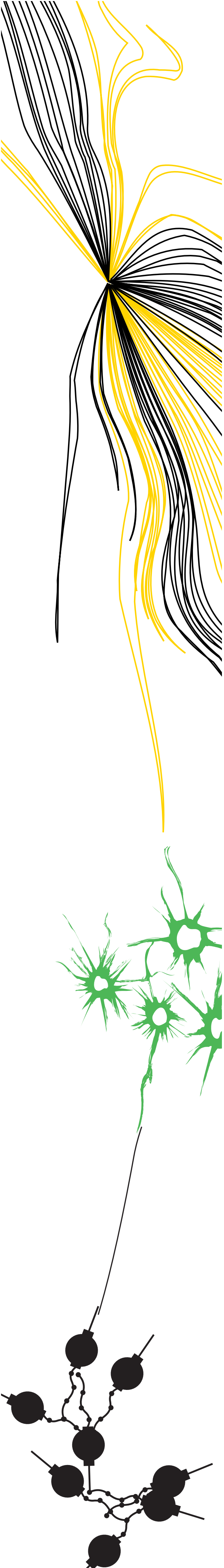


HUMAN IMPEDANCE REGULATION IN HAPTIC CONTROL USING ELECTROMYOGRAPHY COMBINED WITH THE POSTURE OF THE HUMAN ARM TO INCREASE TRANSPARENCY WITHIN TELEMANUPULATION CONTROL

H.G.D. (Huub) Braakman

MSC ASSIGNMENT

Committee:
dr. ir. D. Dresscher
dr. ing. G. Englebienne

July 2021

042RaM2021
Robotics and Mechatronics
EEMathCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Preface

The work presented in this thesis is written to conclude the Master of Science programme in Electrical Engineering within the Robotic and Mechatronics group (RaM) at the University of Twente. The presented work is done for the i-Botics innovation centre, which is a collaboration between TNO and the University of Twente.

At first, I would like to thank my daily supervisor, Dr.ir. Douwe Dresscher, for pointing me in the right direction once things became unclear. Furthermore, I would like to thank Sara Falcone (MSc) and Luc Schootuiterkamp (MSc) for helping me setting up and testing the user study. Likewise, I would like to thank Robin Lieftink (MSc), who helped me understand the basic control of the used robotic arm and haptic device. At last, I would like to thank the participants who were very interested in my research and helped me conduct the user study, which allowed me to collect insightful data.

Summary

Within telerobotics, the goal is to execute tasks remotely with a robotic system. A human operator should control the robotic system at the remote site. To execute the tasks as effectively as possible the operator should be provided with visual and force feedback. In an ideal situation, perfect transparency will be achieved.

To achieve this ideal situation research is done regarding tele-impedance control, which should allow the operator to change the impedance of the robot arm at the remote site. Currently, this is done by normalizing surface EMG signals from a single muscle co-contraction pair in the forearm and using those to linear scale the endpoint stiffness between a lower and upper limit. However, this does not contain any information about the actual endpoint stiffness of the human arm.

To make the robot arm feel more like a natural extension of the human arm, it should also mimic the endpoint impedance of the human arm. This research is focused on that principle. A controller is developed, which is able to calculate the human arm endpoint stiffness by measuring surface EMG signals from the human arm and determining the posture of the arm. This data is obtained using two Myo armbands, one around the arm and the other around the forearm. The resulting endpoint stiffness is different in x, y and $z-$direction.

To compare the two controllers a user study is done. From the user study results, it followed that the newly designed controller shows promising results regarding comfort, prediction, precision, and consistency. However, the results are still very divergent, mainly caused by the orientation of the human arm and the muscle calibration phase. This an issue that could be improved and lead to more consistency in the tele-impedance control method.

Contents

List of Figures

List of Tables

Abbreviations and symbols

Abbreviations

Abbreviation	Definition
API	Application Programming Interface
CNS	Central Nervous System
DARE	Discrete time algebraic Riccati equation
DoF	Degrees of Freedom
EMG	Electromyography
IMU	Inertial Measurement Unit
LQR	Linear Quadratic Regulator
TTC	Time to Task Completion

Symbols - Greek

Symbol	Unit	Definition
Δt	[s]	Discretizing time step
λ_i	[m]	Muscle length, with i the muscle under investigation
$\lambda_{i,0}$	[m]	Initial muscle length of muscle i
$\Lambda(\mathbf{q})$	[kg]	Abbreviation for $\mathbf{J^{-T}}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{J^{-1}}(\mathbf{q})$
$\Lambda_{\mathbf{d}}$	[kg]	Positive definite matrix representing virtual mass
$\mu(\mathbf{q},\dot{\mathbf{q}})$	[kgs^{-1}]	Abbreviation for $\mathbf{J^{-T}}(\mathbf{q})\left(\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})-\mathbf{M}(\mathbf{q})\mathbf{J^{-1}}(\mathbf{q})\dot{\mathbf{J}}(\mathbf{q})\right)\mathbf{J^{-1}}(\mathbf{q})$
Ψ_i	[-]	Frame used for kinematic transformations, where i specifies the frame
ρ	[-]	Moment arm
$\tau_{\mathbf{acc}}$	[Nm]	Torque due to angular acceleration
$\tau_{\mathbf{Cmd}}$	[Nm]	Command torque for the slave robot to mimic the human arm compliance
τ_{DT}	[s]	Communication time delay
$\tau_{\mathbf{ext}}$	[Nm]	Torque caused by external forces
$\tau_{\mathbf{f}}$	[Nm]	Torque due to friction
$\tau_{\mathbf{g}}(\mathbf{q})$	[Nm]	Torque generated by gravity
$\tau_{\mathbf{total}}$	[Nm]	Torque acting on the rotating joint
τ_u	[s]	Time constant of the muscle
$\tau_{\mathbf{vel}}$	[Nm]	Torque due to angular velocity

Symbols - Latin

Symbol	Unit	Definition
$\mathbf{A_c}$	[-]	System matrix continuous time
$\mathbf{A_d}$	[-]	Discretized system matrix
$a(t)$	[-]	Muscle activation signal
$\mathbf{B_c}$	[-]	Input matrix continous time
$\mathbf{B_d}$	[-]	Discretized input matrix
$\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$	[kgm$^2 s^{-1}$rad^{-1}]	Coriolis/centrifugal matrix
$\mathbf{C_c}$	[-]	Output matrix continuous time
$\mathbf{D}(t)$	[kgs^{-1}]	Damping matrix with random entries ($d_{11}(t)$ till $d_{33}(t)$). Used to determine a velocity dependent random force
$\mathbf{D_d}$	[kgs^{-1}]	Positive definite matrix representing virtual damping
$\mathbf{F_{Cmd}}$	[N]	Control command that allows the robot arm to behave as a compliant mass-spring-damper system
$\mathbf{F_{ext}}$	[N]	External force acting on the end-effector
$\mathbf{f}_{\text{external}}(t)$	[N]	External, velocity dependent, random force field in 3D
f^i_{external}	[N]	Random force field component, where i denotes the direction
$\mathbf{F_g}(\mathbf{q})$	[N]	Force due to gravity acting on the end-effector
$\mathbf{f}_{\text{muscle}}(t)$	[N]	Force vector, containing $f^x_{\text{muscle}}(t)$, $f^y_{\text{muscle}}(t)$ and $f^z_{\text{muscle}}(t)$ of the human arm
$f^i_{\text{muscle}}(t)$	[N]	Muscle force of the human arm, where i defines the direction
$\mathbf{F_{total}}$	[N]	Total force acting on the end-effector
$\mathbf{g}$	[m]	Position vector, containing the goal location of the human arm (g^x, g^y and g^z)
g^i	[m]	Goal location of the human arm, where i defines the x, y or z direction
$\mathbf{H^b_a(0)}$	[-]	Homogeneous transformation matrix, from frame a to frame b, for the initial configuration
$\mathbf{H^b_a}(\mathbf{q})$	[-]	Homogeneous transformation matrix, from frame a to frame b
$\mathbf{I}$	[-]	Identity matrix
J	[-]	Accumulated cost
$\mathbf{J_H}(\mathbf{q})$	[-]	Geometric Jacobian matrix of the human arm
$\mathbf{J^\dagger_{H,v}}$	[-]	Pseudo inverse of $\mathbf{J_H}(\mathbf{q})$. It defines a mapping from the end-effector veloctiy $\dot{\mathbf{p}}_{\mathbf{ee}}(t)$ to the joint velocities $\dot{\mathbf{q}}$
$\mathbf{J_M}(\mathbf{q})$	[-]	Muscle Jacobian matrix
$\mathbf{J_R}(\mathbf{q})$	[-]	Jacobian matrix of the robot arm
$\mathbf{K}$, $\mathbf{K}_k$	[-]	Feedback gain matrix
$\mathbf{K_d}$	[kgm$^2 s^{-2}$]	Positive definite matrix representing virtual stiffness
$\mathbf{K_e}$	[Nrad^{-1}]	Endpoint stiffness matrix
K^i_{int}	[Nm^{-1}]	Intrinsic muscle stiffness of the specified muscle (i)
K_J	[Nrad^{-1}]	Joint stiffness
$\mathbf{K_J}$	[Nrad^{-1}]	Joint stiffness matrix
K^i_μ	[Nm^{-1}]	Muscle elasticity of the specified muscle (i)
$\mathbf{K}_\mu$	[Nm^{-1}]	Muscle elasticity/stiffness matrix (containing all muscles under investigation)
$K^i_{\mu,max}$	[Nm^{-1}]	Maximum elasticity/stiffness of the specified muscle (i)
K_v	[Nm^{-1}]	Virtual stiffness
$\mathbf{L}$	[-]	Weighting matrix, positive-definite
L_{la}	[m]	Length of the lower arm of a human
L_{ua}	[m]	Length of the upper arm of a human

Symbols - Latin (continued)

Symbol	Unit	Definition
$\mathbf{M}$	[kg]	Mass matrix containing the mass of the human arm endpoint in x, y and z-direction
$\mathbf{M}(\mathbf{q})$	[kgm^{2}rad^{-1}]	Inertia matrix
$\mathbf{p}(t)$	[m]	Position vector, containing $p^x(t)$, $p^y(t)$ and $p^z(t)$ of the human arm
$\mathbf{p_{ee}}$	[ms^{-1}]	End-effector position, 3D vector
$\mathbf{p_{sp}}$	[ms^{-1}]	Set point position, 3D vector. Contains the path to be followed by the end-effector
$\dot{\mathbf{p}}_{\mathbf{ee}}$	[ms^{-1}]	End-effector velocity, 3D vector
$\tilde{p}_a^b$	[m]	Skew symmetric representation of the position of frame a expressed in frame b
$p^i(t)$	[m]	Position of the human arm, where i specifies the direction
$\mathbf{q}$	[rad]	Robot joint angles
$q_{e,i}$	[rad]	Elbow angle with respect to the x, y, or z axis, where i defines the axis under investigation
q_i	[rad]	Joint angle, where i specifies the joint
$q_{s,i}$	[rad]	Shoulder angle with respect to the x, y, or z axis, where i defines the axis under investigation
$\dot{\mathbf{q}}$	[rads^{-1}]	Robot joint velocities
$\ddot{\mathbf{q}}$	[rads^{-2}]	Robot joint accelerations
R_i	[m]	Radius of a specific joint of the human arm, where i specifies the joint
R_a^b	[-]	Rotation matrix from frame a to frame b
$\mathbf{T}$	[-]	Weighting matrix, positive semi-definite
$\widehat{\mathbf{T}}_{\mathbf{a}}^{\mathbf{c,b}}$	[-]	Unit twist of frame a with respect to frame b expressed in frame c
$\tilde{\widehat{\mathbf{T}}}_{\mathbf{a}}^{\mathbf{c,b}}$	[-]	Skew symmetric matrix of an unit twist of frame a with respect to frame b, expressed in frame c
t_{delay}	[s]	Delay in the communication channel
u	[-]	Normalized muscle activation (EMG)
$\mathbf{u}(t)$	[-]	Motor command, continuous time
$\mathbf{v}(t)$	[ms^{-1}]	Velocity vector, containing $v^x(t)$, $v^y(t)$ and $v^z(t)$ of the human arm
$v^i(t)$	[ms^{-1}]	Velocity of the human arm, where i defines the direction
$\mathbf{V}_k$	[-]	Discrete time algebraic Riccati equation (DARE)
$w^{(q,i)}$	[-]	Weight of the weight matrix $\mathbf{T}$, where q is an element of the state vector $\mathbf{x}_k$, and i the direction in x, y or z
$\mathbf{x}(t)$, $\mathbf{x}_k$	[m, ms^{-1}, N, m]	State vector, continuos time and discrete time respectively, containing $\mathbf{p}(t)$, $\mathbf{v}(t)$, $\mathbf{f}_{\text{muscle}}$(t) and $\mathbf{g}$
$\mathbf{x_R}$	[m]	End-effector position of the robot arm
$\dot{\mathbf{x}}_{\mathbf{R}}$	[ms^{-1}]	End-effector velocity of the robot arm
$\ddot{\mathbf{x}}_{\mathbf{R}}$	[ms^{-2}]	End-effector acceleration of the robot arm
$\mathbf{x_{R,d}}$	[m]	Desired end-effector position of the robot arm
$\mathbf{x_{R,p}}$	[m]	Predicted end-effector position of the robot arm
$\ddot{\mathbf{x}}_{\mathbf{R,d}}$	[ms^{-2}]	Desired end-effector acceleration of the robot arm
$\mathbf{y}(t)$, $\mathbf{y}_k$	[m, ms^{-1}, N, m]	Output vector, continuous time and discrete time respectively)

1 Introduction

The use of teleoperation can extend the human capability to manipulate objects remotely. Within teleoperation, the operator at the control site is provided with the same information as present at the remote site. A more detailed overview can be seen in figure 1.1.

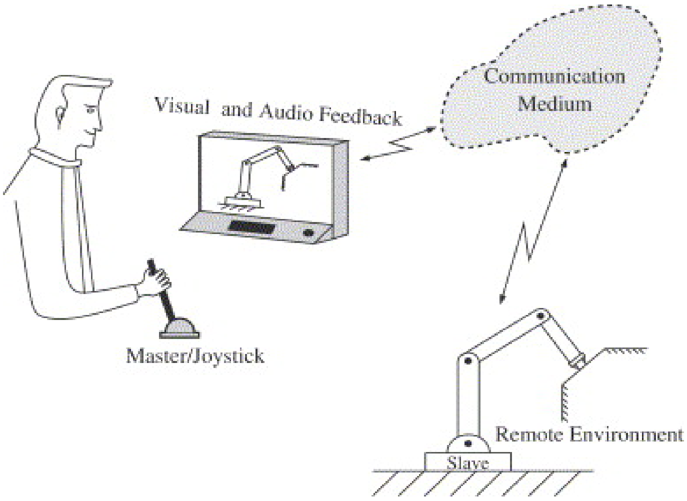


Figure 1.1: Teleoperation principle [1].

Once forces from the remote environment are reflected towards the master, the teleoperation is said to be bilateral [1]. Providing the human operator with a feeling of the remote environment to increase transparency is one of the main objectives for the system, together with stability. However, by the use of bilateral teleoperation, the communication time-delay together with incomplete information of the master and slave site induces instability [8, 9]. The instability causes a decrease in transparency.

A decent amount of research is done in order for the bilateral teleoperation to remain stable in the presence of time delay. However, in most cases, the stability is a trade-off with the transparency of the system[10, 11, 12, 13].

The reduced transparency leads to a decrease inadequate task performance within uncertain environments. Tasks, which humans usually perform without any problems, are no longer intuitively controllable by bilateral teleoperation. In the recent study of [14] impedance modulation of the slave device, by the use of Electromyography (EMG) measurements from muscles in the forearm of the master is used. This research showed that intuitive impedance modulation, through co-contraction of the muscles in the forearm, leads to a more intuitive control in unpredictable environments. However, the system's transparency is still decreasing once time delay is present in the communication channel.

To compensate for the effect of time delay in the communication channel the research of [14] is elaborated in [15], where a bi-directional impedance reflection controller is designed to overcome the negative effects of time delay. This is done by estimating and reflecting the impedance of the operator and the environment. In order to compensate for the delayed motion, a linear trajectory predictor is added. This resulted in a working system with better performance until a time delay of 50 ms. Increasing the time delay further resulted in instabilities in the system. However, without time delay, the bilateral impedance control with a passivity layer from [16] outperforms the bi-directional impedance reflection controller from [15].

Furthermore, from the work of [15] it followed that the measured EMG signals were hardly able to reach the maximum signals measured during calibration. This resulted in an overall, relatively low impedance for the slave controller. Due to this overall low impedance, overshoot occurred more easily. Furthermore, rigid objects were hard to sense since the operator felt only a low force.

To obtain a more realistic and intuitive impedance reflection the methodology of modulating the endpoint stiffness of the slave device will be changed.

In this paper, a new method to reflect the human arm impedance to the slave robot will be evaluated, inspired by the work done by [17]. However, the approach taken in this research will be different. Since, in [17], the exact same experiment on the master site is used to determine the human arm impedance and position and transfer it to the slave site, whilst in this experiment, a haptic device will be used at the master site, which controls the position at the slave site and gives haptic feedback to the user to allow for impedance manipulation. The haptic approach allows the user to interact with the slave environment instead of copying the master environment towards the slave environment. The haptic device will present the user information about the interaction with the environment. This is done by reflecting the force at the slave site to the user. This force feedback triggers the activation of the muscles. By the use of EMG measurements, the joint impedances can be determined. Together with the human arm posture, these joint impedances will be used to determine the endpoint impedance of the human arm. A dynamic model of the robot arm will be used to determine a control command that allows mimicking the generated endpoint impedance profile towards the robot site.

Using the described regulated tele-impedance control the robot arm should feel more like a natural extension of the human arm. This will give a more intuitive representation once the robot arm is in contact with an unpredictable environment. Think of underwater welding, for example. This increase in transparency is likely to increase the work effort.

1.1 Research goal

This research aims to develop a controller that can increase transparency within telemanipulation control by using human arm endpoint impedance generated from EMG data and the posture of the human arm. The goal is reached once the user can change the endpoint impedance of the robot arm to allow for more efficient/intuitive task handling.

To reach the goal a haptic device (Virtuose 6D [5]) and two Myo Gesture Control Armbands [18] are used at the control site. To execute the task a Panda (by Franka Emika) robot arm at the remote site is used. The position of the haptic device and the impedance profile from the human arm is translated to the robot arm at the remote site, where the task will be executed.

Based on the goal of this thesis, the research question is:

- **To what extent will the endpoint impedance of the robot arm, based on activation of the human arm muscles and posture of the human arm, increase the performance and intuitiveness of telemanipulation task execution in an unpredictable environment compared to an impedance profile based on co-contraction of a single muscle pair?**

1.2 Overview

After this brief introduction of the research subject, the document is outlined as follows:

- Chapter 2: Background.
  This chapter gives some background information that allows the reader to understand how some main theory used in this research is determined.

- Chapter 3: Analysis.
  This chapter elaborates on how the research may be realised based on previous research.
- Chapter 4: Design.
  In this chapter, the analysis is worked out more thoroughly, and a design of the controller is realised.
- Chapter 5: Experiments.
  This chapter explains the experiments that are conducted during a user study.
- Chapter 6: Results.
  In this chapter, the results of the user study and the performance of the controller are discussed.
- Chapter 7: Conclusions.
  This chapter concludes the research based on the user study results and the controller's performance.
- Chapter 8: Recommendations.
  In this chapter, recommendations are made for possible further research regarding the topic of this thesis.

2 Background

This chapter contains some general information that is needed to conduct the research. Within this background information, some examples are used, which are based on the human arm. These examples serve for a better understanding of the theory.

2.1 Muscle Jacobian

The muscle Jacobian contains the moment arms that translates the joint velocities to muscle stretch velocity. For multiple muscles and multiple joints the muscle Jacobian results in equation 2.1, [19]. In this equation $\mathbf{J_M}(\mathbf{q}) \in \mathbb{R}^{N \times Q}$, with N the number of muscles and Q the number of joints. In the equation, λ_i and q_i represent the muscle length and joint angle, respectively. The subscript i specifies the muscle and joint under investigation.

$$\mathbf{J_M}(\mathbf{q}) = \begin{bmatrix} \frac{\partial\lambda_1(q)}{\partial q_1} & \frac{\partial\lambda_1(q)}{\partial q_2} & \cdots & \frac{\partial\lambda_1(q)}{\partial q_7} \\ \frac{\partial\lambda_2(q)}{\partial q_1} & \frac{\partial\lambda_2(q)}{\partial q_2} & \cdots & \frac{\partial\lambda_2(q)}{\partial q_7} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial\lambda_N(q)}{\partial q_1} & \frac{\partial\lambda_N(q)}{\partial q_2} & \cdots & \frac{\partial\lambda_N(q)}{\partial q_7} \end{bmatrix} \tag{2.1}$$

As an example, the muscle Jacobian is determined for an elbow joint in a 2D case. Using a simplified model of the human arm (see figure 2.1), the muscle stretch velocity for the flexor muscle and extensor muscle can be determined.

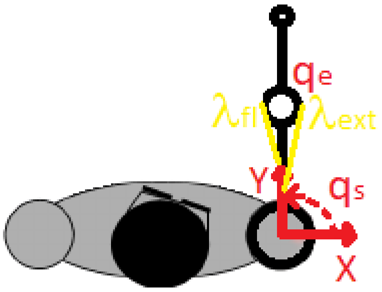


Figure 2.1: Top view of an animated human arm in 2D, with q_s and q_e the shoulder and elbow angle, respectively.

It is assumed that the arm configuration from figure 2.1 is the initial configuration ($q_s = 0$ and $q_e = 0$). The flexor muscle and extensor muscle have a length belonging to this configuration. These are λ_{fl} and λ_{ext} respectively. If a counterclockwise rotation of the endpoint with respect to the elbow joint happens, the extensor muscle will revolve around the elbow joint and therefore stretch. Whereas the flexor muscle will release from the joint and therefore contract. The quantity of stretch/contraction is dependent on the length of the arc [rad] that the muscle revolves around. The length of the arc can be described as $R_i \cdot q_i$, where R_i is the radius of the joint and q_i the joint angle. In this notation, the subscript i specifies the joint under investigation. This gives rise to the following expression for the muscle lengths:

$$\lambda_{ext} = R_e \cdot q_e + \lambda_{ext,0} \tag{2.2}$$

$$\lambda_{fl} = -R_e \cdot q_e + \lambda_{fl,0} \tag{2.3}$$

In the above equations $\lambda_{ext,0}$ and $\lambda_{fl,0}$ are the initial muscle lengths.

The muscle stretch velocity now follows by taking the time derivative of the muscle length. This gives:

$$\frac{d}{dt}\boldsymbol{\lambda} = \frac{\partial \boldsymbol{\lambda}}{\partial \mathbf{q}} \cdot \frac{\partial \mathbf{q}}{\partial t} \tag{2.4}$$

$$\frac{d}{dt}\begin{bmatrix}\lambda_{ext}\\ \lambda_{fl}\end{bmatrix} = \begin{bmatrix}\frac{\partial \lambda_{ext}}{\partial q_s} & \frac{\partial \lambda_{ext}}{\partial q_e}\\ \frac{\partial \lambda_{fl}}{\partial q_s} & \frac{\partial \lambda_{fl}}{\partial q_e}\end{bmatrix} \cdot \begin{bmatrix}\dot{q}_s\\ \dot{q}_e\end{bmatrix} \tag{2.5}$$

$$= \begin{bmatrix}0 & R_e\\ 0 & -R_e\end{bmatrix} \cdot \begin{bmatrix}\dot{q}_s\\ \dot{q}_e\end{bmatrix} \tag{2.6}$$

From 2.5 and 2.6 the muscle Jacobian for the elbow follows as:

$$\mathbf{J_M}(\mathbf{q}) = \begin{bmatrix}\frac{\partial \lambda_{ext}}{\partial q_s} & \frac{\partial \lambda_{ext}}{\partial q_e}\\ \frac{\partial \lambda_{fl}}{\partial q_s} & \frac{\partial \lambda_{fl}}{\partial q_e}\end{bmatrix} \tag{2.7}$$

$$= \begin{bmatrix}0 & R_e\\ 0 & -R_e\end{bmatrix} \tag{2.8}$$

This muscle Jacobian is a constant matrix and only dependent on the joint radius. This only holds if the muscle is rotating around the joint, as is the case in this simple example. However, the centre of rotation is hard to determine when a muscle spans several joints.

2.2 Human arm Jacobian

In robotics, the Jacobian describes the relationship between the joint angular velocities and the end-effector's linear and angular velocities. In inverse kinematics, this can be used to determine the specific joint angles belonging to the desired end-effector position. Hence, the desired path can be followed by the end-effector of the robot. There are different approaches to obtain the Jacobian, namely the analytical and a geometric approach. For the analytical approach, trigonometry is used to determine the end-effector position. Taking the partial derivative of this end-effector position with respect to the different joints results in the analytical Jacobian. However, this analytical approach becomes very hard to perform once the robotic configuration becomes more complex. Therefore, the geometric method for determining the Jacobian can be used, which is more intuitive.

2.2.1 Analytical Jacobian

To extend the example from section 2.1 the manipulator Jacobian for the human arm is determined, analytically, for a 2D case as well. This can be done more intuitively by changing the posture of figure 2.1 slightly. This results in figure 2.2, where L_{ua} and L_{la} describe the length of the upper arm and lower arm, respectively.

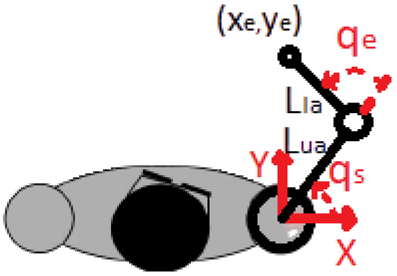


Figure 2.2: The top view of an animated human arm in 2D, with q_s and q_e the shoulder and elbow angle, respectively. The endpoint is defined by (x_e, y_e), and the length of the upper arm and lower arm with L_{ua} and L_{la}, respectively.

Doing some basic trigonometry, based on the above figure, results in the following expression for the endpoint position in 2D:

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \underbrace{\begin{bmatrix} L_{ua}\cos(q_s) + L_{la}\cos(q_s + q_e) \\ L_{ua}\sin(q_s) + L_{la}\sin(q_s + q_e) \end{bmatrix}}_{\mathbf{f}(\mathbf{q})} \tag{2.9}$$

From [20] it follows that the analytical Jacobian can be found by the partial derivative, as in 2.10.

$$\mathbf{J_H}(\mathbf{q}) = \frac{\partial \mathbf{f}(\mathbf{q})}{\partial \mathbf{q}} \tag{2.10}$$

From 2.9 and 2.10 the expression for the analytical Jacobian then follows as:

$$\mathbf{J_H}(\mathbf{q}) = \begin{bmatrix} \frac{\delta x_e}{\delta q_s} & \frac{\delta x_e}{\delta q_e} \\ \frac{\delta y_e}{\delta q_s} & \frac{\delta y_e}{\delta q_e} \end{bmatrix} = \begin{bmatrix} -L_{ua}\sin(q_s) - L_{la}\sin(q_s + q_e) & -L_{la}\sin(q_s + q_e) \\ L_{ua}\cos(q_s) + L_{la}\cos(q_s + q_e) & L_{la}\cos(q_s + q_e) \end{bmatrix} \tag{2.11}$$

2.2.2 Geometric Jacobian

The above approach shows that the Jacobian for a 2D problem, with a low amount of links, can easily be determined by following the analytical approach. However, determining the Jacobian for a 3D problem is not that easy. Therefore, to determine the Jacobian intuitively, the geometric approach is taken.

Setting the shoulder and elbow angles ($q_{s,x}$, $q_{s,y}$, $q_{s,z}$ and $q_{e,y}$) of the human arm to zero gives rise to the initial configuration as in figure 2.3.

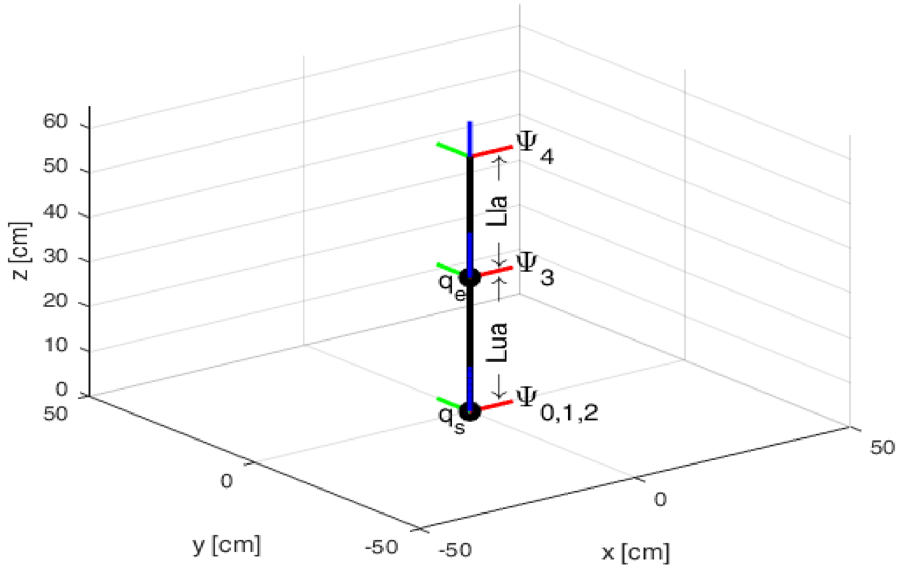


Figure 2.3: Initial configuration of the human arm in 3D, where $q_{s,x}$, $q_{s,y}$, $q_{s,z}$ and $q_{e,y}$ are set to 0 rad. The different frames used are represented by Ψ_i. Frame 0 (Ψ_0) is the reference frame. Frame 1 describes a rotation around the $x-$axis with respect to Ψ_0. Frame 2 (Ψ_2) describes a rotation around the $y-$axis with respect to Ψ_1. Frame 3 (Ψ_3) describes a rotation and translation around the $z-$axis with respect to Ψ_2. Frame 4 (Ψ_4) describes a rotation around the $y-$axis and a translation round the $z-$axis with respect to Ψ_3.

In figure 2.3 the 3 degrees of freedom (DoF) shoulder ball joint is represented with q_s and the 1 DoF elbow joint with q_e. The angles around which a rotation takes place follow from these representations as $q_{s,x}$, $q_{s,y}$, $q_{s,z}$ and $q_{e,y}$. The frames to describe the different transformations and twists are numbered as Ψ_0 till Ψ_4. Furthermore, the upper arm length and lower arm length are defined by L_{ua} and L_{la}, respectively. The upper and lower arm lengths are based on the values from [21], where L_{ua} has approximately a length of 0.3 m and L_{la} a length of 0.27 m.

The frame of interest can be expressed in the inertial reference frame (Ψ_0) by the use of Brockett's product of exponentials formula [22], which gives rise to the homogeneous transformation matrix $\mathbf{H_i^0}(\mathbf{q})$, see equation 2.12.

$$\mathbf{H_i^0}(\mathbf{q}) = e^{\tilde{\widehat{\mathbf{T}}}_\mathbf{1}^{\mathbf{0,0}} q_1} e^{\tilde{\widehat{\mathbf{T}}}_\mathbf{2}^{\mathbf{0,1}} q_2} \dots e^{\tilde{\widehat{\mathbf{T}}}_\mathbf{i}^{\mathbf{0,(i-1)}} q_i} \mathbf{H_i^0}(\mathbf{0}) \tag{2.12}$$

Where,

- q_i specifies the angle around which the transformation takes place
- $\tilde{\widehat{\mathbf{T}}}_\mathbf{i}^{\mathbf{0,(i-1)}}$ is the skew symmetric unit twist of Ψ_i with respect to frame $\Psi_{(i-1)}$, expressed in Ψ_0
- $\mathbf{H_i^0}(\mathbf{0})$ is the transformation matrix which expresses Ψ_i in Ψ_0 for the initial configuration

The unit twist follows from [23] as:

$$\widehat{\mathbf{T}}_\mathbf{i}^{\mathbf{0,(i-1)}} = \begin{bmatrix} \omega_i \\ v_i \end{bmatrix} = \begin{bmatrix} \omega_i \\ -\omega_i \times Q_i \end{bmatrix} \in \mathbb{R}^6 \tag{2.13}$$

The rotational velocity is defined by ω_i, and the translational velocity is defined by v_i. The translational velocity follows from the cross product of the rotational velocity and the position of the rotation axis (Q_i). The matrix representation of this twist can be defined by 2.14, [23], where $\tilde{\omega_i}$ is a skew-symmetric representation of the rotational velocity components.

$$\tilde{\hat{\mathbf{T}}}_{\mathbf{i}}^{\mathbf{0,(i-1)}} = \begin{bmatrix} \tilde{\omega}_i & v_i \\ 0^{1x3} & 0 \end{bmatrix} \tag{2.14}$$

Following from [24], the Rodrigues' formula 2.15 can be used to solve the matrix exponential.

$$e^{\tilde{\hat{\mathbf{T}}}_{\mathbf{i}}^{\mathbf{0,(i-1)}} q_i} = I + \tilde{\hat{\mathbf{T}}}_{\mathbf{i}}^{\mathbf{0,(i-1)}} \sin(q_i) + (\tilde{\hat{\mathbf{T}}}_{\mathbf{i}}^{\mathbf{0,(i-1)}})^2 (1 - \cos(q_i)) \tag{2.15}$$

Endpoint position in space

According to Brockett's product of exponentials formula the final frame (Ψ_4) can be expressed with respect to the inertial reference frame (Ψ_0) as:

$$\mathbf{H_4^0}(\mathbf{q}) = e^{\tilde{\hat{\mathbf{T}}}_{\mathbf{1}}^{\mathbf{0,0}} q_{s,x}} e^{\tilde{\hat{\mathbf{T}}}_{\mathbf{2}}^{\mathbf{0,1}} q_{s,y}} e^{\tilde{\hat{\mathbf{T}}}_{\mathbf{3}}^{\mathbf{0,2}} q_{s,z}} e^{\tilde{\hat{\mathbf{T}}}_{\mathbf{4}}^{\mathbf{0,3}} q_{e,y}} \mathbf{H_4^0}(\mathbf{0}) \tag{2.16}$$

This transformation matrix $\mathbf{H_4^0}(\mathbf{q})$ contains the endpoint position of the human arm in space, which is dependent on the shoulder 3 DoF angles and elbow 1 DoF angle.

