stimulation site and position of the coil. These systems require a physician to move the coil to the desired position. A passive support system keeps the coil in position.

Several TMS-robots have been developed to improve the accuracy and repeatability of stimulation. These TMS-robots use optical tracking in combination with fMRI images to locate and follow the stimulation site. The current TMS robot such as the NeuroMate (ISS/IMMI, Sacramento, CA) [12], [13], Adept Viper s850 (Adept Technology, Inc. Livermore, CA, USA) [14], and Kuka KR3 (Ausburg, Germany) [15] rely on commercially available robotic manipulators to place the stimulator. In the recent years a specialized design for TMS robot is presented [16]. This robot is commercialized by Axilum robotics

These robots are multiple DOF serial type robots. The serial robot types are known for having a large workspace (operational area). Since the motors are mounted on the moving parts, the active mass and required motor power is high. For operation near humans the power of these heavy robots needs to be restrained to reduce the effect of impact.

The application of TMS during activities such as walking poses a challenge since the TMS coil must accurately follow the head during the natural swaying motion. Several research groups mount the TMS coil on the head during treadmill walking. For example, by connecting of the coil to a helmet [5], [17] or by mounting the coil on a harness [18], [19]. The main disadvantages of these methods are that the natural movement of the head is inhibited to some extent and significant slack between the head and the coil may occur, reducing the accuracy and repeatability of the stimulation. Furthermore these methods do not allow a simple relocation of the stimulator for example to another stimulation site or for grid finding. To achieve a safe and accurate robotic TMS stimulation during head movement a novel robot design is needed.

C. Parallel robots

Safety design of the TMS robot requires the highest possible reduction of kinetic energy and therefore of the moving mass. A distinctive innovation of this project is to use of a parallel manipulator –mechanical, moving part of the robot- instead of serial manipulators for TMS stimulation. Serial manipulators consist of a single kinematic chain connecting the end-effector (tool) to the base. Merlet [20, p. 13] defines the class of parallel manipulators as follows: "*A **parallel robot** is made up of an end-effector with n degrees of freedom, and of a fixed base, linked together by at least two independent kinematic chains. Actuation takes place through n simple actuators*". The main advantage over serial robots is that the motors can be mounted on the base, reducing its active mass significantly and, consequently, increasing its intrinsic safety. Parallel manipulators can be much stiffer due to the higher number of connections between the end-effector and the base. Parallel structures generally have a smaller workspace, reducing the dangerous (live) area of the robot.

The range of motion (ROM) of a robot is the theoretically reachable area of the robot. In practice, the manipulator may not be controllable in each point, due to singular configurations. A singular configuration can best be understood as a pose in which the end-point has an uncontrollable degree of freedom, leading to the loss of inherent rigidity.

VI. Outline

This report consists of seven chapters. The *second chapter* consists of the main article of this thesis. Here, the complete design of the robot is presented. The following chapters expand on the theory, choices and results of this chapter. The *third chapter* presents the work done on improving the tracking quality of the robot. Two head tracking methods are presented and compared for accuracy and noise suppression. The *fourth chapter* gives the design considerations on the safety of the robot. An attempt is made to answer the following question: "How can the robot be build such that it poses no threat under any condition or failure?" To understand the function of the Hexa robot, the kinematic model is presented in the *fifth chapter*. In the *sixth chapter*, the controller aspects of the robot are shown. Here, the controller framework with the contact force controller, position controller and state controller are shown. Finally, conclusions are reported in the *seventh chapter*.

Chapter 2.Robotized TMS for application during treadmill walking

Ir. Jan. J. de Jong, Dr. ir. Martijn Wessels, Prof. dr. ir. Herman van der Kooij, Dr. ir. Arno H. A. Stienen

***Abstract*— A novel robot has been designed for the application of Transcranial magnetic stimulation (TMS) during motion therapies such as treadmill walking. As the velocity of the head during treadmill walking exceeds the velocity safety limits of conventional TMS robots, a novel robot design is required that combines high velocity with intrinsic safety. This design consists of a safety spring system, to ensure soft contact between the robot and the subject, a parallel robot mechanism for lightweight, fast and accurate placement of the stimulator, and a high accuracy motion-capturing device for tracking of the subject. An external safety system measures the contact force, and controls the power to the robot to ensure the safety offered by the robot. The system has proven capable of tracking the head of a subject during slow movement (<0.05 m/s). However, faster motions are limited by the bandwidth of prefilter that is required for attenuation of the input measurement noise of the tracking system. Therefore, additional inertial motion sensors and Kalman filtering techniques are recommended to achieve the accurate and high velocity head tracking required for TMS during treadmill walking.**

***Index Terms*—Medical robots, transcranial magnetic stimulation (TMS), motion tracking.**

I. Introduction

Transcranial magnetic stimulation (TMS) is a non-invasive tool for the electromagnetic stimulation of neurological tissue. It uses a strong magnetic pulse, induced through an iron core coil, to excite or inhibit neurons in the brain or spinal tract. This method is used for a wide variety of applications including the treatment of depression, migraine, stroke and Parkinson's disease [21]–[24]. Over recent years, TMS has gained interest as a motor cortex identification tool [1], [3], [25], [26]. It is used to quantify the excitability and connectivity within the brain for example, before, during and after motor training. Furthermore, TMS has shown to have a positive effect on motor recovery following stroke [27]–[30]. Depending on the stimulation procedure, the TMS sessions can last up to 30 minutes. For consistent results over repeated stimulations, it is necessary to maintain the position of the coil center within a few millimeters of the targeted location. However, natural head sway of the subject undergoing TMS and fatigued arms of the practitioner as a result of the 1 to 3 kg coil can interfere with the meeting of this requirement.

To increase the accuracy and repeatability of TMS, and reduce the strain on the clinician, we developed a robotic system for TMS on subjects in motion. This TMS robot can be used during, for example, upper-extremity exercises and treadmill walking using exoskeletons.

Several conventional robots have previously been used for TMS, such as the NeuroMate (ISS/IMMI, Sacramento, CA) [12], [13], Adept Viper s850 (Adept Technology, Inc. Livermore, CA, USA) [14], and Kuka KR3 (Ausburg, Germany) [15]. More recently, Zorn et al. [16] designed a robot that uses a mechanism developed specifically for sedentary TMS. These systems use optical tracking of the subject's head to place the coil at the desired location against the scalp. These robots rely on six or seven-DOF serial manipulators of which the motors are placed at the joints. These serial robots combine a large operational area with high inertia. Therefore, for safe operation near humans, the power and velocity of these heavy robots needs to be restrained to reduce the impact of failure. For example, the maximum velocity of the device presented by Mattheus et al. [15] is approximately 0.1 m/s. These constraints preclude the use of such robots in a dynamic setting, such as TMS during treadmill walking.

In this paper, a novel TMS robot is presented for safe and accurate stimulation during treadmill walking. Firstly, the requirements for this robot are discussed. Secondly, the hardware design of the robot is presented followed by the controller design. Thirdly, the performance of the robot is evaluated based on measurements. Lastly, the paper concludes with a discussion current design and future work.

II. Requirements

A. Safety

For such a robot, safety is of utmost importance since a powerful robotic arm operating in close contact with vital parts of the subject (e.g., the head). To reduce the impact of any electrical, mechanical or software failure, the robot is required to have low kinetic energy and operate at low motor power. Unlike to industrial robots, medical robots are designed to interact physically with humans to help them. A robot must be designed such that it will not harm the subject as a result of any software, electronic or hardware

Table 1. Listing of the notations used in this article

Notation	Meaning
M	Functions or mappings are shown in regular capital.
c	Single values are denoted with a regular lower-case.
$\boldsymbol{a}$	Vectors are denoted with a bold lower-case, This also includes arrays of vectors.
$\boldsymbol{A}$	Matrices are denoted with a bold capital.
c_m	Meaning of subscript is content dependent.
${}_x\boldsymbol{a}$	The pre-subscript denotes x-dimension of vector $\boldsymbol{a}$**.**
${}^i\boldsymbol{A}$	The pre-superscript denotes i-th row of $\boldsymbol{A}$**.**
$\bar{\boldsymbol{a}}$	Averaged vector over the rows.
ψ^i	Reference frame i. This is a physical property, not a value.
${}^k\boldsymbol{p}$	Point k. This is a physical point and not a value.
${}^k\boldsymbol{p}^i$	Point k expressed in frame i. Now it is a vector with a numerical value.
$\boldsymbol{o}_i^j$	Origin of frame i expressed in frame j.
$\boldsymbol{R}_i^j$	Rotation of frame i expressed in frame j.
$\boldsymbol{H}_i^j$	Transformation matrix between two frames. Expresses frame i in frame j: $\boldsymbol{p}^j = \boldsymbol{H}_i^j\boldsymbol{p}^i$.
$\boldsymbol{t}_i^{j,k}$	The twist of frame k with respect to frame i expressed in frame j.

Table 2. Required minimal workspace and performance during treadmill walking.

	ROM	Velocity	Acceleration	Force	Accuracy
Translation	0.15 m	1 m/s	10 m/s^2	50 N	1 mm
Rotation	15 deg	250 deg/s	150 deg/s^2	5 Nm	1 deg

malfunction. This ranges from preventing mechanical hazards by crushing or pinching parts of the human body to electrical dangers from incorrectly connected signal or power lines.

One of the main risks of this robot is a too high interaction force between the subject and robot. It is vital that a highly energetic impact between the skull and the robot should be avoided at all costs. Therefore, the contact force during full operation should be limited to between 25 and 50 N. To detect failure of the control system or a sensor, all vital subsystems, such as the encoders, force sensors, power supplies, and controller, must be double-checked. In case of an emergency stop, the stop conditions must not harm the subject. This implies that the energy in the system must be dissipated quickly.

B. Performance requirement

The required performance of the robot is determined by the motion of the head during treadmill walking. The required range of motion, velocities and acceleration are based on the values reported in literature [18]–[20] and measurements on a 39-year-old subject walking on the treadmill [21]. During walking, the head motion stays within a 150 mm cube. The required rotation is about 15 deg in each direction. Table 2 presents the required performance of the robot.

Different subjects' heights and different stimulation sites require the robot to be placed over a range of 1.6-2.0 m in height. The stimulator must be placed of the complete surface of the skull. This requires the coil to be placed of a half a sphere.

The required accuracy for TMS stimulation is +/- 1 mm. The transmission of the TMS pulse is shown to be both position and orientation dependent due to the non-linear transmission of the magnetic field, to the folding of the local brain structure and to the orientation of the neurons [22], [23].

The TMS robot should be able to hold and press the 2.5 kg coil against the skull during motion with accelerations up to 10 m/s^2. In addition, a contact force of 25 N is desirable to prevent the coil from losing contact during these rapid motions. This implies a total maximal required end-effector force and moment of maximal 50 N and 5 Nm, respectively.

III. Design

The design of the TMS robot is shown in Figure 6. The stimulator is pressed against the stimulation site by a safety spring system. A six-DOF parallel manipulator, known as the Hexa, actively moves the stimulator. For intersession adjustment, the robot can be moved over a circular arch. The head movement is measured by an optical tracker system, which is used as an input to the robot controller. The robot is currently placed over a treadmill but can also be used in other subject settings such as over a chair.

A. Safety Design

The design of the robot is focused on ensuring safe interaction with the subject under all circumstances. Highly energetic collisions between the stimulator coil and the subject's skull is one of the major hazards of a TMS robot. This impact can be prevented by maintaining contact with the head throughout the stimulation session. This way, no kinetic energy can be transferred between the coil and the skull. For this reason, a safety spring system is placed between the robot and the coil. By continuously monitoring the deflection of the spring, and hence the contact force, the system can rapidly shut down and an impact can be prevented.

A low level of kinetic energy near the subject's skull is achieved by reducing the mass of the moving parts. Parallel manipulators with six-DOF have the potential for low kinetic energetic motion, since the motors are mounted on the base. Therefore, only the links and the end effector contribute to the moving mass. Parallel manipulators are known to have small workspaces, which generally leads to high stiffness, high accuracy and a small area in which a collision can occur.

Electrical and software safety is improved by using redundant sensors and an external watchdog system to monitor for controller failure or software lock-up.

A protective cover is placed over the robot manipulator to prevent either the subject or operator from reaching into the areas where a hand or finger can be crushed or become stuck.

B. TMS Coil

The robot is designed to connect to a wide range of commercial available TMS coils. Currently, the Magstim® Double 70mm Air Film Coil (The Magstim Company Limited, Whitlands, UK) is connected to the robot.

C. Spring system

A spring contact system is placed between the coil and the robot to allow soft interaction contact. This system acts as a safety switch, which only allows the robot to move at full speed while the contact force is within preset safety limits. The spring deflection measurement can also be used to control the contact force between the coil and the head.

The spring system uses two conical compression springs used in opposite compression directions to achieve a linear stiffness in the midrange and quadratic stiffness near the end of the range. In the mid-range, the stiffness is approximately 5.700 N/m, while near the ends the stiffness increases to

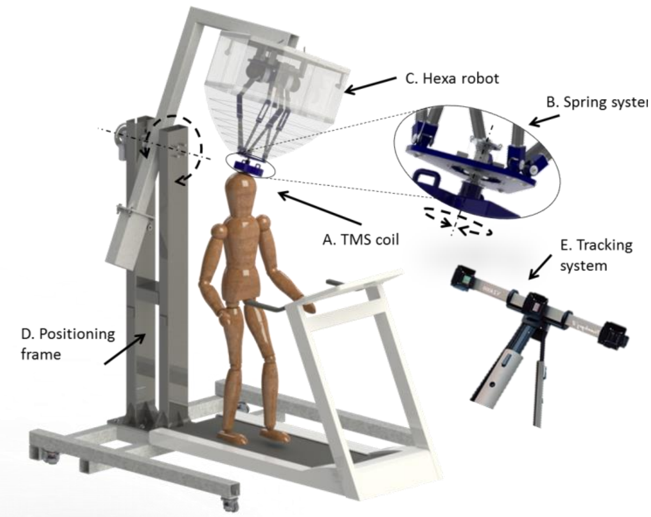


Figure 6 - System overview. Shown here are A) coil, B) the spring system, C) Hexa robot, D) positioning frame and the E) head tracking system.

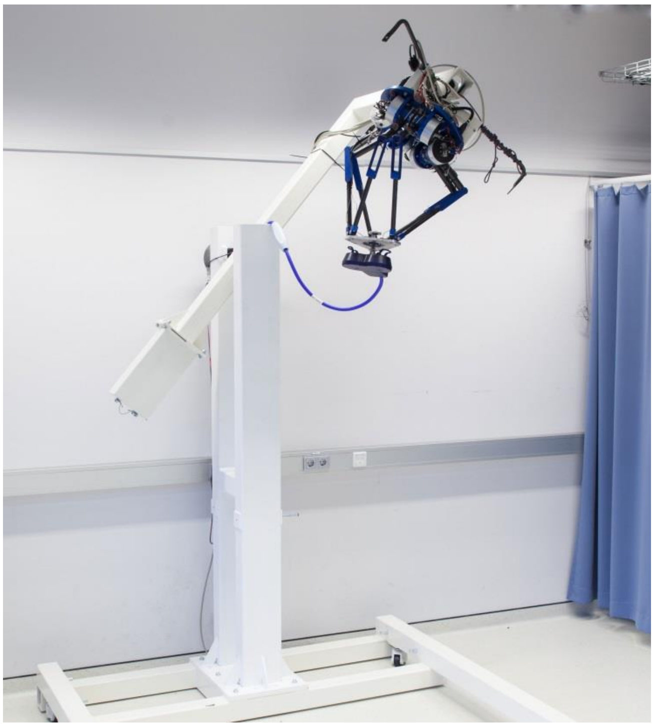


Figure 7 - System realization: The TMS coil (a), the spring system (b), Hexa robot (c), positioning frame (d) photographed without protective cover.

12.000 N/m. The spring compression – and with that the spring force – is measured by two draw wire sensors (UniMeasure ZX-HM and Celesco M150)

The orientation of the coil can be oriented around the axis perpendicular to the skull to allow optimal orientation of the magnetic field with respect to the brain structure.

D. *Hexa robot*

A Hexa robot is used to place the stimulator against the skull during treadmill walking. The Hexa robot is a six-DOF parallel robot in which the end effector is linked to the base by six identical <u>R</u>US kinematic chains. The actuators are located at the base to achieve a minimum moving mass. In [24] a quantitative comparison between the Stewart platform, Hexaglide [25] and the Hexa [26] was made to find the best six-DOF parallel manipulator for this application. The Hexa was chosen for this application as it allows a larger rotational range of motion than offered by the other mechanisms. In the same article [24], the results of the geometrical optimization for the Hexa in this application are described.

Six pancake motors (Printed Motor Works, GN9T) fitted with trochoidal gearboxes (Spinea TwinSpin, TS-60-35) are used to achieve an actuator torque of 15.5 Nm and a velocity of 500 deg/s. This results in an approximate velocity of 3 m/s and 1.200 deg/s, and force and torque of 100 N and 15 Nm respectively at the end-effector. Since the kinematics of the Hexa are non-linear, these values vary strongly within the workspace. Six high-resolution incremental encoders (AEDA-3300) are placed at the motor axes. They achieve a resolution of 1.3 10^{-4} deg at the joint output. This reflects to a 0.01 mm resolution at the end-effector. String potentiometers (Celesco SP3) are used on the output shafts to determine the absolute joint angles. These potentiometers are also used as redundant joint sensors. Each joint has mechanical and electrical end stop, which will prevents collision in case of a run-away motor. The connecting arms are made of carbon tubes with aluminum inserts, resulting in a mechanism with a mass less than 3 kg, excluding the coil. The upper arms and the lower arms are 0.2 m and 0.4 m long respectively.

E. *Support frame*

The Hexa robot can be moved over the stimulation site by a one-DOF arm. The axis of this arm passes through the center of the subject’s head. This allows the robot to be placed over all the stimulation sites and to accommodate for differences in subject lengths. The arm is actuated by a motor with worm gear (Parvalux, PM10MWS) placed on the axis. Switches on the frame allow motor control of this arm. Furthermore, the axis is excluded from the realtime controller because it designed to remain stationary during sessions. Moreover, a rotational potentiometer records the rotation of the axis. A counter-mass balances the arm, which reduces the power required to move the robot and thus increases its mechanical safety. The arm rotation is fixed by a mechanical brake at the output shaft of the motor.

F. *Motion capturing device*

The head position is measured by the Visualeyez Vz4000 (PTI Phoenix, Burnaby, BC, Canada) realtime motion-capturing device. This system uses active optical markers, which are placed, on both the robot and on the subject’s head. The system operates at a frequency of 100 Hz and obtains the marker position with an accuracy of 0.5 mm. The camera system communicates with a non-realtime PC, which sends the data through serial communication to the realtime controller. The lag in the data transfer has been empirically determined at 60 ms.

G. *Electronic safety system*

The robot is equipped with an external safety circuit, which monitors all the essential functions of the robot. The safety system enables motor power only if all the safety criteria are met. The safety criteria monitored by the safety system include the normal contact force, a watchdog to monitor the operation of the control system, electronic motor joint end-stop switches, emergency stop buttons and a light to signal that the system is in operation. The safety system has an autonomous power supply and is galvanically isolated to ensure complete independence from the main electrical circuitry.

IV. CONTROL

Safe, accurate and fast tracking of the head requires an online estimation of the stimulation site position from the 3D marker measurements. This position data is then to be used to control the robot to place the stimulator in this site. Furthermore the contact force data is to be used to improve the feel and safety of the robot. In Figure 8, the control structure to achieve this is presented. In this figure, three controller processes can be identified. The head-tracking block converts the data from the motion-capturing system into a desired robot pose. The robot pose is controlled by the position controller. The desired robot coordinates are modulated by the force controller to enable a safe interaction force between the subject and the robot.

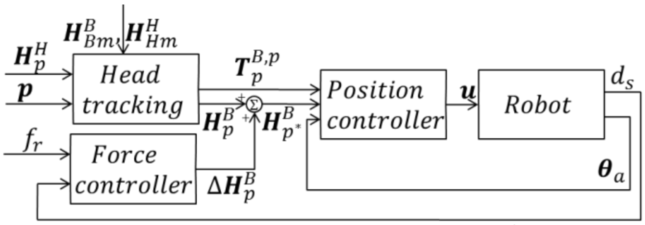


Figure 8 – The main controller diagram. The motion of the subject is measured by the camera, which measures the position ($\boldsymbol{p}^{\mathrm{Vz}}$) of the markers. The desired robot pose ($\boldsymbol{H}_{p}^{B}$) is calculated by head tracking from the measured marker position, offline registered marker frame locations ($\boldsymbol{H}_{Bm}^{B}$,$\boldsymbol{H}_{Hm}^{H}$) and pose of the robot relative to the stimulation site ($\boldsymbol{H}_{p}^{H}$). To attain a proper contact force (f_r) the force controller adds a correction pose ($\Delta\boldsymbol{H}_{p}^{B}$), based on the spring deflexion (d_s), to the desired end-effector pose ($\boldsymbol{H}_{p^*}^{B}$). The position controller steers the robot to the desired pose using the desired twist ($\boldsymbol{T}_{p}^{B,p}$), the joint angles ($\boldsymbol{\theta}_a$) and motor voltages ($\boldsymbol{u}$).

A. Head tracking

Figure 9 describes the calculation required to translate the measured marker position into a desired robot pose. Two frames are fitted through the markers placed on the subject's head and on the robot's base (M_p). How these two frames relate to the actual pose of the robot base and the stimulation site is registered beforehand. From this measurement, the required robot pose is calculated (M_h). However, the measurement noise has a strong effect on the desired pose of the robot. Therefore, the influence of the measurement noise is reduced by a first order tracking filter (F) with a cut-off frequency of 1 Hz.

1) Governing equations

In Figure 10 the transformations are shown which relate the end-effector pose ($\boldsymbol{H}_{P}^{B}$) to the subject head marker position ($\boldsymbol{H}_{Hm}^{Vz}$). There are six frames defined of which the relative transformation is to be measured or calculated in order to find the desired robot pose:

1. φ_{Vz}: Visualeyez reference frame
2. φ_{H}: Stimulation site
3. φ_{Hm}: Head-marker reference frame
4. φ_{B}: Base reference frame
5. φ_{Bm}: Base-marker reference frame
6. φ_{P}: End-effector reference frame

The transformations from one frame to another give rise to the following chain multiplication for calculation of the desired robot pose:

$$M_h\!: \boldsymbol{H}_{P}^{B}(t) = \boldsymbol{H}_{Bm}^{B}\boldsymbol{H}_{Vz}^{Bm}(t)\,\boldsymbol{H}_{Hm}^{Vz}(t)\boldsymbol{H}_{H}^{Hm}\boldsymbol{H}_{P}^{H}(t) \tag{1}$$

In this chain, the head marker frame ($\boldsymbol{H}_{Hm}^{Vz}$) and the robot marker frame ($\boldsymbol{H}_{Vz}^{Bm}$) are derived from the marker measurements. The local robot frame ($\boldsymbol{H}_{Bm}^{B}$) and the local frame at the stimulation site ($\boldsymbol{H}_{H}^{Hm}$) are given by the offline registration. This leaves the platform pose with respect to the stimulation site ($\boldsymbol{H}_{P}^{H}$) to be specified, based on the operating mode.

There are three tracking modes defined to control the motion of the robot. The first is a demo mode, whereas the second mode sets and holds the current relative pose between the robot and the stimulation site. The third mode steers the robot over the predefined stimulation site. The second and the third modes can be used for TMS stimulation.

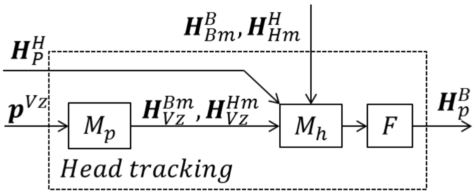

$\boldsymbol{H}_{Bm}^{B}, \boldsymbol{H}_{Hm}^{H}$
$\boldsymbol{H}_{P}^{H}$
$\boldsymbol{p}^{Vz}$
M_p
$\boldsymbol{H}_{Vz}^{Bm}, \boldsymbol{H}_{Vz}^{Hm}$
M_h
F
$\boldsymbol{H}_{p}^{B}$
Head tracking

Figure 9 – The head-tracking controller diagram. The motion of the subject $\boldsymbol{H}_{Vz}^{Hm}$ and the robots base $\boldsymbol{H}_{Vz}^{Bm}$ is calculated from the measured position $\boldsymbol{p}^{Vz}$ of the markers. This is called marker mapping (M_p). The desired pose ($\boldsymbol{H}_{p}^{B}$) of the robot is calculated in transformations mapping (M_h) using the robot ($\boldsymbol{H}_{Bm}^{B}$) subject ($\boldsymbol{H}_{H}^{Hm}$) registration, and desired position of the robot end effector with respect to the stimulation site ($\boldsymbol{H}_{P}^{H}$). A tracking filter (F) is added to reduce the measurement noise.

For demonstration purposes, it is desirable to let the end-effector of the robot make the same movement as the head at a safe distance from the subject. This means that the robot rotates and translates in the same manner as the head. $\boldsymbol{H}_{P}^{H}$ is therefore constant. To achieve such a motion, we defined an initial (set) frame. At the initial time (t_s), $\boldsymbol{H}_{P}^{H}(t_s)$ is calculated by inverting equation (1). As the accuracy is not the highest priority for a demonstration, the motion of the base frame is rejected. This results in:

$$\boldsymbol{H}_{P}^{B}(t) = \boldsymbol{H}_{Vz}^{B}\,(t_s)\boldsymbol{H}_{Hm}^{Vz}(t)\boldsymbol{H}_{Vz}^{Hm}(t_s)\boldsymbol{H}_{B}^{Vz}(t_s)\boldsymbol{H}_{P}^{B}(t_s) \tag{2}$$

in which:

$$\boldsymbol{H}_{Vz}^{B}(t_s) = \boldsymbol{H}_{Bm}^{B}\boldsymbol{H}_{Vz}^{Bm}(t_s) \tag{3}$$

The translational part of $\boldsymbol{H}_{Vz}^{B}(t_s)$, is set to zero to allow the movement to take place at a different position. Note that the registered stimulation site is no longer required as it only depends on the relative pose of the stimulator to the head marker site

For the second tracking mode, where the coil has to follow the set pose exactly, the $\boldsymbol{H}_{Vz}^{B}$ in (2) becomes time dependent. Opposed to the demonstration mode, the translational part in this matrix is now used.

In case we want to track a predetermined stimulation site, we have a slightly different transformation as in previous section. Again, we start with (1). Now the stimulation site with respect to the head markers ($\boldsymbol{H}_{H}^{Hm}$) is specified by the registration of the head (as described below). The relative pose of the stimulator to the head ($\boldsymbol{H}_{P}^{H}$) is specified depending on the operating mode. For example, in order to make contact with the subject, the robot is first steered over the stimulation site and then approaches the skull in a perpendicular fashion. During stimulation, the head pose and the stimulator pose have to be equal. This full tracking mode can be extended to include grid stimulation.

2) Registration

For the second and the third tracking modes, it is necessary to know where the robot and the subject are with respect to the tracking system. It especially important that the pose of the marker frames of with respect to the robot ($\boldsymbol{H}_{Bm}^{B}$) and to the subject ($\boldsymbol{H}_{Hm}^{H}$) is found. This is calculated during the registration part. Registration of landmarks on the robot and on the subject with respect to the markers is done with an optical probe. For the robot landmarks, the top and bottom corners of the base plate (${}^{rl}\boldsymbol{p}^{B}$) are used. These landmarks can be linked to a geometric model to find the origin of the robot. For the robot landmarks, the following relation holds:

$${}^{rl}\boldsymbol{p}^{B} = \boldsymbol{H}_{Bm}^{B}\,{}^{rl}\boldsymbol{p}^{Bm} \tag{4}$$

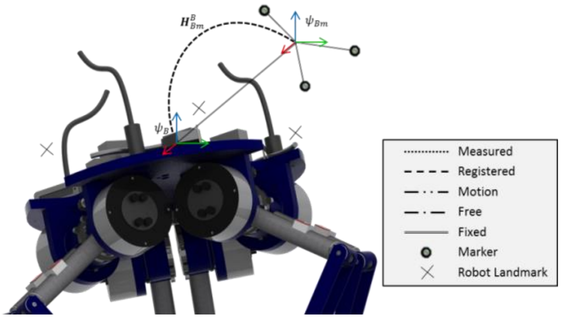


Figure 10 – Robot registration. The robot landmarks are shown here not on their exact location.

The “Procrustes method” (described below) can be used to find the corresponding transformation matrix. A similar method is used to link a 3D MRI image to a head such that the optimal stimulation site can be found. However, this has not yet been implemented. Therefore, the stimulation site and the surface of the skull are measured by use the probe data.

3) *Marker mapping*

During realtime tracking, a frame needs to be fitted through the markers that are placed on the robot base and on the subject’s head. This marker mapping (M_p) uses the “Procrustes” method to map a transformation between an initial point cloud and a moved point cloud, as described in [27] and [28]. This method results in a pose estimate, which minimizes the mapping error in the least squares sense.

The time-variable transformations - denoting the pose of the head markers ($\boldsymbol{H}_{Hm}^{Vz}(t)$) and the base markers ($\boldsymbol{H}_{Vz}^{Bm}(t)$) - need to be calculated. These are the head marker frame (φ^{Hm}) and base marker frame (φ^{Bm}), expressed in the camera frame (φ^{Vz}) at time$t$. The measurement of the i-th marker at time t is denoted as ${}^{i}\boldsymbol{p}^{Vz}(t)$. The centroid of the point cloud is denoted with an overbar. For brevity, this method is shown for the head marker frame only; the base marker frame is treated similarly.

This method uses an initial frame at $t = t_s$ to calculate the transformations.

$$\boldsymbol{H}_{Hm}^{Vz}(t) = \boldsymbol{H}_{s}^{Vz}\boldsymbol{H}_{Hm}^{s}(t) \tag{5}$$

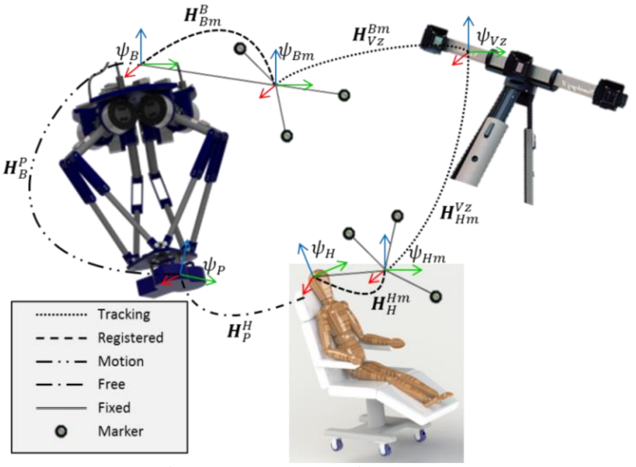


Figure 12 – Transformations required for head tracking. The motion capturing camera measures the pose of the markers on the robot ($\boldsymbol{H}_{Vz}^{Bm}$) and of the subject ($\boldsymbol{H}_{Hm}^{Vz}$). Prior registration provides the location of the markers with respect to the robot ($\boldsymbol{H}_{Bm}^{B}$) and to the stimulation site ($\boldsymbol{H}_{H}^{Hm}$). Finally the position of the robot with respect to head ($\boldsymbol{H}_{H}^{P}$) can be specified to calculate the desired robot position ($\boldsymbol{H}_{B}^{P}$)

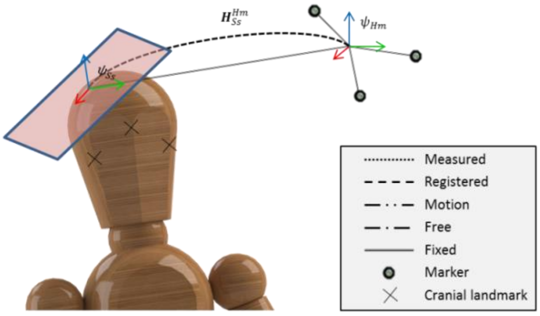


Figure 11 – Subject registration. Here a plane is defined trough the stimulation site to allow proper orientation of the coil. The landmarks on the head are show here.

The frames are fixed at the centroid of the markers. The initial frame is co-axial with the inertial frame:

$$\boldsymbol{H}_{s}^{Vz} = \begin{bmatrix} \boldsymbol{I}_3 & \overline{\boldsymbol{p}}^{Vz}(t_s) \\ 0 & 1 \end{bmatrix}, \boldsymbol{H}_{s}^{Vz} = \begin{bmatrix} \boldsymbol{R}_{Hm}^{s} & \boldsymbol{o}_{Hm}^{s} \\ 0 & 1 \end{bmatrix} \tag{6}$$

The time variant part $\boldsymbol{H}_{Hm}^{s}(t)$is calculated in five steps:

1. Calculate the distance to the centroid of the two point clouds:

$$\begin{aligned} {}^{i}\boldsymbol{X} &= {}^{i}\boldsymbol{p}^{Vz}(t_s) - \overline{\boldsymbol{p}}^{Vz}(t_s) \\ {}^{i}\boldsymbol{Y} &= {}^{i}\boldsymbol{p}^{Vz}(t) - \overline{\boldsymbol{p}}^{Vz}(t) \end{aligned} \tag{7}$$

2. Construct covariance matrix from distance matrices $\boldsymbol{X} = [{}^{1}\boldsymbol{X} \quad \dots \quad {}^{n}\boldsymbol{X}]$ and $\boldsymbol{Y} = [{}^{1}\boldsymbol{Y} \quad \dots \quad {}^{n}\boldsymbol{Y}]$:

$$\boldsymbol{S} = \boldsymbol{X}\boldsymbol{Y}^{T} \tag{8}$$

3. Using singular value decomposition of $\boldsymbol{S}$ to calculate $\boldsymbol{U}$ and $\boldsymbol{V}$:

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{T} \tag{9}$$

4. Compute the rotation matrix $\boldsymbol{R}_{Hm}^{s}$ from $\boldsymbol{U}$ and $\boldsymbol{V}$. The determinant is used to assure the rotation matrix is proper:

$$\boldsymbol{R}_{Hm}^{s} = \boldsymbol{V}\begin{bmatrix} \boldsymbol{I}_2 & 0 \\ 0 & det(\boldsymbol{V}\boldsymbol{U}^{T}) \end{bmatrix}\boldsymbol{U}^{T} \tag{10}$$

5. The translation part is calculated using identity

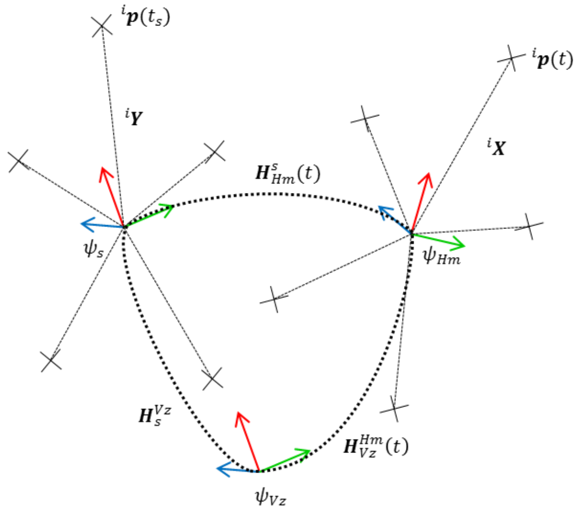

${}^{i}\boldsymbol{p}(t_s)$
${}^{i}\boldsymbol{p}(t)$
${}^{i}\boldsymbol{Y}$
${}^{i}\boldsymbol{X}$
$\boldsymbol{H}_{Hm}^{s}(t)$
ψ_s
ψ_{Hm}
$\boldsymbol{H}_{s}^{Vz}$
$\boldsymbol{H}_{Vz}^{Hm}(t)$
ψ_{Vz}

Figure 13 – The transformation of required for point cloud method. The two uniform but translated point clouds ($\boldsymbol{p}(t_s)$, $\boldsymbol{p}(t)$) at two time steps (t_s, t) are shown. Also, the required translations to express both points in the inertial frame (ψ_{Vz}), set frame (ψ_s) and head frame (ψ_{Hm}). The distance of the marker to the two centroids ($\boldsymbol{X}$, $\boldsymbol{Y}$) are shown as well.