By observing the initial configuration in figure 2.3 and using the expressions from 2.13 and 2.14 the following unit twist matrices are found:

$$\tilde{\hat{\mathbf{T}}}_{\mathbf{1}}^{\mathbf{0,0}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.17}$$

$$\tilde{\hat{\mathbf{T}}}_{\mathbf{2}}^{\mathbf{0,1}} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.18}$$

$$\tilde{\hat{\mathbf{T}}}_{\mathbf{3}}^{\mathbf{0,2}} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.19}$$

$$\tilde{\hat{\mathbf{T}}}_{\mathbf{4}}^{\mathbf{0,3}} = \begin{bmatrix} 0 & 0 & 1 & -L_{ua} \\ 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.20}$$

The transformation matrix $\mathbf{H_4^0}(\mathbf{0})$ follows from figure 2.3 as:

$$\mathbf{H_4^0}(\mathbf{0}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_{ua} + L_{la} \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.21}$$

Substitution of the skew symmetric unit twist matrices, together with equation 2.15 and the initial configuration matrix $\mathbf{H}_4^0(\mathbf{0})$ into 2.16 yields the following expression for the endpoint transformation matrix $\mathbf{H}_4^0(\mathbf{q})$:

$$\mathbf{H}_4^0(\mathbf{q}) = \begin{bmatrix} \mathbf{H}_{4(1)}^0(\mathbf{q}) & \mathbf{H}_{4(2)}^0(\mathbf{q}) & \mathbf{H}_{4(3)}^0(\mathbf{q}) & \mathbf{H}_{4(4)}^0(\mathbf{q}) \end{bmatrix} \tag{2.22}$$

To minimise the notation space several adaptations are made according to table 2.1.

Table 2.1: Clarification of the notation used in several matrices.

Definition:	**Conventional notation:**	**Notation in matrix as:**
Cosine of the shoulder angle with respect to the x-axis	$\cos(q_{s,x})$	$\mathrm{c}_{s,x}$
Cosine of the shoulder angle with respect to the y-axis	$\cos(q_{s,y})$	$\mathrm{c}_{s,y}$
Cosine of the shoulder angle with respect to the z-axis	$\cos(q_{s,z})$	$\mathrm{c}_{s,z}$
Cosine of the elbow angle with respect to the y-axis	$\cos(q_{e,y})$	$\mathrm{c}_{e,y}$
Sine of the shoulder angle with respect to the x-axis	$\sin(q_{s,x})$	$\mathrm{s}_{s,x}$
Sine of the shoulder angle with respect to the y-axis	$\sin(q_{s,y})$	$\mathrm{s}_{s,y}$
Sine of the shoulder angle with respect to the z-axis	$\sin(q_{s,z})$	$\mathrm{s}_{s,z}$
Sine of the elbow angle with respect to the y-axis	$\sin(q_{e,y})$	$\mathrm{s}_{e,y}$

Applying this the the transformation matrix $\mathbf{H}_4^0(\mathbf{q})$ results in:

$$\mathbf{H}_{4(1)}^0(\mathbf{q}) = \begin{bmatrix} \mathrm{c}_{s,y}\cdot\mathrm{c}_{s,z}\cdot\mathrm{c}_{e,y} - \mathrm{s}_{s,y}\cdot\mathrm{s}_{e,y} \\ \mathrm{c}_{e,y}\cdot(\mathrm{c}_{s,x}\cdot\mathrm{s}_{s,z} + \mathrm{c}_{s,z}\cdot\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,y}) + \mathrm{c}_{s,y}\cdot\mathrm{s}_{s,x}\cdot\mathrm{s}_{e,y} \\ \mathrm{c}_{e,y}\cdot(\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,z} - \mathrm{c}_{s,x}\cdot\mathrm{c}_{s,z}\cdot\mathrm{s}_{s,y}) - \mathrm{c}_{s,x}\cdot\mathrm{c}_{s,y}\cdot\mathrm{s}_{e,y} \\ 0 \end{bmatrix} \tag{2.23}$$

$$\mathbf{H}_{4(2)}^0(\mathbf{q}) = \begin{bmatrix} -\mathrm{c}_{s,y}\cdot\mathrm{s}_{s,z} \\ \mathrm{c}_{s,x}\cdot\mathrm{c}_{s,z} - \mathrm{s}_{s,x}\cdot\mathrm{s}_{s,y}\cdot\mathrm{s}_{s,z} \\ \mathrm{c}_{s,z}\cdot\mathrm{s}_{s,x} + \mathrm{c}_{s,x}\cdot\mathrm{s}_{s,y}\cdot\mathrm{s}_{s,z} \\ 0 \end{bmatrix} \tag{2.24}$$

$$\mathbf{H}_{4(3)}^0(\mathbf{q}) = \begin{bmatrix} \mathrm{s}_{s,y}\cdot\mathrm{c}_{e,y} + \mathrm{c}_{s,y}\cdot\mathrm{c}_{s,z}\cdot\mathrm{s}_{e,y} \\ \mathrm{s}_{e,y}\cdot(\mathrm{c}_{s,x}\cdot\mathrm{s}_{s,z} + \mathrm{c}_{s,z}\cdot\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,y}) - \mathrm{c}_{s,y}\cdot\mathrm{s}_{s,x}\cdot\mathrm{c}_{e,y} \\ \mathrm{s}_{e,y}\cdot(\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,z} - \mathrm{c}_{s,x}\cdot\mathrm{c}_{s,z}\cdot\mathrm{s}_{s,y}) + \mathrm{c}_{s,x}\cdot\mathrm{c}_{s,y}\cdot\mathrm{c}_{e,y} \\ 0 \end{bmatrix} \tag{2.25}$$

$$\mathbf{H}_{4(4)}^0(\mathbf{q}) = \begin{bmatrix} (L_{la}+L_{ua})\cdot(\mathrm{s}_{s,y}\cdot\mathrm{c}_{e,y}+\mathrm{c}_{s,y}\cdot\mathrm{c}_{s,z}\cdot\mathrm{s}_{e,y}) - L_{ua}\cdot\mathrm{s}_{s,y}\cdot(\mathrm{c}_{e,y}-1) - L_{ua}\cdot\mathrm{c}_{s,y}\cdot\mathrm{c}_{s,z}\cdot\mathrm{s}_{e,y} \\ (L_{la}+L_{ua})\cdot(\mathrm{s}_{e,y}\cdot(\mathrm{c}_{s,x}\cdot\mathrm{s}_{s,z}+\mathrm{c}_{s,z}\cdot\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,y}) - \mathrm{c}_{s,y}\cdot\mathrm{s}_{s,x}\cdot\mathrm{c}_{e,y}) - L_{ua}\cdot\mathrm{s}_{e,y}\cdot(\mathrm{c}_{s,x}\cdot\mathrm{s}_{s,z}+\mathrm{c}_{s,z}\cdot\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,y}) + L_{ua}\cdot\mathrm{c}_{s,y}\cdot\mathrm{s}_{s,x}\cdot(\mathrm{c}_{e,y}-1) \\ (L_{la}+L_{ua})\cdot(\mathrm{s}_{e,y}\cdot(\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,z}-\mathrm{c}_{s,x}\cdot\mathrm{c}_{s,z}\cdot\mathrm{s}_{s,y}) + \mathrm{c}_{s,x}\cdot\mathrm{c}_{s,y}\cdot\mathrm{c}_{e,y}) - L_{ua}\cdot\mathrm{s}_{e,y}\cdot(\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,z}-\mathrm{c}_{s,x}\cdot\mathrm{c}_{s,z}\cdot\mathrm{s}_{s,y}) - L_{ua}\cdot\mathrm{c}_{s,x}\cdot\mathrm{c}_{s,y}\cdot(\mathrm{c}_{e,y}-1) \\ 1 \end{bmatrix} \tag{2.26}$$

Spatial manipulator Jacobian

According to [25] the spatial manipulator Jacobian can be determined as:

$$\mathbf{J_H}(\mathbf{q}) = \begin{bmatrix} \mathbf{J_{H,1}} & \mathbf{J_{H,2}} & \mathbf{J_{H,3}} & \mathbf{J_{H,4}} \end{bmatrix} \tag{2.27}$$

The different columns follow from:

$$\mathbf{J_{H,i}} = \mathrm{Ad}_{\mathbf{H^0_{i-1}(q)}} \widehat{\mathrm{T}}_{\mathbf{i}}^{\mathbf{(i-1),(i-1)}} \tag{2.28}$$

The adjoint of the transformation matrix $\mathbf{H^j_i(q)}$ follows from [23] as:

$$\mathrm{Ad}_{\mathbf{H^j_i(q)}} = \begin{bmatrix} R_i^j & 0 \\ \tilde{p}_i^j R_i^j & R_i^j \end{bmatrix} \in \mathbb{R}^{6\times 6} \tag{2.29}$$

In this matrix R_i^j represents the rotation matrix of Ψ_i with respect to Ψ_j and $\tilde{p}_i^j$ the skew symmetric expression for the position of Ψ_i with respect to Ψ_j.

The transformation matrices $\mathbf{H^0_3(q)}$, $\mathbf{H^0_2(q)}$, $\mathbf{H^0_1(q)}$ and $\mathbf{H^0_0(q)}$ can be determined in a similar way as is done above in 2.2.2. The different adjoint matrices then simply follow by substitution of the transformation matrices components and endpoint positions in 2.29.

The unit twists $\widehat{\mathrm{T}}_{\mathbf{i}}^{\mathbf{(i-1),(i-1)}}$ can be determined by observing the initial configuration of figure 2.3. These unit twists are expressed in their previous frame. Since there are only rotational joints, the twists should only contain elements for rotation. Hence, the unit twists are:

$$\widehat{\mathrm{T}}_{\mathbf{1}}^{\mathbf{0,0}} = \begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix} \qquad \widehat{\mathrm{T}}_{\mathbf{2}}^{\mathbf{1,1}} = \begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix} \qquad \widehat{\mathrm{T}}_{\mathbf{3}}^{\mathbf{2,2}} = \begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix} \qquad \widehat{\mathrm{T}}_{\mathbf{4}}^{\mathbf{3,3}} = \begin{bmatrix}0\\1\\0\\0\\0\\0\end{bmatrix}$$

Substitution of the different unit twists and different adjoint matrices into 2.28 yields the spatial manipulator Jacobian of the human arm as:

$$\mathbf{J_H(q)} = \left[\underbrace{\begin{matrix}1\\0\\0\\0\\0\\0\end{matrix}}_{\mathbf{J_{H,1}}} \quad \underbrace{\begin{matrix}0\\ \mathrm{c}_{s,x}\\ \mathrm{s}_{s,x}\\0\\0\\0\end{matrix}}_{\mathbf{J_{H,2}}} \quad \underbrace{\begin{matrix}\mathrm{s}_{s,y}\\ -\mathrm{c}_{s,y}\cdot\mathrm{s}_{s,x}\\ \mathrm{c}_{s,x}\cdot\mathrm{c}_{s,y}\\0\\0\\0\end{matrix}}_{\mathbf{J_{H,3}}} \quad \underbrace{\begin{matrix}-\mathrm{c}_{s,y}\cdot\mathrm{s}_{s,z}\\ \mathrm{c}_{s,x}\cdot\mathrm{c}_{s,z}-\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,y}\cdot\mathrm{s}_{s,z}\\ \mathrm{c}_{s,z}\cdot\mathrm{s}_{s,x}+\mathrm{c}_{s,x}\cdot\mathrm{s}_{s,y}\cdot\mathrm{s}_{s,z}\\ -L_{ua}\cdot\mathrm{c}_{s,x}\cdot\mathrm{c}_{s,y}\cdot(\mathrm{c}_{s,x}\cdot\mathrm{c}_{s,z}-\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,y}\cdot\mathrm{s}_{s,z})-L_{ua}\cdot\mathrm{c}_{s,y}\cdot\mathrm{s}_{s,x}\cdot(\mathrm{c}_{s,z}\cdot\mathrm{s}_{s,x}+\mathrm{c}_{s,x}\cdot\mathrm{s}_{s,y}\cdot\mathrm{s}_{s,z})\\ -L_{ua}\cdot\mathrm{c}_{s,x}\cdot\mathrm{s}_{s,z}\cdot\mathrm{c}_{s,y}^2-L_{ua}\cdot\mathrm{s}_{s,y}\cdot(\mathrm{c}_{s,z}\cdot\mathrm{s}_{s,x}+\mathrm{c}_{s,x}\cdot\mathrm{s}_{s,y}\cdot\mathrm{s}_{s,z})\\ L_{ua}\cdot\mathrm{s}_{s,y}\cdot(\mathrm{c}_{s,x}\cdot\mathrm{c}_{s,z}-\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,y}\cdot\mathrm{s}_{s,z})-L_{ua}\cdot\mathrm{c}_{s,y}^2\cdot\mathrm{s}_{s,x}\cdot\mathrm{s}_{s,z}\end{matrix}}_{\mathbf{J_{H,4}}} \right] \tag{2.30}$$

Inverse kinematics

To conclude whether the determined manipulator Jacobian is correct, an approach is taken which determines the desired joint angles based on the manipulator Jacobian and the desired path that needs to be followed by the endpoint of the arm.

For the arm to follow a pattern, a set of setpoints need to be followed. This can be achieved by the use of inverse kinematics, where $\mathbf{\dot{q}} = \mathbf{J_H^{-1}(q)T_4^{0,0}}$. However, the matrix $\mathbf{J_H(q)}$ is not square and hence not non-singular. Furthermore, the endpoint twist has only a relevant translational velocity.

To overcome this problem the approach from [26] is taken, where the manipulator Jacobian $\mathbf{J_H}(\mathbf{q})$ can be transformed to a pseudo inverse that only defines a mapping from the end-effector veloctiy $\dot{\mathbf{p}}_{\mathbf{ee}}$ to $\dot{\mathbf{q}}$. This results in the following expression:

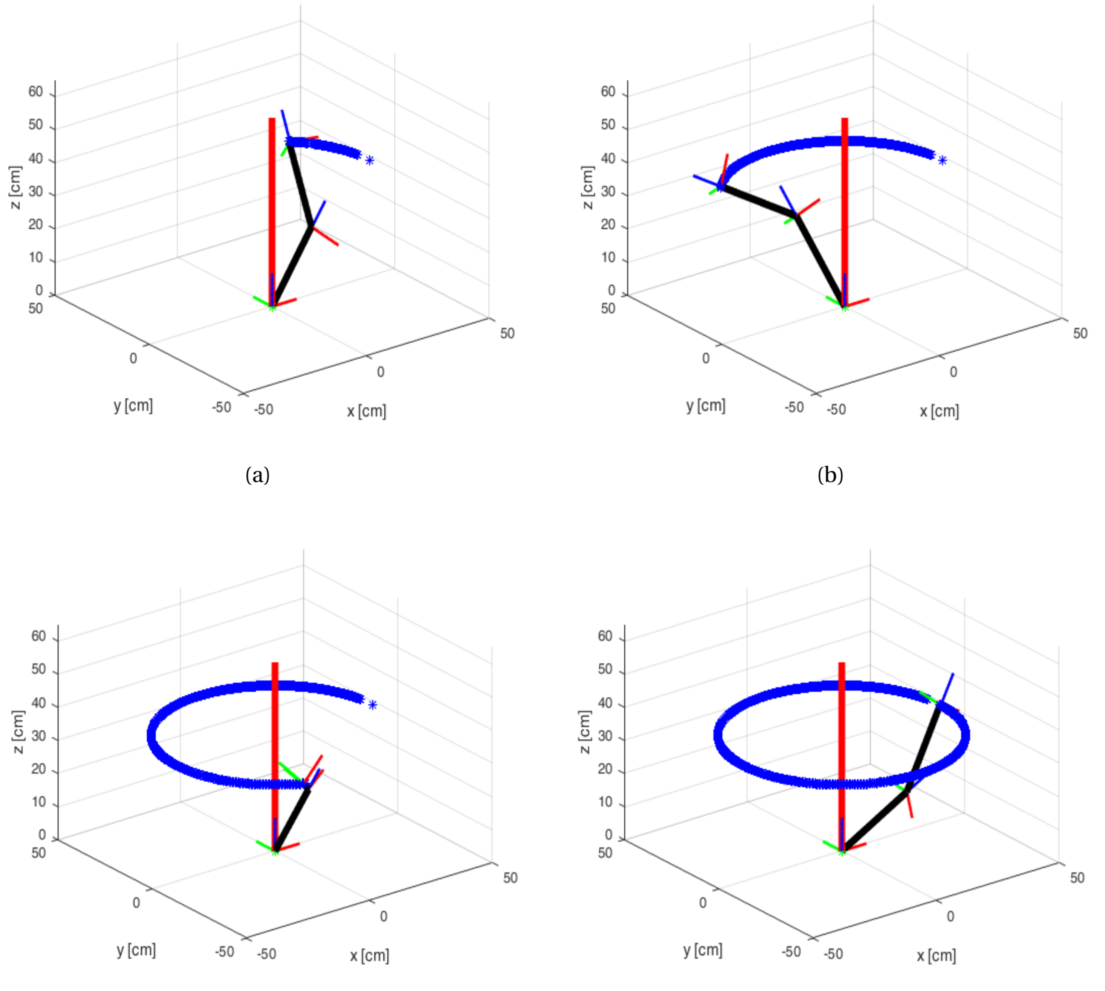
$$\dot{\mathbf{q}} = \mathbf{J}^{\dagger}_{\mathbf{H,v}}(\mathbf{q}) \cdot \dot{\mathbf{p}}_{\mathbf{ee}} \tag{2.31}$$

A way to determine the endpoint velocity is by the difference in setpoint position and the actual position of the endpoint, and the time delay between the devices controlling the setpoint and endpoint positions. Hence,

$$\dot{\mathbf{p}}_{\mathbf{ee}} = \frac{(\mathbf{p}_{\mathbf{sp}} - \mathbf{p}_{\mathbf{ee}})}{\tau_{DT}} \tag{2.32}$$

The endpoint position $\mathbf{p}_{\mathbf{ee}}$ follows from the transformation matrix $\mathbf{H}^{\mathbf{0}}_{\mathbf{4}}(\mathbf{q})$. The setpoint follows from the discretisation of the desired path. If the initial manipulation Jacobian $\mathbf{J_H}(\mathbf{q})$ is correct, the to be set velocity on the arm ($\dot{\mathbf{q}}$) should cause the arm to follow the desired path.

Setting a circle with a height in the z-direction of 0.35 m and a radius in the x,y-direction of 0.40 m, as a specified path, indeed causes the arm model to follow. This can be seen clearly from the Matlab simulation in figure 2.4. Therefore, it can be concluded that the determined manipulator Jacobian of the human arm, $\mathbf{J_H}(\mathbf{q})$, should be correct. In the figure, the red line represents the arm in its initial configuration. To observe the rotation of the arm, the inertial reference frame (Ψ_0), the frame at the elbow joint (Ψ_3), and the frame at the endpoint (Ψ_4) are drawn. The specified path that is followed by the arm is drawn in blue.



(a)

(b)

(c)

(d)

Figure 2.4: In figures *a* till *d*, it is illustrated that the simplified human arm model follows the set circular pattern by changing its shoulder and elbow angles.

3 Analysis

As discussed in section 1, the goal is to translate the endpoint impedance from the human arm towards the robot arm.
The main idea originates from the work of [17], where an impedance profile of the human arm together with a corresponding endpoint position is sent towards the robot arm in real-time. The stiffness profile is based on agonist-antagonist muscle co-activations. This co-activation allows for an increase/decrease in joint and endpoint impedance of the human arm without any force fluctuations at the endpoint. The measured muscular activities (EMG) signals are decomposed into force related and co-activation parts. This co-activation part describes the endpoint impedance caused by muscle co-activation. The total endpoint impedance can then be found by including the intrinsic endpoint stiffness, which is dependent on the posture of the human arm.

A few downsides to the method of [17] are described below:

1. A downside is that the intrinsic stiffness is not scaled with the actual arm posture. They did not implement this since the task performed was accomplished in the vicinity of the isometric posture.

2. Furthermore, the tracking of the human arm endpoint (wrist), shoulder and elbow positions are done by an Optitrack system. This method requires the master to always be within a space that is equipped with cameras.

3. At last, the sEMG measurements are done by the use of separate surface EMG sensors. This does not only require time to install but might also cause limitation in moving space or even measurement uncertainties once the cables are yanked.

To overcome the downsides of the aforementioned method adaptations are made. These adaptations are elaborated on throughout this analysis section.

3.1 Endpoint impedance estimation of the human arm

Regulating the impedance at the slave site allows for increased stability and accuracy, making the overall system more transparent if the impedance at the master site can be tuned intuitively.

The study in [27] shows that human motor skills are used by the Central Nervous System (CNS) to activate individual muscles for task-oriented force generation. According to [28] muscle forces are generated based on activation dynamics and contraction dynamics of the muscle tissue. Furthermore, [27] says that the CNS also uses muscle activation to realise an increase in impedance without changing the endpoint force. Changes in endpoint force are dependent on the length change of the muscles, hence on the contraction dynamics. Therefore, muscular activation can be associated with isometric muscle force [28]. The isometric muscle force is a force excited by the muscle without changing the length of the muscle. The measured activation of the muscles can be related to the joint impedances once a disturbance to the arm endpoint is applied. From previous research, e.g. [29], it is found that muscles have spring-like properties. The key in finding the endpoint impedance of the human arm, therefore, lies in finding the human arm's joint stiffness by using surface EMG readings. Once these joint stiffnesses are found, a kinematic model of the human arm can be used to describe a human arm Jacobian to transform the joint stiffnesses to a form that describes the stiffness of the human arm at its endpoint in a 3D environment. This can be realised by finding a solution, in real-time, for the following equation,[30]:

$$\mathbf{K_e} = \mathbf{J_H}^{-\mathrm{T}}(\mathbf{q})\mathbf{K_J}\mathbf{J_H}^{-1}(\mathbf{q}) \tag{3.1}$$

Where $\mathbf{J_H}$ is the human arm Jacobian matrix and $\mathbf{K_J}$ a matrix that contains the joint stiffnesses.

Applying this methodology covers almost all the capabilities of changing the endpoint impedance of the human arm. According to [31] the human is capable of changing its endpoint impedance in several ways. Namely: by changing its arm configuration, applying a force or co-contraction of the muscle pairs. The endpoint impedance that follows once the operator applies a force is already incorporated in the joint stiffness and human arm Jacobian. This becomes clear once the human arm applies an external force with its endpoint. Applying an external force causes a change in the activation of the human arm muscles, and therefore in the joint stiffnesses. If there is no co-activation (the agonist muscle increases while the antagonist muscle stays the same), the human arm will move, resulting in a position deviation in the endpoint. Hence, the human arm Jacobian will change as well.

3.1.1 Joint stiffness based on surface EMG readings

From [32] it follows that the joint stiffness is dependent on the muscles crossing that joint. An angle change with respect to that joint will cause muscles crossing the joint to be stretched or shortened. Furthermore, the torque applied by the muscle is dependent on the muscle force and moment arm. This follows from the equation for joint stiffness (K_J), based on a single contributing muscle. The equation can be found below, with τ the joint torque, q the joint angle, μ the muscle force, ρ the moment arm, λ the muscle length, and K_μ the muscle stiffness.

$$K_J = \frac{d\tau}{dq} = \frac{\partial\lambda}{\partial q}\cdot\frac{\partial\tau}{\partial\lambda} = \frac{\partial\lambda}{\partial q}\cdot\frac{\partial\tau}{\partial\mu}\cdot\frac{\partial\mu}{\partial\lambda} = \rho^2 K_\mu \tag{3.2}$$

The result of this equation follows from the general torque and length calculations. For this specific case they can be expressed as:

$$d\tau = d\mu\cdot\rho \qquad \text{and} \qquad d\lambda = \rho\cdot d(q) \tag{3.3}$$

Using equation 3.2 to determine the joint stiffness for different joints, based on multiple muscles and moment arms, results in equation 3.4. This joint stiffness matrix for the human arm also follows from [33].

$$\mathbf{K_J} = \mathbf{J_M}^T(\mathbf{q})\mathbf{K}_\mu\mathbf{J_M}(\mathbf{q}) \tag{3.4}$$

In equation 3.4 the muscle Jacobian ($\mathbf{J_M}(\mathbf{q})$) contains the moment arms, which are dependent on small changes in muscle length caused by small changes in the joint angle [33], see equation 3.2, and basically describes the distance of the perpendicular line of action towards the joint of interest. Combining all muscle moment arms results in a muscle Jacobian matrix as in equation 2.1.

Muscle stiffness

For the muscle stiffness holds, according to equation 3.2: $K_\mu = \frac{\partial\mu}{\partial\lambda}$.
However, once an isometric muscle force is considered, the muscle stiffness can be described

by a linear dependence on the muscle activation [28]. This results in an expression for the muscle stiffness as 3.5, [34]:

$$K_{\mu}^{i} = K_{int}^{i} + K_{\mu,max}^{i} \cdot u_i \tag{3.5}$$

Where,

- K_{μ}^{i} describes the total elasticity of the single muscle i
- K_{int}^{i} is the intrinsic muscle elasticity of muscle i. Hence, the elasticity of the muscle when there is no activation
- $K_{\mu,max}^{i}$ is the maximum muscle elasticity of muscle i
- u_i describes the muscle activation of muscle i, based on normalized EMG measurements for the muscle under investigation (i)

Determining the muscle stiffness for every contributing/measured muscle gives rise to a muscle stiffness matrix $\mathbf{K}_{\mu} \in \mathbb{R}^{N \times N}$. The diagonal matrix dimension $N \times N$ follows from the number of muscles N. Every matrix entry represents the muscle stiffness for a specific muscle.

Muscle Jacobian
The method in section 2.1 that determines the muscle Jacobian is based on the assumption that the muscle is aligned with the body parts. Furthermore, it does not allow for muscle moment arm calculations once the muscle spans several joints. Although it gives a good insight into what is going on, it is too oversimplified to be used. To get reliable results within the muscle Jacobian calculations, an OpenSim model will be used that is based on the Upper Extremity model of [35].

3.1.2 Joint stiffness to endpoint stiffness

The only remainder to obtain the human arm endpoint stiffness is the human arm Jacobian. This can easily be obtained by realising a kinematic model which includes rotations of the shoulder joint and the elbow joint. This Jacobian causes the endpoint impedance to be posture dependent once an orientation sensor obtains the specific shoulder and elbow angles.

3.2 Muscles for human impedance estimation

In order to obtain accurate joint stiffnesses regarding the shoulder and elbow, it is desired to measure all contributing muscles. However, this would be unnecessarily complex and user-unfriendly. Think, for instance, of an arm with loads of EMG sensors distributed along the shoulder, arm and forearm with their corresponding wires. This would limit the user's movements and take precious time to install, especially during operator shifts.

To avoid unnecessary hassle and limitations in operating space it would be desirable to use a Myo Gesture Control Armband developed by Thalmic Labs [18]. This device can be worn around the arm and forearm of the operator. The Myo Armband consists of eight surface EMG sensors that allow, if worn correctly, to measure eight different muscles in the arm and forearm.

The muscles that contribute the most to elbow and shoulder flexion, extension, supination and pronation are based on the muscle modelling parameters of [35]. The methodology of using the Myo Gesture Control Armband comes with the downside that some significant muscles that contribute to the shoulder can not be measured due to the limited placement of the Myo Armband. However, some muscles still contribute to the stiffness of the shoulder, which can be measured around the arm. To grasp measurable and significantly contributing muscles to the shoulder and elbow stiffnesses it is desirable to measure the muscles of the arm and forearm that are indicated in figure 3.1 and 3.3 respectively. How these muscles can effectively be measured is dependent on the Myo Armband placement. This will follow in the design section 4.2. The level (height) of these muscles follow from figure 3.2 and 3.3.

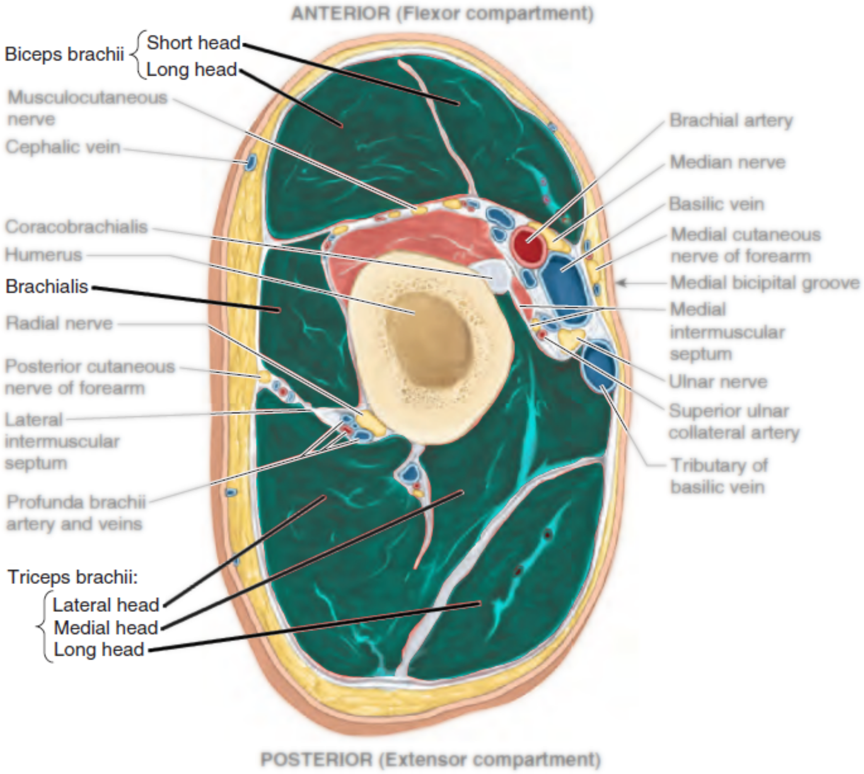

ANTERIOR (Flexor compartment)
Biceps brachii
Short head
Long head
Musculocutaneous nerve
Cephalic vein
Coracobrachialis
Humerus
Brachialis
Radial nerve
Posterior cutaneous nerve of forearm
Lateral intermuscular septum
Profunda brachii artery and veins
Triceps brachii:
Lateral head
Medial head
Long head
Brachial artery
Median nerve
Basilic vein
Medial cutaneous nerve of forearm
Medial bicipital groove
Medial intermuscular septum
Ulnar nerve
Superior ulnar collateral artery
Tributary of basilic vein
POSTERIOR (Extensor compartment)

Figure 3.1: Cross-section of the arm, with the muscles of interest highlighted (green). Edited image, original originates from [2].

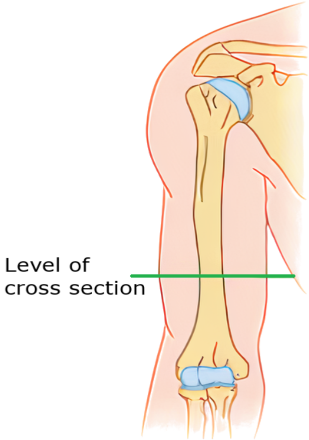


Figure 3.2: Indication of the level where the muscle locations of figure 3.1 match, from [2].

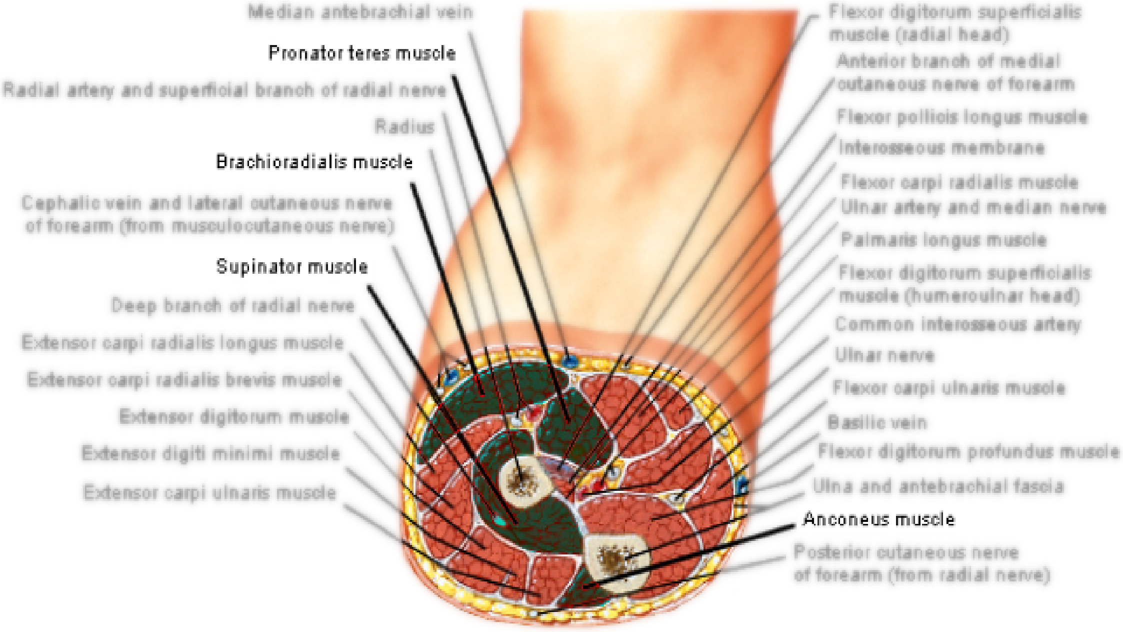

Median antebrachial vein
Pronator teres muscle
Radial artery and superficial branch of radial nerve
Radius
Brachioradialis muscle
Cephalic vein and lateral cutaneous nerve of forearm (from musculocutaneous nerve)
Supinator muscle
Deep branch of radial nerve
Extensor carpi radialis longus muscle
Extensor carpi radialis brevis muscle
Extensor digitorum muscle
Extensor digiti minimi muscle
Extensor carpi ulnaris muscle
Flexor digitorum superficialis muscle (radial head)
Anterior branch of medial cutaneous nerve of forearm
Flexor pollicis longus muscle
Interosseous membrane
Flexor carpi radialis muscle
Ulnar artery and median nerve
Palmaris longus muscle
Flexor digitorum superficialis muscle (humeroulnar head)
Common interosseous artery
Ulnar nerve
Flexor carpi ulnaris muscle
Basilic vein
Flexor digitorum profundus muscle
Ulna and antebrachial fascia
Anconeus muscle
Posterior cutaneous nerve of forearm (from radial nerve)

Figure 3.3: Cross-section of the forearm, with the muscles of interest highlighted (green). Edited image, original originates from [3].

3.3 Hardware

This section contains the required hardware in order to realise and use the desired controller.

3.3.1 Myo Armband surface EMG

The physical Myo Gesture Control Armband developed by Thalmic Labs, which will be used for measuring surface EMG signals, can be seen in figure 3.4. The device consists of eight medical grade stainless steel EMG sensors and a nine-axis Inertial Measurement Unit (IMU). This IMU contains a three-axis gyroscope, a three-axis accelerometer and a three-axis magnetometer. Furthermore, it contains a battery, and the data is transferred over Bluetooth, which avoids the unnecessary hassle of wires [18].

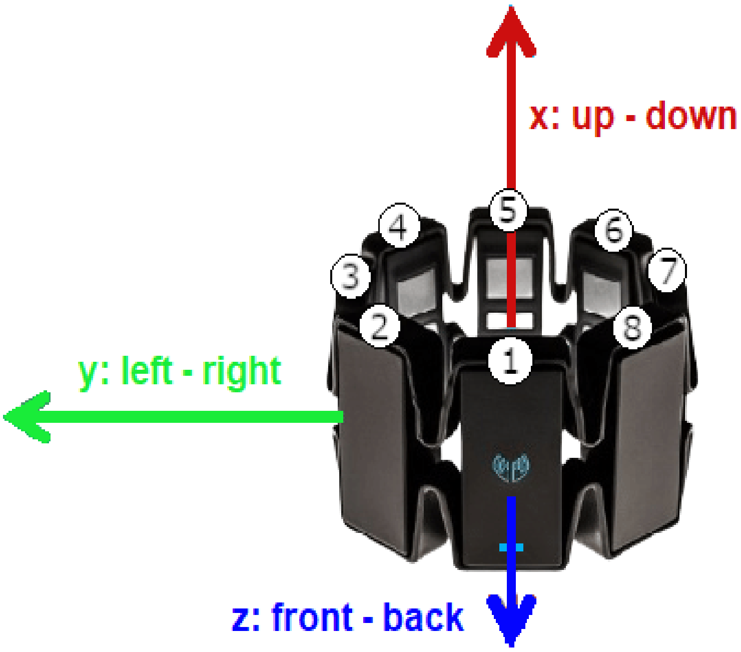


Figure 3.4: Indication of the Myo Gesture Control Armband with its IMU axes and EMG sensors numbered. Edited image, original originates from [4].

3.3.2 Haptic device

Depending on the results, this research may be used within the ANA Avatar XPRIZE competition. Therefore, the haptic device that will be used during this research will be the same as the one used to control the ANA Avatar, which is the Virtuose 6D from Haption [5] mounted in an upside-down configuration. This haptic device can deliver high force feedback in 6 DoF and has a large workspace. This large workspace allows the user to move his/her entire arm to control the haptic device. This allows for a contribution of the posture of the human arm regarding the endpoint stiffness, which will be less the case if a haptic device with a smaller workspace is used. An indication of how the device looks like can be found in figure 3.5.

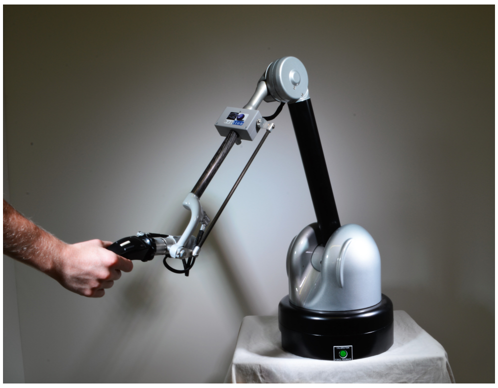


Figure 3.5: The Virtuose 6D haptic device from Haption in its typical configuration[5].

3.3.3 Robot arm

For the slave device, there are two possible options available at the RaM lab. One is the KUKA lwr 4+ and the other is the Franka Emika Panda. Both are 7 DoF devices and can therefore process the possible 6 DoF information from the haptic device. Since the Franka Emika Panda has better integration with the ROS environment, it will be chosen as the slave device.

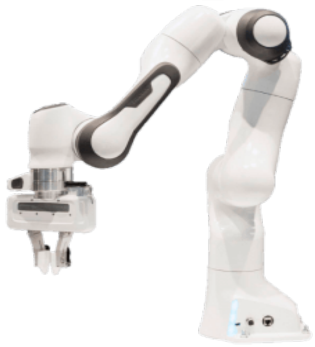


Figure 3.6: The Franka Emika Panda arm, that will be used as slave device. Image obtained from [6].

3.4 Human arm endpoint impedance implementation on a robotic arm

The dynamics of the human arm and the robotic arm differ. The endpoint stiffness of the human arm is associated with its dynamics, as follows from section 3.1. A controller is needed that can change the dynamic behaviour of the robot arm. The control command from this controller should allow the robot arm to mimic the endpoint impedance of the human arm based on the stiffness profiles.

In order to apply the determined stiffness profiles to the robotic arm, the dynamical model of the robot arm should be described first.

Assuming that all the intermediate joints between the endpoint and first angular joint are rigid, the total torque acting on this first joint can be determined using an inverted pendulum model. This results in the equation of motion as in 3.6, which also follows from [36].

$$\tau_{\mathbf{total}} = \tau_{\mathbf{acc}} + \tau_{\mathbf{vel}} + \tau_{\mathbf{g}}(\mathbf{q}) + \tau_{\mathbf{f}} - \tau_{\mathbf{ext}} \tag{3.6}$$

Where:

- $\tau_{\mathbf{total}}$ is the torque that acts on the rotating joint.
- $\tau_{\mathbf{acc}}$ is the torque due to angular acceleration ($\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}}$), with $\mathbf{M}(\mathbf{q})$ the inertia matrix.
- $\tau_{\mathbf{vel}}$ is the torque due to angular velocity $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}}$, with $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ the Coriolis/centrifugal matrix.
- $\tau_{\mathbf{g}}(\mathbf{q})$ is a vector containing the torques due to gravity and is dependent on the orientation of the arm.
- $\tau_{\mathbf{f}}$ is a vector containing the torques due to friction. In case of simulation these can be regarded as ideal (hence, 0).
- $\tau_{\mathbf{ext}}$ is the torque due to the external forces that are applied at the end-effector ($\tau_{\mathbf{ext}} = \mathbf{J_R}^{\mathrm{T}}(\mathbf{q})\mathbf{F_{ext}}$). With $\mathbf{J_R}^{\mathrm{T}}(\mathbf{q})$ the transpose of the manipulator Jacobian of the robot arm.

Therefore, with $\tau_{\mathbf{g}}(\mathbf{q}) = \mathbf{J_R}^{\mathrm{T}}(\mathbf{q})\mathbf{F_g}(\mathbf{q})$, there holds:

$$\begin{aligned}\tau_{\mathbf{total}} &= \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{J_R}^{\mathrm{T}}(\mathbf{q})\mathbf{F_g}(\mathbf{q}) - \mathbf{J_R}^{\mathrm{T}}(\mathbf{q})\mathbf{F_{ext}} \\ &= \mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{J_R}^{\mathrm{T}}(\mathbf{q})(\mathbf{F_g}(\mathbf{q}) - \mathbf{F_{ext}})\end{aligned} \tag{3.7}$$

The joint velocities/accelerations can be translated into the Cartesian domain by using the manipulator Jacobian of the robot arm. Resulting in:

$$\dot{\mathbf{x}}_{\mathbf{R}} = \mathbf{J_R}(\mathbf{q})\dot{\mathbf{q}} \rightarrow \dot{\mathbf{q}} = \mathbf{J_R}^{-1}(\mathbf{q})\dot{\mathbf{x}}_{\mathbf{R}} \tag{3.8}$$

$$\ddot{\mathbf{x}}_{\mathbf{R}} = \mathbf{J_R}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{J}}_{\mathbf{R}}(\mathbf{q})\dot{\mathbf{q}} \rightarrow \ddot{\mathbf{q}} = \mathbf{J_R}^{-1}(\mathbf{q})(\ddot{\mathbf{x}}_{\mathbf{R}} - \dot{\mathbf{J}}_{\mathbf{R}}(\mathbf{q})\dot{\mathbf{q}}) \tag{3.9}$$

Substitution of equations 3.8 and 3.9 into 3.7 yields:

$$\begin{aligned}\tau_{\mathbf{total}} =& \mathbf{M}(\mathbf{q})\mathbf{J_R}^{-1}(\mathbf{q})(\ddot{\mathbf{x}}_{\mathbf{R}} - \dot{\mathbf{J}}_{\mathbf{R}}(\mathbf{q})\mathbf{J_R}^{-1}(\mathbf{q})\dot{\mathbf{x}}_{\mathbf{R}}) \\ &+ \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\mathbf{J_R}^{-1}(\mathbf{q})\dot{\mathbf{x}}_{\mathbf{R}} + \mathbf{J_R}^{\mathrm{T}}(\mathbf{q})(\mathbf{F_g}(\mathbf{q}) - \mathbf{F_{ext}})\end{aligned} \tag{3.10}$$

Where $\tau_{\mathbf{total}} = \mathbf{J_R}^{\mathrm{T}}(\mathbf{q})\mathbf{F_{total}}$, hence:

$$\mathbf{F_{total}} = \Lambda(\mathbf{q})\ddot{\mathbf{x}}_{\mathbf{R}} + \mu(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{x}}_{\mathbf{R}} + \mathbf{F_g}(\mathbf{q}) - \mathbf{F_{ext}} \tag{3.11}$$

With:

$$\Lambda(\mathbf{q}) = \mathbf{J_R}^{-\mathrm{T}}(\mathbf{q})\mathbf{M}(\mathbf{q})\mathbf{J_R}^{-1}(\mathbf{q}) \tag{3.12}$$

$$\mu(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{J_R}^{-\mathrm{T}}(\mathbf{q})\left(\mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{M}(\mathbf{q})\mathbf{J_R}^{-1}(\mathbf{q})\dot{\mathbf{J}}_{\mathbf{R}}(\mathbf{q})\right)\mathbf{J_R}^{-1}(\mathbf{q}) \tag{3.13}$$

Control force

To realise compliant behaviour of the robot arm, it is desired to model the external forces from the environment as an impedance model, where the sensed end-effector motion of the robot arm results in a force applied to the robot arm. This can be realized by a mass-spring-damper system, based on [30] and [37]. Keeping in mind that the acceleration should be measured with respect to a fixed reference frame, this results in equation 3.14.

$$\mathbf{F_{ext}} = \Lambda_{\mathbf{d}}\ddot{\mathbf{x}}_{\mathbf{R}} + \mathbf{D_d}\Delta\dot{\mathbf{x}}_{\mathbf{R}} + \mathbf{K_d}\Delta\mathbf{x_R} \tag{3.14}$$

Where,

- $\Delta\mathbf{x_R} = \mathbf{x_R} - \mathbf{x_{R,d}}$ is the Cartesian displacement, where $\mathbf{x_R}$ is the actual position and $\mathbf{x_{R,d}}$ the desired position of the end-effector of the robot arm.
- $\Lambda_{\mathbf{d}}$ a positive definite matrix representing virtual mass. This virtual mass is the desired mass for a human arm.
- $\mathbf{D_d}$ a positive definite matrix representing virtual damping. This virtual damping is the desired damping to mimic the compliance of the human arm.
- $\mathbf{K_d}$ a positive definite matrix representing virtual stiffness. This virtual stiffness is the desired stiffness to mimic the compliance of the human arm.

By using the impedance model to describe the external forces from the environment, a control law is developed. This control law describes the force that needs to be applied by the end-effector to reach the desired compliant behaviour.

This control law can be obtained in closed loop by rewriting 3.14 into:

$$\ddot{\mathbf{x}}_{\mathbf{R}} = \Lambda_{\mathbf{d}}^{\mathbf{-1}}\left(\mathbf{F_{ext}} - \mathbf{D_d}\Delta\dot{\mathbf{x}}_{\mathbf{R}} - \mathbf{K_d}\Delta\mathbf{x_R}\right) \tag{3.15}$$

Substitution into 3.11 than yields:

$$\begin{aligned}\mathbf{F_{total}} =& \mu(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{x}}_{\mathbf{R}} + \mathbf{F_g}(\mathbf{q}) - \Lambda(\mathbf{q})\Lambda_{\mathbf{d}}^{\mathbf{-1}}\left(\mathbf{D_d}\Delta\dot{\mathbf{x}}_{\mathbf{R}} + \mathbf{K_d}\Delta\mathbf{x_R}\right) &(3.16)\\ &+ \left(\Lambda(\mathbf{q})\Lambda_{\mathbf{d}}^{\mathbf{-1}} - \mathbf{I}\right)\mathbf{F_{ext}} &(3.17)\end{aligned}$$

Force feedback is required in order to reshape the inertia of the end-effector [36]. Furthermore, the whole idea of the regulated tele-impedance controller, which is based on coupling impedance profiles to positions, is to avoid force control. Instead, it is desired to translate the human impedance to the robot arm directly. Therefore, it can be assumed that $\Lambda = \Lambda_{\mathbf{d}}$, which leads to the desired control command in equation 3.18.

$$\mathbf{F_{Cmd}} = \mathbf{F_{total}} = \mu(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{x}}_{\mathbf{R}} + \mathbf{F_g}(\mathbf{q}) - \mathbf{D_d}\Delta\dot{\mathbf{x}}_{\mathbf{R}} - \mathbf{K_d}\Delta\mathbf{x_R} \tag{3.18}$$

This command needs to be applied to the robotic arm to mimic the compliance of the human arm. Where the virtual stiffness follows from EMG data as described in section 3.1. The virtual damping can be set to the desired value that causes the robot arm to behave as a stable system (can be tweaked during testing). The virtual mass does not contribute to the desired control command in equation 3.18, since $\Lambda = \Lambda_{\mathbf{d}}$.
Now that it is known how to obtain the end-effector force based on the arm stiffness, it needs to be converted back to desired joint torques that will realise such an end-effector force. As done earlier, for the gravity torque vector, it follows that the needed command torques follow from:

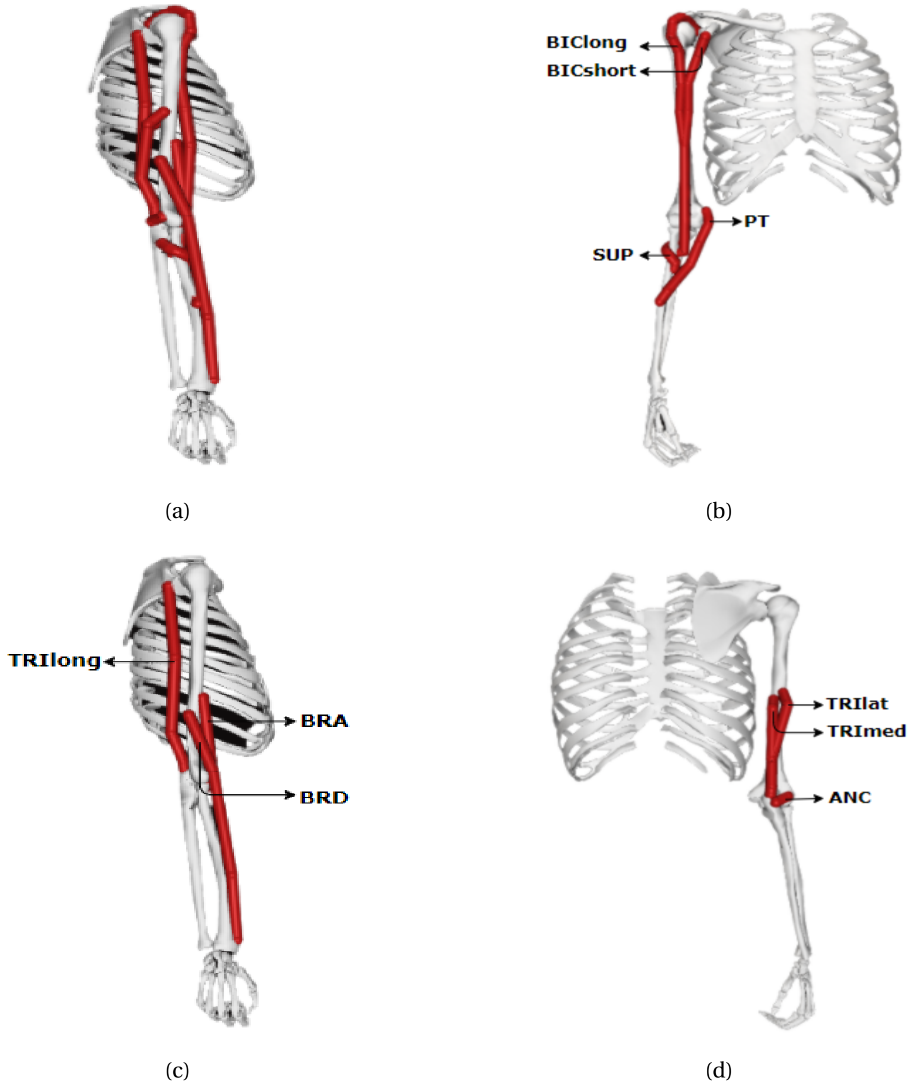
$$\tau_{\mathbf{Cmd}} = \mathbf{J_R}^{\mathrm{T}}(\mathbf{q})\mathbf{F_{Cmd}} \tag{3.19}$$

4 Design

This chapter covers how the proposed methodology from section 3 is realised. Firstly, it will be discussed what is used to determine the needed variables and how this is done. Secondly, an overview is given, which indicates how the different parts of the controller work together. At first, the designed controller is implemented in MATLAB to see whether it works as desired. Afterwards, the controller is implemented in Python in order to work in the ROS environment with the hardware.

4.1 The OpenSim model

The muscles that are used for calculating moment arms and measuring muscle activation by the use of surface EMG electrodes follow from section 3.2. The muscles are defined more clearly in figures 4.1(a) till 4.1(d) below, which are obtained from the used OpenSim Upper Extremity model, based on [35]. The abbreviations, functions, and attachment are given in the overview of table 4.1.



(a)

(b)

(c)

(d)

Figure 4.1: The used OpenSim muscle model, where figure *a* contains all the muscles of interest. The figures *b* till *d* indicate the muscle locations of the muscles under consideration.

Table 4.1: Overview of the muscles under consideration.

Muscle function (main)	**Biarticular**	**Abbre-viation**	**Function**[2]	**Attachment**[2]
Flexor	*Shoulder-elbow joint muscle:* Biceps long head	BIClong	Supinates forearm and, when it is supine, flexes forearm	**Origin:** Supraglenoid tubercle of scapula **Insertion:** Tuberosity of radius and fascia of forearm via bicipital aponeurosis
Flexor	*Shoulder-elbow joint muscle:* Biceps short head	BICshort	Resists dislocation of shoulder	**Origin:** Tip of coracoid process of scapula **Insertion:** Tuberosity of radius and fascia of forearm via bicipital aponeurosis
Extensor	*Shoulder-elbow joint muscle:* Triceps long head	TRIlong	Chief extensor of forearm; long head resists dislocation of humerus; especially important during adduction	**Origin:** Infraglenoid tubercle of scapula **Insertion:** Proximal end of olecranon of ulna and fascia of forearm
	Monoarticular			
Flexor	*Elbow joint muscle:* Brachioradialis	BRD	Relatively weak flexion of forearm; maximal when forearm is in midpronated position	**Origin:** Proximal two thirds of supra-epicondylar ridge of humerus **Insertion:** Lateral surface of distal end of radius proximal to styloid process
Flexor	*Elbow joint muscle:* Brachialis	BRA	Flexes forearm in all positions	**Origin:** Distal half of anterior surface of humerus **Insertion:** Coronoid process and tuberosity ulna
Extensor	*Elbow joint muscle:* Triceps lateral head	TRIlat	Chief extensor of forearm; especially important during adduction	**Origin:** Posterior surface of humerus, superior to radial groove **Insertion:** Proximal end of olecranon of ulna and fascia of forearm
Extensor	*Elbow joint muscle:* Triceps medial head	TRImed	Chief extensor of forearm; especially important during adduction	**Origin:** Posterior surface of humerus, inferior to radial groove **Insertion:** Proximal end of olecranon of ulna and fascia of forearm

Muscle overview (continued)

Muscle function (main)	**Monoarticular**	**Abbre-viation**	**Function**[2]	**Attachment**[2]
Pronator/ Flexor	*Elbow joint muscle:* Pronator teres	PT	Pronates and flexes forearm (at elbow)	**Origin:** Medial epicondyle of humerus (common flexor origin) **Insertion:** Middle of convexity of lateral surface of radius
Extensor	*Elbow joint muscle:* Anconeus	ANC	Assists triceps in extending forearm; stabilizes elbow joint; may abduct ulna during pronation	**Origin:** Lateral epicondyle of humerus **Insertion:** Lateral surface of olecranon and superior part of posterior surface of ulna
Supinator	*Elbow joint muscle:* Supinator	SUP	Supinates forearm; rotates radius to turn palm anteriorly or superiorly (if elbow is flexed)	**Origin:** Lateral epicondyle of humerus; radial collateral and anular ligaments; supinator fossa; crest of ulna **Insertion:** Lateral, posterior, and anterior surfaces of proximal third of radius

4.2 Myo Armband placement

In order to measure the activation of the muscles of interest, it is desired that the placement of the Myo Armband is correct. The level (height) of placement already follows from figures 3.2 and 3.3. However, the rotation of the Myo Armband matters as well. To obtain the correct rotation it is assumed that the Myo Armband is circular and all eight EMG sensors are placed equidistantly. This results in an angle of 45° between each sensor. These circular patterns can be projected on the arm and forearm and rotated so that all muscles of interest are covered. This results in the Myo Armband placement around the arm and forearm as in figure 4.2 and 4.3 respectively. In these figures, the green line above the first EMG sensor indicates that the charging port of the IMU is at the front. Therefore, the numbering of the rest of the EMG sensors follow from figure 3.4. Measuring the supinator muscle in the forearm might become problematic using surface EMG sensors due to the surrounding muscles, which are more on the surface and may interfere with the supinator muscle activations. The exact influence will be discussed in the experiment results section 6.

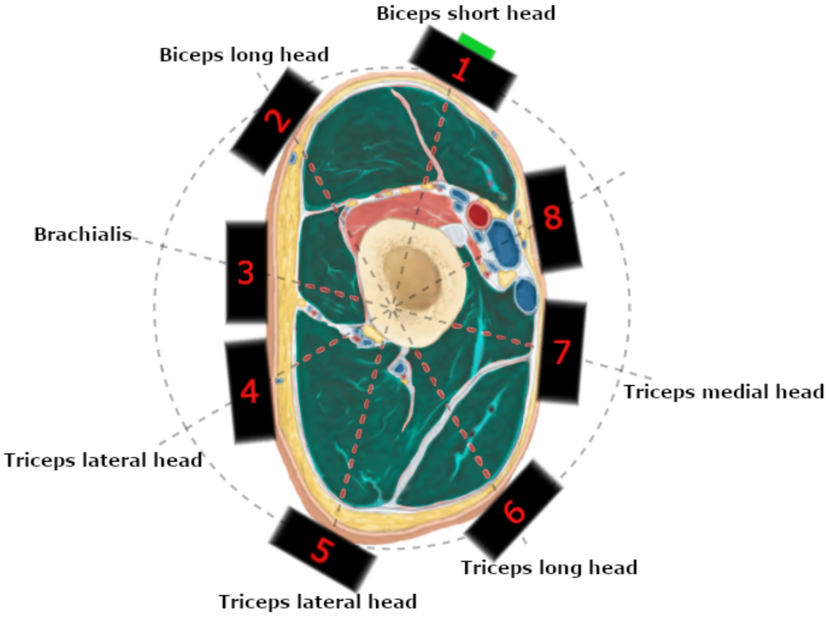


Figure 4.2: Myo Armband placement around the arm, for surface EMG measurements.

Table 4.2: Short overview of the electrode placement on the arm.

Muscle	**number of Electrode**
Biceps short head	1
Biceps long head	2
Brachialis	3
Triceps lateral head	4/5
Triceps long head	6
Triceps medial head	7

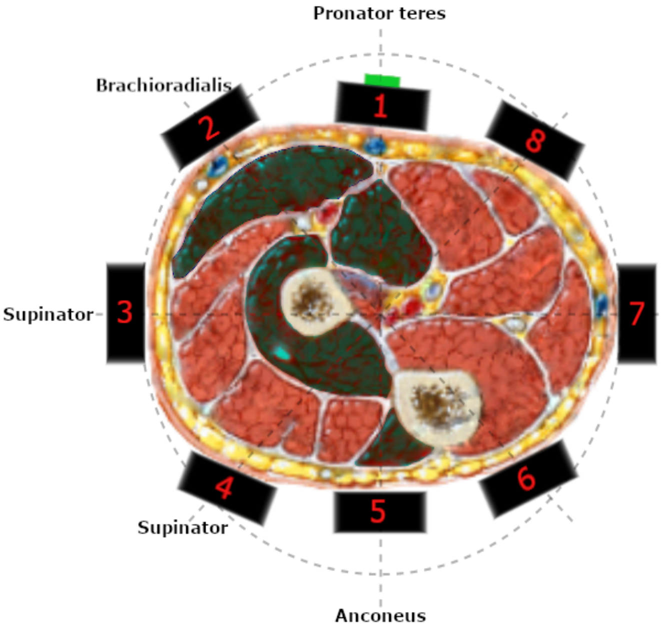


Figure 4.3: Myo Armband placement around the forearm, for surface EMG measurements.

Table 4.3: Short overview of the electrode placement on the forearm.

Muscle	**number of Electrode**
Pronator teres	1
Brachioradialis	2
Supinator	3/4
Anconeus	5

4.3 Muscle Jacobian

Within the developed OpenSim model (the Upper Extremity model), the muscles are connected to the limbs so that it resembles the muscles from the cadaver used in [35]. The muscle moment arms are different for every configuration of the human arm. Obtaining the human arm configuration in real-time and implementing it in the OpenSim model is therefore desired. OpenSim consists of an Application Programming Interface (API), and therefore communication with MATLAB and Python is possible. This allows changing the OpenSim model's configuration and reading out its corresponding muscle Jacobian matrix entries in real-time. The resulting muscle Jacobian that follows in real-time for the muscles of interest can then be defined as equation 4.1. In this equation, the moment arms for biarticular muscles are described by the first three row entries. The fourth till tenth rows describe the moment arms of the muscles that span the elbow. The angles used in calculating the muscle moment arm follow from the OpenSim model, and are defined according to table 4.4.

Table 4.4: Definition of the used abbreviated angles

Notation:	**Definition:**
q_{ea}	Elevation angle
q_{se}	Shoulder elevation angle
q_{sr}	Shoulder rotation angle
q_{ef}	Elbow flexion angle
q_{ps}	Pronation/Supination angle

$$\mathbf{J_M}(\mathbf{q}) = \begin{bmatrix}
\dfrac{\partial \lambda_{BIClong}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{BIClong}(\mathbf{q})}{\partial q_{se}} & \dfrac{\partial \lambda_{BIClong}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{BIClong}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{BIClong}(\mathbf{q})}{\partial q_{ps}} \\
\dfrac{\partial \lambda_{BICshort}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{BICshort}(\mathbf{q})}{\partial q_{se}} & \dfrac{\partial \lambda_{BICshort}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{BICshort}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{BICshort}(\mathbf{q})}{\partial q_{ps}} \\
\dfrac{\partial \lambda_{TRIlong}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{TRIlong}(\mathbf{q})}{\partial q_{se}} & \dfrac{\partial \lambda_{TRIlong}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{TRIlong}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{TRIlong}(\mathbf{q})}{\partial q_{ps}} \\
\dfrac{\partial \lambda_{BRD}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{BRD}(\mathbf{q})}{\partial q_{se}} & \dfrac{\partial \lambda_{BRD}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{BRD}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{BRD}(\mathbf{q})}{\partial q_{ps}} \\
\dfrac{\partial \lambda_{BRA}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{BRA}(\mathbf{q})}{\partial q_{se}} & \dfrac{\partial \lambda_{BRA}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{BRA}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{BRA}(\mathbf{q})}{\partial q_{ps}} \\
\dfrac{\partial \lambda_{TRIlat}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{TRIlat}(\mathbf{q})}{\partial q_{se}} & \dfrac{\partial \lambda_{TRIlat}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{TRIlat}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{TRIlat}(\mathbf{q})}{\partial q_{ps}} \\
\dfrac{\partial \lambda_{TRImed}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{TRImed}(\mathbf{q})}{\partial q_{se}} & \dfrac{\partial \lambda_{TRImed}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{TRImed}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{TRImed}(\mathbf{q})}{\partial q_{ps}} \\
\dfrac{\partial \lambda_{PT}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{PT}(\mathbf{q})}{\partial q_{se}} & \dfrac{\partial \lambda_{PT}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{PT}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{PT}(\mathbf{q})}{\partial q_{ps}} \\
\dfrac{\partial \lambda_{ANC}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{ANC}(\mathbf{q})}{\partial q_{ps}} & \dfrac{\partial \lambda_{ANC}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{ANC}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{ANC}(\mathbf{q})}{\partial q_{ps}} \\
\dfrac{\partial \lambda_{SUP}(\mathbf{q})}{\partial q_{ea}} & \dfrac{\partial \lambda_{SUP}(\mathbf{q})}{\partial q_{se}} & \dfrac{\partial \lambda_{SUP}(\mathbf{q})}{\partial q_{sr}} & \dfrac{\partial \lambda_{SUP}(\mathbf{q})}{\partial q_{ef}} & \dfrac{\partial \lambda_{SUP}(\mathbf{q})}{\partial q_{ps}}
\end{bmatrix} \tag{4.1}$$

4.4 Muscle stiffness

This section describes how the muscle stiffness matrix, resulting from all the contributing muscle stiffnesses, is determined.