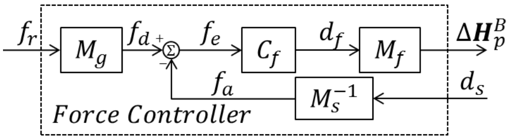


Figure 14 – The force control loop controls a pose offset ($\varDelta\boldsymbol{H}_p^B$) based on the input reference force (f_r) and the current spring deflection (d_s). It compensates for the influence of the gravity (M_g) to calculate the desired force (f_d). The actual spring force (f_a) is estimated using an inverted spring model (M_s). Based on the force error (f_e) the controller (C_f) calculates a deflection offset (d_f) which is mapped to the robots reference frame in M_f.

$$\boldsymbol{o}_{Hm}^s(t) = \overline{\boldsymbol{p}}^{Vz}(t) + (\boldsymbol{R}_{Hm}^s - \boldsymbol{I}_3)\boldsymbol{o}_s^{Vz} - \boldsymbol{R}_{Hm}^s\overline{\boldsymbol{p}}^{Vz}(t_s) \qquad (11)$$

Together, the rotation and translation parts form the time-dependent transformation matrix, which can be used with (5) to calculate the complete transformation matrix.

In the next chapter (Chapter 3), a comparison is made between Orthogonal Procrustes method and a method using Kalman filtering to find the method that most effectively reduces the influence of measurement noise.

B. Force controller

The contact force between the subject and the coil is controlled to ensure a constant contact force. However, the contact force itself cannot be measured directly as the strong magnetic field prevents the placement of a sensor between the subject and the coil. Therefore, the contact force is assumed equal to the spring force. For the low frequency range, this is assumption is valid. However, for the higher frequencies, the acceleration and therefore the inertia of the coil also influences the contact force.

The force controller, shown in Figure 15, adds a pose offset ($\Delta\boldsymbol{H}_p^B$) based on the spring displacement (d_s) and the platform orientation to obtain the reference contact force (f_r).

To obtain an orientation-independent contact force, the gravitational force of mass on the skull must be compensated for. This is done by adding an orientation-dependent gravitational compensation force to the reference force (f_r). This involves the mapping of a gravitational vector in global frame to the platform reference frame by a rotation matrix (${}_z\boldsymbol{R}_O^p$), which only consists of the z-component:

$$M_g\colon f_d = f_r + m_z\boldsymbol{R}_O^p\begin{bmatrix}0\\0\\-9.81\end{bmatrix} \qquad (12)$$

The contact force controller consists of a PI action on the force error (f_e):

$$C_f\colon\, d_f = \frac{k_{f,p}}{k_s}(1 + k_{f,i}\frac{1}{s})f_e \qquad (13)$$

The proportional and integral controller gains are $k_{f,p} = 0.5$, $k_{f,i} = 50$, and the spring constant is $k_s = 5700\ N/m$.

The resulting desired displacement (d_f) is translated by the kinematic mapping to an offset of the desired robot pose ($\varDelta\boldsymbol{H}_p^B$):

$$M_f\colon \varDelta\boldsymbol{H}_p^B = \begin{bmatrix}\boldsymbol{R}_p^B & 0\\0 & 1\end{bmatrix}\begin{bmatrix}0\\0\\d_f\\1\end{bmatrix} \qquad (14)$$

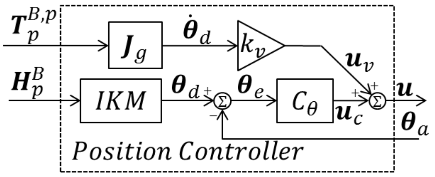


Figure 15 – The position controller calculates the required motor voltages ($\boldsymbol{u}$) based on the desired end-effector pose ($\boldsymbol{H}_p^B$), the desired end-effector twist ($\boldsymbol{T}_p^{B,p}$) and the current joint angles ($\boldsymbol{\theta}_a$). The inverse kinematic model (IKM) gives de desired joint angles ($\boldsymbol{\theta}_d$). The position controller (C_θ) gives an error ($\boldsymbol{\theta}_e$) dependent controller voltage ($\boldsymbol{u}_c$,). The velocity mapping ($\boldsymbol{J}_g$) and the velocity ($\dot{\boldsymbol{\theta}}_d$) feed-forward gain ($k_v$) compensates for the gear friction ($\boldsymbol{u}_v$).

C. Position controller

In Figure 15, the layout of the controller is given. The robot pose is controlled by six identical joint-space PID controllers. The controller values are tuned to achieve a bandwidth of 15 Hz.

The inverse kinematic model (IKM) calculates the desired joint angles ($\boldsymbol{\theta}_d$) as a function of the desired robot pose ($\boldsymbol{H}_p^B$):

$$\boldsymbol{\theta}_d = IKM\left(\boldsymbol{H}_p^B\right) \qquad (15)$$

A velocity feed-forward ($\boldsymbol{u}_v$) is used to compensate for the counter electromotive force and friction effects in the motor and the gear:

$$M_v\colon \boldsymbol{u}_v = k_v\dot{\boldsymbol{\theta}}_d, \qquad \dot{\boldsymbol{\theta}}_d = \boldsymbol{J}_g\boldsymbol{T}_p^{B,p} \qquad (16)$$

This friction compensation (M_v) is dependent on the joint velocities and are calculated from the desired end effector twist ($\boldsymbol{T}_p^{B,p}$) and the global coordinate Jacobian ($\boldsymbol{J}_g$). For the compensation, an empirically determined gain (k_v) suffices.

V. EVALUATION

The system is evaluated starting from low-level performance assessment up to functional evaluation of the head tracking. First, the kinematic model and the pose controller are evaluated. Secondly, the force controller and thirdly the head tracking are evaluated. The evaluation based on TMS stimulation is not conducted in this paper, since the neurological response to TMS has a large variation and is therefore a poor measure of the robot accuracy.

A. Kinematic model and Position controller

To validate the kinematic model and the accuracy of the position controller, a box-shaped trajectory is followed by the robot. A minimal-jerk motion profile of 0.4 s over each edge of 0.16 m is used. This motion reaches a maximum velocity of 1.5 m/s and an maximal acceleration of 40 m/s^2. Accuracy of tracking is determined by comparing the resulting robot pose, as calculated from kinematic model, to the pose measurement by Visualeyez, the optical motion-capturing device.

During this box-motion, it can be seen that the accuracy in the perpendicular direction is +/- 1 mm. These inaccuracies are mainly caused by the calibration. The contribution of the controller error is about +/-0.5 mm in the motion direction.

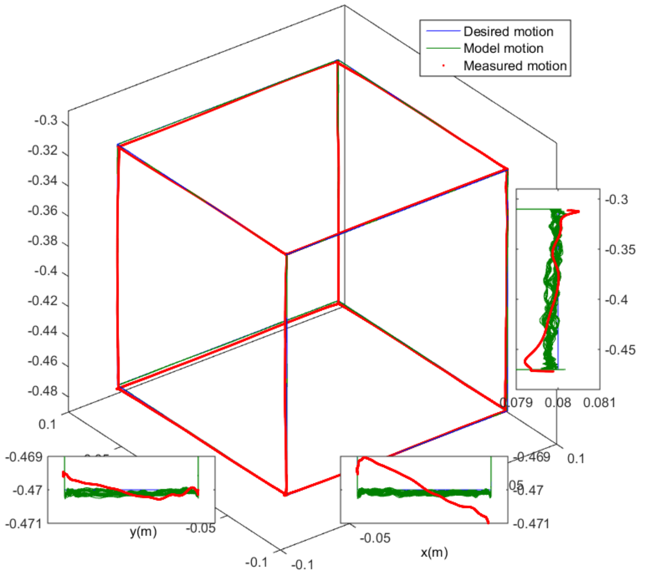


Figure 16 – Box motion of the end effector with respect to the base. The box-motion has sides of 0.16 m and a travel time of 0.4s over each segment. The maximal robot velocity is 1.5 m/s. The desired pose, the pose of the internal model of the robot, and the measured pose are shown in blue red green respectively. Inserts show the accuracy of motion in the y, x and z direction.

The dynamic performance and the crosstalk of the robot are determined by letting the robot follow a swept sine motion in each translation and rotation direction. The trajectory is an exponential swept sine motion from 0.01 Hz to 15 Hz, with an amplitude of 5 mm and 2 deg. These values are chosen such that the amplitudes of the motor motions are comparable. A payload of 2.5 kg is attached to the platform to simulate the load of the stimulator coil.

Figure 18 shows the tracking performance in the worst direction (x) and in the best direction (z) in terms of accuracy. It can be seen that the system follows the desired trajectory up to 15 Hz. It can be observed that there is some stick-slip present, limiting the low velocity motion. This results in incomplete tracking of the signal. Crosstalk can be observed between the input motion in one direction and an undesired motion in the other direction. This tracking error is less than +/- 0.5 mm and +/- 0.1 deg.

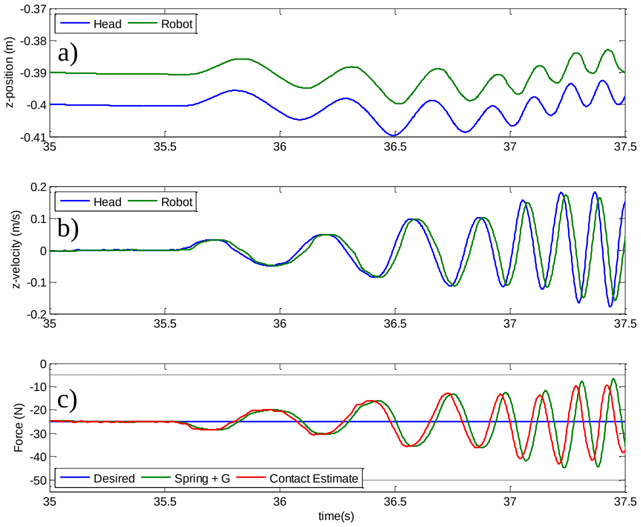


Figure 18 – Evaluation of the spring force controller. Top figure (a) shows the position of the head and the position of the robot. Middle figure (b) shows the velocity of the head and the robot. Bottom figure (c) shows the desired force, the measured spring force, and the estimated contact force.

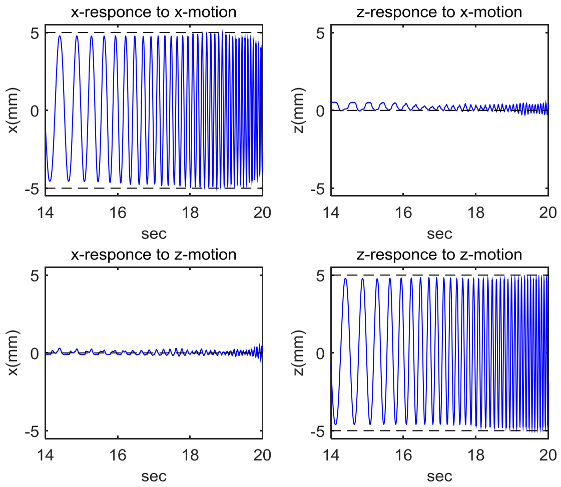


Figure 17 – Response of the system to a 5-mm chirp up to 15 Hz in respectively the x and the z direction. Here the parasitic motion in the other direction is also shown.

B. Spring force controller

The spring force controller is evaluated by imposing a manually generated approximate swept-sine motion type. The spring force and the position of the robot and the head are measured. The contact force is estimated by adding gravitational and inertial forces to the spring force. The motion generated has an amplitude of 5 mm and a velocity up to 0.2 m/s. The spring force is tested without the use of head-tracking to discriminate clearly between the two effects

Figure 17 shows that the robot follows the head at an approximate distance of 1 cm. Furthermore, it can be seen that the contact force achieves the desired force during low velocity motion. At higher frequencies, the contact force variation increases. During the higher frequency motion, the contact force will start leading the spring force due to the added inertial forces. As the input motion approaches the anti-resonance frequency of the system of 7 Hz, the amplitude of the contact force becomes lower than the spring force. In chapter Chapter 6.I. a study is done on the design and performance of the spring force controller. The study shows that force controller can achieve a small bandwidth, since its stability will be compromised by the phase lag of the position controller and the rest of the system.

Figure 17 shows that even during large and fast motions, the contact force stays within the safety bounds of 5 to 50 N. This shows that under the bandwidth limits imposed on controller, sufficient spring force compensation can be achieved.

C. Head tracking

To evaluate the head tracking abilities of the robot, the head is tracked during an alternating slow and fast motion. Here also the force controller is not used to show clearly the effect of the head tracking. The measurements of the head pose are subject to noise, as can be seen in Figure 19-a. The RMS value of pose measurement is 3.5 mm and 0.7 deg. To calculate the tracking error of the robot, the head pose has to be estimated from the measurement data. This is done by offline filtering of the data using a zero-phase filtering paradigm with a cut-off frequency of 2 Hz. This is the post-hoc filtered reference trajectory of Figure 19-left.

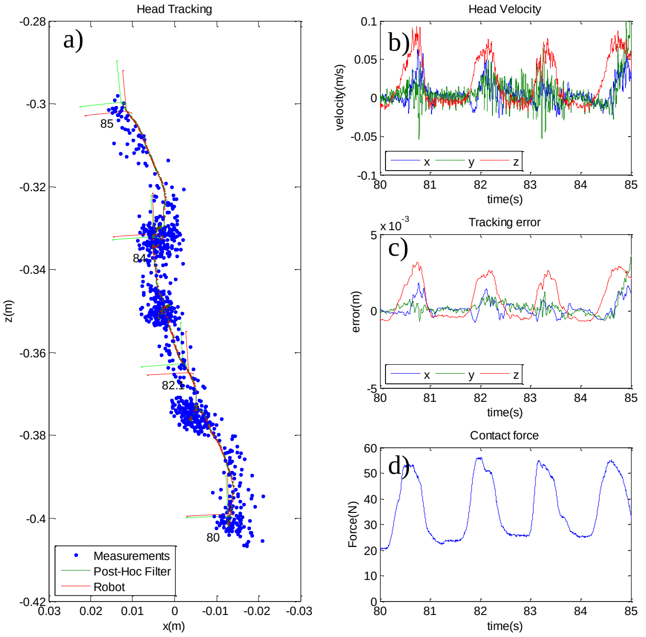


Figure 19 – Evaluation of the head tracking. Subfigure a) shows the measured position of the head. The actual position of the robot and the post hoc head position calculation. b) position shows the velocity of the robot in 3 directions. c) shows the error of tracking. d) shows the resulting contact force.

Figure 19-a shows that the robot follows the head trajectory. The orientation – as shown with the axis system at different times – corresponds with an accuracy of 1 deg. During standstill, the error is less than 0.5 mm as can be seen in Figure 19-c. However, at higher velocities the error increases accordingly. This error in the z-direction results in a compression of the spring with an associated change in contact force. The frequency dependency of the tracking error can be explained by the fact that an input filter with a cutoff frequency of 1 Hz is required to reduce the influence of the measurement noise. Otherwise, this noise will apply a very unpleasant vibration onto the patient's head.

VI. Discussion

The TMS robot is functional and is able to track a subject's head during slow movement. The tracking during faster motion is limited by the input filter, as the marker noise induces vibrations of the end-effector. The tracking quality and the noise suppression can be improved by the use of a Kalman filter, and can further be improved by attaching an IMU on the subject's head.

The addition of the spring force controller results in a more constant contact force. However, its bandwidth is limited by the natural frequency of the spring system and the response of the Hexa manipulator. With an accelerometer placed on the coil, the controller can reduce the influence of the inertia of the coil on the contact force.

The spring contact allows a smooth motion in the perpendicular direction. In the other directions, the friction force of the contact hinders the head movement. This can be solved by lowering the contact force, but also by implementing an elastic element in the tangential directions.

The position controller has been shown to have sufficient bandwidth to achieve the accuracy required for high velocity movement. The inverse kinematic model induces a small error since the geometric model does not correspond fully with the actual robot. This could be improved further by a more accurate calibration of the system.

VII. Conclusion

In this article, the design of a TMS robot for treadmill tracking has been presented. The robot is designed to evaluate human motor adaptation during treadmill walking and improve the motor recovery after stroke using TMS.

The system places the stimulation coil against the skull by the use of an elastic element. This will ensure soft contact and allow a timely shutdown in case the contact is about to be lost, or the contact force becomes too large. A six-DOF parallel manipulator, the Hexa, places the coil against the stimulation site. The motors are located at the base to reduce the moving mass and energy in the system. A frame supports the robot and allows adjustment to suit various subjects and stimulation sites. A 3D motion capturing system is used to measure the pose of the subject's head.

In this article, the transformations and controller strategies, which relate the measurements of the head position to a translation of the robot, are presented. The head-tracking system uses an orthogonal Procrustes method to fit frames through the marker measurements. These frames are related to the position of the stimulation site, with respect to the robot. The position controller steers the robot to this position. The spring force controller augments this position signal to reduce the contact force variations.

The system has been fully built and tested. The mechanism and controller operate satisfactorily for fast motion. However, an input filter is required to reduce the influence of measurement noise on the robot pose. As a result, the bandwidth of the system is significantly limited. Further improvements to the head position measurement, head pose estimation and filtering need to be made before the system can successfully track subjects and apply TMS safely during treadmill walking.

Chapter 3.Comparison between two realtime tracking methods for robotized TMS

Ir. Jan. J. de Jong, Ir. Wietse van Dijk, Prof. dr. ir. Herman van der Kooij,
Dr. ir. Arno H. A. Stienen

***Abstract*— Robotic TMS stimulation is used to investigate the connectivity with in the brain and help motor training via stimulation during activities. A novel robot has been developed for walking training on treadmill. The robot optically measures the motion of the head in realtime and place the stimulator with millimeter accuracies during motor. For accurate and comfortable tracking, two methods of calculating the head pose from the marker data compared here. The first one uses the principle component analysis of the marker point cloud. The second relies on a Kalman filter to track each marker and update a model of the head motion. Both methods result in accurate tracking, while the first method is more susceptible to marker noise and initial bias, while the Kalman filter reduces the noise significantly at cost of increased complexity and calculation times.**

***Index Terms*—Medical robots, transcranial magnetic stimulation (TMS), motion tracking.**

I. Introduction

Transcranial magnetic stimulation (TMS) is a non-invasive tool, used to investigate the brain behavior and help with recovery of several neurological pathologies. These pathologies include depression, Parkinson’s disease and stroke. The TMS pulse is generated by running large currents trough a magnetic coil. These electromagnetic pulses induce electric currents in the brain, which lead to modified brain behavior. TMS has been used to improve post-stroke training for motor relearning [3]. To be able to apply TMS during treadmill motor training a novel TMS robot design is presented in Chapter 2.

The robot uses an optical motion capturing system [42] to measure the motion of the head. From this data, the desired pose of the robot is calculated to place the stimulator against the prespecified area of the head. This system relies on measuring two marker frames; one fixed to the head and one to the robot. From the measurement of these markers, the relative position and orientation of the head to the robots base has to be extracted. The aim of this paper is to find the most effective, marker-tracking algorithm. This method has to reduce the influence of measurement noise, realtime and accurate within 1 mm.

Notation	Meaning
$\boldsymbol{a}$	Vector, can also be list with vectors as rows
${}_{x}a$	The x-dimension (column) of vector $\boldsymbol{a}$
${}^{i}\boldsymbol{a}$	The i-th point (row) of $\boldsymbol{a}$
$\times$	Cross product
$[\boldsymbol{a}\times]$	Skew symmetric or semi-skew symmetric form of the vector
$\otimes$	Quaternion product
$[\boldsymbol{q}\otimes]$, $[\boldsymbol{q}\otimes]^T$	Quaternion product in matrix form. The left and right multiplication.
$\breve{\boldsymbol{a}}$	Appended vector with a 1 or a 0
$\overline{\boldsymbol{a}}$	Averaged vector over the rows
$\widehat{\boldsymbol{a}}$	Estimation of $\boldsymbol{a}$

Table 3. Notation used in this article

The measurement setup uses a non-realtime computer to readout the measured marker data from the camera system. This data is send to the realtime controller to compute the require transformation to steer the robot over the stimulation site. The non-realtime tracking system operates at approximately 100 Hz while the realtime controller operates at 1 kHz. This non-realtime system induces a variable delay of 10-30ms. The camera system can reach a RMS accuracy value for each marker of 0.1 mm in the center of field of vision. Near the edges, the marker noise increases. In addition to the measurement noise, some measurement artifacts are observed. Sometimes the markers are presented mirrored or translated by several mm up to multiple meters. Tracking method can also be hindered by the fact that not all markers are visible all the time [42]. Together with the non-realtime nature, rate transition and measurement and quantification noise, the head-tracking algorithm has to be designed to achieve accurate tracking of 1 mm and smoothness for pleasant interaction.

In this chapter, two methods are compared to track this motion in realtime. The first method is the orthogonal Procrustes method (OP) [41]. It uses the singular value decomposition (SVD) of the displacement vector to find the optimal rotation and translation in the minimal square sense. Sometimes it is also called the Kabsch method or point cloud method. This method is used by another TMS robot design [16]. As this method only uses the current measurement, it is expected that the noise will be propagated to the end-effector placement. The second

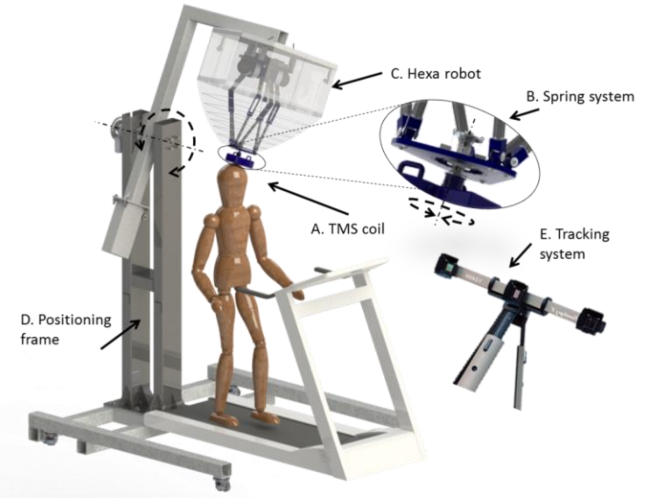


Figure 20 The TMS robot setup with all the auxiliary apparatus such as the head tracking mechanism and controller unit.

method uses a modified Kalman filter (MKF) [43] to update a model of the head motion.

In this article the mathematical notations are be introduced. Then the relations between reference frames are given. The two tracking method are expanded and a compared based on artificial and measured head movements. The realtime testing of the OP method is performed and presented in Chapter 2. The realtime testing of the Kalman filter is not performed yet.

II. Notation

With each rigid body a reference frame (ψ_i) is associated. This reference frame gives an expression for points attached to this frame with respect to other reference frames. When this point is expressed in the reference frame, (i.e. ψ_1 in Figure 21) we denote it by a superscript, i.e. $\boldsymbol{p}^1$. Given a second frame (ψ_2) we can express the previous point in ψ_2 using the rotation matrices and translation vector. Here the check represents the same vector appended with a one for calculus propagation.

$$\breve{\boldsymbol{p}}^2 = \boldsymbol{H}_1^2 \breve{\boldsymbol{p}}^1, \qquad \breve{\boldsymbol{p}}^1 = \begin{bmatrix} \boldsymbol{p}^1 \\ 1 \end{bmatrix}, \qquad \boldsymbol{H}_1^2 = \begin{bmatrix} \boldsymbol{R}_1^2 & \boldsymbol{o}_1^2 \\ 0 & 1 \end{bmatrix} \tag{17}$$

Multiple points are distinguished by a pre-superscript (${}^1\boldsymbol{p}^i$), meaning the first marker expressed in frame i. The different dimensions of each point or matrix is specified with a pre subscript (${}_x\boldsymbol{p}^i$), meaning the length of vector in the x-direction

The relative velocities of points can be expressed in a frame using the twist of the frame to which the point is rigidly attached. The twist is the generalized velocity of a rigid body. It is expressed by the concatenation of the angular velocity ($\boldsymbol{\omega}$) and translational velocity ($\boldsymbol{u}$). $\boldsymbol{v}_1^{3,2}$ is the twist of frame 1 (ψ_1) relative to frame 2 (ψ_2) expressed (seen) by frame 3 (ψ_3)

$$\boldsymbol{v}_1^{3,2} = \begin{bmatrix} \boldsymbol{\omega}_1^{3,2} \\ \boldsymbol{u}_1^{3,2} \end{bmatrix} \tag{18}$$

The derivative of equation (17) gives velocity of the point as seen by frame 2.

$$\dot{\breve{\boldsymbol{p}}}^2 = \dot{\boldsymbol{H}}_1^2 \breve{\boldsymbol{p}}^1 = \dot{\boldsymbol{H}}_1^2 \boldsymbol{H}_2^1 \breve{\boldsymbol{p}}^2 = [\boldsymbol{v}_1^{2,2} \times] \breve{\boldsymbol{p}}^2 \tag{19}$$

Here the boxed cross denotes a semi-skew symmetric notation, a combination of a skew symmetric angular velocity matrix ($[\boldsymbol{\omega} \times]$) and the velocity vector.

$$[\boldsymbol{v} \times] = \begin{bmatrix} [\boldsymbol{\omega} \times] & \boldsymbol{u} \\ 0 & 0 \end{bmatrix} \tag{20}$$

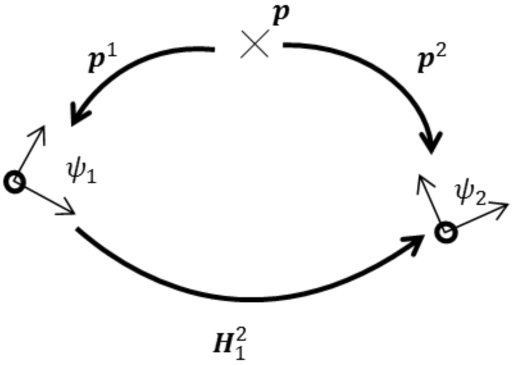


Figure 21. Notation for expression of a point $\boldsymbol{p}$ in two different frames.

The Kalman filter uses unit quaternions to express the rotation in minimal singularity free coordinates. These quaternions are concatenation of a vector part (${}_l\boldsymbol{q}$) and a scalar part (${}_sq$).

$$\boldsymbol{q} = \begin{bmatrix} {}_l\boldsymbol{q} \\ {}_sq \end{bmatrix} = \begin{bmatrix} {}_xq \\ {}_yq \\ {}_zq \\ {}_sq \end{bmatrix} \tag{21}$$

The result of multiple subsequent rotations can be computed from the quaternion product ($\otimes$). In this paper, the matrix formulation of the product is used.

$$\boldsymbol{q}_1^3 = \boldsymbol{q}_2^3 \otimes \boldsymbol{q}_1^2 = [\boldsymbol{q}_2^3 \otimes] \boldsymbol{q}_1^2 = [\boldsymbol{q}_1^2 \otimes]^T \boldsymbol{q}_2^3 \tag{22}$$

These are the two matrix forms, left and right, of quaternion multiplication.

$$\begin{aligned} [\boldsymbol{q} \otimes] &= \begin{bmatrix} [{}_l\boldsymbol{q} \times] & {}_l\boldsymbol{q} \\ -{}_l\boldsymbol{q}^T & 0 \end{bmatrix} + {}_sq\boldsymbol{I}_4 \\ [\boldsymbol{q} \otimes]^T &= \begin{bmatrix} [{}_l\boldsymbol{q} \times]^T & {}_l\boldsymbol{q} \\ -{}_l\boldsymbol{q}^T & 0 \end{bmatrix} + {}_sq\boldsymbol{I}_4 \end{aligned} \tag{23}$$

It should be observed that the (semi-) transpose of this matrix is in fact only a transpose of the skew symmetric part.

Rotation matrices can also be computed from the quaternion.

$$\boldsymbol{R} = ({}_sq^2 - {}_l\boldsymbol{q}^T {}_l\boldsymbol{q})\boldsymbol{I}_3 + 2{}_l\boldsymbol{q}{}_l\boldsymbol{q}^T + 2{}_sq[{}_l\boldsymbol{q} \times] \tag{24}$$

The derivative of the quaternion (the quaternion rate) is the quaternion product of the angular velocity. The accent ($\breve{\boldsymbol{\omega}}$) indicate the extended angular velocity with a zero to make it a quaternion.

$$\dot{\boldsymbol{q}} = \frac{1}{2} \breve{\boldsymbol{\omega}} \otimes \boldsymbol{q} \tag{25}$$

The evolution of quaternions over a certain time with a constant angular velocity can be given from the exponential mapping.

$$\begin{aligned} \boldsymbol{q}(t) &= e^{t\breve{\boldsymbol{\omega}}} \otimes \boldsymbol{q}(0) \\ &= \begin{bmatrix} 1/\theta \sin(t\theta/2)\boldsymbol{\omega} \\ \cos(t\theta/2) \end{bmatrix} \otimes \boldsymbol{q}(0) \end{aligned} \tag{26}$$

With $\theta = \|\boldsymbol{\omega}\|$

To denote all the different operations done on vectors and matrices, a list of notations with the meanings is given in Table 3.

A. Transformations

For realtime head tracking, markers are placed on the robot and on the head to calculate the pose of the head with respect to the pose of the robots base. From this, the desired pose of the robot can be calculated such that the robot can be steered over the stimulation site. In Figure 22 the calculation of the desired robot pose is visualized. This requires an offline registration step to calculate where the markers are placed on the head with respect to stimulation site ($\boldsymbol{H}_H^{Hm}$) and similarly where the markers are placed at the robot base ($\boldsymbol{H}_{Bm}^B$). During realtime tracking the pose of the stimulation site ($\boldsymbol{H}_H^{Vz}(\mathrm{t})$) can be calculated from the measured head marker pose ($\boldsymbol{H}_{Hm}^{Vz}(t)$) and the offline registration.

$$\boldsymbol{H}_H^{Vz}(\mathrm{t}) = \boldsymbol{H}_{Hm}^{Vz}(\mathrm{t}) \boldsymbol{H}_H^{Hm} \tag{27}$$

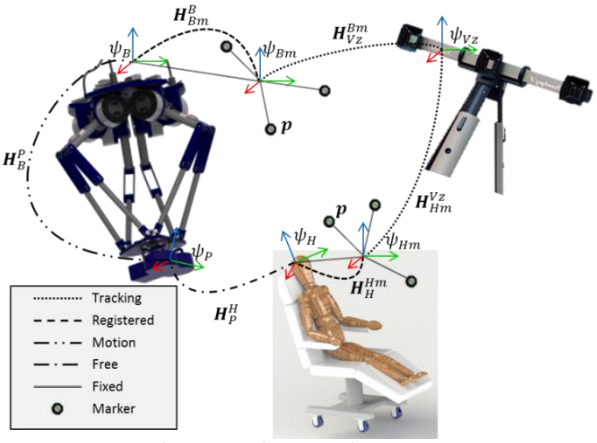


Figure 22 The transformation of required for head tracking

The stimulation site can be related to the base of the robot ($\boldsymbol{H}_H^B(\mathrm{t})$).

$$\boldsymbol{H}_H^B(\mathrm{t}) = \boldsymbol{H}_{Bm}^B \boldsymbol{H}_{Vz}^{Bm}(\mathrm{t}) \boldsymbol{H}_{Hm}^{Vz}(\mathrm{t}) \boldsymbol{H}_H^{Hm} \tag{28}$$

Now in order to calculate the desired robot pose ($\boldsymbol{H}_P^B(\mathrm{t})$) only the relative pose of the stimulator to the stimulation site ($\boldsymbol{H}_P^H(\mathrm{t})$) has to be given. This is depending on the operation modus of the robot. For example, before contact is made the robot should approach the stimulation site perpendicular skull.

$$\begin{aligned} \boldsymbol{H}_P^B(\mathrm{t}) &= \boldsymbol{H}_H^B(\mathrm{t}) \boldsymbol{H}_P^H(\mathrm{t}) \\ &= \boldsymbol{H}_{Bm}^B \boldsymbol{H}_{Vz}^{Bm}(\mathrm{t}) \boldsymbol{H}_{Hm}^{Vz}(\mathrm{t}) \boldsymbol{H}_H^{Hm} \boldsymbol{H}_P^H(\mathrm{t}) \end{aligned} \tag{29}$$

This will be send to the robot position controller in order to achieve the required position with respect to the stimulation site.

The topic of this article is to find the best method for calculation of the transformations from the marker measurements.

III. Methods

To find the time varying matrices $\boldsymbol{H}_{Vz}^{Bm}(\mathrm{t})$ and $\boldsymbol{H}_{Hm}^{Vz}(\mathrm{t})$, the two tracking methods are expanded here.

A. Orthogonal Procrustes method

The orthogonal Procrustes method uses a singular value decomposition of the translation of the markers relative to the centroid to find the rotation. It relates two point clouds with a transformation matrix and assumes the point clouds have only translated and rotated, such that the points are not moved with respect to each other.

For the calculation, this method is only shown for the head marker frame. For the base frame, the method can be treated similar.

This method uses an initial (set) frame (ψ_s) at t_s to start calculating the relative translations from. The initial frame $\boldsymbol{H}_s^{Vz}$ is calculated during offline registration of the stimulations site, and is therefore fixed for duration of the tracking. The relative head pose $\boldsymbol{H}_{Hm}^s(t)$ is time variant

$$\boldsymbol{H}_{Hm}^{Vz}(t) = \boldsymbol{H}_s^{Vz} \boldsymbol{H}_{Hm}^s(t) \tag{30}$$

Both frames are fixed at the centroid of the markers. The initial frame is co-axial with the inertial frame.

$$\boldsymbol{o}_s^{Vz} = \overline{\boldsymbol{p}}^{Vz}(t_s), \qquad \boldsymbol{R}_s^{Vz} = \boldsymbol{I}_3 \tag{31}$$

The time variant part $\boldsymbol{H}_{Hm}^s(t)$is calculated in five steps.

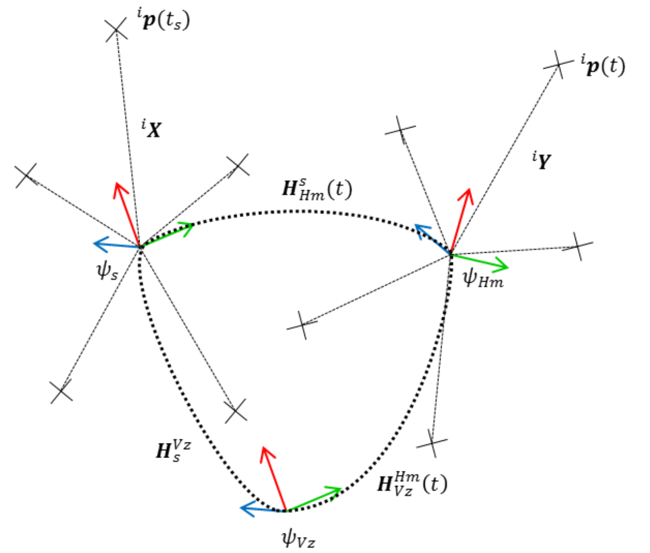


Figure 23 The orthogonal Procrustes method. A time varying transformation matrix ($\boldsymbol{H}_{Hm}^s$) is fitted to describe the translation and rotation during the motion of the markers. The frame at set time (t_s) is ψ_s at and ψ_{Hm} at the current instant (t). The distances to the centroids is $\boldsymbol{X}$ and $\boldsymbol{Y}$.

1. Calculate the distance to the centroid of the two point clouds.

$$\begin{aligned} {}^i\boldsymbol{X} &= {}^i\boldsymbol{p}^{Vz}(t_s) - \overline{\boldsymbol{p}}^{Vz}(t_s), \\ {}^i\boldsymbol{Y} &= {}^i\boldsymbol{p}^{Vz}(t) - \overline{\boldsymbol{p}}^{Vz}(t) \end{aligned} \tag{32}$$

2. Construct covariance matrix from distance matrices. $\boldsymbol{X} = [{}^1\boldsymbol{X} \quad \dots \quad {}^n\boldsymbol{X}]$ and $\boldsymbol{Y} = [{}^1\boldsymbol{Y} \quad \dots \quad {}^n\boldsymbol{Y}]$

$$\boldsymbol{S} = \boldsymbol{X}\boldsymbol{Y}^T \tag{33}$$

3. Using singular value decomposition of $\boldsymbol{S}$ to calculate $\boldsymbol{U}$ and $\boldsymbol{V}$.

$$\boldsymbol{S} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\boldsymbol{T}} \tag{34}$$

4. Compute the rotation matrix $\boldsymbol{R}_{Hm}^s$ from U and V. The determinant is used to assure that the rotation matrix is proper.

$$\boldsymbol{R}_{Hm}^s(t) = \boldsymbol{V} \begin{bmatrix} \boldsymbol{I}_2 & 0 \\ 0 & det(\boldsymbol{V}\boldsymbol{U}^T) \end{bmatrix} \boldsymbol{U}^{\boldsymbol{T}} \tag{35}$$

5. The translation part is calculated using identity

$$\boldsymbol{o}_{Hm}^s(t) = \overline{\boldsymbol{p}}^{Vz}(t) + (\boldsymbol{R}_{Hm}^s - \boldsymbol{I}_3)\boldsymbol{o}_s^{Vz} - \boldsymbol{R}_{Hm}^s \overline{\boldsymbol{p}}^{Vz}(t_s) \tag{36}$$

Together the rotation and translation part form the time dependent transformation matrix.

$$\boldsymbol{H}_s^{Vz} = \begin{bmatrix} \boldsymbol{I}_3 & \overline{\boldsymbol{p}}^{Vz}(t_s) \\ 0 & 1 \end{bmatrix}, \boldsymbol{H}_{Hm}^s(t) = \begin{bmatrix} \boldsymbol{R}_{Hm}^s & \boldsymbol{o}_{Hm}^s \\ 0 & 1 \end{bmatrix} \tag{37}$$

Sometimes one or more markers are not observed due to occlusion of that marker. In order to continue and find the proper transformations, this marker is omitted from the calculation in equation (32) and (36). The invisible markers are omitted in the setting cloud and in the current cloud. Note that for the calculation of (31), the complete set of initial measurements has to be used.

Omitting of a marker from the set of markers will have no effect on the calculated transformation if there is no noise. Recognize that the point clouds, expressed in the local frames, are then equal:

$$\boldsymbol{p}^{s}(t_s) = \boldsymbol{p}^{Hm}(t) \tag{38}$$

Transformation to the global frame and expansion relates the measurements at the two instances ($\boldsymbol{p}^{Vz}(t_s)$ and $\boldsymbol{p}^{Vz}(t)$)

$$\begin{aligned} \boldsymbol{H}_{Vz}^{s}\breve{\boldsymbol{p}}^{Vz}(t_s) &= \boldsymbol{H}_{Vz}^{Hm}\breve{\boldsymbol{p}}^{Vz}(t) \\ \boldsymbol{H}_{Vz}^{s}\breve{\boldsymbol{p}}^{Vz}(t_s) &= \boldsymbol{H}_{s}^{Hm}\boldsymbol{H}_{Vz}^{s}\breve{\boldsymbol{p}}^{Vz}(t) \\ \boldsymbol{H}_{s}^{Hm}\boldsymbol{H}_{Vz}^{s}\breve{\boldsymbol{p}}^{Vz}(t_s) &= \boldsymbol{H}_{Vz}^{s}\breve{\boldsymbol{p}}^{Vz}(t) \end{aligned} \tag{39}$$

Using (31) this relation can be decomposed into o combination of a rotation matrix and translation vectors.

$$\boldsymbol{R}_{Hm}^{s}(\boldsymbol{p}^{Vz}(t_s) + \boldsymbol{o}_{Vz}^{s}) + \boldsymbol{o}_{Hm}^{s} = \boldsymbol{p}^{Vz}(t) + \boldsymbol{o}_{Vz}^{s} \tag{40}$$

Together with proving relation (36), it also shows that any set of combinations of points from both clouds can be used to calculate the relate translation between these clouds. The average is chosen, as it will reduce measurement noise. If the number of visible markers is less than three, this method cannot be used and the robot should stop.

B. Kalman method

The extended Kalman filter is widely used as a state estimator. It recursively models the noise on the measurements and on the system, in order to find the optimal estimation of the state.

A typical extended Kalman filter consist of a prediction phase, in which the next state is predicted from the current state, and a measurement update phase in which the measurement is used to correct the predicted state. The prediction and measurement update are dependent on a model of the noise propagation.

The state and measurement equation consist of:

$$\begin{aligned} \boldsymbol{x}(k+1) &= F\big(\boldsymbol{x}(k)\big) + \boldsymbol{w}(k) \\ \boldsymbol{z}(k) &= B\big(\boldsymbol{x}(k)\big) + \boldsymbol{m}(k) \\ \boldsymbol{y} &= Y(\boldsymbol{x}(k)) \end{aligned} \tag{41}$$

Here $\boldsymbol{x}$ is the state with F the state update function. $\boldsymbol{w}$ is the process noise. $\boldsymbol{z}$ is the measurement, with B the measurement function. $\boldsymbol{m}$ is the measurement noise. The output ($\boldsymbol{y}$) is calculated from the current state by the output function (Y). In the Kalman sequence, the Jacobians $\boldsymbol{F}(k)$ and $\boldsymbol{B}(k)$, which are the derivatives of the state and measurement function, are used to propagate the prediction error and covariance matrices

$$\boldsymbol{F}(k) = \left.\frac{\partial F}{\partial \boldsymbol{x}}\right|_{\hat{x}(k)}, \qquad \boldsymbol{B}(k) = \left.\frac{\partial B}{\partial \boldsymbol{x}}\right|_{\hat{\boldsymbol{x}}(\boldsymbol{k})} \tag{42}$$

The predicted state and covariance matrix are denoted with a subscript "pre", and the measurement states and matrix with a subscript "mea". The discrete time step is denoted with k.

1. Predict the current state from the previous state

$$\hat{\boldsymbol{x}}_{pre}(k) = F\big(\hat{\boldsymbol{x}}_{mea}(k-1)\big) \tag{43}$$

Update the progress in the covariance matrix.

$$\boldsymbol{C}_{pre}(k) = \boldsymbol{F}\boldsymbol{C}_{mea}(k-1)\boldsymbol{F}^{T} + \boldsymbol{C}_{w} \tag{44}$$

2. In the next phase the measurement update is done. This depends on the innovation matrix ($\boldsymbol{S}$), which is used to calculate the Kalman gain matrix ($\boldsymbol{K}$).

$$\begin{aligned} \boldsymbol{S}(k) &= \boldsymbol{B}\boldsymbol{C}_{pre}(k)\boldsymbol{B}^{T} + \boldsymbol{C}_{m} \\ \boldsymbol{K}(k) &= \boldsymbol{C}_{pre}(k)\boldsymbol{B}^{T}\boldsymbol{S}^{-1}(k) \end{aligned} \tag{45}$$

Use the difference between the prediction and actual measurement as the residual

$$\boldsymbol{\varepsilon}(k) = \boldsymbol{z}(k) - B\big(\hat{\boldsymbol{x}}_{pre}(k)\big) \tag{46}$$

Use the Kalman gain as the weight function for the measurement update.

$$\hat{\boldsymbol{x}}_{mea}(k) = \hat{\boldsymbol{x}}_{pre}(k) + \boldsymbol{K}(k)\boldsymbol{\varepsilon}(k) \tag{47}$$

Propagate the measurement to the covariance matrix.

$$\boldsymbol{C}_{m}(k) = \boldsymbol{C}_{pre}(k) - \boldsymbol{K}(k)\boldsymbol{S}(k)\boldsymbol{K}^{T}(k) \tag{48}$$

3. The output ($\boldsymbol{y}$) of this system is calculated at the end of the measurement update.

$$\boldsymbol{y}(k) = Y\big(\hat{\boldsymbol{x}}_{mea}(k)\big) \tag{49}$$

These steps are performed during each iteration.

Before we can build the Kalman filter, we need to define the states and the state evolution

1) States

For the head tracking the states are the position of the head ($\boldsymbol{o}_{Hm}^{Vz}$), the orientation of the head in quaternions ($\boldsymbol{q}_{Hm}^{Vz}$), the twist of the head ($\boldsymbol{v}_{Hm}^{Vz,Vz}(k)$) and the position of the marker with respect to the head ($\boldsymbol{p}^{Hm}$, in local frame). We measure the position of the markers in the global frame $\boldsymbol{p}^{Vz}$.

Different from the OP method the frame is placed ψ^{Hm} is placed in the first marker and aligning the frame with the second and third marker. This leaves the second marker only with one variable and the third with two variables. Later this displacement with respect to the OP method will be corrected.

$$\begin{gathered} \boldsymbol{o}_{Hm}^{Vz}(k) = {}^{1}\boldsymbol{p}^{Vz}(k) \\ {}^{1}\boldsymbol{p}^{Hm}(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \qquad {}^{2}\boldsymbol{p}^{Hm}(k) = \begin{bmatrix} {}^{2}_{x}p^{Hm}(k) \\ 0 \\ 0 \end{bmatrix}, \\ {}^{3}\boldsymbol{p}^{Hm}(k) = \begin{bmatrix} {}^{3}_{x}p^{Hm}(k) \\ {}^{3}_{y}p^{Hm}(k) \\ 0 \end{bmatrix} \end{gathered} \tag{50}$$

Therefore, the state becomes

$$\boldsymbol{x}(k) = \begin{bmatrix} \boldsymbol{o}_{Hm}^{Vz}(k) \\ \boldsymbol{q}_{Hm}^{Vz}(k) \\ \boldsymbol{v}_{Hm}^{Vz,Vz}(k) \\ {}^{2}\boldsymbol{p}^{Hm}(k) \\ \vdots \\ {}^{n}\boldsymbol{p}^{Hm}(k) \end{bmatrix} \tag{51}$$

With a measurement function

$$\boldsymbol{z}(k) = \begin{bmatrix} {}^{1}\hat{\boldsymbol{p}}^{Vz}(k) \\ \vdots \\ {}^{n}\hat{\boldsymbol{p}}^{Vz}(k) \end{bmatrix} \tag{52}$$

The output ($\boldsymbol{y}$) is calculated from the normalized quaternion (24) and position vector to calculate the pose of the head with respect to the camera frame.

$$\boldsymbol{y}(k) = \widehat{\boldsymbol{H}}_{Hm}^{Vz} = \begin{bmatrix} \widehat{\boldsymbol{R}}_{Hm}^{Vz} & \widehat{\boldsymbol{o}}_{Hm}^{Vz} \\ 0 & 1 \end{bmatrix} \tag{53}$$

From now on the frame reference is dropped, except for the marker position in the marker frame ${}^{i}\boldsymbol{p}^{Hm}$ and in the Visualeyez frame ${}^{i}\boldsymbol{p}^{Vz}$**.**

2) State evolution

The evolution of state follows the rules shown below. The position of the next frame is the integration over discrete time Δt from the linear velocity $\boldsymbol{u}$**.**

$$\boldsymbol{o}(k+1) = \boldsymbol{o}(k) + \Delta t \boldsymbol{u}(k) \tag{54}$$

The quaternions evolve according to the exponential mapping of (26). A normalization step of the quaternions is required since the measurement update step (47) does not guarantee unit quaternions. There are also possibilities in which not the quaternions but the quaternion error is used in the state evolution.

$$\boldsymbol{q}(k+1) = \boldsymbol{e}^{\Delta t \widetilde{\boldsymbol{\omega}}}(k) \otimes \frac{\boldsymbol{q}(k)}{\|\boldsymbol{q}(k)\|} \tag{55}$$

The process noise enters the system at the twist level. The twist is assumed to dampen out according the tunable matrix $\boldsymbol{F}_{vv}$**.** Which consist of tuning parameters α and β. These parameters are to be determined empirically. These damping factors are used for safety reasons to prevent motion when no data is present.

$$\begin{gathered} \boldsymbol{v}(k+1) = \boldsymbol{F}_{vv}\boldsymbol{v}(k) + \boldsymbol{w}_v(k), \\ \boldsymbol{F}_{vv} = \begin{bmatrix} \alpha \boldsymbol{I}_3 & 0 \\ 0 & \beta \boldsymbol{I}_3 \end{bmatrix} \end{gathered} \tag{56}$$

The markers are not moving in the local frame.

$${}^{i}\boldsymbol{p}^{Hm}(k+1) = {}^{i}\boldsymbol{p}^{Hm}(k) \tag{57}$$

Together the state evolution function $F(\boldsymbol{x}(k))$ is given.

$$\boldsymbol{x}(k+1) = \begin{bmatrix} \boldsymbol{o}(k) + \Delta t \boldsymbol{u}(k) \\ \boldsymbol{e}^{\Delta t \boldsymbol{\omega}}(k) \otimes \dfrac{\boldsymbol{q}(k)}{\|\boldsymbol{q}(k)\|} \\ \boldsymbol{F}_{vv}\boldsymbol{v}(k) \\ {}^{1}\boldsymbol{p}^{Hm}(k) \\ \vdots \\ {}^{n}\boldsymbol{p}^{Hm}(k) \end{bmatrix} + \boldsymbol{w}(k) \tag{58}$$

With process noise

$$\boldsymbol{w}(k) = \begin{bmatrix} 0 \\ \boldsymbol{w}_T(k) \\ 0 \end{bmatrix} \tag{59}$$

The measurement function is giving the position of all the markers in the global frame. Therefore, we can write:

$${}^{i}\widehat{\boldsymbol{p}}^{Vz}(k) = \ \boldsymbol{H}(k){}^{i}\boldsymbol{p}^{Hm}(k) + \boldsymbol{m}(k) \tag{60}$$

$${}^{i}\widehat{\boldsymbol{p}}^{Vz}(k) = \boldsymbol{R}(k){}^{i}\boldsymbol{p}^{Hm}(k) + \boldsymbol{o}(k) + \boldsymbol{m}(k) \tag{61}$$

For complete measurement, set $B(\boldsymbol{x}(k))$ can be expanded.

$$\begin{bmatrix} {}^{1}\widehat{\boldsymbol{p}}^{Vz}(k) \\ \vdots \\ {}^{n}\widehat{\boldsymbol{p}}^{Vz}(k) \end{bmatrix} = \begin{bmatrix} 0 & \cdots & 0 & \boldsymbol{o}(k) \\ \boldsymbol{R}(k) & \ddots & 0 & \boldsymbol{o}(k) \\ \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & \boldsymbol{R}(k) & \boldsymbol{o}(k) \end{bmatrix} \begin{bmatrix} {}^{2}\boldsymbol{p}^{Hm}(k) \\ \vdots \\ {}^{n}\boldsymbol{p}^{Hm}(k) \\ 1 \end{bmatrix} + \boldsymbol{m}(k) \tag{62}$$

3) Jacobians

The Jacobian matrices are given by differentiation of the $F(\boldsymbol{x}), B(\boldsymbol{x})$ functions. These matrices consist of different submatrices. First, the outline of each matrix will be elaborated, and then the sub matrices will be calculated further. The Jacobian of the state evolution is given as:

$$\begin{gathered} \boldsymbol{F} = \begin{bmatrix} \boldsymbol{I}_3 & 0 & \Delta t \boldsymbol{I}_3 & 0 & 0 \\ 0 & \boldsymbol{F}_{qq} & 0 & \boldsymbol{F}_{q\omega} & 0 \\ 0 & 0 & \alpha \boldsymbol{I}_3 & 0 & 0 \\ 0 & 0 & 0 & \beta \boldsymbol{I}_3 & 0 \\ 0 & 0 & 0 & 0 & \boldsymbol{I} \end{bmatrix} \\ \boldsymbol{F}_{qq} = \left.\frac{\partial \boldsymbol{q}(k+1)}{\partial \boldsymbol{q}(k)}\right|_{\boldsymbol{q}(k)}, \quad \boldsymbol{F}_{q\omega} = \left.\frac{\partial \boldsymbol{q}(k+1)}{\partial \boldsymbol{\omega}(k)}\right|_{\boldsymbol{\omega}(k)} \end{gathered} \tag{63}$$

Two Jacobians of the quaternion evolution have to be given. Within which, the normalization step of (55) has to be taken into account.

$$\boldsymbol{F}_{qq} = [e^{\Delta t \boldsymbol{\omega}} \otimes] \left(\frac{\boldsymbol{I}_4}{\|\boldsymbol{q}\|} - \frac{\boldsymbol{q}\boldsymbol{q}^T}{\|\boldsymbol{q}\|^3} \right) \tag{64}$$

For the Jacobian with respect to the angular velocity, it is sufficient to use the exact time derivative of (25). The rationale of this simplification can be found in Appendix A

$$\boldsymbol{F}_{q\omega} = \frac{\Delta t}{2} [\boldsymbol{q} \otimes]^T \begin{bmatrix} \boldsymbol{I}_3 \\ 0 \end{bmatrix} \tag{65}$$

Also for the Jacobian of the measurement function $\boldsymbol{B}$:

$$\begin{gathered} \boldsymbol{B} = \begin{bmatrix} \boldsymbol{I}_3 & 0 & 0 & 0 & \dots & 0 \\ \boldsymbol{I}_3 & {}^{2}\boldsymbol{B}_{pq} & 0 & \boldsymbol{R}_{Hm}^{Vz}(k) & \ddots & 0 \\ \boldsymbol{I}_3 & \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{I}_3 & {}^{n}\boldsymbol{B}_{pq} & 0 & 0 & \dots & \boldsymbol{R}_{Hm}^{Vz}(k) \end{bmatrix} \\ {}^{i}\boldsymbol{B}_{pq} = \left.\frac{\partial {}^{i}\widehat{\boldsymbol{p}}^{Vz}(k)}{\partial \boldsymbol{q}(k)}\right|_{\boldsymbol{q}(k)} \end{gathered} \tag{66}$$

For the calculation of the last remaining Jacobian, we must take the derivative of the measurement position to the quaternions. The derivative of (61) to quaternions is clearly only depended on the rotation matrix $\boldsymbol{R}$:

$${}^{i}\boldsymbol{B}_{pq} = \left(\left.\frac{\partial \boldsymbol{R}}{\partial \boldsymbol{q}}\right|_{\boldsymbol{q}} \right) {}^{i}\boldsymbol{p}^{Hm} \tag{67}$$

Now the contribution of each quaternion element in $\boldsymbol{R}$ is differentiated from equation (24).

$$\begin{aligned} \frac{\partial \boldsymbol{R}}{\partial {}_{x}q} &= 2(-{}_{x}q\boldsymbol{I}_{\boldsymbol{3}} + \boldsymbol{n}_{x\,l}\boldsymbol{q}^T + {}_{l}\boldsymbol{q}\boldsymbol{n}_x^T + {}_{s}q[\boldsymbol{n}_x \times]) \\ \frac{\partial \boldsymbol{R}}{\partial {}_{y}q} &= 2(-{}_{y}q\boldsymbol{I}_{\boldsymbol{3}} + \boldsymbol{n}_{y\,l}\boldsymbol{q}^T + {}_{l}\boldsymbol{q}\boldsymbol{n}_y^T + {}_{s}q[\boldsymbol{n}_y \times]) \\ \frac{\partial \boldsymbol{R}}{\partial {}_{z}q} &= 2(-{}_{z}q\boldsymbol{I}_{\boldsymbol{3}} + \ \boldsymbol{n}_{z\,l}\boldsymbol{q}^T + {}_{l}\boldsymbol{q}\boldsymbol{n}_z^T + {}_{s}q[\boldsymbol{n}_z \times]) \\ \frac{\partial \boldsymbol{R}}{\partial {}_{s}q} &= 2({}_{s}q\boldsymbol{I}_{\boldsymbol{3}} + [{}_{l}\boldsymbol{q} \times]) \end{aligned} \tag{68}$$

In which the selection vectors are $\boldsymbol{n}_x = [1 \quad 0 \quad 0]^T, \boldsymbol{n}_y = [0 \quad 1 \quad 0]^T, \boldsymbol{n}_z = [0 \quad 0 \quad 1]^T$

The contribution of each element can now be added to come to the complete Jacobian.

$$ {}^i\boldsymbol{B}_{pq} = \left[\frac{\partial \boldsymbol{R}}{\partial_x q}\,{}^i\boldsymbol{p}^{Hm} \quad \frac{\partial \boldsymbol{R}}{\partial_y q}\,{}^i\boldsymbol{p}^{Hm} \quad \frac{\partial \boldsymbol{R}}{\partial_z q}\,{}^i\boldsymbol{p}^{Hm} \quad \frac{\partial \boldsymbol{R}}{\partial_s q}\,{}^i\boldsymbol{p}^{Hn}\right. \tag{69} $$

Now all the Jacobians are calculated.

4) *Filter parameters*

There are a few parameters left to be chosen. These parameters can be used to tune the Kalman filter. These are the standard deviation of the measurement noise (σ_v) and process noise (σ_w). The damping facors α, and β are responsible for the translational and rotational velocity damping. The measurement noise are identified on forehand to be 1 mm

5) *Detecting jumpy markers*

Since the variance of the measured markers is known, we can improve the MKF method to detect which marker is jumped. In this case we make an allowable spread of the error depended on the current covariance matrix. This can be done by calculating the Mahalanobis distance of the measurement residual $\boldsymbol{\varepsilon}(k)$. This uses the inverse of the innovation matrix to calculate for each marker measurement the normalized distance.

$$ {}^i d = \sqrt{{}^i\boldsymbol{\varepsilon}^T(k)\boldsymbol{S}^{-1}(\boldsymbol{k})\,{}^i\boldsymbol{\varepsilon}(k)} \tag{70} $$

If this distance is larger than a specified value, we call this a marker jumped. We take here the value three, meaning the residual is three times the covariance distance.

C. *Correct to the same frame*

As mentioned before the OP method has it local frame oriented along the initial frame and rotated from that. It is placed in the center of mass. The MKF method uses a frame placed in the first marker and oriented along the second and third marker. In order to compare the methods the systems have to be rotated to the same frame.

$$ \boldsymbol{H}^{Vz}_{Hm,\mathrm{OP}} = \boldsymbol{H}^{Vz}_{Hm,MKF}\boldsymbol{H}^{Hm,MKF}_{Hm,\mathrm{OP}} \tag{71} $$

The constant correction matrix $\boldsymbol{H}^{Hm,\mathrm{OP}}_{Hm,MKF}$ can be calculated using the cross-matrix method (Refer to Appendix D) of the averaged local markers of the OP method. Another option is to calculate the difference between the two frames, use a low pass filter and correct MKF calculation.

$$ \boldsymbol{H}^{Hm,MKF}_{Hm,\mathrm{OP}} = \boldsymbol{H}^{Hm,MKF}_{Vz}\boldsymbol{H}^{Vz}_{Hm,\mathrm{OP}} \tag{72} $$

D. *Test setup and evaluation measures*

To test and compare the quality of the two proposed tracking methods three different test setups are used. We compare the accuracy and the smoothness of the signal, as we are interested in accurateness and noise suppression.

A random multisine is generated to resemble the motion of the head. From this the marker motion are simulated and the measurement noise is added. Now the actual head motion is known and we can compare the estimated head motion from the two methods.

The second evaluation is done on actual head motion data obtained from a subject walking on a treadmill. The data is processed offline. The actual pose of the head is not known but we can see how the methods operate under semi-realistic circumstances. The smoothness of both signals can be compared. Worst-case data of a subject walking at 6 km/h is used.

For the last method, the two methods are implemented into the controller algorithm of the robot for realtime testing. It can be seen how both methods perform under the non-constant frame rate of the tracker system. The orthogonal Procrustes method has been implemented successfully. As the implementation of the Kalman Filter into the realtime controller has not been done a comparison of the realtime performance cannot be conducted.

IV. Results

The results of the artificial data and the offline filtering of actual measurements are presented here. The on-line tracking is not performed by the MKF method.

A. *Artificial data*

A multisine is generated and the two methods are applied to the data. The result can be viewed In Figure 24 and Figure 25 it can be seen that the modified Kalman method is slightly more accurate than the orthogonal Procrustes method. The RMS values of the error can be found in Table

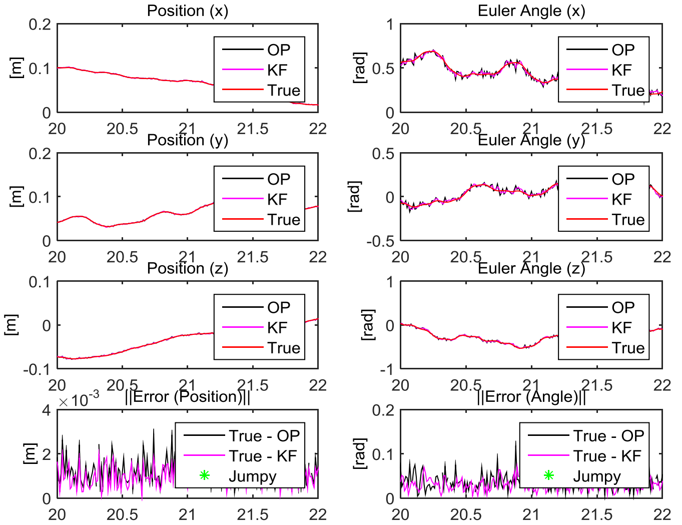


Figure 24. Pose estimation during artificial data. On the left, the position and to the right, the Euler angles around the x-y-z axis. Bottom row gives the position and orientation error of the pose estimators.

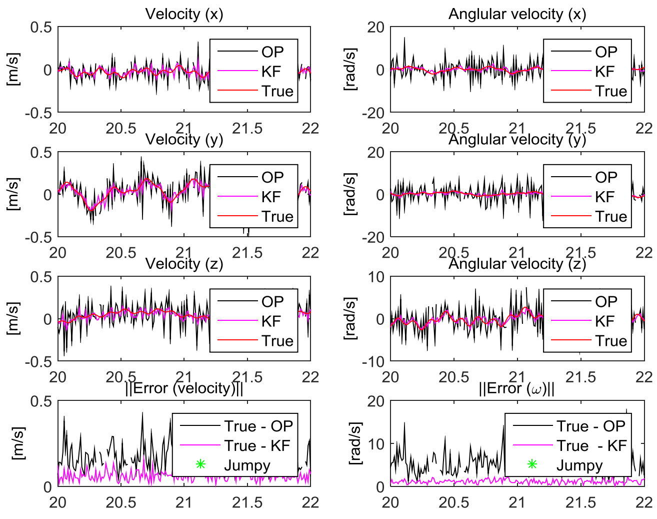


Figure 25. Velocity profiles of the two methods during artificial data. On the left side is the linear velocity while to the right is the angular velocity. Also the error in velocity tracking is plotted

4. For the velocity, the errors are even bigger. This means that MKF is in this case not only more accurate but also especially smoother.

Error(RMS)	OP	MKF
Position (mm)	1.4	1.0
Orientation (mrad)	44	35
Velocity (m/s)	0.20	0.06
Angular velocity (rad/s)	6.2	1.3

Table 4. RMS values of the error of artificial data

B. *Offline filtering*

The resulting tracking can be found in Figure 27 and Figure 26. From these data can be seen that estimation of the two methods generally stay within 1 mm distance and 0.03 rad from each other. Furthermore, it can be seen that there are some jumpy markers. They are denoted with a green star, this is problem originates the camera system.

Since the acceleration of the OP method is larger than the MKF method, it can be seen that the MKF method is smoother. The suppression of the measurement noise comes at a cost of inducing a small delay into the system, as can be seen in Figure 27. This has to do with the choice of tuning parameters of the Kalman filters. These values are chosen such that the suppression of noise is strong and the accuracy/difference with the OP method stays within 1 mm for worst-case scenario.

C. *Actual Implementation*

Currently only the motion is tracked using the OP method. Refer to Chapter 2 for the results on realtime tracking using the OP method. The results of the Kalman filter are not available as the implementation of the Kalman filter is lacking. The main problem with the implementation is of technical nature, which requires more debugging.

V. Discussion Conclusion

It can be concluded that the extended Kalman method results in improved tracking when compared to the Orthogonal Procrustes method. It is both smoother and more accurate under artificial data and smoother during offline tracking.

In further improvement can be made by adding an accelerometer to the subject’s head. This can give additional pose information and acceleration. This requires an expansion of the current algorithm.

The method can be also improved by using the fact that the head motion during treadmill walking is a periodic signal that can be predicted by adaptive frequency oscillators [44], [45].

At this moment, the MKF method is not yet implemented successfully into the realtime controller. This can be done in the near future

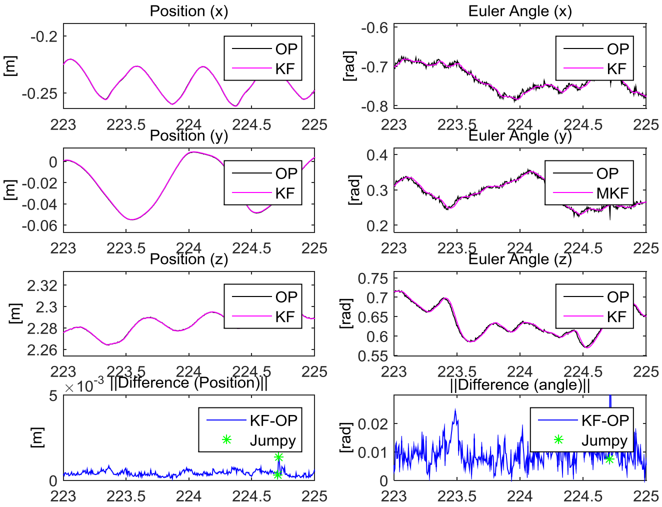


Figure 26. Pose estimation during offline tracking. On the left, the position and to the right, the Euler angles around the x-y-z axis. Bottom row gives the position and orientation error of the difference between the estimators.

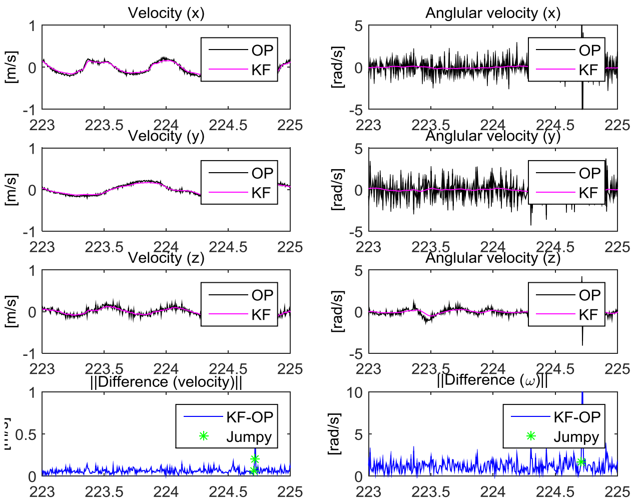


Figure 27. Pose estimation of the two methods during offline processing of measured data. On the left side is the position while to the right is the Euler angle. Also the difference in velocity tracking is plotted

Chapter 4. Safety design of medical robots, applied to a TMS robot

I. Introduction

Safety is of utmost importance in the design of medical robots. In the Chapter 2, it was shown the safety was a design consideration on every level of the process. It this chapter the safety design of the TMS robot is expanded and the safety measures are presented.

Medical robots are designed to mechanically interact with the subject or patient. They cannot be simply fenced away as industrial robots often are. The questions arise: "How can the safety of the subject be ensured in all conditions?" Can we quantify and reduce the risk of harm to the subject to acceptable level? What would happen if a malfunction, mechanical, electronic, software, or man-made, would occur?

Kazanzides [8] points out that for industrial robots it is desired to continue to work under fault conditions. They are designed to be fault-tolerant. For medical robots it is often enough to stop the robot in case of a malfunction. This is called fail-safe. This means that a malfunction should be detected before it can do harm. It is therefore desired to define what the possible malfunctions and errors are and what the consequence would be. The failure mode effect analysis (FMEA) is a tool often used for identifying the risks of a system. A FMEA quantifies the chance and consequence of the risk. The risk of a system is defined as the chance of a malfunction multiplied with the consequence of this malfunction. A FMEA is generally made during a brainstorm with the stakeholders of a project. Experts of each design field are required to give a good estimation of the risk.

In the literature on safety design of medical robots [8]–[10], [46]–[48], redundant sensors and watchdogs are proposed to increase the change of detecting sensor or controller failure. It is advised to make use of a redundant controller as an observer. The output of the redundant system has to be compared continuously. A difference in output points to an internal malfunction.

The psychological aspects of the safety have received only limited attention in the design of medical robots. Especially in this TMS robot, the perception of safety will have strong influence on the usability of the robot. The

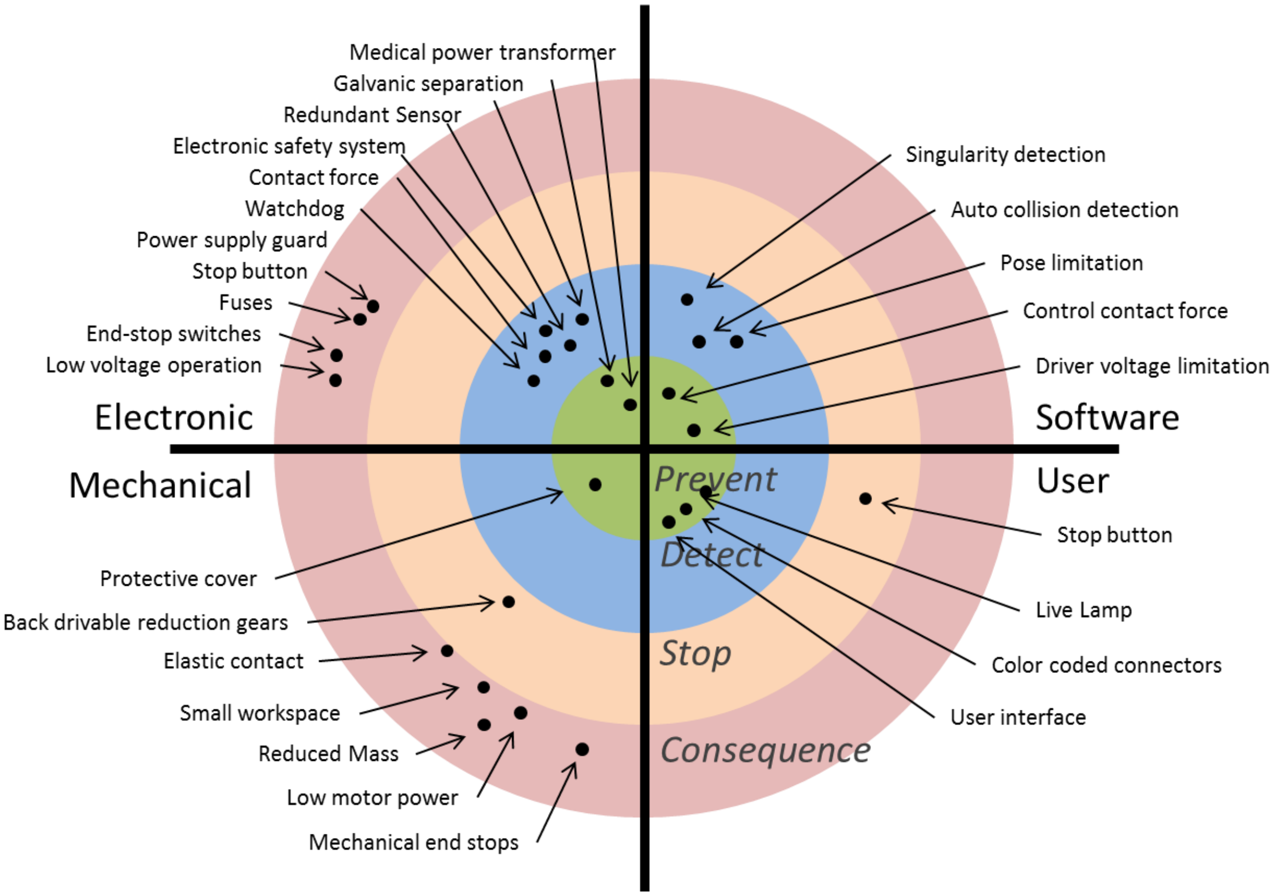


Figure 28. The safety measures are categorized according to the safety level and malfunction domains.

subjects must have the perception that the robot is be safe in order for them to be willing to take part in the treadmill training It might even influence the walking behavior of the subject. If the subjects are afraid of the robot, hovering over they can respond by ducking way.