A muscle stiffness matrix can be realised by the use of equation 3.5. The intrinsic stiffness and maximum elasticity that belong to the muscles, together with the muscle activation, will be different for every user. Therefore, it is most reliable to obtain these values during a calibration phase of the controller. However, intrinsic muscle stiffness and maximum elasticity are hard to obtain since this would require information such as muscle lengths, change in muscle lengths and force applied by the specific muscle. Therefore, the intrinsic stiffnesses and maximum elasticities for the muscles of interest are based upon the muscle modelling parameters used in the research of [35], where it is assumed that the optimal sarcomere length is 2.7 µm. By using the peak muscle force data and the optimal fiber length (table 4.5), combined with the steepest slope (red circle) in the length-tension relation curve from figure 4.4, an approximation of the muscle stiffnesses can be made. An example calculation for this approximation is done for the Triceps long head (TRIlong) muscle, as shown in equation 4.2. All the resulting muscle stiffnesses can be found in table 4.6.

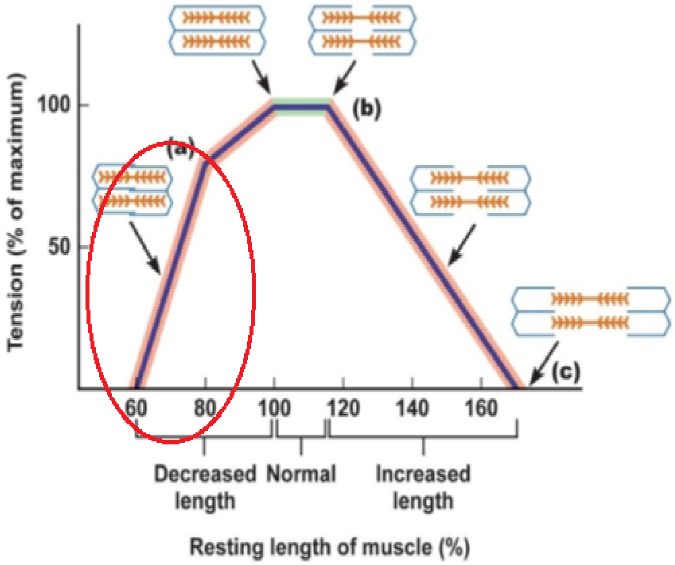


Figure 4.4: Muscle length-tension curve, obtained from [7].

Table 4.5: Muscle specific parameters.

Muscle	**Peak force** [N], from [35]	**Optimal fiber length** [m], from [35]
TRIlong	798.5	0.134
TRIlat	624.3	0.114
TRImed	624.3	0.114
BIClong	624.3	0.116
BICshort	435.6	0.132
BRD	261.3	0.173
BRA	987.3	0.086
PT	566.2	0.049
ANC	350.0	0.027
SUP	476.0	0.033

$$\begin{aligned} K_{\mu}^{\text{TRIlong}} &= \frac{\Delta\text{Tension}}{\Delta\text{muscle length}} \\ &= \frac{(0.8\cdot 798.5)}{(0.8\cdot 0.134)-(0.6\cdot 0.134)} \\ &= 23\,835.82\,\text{N}\,\text{m}^{-1} \end{aligned} \tag{4.2}$$

Table 4.6: Muscle stiffness values

Muscle	**Muscle stiffness** (K_μ) [Nm-1]
TRIlong	23835.82
TRIlat	21905.26
TRImed	21905.26
BIClong	21527.59
BICshort	13200
BRD	6041.18
BRA	45920.93
PT	46220.41
ANC	51851.85
SUP	57696.97

The obtained muscle stiffnesses (table 4.6) can be used to determine an approximation for the intrinsic muscle stiffnesses and the maximum elasticities. The proposed method is based on maximum and minimum activation data of the muscles obtained during a calibration phase for normalising the EMG data. This will be more elaborated on in section 4.4.1, where u_{max} and u_{min} describe the maximum and minimum EMG data for a specified muscle, respectively. Using these u_{max} and u_{min} values allow for an intrinsic muscle stiffness calculation which will be different for every user. This method assumes that with minimum muscle activation, the resulting muscle stiffness must be the intrinsic muscle stiffness. Therefore, the part belonging to the minimum activation is taken from the determined muscle stiffness in table 4.6 as the intrinsic muscle stiffness. The intrinsic stiffness can be determined with this approach according to equation 4.3, where superscript i defines the specific muscle.

$$K^i_{int} = \frac{u^i_{min}}{u^i_{max}} \cdot K^i_\mu \tag{4.3}$$

The maximum muscle elasticity can be found using the muscle stiffness values from table 4.6, combined with equation 3.5 and equation 4.3. This results in equation 4.4.

$$K^i_{\mu,max} = K^i_\mu - K^i_{int} \tag{4.4}$$

4.4.1 Muscle activation

It follows from equation 3.5 that the muscle activation (u_i) is a normalised EMG reading. Therefore, the maximum and minimum muscle activation for every contributing muscle needs to be known. These can be obtained once the EMG of every contributing muscle is measured during a calibration phase. The muscle activation can then be normalised for every muscle according to the following equation:

$$u_i = \frac{(u_{data} - u_{min})}{(u_{max} - u_{min})} \tag{4.5}$$

Where, u_{data} is the raw EMG data after applying a Low Pass Filter and u_{min} and u_{max} describe the minimum and maximum activation of the muscle respectively.

From previous research of [14] it followed that a second-order Butterworth with a low-pass corner frequency at 5 [Hz] keeps the important information intact as much as possible, whilst not too much delay is introduced to the signal.

Combining the intrinsic muscle stiffness, maximum muscle elasticity and the normalised muscle activation, according to equation 3.5 ($K_{\mu}^{i} = K_{int}^{i} + K_{\mu,max}^{i} \cdot u_i$), results in the muscle stiffness matrix as 4.6. The order of the diagonal entries of the matrix is dependent on the muscle Jacobian from equation 4.1.

$$\mathbf{K}_{\mu} = \mathrm{diag}(K_{\mu}^{BIClong}, K_{\mu}^{BICshort}, K_{\mu}^{TRIlong}, K_{\mu}^{BRD}, K_{\mu}^{BRA}, K_{\mu}^{TRIlat}, K_{\mu}^{TRImed}, K_{\mu}^{PT}, K_{\mu}^{ANC}, K_{\mu}^{SUP}) \tag{4.6}$$

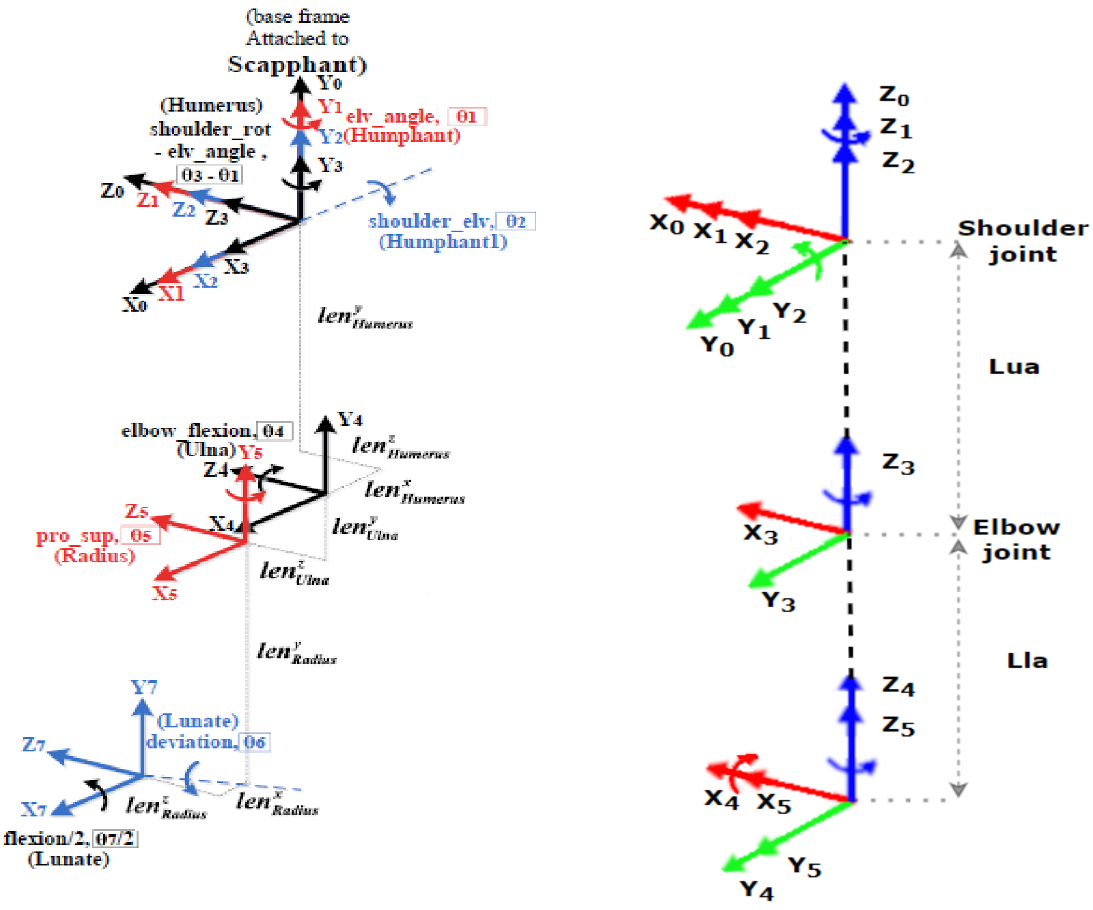
4.5 Joint stiffness to endpoint stiffness

As discussed, the transformation from joint stiffness $\mathbf{K_J}$ to the endpoint stiffness $\mathbf{K_e}$ can be described by 4.7.

$$\mathbf{K_e} = \mathbf{J_H}^{-\mathrm{T}}(\mathbf{q})\mathbf{K_J}\mathbf{J_H}^{-1}(\mathbf{q}) \tag{4.7}$$

Where $\mathbf{J_H}(\mathbf{q})$ is the manipulator Jacobian of the human arm, which describes the endpoint position based on the lengths and configuration of the limbs. An analytical and a geometric approach can be taken to determine the Jacobian of the human arm. These approaches and their results for a simplified human arm configuration are described in section 2.2. A similar approach can be taken for the human arm Jacobian based on similar rotations as the OpenSim model. However, assuming that the wrist joint is fixed, the 7-DoF kinematic OpenSim model (figure 4.5(a)) can be simplified to a 5-DoF (see figure 4.5(b)). This allows endpoint stiffness calculation based on the configuration of the human arm alone. Furthermore, a calculation of the human arm Jacobian outside of the OpenSim software makes the controller more independent of the OpenSim software. This could be handy if there is chosen to determine the human arm endpoint stiffness only on the posture of the human arm, which can be realised by setting the joint stiffness matrix as an identity matrix, hence no OpenSim is required to determine the muscle moment arm Jacobian.

In determining the human arm Jacobian, there is no need to dive into anatomical relations in the forearm since a Myo Armband will be worn around the whole forearm and not the radius bone only (as figure 4.5(a) indicates). Therefore, the forearm can be simplified as just one limb which extends the upper arm once in initial position (**Lla** in figure 4.5(b)). The simplified kinematic model is based on the right-hand coordinate frame. This allows for easier integration with the endpoint positions that follow from the haptic device.



(a)

(b)

Figure 4.5: *(a)* Kinetmatic model used for the OpenSim model, obtained from [38]. *(b)* Kinematic model used for human arm Jacobian, based on the right hand coordinate frame.

Within the kinematic model of figure 4.5(b) the different frames can be expressed as follows:

- Frame 0 (Ψ_0) is the inertial reference frame.
- Frame 1 (Ψ_1) is used to describe a rotation around the $z-$axis of the shoulder with respect to frame 0 (Ψ_0).
- Frame 2 (Ψ_2) is used to describe a rotation around the $y-$axis of the shoulder with respect to frame 1 (Ψ_1).
- Frame 3 (Ψ_3) is used to describe a translation and rotation around the $z-$axis of the shoulder with respect to frame 2 (Ψ_2).
- Frame 4 (Ψ_4) is used to describe the translation and flexion of the elbow joint with respect to frame 3 (Ψ_3).
- The endpoint frame, frame 5 (Ψ_5), describes supination or pronation (rotation) of the elbow with respect to frame 4 (Ψ_4).

For ease of notation these frame/arm rotations are simplified according to table 4.7. Where the numbers indicate the specific frame rotation with respect to the previous frame.

Table 4.7: Frame/arm rotation notation.

Notation:	**Definition:**
q_{s,z_1}	Rotation around the z-axis of the shoulder joint
q_{s,y_2}	Rotation around the y-axis of the shoulder joint
q_{s,z_3}	Rotation around the z-axis of the shoulder joint
q_{e,x_4}	Rotation around the x-axis of the elbow joint
q_{e,z_5}	Rotation around the z-axis of the elbow joint

Using the simplified kinematic model (figure 4.5(b)) with its frame descriptions, and the methodology from section 2.2.2, gives rise to the expression for the geometric human arm Jacobian, which is independent of the OpenSim software. Furthermore, the determined H-matrices are used to develop MATLAB and Python models, which presents the configuration of the arm, as can be seen in figure 4.6 below. Doing so allows comparing the MATLAB/Python model and the OpenSim model for different angle inputs.

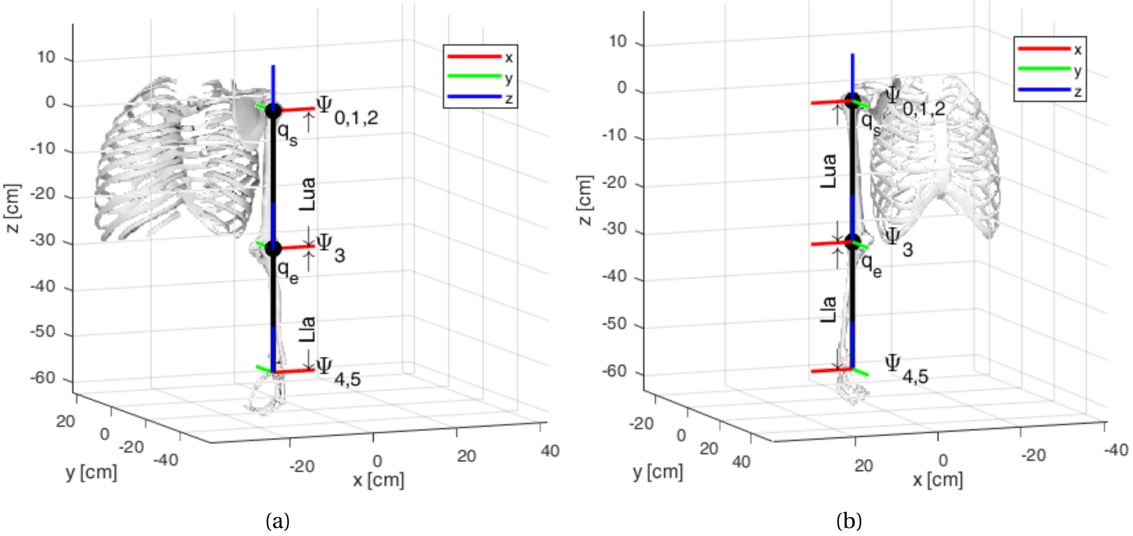


Figure 4.6: Initial configuration of the human arm in 3D with the OpenSim bodies included as a reference. The angles q_{s,z_1}, q_{s,y_2}, q_{s,z_3}, q_{e,x_4} and q_{e,z_5} are the rotations of the shoulder and the elbow respectively, and follow from figure 4.5(b). In the initial configuration these are set to 0 [rad]. The different frames used to determine the arm Jacobian are represented by Ψ_i.

The angle inputs that result in a certain configuration for the MATLAB/Python model are obtained by the use of the internal IMU's orientation data of the Myo Armbands, as will be explained in section 4.5.1. However, the endpoint frame of the kinematic model from figure 4.5(b) is still rotated with respect to the inertial reference frame. To make transferring the impedance to the robotic arm and interpreting the results easier, the endpoint frame is only translated with respect to the inertial reference frame (Ψ_0). Furthermore, the setpoint obtained from the haptic device is with respect to the inertial reference frame. The approach from 2.2.2 is taken to correct for the rotated end-effector frame. Furthermore, this approach allows the calculation of a Moore-Penrose pseudo-inverse Jacobian, which is square and non-singular, and can therefore be used for the desired endpoint stiffness calculation according to equation 4.7. The Moore-Penrose pseudo-inverse is not dynamically consistent. However, this should not make a difference since this pseudo-inverse will not be used to determine new angles. The new angles follow from the Myo Armbands (as will be elaborated on in section 4.5.1 and 4.5.2) to avoid the inconsistency of the Moore-Penrose pseudo-inverse. They are directly substituted in the pseudo-inverse to determine the final endpoint stiffness.

The obtained MATLAB/Python configuration can be used to determine the angles needed for the OpenSim model to reach (approximately) the same configuration. This is desired since the Muscle Jacobian is dependent on this configuration.
From figure 4.5 it follows that the OpenSim angles follow from the MATLAB/Python angles according to table 4.8.

Table 4.8: OpenSim angles following from the kinematic MATLAB/Python model.

OpenSim angle	**MATLAB/Python angle**
Elevation angle	$q_{s,z_1} + \pi$
Shoulder elevation	q_{s,y_2}
Shoulder rotation	$q_{s,z_3} + q_{s,z_1}$
Elbow flexion	q_{e,x_4}
Pronation/Supination	q_{e,z_5}

Implementing the measurement results from the Myo Armband and the haptic device into the endpoint stiffness equation for the human arm 4.7 gives symmetric matrix as 4.8:

$$\mathbf{K_e} = \begin{bmatrix} K_{xx} & K_{xy} & K_{xz} & K_{x,q_x} & K_{x,q_y} & K_{x,q_z} \\ K_{yx} & K_{yy} & K_{yz} & K_{y,q_x} & K_{y,q_y} & K_{y,q_z} \\ K_{zx} & K_{zy} & K_{zz} & K_{z,q_x} & K_{z,q_y} & K_{z,q_z} \\ K_{q_x,x} & K_{q_x,y} & K_{q_x,z} & K_{q_x,q_x} & K_{q_x,q_y} & K_{q_x,q_z} \\ K_{q_y,x} & K_{q_y,y} & K_{q_y,z} & K_{q_y,q_x} & K_{q_y,q_y} & K_{q_y,q_z} \\ K_{q_z,x} & K_{q_z,y} & K_{q_z,z} & K_{q_z,q_x} & K_{q_z,q_y} & K_{q_z,q_z} \end{bmatrix} \tag{4.8}$$

The matrix entries describing a translational spring can be used to develop a stiffness ellipsoid, which makes interpretation of the results easier. This stiffness ellipsoid is inversely proportional to the stiffness.
As an example, a stiffness ellipsoid at the endpoint of the human arm is given (figure 4.7). In this example, it is assumed that all the muscles are fully activated, the intrinsic muscle stiffness is the same for all the muscles, and the desired endpoint position is known. This results in a stiffness profile that is focused on the configuration of the arm. Comparison of this ellipsoid with figure 2a of earlier research on endpoint stiffness for a prosthetic limb ([39]) indicates that the endpoint stiffness calculation is more or less on the right track.

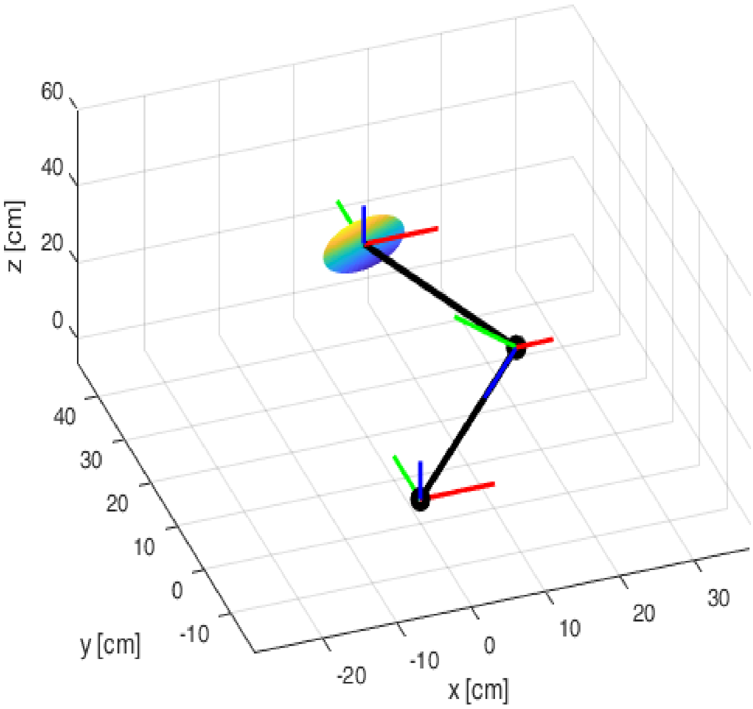


Figure 4.7: Stiffness ellipsoid for an example configuration based on the position x=0.05 m, y=0.35 m and z=0.25 m.

4.5.1 Obtain shoulder and elbow angles

In order to obtain the real-time configuration of the human arm, it is necessary to either know the endpoint location of the human arm or know the shoulder and elbow angles. Via the endpoint location, the angles can be determined by the use of an inverse Jacobian. However, this configuration might not precisely match the actual human arm position since several configurations are possible for the human arm to reach a specific point. Therefore, it is desired to obtain the angles of the human arm directly. This can be done by the use of the gyroscope that is present in the IMU of the Myo Armband. The orientation of the Myo is presented as a quaternion. However, the orientation of the arm and forearm are represented by a rotation matrix. Therefore, the obtained quaternions from the Myo Armbands should be converted to rotation matrices. According to [40] this can be done by the use of the following equation:

$$\text{Rotation matrix} = \begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2\left(q_1q_2-q_0q_3\right) & 2\left(q_1q_3+q_0q_2\right) \\ 2\left(q_2q_1+q_0q_3\right) & q_0^2-q_1^2+q_2^2-q_3^2 & 2\left(q_2q_3-q_0q_1\right) \\ 2\left(q_3q_1-q_0q_2\right) & 2\left(q_3q_2+q_0q_1\right) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \tag{4.9}$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{4.10}$$

Where, q_0, q_1, q_2 and q_3 are the four elements of the quaternion and are commonly known as Euler parameters. The first Euler parameter, q_0, is a scalar part and q_1, q_2, and q_3 are the vector parameters that describe a vector in three-dimensional space, x, y and z, respectively.

IMU frame correction

In figure 3.4 the reference frame of the Myo Armband can be found. Once the Myo Armband is worn around the arm/forearm (according to figure 4.2/4.3) it follows that the reference frame is different for the Myo Armband than the desired reference frame for the kinematic model of the human arm (figure 4.6). The difference between these two frames can be seen more clearly in the top view comparison of figure 4.8.

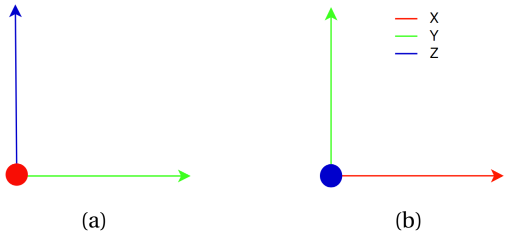


Figure 4.8: Different reference frames. In (*a*) the reference frame for the Myo can be found, and in (*b*) the reference frame of the kinematic model can be found.

A correction for this problem is found by simply interchanging the columns of the rotation matrix that follows from the IMU. Hence, the rotation matrix that expresses a rotation with respect to the kinematic frame 4.8(b) can be found by the rotation matrix as:

$$\text{Rotation matrix kinematic frame} = \begin{bmatrix} r_{12} & r_{13} & r_{11} \\ r_{22} & r_{23} & r_{21} \\ r_{32} & r_{33} & r_{31} \end{bmatrix} \tag{4.11}$$

4.5.2 Orientation calibration

The initial configuration of figure 4.6 is assumed for the kinematic model. Therefore, it is desired that the inertial reference frames of the Myo Armbands correspond with this initial configuration. Hence, the Myo Armbands of the arm and forearm should be calibrated once the initial posture of the human arm equals the initial configuration. If this will not be done, the arm and forearm rotations are calculated from a frame of reference obtained during the Myo startup period.

The rotation of the arm/forearm with respect to the startup frame can be expressed as $R^{startup\_frame}_{arm\_movement}$. To obtain the arm/forearm movement with respect to the inertial reference frame, hence the frame of the initial configuration in figure 4.6, a rotation matrix is needed that converts the startup frame to the reference frame of the initial configuration. This matrix is obtained by measuring the rotation matrix for a certain amount of time once the arm is in the initial configuration. The rotation that is found can be described as the rotation of the reference frame expressed in the startup frame of the Myo $\left(R^{startup\_frame}_{ref\_frame\_arm}\right)$. Taking the inverse of this calibrated rotation matrix will then result in the arm/forearm movement with respect to the frame of the initial configuration. This leads to equation 4.12 and 4.13.

$$R^{ref\_frame\_arm}_{arm\_movement} = \left(R^{startup\_frame}_{ref\_frame\_arm}\right)^{-1} \cdot R^{startup\_frame}_{arm\_movement} \tag{4.12}$$

$$R^{ref\_frame\_forearm}_{forearm\_movement} = \left(R^{startup\_frame}_{ref\_frame\_forearm}\right)^{-1} \cdot R^{startup\_frame}_{forearm\_movement} \tag{4.13}$$

It is desired that the rotation of the forearm is with respect to the frame of the arm. Otherwise, there is a rotation of the forearm once the arm is moved, which is not desired. Due to the calibration method, the reference frame of the arm and forearm are almost identical. Therefore, the following equation (4.14) can be used to express the movement of the forearm with respect to the current frame of the arm.

$$R^{arm\_movement}_{forearm\_movement} = \left(R^{ref\_frame\_arm}_{arm\_movement}\right)^{-1} \cdot R^{ref\_frame\_forearm}_{forearm\_movement} \tag{4.14}$$

4.5.2.1 Shoulder angles

As follows from section 4.5 the rotation matrix for the arm contains a rotation around the z-axis, y-axis and (again) z-axis, respectively. This results in:

$$R_{z_1,y_2,z_3} = \begin{bmatrix} \mathrm{c}q_{s,z_1}\mathrm{c}q_{s,y_2}\mathrm{c}q_{s,z_3} - \mathrm{s}q_{s,z_1}\mathrm{s}q_{s,z_3} & -\mathrm{c}q_{s,z_1}\mathrm{c}q_{s,y_2}\mathrm{s}q_{s,z_3} - \mathrm{s}q_{s,z_1}\mathrm{c}q_{s,z_3} & \mathrm{c}q_{s,z_1}\mathrm{s}q_{s,y_2} \\ \mathrm{s}q_{s,z_1}\mathrm{c}q_{s,y_2}\mathrm{c}q_{s,z_3} + \mathrm{c}q_{s,z_1}\mathrm{s}q_{s,z_3} & -\mathrm{s}q_{s,z_1}\mathrm{c}q_{s,y_2}\mathrm{s}q_{s,z_3} + \mathrm{c}q_{s,z_1}\mathrm{c}q_{s,z_3} & \mathrm{s}q_{s,z_1}\mathrm{s}q_{s,y_2} \\ -\mathrm{s}q_{s,y_2}\mathrm{c}q_{s,z_3} & \mathrm{s}q_{s,y_2}\mathrm{s}q_{s,z_3} & \mathrm{c}q_{s,y_2} \end{bmatrix} \tag{4.15}$$

Converting the quaternions from the Myo that is worn around the arm to a rotation matrix, by the use of equation 4.9 and 4.11, gives the solution to the rotation matrix of 4.15. This can be used to determine the shoulder angles for the kinematic model. After calculation, and according to [41], this results in the following equations. In these equations Atan2() is used to retain sign information.

$$q_{s,y_2} = \mathrm{Atan2}\left(\sqrt{r_{32}^2 + r_{33}^2}, r_{31}\right) \tag{4.16}$$

$$q_{s,z_1} = \mathrm{Atan2}\left(r_{21}/sq_{s,y_2}, r_{11}/sq_{s,y_2}\right) \tag{4.17}$$

$$q_{s,z_3} = \mathrm{Atan2}\left(r_{33}/sq_{s,y_2}, -r_{32}/sq_{s,y_2}\right) \tag{4.18}$$

4.5.2.2 Elbow angles

The same can be done for the Myo Armband that is worn around the forearm. The rotation matrix for the kinematic model can be defined as:

$$R_{x_4,z_5} = \begin{bmatrix} \mathrm{c}q_{e,z_5} & -\mathrm{s}q_{e,z_5} & 0 \\ \mathrm{c}q_{e,x_4}\mathrm{s}q_{e,z_5} & \mathrm{c}q_{e,x_4}\mathrm{c}q_{e,z_5} & -\mathrm{s}q_{e,x_4} \\ \mathrm{s}q_{e,x_4}\mathrm{s}q_{e,z_5} & \mathrm{s}q_{e,x_4}\mathrm{c}q_{e,z_5} & \mathrm{c}q_{e,x_4} \end{bmatrix} \tag{4.19}$$

The solutions of interest are:

$$q_{e,x_4} = \arccos\left(r_{31}\right) \tag{4.20}$$

$$q_{e,z_5} = -\arcsin\left(r_{13}\right) \tag{4.21}$$

These are chosen since they are in line with the physical limitations of the elbow. The elbow can not extend any further than the configuration from figure 4.6, hence the angle q_{e,x_4} can not be negative. Furthermore, the elbow flexion is limited as well. Therefore, the solution for q_{e,x_4} should be equation 4.20 since this solution has a range of $0° \leq q_{e,x_4} \leq 180°$. There are some limitations for supination and pronation of the elbow as well causing a rotation limitation of approximately $-90° \leq q_{e,z_5} \leq 90°$. This automatically requires the solution as in 4.21, which covers the same range.

4.6 Controller calibration within the ROS environment

Now that everything is known, the controller is implemented in the ROS environment, allowing communication between all the hardware used and the written control nodes. As discussed earlier, in section 4.4 the upper and lower limits of the EMG signals are needed for every contributing muscle in order to normalise all the muscle activations. In this way, they can be used to determine the specific muscle stiffness.

In figure 4.9 a block diagram can be seen which shows the hardware (blue blocks), communication protocols (orange circles), ROS nodes (purple blocks), and subscribed data (arrows) which are necessary to perform a muscle calibration. However, the first thing done is initialising the orientation sensors of the Myo's after they are placed on the arm and forearm. By aligning the arm with the body and placing the palm of the hand on the side of the thigh, the initialisation configuration is obtained. The frames for both Myo's in this configuration are determined and saved such that they can be used as a reference frame.

If everything is launched as is indicated in the block diagram (4.9), the operator is able to press the spacebar to start the calibration (and backspace to redo a calibration step). Once the spacebar is pressed, the current endpoint position of the haptic device (Virtuose) is taken as the starting position, and the haptic device will linearly increase its endpoint force in one of the directions indicated in figure 4.10. During the increase in endpoint force, the operator will try to counteract the force in the specific direction by increasing his/her endpoint stiffness as well. While the operator counteracts the endpoint force, all the EMG signals from the contributing muscles are measured. Once a deviation of 0.08 m between the starting point of the calibration and current position is reached the calibration in that direction is stopped, as well as the EMG data acquisition. The whole process starts again once the operator presses the spacebar again. However, the applied force by the Virtuose is now under an angle of 45. This whole process will be done under several angles, as can be seen in figure 4.10. The forces applied to the human arm from different directions allow to obtain the optimal upper and lower EMG limits of the muscles, since for every direction, another human arm endpoint stiffness is desired and therefore, the activation of the muscles will be different for every direction.

The obtained upper and lower limits of the muscle activations are stored on the ROS parameter server. This allows to shut down the calibration without losing the determined upper and lower EMG limits.

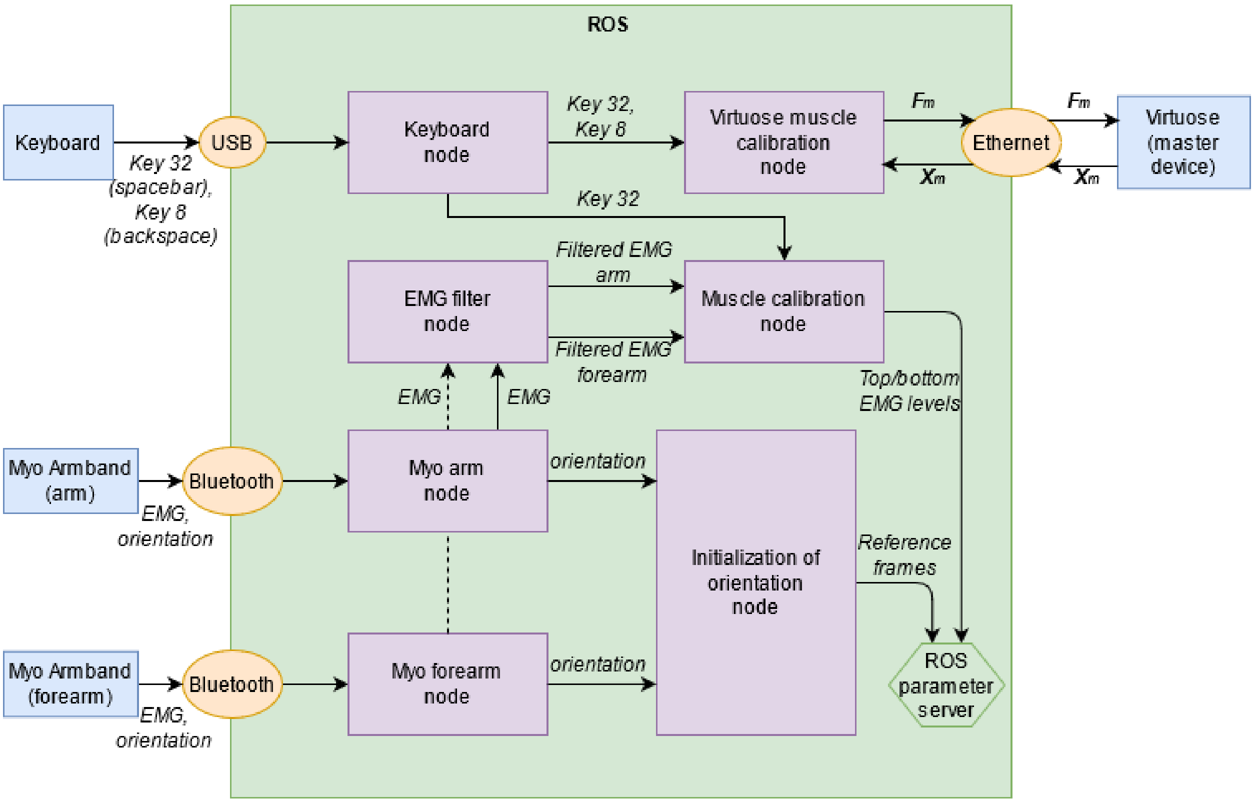


Figure 4.9: Block diagram showing the needed hardware, communication, ROS nodes and subscribed data in order to perform the initialisation of the orientation sensor and the muscle calibration to determine the upper and lower limits of all the muscles under consideration. F_m and X_m indicate the force command of the Virtuose and the position of the Virtuose, respectively.

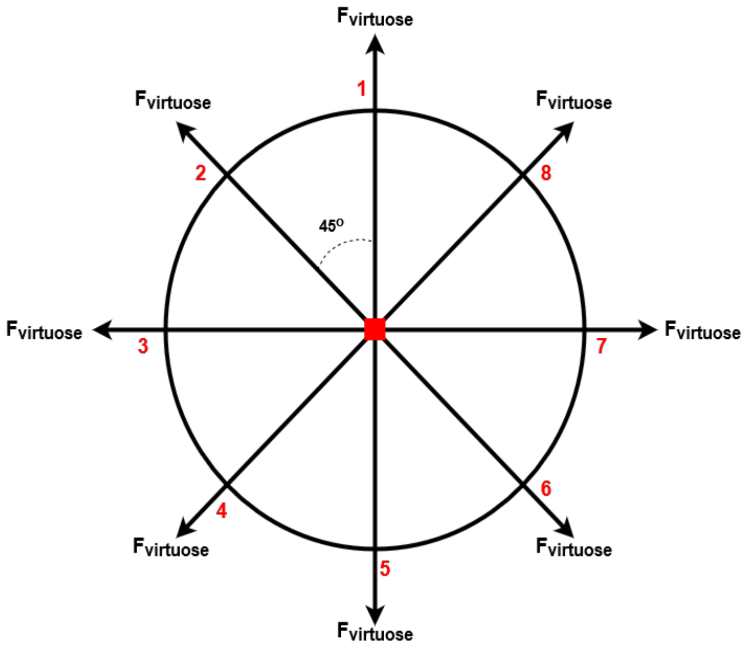


Figure 4.10: Directions in which the Virtuose will linearly increase its endpoint stiffness. The red dot indicates the starting position of the calibration.

After obtaining the upper and lower EMG limits, a new calibration script can be launched. The block diagram for this calibration method can be found in figure 4.11. The performance of this calibration method is the same. However, this time the calibration is used to scale the endpoint stiffness that follows from the endpoint stiffness from the human arm. Loading the upper and lower limits of the EMG from the ROS parameter server, together with the reference frames of both Myo's, allow the designed controller (Human arm endpoint stiffness node) to determine the endpoint stiffness of the human arm. During the calibration (where the operator again tries to counteract the forces applied by the Virtuose), the upper and lower limits of the human arm endpoint stiffness are determined. These limits are saved to the ROS parameter server such that they can be used to scale the human arm endpoint stiffness within a range of 200 to 800 $\mathrm{N\,m^{-1}}$ before it is applied to the robot arm. According to the datasheet of the Franka Emika Panda, the adjustable translational stiffness can range from 0 till 3000 $\mathrm{N\,m^{-1}}$. However, with safety reasons in mind, the upper and lower limits of the Franka Emika Panda robot arm are determined empirically and set to a range of 200 till 800 $\mathrm{N\,m^{-1}}$.

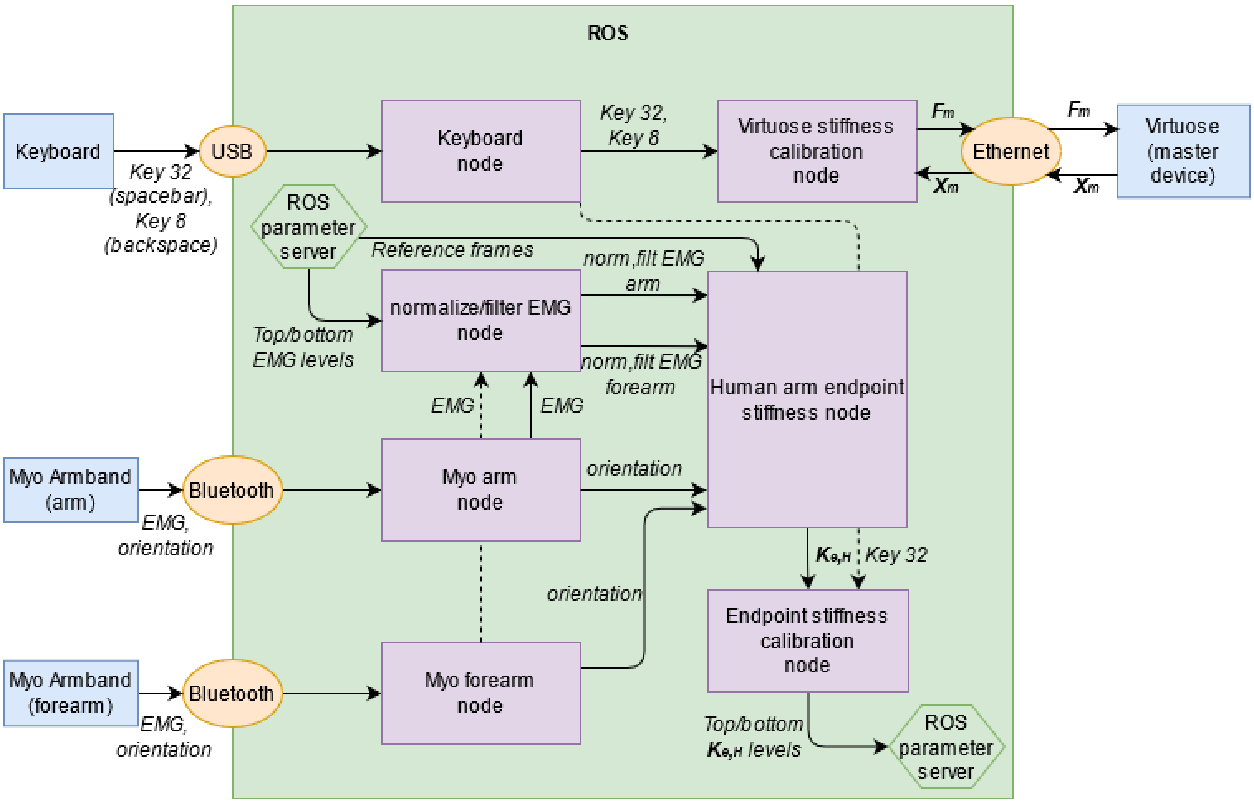


Figure 4.11: Block diagram showing the needed hardware, communication, ROS nodes and subscribed data in order to perform a calibration that obtains the upper and lower limits of the human arm endpoint stiffness.

4.7 The ROS telemanipulation impedance controller

Once the calibrations are performed and the desired data for the orientation of the Myo's, upper and lower boundaries of the muscle activation, and human arm endpoint stiffness are determined, the telemanipulation control can take place. The block diagram which indicates how this control takes place can be found in figure 4.12. As can be seen, the scaled endpoint stiffness is subscribed to by the Franka impedance controller, where it is used to determine the endpoint force for the robotic arm based on the position difference between the master and slave site. By using the Franka Emika Panda Jacobian, the endpoint force will be converted to joint torques that allow the robot arm to execute the commanded endpoint force. The endpoint force is fed back to the master site to feel the robot arm's behaviour and environment

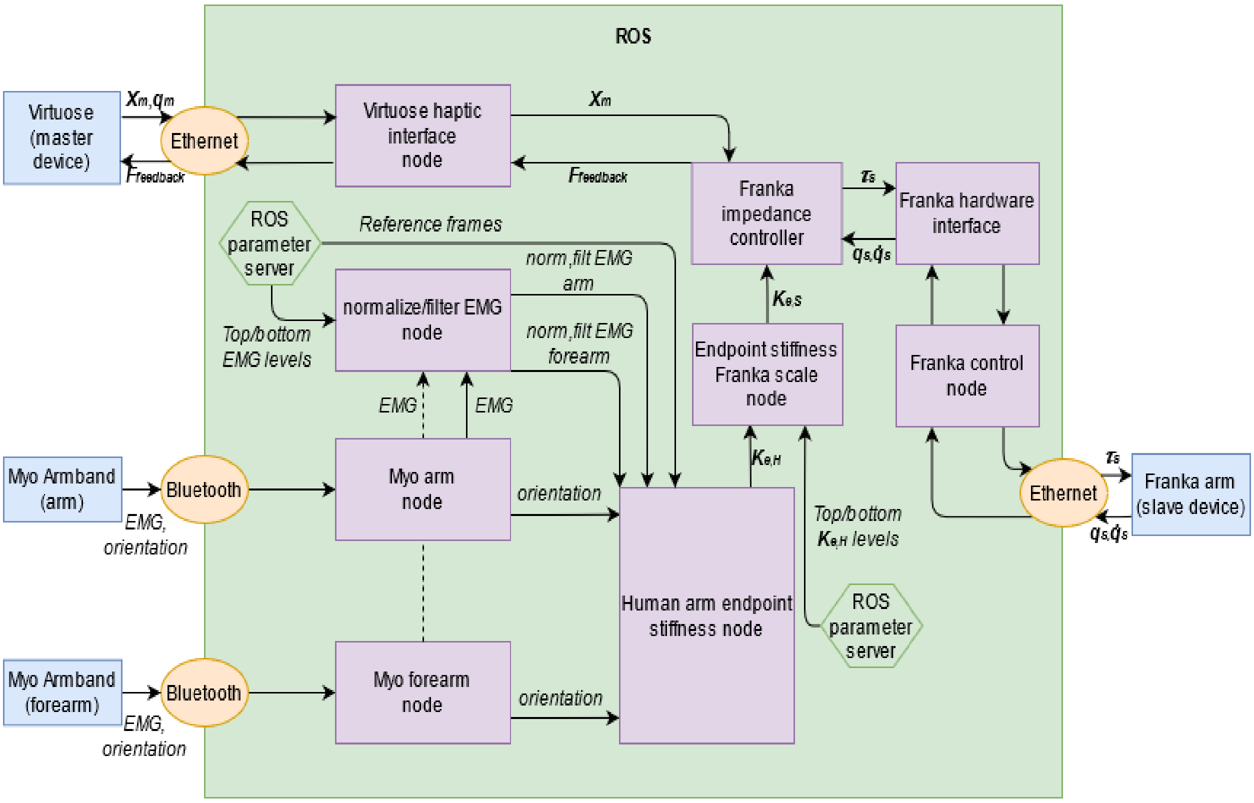


Figure 4.12: Block diagram showing the needed hardware, communication, ROS nodes and subscribed data in order to perform telemanipulation impedance control.

5 Experiments

This section provides information about the experiments that are performed. The experiments are designed in such a way that it gives insights into the influence of the endpoint impedance of the robot arm on perceived efficiency (being fast) and the perceived effectiveness (being precise). Furthermore, they should include the increase or decrease in the intuitiveness of telemanipulation task execution in an unpredictable environment. The results of these specific experiments give rise to an answer to the research question, which stated:

– *"To what extent will the endpoint impedance of the robot arm, based on activation of the human arm muscles and posture of the human arm, increase the performance and intuitiveness of telemanipulation task execution in an unpredictable environment compared to an impedance profile based on co-contraction of a single muscle pair?"*

The research done is based on transferring the endpoint stiffness of the human arm towards the slave robot. This does not take into account the wrist movement and hence the hand of the operator. To exclude possible wrist rotations as much as possible the experiments are designed in such a way that they should not allow any rotation in the end-effector. Hence, rotation of the gripper has no result on any movement of the end-effector of the slave robot.

The results of the newly designed controller are compared with an endpoint stiffness controller from the previous research of [14], which determines the endpoint stiffness based on the co-contraction of a single muscle pair in the forearm. A user study is done to know whether an increase or decrease in the intuitive control, perceived effectiveness, and perceived efficiency of the robot arm endpoint is experienced. During this user study, a group of 16 people (right-handed) performed the same experiments for both controllers. However, the data of 14 people (3 females and 11 males) will be analysed since the data of 2 participants is corrupted. The age of the participants ranges from 19 to 25 years, with a mean age of 23.29 years and a standard deviation of 1.64 years. The controller order was chosen randomly for each participant. The same holds for the heavy and light blocks used for the experiments. During the experiments, the participant followed the slave robot's movements on a screen while controlling the haptic device. This gave the idea that they were not physically present at the robot arm site.
After the experiments were performed for one specific controller, the participants were asked to fill in a questionnaire on the QuestionPro platform. This questionnaire is based on [42] and a PDF version of the used questionnaire can be found in Appendix C. The questions are answered based on a seven-point Likert scale ranging from 1 to 7, where 1 corresponds with "strongly agree" and 7 with "strongly disagree". The questions can be subdivided into three different scales: Perceived Usefulness (PU, question 7 to 10), Perceived Ease of Use (PEU, question 11 to 12) and Comfort (C, question 13 to 14).
The Likert scales that follow from both questionnaires are used to perform a paired sample t-test to see any statistically significant differences between the two controllers. For this paired sample t-test, a p-value threshold of 0.05, the most common threshold value, is used.

To obtain good insights into the difference between the two controllers (concerning the perceived effectiveness and perceived efficiency), the experiments should at least focus on tasks that require a change in endpoint stiffness if it is desired to perform the task optimally. This leads to two different tasks, one that focuses on accurate performance and one that focuses on performance time. These tasks can be described as follows:

Task 1: Path to precision
Push a physical block with a specific weight along a marked area. The path that the block should follow is indicated with tape. Before the block is pushed, a "start block" button should

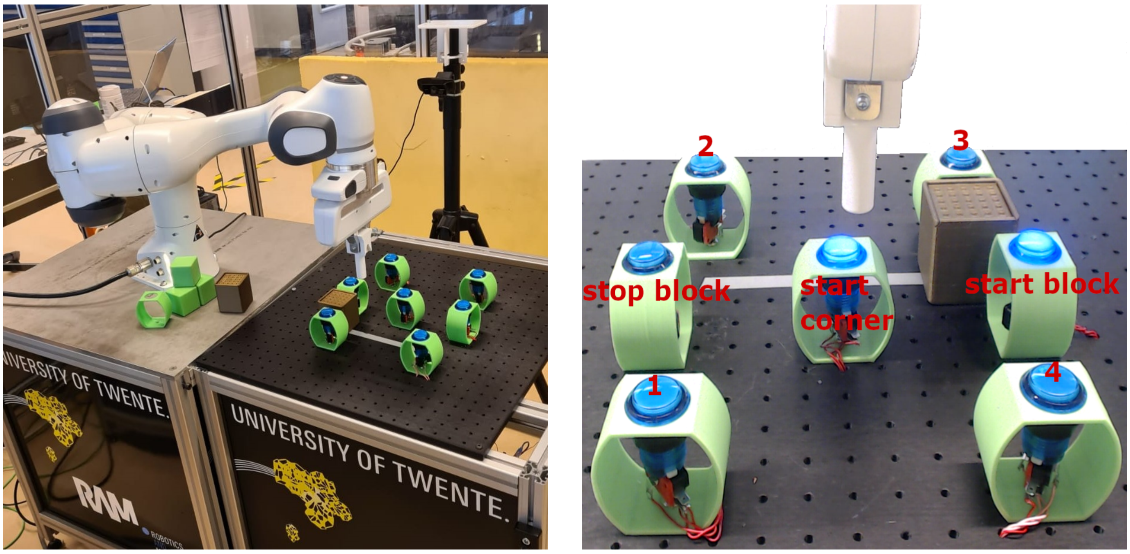
be pressed. After reaching the end of the marked area, the “stop block” button should be pressed.

Task 2: Corner reaction
Start at the centre of a square and press the “start corner” button. After a few seconds, one of the corners will light up. Reach to the corner and press the arcade button to turn the LED off. After the button in the corner is pressed, return to the centre and repeat the task. Perform the task four times.

For the first task, two blocks with the same dimensions and look, but different weights, will be used. The heavy block is filled with lead and weights 1.14 kg, whereas the light block is empty and weights 0.047 kg. For every user, the order in which a block has to be pushed first is pre-determined by using a random generator. The participant does not know that the two blocks have different weights. At the start of the task the user needs to push an arcade button which starts the collection of data. Once the task is performed, he/she needs to push a second arcade button, which will stop the data collection.

For the second task, data collection is started once the “start corner” button is pressed, and data collection is stopped once one of the “corner buttons” is pressed. The order in which the LEDs light up is determined randomly to avoid that the participant may know the pattern.

The setup of both experiments can be seen in figure 5.1.



(a) Franka Emika Panda arm with the experiment setup and the recording webcam.

(b) Experiments with an indication of the buttons.

Figure 5.1: Setup of the two experiments, where (in figure b) 1 till 4 define the four corners which are randomly chosen once the “start corner” button is pushed. The “start block” button defines the button that has to be pushed before the block will be pushed over the indicated grey line. Afterwards, the “stop block” button needs to be pushed.

To analyse the perceived effectiveness and perceived efficiency for both controllers at least several variables should be measured within the experiments. These variables are:

- Elapsed time
- Impedance controller actions applied to the robotic arm

- endpoint force of the robotic arm, as a result of the impedance controller
- Position of the haptic master device along x, y and z
- Position of the robotic arm along x, y and z

Overall the experiment, per participant, is approximated to take ± 37 minutes. This is based on the designed protocol, which can be found in Appendix B. The time per participant (following from the protocol) does not include any added time due to unforeseen hardware malfunction. Therefore, the time reserved per participant is set to 1 hour. This gives an additional 23 minutes per participant.

6 Results

This chapter covers the results obtained from the user study. Firstly, the results of the questionnaires are discussed and substantiated with some performance results. Secondly, the performance results will be analysed further. The results are analysed such that they answer the following questions:

Questionnairre results:

- Is there a significant difference in time to task completion (TTC) between the two controllers?
- Do the different questionnaire scales show any significant difference between the two controllers?

Performance results:

- Does the designed controller work as expected, and do the results show any benefit?
- Can the deep muscles in the forearm be effectively measured by the use of sEMG? If not, what influence does this have?

6.1 Questionnaire results

6.1.1 TTC time

From the user study questionnaire answers, it follows that all participants were right-handed and did not encounter any physical limitations that could hinder them during the performance of the different tasks. The statistics that follow from the questionnaire can be found in table 6.1. Furthermore, Pearson's correlation coefficients ($\mathbf{r_1}$, till $\mathbf{r_4}$) are calculated for every question, and its corresponding scale, with respect to the time to task completion (TTC) for both controllers. The TTC is based on the time it takes to go from the “start corner” button (once pressed) to button “1”, “2”, “3”, or “4” and press it.

Table 6.1: Statistics for the paired t-test and correlation between the scales and task time, based on the used Likert scale (1 to 7). In this table PU defines the Perceived Usefulness, PEU the Perceived Ease of Use and C the Comfort.

Question	Scale	Controller	Mean	N	Std. Deviation	Std. Error Mean	r_1	r_2	r_3	r_4
Q7	PU	Posture co-contraction	2,14	14	0,66	0,18	-0,12	0,48	0,39	-0,07
		Single muscle pair co-contraction	2,71	14	1,49	0,40	0,34	-0,14	-0,11	-0,45
Q8	PU	Posture co-contraction	2,93	14	1,38	0,37	-0,03	0,51	0,44	-0,27
		Single muscle pair co-contraction	3,29	14	1,68	0,45	0,18	-0,27	-0,16	-0,59
Q9	PU	Posture co-contraction	2,93	14	1,54	0,41	0,10	0,46	0,31	-0,46
		Single muscle pair co-contraction	3,36	14	1,95	0,52	0,08	-0,46	-0,08	-0,27
Q10	PU	Posture co-contraction	2,86	14	1,46	0,39	-0,60	0,02	0,27	-0,23
		Single muscle pair co-contraction	3,21	14	1,58	0,42	0,21	-0,45	-0,21	-0,17
Q11	PEU	Posture co-contraction	2,64	14	1,39	0,37	0,02	0,47	0,42	-0,15
		Single muscle pair co-contraction	2,93	14	1,69	0,45	0,07	-0,22	-0,23	-0,44
Q12	PEU	Posture co-contraction	1,93	14	0,83	0,22	0,01	0,57	0,67	-0,0
		Single muscle pair co-contraction	2,07	14	1,00	0,27	0,04	-0,47	-0,29	-0,57
Q13	C	Posture co-contraction	5,36	14	1,39	0,37	-0,07	-0,31	-0,20	-0,31
		Single muscle pair co-contraction	4,07	14	1,59	0,43	-0,24	0,26	0,36	0,43
Q14	C	Posture co-contraction	4,21	14	1,85	0,49	-0,02	-0,12	-0,33	0,20
		Single muscle pair co-contraction	3,29	14	1,59	0,42	-0,16	0,14	0,46	0,10

The Pearson's correlation coefficients show that there is indeed some significant correlation between the given answers and the TTC. However, the answers do not always seem in line with the expectations. For example, the results of Q10 for the posture co-contraction controller shows a negative correlation coefficient of -0.60 ($\mathbf{r_1}$) while reaching and pressing button "1". This indicates that the more precise the controller feels, the longer it takes to complete the task, which is not expected. This mainly has to do with the fact that the users found it quite hard to use depth perception to find the exact location of the buttons with the 2-dimensional visual feedback coming from a webcam. This depth perception issue might become less relevant when the experiment is done for the second time. Therefore, boxplots are made for the posture co-contraction controller (figure 6.1), which show the TTC for the test subjects that used the posture co-contraction controller as the first controller and the test subjects that used the posture co-contraction controller as the second controller. The boxplots shown contain all the button experiments performed with the specific controller. The same boxplots can be made for the single muscle pair co-contraction controller, resulting in figure 6.2. The TTC for pushing the heavy and light block will not be analysed since some participants tend to get stuck between the "start block", "start corner" and "stop block" buttons (figure 5.1b), which influences the TTC a lot and is therefore hard to analyse.

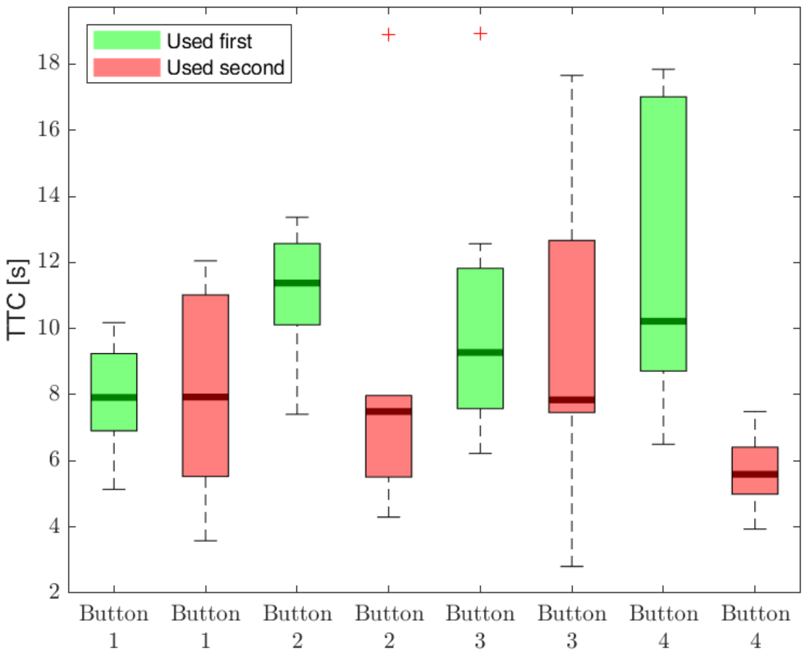


Figure 6.1: TTC boxplots for the button experiments while using the posture co-contraction controller. The results of the users that used the posture co-contraction controller as their first controller are shown in the green boxplots, and the users that used the posture co-contraction controller as their second controller are shown in the red boxplots.

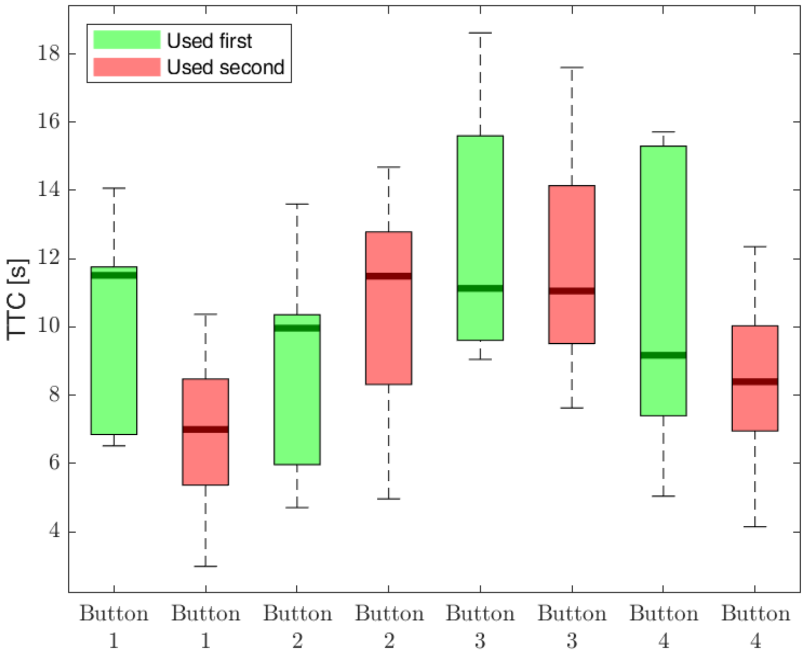


Figure 6.2: TTC boxplots for the button experiments while using the single muscle pair co-contraction controller. The results of the users that used the single muscle pair co-contraction controller as their first controller are shown in the green boxplots, and the users that used the single muscle pair co-contraction controller as their second controller are shown in the red boxplots.

From the posture co-contraction results in figure 6.1, in almost all cases, the median is lower or approximately the same once the controller is used as second. This indicates that the depth perception issue seems to decrease once the experiment is trained. The same holds for the single muscle pair co-contraction controller in figure 6.2, except button 2, which might be caused by a low overall stiffness (which will follow from figure 6.9) and hence the controller may feel less precise, which makes it harder to control.
Following the boxplots results, it is assumed that once a specific controller is used for the second time, the results are the most reliable. Doing so allows comparing the TTC for both controllers in the button experiment. This shows promising results for the newly designed posture co-contraction controller concerning the TTC if the median value is assumed. Although the TTC spread for some buttons is quite big (e.g. the third quartile of the green boxplot of button "4" in figure 6.2), it remains hard to draw any hard conclusions. More users are needed to give a more trustworthy median, which answers the first question.

6.1.2 Paired t-test to show significant difference

A paired t-test can be performed by using further information from the obtained statistics in table 6.1. Doing so results in table 6.2.

Table 6.2: Paired t-test, based on the used Likert scale (1 to 7).

			Paired differences			t	df	p
			Mean	Std. Deviation	Std. Error Mean			
Q7	Perceived Usefulness (PU)	Posture co-contraction - Single muscle pair co-contraction	-0.57	1.09	0.29	-1.96	13	0,071
Q8	Perceived Usefulness (PU)	Posture co-contraction - Single muscle pair co-contraction	-0.36	0.63	0.17	-2.11	13	0,055
Q9	Perceived Usefulness (PU)	Posture co-contraction - Single muscle pair co-contraction	-0.43	0.85	0.23	-1.89	13	0,082
Q10	Perceived Usefulness (PU)	Posture co-contraction - Single muscle pair co-contraction	-0.36	0.74	0.20	-1.79	13	0,096
Q11	Perceived Ease of Use (PEU)	Posture co-contraction - Single muscle pair co-contraction	-0.29	0.73	0.19	-1.47	13	0,165
Q12	Perceived Ease of Use (PEU)	Posture co-contraction - Single muscle pair co-contraction	-0.14	0.36	0.10	-1.47	13	0,165
Q13	Comfort (C)	Posture co-contraction - Single muscle pair co-contraction	1.29	1.14	0.30	4.22	13	0,001
Q14	Comfort (C)	Posture co-contraction - Single muscle pair co-contraction	0.93	0.83	0.22	4.19	13	0,001

From the results in table 6.1 and 6.2, it follows that only significant differences can be found in the Comfort scale ($p<0.05$). This shows that the posture co-contraction controller feels more comfortable during the task performance.

The questionnaire contains an additional question (Q15) which asks the participants to give at least three adjectives to describe the currently used controller. The obtained results are categorised into several adjectives and shown in figure 6.3. Some participants gave adjectives that fall under the same category and are therefore only counted once.
From the results in figure 6.3, it follows that for almost every positive adjective category, the newly designed posture co-contraction controller scored the highest. However, once the adjective "heavy" was chosen, it was noted that this was mainly in a specific direction. This was not the case for the single muscle pair co-contraction controller.

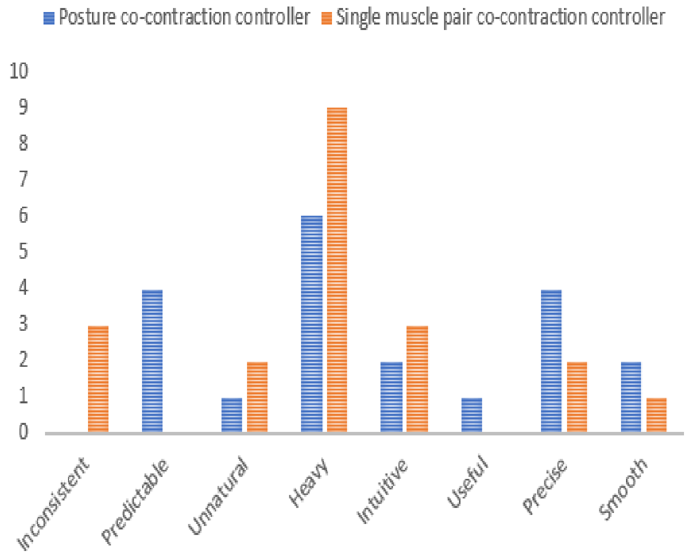


Figure 6.3: Results for the adjectives to describe the controllers during the user study.

Based on the questionnaire results, the answer to the second question would be that there is, according to the paired t-test, a significant difference in the comfort scale. However, subdividing the comfort scale into several adjectives does not always seem in favour of the newly designed controller.

6.2 Performance results

To analyse the performance of the designed controller, the end-effector position, stiffness, and force of the Franka Emika Panda slave arm are compared with the already existing controller. Furthermore, the error between the position of the haptic device and the slave arm are compared as well. The results for all users can be found in Appendices H till M. Appendices H till I cover the results obtained for both controllers once the “start corner” button is pressed, the path towards “button 1” is taken and “button 1” is pressed. The results for pushing the heavy block and light block (for both controllers) can be found in Appendices J till K, and L till M, respectively.

6.2.1 Performance results of all users

In this section, the performance results of the endpoint stiffness are analysed for all users. This is done by using boxplots that contain the endpoint stiffnesses of all participants for the different experiments. Firstly, the endpoint stiffness of the newly designed posture co-contraction controller will be analysed. Secondly, the single muscle pair co-contraction controller results are analysed and compared with the newly designed controller.