A. *Safety classification*

The safety measures are classified on four levels and in four domains. The levels indicate at what time in the failure cascade the safety measure becomes active. The domains indicate on which physical domain the safety measure relies.

1) *Safety levels*

The risk of a failure can be reduced on four levels; the four safety levels. Firstly, to reduce the risk the malfunction has to be *prevented* al together. This requires for example a maintenance program, correct usage of the apparatus and high quality parts. Secondly, the chance of timely *detection* is to be increased. As stated previously this requires redundant sensors detect a sensor failure. Redundant controllers or watchdogs can be implemented to guard the controllers. Consequently, in case of a detected error, an emergency stop should be made. The *stop conditions* should not cause any harm to the subject or damage the system. This means that the kinetic and electrical energy in the system should be dissipated quickly. The system should not collapse onto the subject. Fourthly, if the safety systems fail to detect the error the robot should be designed in such a way that the *consequence* of such a malfunction is reduced as much as possible. This can be done by reducing the mass of the system, using flexible contact, using low power electronics, and back drivable motors.

2) *Malfunction domains*

There are four domains in which a malfunction can occur: mechanical, electrical, software and, user level. With mechanical failure, one can think of broken parts of the robot. Electrical failure includes broken sensors, connectors or power supplies. Failure in the software might be locked up computer, instability of the controller or calibration error. Man-made errors are the most difficult to predict and to prevent. This includes putting fingers or hands in between moving parts of the robot, overriding safety limits, not following instructions, and failing timely maintenance.

The consequences of these failures can also be categorized in these four domains. Note that failure in one domain often leads to malfunctions in another domain. For example, an instable controller can result in very strong oscillations of the end-effector with a possible mechanical failure as a consequence. The end result, which is to be prevented, is the hazard at the human level, that is, the harm to the subject or operator.

II. SAFETY DESIGN

A. *FMEA*

To ensure the safety of the user of the TMS robot at all time, the malfunctions are identified in a FMEA. The major risks are to be counteracted by safety measures. This FMEA process is done at stages throughout the design of the robot.

The major hazards of the robot include sensor and controller failure. If the controller receives faulty information it will move the robot erroneously. This can result in high velocities and acceleration and placement errors. These can have internal collision or an impact with the subject as a consequence. Another hazard can occur when the subject places their hand of fingers in between moving parts of robot. Instability of the position controller can result in strong oscillations of the robot with a possible mechanical failure as a consequence.

B. *Safety design principles*

The safety measures are classified on four levels and in four domains. The levels indicate at what time in the failure cascade the safety measure becomes active. The domains indicate on which physical domain the safety measure relies.

1) *Contact guarantee strategy*

Impact between the stimulator and the skull is one result in major injury. To prevent a high energetic collision, a spring mechanism is employed that while ensure contact between the stimulator and skull during fast motions. No kinetic energy between the stimulator and the subject can be exchanged as they move with equal velocities.

To achieve this a spring system is designed which presses the coil against the skull with approximately 25 N. By measuring the deflexion of the spring, an estimation of the contact force can be given. Full velocity and robot power is only allowed when the contact force is within safety limits. If the contact is about to be lost due to reduced deflexion an emergency stop is to be performed. Between the stop and the safe range an alarm range is placed. The robots maximal power is reduced, and a sound heard to warn the subject and the operator to restore the contact force. Also when the contact force becomes too large an emergency stop has to be performed. The spring system also will give a more soft feeling to the robot small placement errors of the robot will not be transmitted directly to the skull. The safety boundaries have to be chosen based on experience gained with testing of subjects. The system should not be too safe, such that a premature emergency stop is made. With a result that the users might be compelled to override the safety mechanism. The spring force measurement can be used to control the robot to reduce the fluctuations of contact force and stay closer to the desired contact force. This controller is designed in another document. There the stability issues and performance limitations are discussed.

2) *Light weight design*

The consequence of the impact with the head during malfunction is reduced by demanding a light-weight design of the robot. By reducing the moving mass as much as possible the kinetic energy in the mechanism is minimized. A light weight structure is achieved by selecting a 6 DOF parallel robot to place the stimulator. Owing to the parallel structure, the actuators can be placed on the base, resulting in a much lower moving mass compared to serial mechanisms.

C. *Safety measures*

Using the FMEA several risk are quantified and countermeasures are taken. They are divided in the four domains introduced previously.

1) Mechanical

Main mechanical hazard for the subject is a high energetic collision. As the robot is designed to move at high velocities, an impact can be prevented by guaranteeing contact during full power motion. This is done by placing an *elastic contact* between the coil and the robot. The deflection of elastic contact is measured to estimate the contact force. When the contact force exceeds a minimal or maximal force, the robot should be stopped. This soft contact also increases the comfort of the user as small placement deviations, result in damped force oscillations.

To reduce the consequence of failure, the energy in the system is be reduced by *reducing the mass* of the moving parts of the robot. This is achieved by choosing a parallel manipulator. A parallel robot has the motors and gear placed at the base. The moving mass is a sum of the mass connective rods, the platform and the coil. The connective rods are made from carbon tubes and aluminum. The platform is made from aluminum. This results in a moving mass of the robot (without the coil) of 2.5 kg. Reduced mass also results in *less required motor power*. This is also safer as it reduces the effect of a run-away motor.

Parallel robots have the advantage that they have relatively *small workspaces.* The robot cannot reach places outside it is intended range of motion. This means that the live or dangerous area is also small.

Back drivable reduction gears with a transmission ratio of 1/35 are used. These gears have enough damping to dissipate the energy in the system in case of an emergency stop. The back drivability means the subject can move the end-effector by hand in case the power is off. This will prevent the subject from being stuck beneath the robot. The stiction of the gears prevents the robot from collapsing. Other medical robots rely on emergency brakes on the motors to achieve this result.

End-stops are placed at the joints to prevent the robot from damaging itself during a run-away. This also limits the workspace of the robot to the required range of motion. Electric end-end stop switches can be used to detect malfunction and power down the system.

To prevent the subject from getting its hand crushed by the moving arms of the robot a *protective cover* is placed over the Hexa. It is made from heavy fabric with metal rings.

2) Electrical

The function of the vital sensors is checked by using *redundant sensors*. Two different sensors modalities are used to measure the most important features. For example, an incremental encoder is measured in comparison to a potential meter. This way, a conjoint failure due to a mutual external factor is reduced as much as possible.

An *external electrical safety* system is designed to guard the electronic safety of the system. This safety system performs a number of safety test and redundantly measures sensors and operation of the real-time controller. The system measures and compares the *contact force* on the subject and governs the power to the robot. It will only allow full power if the contact with the subject is ensured. It measures the motor *end-stop switches*, compares the redundant spring force displacement sensors to *detect sensor failure*. It also checks the *power supply guard*. It has a *watchdog* to check for lock-up of the robot controller. The external safety system is *galvanic separated* from the rest of the system. S*top buttons* allow the subject and operator to cut the power to the robot, enforcing an emergency stop in case of a sense of danger.

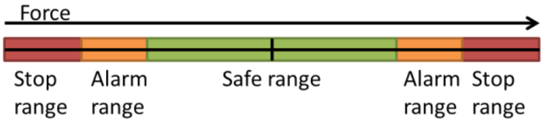


Figure 29. Force ranges for the contact force. The robot is allowed to move at full power in the safe range. In the alarm range the power to the motors is cut. When the force reaches stop range, an emergency stop is made.

The complete robot is isolated from the common power lines by a *medical power transformer*. This to prevent interference from and to other equipment. The motors operate at a *low voltage*. This reduces the hazard of short circuit. The currents to the power supply and the motors are limited by *fuses.*

3) Software

At the software level, many sensors can be measured and complicated calculations can be performed. This gives the possibility to implement more complex safety measures. However, it should be noted that the software itself cannot prevent software failure, such as controller lock-up. The software safety is designed to be stricter than the external electronic safety. This allows a more smooth software emergency stop and makes the external safety system the last guard. The *redundant sensors* are checked to detect sensor failure. This means comparing spring displacement sensors and the joint angle sensors. The *contact force is measured and controlled* to ensure a constant contact force. When the contact force is in the middle of the safety zone, the distance to contact loss is maximal. This makes the system more failsafe. The software makes an emergency stop of the stop range is exceeded. One should notice that this is redundant to the electronic safety system. The boundaries on software level are stricter than the electronic boundaries. This prevents the electronic system from intervening. The workspace of the robot is limited by singularities. When the robot is in such a singularity configuration, it will lose stiffness in one or more directions. This renders the robot incontrollable. To prevent this, a *singularity detection* is built in to the controller software. It detects when the condition number of the Jacobian (a measure for distance to a singularity), is below a certain

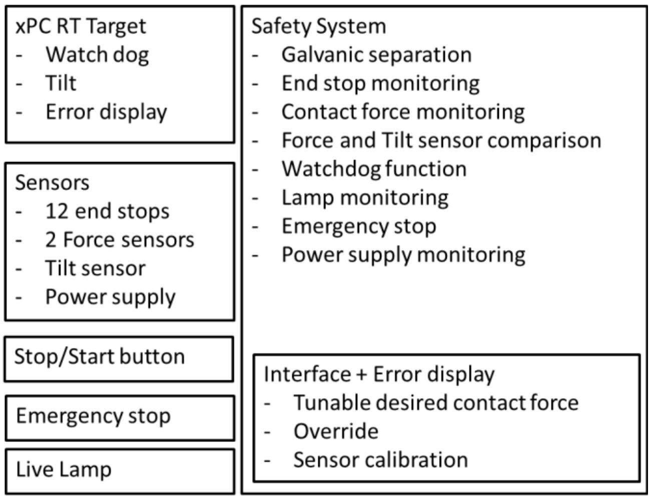


Figure 30. Schematic overview of interconnection of the safety system with other parts of the design.

level. See Chapter 5.IV for more information on singularities and the condition number Another hazard for when the robot reaches an undesired pose is that arms of the robot can hit each other. An *auto collision detection* checks if the distance between arms of the robot is less than a safety minimum. The *driver voltages are limited* to prevent too strong motor action in case of malfunction or user error. The *pose is limited*, in which the robot is allowed to move. If it exceeds this range, a safety stop is performed.

4) User level

The safety at user level consists of a *stop button* to allow the subject to stop the system at any time. A *live lamp*, with a slow pulse, shows that the robot is live and can move. To prevent a connection mistake all wires, and connectors and housing are *color-coded*. A *User interface* has been developed to make the system more usable and to shield the code away from the operator.

D. Safety perception

As stated in the introduction the perception of safety of the user is of utmost importance. The subject has to have the feeling the robot is to be trusted. The current design is aimed to make the robot as inconspicuous as possible. This is done by slender design which leaves a large field of view to the subject, shielding away the complicated mechanics and electronics by the protective cover and, by reduction of the annoying or persistent sounds and light. Also in the control the robot should be as unobtrusive as possible, a smooth motion trajectories during starting phase and tracking. The contact force control is must give the subject the feeling they can stop or push away the robot at any moment. It should however be noted that this design has to be evaluated and optimized for this psychological. For new version of this system, a redesign with respect to aspect is strongly suggested.

III. Validation and Evaluation

In this document the risks of the TMS have been identified. Several safety measures have been presented and implemented into the robot and control to reduce these risks. The residual risk is identified with the FMEA analysis. The FMEA is currently performed by the designing party only. In the future also the users, subjects, operators and other specialists should be included into evaluation of the risks. The practical evaluation of the safety system is not yet done systematically. No safety tests have been performed to see how the system functions under undesired circumstances and what kind of risk that poses for the subject. To guarantee the medical safety several tests have to be performed. These tests include electromagnetic Compatibility (EMC) testing, failure testing, user testing, and subject testing. The goal of this safety validation is to achieve CE mark for medical devices.

The EMC testing is required to show that the device does not electronically interfere with other devices in the hospital. The EMC testing has to be performed by an external party as the university does not have these equipment. Electronic safety testing also includes the test of the external safety system. Different failures of the sensors have to be simulated to see how the robot response to those.

User tests have to be performed to guarantee that the operator of the robot can use the system in a safe manner. This has to show that the documentation and manuals are clear and that the user interface will prevent undesired use. Before the system can be tested with humans, several dummy tests will have to be performed to see if the robot can detect malfunction and perform emergency stops in a safe manner. This includes a test of the subject quickly ducking away from the robot.

The spring safety mechanism has to be tested and tuned to allow the proper response during normal operation. This includes selection of the contact force and force boundaries.

Even after thorough testing, complete safety cannot be guaranteed completely. Continuous improvement and risk validation is required.

IV. Conclusion and Discussion

Throughout the design of robot, safety has been a paramount priority. At all stages of the design safety measures are taken to prevent, detect, and handle malfunction and minimize the consequences of failure. The implemented safety measures reduce the risk significantly as evaluated by the FMEA. However, these safety measures and FMEA still have to be evaluated by an external party. In the future, a maintenance system has to be made to ensure timely maintenance. The electronic safety system should be tested more systematically for malfunction and usability. An evaluation and, if necessary, a redesign with respect to the psychological aspects of the robots is advised.

A complete guarantee of safety at all-time can never be given. The safety design of a system is never finished as during operation new risks can and have to be identified.

Chapter 5.Kinematic analysis of the Hexa manipulator with singularity and auto-collision detection

The Hexa robot places the TMS coil over the targeted stimulation site. To control the robot to this position, the relation between the position and orientation of the coil and the joint angles has to be given. For robots, there are two types of kinematic relations. The forward (or direct) kinematic model (FKM/DKM) gives the end-effector pose a function of the given joint angles. The inverse kinematic model (IKM) gives converse of this relation; namely, the joint angles as function of the given end-effector pose. For serial robot types, the closed form of the FKM can be found relatively straight forward. The IKM cannot not always be found as easily, and in some cases, a closed form is impossible to find. For parallel robots the opposite hold: the IKM can be found easily and the FKM is difficult if not impossible to find. The derivative of the kinematic relations will give the input-output equations of the robot. The IO equations relate the joint velocities to the end-effector twist and the input joint forces to the end-effector wrist. Usually, these relations are the Jacobians of the kinematic model.

The Hexa robot (Figure 31, *left*) is a parallel mechanism of the 6-R̲US type. Six identical arm connect the end-effector via an actuated revolute joint (R̲) at $\boldsymbol{a}$, an universal joint (U) at $\boldsymbol{b}$ and a spherical joint (S) at $\boldsymbol{c}$. Each leg has five passive degrees of freedom and one actuated constraint from the motor. This actuated constraint lies collinear with the forearm. This means that the forearm can only push and pull on the platform. No static transverse forces are present. Connecting these six arms together, fully constrain the end-effector motion.

There are certain poses in which the platform is no longer fully constraint and the platform gains or loses a mobility, this is called a singularity. These singularities result in a degeneration of the input-output velocity relations of the robot. These pose have to be found and avoided. Another dangerous poses, in which the links of the robot are in danger of colliding with itself (auto collision), are to be avoided.

In this chapter the kinematic relations will be given. From these kinematic relations the input-output velocity model can be derived. Using these input-output relations the singularities can be found and understood. The kinematic relation is also used to find the pose in which the auto-collision can occur.

I. Inverse Kinematic Model

The inverse kinematic model gives the joint angles ($\boldsymbol{\theta}$) as a function of the robot pose ($\boldsymbol{x}$). The robot pose consist of end-effector position ($\boldsymbol{o}$) and platform orientation in (zyx) Euler angles ($\boldsymbol{\phi}$). $\boldsymbol{x}^T = [\boldsymbol{o}^T \quad \boldsymbol{\phi}^T]$. The end-effector pose is given by a homogenous matrix.

$$\boldsymbol{H}_p^B(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{R}_{zyx}(\boldsymbol{\phi}) & \boldsymbol{o} \\ 0 & 1 \end{bmatrix} \tag{73}$$

The rotation matrix $\boldsymbol{R}_{zyx}(\boldsymbol{\phi})$ denotes the rotation in Euler angles, with respectively x, y, and z rotation.

$$\begin{gathered} \boldsymbol{R}_{zyx}(\boldsymbol{\phi}) = \boldsymbol{R}_x \boldsymbol{R}_y \boldsymbol{R}_z \\ \boldsymbol{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & c & -s \\ 0 & s & c \end{bmatrix}, \quad \boldsymbol{R}_y = \begin{bmatrix} c & 0 & s \\ 0 & 1 & 0 \\ -s & 0 & c \end{bmatrix}, \\ \boldsymbol{R}_z = \begin{bmatrix} c & -s & 0 \\ s & c & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{aligned} s &= \sin(\phi), \\ c &= \cos(\phi) \end{aligned} \end{gathered} \tag{74}$$

In Figure 31 the notation of the various parts of the robot is given. For simplicity reasons the notation of pose and

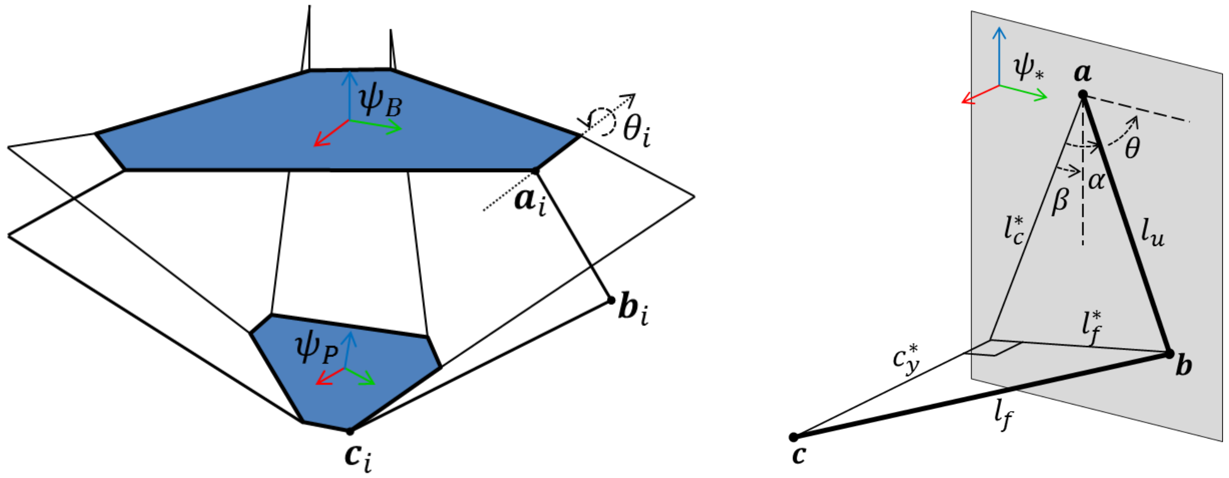


Figure 31 The kinematic model of the Hexa. *Left* shows the position of the shoulder ($\boldsymbol{a}_i$), elbow ($\boldsymbol{b}_i$), and the wrist ($\boldsymbol{c}_i$). *Right* shows the plane normal to the joint axis of ($\boldsymbol{a}_i$). Here also the lengths of the upper arm and forearm are shown. The auxiliary lengths l_f^*, l_c^*, and c_y^* in this plane are used to calculate the angles α, and β which will give the joint angle θ of joint i.

points differs slightly from the rest of this document. The sub- and superscript will be explained further on.

The IKM is calculated by defining a rotated reference frame (denoted with a *), as shown in Figure 31 right. The frame is rotated around the z-axis such that the y-axis is parallel with the rotation axis of the actuator joint $(\boldsymbol{a}_i)$. This is done for each joint $(i = [1 \dots 6])$. From now on, the i notation is dropped. The angle between the joint frame and the base frame is denoted with an angle γ.The projected length of the forearm (l_f^*) on this x^*z-plane and the projected length (l_c^*) from the wrist $(\boldsymbol{c})$ to the shoulder $(\boldsymbol{a})$ on this plane is calculated. The lengths l_f^*, l_c^* and l_u allow the calculation of angles α and β yielding θ. This is done for all arms.

The positions of the wrist $(\boldsymbol{c})$ and the shoulder $(\boldsymbol{a})$, expressed in the base frame, are calculated from the length and the width of the base and the platform.

$$\boldsymbol{a} = \boldsymbol{l}_b = \begin{bmatrix} l_{B_x} \\ l_{B_y} \\ 0 \end{bmatrix}, \qquad \boldsymbol{b} = \boldsymbol{R}_z(\gamma)\boldsymbol{R}_y(\theta)\begin{bmatrix} 0 \\ 0 \\ l_u \end{bmatrix} + \boldsymbol{a}, \tag{75}$$
$$\boldsymbol{c} = \boldsymbol{H}_p^B(\boldsymbol{x})\boldsymbol{l}_p = \boldsymbol{H}_p^B(\boldsymbol{x})\begin{bmatrix} l_{p_x} \\ l_{p_y} \\ 0 \end{bmatrix}$$

With the rotation matrix $\boldsymbol{R}_z(\gamma_i)$ the position of the shoulder $(\boldsymbol{a}_i^*)$ and the wrist $(\boldsymbol{c}_i^*)$ in the rotated reference frame can be calculated.

$$\boldsymbol{a}^* = \boldsymbol{R}_z(\gamma)\boldsymbol{a}, \qquad \boldsymbol{c}^* = \boldsymbol{R}_z(\gamma)\boldsymbol{c} \tag{76}$$

The lengths of the projected triangle are calculated.

$$\left(l_f^*\right)^2 = \left(l_f\right)^2 - \left(c_y^*\right)^2, \qquad l_c^* = ||\begin{bmatrix} c_x^* \\ 0 \\ c_z^* \end{bmatrix}|| \tag{77}$$

The cosine law can be used to calculate the auxiliary angles α and β from the projected triangle.

$$\alpha = \cos^{-1}\left(\frac{l_u^2 - (l_c^*)^2 + \left(l_f^*\right)^2}{2 l_u l_c^*}\right) \tag{78}$$

Similar for the β_i, the angle between the projected point $\boldsymbol{c}$ and the vertical is given.

$$\beta = \tan^{-1}\left(\frac{c_x^*}{c_z^*}\right) \tag{79}$$

Together, the two angles give the angle of the actuator joint with the horizontal plane.

$$\theta_i = -\alpha_i + \beta_i + \frac{\pi}{2} \tag{80}$$

II. Forward Kinematic Model (FKM)

Currently, there exists no closed form forward kinematic model of the Hexa robot. A closed form forward kinematic model is difficult or even impossible to find for some parallel robot structures. There exist solutions for the Steward platform [49], [50]. These solutions depend on the solving of polynomial of order 40, or formulation of an eight dimensional geometrical space. These solution currently only apply to a Steward platform of which the attachment points of the base lie in plane and are pair wise symmetric. The lower part of the Hexa (elbow, forearms, wrist, and platform) is kinematically similar to the Steward platform. The difference is that the lengths are fixed but the elbow can move over a circle. A solution for the Hexa could be found if it is possible to expand the solution for the Steward platform to allow a non-symmetrical, non-planar base. This solution has not yet been found.

Therefore, an iterative Newton Raphson method is used to calculate the end-effector pose $\boldsymbol{x}$ from a given set of joint angles $\boldsymbol{\theta}$.

First, an initial pose is guessed. For this guess, the motor angles are calculated using the inverse kinematic model. Here a hat denotes the estimation.

$$\widehat{\boldsymbol{\theta}}(\boldsymbol{k}) = IKM\left(\widehat{\boldsymbol{x}}(k)\right) \tag{81}$$

Then the joint error is calculated

$$\boldsymbol{\theta}_{\boldsymbol{e}}(k) = \widehat{\boldsymbol{\theta}}(k) - \boldsymbol{\theta} \tag{82}$$

The inverse Jacobian reflects this error to the end-effector pose. This is the update of pose estimation.

$$\widehat{\boldsymbol{x}}(k+1) = \widehat{\boldsymbol{x}}(k) - \boldsymbol{J}_e^{-1}\boldsymbol{\theta}_{\boldsymbol{e}}(k) \tag{83}$$

This process is iterated until a sufficiently small end-effector error is found.

This method has shown to be quite computational demanding; it requires calculation and inversion of the Jacobian at each iteration. To be able to use it in a realtime implementation two simplifications can be made. Firstly, it can be assumed that the Jacobian is constant for the course iteration. The Jacobian and its inverse have to be calculated only once, reducing the computational load strongly. The second simplification is that at each time step one iteration is done. The downside of this approach is that it will take time before the iteration settles to a close enough value. Luckily, this will require only four time steps to approach an accuracy in the order of thousands of a millimeter. In addition, the motion of the robot will be reflected into an error of pose estimation. Therefore, the procedure is upgraded to include the joint velocities.

$$\widehat{\boldsymbol{x}}(k+1) = \widehat{\boldsymbol{x}}(k) - \boldsymbol{J}_{\boldsymbol{e}}^{-1}\left(\boldsymbol{\theta}_{\boldsymbol{e}}(k) + \dot{\boldsymbol{\theta}}\right) \tag{84}$$

In the first method, the amount of required computational time is not constant. This might cause an undesired computational overload, which is much more dangerous than a pose estimation error of which the magnitude is known. Therefore, the later method is chosen.

III. Jacobians

The Jacobian relates the end-effector twist to the joint velocities and vice versa. This also holds for small displacements such as small pose errors. The pose derivative in global base coordinates is given $\dot{\boldsymbol{x}}_g{}^T = [\boldsymbol{\omega}^T \quad \dot{\boldsymbol{o}}^T]$. This means that not the Euler rate, but the angular velocities are used as global derivative variable:

$$\begin{bmatrix} \dot{\theta}_1 \\ \vdots \\ \dot{\theta}_6 \end{bmatrix} = \begin{bmatrix} \frac{d\theta_1}{dx_1} & \cdots & \frac{d\theta_1}{dx_6} \\ \vdots & \ddots & \vdots \\ \frac{d\theta_6}{dx_1} & \cdots & \frac{d\theta_6}{dx_6} \end{bmatrix} \begin{bmatrix} \dot{x}_{g,1} \\ \vdots \\ \dot{x}_{g,6} \end{bmatrix} \tag{85}$$

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}_g \dot{\boldsymbol{x}}_g \tag{86}$$

$$\boldsymbol{\tau} = \boldsymbol{J}_g^{-T}\boldsymbol{w} \tag{87}$$

Using the virtual work theorem it can be proven that the Jacobian can be used to calculate the mapping between the static joint torques ($\boldsymbol{\tau}$) and end-effector wrench ($\boldsymbol{w}$).

For the Hexa robot, the number of joints is equal to the order of end-effector freedom. Therefore, the Jacobian is square with an order of six. The singularities of a mechanism show up in the Jacobian, it loses rank and it will no longer be invertible. The singularities can be found where the determinant and condition number of the Jacobian are zero.

The Jacobian of the Hexa is calculated by extension of the method introduced by Codourey [51] for the Delta robot. The method uses the time derivative of the constraint equations, to find two matrices that relate the dependency of $\dot{\boldsymbol{x}}$ to $\dot{\boldsymbol{\theta}}$. As the Delta robot has no end-effector rotation, the method has to be expanded to incorporate the extra rotation of the Hexa.

The constraint equation used here, is that the distance between wrist and the elbow is equal to the length of the forearm.

$$\|\boldsymbol{c}_i - \boldsymbol{b}_i\|^2 - l_f^2 = 0, \qquad i = [1 \dots 6] \tag{88}$$

Can be rewritten into:

$$\boldsymbol{z}_i^T \boldsymbol{z}_i - l_f^2 = 0 \tag{89}$$

Using (75) this can be rewritten as:

$$\boldsymbol{z} = \boldsymbol{H}_P^B(\boldsymbol{x})\boldsymbol{l}_p + \boldsymbol{R}_z(\gamma)\boldsymbol{R}_y(\theta)\begin{bmatrix}0\\0\\l_u\end{bmatrix} + \boldsymbol{a} \tag{90}$$

Here again the notation with the i-th arm is omitted for brevity. It should also be noted that from now on, the rotation directions are not shown for the rotation matrices.

The derivative of the constraint equation is zero as the length of the lower arm is constant. The time derivative of the constraint equation (89) is:

$$\dot{\boldsymbol{z}}^T\boldsymbol{z} + \boldsymbol{z}^T\dot{\boldsymbol{z}} = 0 \tag{91}$$

Since the commutative property, we can reduce this to

$$\boldsymbol{z}^T\dot{\boldsymbol{z}} = 0 \tag{92}$$

Expanding the derivative of $\boldsymbol{z}$ results in:

$$\dot{\boldsymbol{z}} = \dot{\boldsymbol{o}} + \boldsymbol{\omega} \times \boldsymbol{R}(\boldsymbol{\phi})\boldsymbol{l}_p + \boldsymbol{R}(\gamma)\left(\begin{bmatrix}0\\\dot{\theta}\\0\end{bmatrix} \times \boldsymbol{R}(\theta)\begin{bmatrix}0\\0\\l_u\end{bmatrix}\right) \tag{93}$$

Rewriting gives:

$$\dot{\boldsymbol{z}} = \dot{\boldsymbol{o}} + [\boldsymbol{R}(\boldsymbol{\phi})\boldsymbol{l}_p \times]^T\boldsymbol{\omega} + \boldsymbol{R}(\gamma)\begin{bmatrix}0\\1\\0\end{bmatrix} \times \boldsymbol{R}(\theta)\begin{bmatrix}0\\0\\l_u\end{bmatrix}\dot{\theta} \tag{94}$$

This can be rewritten into a pose dependent part and an actuator joint part:

$$\dot{\boldsymbol{z}} = \boldsymbol{M}\dot{\boldsymbol{x}}_g - \boldsymbol{N}\dot{\theta} \tag{95}$$

With:

$$\begin{gathered}\boldsymbol{M} = \big[[\boldsymbol{R}(\boldsymbol{\phi})\boldsymbol{l}_p \times]^T \quad \boldsymbol{I}_3\big],\\ \boldsymbol{N}_i = -\boldsymbol{R}(\gamma)\begin{bmatrix}0\\1\\0\end{bmatrix} \times \boldsymbol{R}(\theta)\begin{bmatrix}0\\0\\l_u\end{bmatrix}\end{gathered} \tag{96}$$

This results in the constraint equation for each arm:

$$\boldsymbol{z}_i^T\dot{\boldsymbol{z}}_i = \boldsymbol{z}_i^T\boldsymbol{M}_i\dot{\boldsymbol{x}}_g - \boldsymbol{z}_i^T\boldsymbol{N}_i\dot{\theta}_i = 0 \tag{97}$$

Collecting the constraints for all arms gives a matrix equality:

$$\begin{bmatrix}\boldsymbol{z}_1^T\boldsymbol{M}_1\\\vdots\\\boldsymbol{z}_6^T\boldsymbol{M}_6\end{bmatrix}\dot{\boldsymbol{x}}_g - \begin{bmatrix}\boldsymbol{z}_1^T\boldsymbol{N}_1 & \dots & 0\\\vdots & \ddots & \vdots\\0 & \dots & \boldsymbol{z}_6^T\boldsymbol{N}_6\end{bmatrix}\dot{\boldsymbol{\theta}} = 0 \tag{98}$$

From this equality the Jacobian can be found

$$\dot{\boldsymbol{\theta}} = \boldsymbol{J}\dot{\boldsymbol{x}}_g \tag{99}$$

It can be clear that the Jacobian will be:

$$\boldsymbol{J}_g = \begin{bmatrix}\boldsymbol{z}_1^T\boldsymbol{N}_1 & \dots & 0\\\vdots & \ddots & \vdots\\0 & \dots & \boldsymbol{z}_6^T\boldsymbol{N}_6\end{bmatrix}^{-\boldsymbol{1}}\begin{bmatrix}\boldsymbol{z}_1^T\boldsymbol{M}_1\\\vdots\\\boldsymbol{z}_1^T\boldsymbol{M}_6\end{bmatrix} \tag{100}$$

And the inverse:

$$\boldsymbol{J}_g^{-\boldsymbol{1}} = \begin{bmatrix}\boldsymbol{z}_1^T\boldsymbol{M}_1\\\vdots\\\boldsymbol{z}_1^T\boldsymbol{M}_6\end{bmatrix}^{-\boldsymbol{1}}\begin{bmatrix}\boldsymbol{z}_1^T\boldsymbol{N}_1 & \dots & 0\\\vdots & \ddots & \vdots\\0 & \dots & \boldsymbol{z}_6^T\boldsymbol{N}_6\end{bmatrix} \tag{101}$$

The Jacobian (and its inverse) is now expressed in angular velocities of the end-effector. To find the Euler rate, used for the FKM, the Jacobian should be rotated into the relative frames of the Euler angles. This is done by the $\boldsymbol{Q}$ and the $\boldsymbol{E}$ matrix.

$$\boldsymbol{J}_e = \boldsymbol{J}_g\boldsymbol{E}, \qquad \boldsymbol{\omega} = \boldsymbol{Q}\dot{\boldsymbol{\phi}}, \qquad \boldsymbol{E} = \begin{bmatrix}\boldsymbol{I}_3 & 0\\0 & \boldsymbol{Q}^{-\boldsymbol{1}}\end{bmatrix}, \tag{102}$$

With the $\boldsymbol{Q}$ matrix

$$\boldsymbol{Q} = \begin{bmatrix}\dfrac{c(\phi_3)}{c(\phi_2)} & \dfrac{s(\phi_3)}{c(\phi_2)} & 0\\ -s(\phi_3) & c(\phi_3) & 0\\ c(\phi_3)t(\phi_2) & s(\phi_3)t(\phi_2) & 1\end{bmatrix} \tag{103}$$

IV. Singularity analysis

The singularities are already understood as poses in which the end-effector gains or loses a mobility. In addition, it has been seen that this reflects in a Jacobian with a reduced column or row rank. Singularities can be understood as transition between different types of modi of a robot. To find and avoid all singularity poses, first the different types of singularities are described.

Merlet [20] describes three types of singularities. The *redundant input* singularity (RI) is pose in which the end-effector does not move while the actuators make a movement. The second, the *redundant output* singularity (RO), is a pose in which the end-effector can make a movement while the actuators are locked. The third singularity is the *redundant passive motion* singularity (RPM) which allows movement of a passive joint without movement of the end-effector or actuators.

The singularity detection checks if the condition number (104) of the Jacobian is sufficiently low. If the Jacobian is above an empirically determined value, an emergency stop will be made. The condition number is defined as the ratio between the largest and the smallest eigenvalue of a matrix.

$$\kappa(\boldsymbol{J}) = \lambda_{max}(\boldsymbol{J})/\lambda_{min}(\boldsymbol{J}) \tag{104}$$

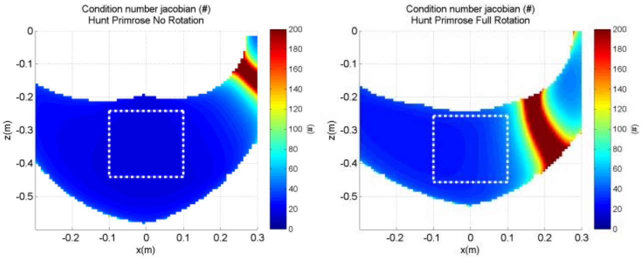


Figure 32 The condition number of the Jacobian during no (*left*) and full (*right*) rotation. The red parts indicate a position of singularities. The whit box denotes the required workspace of the robot

This condition number can be used to numerically find the pose of the singularities, as shown in Figure 32. The figure shows there exist singularities in the range of motion of the robot. They are located at the edge of the range of motion and straight through a part of the range of motion. They however are located outside the required range of motion of the robot (the white box). To understand the singularities and completely prove that all singularities are outside the required workspace, a geometrical description of the singularities is to be given. For this Hexa the complete geometrical description of the singularities are not given yet. The nature of the existing singularities is described using Merlet distinction.

The RI singularities in a Hexa robot can be found at the boundaries of the range of motion. Here at least one of the arms reaches is maximally outstretched position. This means that the end-effector velocity goes though zero while the actuator joints are moving.

RO singularities also exist in the Hexa robot. The singularities however are more difficult to find. For simplicity reasons the arms of the Hexa are assumed to pair at one point on the platform. As each arm constraints the motion of the platform in the length direction of the lower arm, the translation of the connection point in the plane normal to this arm is free. By connecting two of these arms at one point, the connection point can only translate over the intersection of the two planes. If this motion cannot be counteracted by the other pairs, we have a singular RO pose. This is one-dimensional freedom of a pair. For the true Hexa the two arms do not intersect at the platform, but the reasoning in understanding the singularities are similar, but more intricate. Currently two types of RO singularities are identified. Figure 34 left and center show the two types of RO singularities.

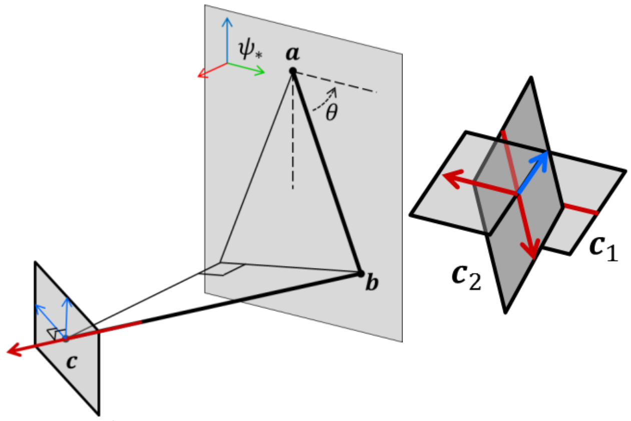


Figure 33. *Left* shows the translational constraint (red) and translational freedom (blue) of one leg. *Right* shows the intersection of two legs at one point and the resulting freedom.

The first singularity is called *planar RO*. This happens when an arm pair is in plane with the platform, as seen in Figure 34 left. This results in an out of plane freedom of the connection point and a rotational freedom of the platform around the connection line of the two other pairs.

A *pencil RO* happens when the end-effector is rotated around the z-axis until all pair freedoms are tangential to the z-axis of the platform. Now, the actuators cannot support a wrench (force and couple) around the z-axis. This singularity is also located at the outside of the workspace. Such that during normal operation this will have no influence.

One more singularity deserves attention. This singularity has been found only recently. This *RRR serial singularity* occurs when a forearm is normal to the end-effector plate. Then two of the three revolute joints are inline, resulting in a loss of mobility in the direction normal to the plane trough the three revolute joints, as can be seen in Figure 34. This means the RRR joint does no longer behave as a spherical joint. Close to this, singularity the velocity transmission of the RRR joint degrades, leading to higher stresses inside the joints and links. This singularity cannot be found using the condition number of the Jacobian because there the RRR joint is modeled as perfect spherical joint. This singularity is quite a problem as it occurs in the middle of workspace of the robot. The singularity explains strange bumps and noise

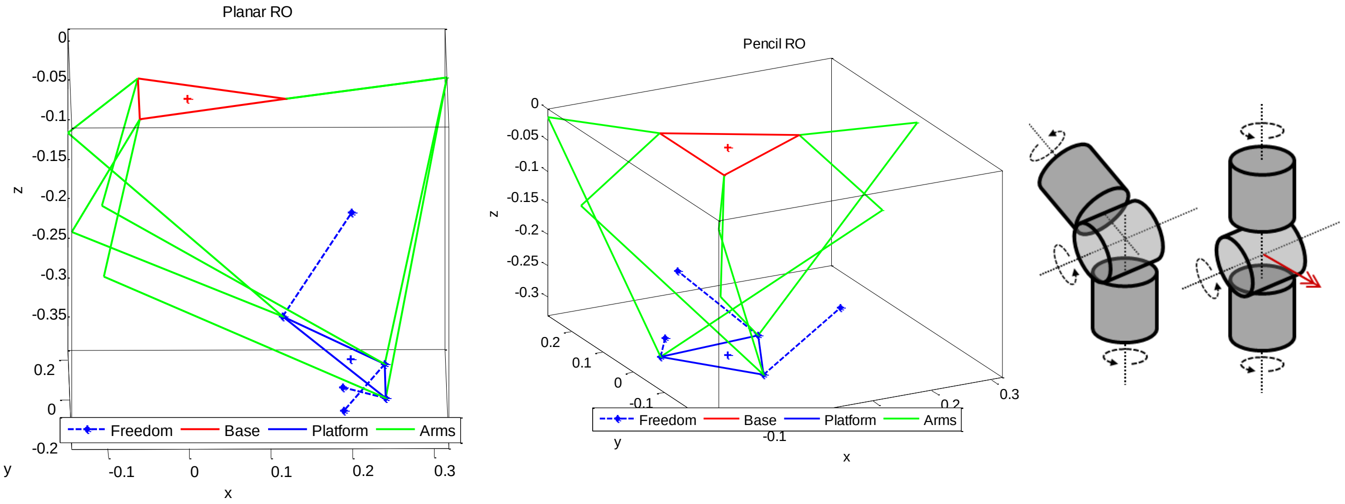


Figure 34 Here the different types of singularities are shown. *Left* shows the planar RO singularity with one pairwise arm freedom (dashed blue) normal to the platform. *Middle* shows the pencil RO singularity with in dashed blue the pairwise freedom of the connection point which results in an axial (z) screw motion. *Right* shows the RRR singularity in which there is no rotation possible around the red axis.

observed during otherwise smooth motion.

Currently, no method is implemented to find and avoid this singularity. A solution might be to send a little offset in the robot pose in order to steer around the singularity. This will induce some pose error but avoids wear of the robot and shaking behavior. During a redesign of a new robot, the joint should be placed at angle to move this singularity outside the workspace.

V. Auto collision detection

In certain poses of the Hexa robot internal collisions between arms can occur. These are called interferences. In order to avoid these poses, the distance between the arms should be kept higher than the radii of the interfering segments. An emergency stop should be performed when the segments are too close to each other. In this section the procedure to calculate the segment distance is explained.

1) *Minimal Distance between two infinite lines*

First, the general case of the calculating the distance of two infinite lines is explained. This procedure uses the property that shortest distance vector between two infinite lines is orthogonal to both lines.

There are two lines defined as shown in Figure 35 with each a point $\boldsymbol{a}$ $(\boldsymbol{a}_1, \boldsymbol{a}_2)$ and a point $\boldsymbol{b},$ $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ on the line. On both lines there is third point $\boldsymbol{c}$ $(\boldsymbol{c}_1, \boldsymbol{c}_2)$ which is the beginning and end of the shortest distance vector between the two lines. The points $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ are parameterized by scaling factors s_1 and s_2.

$$\begin{aligned} \boldsymbol{c}_1 &= \boldsymbol{a}_1 + s_1 \boldsymbol{u}_1 \\ \boldsymbol{c}_2 &= \boldsymbol{a}_2 + s_2 \boldsymbol{u}_2 \end{aligned} \tag{105}$$

With line direction vectors

$$\begin{aligned} \boldsymbol{u}_1 &= \boldsymbol{b}_1 - \boldsymbol{a}_1 \\ \boldsymbol{u}_2 &= \boldsymbol{b}_2 - \boldsymbol{a}_2 \\ \boldsymbol{w}_1 &= \boldsymbol{a}_2 - \boldsymbol{a}_1 \end{aligned} \tag{106}$$

Here $\boldsymbol{w}_1$ points form $\boldsymbol{a}_2$ to $\boldsymbol{a}_1$. Similarly, a vector for the intersection line can be defined.

$$\boldsymbol{u}_3 = \boldsymbol{c}_2 - \boldsymbol{c}_1 \tag{107}$$

Expanding (107) using (106) we can write

$$\begin{aligned} \boldsymbol{u}_3 &= \boldsymbol{a}_2 + s_2 \boldsymbol{u}_2 - \boldsymbol{a}_1 + s_1 \boldsymbol{u}_1 \\ &= \boldsymbol{w}_1 + s_2 \boldsymbol{u}_2 - s_1 \boldsymbol{u}_1 \end{aligned} \tag{108}$$

Since the orthogonality property of line 3 with respect to both lines, we can write

$$\begin{aligned} \boldsymbol{u}_1 \cdot \boldsymbol{u}_3 &= 0 \\ \boldsymbol{u}_2 \cdot \boldsymbol{u}_3 &= 0 \end{aligned} \tag{109}$$

By inserting (108) in both equations of (109) we end up with a system of equations.

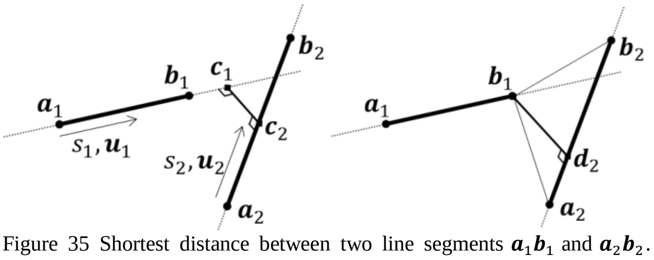


Figure 35 Shortest distance between two line segments $\boldsymbol{a}_1\boldsymbol{b}_1$ and $\boldsymbol{a}_2\boldsymbol{b}_2$. Left shows the general case in which the shortest distance may lay outside the segment. Right shows the distance calculation in that case using the altitude of a triangle $\boldsymbol{a}_1\boldsymbol{b}_1\boldsymbol{b}_2$

$$\begin{aligned} \boldsymbol{u}_1 \cdot \boldsymbol{w}_1 - s_2(\boldsymbol{u}_1 \cdot \boldsymbol{u}_2) + s_1(\boldsymbol{u}_1 \cdot \boldsymbol{u}_1) &= 0 \\ \boldsymbol{u}_2 \cdot \boldsymbol{w}_1 - s_2(\boldsymbol{u}_2 \cdot \boldsymbol{u}_2) + s_1(\boldsymbol{u}_2 \cdot \boldsymbol{u}_1) &= 0 \end{aligned} \tag{110}$$

From this s_1 and s_2 are solved

$$\begin{aligned} s_1 &= -\frac{q_1 q_3 - q_2 q_4}{q_1 q_3 - {q_2}^2} \\ s_2 &= -\frac{q_1 q_2 - q_3 q_4}{q_1 q_3 - {q_2}^2} \end{aligned} \tag{111}$$

With coefficients:

$$\begin{aligned} q_1 &= \boldsymbol{u}_1 \cdot \boldsymbol{u}_1 \\ q_2 &= \boldsymbol{u}_1 \cdot \boldsymbol{u}_2 \\ q_3 &= \boldsymbol{u}_2 \cdot \boldsymbol{u}_2 \\ q_4 &= \boldsymbol{u}_1 \cdot \boldsymbol{w}_1 \\ q_5 &= \boldsymbol{u}_2 \cdot \boldsymbol{w}_1 \end{aligned} \tag{112}$$

Now the minimal distance (l_{min}) from between the two lines can be calculate by taking the distance between $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$.

$$l_{min} = \|\boldsymbol{c}_2 - \boldsymbol{c}_1\| \tag{113}$$

2) *Minimal Distance between two line segments*

These scaling factors give the position of $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ for which the length of line 3 is minimal. However the scaling factors can be more than 1 or less than 0 denoting that the shortest distance is outside segment $\boldsymbol{a}_1\boldsymbol{b}_1$ and/or $\boldsymbol{a}_2\boldsymbol{b}_2$. To find the shortest distance between two segments, the procedure has the be expanded.

Consider there are three cases. The minimal distance points between the infinite lines through $\boldsymbol{a}_1\boldsymbol{b}_1$ and through $\boldsymbol{a}_2\boldsymbol{b}_2$ are:

1. On both segments;
2. On one; or,
3. Not on any segment

In case 1, the minimal distance can be calculated from the previous procedure.

In case 2, the minimal distance is only on one of the segments. The minimal length is defined as the minimal distance from the one of the end-points to a point on the other segment (or end-point). These three points ($\boldsymbol{a}_1$, $\boldsymbol{b}_1$, and $\boldsymbol{c}_2$, or $\boldsymbol{a}_2$, $\boldsymbol{b}_2$, and $\boldsymbol{c}_1$) form a triangle from which the distance can be calculated using the altitude of the triangle. Because it is not clear on forehand which point will be the end-point, both altitudes have to be calculated. Now we have to select the smallest distance point, which still falls inside the segment.

In case 3, no minimal distance point is on the segment. Then both end-points closest to the minimal distance point are taken as the minimal distance points.

3) *Implementation*

For the Hexa, interference can occur between two arms pairs that are connected to the same side on the base. This is arms 1 and 6, 2 and 3 and 4 and 5. Three segments in each arm that can cause interference with a segment on the other arm. The upper arm with elbow plate, the elbow plate with the forearm, and the elbow plate with the elbow plate of the other arm. This gives five possibilities per pair, and a total of 15 interference combinations. These combinations are to be checked continuously for interference.

When the distance between these segments is less than the safety distance plus the radii of both segments an emergency stop is executed. The radii for the segments are calculated by circle in which the whole segment fits. For the forearm, the forearm-wrist is taken as the crucial part.

Since the interference calculation is too time consuming, only one interference possibility is checked during one time step. Fifteen time steps are needed to calculate all the interference distances of the Hexa robot. Using a sample time of 0.001 ms, it takes maximally 0.015 second to detect interference. For the maximal velocities of the robot, a safety distance of 5 mm is sufficient.

VI. Conclusions

In this chapter, the kinematic relations of the Hexa robot where shown. The relation between the end-effector position and the joint angles has been derived. For the forward relation, an iterative procedure is used to calculate the end-effector from the joint angles. The kinematic description is used to calculate the velocity input-output relations. These relations can be used to calculate the static force transmission of the system. There exist certain poses in which these relations degrade and the robot is found to be in a singularity. The position of these singularities are identified and found to be outside the workspace required for TMS during treadmill walking. There exist however a singularity inside the workspace which was not seen earlier. This RRR singularity requires a redesign of the platform or joints. A method to detect internal collisions has been proposed here.

Chapter 6.Controller design

In this thesis, several aspects of controlling the robot are already discussed. This chapter shows how these parts come together to steer the robot over the stimulation site.

The controller can be divided into several subsections. The *head tracking* part to calculated the position of the stimulation site with respect to the robot, the *force controller* that aims to keep a constant contact force between the skull and the robot and a *position controller* to control the robot to the desired position. In this chapter, the force and the position controller expanded. More background details and result are shown. In the last section some words on the actual implementation of the controller into software are given.

I. Spring force controller

To ensure a safe and pleasant contact a spring is placed between the manipulator and the coil For safety reasons, the systems makes an emergency stop if contact force too larger or the contact is almost lost.

For soft interaction contact, we want to control the contact force between the subject and the coil. However, the contact force cannot directly be measured, as a sensor cannot be placed close to the coil. Instead, the spring deflection is measured and the spring force is estimated. Care should be taken in designing a controller, as the contact force is dependent on the mass, gravity and the damping of the spring this requires knowledge of the pose, velocity and acceleration that in turn, are noisy signals.

Without force control, the properties of the spring mass system and the delay, such as induced by the marker filter, result in a non-constant contact force during head tracking. The mass of the coil induces acceleration dependent force on the head. The acceleration force is not measured by the spring system and can therefore cause the contact lost without sensing this. To prevent contact loss during high velocity tracking, pensioning is required to be able keep to always a contact even under varying contact force. The aim of the force control is to make the contact force more constant and to reduce the influence on measurement noise and placement error.

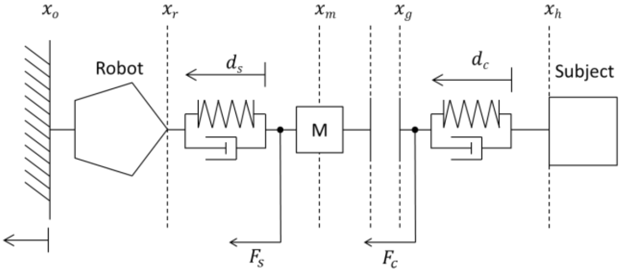


Figure 36. Contact definition. With the robot the spring displacement (d_s) and skin compression (d_c). The pose are the head pose (x_h) position of the gap (x_g), position of the mass/coil (x_m), robot pose (x_r) and the base (x_0). The relevant forces are the forces exerted by the spring system (F_s) and by the contact (F_c)

1) *Head contact model*

To gain insight in the system and to be able to design a controller a model is described here. It uses some simplification to be able to design the controller.

The mass of the coil is around 2.5 kg, dependent on how the TMS wire is connected to the system. The damping is 50 Ns/m and the spring around stiffness is 5200 N/m. Since nonlinear springs are used, this stiffness increases during deflection.

In Figure 36 the definition of the model is given. We treat the system as an actuator, the robot in contact with a spring mass damper system with a possible gap and again the spring damper of the subject's skull. It should be noted that the spring-damper of the skull is of course much stiffer than that of the linear guide. For simplicity reasons the contact impedance is assumed to be infinitely high. Then the position of the mass is equal to the position of the head. $d_c = 0 \rightarrow x_m = x_h$

First, a force balance is made. This is all done without lengths. The force in the spring and the contact force are proportional to the acceleration of the mass. We assume that there is contact such that x_g has no importance.

$$m\ddot{x}_m = F_s + F_c \tag{114}$$

The contact force and the spring force are depended on their relative deflection and deflection velocity.

$$F_s = (k_s + c_s s)d_s \tag{115}$$

The deflection is then again depended on the robot position and mass, the deflection is dependent on the position of the head with respect to the mass. This only applies when there is a contact.

$$d_s = x_r - x_h \tag{116}$$

This leads to a force balance of

$$F_c = (m\,s^2 + c_s s + k_s)x_h(t) - (k_s + sc_s)x_r(t) \tag{117}$$

2) *Closed loop model*

Figure 37 shows the complete closed loop scheme. Here the robot plant, marker filter complete the closed loop. Including the position controller x_r into equation (117), a more general input output relation can be derived based. The marker tracking system (F_m) is modeled as a low pass filter of 2 Hz. The robot plant P_r is also modeled as a low pass filter with a bandwidth of 20 Hz.

In case no contact control is used ($C = 0$) the contact force is:

$$y = (H_h + H_s P_r F_m)x_h \tag{118}$$

With:

$$H_h = \begin{bmatrix} m\,s^2 + c_s s + k_s \\ -1 \end{bmatrix}, \tag{119}$$

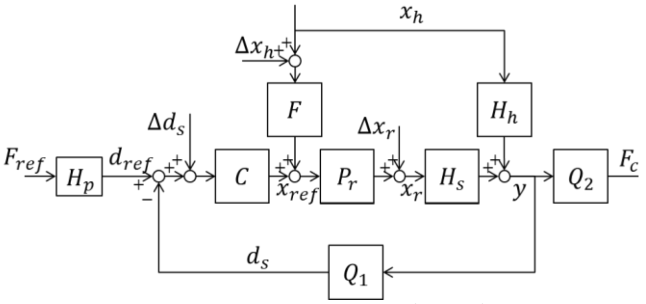


Figure 37. Contact Force control scheme. The force reference gives the desired deflexion (H_p) The controller (C) calculates the desired robot pose (x_{ref}) based on desired contact force (F_{ref}) and spring compression (d_s). Here P_r and H_s are the dynamics of the robot and of the spring, respectively. H_h is the contribution of head displacement on the contact force. $y = [F_c \quad d_s]$. The selection matrices are Q_1 and Q_2. The head position x_h is filtered by the head tracking system F. The disturbances are $\Delta x_h, \Delta x_r, \Delta d_s$ which are the head tracking noise, robot error, and displacement sensor noise, respectively.

$$H_s = \begin{bmatrix} -c_s s - k_s \\ 1 \end{bmatrix}, \qquad y = \begin{bmatrix} F_c \\ d_s \end{bmatrix}$$

Now by closing the contact force control loop we obtain equilibrium:

$$\begin{aligned} y &= (H_h + H_s P_r F_m) x_h \\ &+ H_s P_r C \big(H_p F_{ref} - Q_1 y\big) \end{aligned} \tag{120}$$

From which the contact force can be derived as a function of reference force (F_{ref}) and the head position (x_h)

$$\begin{aligned} F_c &= Q_2 (I + H_s P_r C Q_1)^{-1} \cdot \\ &\Big((H_h + H_s P_r F_m) x_h + H_s P_r C H_p F_{ref}\Big) \end{aligned} \tag{121}$$

The loop gain of the force controller is.

$$L = Q_1 H_s P_r C \tag{122}$$

3) *Requirements*

The aim of the contact force controller is to increase the bandwidth and amplitude for which the system maintains contact. Higher bandwidth force control bandwidth also expects to result in higher marker noise suppression and reduced required pretension.

The frequency content of the head motion is assumed to be of x_h of 0.05 m of a low pass bandwidth with is assumed to have a cutoff frequency of 1.5 Hz. We require petitioning to assure contact during this motion. In Figure 38 the force amplification from mm to force is given for the no marker tracing, marker tracking only, and by adding tracking.

Without force control, the contact force during normal operation can reach up to 125 N. In order to prevent contact los a pretention of at least that value is required. With better tracking this value can be reduced, however still about 50 N is required.

The requirements are not specified in bandwidth of tracking, as the force reference is a DC value. We like to have minimal pretension for a bandwidth up to 10 Hz.

4) *Controllers*

The basic control scheme employed Figure 37 gives an overview of the different transfer functions. The goal of the controller is to end up with a system behavior with a unitary DC gain of the force reference and suppress the output disturbance as much as possible. Here there are three controllers (PD, PI, PID) evaluated without marker tracking and the PID controller is evaluated with three different marker tracking filters.

It can be seen that the PID controller has the most constant force amplitude at low frequency. This comes at the cost of using a differentiating action, which increases the measurement noise. In addition, the attenuation at lower frequencies increases the changing force at higher frequencies.

For the prefiltering, it can be seen that increasing the tracking bandwidth the influence of the inertia on the head becomes more prominent as the anti-frequency of the mass—spring-damper system is removed. This means low-frequency tracking comes at a cost at higher frequencies.

5) *Implementation tests*

The PID controller plus marker tracking can be implemented into the robot. However, these tests are not conducted yet. Currently only the PI ($p_k = 0.5, p_i = 10$) results are shown. For the evaluation, it is quite difficult to get proper input motion to compare the system in real live. We cannot use the same motion and it is quite difficult to assert the contact force. We can only report the spring displacement and report contact feel. Furthermore, we cannot measure the contact force directly and we use a model approximation for that.

6) *Conclusion and discussion*

The closed loop spring force control of the TMS robot showed to give a more constant contact force. Several controllers were suggested, which can reduce the contact force during motion. However, this is also quite limited

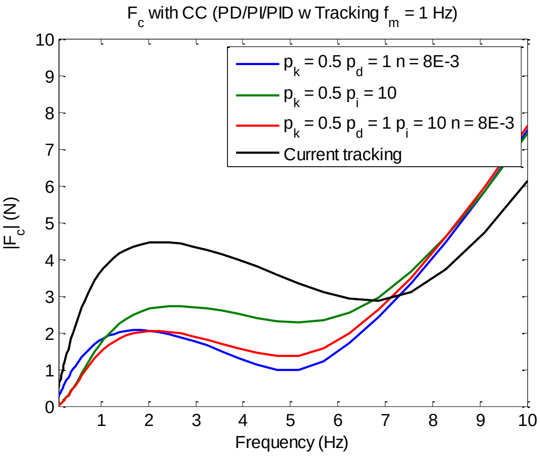


Figure 38 Contact force amplification of 1 mm motion with closed loop control and tracking.

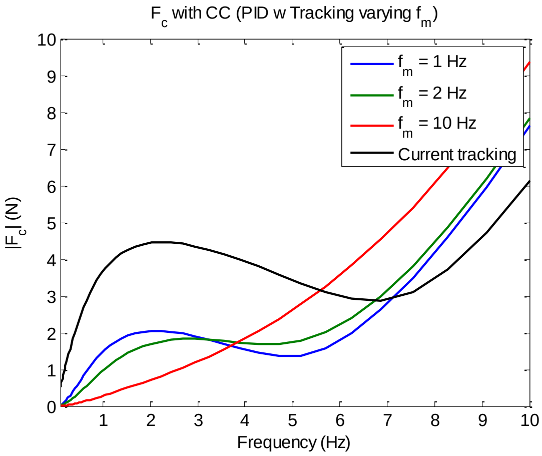


Figure 39 Contact force amplification of 1 mm motion with closed loop control and tracking for varying prefilter cut off frequency

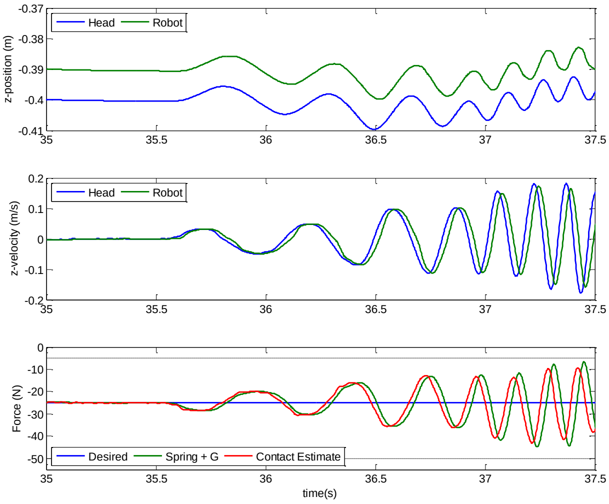


Figure 40 Actual measurements of force control.

since the inertia is not cancelled and will always show up at higher frequencies. Therefore, there is a tread-off between low and high frequency attenuation of the signal. More investigation in this trade-off is required to find an optimal control strategy.

The implementation has been limited to PI control without positioning feed forward. In the future, this has to be investigated more accurately. In addition, this has to be tuned to feel experience of the subjects.

The response at higher frequencies can be improved by adding mass canceling. This however requires second differentiation of the potentiometer signal, which is in practice not feasible. Therefore, accelerometers at the skull or at the coil are suggested to accurately measure this. This requires also careful placement and signal processing to incorporate the influence of the gravity.

II. Position Controller

The position controller is responsible for finding the motor voltages steering the robot in the proper pose depending on the desired end-effector pose and the current robot pose.

There are several controller designs possible. One can think of a control loop in end-effector space. This allows the tuning of the parameters also in the end-effector space meaning the gravitational disturbance can be rejected more easily as it is constant. The problem however is that the direct kinematic model, mapping the joint space to the end-effector space, cannot be written in closed loop form and therefore requires an iterative procedure to find the correct pose. This will induce velocity dependent errors in the control algorithm if this is used for pose control.

The position controller is used in the joint space. This result in six identical PID controllers on the motor angle.

Another option is to use an inner loop on the current sensor to employ current control. The current to the motor is linear with the force of the motors and therefore allow the use of a dynamical model to cancel out non-linear effects in the controller.

As will be shown later, the main cause of controller error is the friction in the motors and the gears. As the friction force is approximately linear with velocity, it can be canceled using a friction feed forward.

In the next section, the bandwidth of joint space PID control will be compared with the same controller with friction feed forward. To that end, a swept sine motion profile in one direction the end-effector space will be tracked and the response of the robot in the other directions will be measured. This will be done for all six dimensions. The amplitude of the swept sine is chosen such that the amplitude of the joint motion is comparable.

In Figure 41the resulting swept sine is shown. Here only PI controller is used with the following controller values. It can be seen that the amplitude of tracking is high up to. This shows that the bandwidth up to 15 Hz is reachable for the robot. It can be seen that there is some crosstalk between the directions. This has to do that the inaccuracies in tracking in the joint coordinates lead to a small motion in the other directions. It can be seen that the crosstalk is not the same for all directions. This has to do with the kinematic coupling at that pose.

There is a limitation of the controller not shown here. The performance changes over the workspace of the robot. Since in the middle of the workspace the force and velocity transmission is equal for all the joints the performance is there the highest. However outside the central line this distribution changes and hence the performance of the system. Currently no pose dependent controller has been conceived. Therefore, the joint controller should be tuned conservative to avoid instabilities over the complete workspace.

Currently a position controller has been designed which can reach a bandwidth of more than 15 Hz which is sufficient for the application of TMS control. It can be seen that the robot can reach a higher bandwidth than mechanically admissible. It has been shown that the use of friction compensation improves the bandwidth over only PID. This high bandwidth has the downside that it passes through the high frequency noise at the input.

III. Implementation

The realtime part of the controller is implemented at an xPC target of Matlab Simulink. This works using two PCs of which the first is responsible for the realtime execution while the second one allows a UI interface and the programming toolbox. The second PC, the Host, compiles

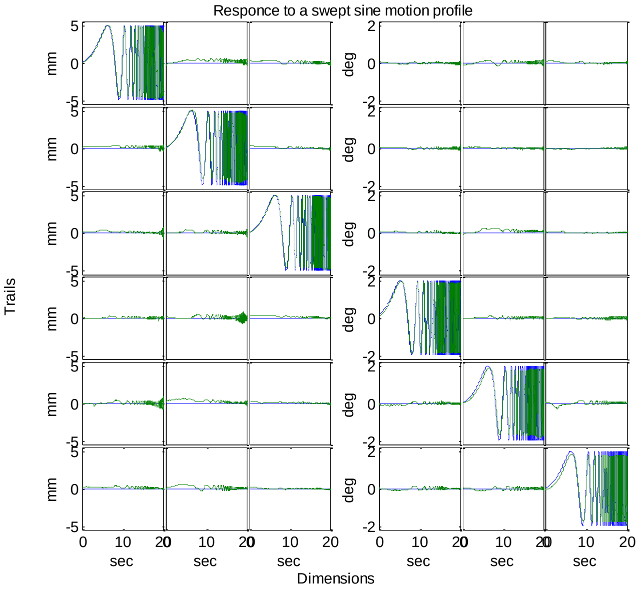


Figure 41. The response of a swept sine motion profile without velocity feedforward.

the Simulink model to c-code, which can be executed on a realtime basis on a DOS Target. The Simulink model on the host can function as user interface and can be used to change values on the fly and read out data. The UI however is limited and gives very basic visual information through engineering diagrams.

1) Overview of Simulink diagram

In (35) an overview of the Simulink scheme can be found. Several subsections are identified.

a. User Interface,
b. State controller,
c. Plot and log,
1. Head tracking,
2. Path generation,
3. Controller,
4. DAQ,
5. Sensor Data Processing.

The subsections identified with the letters: the User Interface, State controller and Plot and log, are used for controlling and reporting the robot at a meta or highest level. In *User Interface* all the tunable variables are given. On set for example the homing position, the operating mode and start the head tracking. In the future, this UI will be implemented into a different program to run independent from the xPC implementation. This would allow a much smoother intuitive interface with the robot. This new UI will communicate with this sub process and set the variables identified here. The *State controller* contains all the high-level operations such as safety features, operating modi. In the next sections, these operating modi are shown. The *Plot and log* sub block allows the readout of parameters.

The sub blocks identified with numbers are the low-level processes. Here all the calculation, communication and control are performed. The first block, *Head tracking* calculates the head pose from the marker data here the OP and MKF methods are implemented including all the relative transformations. The *Path generation* block interpolates the required set point of the robot if required. For example, during homing state the path generation allows a smooth trajectory from the current pose to the homing pose. In *Controller* the force and position controller are implemented. It calculates the desired motor voltages depending on the desired pose and force of the robot. The *DAQ* (data acquisition) block performs the input output of the system. Here the encoder values are read and the desired motor voltages set. In the *Sensor Data Processing* the calibration of the sensors are implemented. Here also the current pose of the robot are calculated.

2) High level controller

The state controller determines in which operating mode the robot currently is. There are currently nine modes and are implemented. In Table 5 these operating modes are listed and their function described. The operating modes are controlled by a state machine implemented in flow chart. This scheme can be found in Figure 42.

First, the robot starts in the off phase. If power on button is pressed in the UI, the robot starts calibration. After this is finished the robot moves to its homing position. From here, it can go to operational. If no subject is present and no contact force is detected the system can move in free mode. This can be used for diagnostic and demonstration purposes.

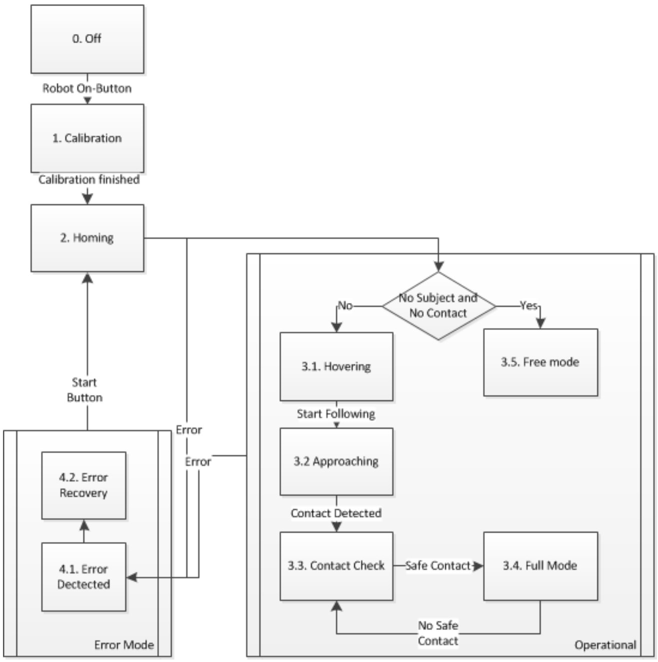


Figure 42 – State flow of the state machine. For clarity, the arrows going to the off phase are omitted.

For the head tracking the robot first goes to hovering mode. This will place the coil over the stimulation site without making contact. If the operator decides to start making contact, the robot will approach the stimulation site normally until contact is detected. Then it will go into a contact check mode. This means that the robot will track the head but the contact force is not in the safe bounds. If the contact force is within the safe bounds, the power to the robot is finally turned on maximally and the robot is allowed to move at full speed.

3) User Interface

Currently a very rudimentary UI interface has been implemented into Simulink. This means that the user has to have Simulink connected to the xPC target in order to change parameters or give commands. A draft of a user interface has been made in python. Using python2xPC

Table 5. Listing of the operating modes of the robot

Operating mode		Function
0.	Off	Power to robot is off. Waiting to be switched on.
1.	Calibration	First phase is start-up calibration of the sensors and calculating robot posture
2.	Homing	Power to the robot is minimal on. Moving to initial position
3.	Operational	System is ready to be used.
3.1.	Hovering	The robot moves over the stimulation site, minimal power and no contact
3.2.	Approaching	The robot approaches stimulation site in a perpendicular path to the head minimal power and minimal contact
3.3.	Contact Check	When the contact force is within contact check, limits the power. Safety measure
3.4	Full Contact	When the contact force is within safe bounds robot is in full power mode. Allows fast tracking
3.5.	Free Mode	When no subject is present and no contact force is measured the robot is allowed to move with full power. Diagnostic purposes
4.	Error	Error modes.
4.1	Error Detected	When an error is detected the robot power is shut down. Waits for a cool down periode to move to recovery
4.2	Error Recovery	Waits for the operator to start the system again.

toolbox developed for the lopes 2 the xPC target could be controlled directly without the need of a Simulink or Matlab instance to be open. This will reduce the load on the windows PC resulting in better communication with the Visualeyez system and a higher safety since parameters are shielded from the user. In addition, the ease of use will increase since the Simulink UI is only quite understandable for engineers.

IV. Discussion and Conclusion

In this section, various parts of the controller of the TMS robot are shown. The contact force controller has been investigated. It is shown that the bandwidth of tracking can only be limited improved by adding the contact force controller. This is due to the anti-frequency of the mass spring system. This frequency can only be moved by actually measuring and controlling the acceleration of the mass. This requires an extra sensor to be placed on the coil or the head. It might be fruitful to see if this force measurement instead might be used in the Kalman filter to gain a better pose and contact force estimation.