Posture co-contraction controller

The endpoint stiffness results for the posture co-contraction controller in x,y and z-direction, based on all participants for the different experiments, can be found in the boxplots of figure 6.4 till 6.6, respectively. These boxplots clearly show a considerable difference per participant regarding their arm endpoint stiffness. This is as expected due to the calibration phase, where the top and bottom muscle activations are determined. Some participants were able to withstand a higher applied force by the Virtuose but could not apply the same amount of force during the experiments simply because the robotic arm did not need an excessive amount of force to overcome its own weight/friction. So participants who had a high upper activation limit for the muscles (due to their excellent withstanding of the force in the calibration phase) eventually delivered less stiffness to the robotic arm since they would never reach such an excessive muscle activation again. The opposite is true for the participants with a lower upper activation limit. They were able to deliver a higher endpoint stiffness to the robotic arm. The boxplots (especially the x-directional stiffness boxplot in figure 6.4) clearly show that quartile one and quartile three have a huge difference from the median. However, this difference is influenced by the human arm posture as well. This becomes clear by looking at the boxplots for the muscle activations (figures 6.7 and 6.8). These boxplots show the muscle activations of all participants during the "button" and "block" experiments for the posture co-contraction controller. As can be seen, the quartiles one and three do not have such an excessive deviation as figure 6.4 till 6.6 suggests. Although, a combination of all these muscles, together with the configuration of the human arm, eventually will. Hence, the difference in endpoint stiffness is different per user and strongly dependent on the calibration. Furthermore, moving the robotic arm in the y-direction (in figure 5.1b, this is from the "start corner" button towards the side of buttons "1" and "4", or from the "start corner" button towards the side of buttons "2" and "3", respectively) was experienced as a less resistive movement than moving in the x-direction. The boxplots in figure 6.5 show this as well since the overall stiffness delivered to the robotic arm is way less than in the x-direction. The higher stiffness in the z-direction (observed in figure 6.6) can be explained by the weight of the robotic arm that has to be overcome once its endpoint is moving upwards.

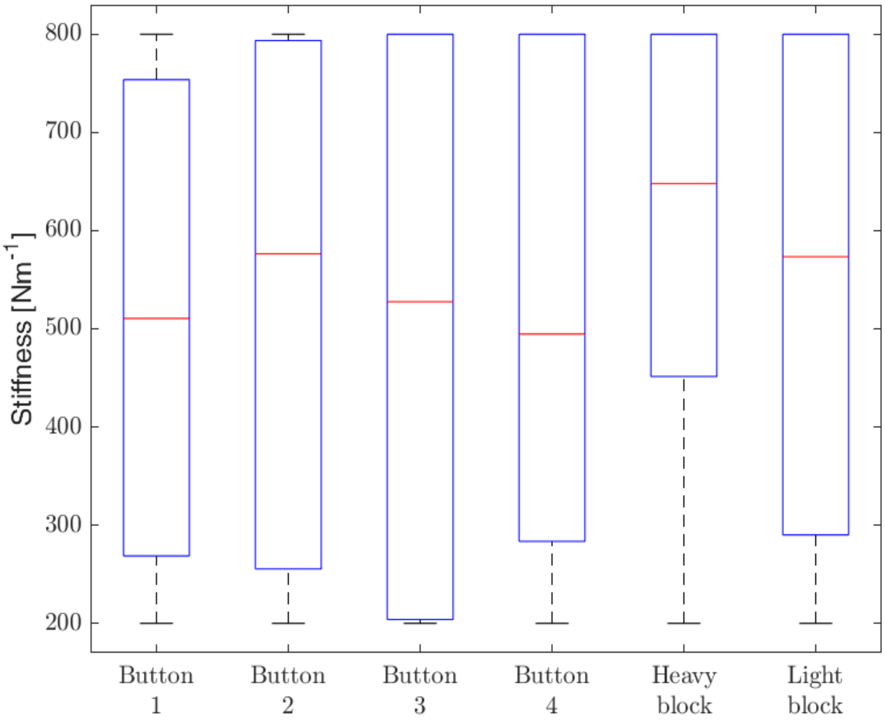


Figure 6.4: Boxplots of the posture co-contraction controller's x-directional stiffness, based on the stiffnesses of all users.

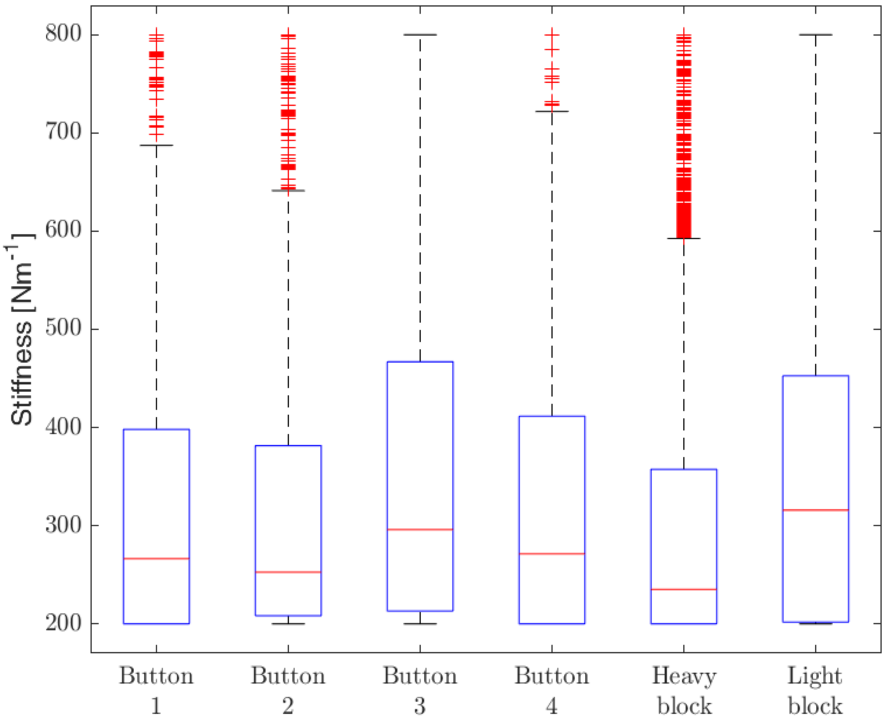


Figure 6.5: Boxplots of the posture co-contraction controller's y-directional stiffness, based on the stiffnesses of all users.

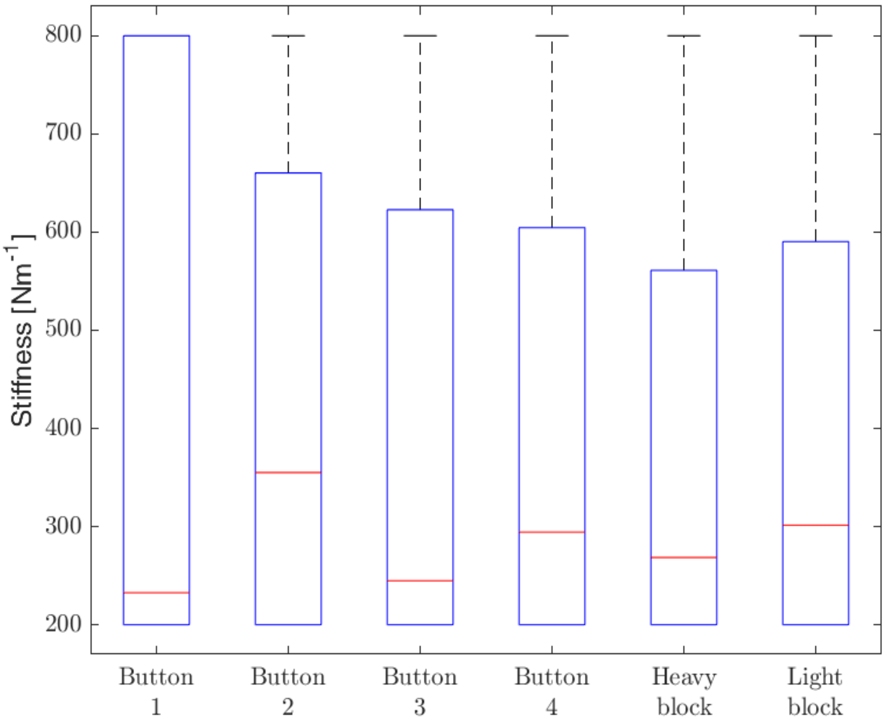


Figure 6.6: Boxplots of the posture co-contraction controller's z-directional stiffness, based on the stiffnesses of all users.

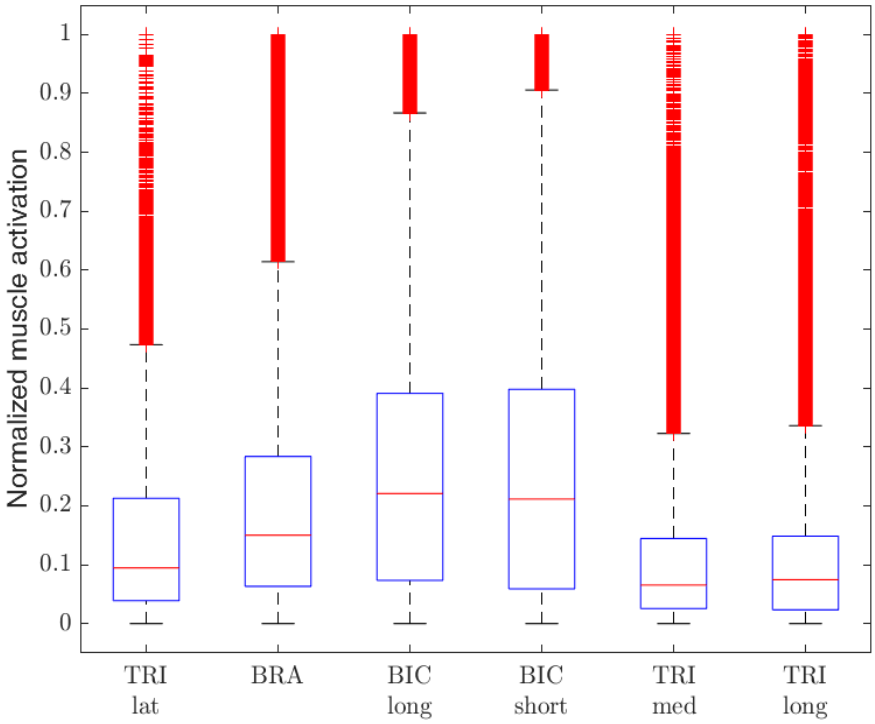


Figure 6.7: Boxplots of the muscle activations of the arm while using the posture co-contraction controller. The data is based on the muscle activations of all participants.

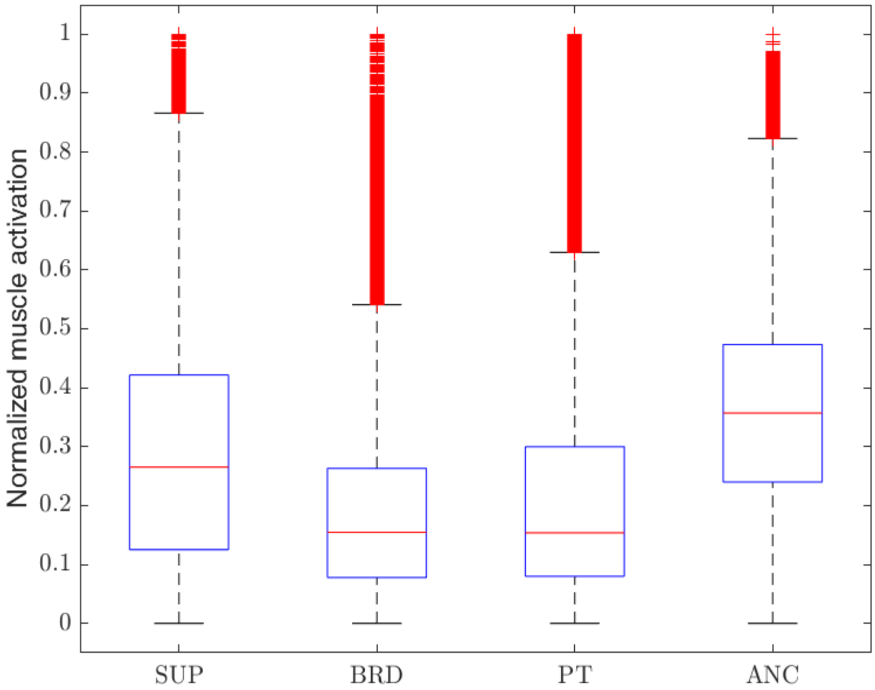


Figure 6.8: Boxplots of the muscle activations of the forearm while using the posture co-contraction controller. The data is based on the muscle activations of all participants.

Single muscle pair co-contraction controller

The stiffness is the same for the single muscle pair co-contraction controller in x,y and z-directions. The boxplot in figure 6.9 shows the results based on the data of all users. It can clearly be seen that the median is way lower than for the posture co-contraction controller, especially in the x-direction, and that the deviation in endpoint stiffness is way less between the different users (e.g. quartile one and quartile three).

The same upper and lower limits for the muscle activations, resulting from the calibration, are used for the single muscle pair co-contraction controller. However, the endpoint stiffness is now entirely dependent on the muscle activations. Figure 6.10 shows a boxplot of the muscle activations during all the experiments. From this figure, it can be seen the brachioradialis muscle (BRD) has a very low median. The single co-contraction controller compares the brachioradialis (BRD) and anconeus (ANC) muscle activations and takes the lowest value to ensure that both muscles are activated equally such that the human arm will not apply an external force. From the results of figure 6.9, this would mean that the linear scaling of the brachioradialis muscle mainly determines the endpoint stiffness.

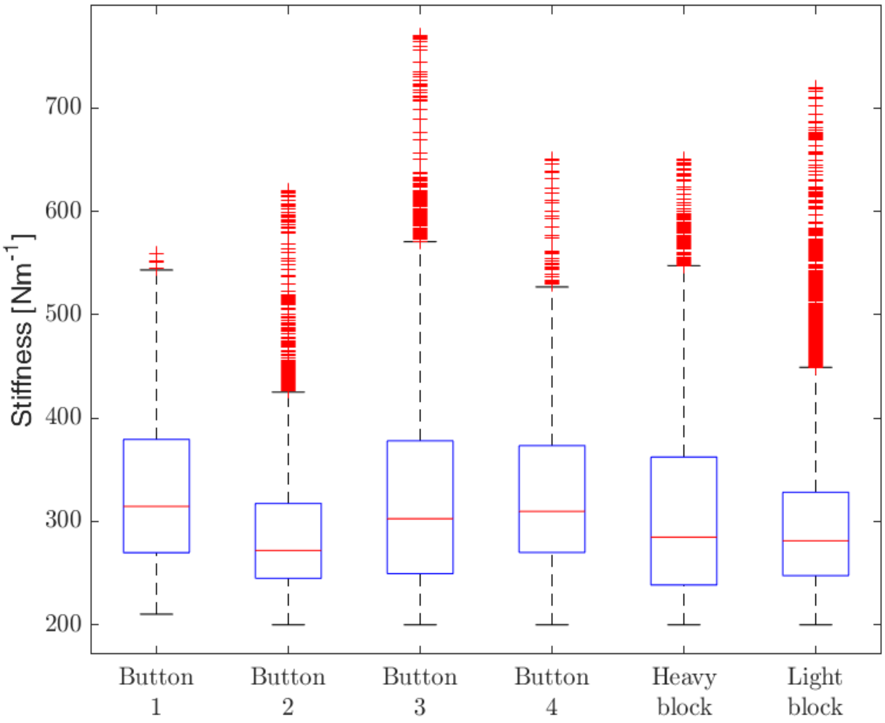


Figure 6.9: Boxplots of the single muscle pair co-contraction controller's three dimensional stiffness, based on the stiffnesses of all users.

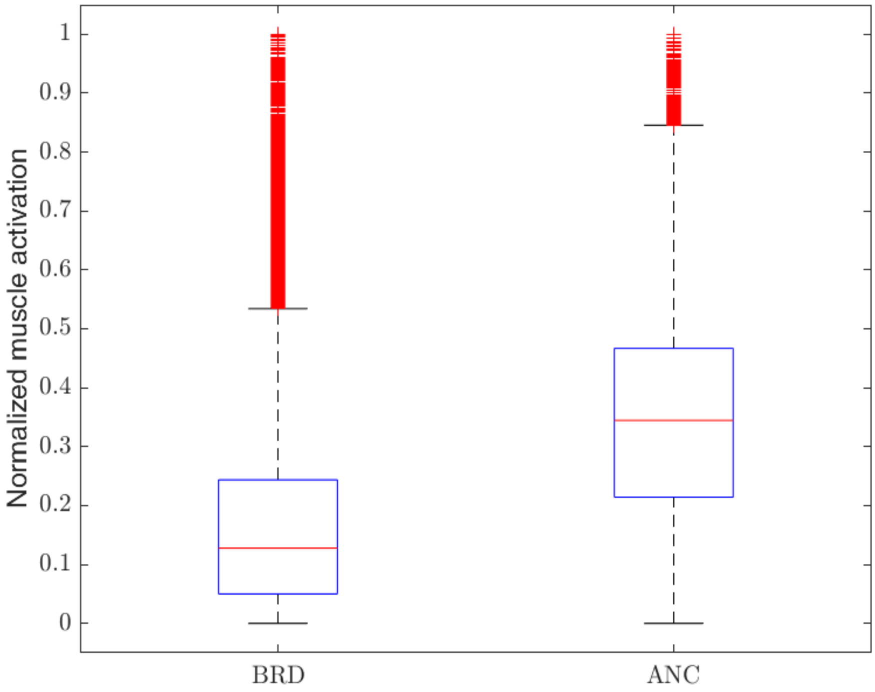


Figure 6.10: Boxplots of the muscle activations of the forearm while using the single muscle pair co-contraction controller. The data is based on the muscle activations of all participants.

For the first performance question, based on the results of all users, it can be said that the results from the posture co-contraction controller and the single muscle pair co-contraction controller are both strongly dependent on the calibration procedure. However, the posture co-contraction controller endpoint stiffness range is bigger due to the contribution of the human arm configuration and more muscles.

6.2.2 Performance results single user

To make the results more clear the results for one user are published here. User 6 is used during this analyses, and his/her results for the tasks reaching corner one and pushing the heavy and light blocks can be found in figures 6.11 till 6.13, respectively.

From the first row of figure 6.11, it can be seen clearly that the end-effector force follows from the position deviation between the slave arm and the haptic device (error). For the posture co-contraction controller, the force in z-direction is the same as the error in the z-direction but multiplied by the continuous stiffness of 800 $\mathrm{N\,m^{-1}}$. This shows that the stiffness following from the human posture and co-contraction of the muscles indeed influences the end-effector force. The same holds for the $x-$ and $y-$values but are slightly harder to observe.

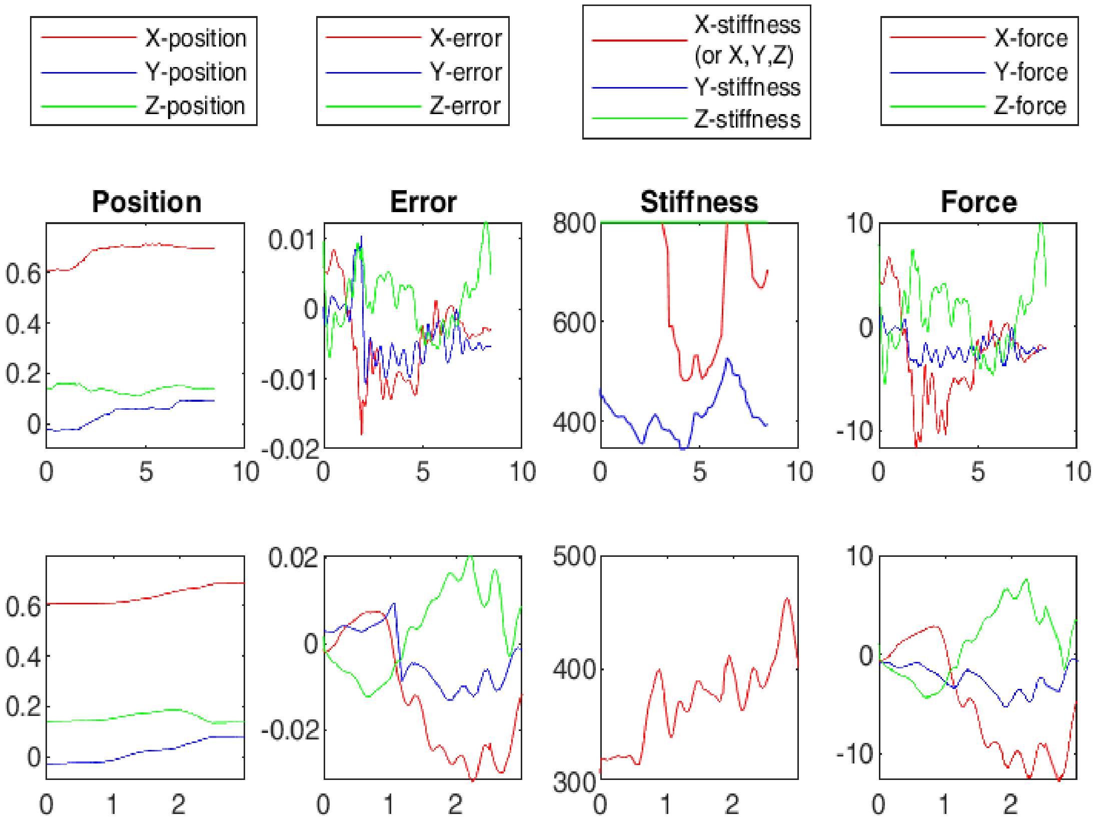


Figure 6.11: Position, position error, stiffness and force plots for the posture muscle co-contraction (top row) and the single muscle pair co-contraction controllers (bottom row). The $x-$axis values show the time taken to perform the task, and the $y-$axis shows the position, error, stiffness and force values in their SI units. The results are obtained by moving towards push-button one and pressing the button.

The path along which the blocks have to be pushed is in the $x-$direction, as can be observed from the position plots in figure 6.12 and 6.13. The position at 0 s indicates the position of the "start block" button, and the position at the last timestamp indicates the position of the "stop block" button. This results in the returning $y-$direction pattern.
The blocks have to be pushed along the $x-$direction. Therefore, it is expected that a higher stiffness is generated in this direction as well. To overcome the weight of the blocks a higher force needs to be applied. This force is generated by either a larger error or a higher stiffness. Since the participant is applying a precise control movement, his/her velocity is slow, resulting in a small error. Therefore, this automatically requires an increase in end-effector stiffness. The $x-$directional stiffness in figures in figures 6.12 and 6.13 (row 1 column 3) show this as well. Furthermore, it can be seen that the weight of the blocks influences the $x-$directional stiffness. The same holds for any obstacles that are encountered. This follows from the steep line from $x-$directional stiffness in figure 6.13 where the light block was pushed against the "stop block" button.

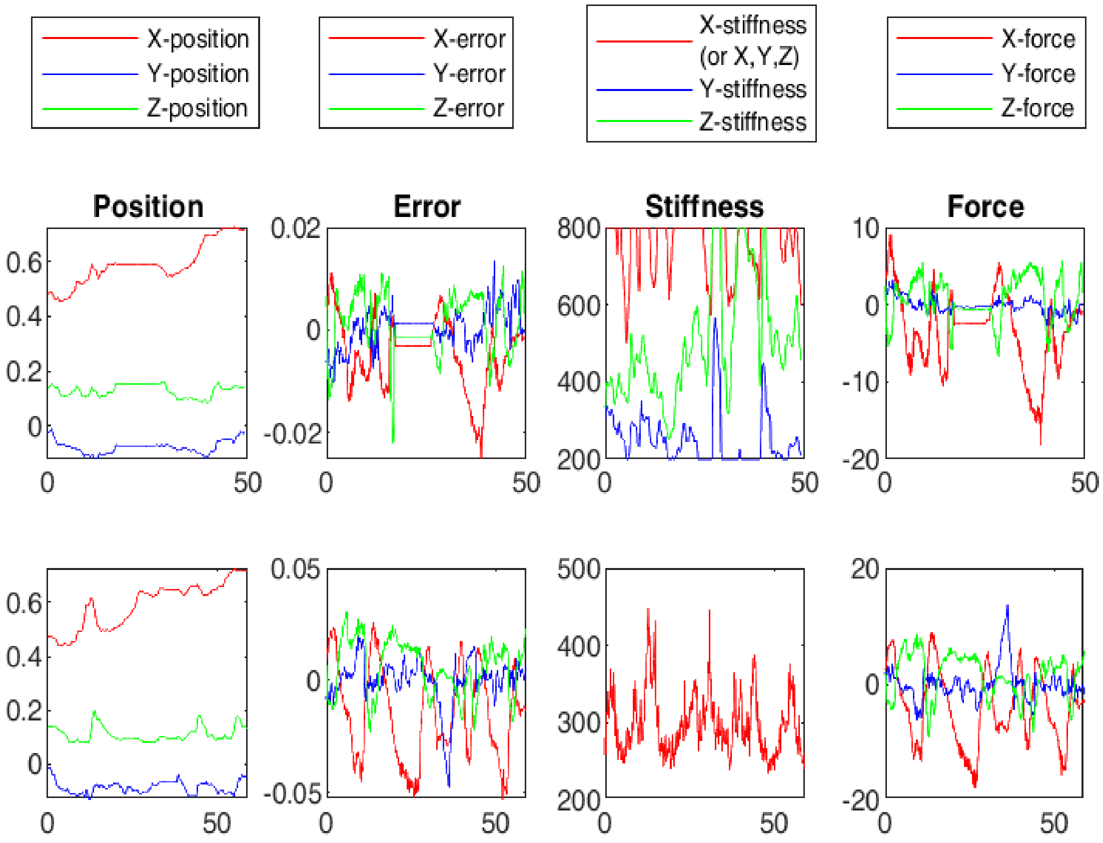


Figure 6.12: Position, position error, stiffness and force plots for the posture muscle co-contraction (top row) and the single muscle pair co-contraction controllers (bottom row). The $x-$axis values show the time taken to perform the task, and the $y-$axis shows the position, error, stiffness and force values in their SI units. The results are obtained during the pushing of the heavy block task.

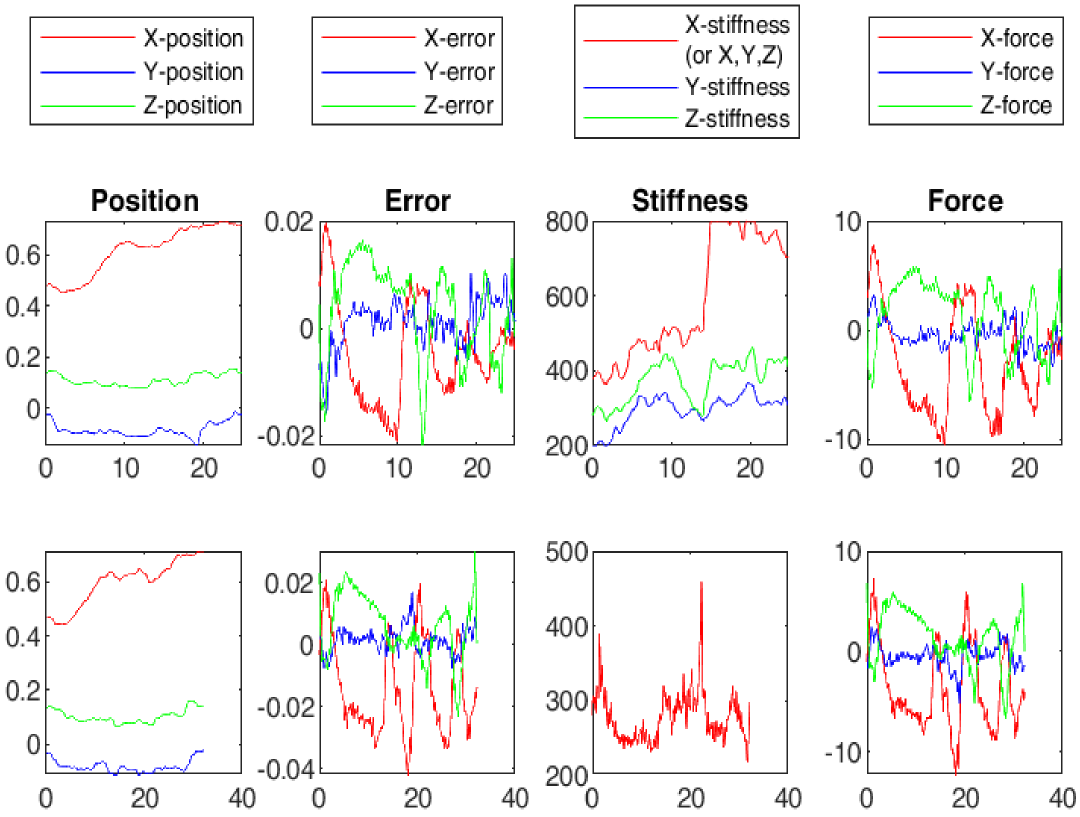


Figure 6.13: Position, position error, stiffness and force plots for the posture muscle co-contraction (top row) and the single muscle pair co-contraction controllers (bottom row). The $x-$axis values show the time taken to perform the task, and the $y-$axis shows the position, error, stiffness and force values in their SI units. The results are obtained during the pushing of the light block task.

The results for this specific user show that the designed posture muscle co-contraction controller has a clear benefit in increasing the end-effector stiffness in the desired direction. Furthermore, this specific user was also able to sense the different weights of the blocks. However, this did not hold for all of the participants. This became very clear from combining the questionnaire answers with the actual performance, as can be seen in figure 6.14 and 6.15. These divergent results are mainly due to the upper and lower limit of the muscle activations, as discussed earlier, and the slightly different arm orientation of the participants while controlling the haptic device. For example, although the starting point of the haptic device was the same for every participant, they can still orientate their elbow differently during the controlling phase. Furthermore, the different arm lengths of the participants cause a slightly different arm orientation if the starting point remains the same for every user. Also, a somewhat different arm posture during the inertial frames for the orientation sensor turns out to have a significant influence on the overall endpoint stiffness. Although all the participants were ordered to obtain the same posture during the inertial frame calibration, slight orientation differences are hard to overcome.

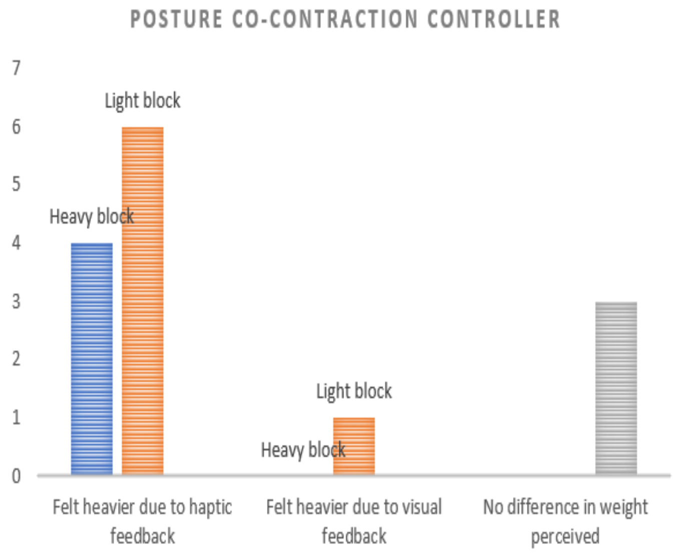


Figure 6.14: Perceived weights of the used blocks by the participants for the posture muscle co-contraction controller.

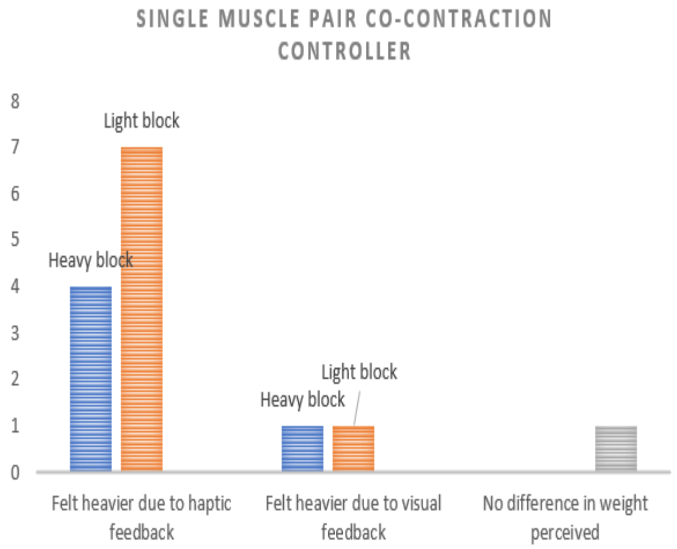


Figure 6.15: Perceived weights of the used blocks by the participants for the single muscle pair co-contraction controller.

6.2.3 Measuring deep muscles

Several normalized EMG activations are plotted in figure 6.16 to see whether the supinator (SUP) muscle, located deep in the forearm, can effectively be measured. These results are based on participant 6 while doing the “button 1” experiment. According to figure 4.3, the supinator muscle ought to be measured with electrode number three or four. To avoid possible interference with the brachioradialis (BRD) muscle, measured by electrode number two, the controller uses the 4th electrode to determine the supinator activations instead of the 3rd electrode.