The position controller on the joint space has shown to achieve quite a large bandwidth. Together with friction canceling, a bandwidth of more than 15 Hz has been achieved. This comes however at a cost of higher sensitivity to measurement or input noise. In addition, there might be some resonance issues present at different places of the workspace. It might be worthwhile to investigate more pose depending controller to cancel out these pose depended contact force. More tuning might result in a less aggressive controller that has a more fluent following behavior but less bandwidth.

The implementation of these controllers into Matlabs xPC requires the use of a quite intricate High-level controller in which the state machine and safety are taken care off. This resulted in quite large amount of states and modes in which the robot can be operated. More work is required to simplify this and allow a more simple interaction. For the same reason a more intuitive user interface has been proposed.

Chapter 7.General conclusions and discussion

In the previous chapters, the design of a TMS robot is presented and evaluated. The TMS robot is designed to apply TMS during treadmill walking. The coil is positioned by a six-DOF parallel manipulator. The coil is pressed against the skull by a spring system for compliant contact. The manipulator is supported by a circular arc frame. The motion of the head is measured by a 3D optical tracker system. To guarantee safety of the subject, an external safety system monitors the operation of the robot.

To assure the safety of the subject under all conditions a FMEA was conducted to find all possible malfunctions and risk factors. Throughout the design process, the safety design has been a paramount priority. Resulting in safety measures at all levels of the system. These safety measures include an advanced external safety system to redundantly check the functionality of the primary controller and sensors. A safety spring system is placed between the robot and the subject. This allows monitoring of the contact force such that a timely emergency stop can be made before the contact is lost. This prevents a high energetic collision, as the robot is only allowed to move when the contact is guaranteed. The use of a parallel manipulator structure gives the system minimal moving mass, and therefore energy while limiting the workspace of the robot to the minimal.

The robot is found to be able to follow the head during a slow motion. The noise on the head measurement system required strong filtering to allow comfortable interaction between the robot and the subject. This filtering strongly limited the bandwidth of the system. In order to solve these problems a modified Kalman filtering is proposed.

I. Recommendations

A. *Current robot*

The tracking motion is generated by a swift and accurate Hexa manipulator. The supporting frame is not stiff enough to reject this powerful motion. This leads to undesired vibrations at the end-effector and a reduction in accuracy. The supporting frame spans an effective length of more than 2 meter. To achieve the necessary stiffness over such a length, a complete redesign of the supporting frame is required. Another drawback of the current design is the minimal possibility to adjust the support height of the frame for different subjects.

The controller algorithm of the robot allows the placement of the end-effector with an accuracy exceeding 1 mm. For TMS applications, this is sufficient. The main improvement lies in the improvement of the head-tracking algorithm. The measurement noise is not sufficiently rejected. This can be achieved by incorporating a Kalman filter and adding accelerometers or IMU to measure the head acceleration and orientation. Effective sensor fusion algorithms exist to allow realtime data processing. In Comparison between two realtime tracking methods for robotized TMS such an algorithm is presented and implemented into the controller of the TMS robot. More evaluation is need and the extension with inertial sensors is to be added.

The Visualeyez cameras communicate via Visualeyez software on the windows PC. This has the issue that it cannot run realtime. In addition, the communication between the windows PC and the xPC target via UDP result in time delay and package loss. To improve this, the read-out of the Visualeyez cameras should be done on the xPC target. This requires the software to be programmed in C. There is support from Visualeyez that allows this.

The pose controller is currently implemented in the joint space. This allows a relative simple controller scheme of six identical SISO systems, .Since the relative large gearing after the motors this gives sufficient results. For further improvements, the pose dependency of load and velocity transmission should be included into the controller. This can be done with a MIMO scheme.

Currently the user interface of the robot is given by Simulink model. This intricate model does not work fluently or intuitively. It is not useable by non-technical educated operator. This also poses a safety risk since the overwriting of certain parameters can lead to undesirable movement of the robot. To be able to test the TMS robot for a longer period a graphical user interface (GUI) is to be designed. The GUI should speed up the initiation process of the head tracking considerable, as its interface is more intuitive than the Matlab scripts required for registration of stimulation site and cranial landmarks.

The psychological aspects of the robot have determines whether a subject is willing to use the TMS robot. This mainly has to do on how safe and trustworthy the robot is perceived. It is a strong robotic manipulator which can exert 200 N on the subject, where there no safety measures. The protective hood around the moving parts, and the soft interaction feel might improve the credibility of the robot. It might be interesting to study the safety perception of this robot. For a commercial version of this product, it is even more important.

B. *Future design*

The parts used for the robot are of very high quality. This comes at a financial price. For example, the motors and gearboxes alone cost 16.000 euro in total. For an experimental research tool, these high quality parts are required. If this robot is to be implemented in a more commercial setting such as product of ANT-Neuro, the system has to redesigned to reduce the cost and complexity of the system.

The electrical circuitry of the robot is designed incrementally. This means that several add-ons to the electronics where made during the process of designing. There are many sensors and therefore also a lot of wires which are bundled in to several cables. This however is quite complicated and difficult to adjust and troubleshoot. An improvement can be made by using bus like allocated I/O such as ethercat modules. This reduces the amount of wires and cables and makes the system therefore more robust and fail-safe.

The spring safety system has one elastic direction. In the other directions, it has a rigid connection. The spring force is measured only in that direction. This leads to a soft contact feel in the normal direction to the coil. In the other directions, this contact is very stiff. This also has to do with the friction between the coil and the skull. To let the subject feel free in those directions, a multiple DOF spring system is desirable. Such a system can mechanically be similar to a remote center of compliance used for peg in hole insertion tasks of robots. The spring stiffness in the other directions can be much higher than in the normal direction. If the spring deflection can be measured this can result in a much softer interaction feel in the other direction.

C. Evaluation

The spring contact control improves the soft interaction feel of the robot. The functionality of the spring force controller cannot be evaluated fully, without the use of an external force sensor and a manipulator that can accurately make a repeatable motion such that the contact force can be compared with and without compensation. This influence is worsened by the fact that the contact force at higher frequencies is strongly dependent on the inertia of the coil. One should also notice that the head also has in impedance as it has the tendency to move with increasing contact force. However, the lead of the contact force to the measured spring force means that the contact force can be lost earlier than the measured spring force.

The electronic safety system is designed to monitor the function of the controller and the sensors. This safety system measures the spring force and compares redundant sensors to check for sensor failure. This safety system is not yet evaluated systematically. This requires the checking of all possible failure modes to see if the safety system has the proper reaction.

The full evaluation of the TMS robot requires the comparison of the effectiveness of manual TMS with robotized TMS. From literature and experience, it is known that the effectiveness of manual TMS is very variable. This prohibits the real validation of the TMS robots stimulation accuracy. For the validation of the TMS robot is desirable to show that it is at least as effective as manual stimulation under similar circumstances.

The proof that the robot can stimulate the head during treadmill walking has yet to be given. This mainly requires a better noise suppression.

II. Outlook

Comparison of the TMS robot to commercially available TMS supports and robot shows that the robot is faster, more accurate and versatile than the existing solutions. A commercial redesign of this TMS robot has potentially a technical edge over the competitors.

Research has shown that TMS can be used to relieve a wide variety of neurological pathologies and that it can be used to investigate the neurological behavior inside the brain. The variability of the stimulation effect, limits the usability of the technique. Part of this variability might be caused by placement error. The accuracy of this TMS robot might help to identify this influence and potentially lead to better TMS efficacy.

When the TMS robot can clinically be applied to motor relearning after stroke, the potential of TMS for improving motor training can be investigated. If the findings of previous research will hold, then the TMS robot has a great potential to be used as a revalidation tool in a clinical setting. The motor training of stoke patients can then be improved and the subjects can have higher functional outcome of training and therefore a better quality of live.

References

[1] E. M. Wassermann and S. H. Lisanby, "Therapeutic application of repetitive transcranial magnetic stimulation: a review.," *Clin. Neurophysiol.*, vol. 112, no. 8, pp. 1367–77, Aug. 2001.

[2] J. L. R. Martin, M. J. Barbanoj, T. E. Schlaepfer, E. Thompson, V. Pérez, and J. Kulisevsky, "Repetitive transcranial magnetic stimulation for the treatment of depression. Systematic review and meta-analysis.," *Br. J. Psychiatry*, vol. 182, no. March, pp. 480–91, Jun. 2003.

[3] L. W. Forrester, D. F. Hanley, and R. F. Macko, "Effects of treadmill exercise on transcranial magnetic stimulation-induced excitability to quadriceps after stroke.," *Arch. Phys. Med. Rehabil.*, vol. 87, no. 2, pp. 229–34, Feb. 2006.

[4] M. Camus, J. Pailhous, and M. Bonnard, "On-line flexibility of the cognitive tuning of corticospinal excitability: a TMS study in human gait.," *Brain Res.*, vol. 1076, no. 1, pp. 144–9, Mar. 2006.

[5] M. Bonnard, M. Camus, T. Coyle, and J. Pailhous, "Task-induced modulation of motor evoked potentials in upper-leg muscles during human gait: a TMS study," *Eur. J. Neurosci.*, vol. 16, no. 11, pp. 2225–2230, Dec. 2002.

[6] P. Talelli, R. J. Greenwood, and J. C. Rothwell, "Arm function after stroke: neurophysiological correlates and recovery mechanisms assessed by transcranial magnetic stimulation.," *Clin. Neurophysiol.*, vol. 117, no. 8, pp. 1641–59, Aug. 2006.

[7] J. F. Veneman, R. Kruidhof, E. E. G. Hekman, R. Ekkelenkamp, E. H. F. Van Asseldonk, and H. van der Kooij, "Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation.," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 3, pp. 379–86, Sep. 2007.

[8] P. Kazanzides, "Safety design for medical robots.," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2009, pp. 7208–11, Jan. 2009.

[9] K. Fodero, H. H. King, M. J. H. Lum, C. Bland, J. Rosen, M. Sinanan, and B. Hannaford, "Control system architecture for a minimally invasive surgical robot.," *Stud. Health Technol. Inform.*, vol. 119, pp. 156–8, Jan. 2006.

[10] U. Laible, "A fail-safe dual channel robot control for surgery applications," *Saf. Sci.*, vol. 42, no. 5, pp. 423–436, Jun. 2004.

[11] S. F. W. Neggers, T. R. Langerak, D. J. L. G. Schutter, R. C. W. Mandl, N. F. Ramsey, P. J. J. Lemmens, and a Postma, "A stereotactic method for image-guided transcranial magnetic stimulation validated with fMRI and motor-evoked potentials.," *Neuroimage*, vol. 21, no. 4, pp. 1805–17, Apr. 2004.

[12] S. Narayana, P. T. Fox, N. Tandon, J. L. Lancaster, J. R. III, M. B. Iyer, and W. Constantine, "Use of neurosurgical robot for aiming and holding in cortical TMS experiments," *Neuroimage*, vol. 11, no. 5, Supplement 1, pp. S471 – S471, 2000.

[13] J. L. Lancaster, S. Narayana, D. Wenzel, J. Luckemeyer, J. Roby, and P. Fox, "Evaluation of an image-guided, robotically positioned transcranial magnetic stimulation system.," *Hum. Brain Mapp.*, vol. 22, no. 4, pp. 329–40, Aug. 2004.

[14] M. Finke, T. Fadini, S. Kantelhardt, a Giese, L. Matthaus, and a Schweikard, "Brain-mapping using robotized TMS.," *Conf. Proc. IEEE Eng. Med. Biol. Soc.*, vol. 2008, pp. 3929–32, Jan. 2008.

[15] L. Matthäus, P. Trillenberg, C. Bodensteiner, A. Giese, and A. Schweikard, "Robotized TMS for motion compensated navigated brain stimulation," *Brain*.

[16] L. Zorn, P. Renaud, B. Bayle, L. Goffin, C. Lebossé, M. de Mathelin, and J. Foucher, "Design and evaluation of a robotic system for transcranial magnetic stimulation.," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 3, pp. 805–15, Mar. 2012.

[17] C. Iglesias, J. B. Nielsen, and V. Marchand-Pauvert, "Corticospinal inhibition of transmission in propriospinal-like neurones during human walking.," *Eur. J. Neurosci.*, vol. 28, no. 7, pp. 1351–61, Oct. 2008.

[18] M. Schubert, a Curt, L. Jensen, and V. Dietz, "Corticospinal input in human gait: modulation of magnetically evoked motor responses.," *Exp. Brain Res.*, vol. 115, no. 2, pp. 234–46, Jun. 1997.

[19] D. Barthelemy and J. B. Nielsen, "Corticospinal contribution to arm muscle activity during human walking.," *J. Physiol.*, vol. 588, no. Pt 6, pp. 967–79, Mar. 2010.

[20] J.-P. Merlet, *Parallel robots*, Second. Springer-Verlag New York Inc, 2006.

[21] T. Wagner, J. Rushmore, U. Eden, and A. Valero-Cabre, "Biophysical foundations underlying TMS: setting the stage for an effective use of neurostimulation in the cognitive neurosciences.," *Cortex.*, vol. 45, no. 9, pp. 1025–34, Oct. 2009.

[22] H. Nollet, "Transcranial magnetic stimulation: review of the technique, basic principles and applications," *Vet. J.*, vol. 166, no. 1, pp. 28–42, Jul. 2003.

[23] M. C. Ridding and J. C. Rothwell, "Is there a future for therapeutic use of transcranial magnetic stimulation?," *Nat. Rev. Neurosci.*, vol. 8, no. 7, pp. 559–67, Jul. 2007.

[24] M. Hallett, "Transcranial magnetic stimulation: a primer.," *Neuron*, vol. 55, no. 2, pp. 187–99, Jul. 2007.

[25] N. Takeuchi, T. Chuma, Y. Matsuo, I. Watanabe, and K. Ikoma, "Repetitive transcranial magnetic stimulation of contralesional primary motor cortex improves hand function after stroke.," *Stroke.*, vol. 36, no. 12, pp. 2681–6, Dec. 2005.

[26] T. H. Emara, R. R. Moustafa, N. M. Elnahas, a M. Elganzoury, T. a Abdo, S. a Mohamed, and M. a Eletribi, "Repetitive transcranial magnetic stimulation at 1Hz and 5Hz produces sustained improvement in motor function and disability after ischaemic stroke.," *Eur. J. Neurol.*, vol. 17, no. 9, pp. 1203–9, Sep. 2010.

[27] D. a Nowak, C. Grefkes, M. Dafotakis, S. Eickhoff, J. Küst, H. Karbe, and G. R. Fink, "Effects of low-frequency repetitive transcranial magnetic stimulation of the contralesional primary motor cortex on movement kinematics and neural activity in subcortical stroke.," *Arch. Neurol.*, vol. 65, no. 6, pp. 741–7, Jun. 2008.

[28] E. M. Khedr, J. C. Rothwell, O. a. Shawky, M. a. Ahmed, and A. Hamdy, "Effect of daily repetitive transcranial magnetic stimulation on motor performance in Parkinson's disease," *Mov. Disord.*, vol. 21, no. 12, pp. 2201–2205, Dec. 2006.

[29] S. S. Geertsen, A. T. Zuur, and J. B. Nielsen, "Voluntary activation of ankle muscles is accompanied by subcortical facilitation of their antagonists.," *J. Physiol.*, vol. 588, no. Pt 13, pp. 2391–402, Jul. 2010.

[30] N. T. Petersen, J. E. Butler, V. Marchand-Pauvert, R. Fisher, a Ledebt, H. S. Pyndt, N. L. Hansen, and J. B. Nielsen, "Suppression of EMG activity by transcranial magnetic stimulation in human subjects during walking.," *J. Physiol.*, vol. 537, no. Pt 2, pp. 651–6, Dec. 2001.

[31] N. T. Petersen, H. S. Pyndt, and J. B. Nielsen, "Investigating human motor control by transcranial magnetic stimulation.," *Exp. Brain Res.*, vol. 152, no. 1, pp. 1–16, Sep. 2003.

[32] M. a Dimyan and L. G. Cohen, "Contribution of transcranial magnetic stimulation to the understanding of functional recovery mechanisms after stroke.," *Neurorehabil. Neural Repair*, vol. 24, no. 2, pp. 125–35, Feb. 2010.

[33] R. L. Waters, J. Morris, and J. Perry, "Translational motion of the head and trunk during normal walking," *J. Biomech.*, vol. 6, no. 2, pp. 167–170, 1973.

[34] J. J. Kavanagh, S. Morrison, and R. S. Barrett, "Coordination of head and trunk accelerations during walking.," *Eur. J. Appl. Physiol.*, vol. 94, no. 4, pp. 468–75, Jul. 2005.

[35] E. Hirasaki, S. T. Moore, T. Raphan, and B. Cohen, "Effects of walking velocity on vertical head and body movements during locomotion.," *Exp. Brain Res.*, vol. 127, no. 2, pp. 117–30, Jul. 1999.

[36] J. J. de Jong, "Design and Control of a Robot for TMS during Treadmill Walking," Univeristy of Twente, 2012.

[37] J. D. Bijsterbosch, A. T. Barker, K.-H. Lee, and P. W. R. Woodruff, "Where does transcranial magnetic stimulation (TMS) stimulate? Modelling of induced field maps for some common

cortical and cerebellar targets.," *Med. Biol. Eng. Comput.*, vol. 50, no. 7, pp. 671–81, Jul. 2012.

[38] I. Laakso, A. Hirata, and Y. Ugawa, "Effects of coil orientation on the electric field induced by TMS over the hand motor area.," *Phys. Med. Biol.*, vol. 59, no. 1, pp. 203–18, Jan. 2014.

[39] J. J. de Jong, A. H. A. Stienen, and H. van der Kooij, "Method for Numerical Evaluation of Performance Characteristics of 6-DOF Parallel Robots usable for TMS Positioning," *Phys. Ther.*

[40] M. Honegger, a. Codourey, and E. Burdet, "Adaptive control of the Hexaglide, a 6 dof parallel manipulator," *Proc. Int. Conf. Robot. Autom.*, vol. 1, no. April, pp. 543–548, 1997.

[41] F. Pierrot, P. Dauchez, and A. Fournier, "HEXA: A fast six-DOF fully-parallel robot," in *Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, 1991, pp. 1158–1163.

[42] D. W. Eggert, a. Lorusso, and R. B. Fisher, "Estimating 3-D rigid body transformations: a comparison of four major algorithms," *Mach. Vis. Appl.*, vol. 9, no. 5–6, pp. 272–290, Mar. 1997.

[43] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-D point sets," *Pattern Anal. …*, no. 5, pp. 698–700, 1987.

[44] A. Cappozzo, U. Della Croce, A. Leardini, and L. Chiari, "Human movement analysis using stereophotogrammetry. Part 1: theoretical background.," *Gait Posture*, vol. 21, no. 2, pp. 186–96, Feb. 2005.

[45] E. Todorov, "Probabilistic inference of multijoint movements, skeletal parameters and marker attachments from diverse motion capture data.," *IEEE Trans. Biomed. Eng.*, vol. 54, no. 11, pp. 1927–39, Nov. 2007.

[46] L. Righetti, J. Buchli, and A. J. Ijspeert, "Dynamic Hebbian learning in adaptive frequency oscillators," *Phys. D Nonlinear Phenom.*, vol. 216, no. 2, pp. 269–281, Apr. 2006.

[47] W. van Dijk, H. van der Kooij, B. Koopman, and E. H. F. van Asseldonk, "Improving the transparency of a rehabilitation robot by exploiting the cyclic behaviour of walking.," *IEEE Int. Conf. Rehabil. Robot.*, vol. 2013, p. 6650393, Jun. 2013.

[48] C. Harper and G. Virk, "Towards the Development of International Safety Standards for Human Robot Interaction," *Int. J. Soc. Robot.*, vol. 2, no. 3, pp. 229–234, Jun. 2010.

[49] W. Ng, "On safety enhancements for medical robots," *Reliab. Eng. Syst. Saf.*, vol. 54, no. 1, pp. 35–45, Oct. 1996.

[50] P. Poignet, E. Dombre, O. Merigeaux, F. Pierrot, and G. Duchemin, "Design and control issues for intrinsically safe medical robots," *Ind. Robot An Int. J.*, vol. 30, no. 1, pp. 83–88, 2003.

[51] X. Huang, Q. Liao, and S. Wei, "Closed-form forward kinematics for a symmetrical 6-6 Stewart platform using algebraic elimination," *Mech. Mach. Theory*, vol. 45, no. 2, pp. 327–334, Feb. 2010.

[52] M. Husty, "An algorithm for solving the direct kinematics of general Stewart-Gough platforms," *Mech. Mach. Theory*, vol. 31, no. 4, 1996.

[53] A. Codourey, "Dynamic modelling and mass matrix evaluation of the DELTA parallel robot for axes decoupling control," *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. IROS '96*, vol. 3, pp. 1211–1218, 1996.

[54] K. G. Holt, R. Ratcliffe, and S. F. Jeng, "Head stability in walking in children with cerebral palsy and in children and adults without neurological impairment.," *Phys. Ther.*, vol. 79, no. 12, pp. 1153–62, Dec. 1999.

[55] T. Pozzo, A. Berthoz, L. Lefort, and E. Vitte, "Head stabilization during various locomotor tasks in humans," *Exp. brain Res.*, vol. 85, no. 1, pp. 208–217, 1991.

[56] L. Laudani, a Casabona, V. Perciavalle, and a Macaluso, "Control of head stability during gait initiation in young and older women.," *J. Electromyogr. Kinesiol.*, vol. 16, no. 6, pp. 603–10, Dec. 2006.

[57] A. P. Mulavara, M. C. Verstraete, and J. J. Bloomberg, "Modulation of head movement control in humans during treadmill walking.," *Gait Posture*, vol. 16, no. 3, pp. 271–82, Dec. 2002.

[58] M. D. Latt, H. B. Menz, V. S. Fung, and S. R. Lord, "Walking speed, cadence and step length are selected to optimize the stability of head and pelvis accelerations.," *Exp. Brain Res.*, vol. 184, no. 2, pp. 201–9, Jan. 2008.

[59] H. B. Menz, S. R. Lord, and R. C. Fitzpatrick, "Acceleration patterns of the head and pelvis when walking on level and irregular surfaces.," *Gait Posture*, vol. 18, no. 1, pp. 35–46, Aug. 2003.

Nomenclature

Abbreviations

Abbreviation	Meaning
DAQ	Data acquisition
DKM	Direct kinematic model (Same as FKM)
DOF	Degree of freedom
EKF	Extended Kalman filter method
FKM	Forward kinematic model
FMEA	Failure mode effect analysis
IKM	Inverse kinematic model
IO	Input-output
MEP	Motor evoked potential
OP	Orthogonal Procrustes method
RI	Redundant input singularity
RO	Redundant output singularity
ROM	Range of motion
Ro-TMS	Robotized transcranial magnetic stimulation
RPM	Redundant passive motion singularity
rTMS	Repetitive transcranial magnetic stimulation
SVD	Singular value decomposition
TMS	Transcranial magnetic stimulation
UI	User Interface

List of notations

Notation	Meaning
$\boldsymbol{a}$	Any vector, can also be list with vectors as rows
A	Mappings and (transfer) functions are denoted with a regular capital
$\boldsymbol{A}$	Matrices are denoted with a bold capital
${}_x a$	The x-dimension (or column) of vector $\boldsymbol{a}$. Also works with matrices
${}^i\boldsymbol{a}$	The i-th point (or row) of $\boldsymbol{a}$ also works with matrices
$\boldsymbol{a}_i$	Sub-script has a context dependent meaning.
$\boldsymbol{a}_i^j$	Vector $\boldsymbol{a}$ is an expression from frame i to frame j, also works with matrices
$[\boldsymbol{a}\times]$	Skew symmetric matrix or semi-skew symmetric matrix form of the vector
$[\boldsymbol{a}\otimes]$, $[\boldsymbol{a}\otimes]^T$	Quaternion product in matrix form. The left and right multiplication respectively.
$\breve{\boldsymbol{a}}$	Appended vector with a 1 or a 0
$\overline{\boldsymbol{a}}$	Averaged vector over the rows
$\widehat{\boldsymbol{a}}$	Estimation of $\boldsymbol{a}$
$\dot{\boldsymbol{a}}, \ddot{\boldsymbol{a}}, \dddot{\boldsymbol{a}}$	First, second, third time derivative of $\boldsymbol{a}$

List of symbols

Symbol	Meaning
$\times$	Cross product
$\otimes$	Quaternion product
ψ_i	Reference frame i
$\boldsymbol{p}^i$	Point (usually a marker) expressed in frame i
$\boldsymbol{H}_i^j$	Transformation matrix from frame i to frame j
$\boldsymbol{R}_i^j$	Rotation matrix from frame i to frame j
$\boldsymbol{o}_i^j$	Translation of the origin of frame i to frame j

Symbol	**Meaning**
$\boldsymbol{v}_i^{j,k}$	Twist of frame i with respect to j expressed in frame k
$\boldsymbol{\omega}$	Angular velocity
$\boldsymbol{u}$	Translational velocity
$\boldsymbol{I}_n$	Identity matrix with n the size of the square matrix.
t	Time
Chapter 2. Robotized TMS for application during treadmill walking	
φ_{Vz}	Visualeyez reference frame
φ_H	Stimulation site.
φ_{Hm}	Head-marker reference frame
φ_B	Base reference frame
φ_{Bm}	Base-marker reference frame
φ_P	End-effector reference frame
M_p	Marker tracking
M_h	Tracking transformations
F	Tracking filter
${}^{rl}\boldsymbol{p}$	Robot landmarks
t_s	Setting time (initial time)
φ^s	Setting frame
$\boldsymbol{X}, \boldsymbol{Y}$	Centroid of marker frame at current time and set time
$\boldsymbol{S}$	Covariance matrix,
$\boldsymbol{U}, \boldsymbol{\Sigma}, \boldsymbol{V}$	Singular value decomposition matrices
M_g	Gravity compensation
M_s	Spring model
C_f	Force controller
M_f	Force compensation mapping
d_s	Spring deflection
f_a	Actual spring force
f_r	Spring reference force
f_d	Desired spring force
$\Delta\boldsymbol{H}_p^B$	Spring controller pose offset
m	Mass of the coil
$k_{f,p}, k_{f,i}$	Proportional and integral controller gains for the force controller
k_s	Spring constant
$\boldsymbol{J}_{\boldsymbol{g}}$	Jacobian matrix of kinematic relation in global coordinates
C_θ	Joint angle controller
k_v	Friction compensation
$\boldsymbol{u}, \boldsymbol{u}_v, \boldsymbol{u}_c$	Motor voltages, controller and velocity feedforward.
$\boldsymbol{\theta}_a$	Actual joint angles
$\boldsymbol{\theta}_d$	Desired joint angles
Chapter 3 Comparison between two realtime tracking methods for robotized TMS	
$\boldsymbol{q}$	Quaternions
${}_{l}\boldsymbol{q}$	Quaternions vector part
${}_{s}q$	Quaternions scalar part
θ	2-Norm of the angular velocity
$\boldsymbol{x}$	Kalman state
$\boldsymbol{z}$	Measurement
$\boldsymbol{y}$	Output
F, B, Y	State evolution function, measurement function, output function
$\boldsymbol{w}, \boldsymbol{m}$	Process noise and measurement noise
k	Discrete time
$\boldsymbol{F}, \boldsymbol{B}$	State Jacobian and measurement Jacobian
$\hat{\boldsymbol{x}}_{pre}, \hat{\boldsymbol{x}}_{mea}$	Estimate state after prediction and measurement update
$\boldsymbol{C}_{pre}, \boldsymbol{C}_{mea}, \boldsymbol{C}_w,$	Prediction, measurement, process and measurement covariance matrices
$\boldsymbol{S}$	Innovation matrix
$\boldsymbol{K}$	Kalman gain matrix
$\boldsymbol{\varepsilon}$	Measurement residual
$\boldsymbol{F}_{qq}, \boldsymbol{F}_{q\omega}, \boldsymbol{F}_{vv}$	Quaternion to quaternion and angular velocity to quaternion and twist to twist Jacobians
$\boldsymbol{B}_{pq}$	Jacobian of quaternion to point measurement.
$\boldsymbol{n}_x, \boldsymbol{n}_y, \boldsymbol{n}_z$	Selection vectors in the relative directions.
d	Mahalanobis distance

Symbol	Meaning
Chapter 5 Kinematic analysis of the Hexa manipulator	
$\boldsymbol{\theta}$	Joint angles
$\boldsymbol{x}$	Robot pose
$\boldsymbol{o}$	End-effector position
$\boldsymbol{\phi}$	End-effector orientation in Euler angles
$\boldsymbol{H}_p^B$	Transformation matrix from platform to base is pose dependent
$\boldsymbol{R}_{zyx}$	Rotation matrix derived from three Euler angles in the z-y-x sequence
$\boldsymbol{R}_x, \boldsymbol{R}_y, \boldsymbol{R}_z$	Different rotation matrices around dimension specified
s, c	Short hand for sine and cosine function
$\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i$	Position of the i-th shoulder, elbow and wrist in base frame.
l_u	Length of the upper arm
l_f	Length of the forearm
l_f^*, l_c^*	Lengths projected onto the x^*z plane
$\boldsymbol{l}_B$	Dimension of the base in base frame
$\boldsymbol{l}_p$	Dimension of the platform in platform frame
γ	Angle of rotation to the x^*z plane
α, β	Auxiliary angles for inverse kinematic method
$\boldsymbol{\theta_e}$	Residual of joint angle estimation
$\boldsymbol{J_e}$	Jacobian to the Euler rate
$\boldsymbol{J}_g$	Jacobian to pose derivative.
$\dot{\boldsymbol{x}}_g$	Pose derivative with linear velocity and angular velocity part.
$\boldsymbol{z}$	Vector pointing from the wrist to the elbow
$\boldsymbol{M}_i, \boldsymbol{N}_i$	Auxiliary matrices for defining Jacobian, with end-effector and joint angle part respectively.
$\boldsymbol{Q}, \boldsymbol{E}$	Conversion matrices from global velocities to Euler rate.
κ	Condition number of a matrix.
λ	Eigenvalues of a matrix
Chapter 6 Controller design	
x_m	Position of the coil
x_h	Position of the head
x_r	Position of the robot
x_o	Position of the base
d_s	Spring displacement
d_{ref}	Desired spring deflexion
d_c	Skin compression
F_s	Spring force
F_c	Contact force
F_{ref}	Force reference
s	Laplace operator
m	Mass of the coil
k_s, c_s	Spring stiffness and damping
k_c, c_c	Skin stiffness and damping
C	Contact force controller
H_p	Static spring model
H_h	Actual head contact dynamics
H_s	Actual head contact dynamics
F	Prefilter and tracking dynamics
P_r	Dynamics of the robot
Q_1, Q_2	Selection matrices
$\Delta x_h, \Delta x_r, \Delta d_s$	Disturbances

Summary

Transcranial Magnetic Stimulation (TMS) uses a strong electromagnetic field induced by a magnetic coil to generate small currents inside the brain. The resulting brain activity can be used to identify motor responses, relieves several neurological diseases and may assist in motor rehabilitation. As this stimulation needs to be both precise and repeatable, computer aided neuronavigation and robotized TMS have been developed. Most of the robots currently used for TMS are powerful, industrial robots. Because of safety issues, the speed of these robots is limited excluding them to be used during motor training activities. The goal of the TMS robot, presented in this report, is to overcome these issues and enable the use of robotized TMS during treadmill walking.

For a medical robot, safety considerations are paramount. A TMS robot is designed to operate in contact with the head. Therefore, it is essential that the robot cannot fail in way that will hurt the subject. Safety measures in this design include a soft interaction contact switch, external safety circuit to monitor the operation of the control system, a mechanical design with minimal mass and redundant sensors.

The robot is designed to follow the head movement of a subject walking on a treadmill. The range of motion required for the robot is a cube of 0.16 m with rotation of +/- 15 degrees in all direction. This range of motion also allows the robot to stimulate over a grind of 7.5x7.5 cm on the head to find the optimal stimulation site. The velocities and accelerations required for the robot are 1 m/s and 150 deg/s and 1 m/s^2 and 250 deg/s^2. The accuracy of stimulation is desired to be 1 mm.

The robot presses the coil against the skull through a safety spring system. Full power to the robot is only allowed when the contact forces are within safety boundaries. This ensures a timely shut down in case of contact loss and prevents high energetic impacts. The coil and safety spring system follow the movement of the head by use of a Hexa manipulator that keeps the focal point of the coil on the same point within the head. This Hexa manipulator is a six-armed parallel mechanism with base mounted motors. This reduces the moving mass to a minimum. The manipulator has a workspace limited to the range of motion required for tracking the head during treadmill walking. To compensate for different subject lengths and stimulation sites the Hexa robot can be moved around the head by the supporting frame.

A 3D optical tracker system measures the position of optical markers on the head of the subject and on the base of the robot. “Point cloud” estimation allows the calculation of the transformations required to steer the robot over the stimulation site. The measured spring force deflection is used to adjust the robot pose such that a constant contact force is maintained.

The evaluation of the robot shows that the Hexa manipulator can be steered with sub millimeter accuracy over a trajectory with dimensions of the required workspace and dynamical performance exceeding the requirements with maximal velocities 1.5 m/s and acceleration of more than 40 m/s^2. A force controller proved to increase the smoothness of the contact force up to a bandwidth of 7 Hz. The tracking of the head was successfully performed. The noise of the optical tracker mechanism however required filtering and therefore limits the bandwidth of the system to 1 Hz. This is not sufficient for tracking the head during treadmill walking.

The bandwidth of tracking can be improved by the use of more elaborated filtering methods such as Kalman tracking, use of accelerometers on the subject and by improvement of the communication with the tracker.

Samenvatting

Transcraniële Magnetische Stimulatie (TMS) wekt, door middel van een sterk fluctuerend magnetisch veld, kleine elektrische stromen op in de hersenen. Deze stromen zorgen ervoor dat dat een hersengebied lokaal wordt gestimuleerd of geïnhibeerd en kan zodoende leiden tot een veranderde neurale activiteit. Vanwege de vereiste nauwkeurigheid en herhaalbaarheid van stimulatie zijn neuronavigatie en robotische TMS ontwikkeld. De gebruikte robots zijn krachtige industriële robots, waarvan de snelheid is gelimiteerd om de veiligheid te kunnen garanderen. Om toch snelle bewegingen van het hoofd te kunnen volgen tijdens revalidatie training, werd er een nieuwe robot ontwikkeld. Deze TMS robot wordt gepresenteerd in dit verslag

Deze robot is ontworpen om in contact met het hoofd te functioneren. Het is daarom essentieel dat de robot in geen geval een gevaar vormt voor de patiënt. Als veiligheidsmaatregel is de robot uitgerust met een veerkracht-interactie-schakelaar. Verder wordt de aansturing gecontroleerd door een extern veilheidscircuit. De sensoren zijn redundant uitgerust zodat sensorfouten opgespoord kunnen worden. De robot is ontworpen met zo min mogelijk bewegende massa waardoor de kinetische energie laag blijft en de gevolgen van impact minimaal.

De robot is ontworpen om het hoofd te volgen tijdens het lopen op een loopband. Het bewegingsbereik is een kubus van 0.16 m en 15 graden in alle richtingen. Dit bereik geeft de robot de mogelijk om een raster te stimuleren zodat de optimale stimulatie plek bepaald kan worden. De vereiste snelheden en versnelling zijn 1 m/s en 150 graden/s en 1 m/s^2 en 250 graden/s^2. De nauwkeurigheid van stimulatie moet hoger zijn dan 1 mm.

De robot drukt de stimulator tegen het hoofd met een veerveiligheidssysteem. Dit zorgt voor een elastisch en zacht contact. Het volledig vermogen wordt alleen ingeschakeld als de contactkracht binnen veilige grenzen ligt. Door het bijtijds uitschakelen van de robot wordt voorkomen dat het systeem in vol vermogen een botsing kan maken met de schedel. De beweging van het hoofd wordt gevolgd door een Hexa-manipulator zodat de spoel op dezelfde positie van het hoofd gehouden wordt. Deze Hexa robot is een zesarmige parallelle mechanisme waarvan de motoren op de basis zijn geplaatst. Hierdoor wordt de bewegende massa beperkt tot een minimum. Het werkgebied van de robot is begrensd tot het werkgebied vereist voor het volgen van het hoofd. Om te compenseren voor verschillende lengtes van personen en stimulatiegebieden kan de Hexa-manipulator doormiddel van een boogvormig frame verplaatst worden.

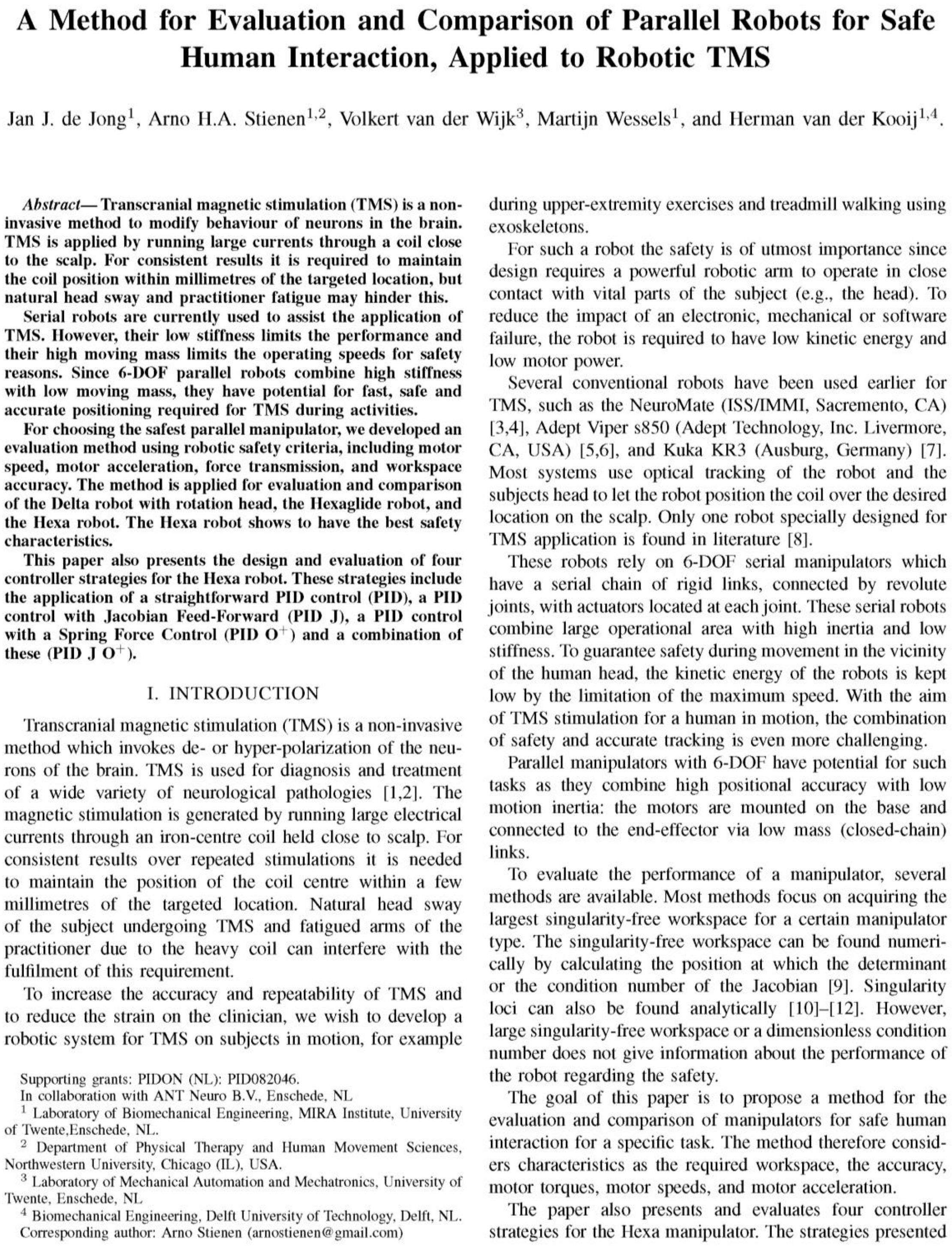
Een 3D optisch camera systeem meet de positie van de optische markers op het hoofd en op de basis van de robot. De “Point Cloud” methode wordt gebruikt om de transformaties te berekenen tussen gemeten posities van de markers en de gewenste positie van de robot. De gemeten veercontactkracht wordt via een regellus zo constant mogelijk te houden.

De evaluatie van de robot laat zien dat de Hexa met een hoge nauwkeurigheid over een pad gestuurd kan worden. Het kan het werkgebied bereiken dat nodig is voor het volgen van een hoofd tijdens lopen op een loopband. De maximale snelheden en versnellingen liggen boven de 1.5 m/s en 40 m/s^2. Contactkrachtsturing leidt tot een vloeiende contractkracht met een maximale bandbreedte van 7 Hz. Het volgen van het hoofd is succesvol uitgevoerd. Helaas leidt de ruis in het optisch meetsysteem tot ongewenste oscillaties en onnauwkeurigheden. Daarom is een filter nodig op de gemeten hoofdpositie, daardoor wordt de bandbreedte van het gehele systeem gereduceerd tot 1 Hz. Dit is niet voldoende voor het volgen van het hoofd tijdens loopbeweging op een loopband.

Deze bandbreedte kan verhoogd worden door betere filtermethoden, zoals Kalman filters, en door het gebruik te maken van accelerometer of gyroscopen en door verbeterde communicatie met het optische meetsysteem.

Appendix

Appendix A. A METHOD FOR EVALUATION OF PARALLEL ROBOTS FOR SAFE HUMAN INTERACTION, APPLIED TO ROBOTIC TMS



A Method for Evaluation and Comparison of Parallel Robots for Safe Human Interaction, Applied to Robotic TMS

Jan J. de Jong[1], Arno H.A. Stienen[1,2], Volkert van der Wijk[3], Martijn Wessels[1], and Herman van der Kooij[1,4].

***Abstract*— Transcranial magnetic stimulation (TMS) is a non-invasive method to modify behaviour of neurons in the brain. TMS is applied by running large currents through a coil close to the scalp. For consistent results it is required to maintain the coil position within millimetres of the targeted location, but natural head sway and practitioner fatigue may hinder this.**

Serial robots are currently used to assist the application of TMS. However, their low stiffness limits the performance and their high moving mass limits the operating speeds for safety reasons. Since 6-DOF parallel robots combine high stiffness with low moving mass, they have potential for fast, safe and accurate positioning required for TMS during activities.

For choosing the safest parallel manipulator, we developed an evaluation method using robotic safety criteria, including motor speed, motor acceleration, force transmission, and workspace accuracy. The method is applied for evaluation and comparison of the Delta robot with rotation head, the Hexaglide robot, and the Hexa robot. The Hexa robot shows to have the best safety characteristics.

This paper also presents the design and evaluation of four controller strategies for the Hexa robot. These strategies include the application of a straightforward PID control (PID), a PID control with Jacobian Feed-Forward (PID J), a PID control with a Spring Force Control (PID O^{+}) and a combination of these (PID J O^{+}).

I. INTRODUCTION

Transcranial magnetic stimulation (TMS) is a non-invasive method which invokes de- or hyper-polarization of the neurons of the brain. TMS is used for diagnosis and treatment of a wide variety of neurological pathologies [1,2]. The magnetic stimulation is generated by running large electrical currents through an iron-centre coil held close to scalp. For consistent results over repeated stimulations it is needed to maintain the position of the coil centre within a few millimetres of the targeted location. Natural head sway of the subject undergoing TMS and fatigued arms of the practitioner due to the heavy coil can interfere with the fulfilment of this requirement.

To increase the accuracy and repeatability of TMS and to reduce the strain on the clinician, we wish to develop a robotic system for TMS on subjects in motion, for example during upper-extremity exercises and treadmill walking using exoskeletons.

For such a robot the safety is of utmost importance since design requires a powerful robotic arm to operate in close contact with vital parts of the subject (e.g., the head). To reduce the impact of an electronic, mechanical or software failure, the robot is required to have low kinetic energy and low motor power.

Several conventional robots have been used earlier for TMS, such as the NeuroMate (ISS/IMMI, Sacremento, CA) [3,4], Adept Viper s850 (Adept Technology, Inc. Livermore, CA, USA) [5,6], and Kuka KR3 (Ausburg, Germany) [7]. Most systems use optical tracking of the robot and the subjects head to let the robot position the coil over the desired location on the scalp. Only one robot specially designed for TMS application is found in literature [8].

These robots rely on 6-DOF serial manipulators which have a serial chain of rigid links, connected by revolute joints, with actuators located at each joint. These serial robots combine large operational area with high inertia and low stiffness. To guarantee safety during movement in the vicinity of the human head, the kinetic energy of the robots is kept low by the limitation of the maximum speed. With the aim of TMS stimulation for a human in motion, the combination of safety and accurate tracking is even more challenging.

Parallel manipulators with 6-DOF have potential for such tasks as they combine high positional accuracy with low motion inertia: the motors are mounted on the base and connected to the end-effector via low mass (closed-chain) links.

To evaluate the performance of a manipulator, several methods are available. Most methods focus on acquiring the largest singularity-free workspace for a certain manipulator type. The singularity-free workspace can be found numerically by calculating the position at which the determinant or the condition number of the Jacobian [9]. Singularity loci can also be found analytically [10]–[12]. However, large singularity-free workspace or a dimensionless condition number does not give information about the performance of the robot regarding the safety.

The goal of this paper is to propose a method for the evaluation and comparison of manipulators for safe human interaction for a specific task. The method therefore considers characteristics as the required workspace, the accuracy, motor torques, motor speeds, and motor acceleration.

The paper also presents and evaluates four controller strategies for the Hexa manipulator. The strategies presented

Supporting grants: PIDON (NL): PID082046.
In collaboration with ANT Neuro B.V., Enschede, NL

[1] Laboratory of Biomechanical Engineering, MIRA Institute, University of Twente,Enschede, NL.
[2] Department of Physical Therapy and Human Movement Sciences, Northwestern University, Chicago (IL), USA.
[3] Laboratory of Mechanical Automation and Mechatronics, University of Twente, Enschede, NL
[4] Biomechanical Engineering, Delft University of Technology, Delft, NL.
Corresponding author: Arno Stienen (arnostienen@gmail.com)

are: PID control in conjunction with Jacobian based feedforward compensation, contact force control, and a combination of these methods.

First the method is presented, then the selected manipulators for evaluation and comparison are described. The manipulators are evaluated and compared and the results are discussed. Subsequently the controller designs are presented and discussed. s

II. METHOD

In order to make a quantitative comparison among various robot types or between geometries, a numerical evaluation of the robotic safety is proposed.

The safety of the robot in each pose can by quantified using five Robotic Safety Indexes (RSI): the maximum motor speed, maximum motor acceleration, maximum motor torque, allowed motor error and the condition number of the Jacobian.

Together these five measures give a indication of the kinematic energy on the end-effector, the motor power and the accuracy within the workspace.

The required motor power can be derived from the maximal motor speed and the maximal motor torque. Motor torque also gives an estimation of the reflected mass from the end-effector, making it a measure for the kinetic energy on the end-effector. The allowed motor error gives a measure for the accuracy of the system. The closeness to singularity can be estimated from the condition number and the maximal motor acceleration to show whether the workspace is large enough.

To calculate the RSIs, first the required workspace and end-effector speeds, accelerations, forces and toques are defined. From these end-effector values the resulting motor values (speed, acceleration, torque and error) can be calculated using the Jacobian (Equation 1 - Equation 6). The RSIs are given by the maximum values of these relations.

The Jacobian ($\mathbb{J}$) and its derivative[1] ($\dot{\mathbb{J}}$) are found by partial first and second-order differentiation of the end-effector pose to the motor coordinates.

$$\mathbb{J}(\mathbf{X}) = \begin{bmatrix} \frac{\delta\Theta_1}{\delta X_1} \cdots \frac{\delta\Theta_1}{\delta X_k} \\ \vdots \quad \ddots \quad \vdots \\ \frac{\delta\Theta_n}{\delta X_1} \cdots \frac{\delta\Theta_n}{\delta X_k} \end{bmatrix} \tag{1}$$

Robotic Safety Indexes are given by the evaluation of the five relations:

1) The relation between the motor speeds ($\dot{\boldsymbol{\Theta}}$) and the generalized end-effector velocity ($\dot{\mathbf{X}}$).

$$\dot{\boldsymbol{\Theta}} = \mathbb{J}\dot{\mathbf{X}} \tag{2}$$

2) The relation between the motor acceleration ($\ddot{\boldsymbol{\Theta}}$) and the end-effector acceleration ($\ddot{\mathbf{X}}$).

$$\ddot{\boldsymbol{\Theta}} = \mathbb{J}\ddot{\mathbf{X}} + \dot{\mathbf{X}}^T\dot{\mathbb{J}}\dot{\mathbf{X}}. \tag{3}$$

3) The relation between the generalized motor force ($\boldsymbol{\tau}$) and end-effector force ($\boldsymbol{\mathcal{F}}$).

$$\boldsymbol{\tau} = \mathbb{J}^{-T}\boldsymbol{\mathcal{F}} \tag{4}$$

4) The relation between the allowed motor error ($\boldsymbol{\varepsilon}_m$) and the end-effector pose error ($\boldsymbol{\varepsilon}_{ee}$) gives the error transfer. For this RSI the minimal allowable motor error is taken.

$$\boldsymbol{\varepsilon}_m = \mathbb{J}\boldsymbol{\varepsilon}_{ee} \tag{5}$$

5) The closeness to singularity of the matrix can be calculated by the condition number $\kappa(\mathbb{J})$ of the Jacobian.

$$\kappa(\mathbb{J}) = \|\mathbb{J}\| \cdot \|\mathbb{J}^{-1}\| \tag{6}$$

III. MANIPULATOR DESCRIPTIONS

The Robotic Safety Indexes method is applied to the evaluation and comparison of three parallel robot concepts of Figure 1: the Delta robot with rotation head, the Hexaglide, and the Hexa robot.

1) Delta robot with rotation head: The Delta robot with rotational head (3-$\underline{\text{R}}$RP$_\text{a}$R-$\underline{\text{RRR}}$)[2] consists of a parallel 3-DOF translational structure (the Delta robot) combined with a 3-DOF rotational head. The actuators of the rotation head are located on the moving platform of the Delta robot. This 6-DOF Delta robot has recently been developed for pick-and-place applications [13] and is used as a surgical tool [14].

2) Hexaglide: The 6-DOF parallel manipulator Hexaglide (6-$\underline{\text{P}}$US[3]) uses linear motors to actuate six legs. The position and orientation of the glider motor can be arranged to create specific workspaces. The Hexaglide is a simple 6-DOF parallel structure as only two universal joints per kinematic chains are used. The manipulator used is for machine milling applications[4].

3) Hexa: The Hexa (6-$\underline{\text{R}}$US) manipulator [15] consists of six articulated kinematic chains. The upper arms are actuated by rotational motors, which determine the position and orientation of manipulator platform. Two commercial applications of this robot type are known, both use the 6-$\underline{\text{R}}$US system for actuation of a flight simulator[5].

The 6-DOF parallel manipulator should continuously and accurately positions the coil to follow the motion of the head. To compensate for different subjects and stimulation sites a passive positioning system places this robotic manipulator at a predetermined height and angle. The required range of motion, speed and acceleration for the TMS robot are

[1] The $\dot{\mathbb{J}}$ is a n x m x n matrix, with n is the number of DOFs and m the number of actuators. The calculation each element of $\ddot{\Theta}_\text{i}$ is done by matrix multiplication of each page ($\dot{\mathbb{J}}_\text{i}$) with $\dot{\mathbf{x}}$ and $\dot{\mathbf{x}}^T$

[2] The technical name for a robot is given by the subsequent joint types. (R:Rotational Joints, P: Prismatic Joint,S:Spherical Joint,U: Universal Joint, P_a Parallelogram structure. Actuated joints are underlined)

[3] In literature other nomenclature for the Hexaglide are found: 6-$\underline{\text{P}}$SU and 6-$\underline{\text{P}}$SS, the functional behaviour is similar. This also applies for the Hexa.

[4] http://www.iwf.mavt.ethz.ch/

[5] http://www.servos.com, http://www.fidelityflight.com/

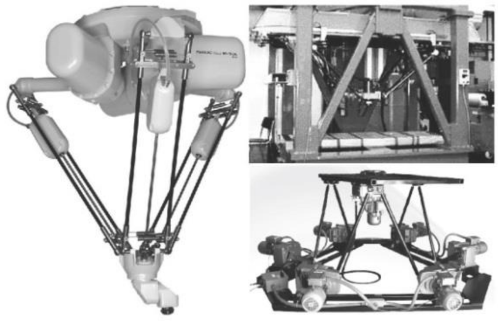


Fig. 1. Three 6-DOF manipulator concepts. Left: Delta robot with rotation head (FANUC M-3iA/6A). Right top: Hexaglide (IWF, ETH, Zurich). Right bottom: Hexa motion platform (Servos & Simulations Inc. 710-7).

TABLE I

SPECIFICATION OF THE ACTIVE WORKSPACE OF A TMS ROBOT.

DOF	Workspace[6]	$\dot{\mathbf{X}}$	$\ddot{\mathbf{X}}$	$\mathcal{F}$	ε_{ee}.
x	0.10	1.50	3.00	10.5	0.5
y	0.10	1.50	3.00	10.5	0.5
z	0.10	1.50	10.0	68.7	0.5
Φ_x	0.51	4.73	59.7	5.37	13.0
Φ_y	0.51	4.73	59.7	5.37	13.0
Φ_z	0.22	4.73	59.7	1.19	-

[6] The workspace is given by the maximal amplitude of the motion in each direction. X, y, and z denotes the frontal, lateral and vertical direction. Φ_x,Φ_y, and Φ_z are the rotations around the respective axes. The required end-effector velocity is denoted with a $\dot{\mathbf{X}}$, the required acceleration by $\ddot{\mathbf{X}}$, the end-effector wrench is given by $\mathcal{F}$ and end-effector error is given by ε_{ee}. The translational workspace, speed, acceleration, wrench and error are given in, respectively, m, m/s, m/s^2, N and mm. The rotational values are given in rad, rad/s, rad/s^2, Nm, and $mrad$, respectively.

based on literature on head movement [16]–[19] and on experiments with paraplegic subjects wearing the robotic exoskeleton LOPES [20]. The required end-effector wrench is calculated from the maximal acceleration of a typical coil of 3.5 kg. The resulting end-effector values can be found in Table I.

For a fair comparison of the performances, the dimensions of the three robots should be comparable. We chose the dimensions such that distance between base and end-effector is 0.6 m in outstretched pose. Since the robot arms can be manufactured from lightweight carbon tubes the weight of the moving parts can be neglected with respect to the weight of the coil. For the Delta robot however, an additional weight of 4 kg for the rotational head has to be taken into account, since the motors of the rotation head are moved during translation. Both the Delta robot and the Hexa use reduction gears with a gear ratio of 1:35.

IV. EVALUATION AND COMPARISON

For each concept the Jacobian are formulated. The corresponding RSI values are calculated by substituting the values of Table I into Equation 2-Equation 6. The maximal values of each RSI at full rotation inside the workspace are shown in Table II.

TABLE II

THE MAXIMUM ROBOTIC SAFETY INDEXES INSIDE THE OPTIMAL WORKSPACE OF THE THREE CONCEPTS AT MAXIMAL ROTATION

RSI[7]	Delta	H-G	Hexa
1. $\dot{\Theta}_{max}$	$1.07\ 10^3$	4.60	$2.22\ 10^3$
2. $\ddot{\Theta}_{max}$	$8.40\ 10^3$	35.3	$1.55\ 10^4$
3. τ_{max}	1.62	$9.86\ 10^5$	1.43
4. $\varepsilon_{m,min}$	35.7	$3.88\ 10^{-5}$	7.18
5. $\kappa(\mathbb{J})$	16.3	$1.07\ 10^5$	127

[7] The motor speed ($\dot{\Theta}_{max}$), acceleration ($\ddot{\Theta}_{max}$), torque(τ_{max}) and allowed error ($\varepsilon_{m,min}$) of the Delta Robot (Delta) and Hexa are respectively given in rad/s, rad/s^2, Nm and $mrad$ for the Hexaglide (H-G) these are given in m/s, m/s^2, N mm. The former manipulator types rely on rotational actuators while the latter use linear motors.

Because of limited space, only the motor force plots are shown in Figure 2. The X- and Y-axis of the plot indicate the position within the slice trough the workspace. The colours in the generalised force plots are normalized on the maximal motor force of the three actuators with similar motor power. The space inside the white box represents the optimal workspace in which the required motor power is minimal.

Table II shows that the *Delta robot* requires lower motor speed and less acceleration but higher torque to obtain the same end-effector movement than the Hexa. The allowed error motor is larger than that of the other concepts. Figure 2 shows that at the top and near the edges of the range of motion high motor torques are required due to the singularities. The extra mass of the rotation head gives the Delta robot higher kinetic energy, increasing the required torque and reducing the intrinsic safety.

High motor torque values appear through the middle of the workspace of the *Hexaglide*, revealing a singularity. This singularity occurs during rotation when multiple kinematic chains become co-planar with the end-effector. The singularity decreases the robotics safety strongly, making this robot type unsuitable for the required rotational task.

The XZ-plane of the generalised force plot for the *Hexa* shows that total range of motion is smaller than that of the Delta Robot and Hexaglide, but large enough for required workspace. The motor torque is more favourable than the other concepts. The force map reveals a singularity outside the active workspace of the Hexa, which does not influence the RSI. Therefore this concept is chosen for optimization and further design, as the kinetic performance is sufficient and the robotic safety is the highest, compared to the Delta robot and the Hexaglide.

Low motor speed, acceleration, torque and error transfer are desirable for a robotic manipulator. However, Equation 2-Equation 6 show that there is a trade-off among the RSIs. For example, a larger motor arm generally leads to a decrease in the motor speed but also to an increased motor torque and error.

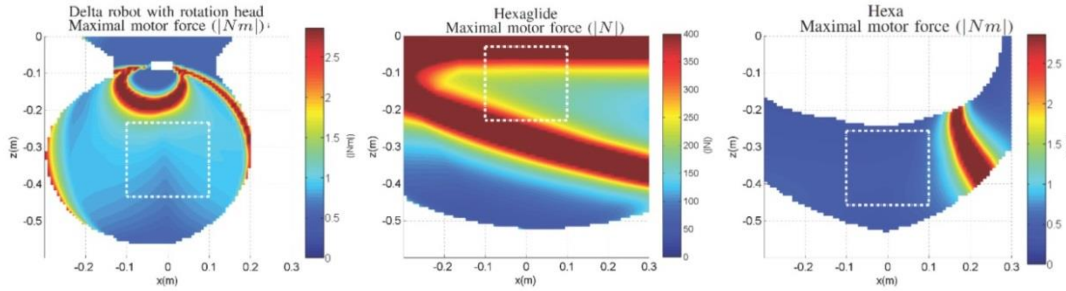


Fig. 2. The XZ plane of the motor generalised force plots show the maximal motor force over the workspace during maximal desired end-effector rotation. The space inside the white box represent the optimal workspace. Middle: Hexaglide. Right: Hexa. For the Delta robot and the Hexa the motor force is given in Nm. For the Hexaglide the force is given in N.

V. Controller Design

Safe tracking of the head can only be achieved by controlling software that is able to detect and handle errors and malfunctions. The controller should be robust enough to reject disturbances and model variations to guarantee stable control over the required workspace.

The input for the controller is derived from head position which is measured with the commercially available Visor TMS neuronavigation system (ANT Neuro B.V., Enschede, The Netherlands).

A spring is placed between the coil and the manipulator to ensure a constant contact between the coil and the head, to prevent collision and to ensure soft human-robot interaction. The contact force between the coil and the head is monitored by measuring the spring deflection.

The dynamic properties of the Hexa mechanism are investigated with the aid of a 3d rigid body model in 20-sim (Edition 4.1, Controllab Products B.V., Enschede, The Netherlands).

A. Four Controllers

The disturbances induced by model uncertainties and force perturbations (e.g. variations in skull contact force), are expected to be amplified by the non-linear kinematic and dynamic behaviour of the parallel manipulator. Conventional controller strategies, such as feedback linearisation and compliance control, use joint-spaced dynamic models to calculate required motor torques. For some parallel robots, these joint-spaced models are not available and for others they require high computation power [9, p. 286], [21], therefore other strategies are to be used. Four controllers are introduced to compensate for these disturbances and allow adequate response times.

1) PID Control: Under the assumption that the non-linear behaviour is reduced by the reduction gears, a joint spaced PID controller is proposed (see Figure 3). The root locus and Nichols charts of the linearised response of the dynamic model are used to determine the PID values. The controller parameters are aimed at high stability margins. The stability can not be proven rigorously for all the possible poses [22] and is therefore demonstrated experimentally. The resulting PID controller values are found to be $\mathbb{K}_P = 8.24$, $\mathbb{K}_I = 0.10$, $\mathbb{K}_D = 1$ and a tameness factor $\chi = 0.007$.

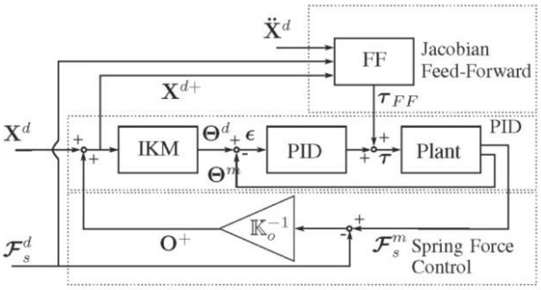


Fig. 3. Different optional elements of the PID controller (PID) with Jacobian Feed-Forward (FF) and Spring Force Control. Here the IKM stands for the Inverse Kinematic Model, Plant stands for the dynamic model of the Hexa including gears and motors. $\mathbf{X}^d$ and $\ddot{\mathbf{X}}^d$ are the desired pose and acceleration. $\boldsymbol{\Theta}^d$ and $\boldsymbol{\Theta}^m$ stand for the desired and measured motor angle. $\boldsymbol{\tau}$ and $\boldsymbol{\tau}_{FF}$ are the motor torque and feed-forward torque. $\boldsymbol{\mathcal{F}}_s^d$ and $\boldsymbol{\mathcal{F}}_s^m$ denote the desired and measured spring force. $\mathbb{K}_o^{-1}$ and $\mathbf{O}^+$ stand for stiffness matrix and offset, respectively.

2) PID with Jacobian Feed-Forward: The knowledge of the plant (robotic manipulator) can be included into the control for chain decoupling and compensation of known disturbances. However, the dynamic and kinematic models of parallel structures in general require too much computation time to include into an on-line loop, therefore several simplifications and assumptions are introduced here.

It is assumed that Coriolis and centrifugal effects have little or no influence because the required speeds and accelerations are small. The error between the desired and the actual pose is assumed to be small. Further, it is assumed that the total moving mass is located at the position of the coil. These assumptions allow the use of the Jacobian to calculate the required motor torques.

The corresponding control scheme can be found in Figure 3, "Jacobian Feed-Forward". The required end-effector wrench is calculated and substituted into Equation 4 resulting in Equation 7 which produces the feed-forward torque

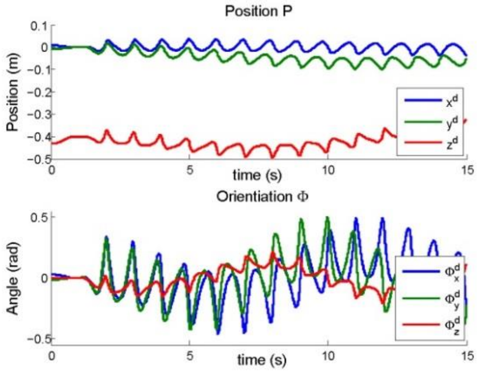


Fig. 4. The trajectory for the evaluation of the controller. The top figure depicts the desired position and the bottom figure shows the desired orientation of the coil.

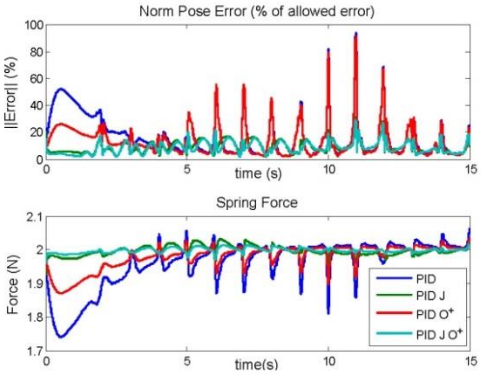


Fig. 5. Top figure shows the normalized error for the four controllers. Bottom figure shows resulting spring force. The PID stands for PID controlled, J and O^+ denote Jacobian Feed-Forward and Spring Force Control respectively.

($\boldsymbol{\tau}_{FF}$). $\ddot{\mathbf{X}}^d$ denotes the desired end-effector acceleration, $\mathbb{M}$ the generalized mass matrix and $\mathbf{G}$ the gravitational acceleration. The desired spring force $\boldsymbol{\mathcal{F}}_s^d$ is added to this equation to compensate for the spring force disturbance.

$$J : \boldsymbol{\tau}_{FF} = \mathbb{J}^{-T}\left[\mathbb{M}\left(\ddot{\mathbf{X}}^d + \mathbf{G}\right) + \boldsymbol{\mathcal{F}}_s^d\right] \tag{7}$$

3) PID with Spring Force Control: It may be expected that the low update frequency and low resolution of the optical tracking system and the flexible connection of the markers to the head will induce some pose error. This error is absorbed by the manipulator spring resulting in an altered spring force.

To obtain a more constant contact force, the measured spring $\boldsymbol{\mathcal{F}}_c^m$ force is used to change the motion trajectory by an offset $\mathbf{X}^{d+}$, allowing safe stimulation regardless of pose errors (Figure 3, "Spring Force Control"). The control law is given by Equation V-A.3, in which $\mathbb{K}_o$ is the pose-dependent virtual stiffness matrix.

$$O^+ : \mathbf{X}^{d+} = \mathbf{X}^d + \mathbb{K}_o^{-1}\left(\boldsymbol{\mathcal{F}}_s^d - \boldsymbol{\mathcal{F}}_s^m\right) \tag{8}$$

4) PID with Jacobian and Spring Force Control: The fourth controller presented here is a combination of the three controllers described earlier. Its control scheme may be found in Figure 3 by combining all optional components.

B. Controller Evaluation

The four controllers are implemented into 20-sim and connected a 3d rigid body model. A typical motion profile for walking is applied to the model. The motion profile consists of a multisine with frequencies between 1 and 10 Hz and a maximum acceleration of 10 m/s^2 (see Figure 4). The normalized percentage of the tracking error and the spring force are used as measures for the performance of the controllers.

The tracking error and spring force are plotted in Figure 5. It can be observed that the four controllers all follow the trajectory within the error margins of Table I. This suggests that the kinematic model is sufficiently accurate and all the controllers are stable.

Inspection of the error plot of Figure 5 shows that the constant gravitational disturbance is compensated by the integral action of the PID controller. The largest error is observed during maximal acceleration; the dynamic error. This error is pose-dependant and results in changed spring force (Figure 5 bottom).

The addition of the Jacobian Feed-Forward reduces the dynamic and gravitational error strongly, resulting in a more constant spring force.

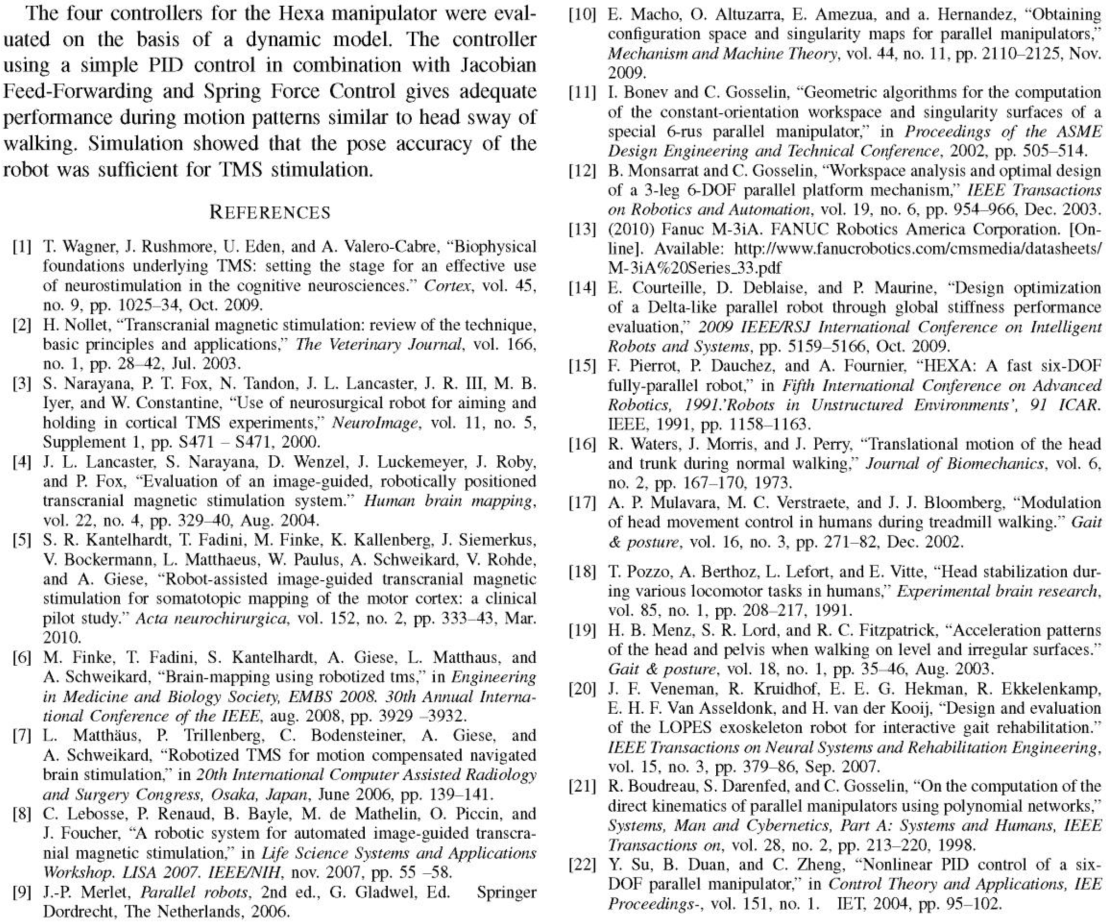
The Spring Force Control reduces the gravitational error, the spring force variation, but does not result in a reduced dynamic error.

The combination of the Jacobian Feed-Forward and the Spring Force Control results in largest error reduction and the smoothest spring force. Therefore this system is chosen to be most suitable for controlling the Hexa.

VI. Discussion and Conclusion

The application of TMS during activities such as treadmill walking gives need for robotic guided TMS. For this purpose, a method was presented in this paper to evaluate and compare parallel robots for safe human interaction.

The method considers the motor speed, motor acceleration, motor torque, and workspace accuracy as measures for the robotic safety and quality of the workspace. This method was applied for evaluation and comparison of the Delta robot, Hexaglide and the Hexa. The Hexa manipulator was found to most suitable. be the best of the three for safe human interaction. This is because the kinetic energy and consequently the required motor torque was lower than the other concepts and no singular was found in the required workspace.



The four controllers for the Hexa manipulator were evaluated on the basis of a dynamic model. The controller using a simple PID control in combination with Jacobian Feed-Forwarding and Spring Force Control gives adequate performance during motion patterns similar to head sway of walking. Simulation showed that the pose accuracy of the robot was sufficient for TMS stimulation.

REFERENCES

[1] T. Wagner, J. Rushmore, U. Eden, and A. Valero-Cabre, "Biophysical foundations underlying TMS: setting the stage for an effective use of neurostimulation in the cognitive neurosciences." *Cortex*, vol. 45, no. 9, pp. 1025–34, Oct. 2009.

[2] H. Nollet, "Transcranial magnetic stimulation: review of the technique, basic principles and applications," *The Veterinary Journal*, vol. 166, no. 1, pp. 28–42, Jul. 2003.