The developed controller is not based on wrist flexion and extension, but takes the wrist as endpoint. Therefore, the hand is considered not to move with respect to the endpoint. Hence, it is expected that the activation of the supinator outperforms the activation of the muscles that cause extension at the wrist (the shallow muscles covered by electrodes three and four). If this is the case, the measured EMG signals should be approximately the same for electrode three and electrode four. Looking at figure 6.16, it can be seen that this is not the case. This is mainly due to participants applying a firm grip on the Virtuose gripper to keep their wrist aligned with the forearm, resulting in high activations of the shallow muscles. Therefore, the supinator muscle is not effectively measured. In this experiment, this is not of great importance since the supinator muscle and pronator teres (PT) muscle are used to determine the rotational stiffness in the endpoint, which is not used. However, if it is desired to implement rotational stiffnesses, the surface EMG methodology does not seem the way to go.

So, the answer to the second performance question would be that sEMG can not be used to effectively measure the deep muscles in the forearm. However, this does not seem to be a problem with the current tasks that only require mimicking the translational endpoint stiffnesses.

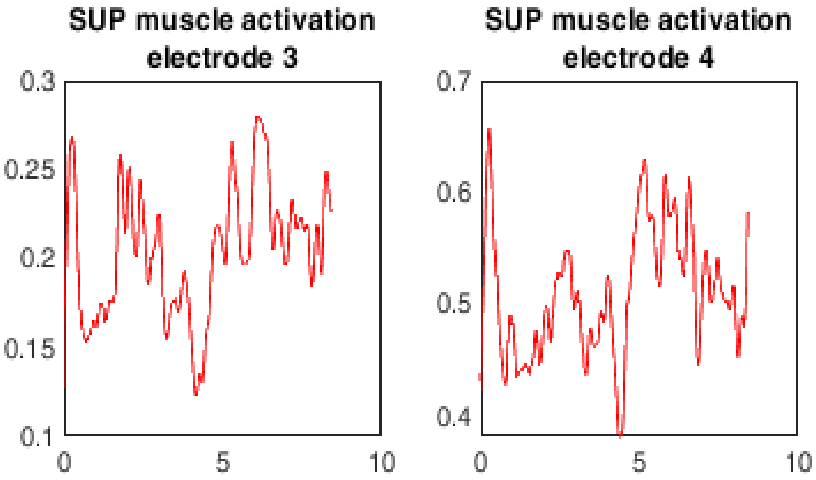


Figure 6.16: Possible muscle activations of the supinator (SUP) muscle while reaching button "1". The y-axis defines the normalized muscle activation, and the x-axis the TTC [s].

7 Conclusions

This thesis focused on implementing a control method that is able to obtain a more realistic and intuitive impedance reflection towards the slave site within telemanipulation control. A big downside to the existing controller based on the co-contraction of a single muscle pair was that it applied the same end-effector stiffness in x, y and $z-$direction. Furthermore, the measured EMG signals were hardly able to reach their upper limits measured during calibration. This caused the result that rigid objects were hard to sense due to an overall low stiffness profile.

To overcome these problems a new control method is analysed, designed and observed. This control method focuses on the contraction of several muscles in the entire arm. These contractions lead to different joint stiffnesses. This, together with the human arm's posture, eventually leads to an endpoint stiffness of the human arm. This endpoint stiffness differs in x, y and $z-$direction and is scaled such that it can be applied to the slave site.

The results show that the participants favour the newly designed control method based on comfort, predictability, precision and consistency. This also follows from the decrease in time to task completion (TTC). However, this might not be fully reliable due to the small number of participants. Furthermore, the new control can increase the endpoint stiffness in a direction corresponding to the task and is able to reach the upper limits of its defined end-effector stiffness limit.

However, in its current form, there are still too many inconsistencies. The results per participant are very divergent. This has mainly to do with the calibration phase to determine the upper and lower limits of the muscle activations. Furthermore, the controller is very sensitive to different arm orientations. This orientation sensitivity also follows once the inertial frames of the orientation sensors are recalibrated. This influences the endpoint stiffness, which should not be the case once the posture of the arm during recalibration is the same. These issues have to be resolved before the current method can be used with consistent results. At last, if it is desired to determine rotational stiffness of the human arm endpoint as well, it would be desired to evaluate other methods that do not rely on surface EMG measurements.

8 Recommendations

In this section, some future improvements and shortcomings of the designed controller will be discussed.

Currently, the designed controller is entirely focused on the use of the right arm. This is due to the OpenSim model that is implemented, see 4.1, to calculate the muscle moment arms. To resolve this a model has to be implemented, which includes muscle moment arm calculations for the muscles in the left arm.

Furthermore, the weight and orientation of the slave robot have a significant influence on the force feedback that the user feels and, therefore, a significant influence on perceiving the environment. This can be compensated for by, for instance, not setting the virtual mass of the human arm equal to the inertia of the robot's arm end-effector ($\Lambda \neq \Lambda_{\mathbf{d}}$). However, this requires constant reshaping of the robotics arm inertia and the mass of the human arm. Another method can be the bilateral impedance reflection principle from [15]. However, this bilateral impedance reflection does not behave well under time delays bigger than 50 ms since this resulted in instabilities in the system due to the simplicity of the trajectory predictor. A possible solution to this problem might be a trajectory prediction on the user site. A trajectory prediction based on the optimal control principle of the human arm might be worthwhile in this case. A thoroughly worked out principle has already been done in Appendix A, but due to time restrictions not implemented.

Appendices

A Trajectory prediction

Within the work of [15] a simple linear trajectory predictor is used, based on velocity and time delay, to compensate for the time delay in the communication channel. The time delay works in both ways. This requires a prediction for the endpoint position of the human arm at the robot site, and a prediction of the end-effector of the robot arm at the operator site.

From [43, 44, 45, 46, 47] it follows that the human arm always follows the most cost-efficient way towards the desired point. Therefore, to get a good prediction of the human arm endpoint position, the prediction should be based on the natural movement of the human arm. The robot position controller can be adopted as desired and does not necessarily need the same cost-effective behaviour as the robot arm. Therefore, the linear prediction for the robot arm trajectory can remain the same.

A.1 Trajectory prediction of the robot arm

According to [15] the simple linear trajectory predictor is as follows:

$$x_{R,p} = x_R + \left(\dot{x}_R \cdot t_{\text{delay}}\right) \tag{8.1}$$

Where, $x_{R,p}$ is the predicted position, x_R the current position and $\dot{x}_R$ the velocity of the end-effector of the robot arm. The amount of time the trajectory needs to be predicted depends on the delay in the communication channel t_{delay}.

A.2 Trajectory prediction of the human arm

In this section, a model of the endpoint movement of the human arm will be discussed. As an example, the model will be used in a reaching movement.

A.2.1 Human arm dynamics

The dynamics which follow from the CNS can be described by a state-space representation [32]. This gives:

$$\dot{\mathbf{x}}(t) = \mathbf{A_c}\mathbf{x}(t) + \mathbf{B_c}\mathbf{u}(t) \tag{8.2}$$

$$\mathbf{y}(t) = \mathbf{C_c}\mathbf{x}(t) \tag{8.3}$$

Where,

- $\dot{\mathbf{x}}(t) = \frac{d}{dt}\mathbf{x}(t)$
- $\mathbf{x}(t)$ is the state vector containing the position ($\mathbf{p}(t)$), velocity ($\mathbf{v}(t)$), muscle force ($\mathbf{f}_{\text{muscle}}(t)$) and goal position ($\mathbf{g}$) of the endpoint in 3D
- $\mathbf{A_c}$ is the system matrix (internal model of the human in this case)
- $\mathbf{B_c}$ is the input matrix
- $\mathbf{u}(t)$ is the motor command (muscle excitation)
- $\mathbf{y}(t)$ is the output vector

- $\mathbf{C_c}$ is the output matrix

To determine the matrices $\mathbf{A_c}$, $\mathbf{B_c}$ and $\mathbf{C_c}$ the following approach from subsection A.2.2 till A.2.2 is taken.

A.2.2 Determine the individual elements of the derivative of the state vector

Position derivatives
The state vector $\mathbf{x}(t)$ contains positions and velocities. The position within the derivative of the state vector is therefore simply equal to the translational velocity in the state vector itself. Hence, equation 8.4 holds.

$$\dot{p}^{(i)}(t) = v^{(i)}(t) \tag{8.4}$$

This results in the matrix representation for the translational velocity as:

$$\begin{bmatrix} \dot{p}^{(x)}(t) \\ \dot{p}^{(y)}(t) \\ \dot{p}^{(z)}(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v^{(x)}(t) \\ v^{(y)}(t) \\ v^{(z)}(t) \end{bmatrix} \tag{8.5}$$

Determining the velocity derivatives
According to Newton's second law, $\sum F = m \cdot a$, the following holds:

$$\mathbf{f}_{\text{muscle}}(t) + \mathbf{f}_{\text{external}}(t) = \mathbf{M}\dot{\mathbf{v}}(t) \tag{8.6}$$

Where,

- $\mathbf{f}_{\text{external}}(t)$ is an external, velocity dependent, random force field
- $\mathbf{M}$ is a mass matrix containing the mass of the human arm endpoint in x, y and z direction
- $\dot{\mathbf{v}}(t)$ is the acceleration of the human arm endpoint

Rewriting 8.6 towards an expression of the velocity derivative results in:

$$\dot{\mathbf{v}}(t) = \mathbf{M}^{-1}\mathbf{f}_{\text{muscle}}(t) + \mathbf{M}^{-1}\mathbf{f}_{\text{external}}(t) \tag{8.7}$$

With $\mathbf{M}^{-1}$ the inverse of the mass matrix $\mathbf{M}$, hence:

$$\mathbf{M}^{-1} = \begin{bmatrix} \frac{1}{m_x} & 0 & 0 \\ 0 & \frac{1}{m_y} & 0 \\ 0 & 0 & \frac{1}{m_z} \end{bmatrix} \tag{8.8}$$

The external, velocity dependent, random force field can be determined by:

$$\mathbf{f}_{\text{external}}(t) = \mathbf{D}(t)\mathbf{v}(t) \tag{8.9}$$

$$\begin{bmatrix} f^{(x)}_{\text{external}}(t) \\ f^{(y)}_{\text{external}}(t) \\ f^{(z)}_{\text{external}}(t) \end{bmatrix} = \begin{bmatrix} d_{11}(t) & d_{12}(t) & d_{13}(t) \\ d_{21}(t) & d_{22}(t) & d_{23}(t) \\ d_{31}(t) & d_{32}(t) & d_{33}(t) \end{bmatrix} \begin{bmatrix} v^{(x)}(t) \\ v^{(y)}(t) \\ v^{(z)}(t) \end{bmatrix} \tag{8.10}$$

To make interpretation of the results in a later stage easier it is chosen that the matrix entries d_{11} till d_{33} are random values which maximum and minimum magnitudes can be set as desired. Furthermore, the amount of time the force needs to stay constant can be set as well. Therefore the time dependency of the elements.

The velocity derivative now follows from 8.7, 8.8 and 8.10 as:

$$\begin{bmatrix} \dot{v}^{(x)}(t) \\ \dot{v}^{(y)}(t) \\ \dot{v}^{(z)}(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{m_x} & 0 & 0 \\ 0 & \frac{1}{m_y} & 0 \\ 0 & 0 & \frac{1}{m_z} \end{bmatrix} \begin{bmatrix} f^{(x)}_{\text{muscle}}(t) \\ f^{(y)}_{\text{muscle}}(t) \\ f^{(z)}_{\text{muscle}}(t) \end{bmatrix} + \begin{bmatrix} \frac{d_{11}(t)}{m_x} & \frac{d_{12}(t)}{m_x} & \frac{d_{13}(t)}{m_x} \\ \frac{d_{21}(t)}{m_y} & \frac{d_{22}(t)}{m_y} & \frac{d_{23}(t)}{m_y} \\ \frac{d_{31}(t)}{m_z} & \frac{d_{32}(t)}{m_z} & \frac{d_{33}(t)}{m_z} \end{bmatrix} \begin{bmatrix} v^{(x)}(t) \\ v^{(y)}(t) \\ v^{(z)}(t) \end{bmatrix} \tag{8.11}$$

derivatives of the muscle force for x, y and z

Muscle force can be modelled as a process that consists of two stages. These stages separate the activation dynamics from the contraction dynamics of the muscle. Furthermore, the activation of the muscle can be associated with the isometric muscle force[28].

According to [48] the muscle activation can be related by 8.12. Where $a(t)$ is the activation signal, $u(t)$ is the excitation signal and τ_u the time constant for activation or deactivation of the muscle. From [48] it follows that the time constant for activation can reach a maximum of ≈10 ms and the time constant for deactivation a maximum of ≈40ms. These values are greater for muscles with slow muscle fibers and larger size. This makes the time constant hard to determine. Therefore, the time constant will be based on the time constant that follows from the deactivation of elbow flexion and extension. This time constant follows from the previous research of [49] and turns out to be 30 ms.

$$\frac{da(t)}{dt} = \frac{u(t) - a(t)}{\tau(a(t), u(t))} \tag{8.12}$$

Since the activation of the muscle and isometric muscle force are related [28] the isometric muscle force can be described as:

$$\dot{f}^{(i)}_{\text{muscle}}(t) = \frac{u^{(i)}(t) - f^{(i)}_{\text{muscle}}(t)}{\tau_u} \tag{8.13}$$

Where,

- (i) represents the x, y or z component
- $\dot{f}^{(i)}_{\text{muscle}}(t) = \frac{d}{dt} f^{(i)}_{\text{muscle}}(t)$
- τ_u is the muscle time constant for activation/deactivation of the muscle, based on [49]

According to 8.13 this gives the following expression for the different elements:

$$\begin{bmatrix} \dot{f}^{(x)}_{\text{muscle}}(t) \\ \dot{f}^{(y)}_{\text{muscle}}(t) \\ \dot{f}^{(z)}_{\text{muscle}}(t) \end{bmatrix} = \begin{bmatrix} \frac{-1}{\tau_u} & 0 & 0 \\ 0 & \frac{-1}{\tau_u} & 0 \\ 0 & 0 & \frac{-1}{\tau_u} \end{bmatrix} \begin{bmatrix} f^{(x)}_{\text{muscle}}(t) \\ f^{(y)}_{\text{muscle}}(t) \\ f^{(z)}_{\text{muscle}}(t) \end{bmatrix} + \begin{bmatrix} \frac{1}{\tau_u} & 0 & 0 \\ 0 & \frac{1}{\tau_u} & 0 \\ 0 & 0 & \frac{1}{\tau_u} \end{bmatrix} \begin{bmatrix} u^{(x)}(t) \\ u^{(y)}(t) \\ u^{(z)}(t) \end{bmatrix} \tag{8.14}$$

Goal location derivative

For testing purposes it is assumed that the goal location is a known and steady point within the xyz-plane. The derivative of a steady point is zero. Therefore:

$$\begin{bmatrix} \dot{g}^{(x)} \\ \dot{g}^{(y)} \\ \dot{g}^{(z)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{8.15}$$

Resulting internal model, input and output matrices

Combining the determined expressions in 8.5, 8.11, 8.14 and 8.15 give rise to the continuous time state space model in 8.16 and 8.17

$$\underbrace{\begin{bmatrix} \dot{p}^{(x)}(t) \\ \dot{p}^{(y)}(t) \\ \dot{p}^{(z)}(t) \\ \dot{v}^{(x)}(t) \\ \dot{v}^{(y)}(t) \\ \dot{v}^{(z)}(t) \\ \dot{f}^{(x)}_{\text{muscle}}(t) \\ \dot{f}^{(y)}_{\text{muscle}}(t) \\ \dot{f}^{(z)}_{\text{muscle}}(t) \\ \dot{g}^{(x)} \\ \dot{g}^{(y)} \\ \dot{g}^{(z)} \end{bmatrix}}_{\dot{\mathbf{x}}(t)} = \underbrace{\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{d_{11}(t)}{m_x} & \frac{d_{12}(t)}{m_x} & \frac{d_{13}(t)}{m_x} & \frac{1}{m_x} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{d_{21}(t)}{m_y} & \frac{d_{22}(t)}{m_y} & \frac{d_{23}(t)}{m_y} & 0 & \frac{1}{m_y} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{d_{31}(t)}{m_z} & \frac{d_{32}(t)}{m_z} & \frac{d_{33}(t)}{m_z} & 0 & 0 & \frac{1}{m_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_u} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_u} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{-1}{\tau_u} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{A_c}} \underbrace{\begin{bmatrix} p^{(x)}(t) \\ p^{(y)}(t) \\ p^{(z)}(t) \\ v^{(x)}(t) \\ v^{(y)}(t) \\ v^{(z)}(t) \\ f^{(x)}_{\text{muscle}}(t) \\ f^{(y)}_{\text{muscle}}(t) \\ f^{(z)}_{\text{muscle}}(t) \\ g^{(x)} \\ g^{(y)} \\ g^{(z)} \end{bmatrix}}_{\mathbf{x}(t)} + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{\tau_u} & 0 & 0 \\ 0 & \frac{1}{\tau_u} & 0 \\ 0 & 0 & \frac{1}{\tau_u} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{\mathbf{B_c}} \underbrace{\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix}}_{\mathbf{u}(t)} \tag{8.16}$$

The output vector $\mathbf{y}(t)$ is equal to the state vector $\mathbf{x}(t)$. This will be the case if it is assumed that the position, velocity and forces are fully observable. For testing purposes, the goal location is a steady and known point in space. Therefore, it will not need to be observable yet. Since every entry of the state vector is observable or already known, the output matrix will be an identity matrix. Hence, $\mathbf{C_c} = I_{12}$.

$$\underbrace{\begin{bmatrix} p^{(x)}(t) \\ p^{(y)}(t) \\ p^{(z)}(t) \\ v^{(x)}(t) \\ v^{(y)}(t) \\ v^{(z)}(t) \\ f^{(x)}_{\text{muscle}}(t) \\ f^{(y)}_{\text{muscle}}(t) \\ f^{(z)}_{\text{muscle}}(t) \\ g^{(x)} \\ g^{(y)} \\ g^{(z)} \end{bmatrix}}_{\mathbf{y}(t)} = \underbrace{\begin{bmatrix} 1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \end{bmatrix}}_{\mathbf{C_c}} \underbrace{\begin{bmatrix} p^{(x)}(t) \\ p^{(y)}(t) \\ p^{(z)}(t) \\ v^{(x)}(t) \\ v^{(y)}(t) \\ v^{(z)}(t) \\ f^{(x)}_{\text{muscle}}(t) \\ f^{(y)}_{\text{muscle}}(t) \\ f^{(z)}_{\text{muscle}}(t) \\ g^{(x)} \\ g^{(y)} \\ g^{(z)} \end{bmatrix}}_{\mathbf{x}(t)} \tag{8.17}$$

A.2.3 Continuous to discrete time

Several methods to make a discrete-time approximation can be used. The forward Euler method will suffice once a small-time step is taken. The forward Euler follows from 8.18.

$$\dot{\mathbf{x}}(t) = \frac{\mathbf{x}(t+\Delta t) - \mathbf{x}(t)}{\Delta t} \tag{8.18}$$

Assuming that $t = k\Delta t$ and $(t+\Delta t) = (k+1)\Delta t$ then gives:

$$\mathbf{x}(t+\Delta t) = \dot{\mathbf{x}}(t)\Delta t + \mathbf{x}(t) \tag{8.19}$$

$$= (\mathbf{I} + \mathbf{A}_c\Delta t)\,\mathbf{x}(t) + (\mathbf{B}_c\Delta t)\,\mathbf{u}(t) \tag{8.20}$$

Hence,

$$\mathbf{x}_{k+1} = (\mathbf{I} + \mathbf{A}_c\Delta t)\,\mathbf{x}_k + (\mathbf{B}_c\Delta t)\,\mathbf{u}_k \tag{8.21}$$

$$= \mathbf{A_d}\mathbf{x}_k + \mathbf{B_d}\mathbf{u}_k \tag{8.22}$$

$$\mathbf{y}_k = \mathbf{C}_c\mathbf{x}_k \tag{8.23}$$

$$= \mathbf{C_d}\mathbf{x}_k \tag{8.24}$$

Following from 8.21 and 8.23 the discretized state matrix, input matrix and output matrix can be defined as:

$$\mathbf{A_d} = \mathbf{I} + \mathbf{A}_c\Delta t \tag{8.25}$$

$$\mathbf{B_d} = \mathbf{B}_c\Delta t \tag{8.26}$$

$$\mathbf{C_d} = \mathbf{C}_c \tag{8.27}$$

A.2.4 Optimal control

Whilst performing a specific task, many redundant muscles do not contribute to the task's outcome. From [43, 44, 45, 46, 47] it follows that the CNS selects the muscle patterns contributing to a task based on minimizing the effort, hence minimizing the cost. Optimal control is often used to operate dynamic systems at a minimal cost. The system in 8.2 is a linear system. To provide optimal controlled feedback to a linear system the Linear Quadratic Regulator (LQR) control method is used. The idea behind this method is to find the gain matrix **K**. This gain matrix is used to determine the new optimal motor command for x, y and z direction ($\mathbf{u}(t) = -\mathbf{K}\mathbf{x}(t)$).
The methodology in deriving the gain matrix **K** is based on the infinite horizon LQR problem of linear discrete-time systems from [50].

Defining a quadratic cost function

According to [51] the quadratic accumulated cost is given by:

$$J = \sum_{k=1}^{N} \mathbf{x}_k^{\mathrm{T}} \mathrm{T} \mathbf{x}_k + \mathbf{u}_k^{T} \mathbf{L} \mathbf{u}_k \tag{8.28}$$

Where,

- The weighting matrix **T** is positive semi-definite, hence $\mathbf{T} = \mathbf{T}^{\mathrm{T}}$ with eigenvalues ≥ 0
- The weighting matrix **L** is positive-definite, hence $\mathbf{L} = \mathbf{L}^{\mathrm{T}}$ with eigenvalues > 0

During movement, there is no need to keep the endpoint fixed at a specific location. Therefore, the cost during movement can be fully described by the muscle activation that is needed to reach the desired goal location in x, y and z. Keeping the endpoint fixed comes with a high cost. However, if it is desired to keep the endpoint fixed, the deviation between the actual endpoint location and the desired location is minimal. Therefore, there is not much muscle activation needed to overcome this minimal error. Hence, the cost function in the stationary case is almost entirely dependent on the desired behaviour of the endpoint (fixed at a point or fixed on a line, for example). The weights to define the importance of the stationarity, velocity or exerted force of the endpoint are described by the matrix **T**. Based on this, the expression from 8.28 can be expanded. To keep the cost low, the weighting matrix **T** is equal to zero during the movement towards the goal location. However, once the goal location is reached the weight matrix **T** will have the predetermined values $w^{(p,i)}$, $w^{(v,i)}$ and $w^{(z,i)}$, where i defines the orientation, x, y or z. Furthermore, if it is assumed that the goal location is a point in space, the weight values for the position, velocity and force are the same in every direction. This methodology gives rise to an expanded quadratic accumulated cost function as:

$$\begin{aligned} J = \sum_{k=1}^{N} \Bigg(& \left[w^{(p,x)} \left(p_k^{(x)} - g_k^{(x)} \right)^2 + w^{(p,y)} \left(p_k^{(y)} - g_k^{(y)} \right)^2 + w^{(p,z)} \left(p_k^{(z)} - g_k^{(z)} \right)^2 \right] \\ & + \left[w^{(v,x)} \left(v_k^{(x)} \right)^2 + w^{(v,y)} \left(v_k^{(y)} \right)^2 + w^{(v,z)} \left(v_k^{(z)} \right)^2 \right] \\ & + \left[w^{(f,x)} \left(f_k^{(x)} \right)^2 + w^{(f,y)} \left(f_k^{(y)} \right)^2 + w^{(f,z)} \left(f_k^{(z)} \right)^2 \right] \\ & + \left[\left(u_k^{(x)} \right)^2 + \left(u_k^{(y)} \right)^2 + \left(u_k^{(z)} \right)^2 \right] \Bigg) \end{aligned} \tag{8.29}$$

From the above expansion and 8.28 it follows that the weighting matrices **T** and **L** can be defined as:

$$\mathbf{T}=\begin{bmatrix} w^{(p,x)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w^{(p,x)} & 0 & 0 \\ 0 & w^{(p,y)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w^{(p,y)} & 0 \\ 0 & 0 & w^{(p,z)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -w^{(p,z)} \\ 0 & 0 & 0 & w^{(v,x)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w^{(v,y)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & w^{(v,z)} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & w^{(f,x)} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & w^{(f,y)} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w^{(f,z)} & 0 & 0 & 0 \\ -w^{(p,x)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w^{(p,x)} & 0 & 0 \\ 0 & -w^{(p,y)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w^{(p,y)} & 0 \\ 0 & 0 & -w^{(p,z)} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & w^{(p,z)} \end{bmatrix} \tag{8.30}$$

$$\mathbf{L}=\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{8.31}$$

The goal is to find the motor commands ($\mathbf{u_1}, \mathbf{u_2}...\mathbf{u_N}$) that minimizes this total cost.
Following the dynamic programming method of [50] it is found that these commands follow from:

$$\mathbf{u}_k = -\mathbf{K}_k \mathbf{x}_k \tag{8.32}$$

Feedback gain matrix and analytical Riccati solution

The feedback gain matrix $\mathbf{K}_k$ can be expressed as[50]:

$$\mathbf{K}_k = \left(\mathbf{B_d}^{\mathrm{T}}\mathbf{V}_{k+1}\mathbf{B_d} + \mathbf{L}\right)^{-1}\mathbf{B_d}^{\mathrm{T}}\mathbf{V}_{k+1}\mathbf{A_d} \tag{8.33}$$

Where $\mathbf{V}_{k+1}$ is the discrete time algebraic Riccati equation (DARE). Based on the dynamic programming method the solution of the DARE is found backwards in time [50] as:

$$\mathbf{V}_k = (\mathbf{A_d} - \mathbf{B_d}\mathbf{K}_k)^{\mathrm{T}}\mathbf{V}_{k+1}(\mathbf{A_d} - \mathbf{B_d}\mathbf{K}_k) + \mathbf{K}_k^{\mathrm{T}}\mathbf{L}\mathbf{K}_k + \mathbf{T} \tag{8.34}$$

This DARE can be simplified by first expanding 8.34 into.

$$\mathbf{V}_k = \mathbf{A_d}^{\mathrm{T}}\mathbf{V}_{k+1}(\mathbf{A_d} - \mathbf{B_d}\mathbf{K}_k) - \mathbf{B_d}^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}}\mathbf{V}_{k+1}(\mathbf{A_d} - \mathbf{B_d}\mathbf{K}_k) + \mathbf{K}_k^{\mathrm{T}}\mathbf{L}\mathbf{K}_k + \mathbf{T} \tag{8.35}$$

Doing the following assumption gives the expression as:

$$\begin{aligned} \mathbf{V}_k &= \mathbf{A_d}^{\mathrm{T}}\mathbf{V}_{k+1}(\mathbf{A_d} - \mathbf{B_d}\mathbf{K}_k) \underbrace{-\mathbf{B_d}^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}}\mathbf{V}_{k+1}(\mathbf{A_d} - \mathbf{B_d}\mathbf{K}_k) + \mathbf{K}_k^{\mathrm{T}}\mathbf{L}\mathbf{K}_k}_{=0} + \mathbf{T} && (8.36) \\ &= \mathbf{A_d}^{\mathrm{T}}\mathbf{V}_{k+1}(\mathbf{A_d} - \mathbf{B_d}\mathbf{K}_k) + \mathbf{T} && (8.37) \end{aligned}$$

From the assumption it follows that:

$$
\begin{aligned}
-\mathbf{B_d}^{\mathrm{T}}\mathbf{K}_k^{\mathrm{T}}\mathbf{V}_{k+1}\left(\mathbf{A_d}-\mathbf{B_d}\mathbf{K}_k\right)+\mathbf{K}_k^{\mathrm{T}}\mathbf{L}\mathbf{K}_k &= 0 && (8.38)\\
\mathbf{L}\mathbf{K}_k &= \mathbf{B_d}^{\mathrm{T}}\mathbf{V}_{k+1}\left(\mathbf{A_d}-\mathbf{B_d}\mathbf{K}_k\right) && (8.39)\\
\mathbf{L}\mathbf{K}_k &= \mathbf{B_d}^{\mathrm{T}}\mathbf{V}_{k+1}\mathbf{A_d}-\mathbf{B_d}^{\mathrm{T}}\mathbf{V}_{k+1}\mathbf{B_d}\mathbf{K}_k && (8.40)\\
\left(\mathbf{B_d}^{\mathrm{T}}\mathbf{V}_{k+1}\mathbf{B_d}+\mathbf{L}\right)\mathbf{K}_k &= \mathbf{B_d}^{\mathrm{T}}\mathbf{V}_{k+1}\mathbf{A_d} && (8.41)\\
\mathbf{K}_k &= \left(\mathbf{B_d}^{\mathrm{T}}\mathbf{V}_{k+1}\mathbf{B_d}+\mathbf{L}\right)^{-1}\mathbf{B_d}^{\mathrm{T}}\mathbf{V}_{k+1}\mathbf{A_d} && (8.42)
\end{aligned}
$$

Where 8.42 is again the feedback gain as in 8.33. Therefore, the assumption should hold, giving the simplified DARE as in 8.37.

A.2.5 Predicted trajectory

The predicted trajectory now follows from the new motor commands determined by the feedback gain. Hence, 8.22 can be written as:

$$\mathbf{x}_{k+1} = (\mathbf{A_d}-\mathbf{B_d}\mathbf{K}_k)\mathbf{x}_k \qquad (8.43)$$

Where $\mathbf{K}_k$ is predicted backwards in time as described in section A.2.4. To determine the backwards iteration, the final values $\mathbf{K}_N$ and $\mathbf{V}_N$ need to be known.
There is no need for a gain matrix at iteration instance N since the final prediction $\mathbf{x}_N$ is not dependent on $\mathbf{K}_N$ but on $\mathbf{K}_{N-1}$, following from 8.43. Furthermore, the DARE is determined backwards in time and therefore also dependent on $\mathbf{K}_{N-1}$. Hence, $\mathbf{K}_N$ can be set to zero.

Setting $\mathbf{K}_N = 0$ and knowing that $\mathbf{V}_{N+1}$ does not exists gives rise to the final condition for the DARE, namely $\mathbf{V}_N = \mathbf{T}$, according to 8.37.

B Protocol

Protocol that needs to be followed by the participant

- The participant visits the RaM lab at the University of Twente at the predetermined time. ± 0 min.

- The participant should wear a (non-medical) face mask, which is similar to those that are mandatory in Dutch public covered areas as of December 1, 2020. ± 0 min.

- The participant should answer questions from the COVID-19 checklist, see appendix D (Publication date 03/25/2021 - 15:14) obtained from the RIVM website. ± 1 min.

- The participant is asked to read a form which explains what is expected from him/her, what we will do and why we will do it. This form can be found in appendix E. ± 1 min.

- The participant is asked to read and fill in a consent form, see appendix F for the consent form. ± 3 min.

- The participant should take place on a chair near the haptic device. ± 0 min.

- The Myo armband/armbands are placed on the arm of the participant. ± 1 min.

- The participant is asked to align his/her arm with the body such that the palm of the hand is aligned with the side of the outer thigh. He/she is asked to hold this posture for a few seconds whilst an orientation calibration is done. ± 1 min.

- The participant is asked to grab the haptic device and is instructed to perform several calibrations. ± 5 min.

- After the instructions are made clear one of the two tasks is performed. ± 5 min.

- Again, instructions are given and the other task is performed. ± 5 min.

- After the tasks are performed a questionnaire will be filled out by the participant. ± 2 min.

- The same procedure follows. However, the calibration step can be skipped. Hence, the participant is asked to grab the haptic device and perform one of the two tasks. ± 5 min.

- Again, instructions are given and the other task is performed. ± 5 min.

- After the tasks are performed a questionnaire will be filled out by the participant. ± 2 min.

- The Myo armband/armbands brace be taken off and will be cleaned for use with the next participant. ± 1 min.

__________ +

± 37 min.

C Questionnaire

Human impedance regulation in Haptic Control using EMG

What is your age?

What is your gender?

1. Male
2. Female

Are you right handed?

1. Yes
2. No

Do you have any physical limitations that may influence the way in which you apply motion control or perceive things? (answer with no or yes, namely...)

Did you feel any difference regarding the weights of the blocks?