[3] S. Narayana, P. T. Fox, N. Tandon, J. L. Lancaster, J. R. III, M. B. Iyer, and W. Constantine, "Use of neurosurgical robot for aiming and holding in cortical TMS experiments," *NeuroImage*, vol. 11, no. 5, Supplement 1, pp. S471 – S471, 2000.

[4] J. L. Lancaster, S. Narayana, D. Wenzel, J. Luckemeyer, J. Roby, and P. Fox, "Evaluation of an image-guided, robotically positioned transcranial magnetic stimulation system." *Human brain mapping*, vol. 22, no. 4, pp. 329–40, Aug. 2004.

[5] S. R. Kantelhardt, T. Fadini, M. Finke, K. Kallenberg, J. Siemerkus, V. Bockermann, L. Matthaeus, W. Paulus, A. Schweikard, V. Rohde, and A. Giese, "Robot-assisted image-guided transcranial magnetic stimulation for somatotopic mapping of the motor cortex: a clinical pilot study." *Acta neurochirurgica*, vol. 152, no. 2, pp. 333–43, Mar. 2010.

[6] M. Finke, T. Fadini, S. Kantelhardt, A. Giese, L. Matthaus, and A. Schweikard, "Brain-mapping using robotized tms," in *Engineering in Medicine and Biology Society, EMBS 2008. 30th Annual International Conference of the IEEE*, aug. 2008, pp. 3929 –3932.

[7] L. Matthäus, P. Trillenberg, C. Bodensteiner, A. Giese, and A. Schweikard, "Robotized TMS for motion compensated navigated brain stimulation," in *20th International Computer Assisted Radiology and Surgery Congress, Osaka, Japan*, June 2006, pp. 139–141.

[8] C. Lebosse, P. Renaud, B. Bayle, M. de Mathelin, O. Piccin, and J. Foucher, "A robotic system for automated image-guided transcranial magnetic stimulation," in *Life Science Systems and Applications Workshop. LISA 2007. IEEE/NIH*, nov. 2007, pp. 55 –58.

[9] J.-P. Merlet, *Parallel robots*, 2nd ed., G. Gladwel, Ed. Springer Dordrecht, The Netherlands, 2006.

[10] E. Macho, O. Altuzarra, E. Amezua, and a. Hernandez, "Obtaining configuration space and singularity maps for parallel manipulators," *Mechanism and Machine Theory*, vol. 44, no. 11, pp. 2110–2125, Nov. 2009.

[11] I. Bonev and C. Gosselin, "Geometric algorithms for the computation of the constant-orientation workspace and singularity surfaces of a special 6-rus parallel manipulator," in *Proceedings of the ASME Design Engineering and Technical Conference*, 2002, pp. 505–514.

[12] B. Monsarrat and C. Gosselin, "Workspace analysis and optimal design of a 3-leg 6-DOF parallel platform mechanism," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 6, pp. 954–966, Dec. 2003.

[13] (2010) Fanuc M-3iA. FANUC Robotics America Corporation. [Online]. Available: http://www.fanucrobotics.com/cmsmedia/datasheets/M-3iA%20Series_33.pdf

[14] E. Courteille, D. Deblaise, and P. Maurine, "Design optimization of a Delta-like parallel robot through global stiffness performance evaluation," *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5159–5166, Oct. 2009.

[15] F. Pierrot, P. Dauchez, and A. Fournier, "HEXA: A fast six-DOF fully-parallel robot," in *Fifth International Conference on Advanced Robotics, 1991.'Robots in Unstructured Environments', 91 ICAR*. IEEE, 1991, pp. 1158–1163.

[16] R. Waters, J. Morris, and J. Perry, "Translational motion of the head and trunk during normal walking," *Journal of Biomechanics*, vol. 6, no. 2, pp. 167–170, 1973.

[17] A. P. Mulavara, M. C. Verstraete, and J. J. Bloomberg, "Modulation of head movement control in humans during treadmill walking." *Gait & posture*, vol. 16, no. 3, pp. 271–82, Dec. 2002.

[18] T. Pozzo, A. Berthoz, L. Lefort, and E. Vitte, "Head stabilization during various locomotor tasks in humans," *Experimental brain research*, vol. 85, no. 1, pp. 208–217, 1991.

[19] H. B. Menz, S. R. Lord, and R. C. Fitzpatrick, "Acceleration patterns of the head and pelvis when walking on level and irregular surfaces." *Gait & posture*, vol. 18, no. 1, pp. 35–46, Aug. 2003.

[20] J. F. Veneman, R. Kruidhof, E. E. G. Hekman, R. Ekkelenkamp, E. H. F. Van Asseldonk, and H. van der Kooij, "Design and evaluation of the LOPES exoskeleton robot for interactive gait rehabilitation." *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 15, no. 3, pp. 379–86, Sep. 2007.

[21] R. Boudreau, S. Darenfed, and C. Gosselin, "On the computation of the direct kinematics of parallel manipulators using polynomial networks," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 28, no. 2, pp. 213–220, 1998.

[22] Y. Su, B. Duan, and C. Zheng, "Nonlinear PID control of a six-DOF parallel manipulator," in *Control Theory and Applications, IEE Proceedings-*, vol. 151, no. 1. IET, 2004, pp. 95–102.

Appendix B. RANGE OF MOTION OF THE HEAD DURING TREADMILL WALKING

A. Introduction

In order to determine the required range of motion and velocity and acceleration of the robot, we need to know the range of motion of the head and the dynamics such as velocity and acceleration. A literature study on the range of motion is done in the appendix of the master thesis [37]. The result is reported in Table 6. To validate the literature study and to measure all the different parameters in the same time, a measurement study was done here.

B. Methods

The test are done on a 46 year old healthy subject during treadmill walking. The head motion is measured using 3D motion capturing system Visualeyez. The subject walked on different speeds ranging from 1 km/h to 12 km/h. The motion of the vertex of the head were measured and reported here.

C. Results

Figure 44-Figure 46 show a typical motion profile of a subject walking at 5 km/h. The motion of the head resembles a butterfly. During one step cycle the human sways from left to right once and twice up to down. The motion is also bended a little around the vertical axis as the subject also looks from left to right.

It should be noted that the orientation of the head has a drift over several steps. This happens when the subject is not tasked keep focus on one spot on the wall.

The range of motion and velocity the subject where measured with treadmill walking with different walking velocities. Figure 47 shows a clear relation between the range of motion and the walking velocity also the walking speed has a relation with the velocity of the head. Also the acceleration of the head has a trend reaching 8 m/s^2 . This acceleration however is difficult to calculate for it requires differentiation twice. The rotation acceleration is not given as it shows only noise. The estimated maximal values are shown in Table 6.

D. Conclusion

From the values from literature and the measured values the range of motion and the dynamic behavior of the head during walking can be quantified. This can form the basis of the list of requirements of the design of a robot during walking.

Table 6. Translation, rotation and acceleration of the head found in literature and test.

Name	x**(m)**	y**(m)**	z**(m)**	ϕ_x**(deg)**	ϕ_y**(deg)**	ϕ_z**(deg)**	$\ddot{x}$**(m/s^2)**	$\ddot{y}$**(m/s^2)**	$\ddot{z}$**(m/s^2)**	**Velocity (m/s)**
Holt [52]	-	-	0.03 (-)	-	-	-	-	-	-	-
Hirasaki [33]	-	-	0.04 (0.02)	-	3.25(-)	-	-	-	7.0 (0.3)	2.0
Waters [31]	0.02(-)	0.11(-)	0.10 (-)	-	-	-	-	-	1.4 (-)	1.57
Pozzo [53]	-	-	0.09 (-)	-	8.5 (2.5)	-	-		7.0 (-)	Walking
Laudani [54]	-	-	-	-	3.2 (-)		-	-	-	First step
Mulavara [55]	-	-	-	2.01(0.102)	2.36(0.14)	2.10 (0.15)	-	-	-	1.79
Kavanagh [32]	-	-	-	-	-	-	2.0 (-)	2.0 (-)	4.0 (-)	1.20
Latt[56]	-	-	-	-	-	-	1.5 (-)	1.0 (0.5)	3.5 (5.0)	2.2
Menz [57]	-	-	-	-	-	-	3.0 (-)	3.0(-)	7.5(-)	-
Subject (5 km/h)	0.05	0.05	0.05	8	8	8	-	-	-	2.5
Concluding Values	0.08	0.08	0.08	15	15	15	10	10	10	

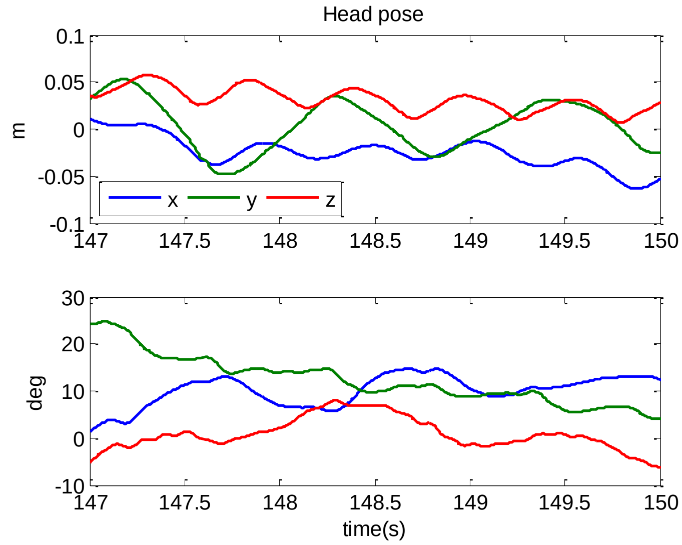


Figure 43. Head pose during walking of 5 km/h.

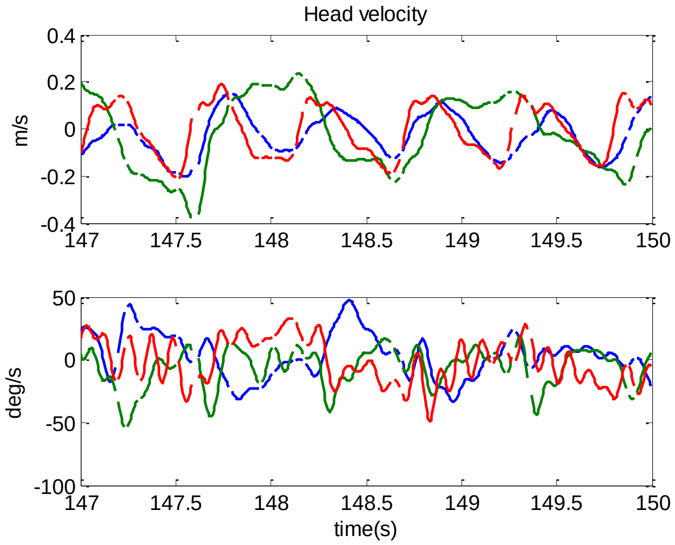


Figure 44. Head velocity during walking of 5 km/h.

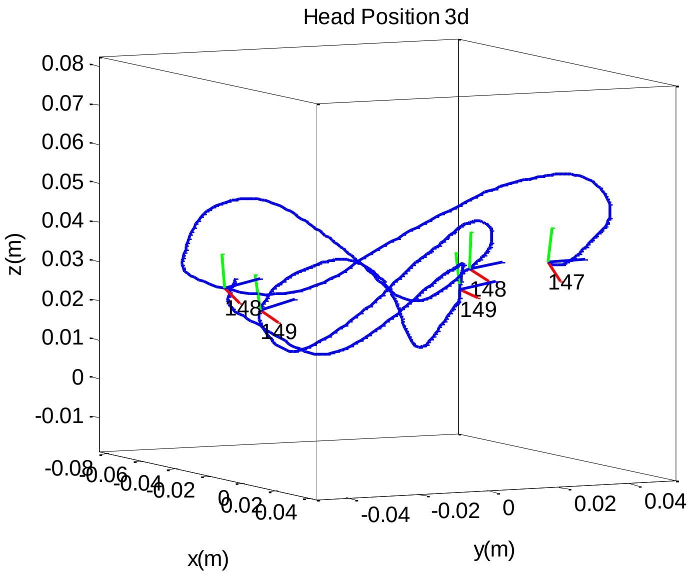


Figure 45. Head position shown here in 3D during walking of 5 km/h. Five orientations are given by the axis.

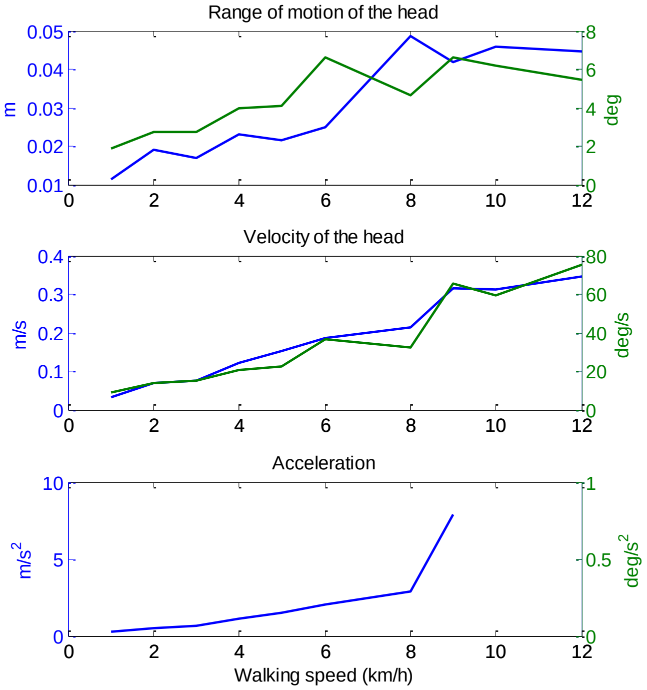


Figure 46. Relation between the walking speed and the range of motion (normalized), the head velocity and acceleration of the head

Appendix C. EVALUATION OF VISUALEYEZ, THE OPTICAL TRACKER SYSTEM

A. *Introduction*

The Visualeyez camera system is used to measure the pose of the head relative to the base of the robot. Usually this means that there are 3 markers placed on the base an 6 markers placed on the head of the subject. The Visualeyez system now only runs on a non-realtime windows PC. The is recorded on the windows PC and send over UDP trough to the realtime xPC Target using a C program called "sendVZtoUPD.exe" This program has been made at the BW group by Nikolai Melikian. The xPC target controls the robot and steers it the desired position.

There are some issues noticed during the operation with Visualeyez. First the *noise level* seemed to be higher than the specified 0.1 mm RMS in the specs. Furthermore a time delay up to 60 ms has been observed between the measurement and receiving it on the xPC target. Thirdly sometimes the marker seem to *jump* from one place to the other more than couple of centimeter up to meters within sample time. The first two issues are being investigated here to find the cause and find settings to reduce these issues. The jumpy markers are not yet investigated it was not clear how these test should be done.

B. *Methods*

Several test are to be done to determine the nature of the issues. For each issue the settings of the baseline measurement are given and also the variables for each test are introduced. This means that in each test all but one variable are changed.

1) *Noise level*

First of all the noise level during stand still is measured for different settings. This is done by placing the markers on a tripod with a spherical head. The parameters which are expected to have influence on the noise are;

1. Sampling time, time between measurements
2. Measurement time, measurement time for each marker
3. Distance to the camera in the normal direction
4. X-position, distance from the center of the cameras
5. Orientation with respect the normal of the camera
6. Lamp on or off.

Setting	Baseline value	Variable values	Unit
Sample time	100	[10 50 100 150]	Hz
Measurement time	2.250	[0.240,1.250, 2.250]	ms
z-Position	1.4	[0.8, 1.4,2.30]	m
x-Position	0	[0,0.5,0.9]	m
Orientation	0	[0,30,60,90]	deg

Table 7. The TMS robot setup with all the auxiliary apparatus such as the head tracking mechanism and controller unit.

The amount of noise is identified using the estimate of the standard deviation for each setting in the three directions of the camera. It is also investigated whether a high pas filter is required to remove the motion artifacts.

2) *Time delay*

A time delay of 60 ms is observed between Visualeyez and the robot this can have several causes. One thing under investigation is that Visualeyez measures every LED at a different time but sends them/saves them at once.

1. VZ tracker in itself. Internal process takes some time
2. Com to USB. A virtual comport is used with USB, this might cause some of the delay since it buffers.
3. Windows process. The windows process takes some time. Other processes like mouse movement or displaying stuff can make interrupts.
4. Saving the data on windows, VZ to Harddisk
5. VZ display, visualization on windows
6. UDP transmission. The data is send over by UDP, this is also possible cause for delay as it might use some buffering process/interrupts oid.
7. Time of sampling
8. Sampling rate
9. Number of markers

We can estimate delay by letting the robot make a chirp movement and let the robot measure that. That data is send through to the robot and saved there. The time delay can be deduced from the shift of the robot movement. The time delay over complete measurement and also the variance in time delay is to be investigated

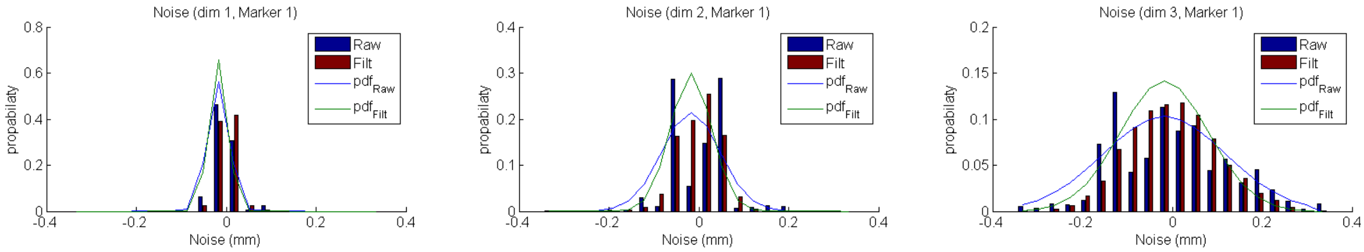


Figure 47. Baseline plot of noise. Showing raw data (Raw) and high pass filtered data (Filt) together with the gaussian fits (pdf-raw, pdf-filt). Here only for marker 1 is shown, similar plots are there for markers 2 and 3.

Setting	Baseline value	Variable values	Unit
Sample time	100	[10 50 10]	Hz
Measurement time	0.769	[0.240,0.500, 0.769]	ms
Graphics	On	[on, off]	-
Saving	On	[on, off]	-
Display VzSend	Off	[on, off]	-
Connected to xPC	Off	[on, off]	-

Table 8. Settings of the test to find the optimal settings for reduction of time-delay.

C. *Results*

The results of the different test concerning the noise level and the time delay are reported here.

1) *Noise level*

In Figure 48 an example plot of the noise (Baseline measurement). The difference with the mean (raw) and the high pass filtered signal (filt) are shown for the 3 dimensions (x,y,z in the camera frame) and of the 3 markers (501-503). To this error signals a Gaussian distribution is plotted (pdf-raw and pdf-filt).

It can be seen that the noise more or less represent a Gaussian bell shaped curve. Since the output values of Visualeyez are on a grid, its resolution not all intermediate values are seen in the unfiltered data. From now on the width of the Gaussian curve is used. This, the standard deviation, of the raw (unfiltered) data as a measure for noise.

It is also seen, which holds in general that the z-direction is most subject to noise. This can be explained by the fact that this is away from the camera and most difficult in triangulation.

The resulting bar diagrams showing the std of the noise for each condition can be found in the appendix. In the table below the results are summarized.

Setting	Influence
Sample time	No Influence
Measurement time	No Influence
z-Position	Noise worse with distance
x-Position	More noise near edge
Orientation	No influence

Table 9. Resulting influence of settings on the noise level.

2) *Time delay*

The measurement of the robot is measured with Visualeyez and send trough UDP to the robot. The robot also has an internal measurement of the pose. These two measurements are compared to see the time delay.

In figure 2 the time series of the measurement data are shown. There are some observations already visible.

Measurement of markers are send trough as soon as new info is available. This can clearly be seen in the 5th graph. Here the sample rate is 10 Hz. It can be seen that each marker point reaches the robot around this frequency, but shifted with respect to each other. The hypothesis was that the markers are only send trough to the robot when all markers are measured. This is shown not to be true. This also means that the time delay for each marker is similar.

These measurements have to be confirmed still. It is especially strange that the connect to xPC has a positive effect on the time delay. It might be that there are some other settings also changed.

Setting	Influences
Base line	32ms
Frame rate	Less to send, less overhead time
Measure time	More idle time, less delay
Graphics off	More idle time, less delay
Display Vzsend	More processes more delay
Connected to xPC	???
Saving off	Less processes, less delay

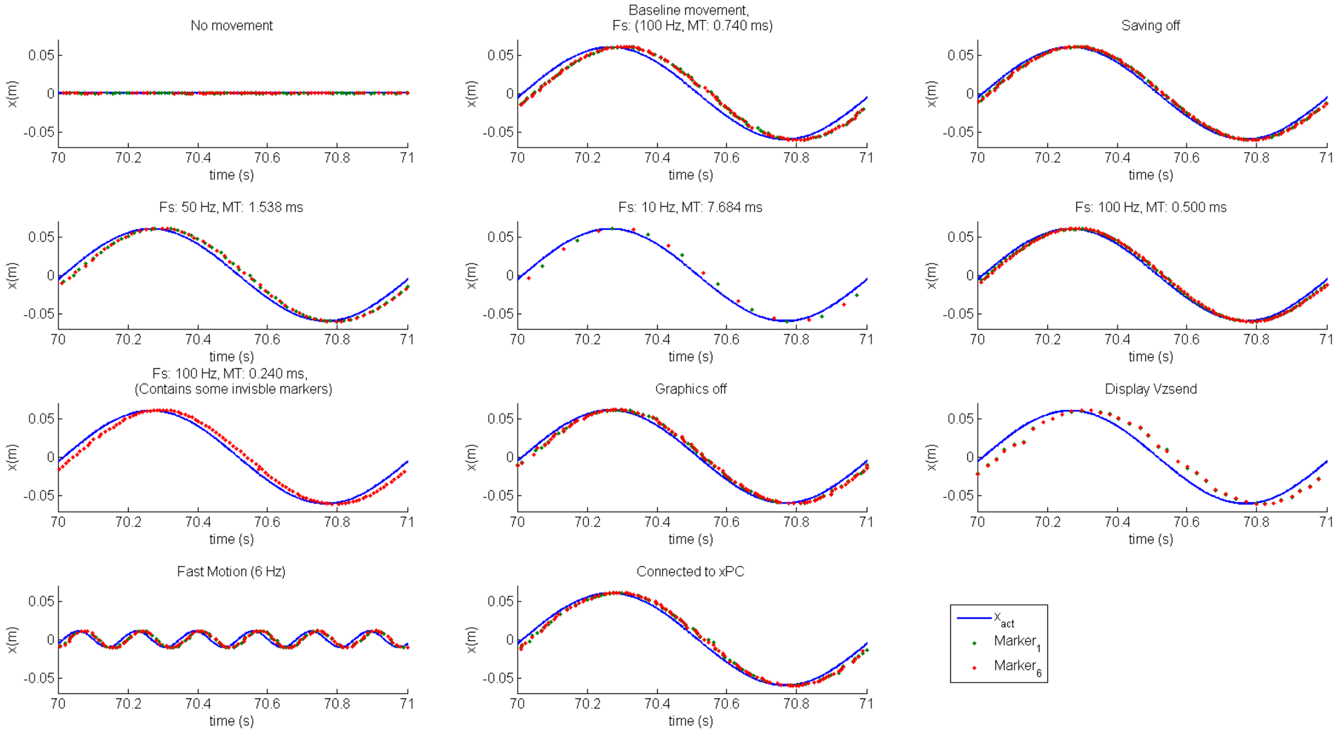


Figure 48. All time series of the measurements. The actual position of the robot is plotted together with the marker data. Only markerdata 1 and 6 are shown.

Table 10. The result of the influence of various settings on the time delay.

D. Conclusions and discussions

From these measurement several conclusions can be drawn.

1) Noise

There is far less noise as expected from experience. The average noise is around 0.05 mm std. This is quite good. This means that around 60 % of the markers position stay within 0.05 mm. However the noise is very dependent on the position in the FoV of the camera's. Farther away and to the sides the noise increases a lot, up to 0.3 mm. Other settings did not seem to have much influence on the noise level.

It might prove the effort to be able to model the noise level depending on the position in the FoV if a Kalman filter is used.

2) Time-Delay

The time delay is possible of about 10 ms. A lot of factors have influence on the time delay.

One major surprise was that the markers are send on the fly to the robot. Whenever there is a new marker measured this data is send via UDP to the robot. This means that the marker cloud is changing over time during motion. There might be some investigation how to incorporate this in to the control loop.

It might be worth noticing that these measures have to be confirmed in second measurements.

It should however be noted that the jumpy marker has not yet been identified. More work should be done to find the origin and solution this strange glitch in the system.

There is now a set of optimal settings defined as can be found in Table 11. The concluding optimal settings for Visualeyez setup. This includes a trade-off for example the frame rate is not desired to be low as it will degrade the fluency of tracking. The results from the investigation on the jumping markers have to be applied here.

Setting	**Value**
Frame rate	100 Hz
Measurement time	0.769 ms
z-Position	1 m
x-Position	Center
Orientation	~
Graphics	Off
Display VZsend	Off
Connected to xPC	Off
Saving	Off

Table 11. The concluding optimal settings for Visualeyez setup.

Appendix D. Additional information on Kalman filter

A. *Jacobian of the quaternion evolution.*

In the formulation of the Jacobian of the quaternion evolution with respect to angular velocity is a simplification with respect to discrete time. During this discrete time the quaternion evolves according the exponential mapping. It is therefore not immediately clear that the linear version (65) can be used. In this section the derivative of the exponential mapping for discrete time is given and the approximation using Taylor expansions show that the linear approximation is a valid choice for small time steps.

Equation (26) can be written using the right quaternion multiplication matrix. In which the resulting right vector contains all the derivative variables.

$$\boldsymbol{F}_{q\omega} = [\boldsymbol{q}(k) \otimes]^T \frac{\partial}{\partial \boldsymbol{\omega}} \begin{bmatrix} \frac{\boldsymbol{\omega}}{\|\boldsymbol{\omega}\|} \sin\left(\frac{\Delta t \|\boldsymbol{\omega}\|}{2}\right) \\ \cos\left(\frac{\Delta t \|\boldsymbol{\omega}\|}{2}\right) \end{bmatrix} \tag{123}$$

This gives the total Jacobian matrix

$$\begin{aligned} \boldsymbol{F}_{q\omega} = [\boldsymbol{q}(k) \otimes]^T \Bigg(& \begin{bmatrix} \frac{\Delta t}{\|\boldsymbol{\omega}\|^2} \cos\left(\frac{\Delta t \|\boldsymbol{\omega}\|}{2}\right) \boldsymbol{\omega}\boldsymbol{\omega}^T \\ -\frac{\Delta t}{2\|\boldsymbol{\omega}\|} \sin\left(\frac{\Delta t \|\boldsymbol{\omega}\|}{2}\right) \boldsymbol{\omega}^T \end{bmatrix} \\ & + \frac{1}{\|\boldsymbol{\omega}\|} \sin\left(\frac{\Delta t \|\boldsymbol{\omega}\|}{2}\right) \begin{bmatrix} \boldsymbol{I_3} - \frac{\boldsymbol{\omega}\boldsymbol{\omega}^T}{\|\boldsymbol{\omega}\|^2} \\ 0 \end{bmatrix} \Bigg) \end{aligned} \tag{124}$$

Taylor expansion of this matrix with respect to Δt up to the order 3 gives for the trigonometric function

$$\Delta t \cos\left(\frac{\Delta t \|\boldsymbol{\omega}\|}{2}\right) \approx \Delta t, \qquad \Delta t \sin\left(\frac{\Delta t \|\boldsymbol{\omega}\|}{2}\right) \approx \frac{\Delta t^2 \|\boldsymbol{\omega}\|}{2}, \\ \sin\left(\frac{\Delta t \|\boldsymbol{\omega}\|}{2}\right) \approx \frac{\Delta t \|\boldsymbol{\omega}\|}{2}$$

Since:

$$\frac{\Delta t}{2\|\boldsymbol{\omega}\|^2} \boldsymbol{\omega}\boldsymbol{\omega}^T + \frac{\Delta t \|\boldsymbol{\omega}\|}{2} \left(\frac{1}{\|\boldsymbol{\omega}\|} \boldsymbol{I_3} - \frac{\boldsymbol{\omega}\boldsymbol{\omega}^T}{\|\boldsymbol{\omega}\|^3} \right) = \frac{\Delta t}{2} \boldsymbol{I_3}$$

The Jacobian can be approximated with:

$$\boldsymbol{F}_{q\omega} = [\boldsymbol{q}(k) \otimes]^T \begin{bmatrix} \frac{\Delta t}{2} \boldsymbol{I}_3 \\ \frac{\Delta t^2}{4} \boldsymbol{\omega}^T \end{bmatrix} \tag{125}$$

As the lower row of the matrix is depending on the square of the small discrete time step it is approximated to zero.

B. *Cross method*

A more simplified method of fitting a frame in a set of points is the cross method. It places the frame at the first point and uses the difference to the other points to fit a orthogonal base through the first 3 points. Inspect Figure 50 to see graphical explanation of the method. Its name is derived from the cross product which is used for orthogonality.

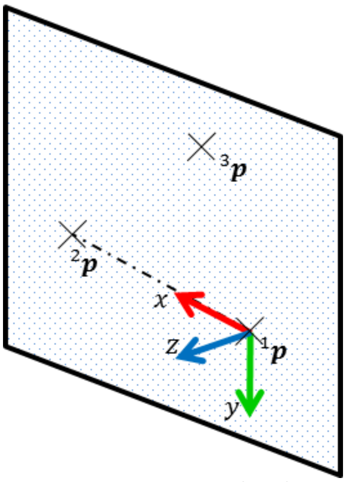


Figure 49 Cross method. It places the origin of the frame at the first point ($^1\boldsymbol{p}$), orients the first axis (x) in the direction of the second point ($^2\boldsymbol{p}$). And makes the $x - y$ plane intersect with the third point($^3\boldsymbol{p}$).

For the first three points of the point cloud the following relation hold:

$$\begin{bmatrix} 0 & {}^2_x p^{Hm} & {}^3_x p^{Hm} \\ 0 & 0 & {}^3_y p^{Hm} \\ 0 & 0 & 0 \end{bmatrix} = \boldsymbol{H}_{Vz}^{Hm} \begin{bmatrix} {}^1\check{\boldsymbol{p}}^{Vz} & {}^2\check{\boldsymbol{p}}^{Vz} & {}^3\check{\boldsymbol{p}}^{Vz} \end{bmatrix} \tag{126}$$

This means that the first marker is placed at the origin. The second is located along the x-direction. And the x-y plane is fit through the x-axis and the third marker.

The inverse of the transformation matrix given such that:

$$(\boldsymbol{H}_{Vz}^{Hm})^{-1} = \boldsymbol{H}_{Hm}^{Vz} = \begin{bmatrix} {}_x\boldsymbol{R} & {}_y\boldsymbol{R} & {}_z\boldsymbol{R} & {}^1\check{\boldsymbol{p}}^{Vz} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{127}$$

With rotation matrices:

$$\begin{aligned} {}_x\boldsymbol{R} &= \frac{{}^2\check{\boldsymbol{p}}^{Vz} - {}^1\check{\boldsymbol{p}}^{Vz}}{\|{}^2\check{\boldsymbol{p}}^{Vz} - {}^1\check{\boldsymbol{p}}^{Vz}\|}, \\ {}_z\boldsymbol{R} &= \frac{{}_x\boldsymbol{R} \times ({}^3\check{\boldsymbol{p}}^{Vz} - {}^1\check{\boldsymbol{p}}^{Vz})}{\|{}_x\boldsymbol{R} \times ({}^3\check{\boldsymbol{p}}^{Vz} - {}^1\check{\boldsymbol{p}}^{Vz})\|}, \\ {}_y\boldsymbol{R} &= {}_z\boldsymbol{R} \times {}_x\boldsymbol{R} \end{aligned} \tag{128}$$

Appendix E. xPC CONTROLLER SCHEME OF TMS ROBOT

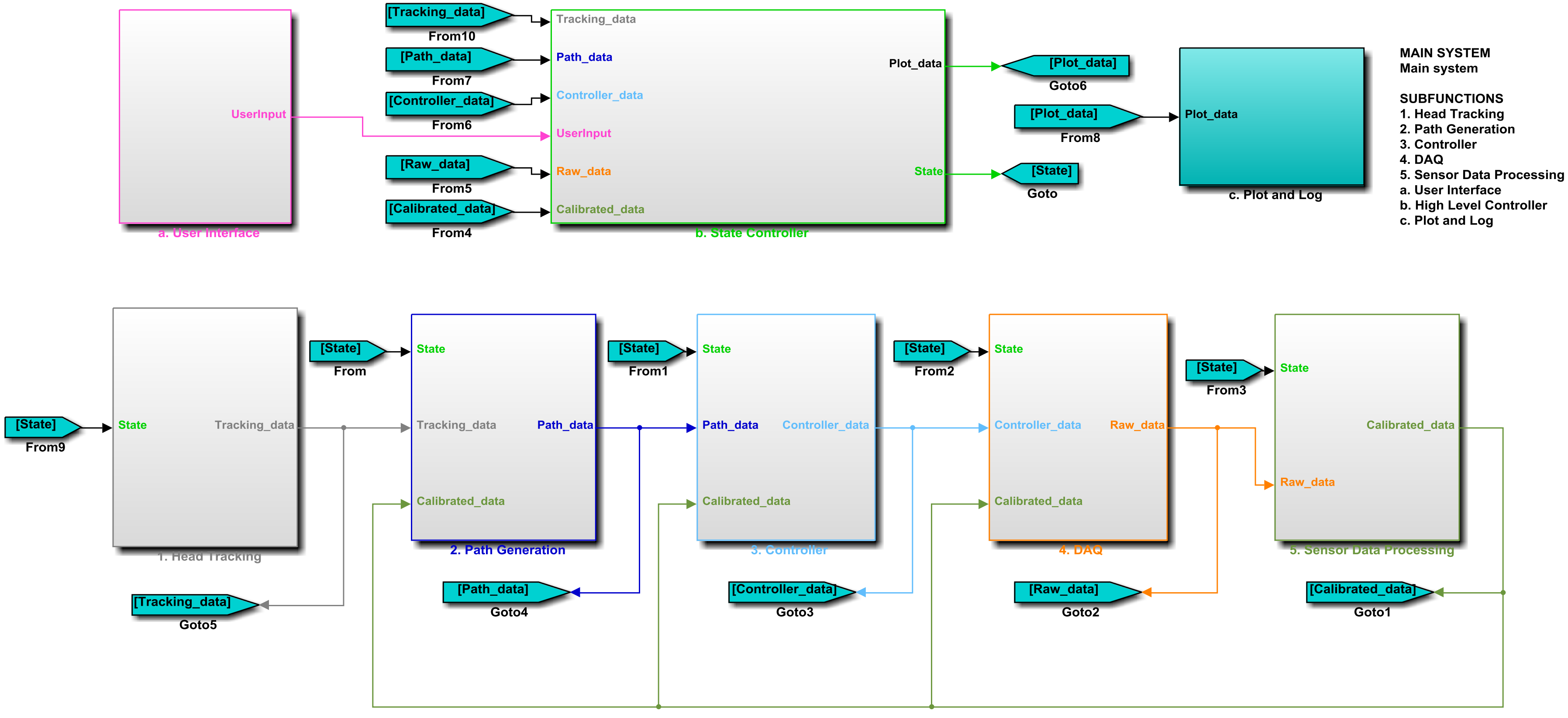


Figure 50 – Highest hierarchical scheme of the controller as implemented in to xPC Simulink. Here several subsections can be recognized. a. User Interface, b. State controller, c. Plot and log, 1. Head tracking, 2. Path generation, 3. Controller, 4. DAQ, 5. Sensor Data Processing.