1. Yes, the first one felt heavier
2. Yes, the second one felt heavier
3. No, I didn’t feel any difference at all

If you perceived a difference, it was due to:

1. The visual feedback
2. The haptic feedback
3. Both of them
4. I did not perceive a difference between the blocks

The current control mode is useful

1. Strongly agree
2. Agree
3. More or less agree
4. Undecided
5. More or less disagree
6. Disagree
7. Strongly disagree

Human impedance regulation in Haptic Control using EMG

The current control mode enhances my working performance

1. Strongly agree
2. Agree
3. More or less agree
4. Undecided
5. More or less disagree
6. Disagree
7. Strongly disagree

I accomplished the given task rapidly

1. Strongly agree
2. Agree
3. More or less agree
4. Undecided
5. More or less disagree
6. Disagree
7. Strongly disagree

I was able to perform precise motions with this control mode

1. Strongly agree
2. Agree
3. More or less agree
4. Undecided
5. More or less disagree
6. Disagree
7. Strongly disagree

This control mode is easy to use

1. Strongly agree
2. Agree
3. More or less agree
4. Undecided
5. More or less disagree
6. Disagree
7. Strongly disagree

It was easy to learn to use this control mode

1. Strongly agree
2. Agree
3. More or less agree
4. Undecided
5. More or less disagree
6. Disagree
7. Strongly disagree

I feel physically uncomfortable in using this control mode

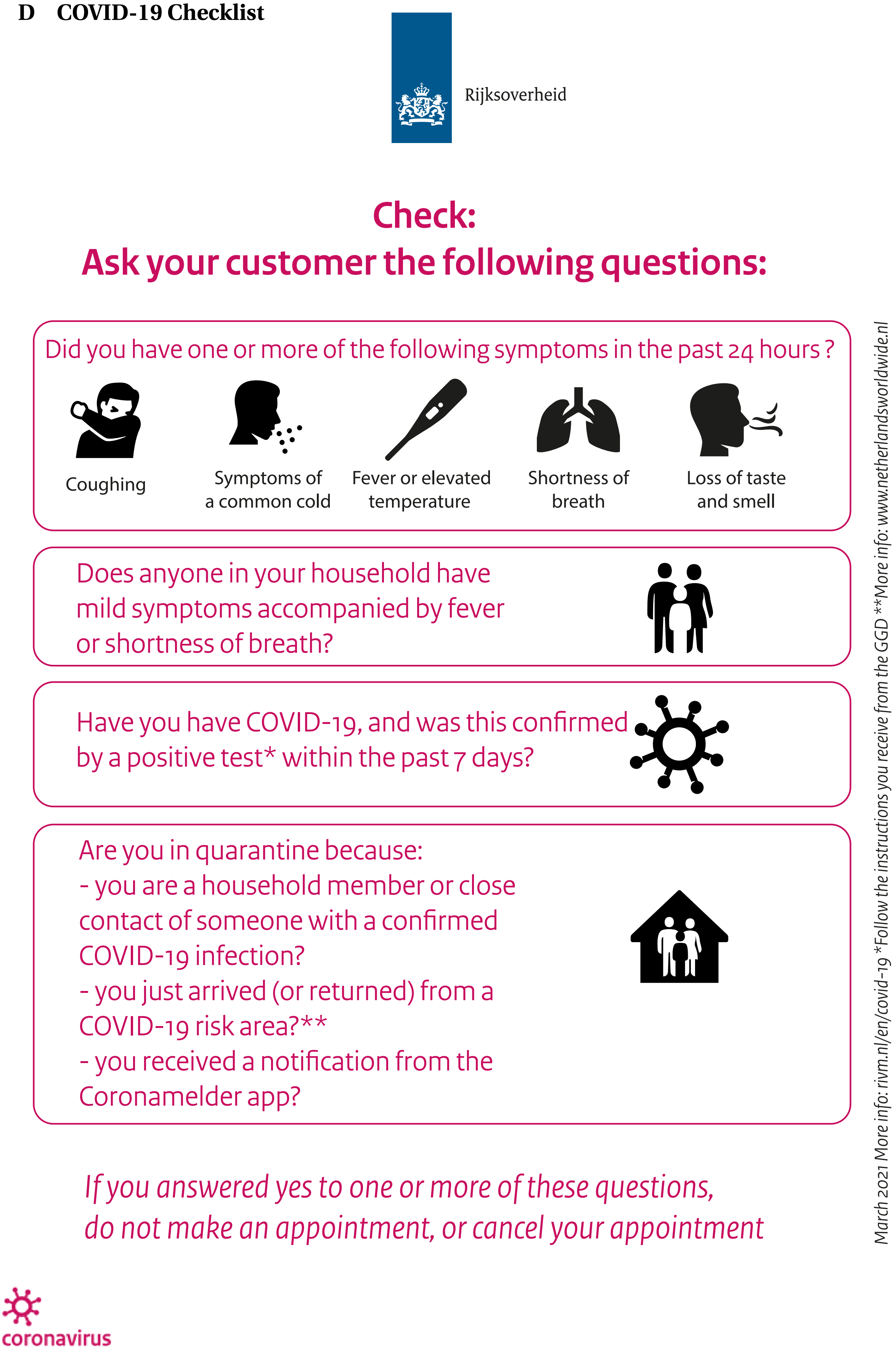
1. Strongly agree
2. Agree
3. More or less agree
4. Undecided
5. More or less disagree
6. Disagree
7. Strongly disagree

I feel tense in using this control mode

1. Strongly agree
2. Agree
3. More or less agree
4. Undecided
5. More or less disagree
6. Disagree
7. Strongly disagree

Describe with 3 to 5 adjectives the controller that you just experienced

D COVID-19 Checklist



Rijksoverheid

Check:
Ask your customer the following questions:

Did you have one or more of the following symptoms in the past 24 hours ?

Coughing

Symptoms of a common cold

Fever or elevated temperature

Shortness of breath

Loss of taste and smell

Does anyone in your household have mild symptoms accompanied by fever or shortness of breath?

Have you have COVID-19, and was this confirmed by a positive test* within the past 7 days?

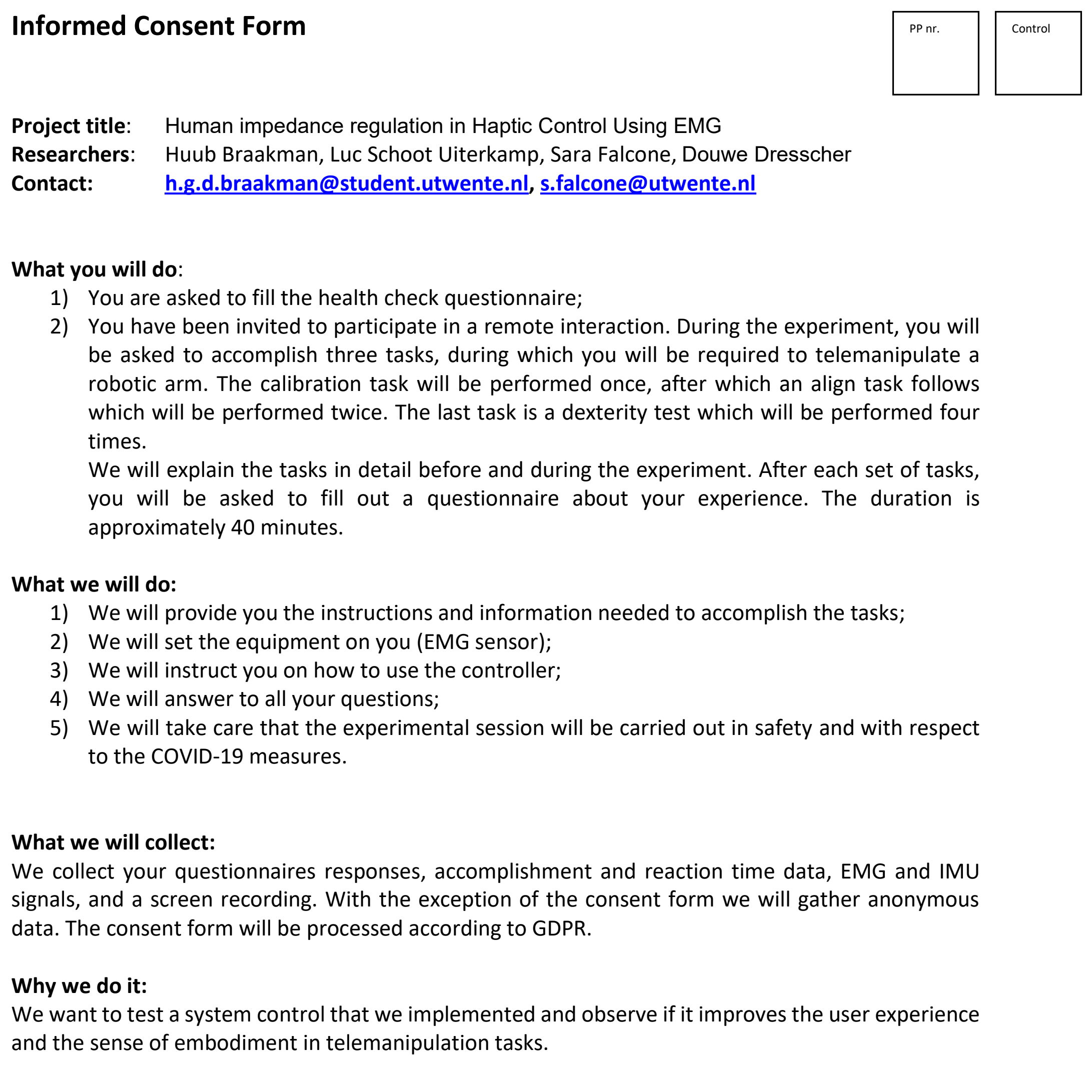
Are you in quarantine because:
- you are a household member or close contact of someone with a confirmed COVID-19 infection?
- you just arrived (or returned) from a COVID-19 risk area?**
- you received a notification from the Coronamelder app?

If you answered yes to one or more of these questions, do not make an appointment, or cancel your appointment

coronavirus

March 2021 More info: rivm.nl/en/covid-19 *Follow the instructions you receive from the GGD **More info: www.netherlandsworldwide.nl

E Informed Consent form



Informed Consent Form

PP nr.

Control

Project title: Human impedance regulation in Haptic Control Using EMG
Researchers: Huub Braakman, Luc Schoot Uiterkamp, Sara Falcone, Douwe Dresscher
Contact: **h.g.d.braakman@student.utwente.nl**, **s.falcone@utwente.nl**

What you will do:

1) You are asked to fill the health check questionnaire;
2) You have been invited to participate in a remote interaction. During the experiment, you will be asked to accomplish three tasks, during which you will be required to telemanipulate a robotic arm. The calibration task will be performed once, after which an align task follows which will be performed twice. The last task is a dexterity test which will be performed four times.
   We will explain the tasks in detail before and during the experiment. After each set of tasks, you will be asked to fill out a questionnaire about your experience. The duration is approximately 40 minutes.

What we will do:

1) We will provide you the instructions and information needed to accomplish the tasks;
2) We will set the equipment on you (EMG sensor);
3) We will instruct you on how to use the controller;
4) We will answer to all your questions;
5) We will take care that the experimental session will be carried out in safety and with respect to the COVID-19 measures.

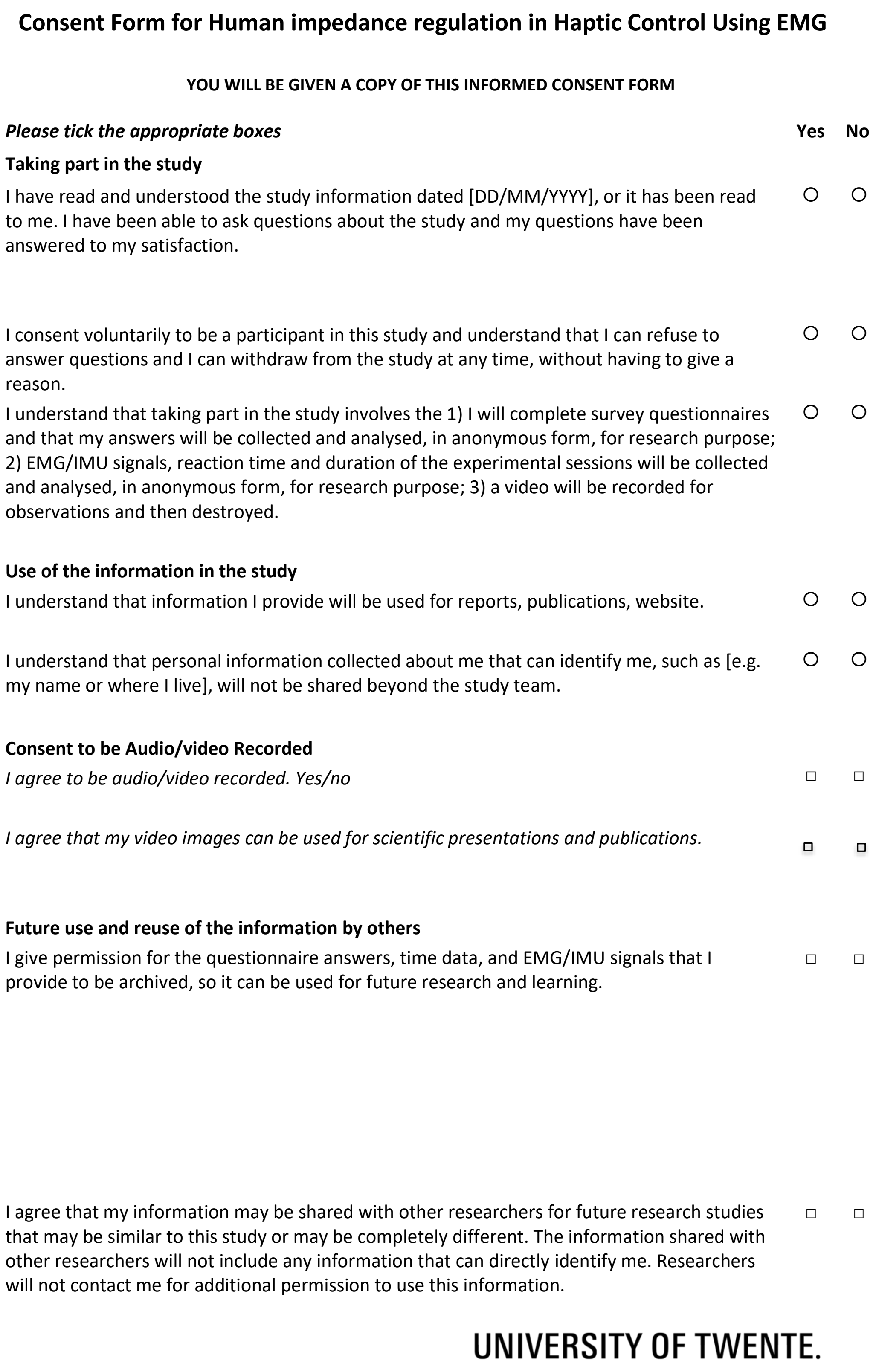
What we will collect:

We collect your questionnaires responses, accomplishment and reaction time data, EMG and IMU signals, and a screen recording. With the exception of the consent form we will gather anonymous data. The consent form will be processed according to GDPR.

Why we do it:

We want to test a system control that we implemented and observe if it improves the user experience and the sense of embodiment in telemanipulation tasks.

F Consent form



Consent Form for Human impedance regulation in Haptic Control Using EMG

YOU WILL BE GIVEN A COPY OF THIS INFORMED CONSENT FORM

Please tick the appropriate boxes	**Yes**	**No**
Taking part in the study		
I have read and understood the study information dated [DD/MM/YYYY], or it has been read to me. I have been able to ask questions about the study and my questions have been answered to my satisfaction.	○	○
I consent voluntarily to be a participant in this study and understand that I can refuse to answer questions and I can withdraw from the study at any time, without having to give a reason.	○	○
I understand that taking part in the study involves the 1) I will complete survey questionnaires and that my answers will be collected and analysed, in anonymous form, for research purpose; 2) EMG/IMU signals, reaction time and duration of the experimental sessions will be collected and analysed, in anonymous form, for research purpose; 3) a video will be recorded for observations and then destroyed.	○	○
Use of the information in the study		
I understand that information I provide will be used for reports, publications, website.	○	○
I understand that personal information collected about me that can identify me, such as [e.g. my name or where I live], will not be shared beyond the study team.	○	○
Consent to be Audio/video Recorded		
I agree to be audio/video recorded. Yes/no	□	□
I agree that my video images can be used for scientific presentations and publications.	□	□
Future use and reuse of the information by others		
I give permission for the questionnaire answers, time data, and EMG/IMU signals that I provide to be archived, so it can be used for future research and learning.	□	□
I agree that my information may be shared with other researchers for future research studies that may be similar to this study or may be completely different. The information shared with other researchers will not include any information that can directly identify me. Researchers will not contact me for additional permission to use this information.	□	□

UNIVERSITY OF TWENTE.

Signatures

________________________ ________________________ _________

Name of participant

I have accurately read out the information sheet to the potential participant and, to the best of my ability, ensured that the participant understands to what they are freely consenting.

___________________________ ____________________ _________

Researcher name Signature Date

Study contact details for further information: Huub Braakman
h.g.d.braakman@student.utwente.nl**, Sara Falcone*,* s.falcone@utwente.nl;**

Contact Information for Questions about Your Rights as a Research Participant

If you have questions about your rights as a research participant, or wish to obtain information, ask questions, or discuss any concerns about this study with someone other than the researcher(s), please contact the Secretary of the Ethics Committee of the Faculty of EEMCS at the University of Twente by **ethics-comm-ewi@utwente.nl**.

G User study planning and observations

Users:	*Date:*	*Start time:*	*Start with controller*[1]	*Start with block*[2]	*Path errors (obtained during experiment):*
1	17-5-2021	12:30	1	Light	1 (fell over, controller 2 heavy block)
2	17-5-2021	13:30	1	Heavy	1(fell over, controller 2 heavy block)
3	17-5-2021	14:30	2	Heavy	1(fell over, controller 2 heavy block)

4	17-5-2021	15:30	2	Light	2x Pushed too hard on the block, force safety mode (controller 2 heavy block)
					1(fell over, controller 2 heavy)
					1(fell over, controller 1 light block)

5	17-5-2021	16:30	2	Light	--
6	18-5-2021	09:30	1	Heavy	1(fell over, controller 1 light block)
7	18-5-2021	10:30	1	Heavy	2x Pushed too hard on the block, force safety mode (controller 1 light block and controller 2 heavy block) 1(fell over, controller 2 light block)
8	18-5-2021	11:30	1	Light	1(fell over, controller 1 light block)
9	18-5-2021	12:30	1	Heavy	1x Pushed too hard on the block, force safety mode (controller 1 heavy block)
10	18-5-2021	13:00	2	Heavy	--
11	18-5-2021	14:00	2	Heavy	2x Pushed too hard on the block, force safety mode (controller 2 heavy block)
~~*12*~~	~~19-5-2021~~	~~11:00~~	~~2~~	~~Light~~	-
12	19-5-2021	12:00	1	Light	1x Pushed too hard on the block, force safety mode (controller 1 heavy block)
13	19-5-2021	13:30	1	Heavy	--
~~*15*~~	~~19-5-2021~~	~~14:15~~	~~2~~	~~Light~~	-
14	19-15-2021	15:00	2	Heavy	1(fell over, controller 2 light block)

*[1] Controller order determined by the use of a random generator (https://www.random.org/).
1 defines the posture/co-contraction controller and 2 defines the controller based on co-contraction of a single muscle pair.

*[2] Weight order determined by the use of the same random generator (https://www.random.org/).
The heavy block has a weight of 1.139 [kg] and the light block has a weight of 0.047 [kg].

H Push button 1: results for the posture co-contraction controller

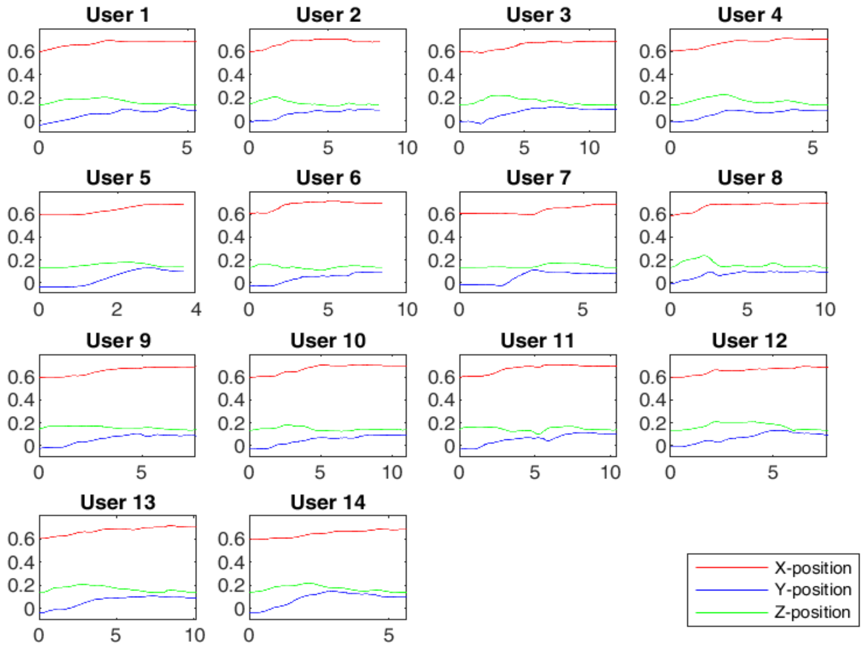


Figure H.1: End-effector position of the Franka Emika Panda arm during the task of pushing button 1 once started in the middle. The results are obtained during the use of the posture co-contraction controller.

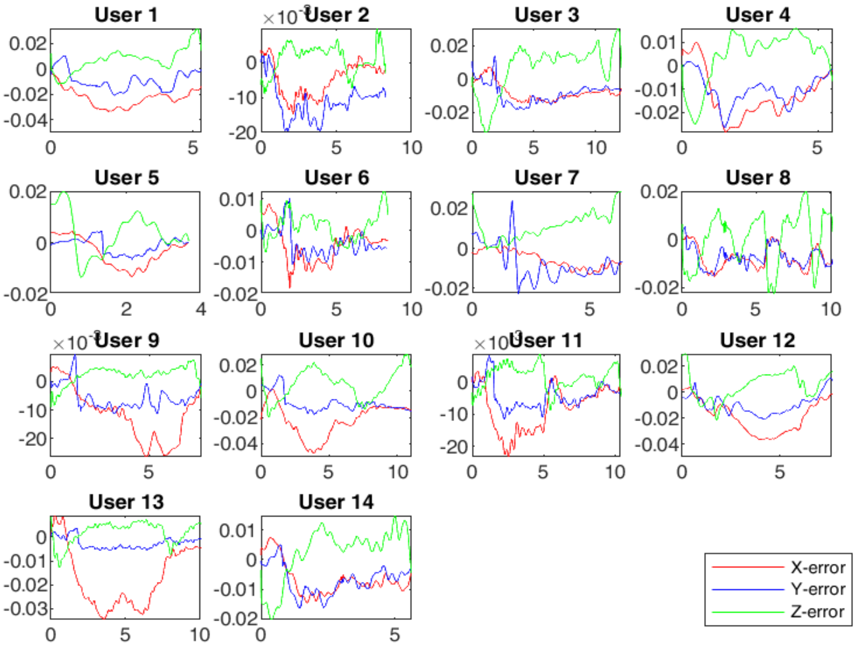


Figure H.2: End-effector error between the the Franka Emika Panda arm and the Virtuose haptic device during the task of pushing button 1 once started in the middle. The results are obtained during the use of the posture co-contraction controller.

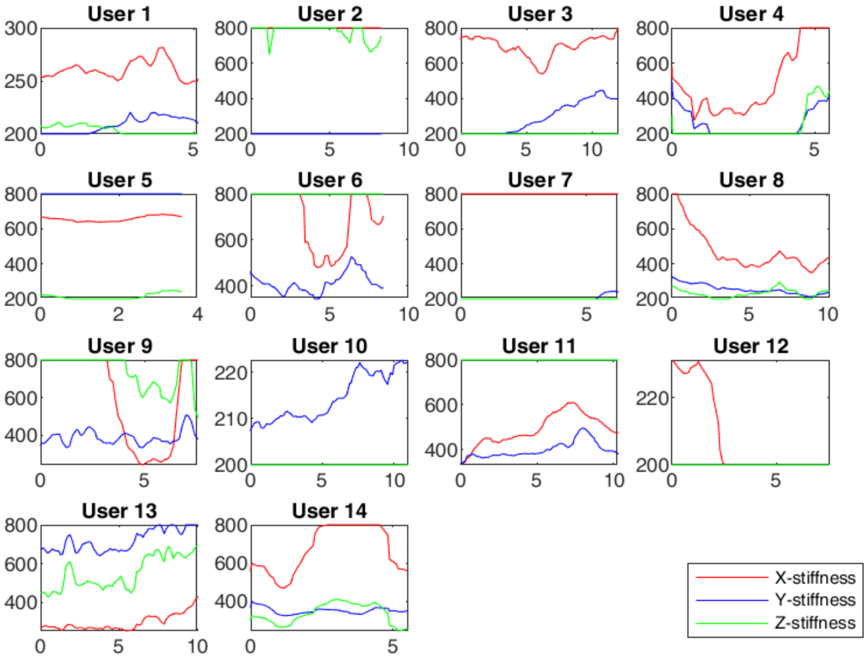


Figure H.3: End-effector stiffness applied to the Franka Emika Panda arm during the task of pushing button 1 once started in the middle. The results are obtained during the use of the posture co-contraction controller.

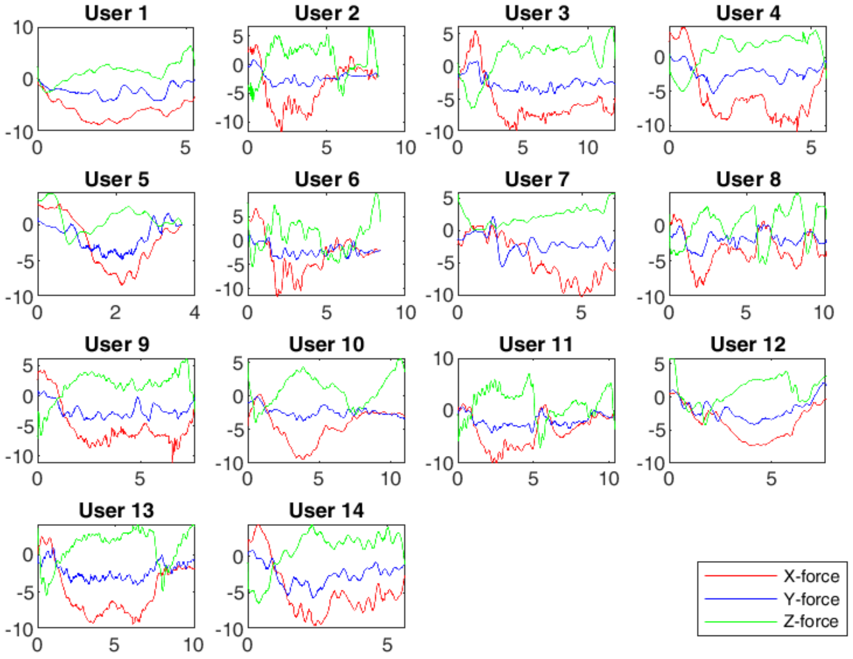


Figure H.4: End-effector force applied to the Franka Emika Panda arm during the task of pushing button 1 once started in the middle. The results are obtained during the use of the posture co-contraction controller.

I Push button 1: results for the single muscle pair co-contraction controller

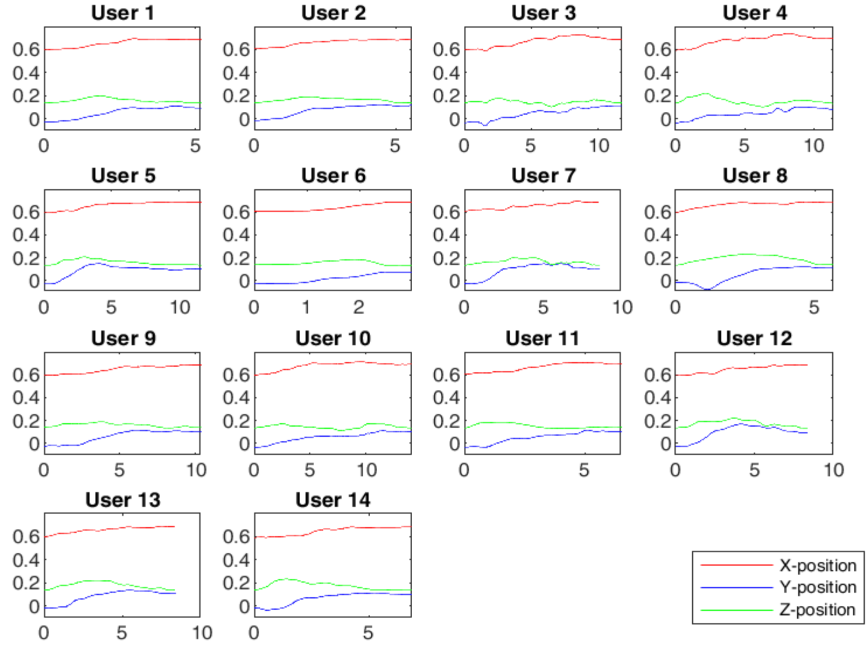


Figure I.1: End-effector position of the Franka Emika Panda arm during the task of pushing button 1 once started in the middle. The results are obtained during the use of the single muscle pair co-contraction controller.

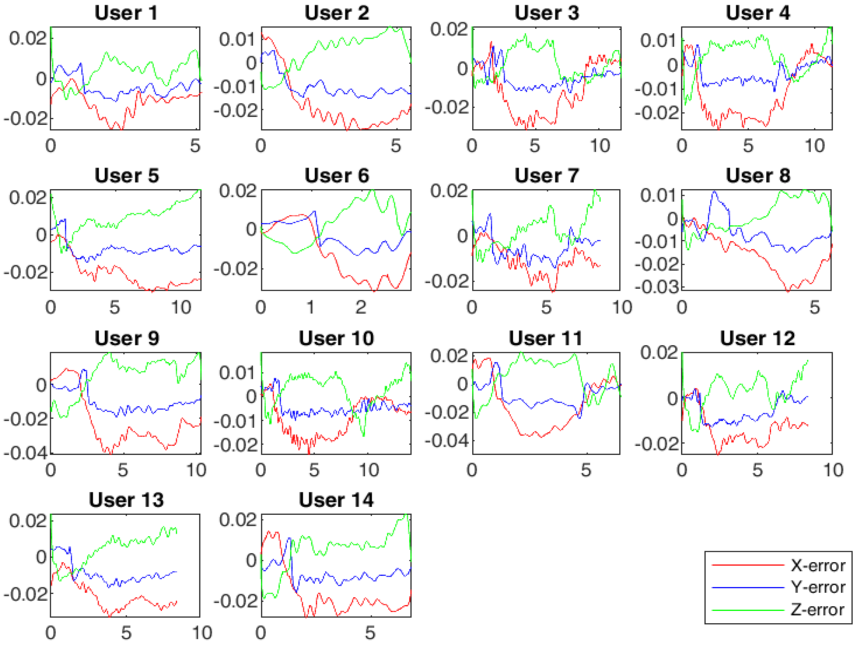


Figure I.2: End-effector error between the the Franka Emika Panda arm and the Virtuose haptic device during the task of pushing button 1 once started in the middle. The results are obtained during the use of the single muscle pair co-contraction controller.

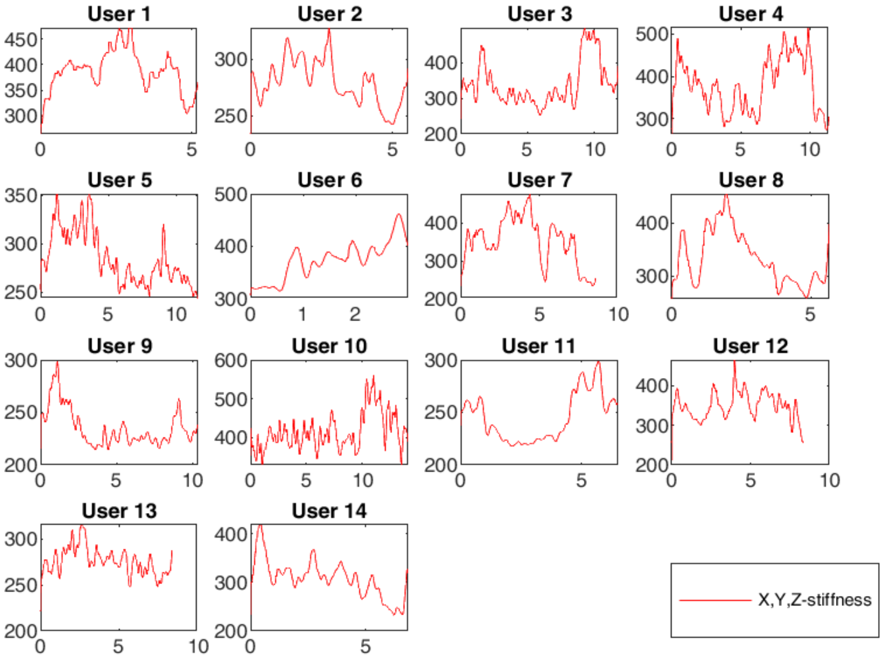


Figure I.3: End-effector stiffness applied to the Franka Emika Panda arm during the task of pushing button 1 once started in the middle. The results are obtained during the use of the single muscle pair co-contraction controller.

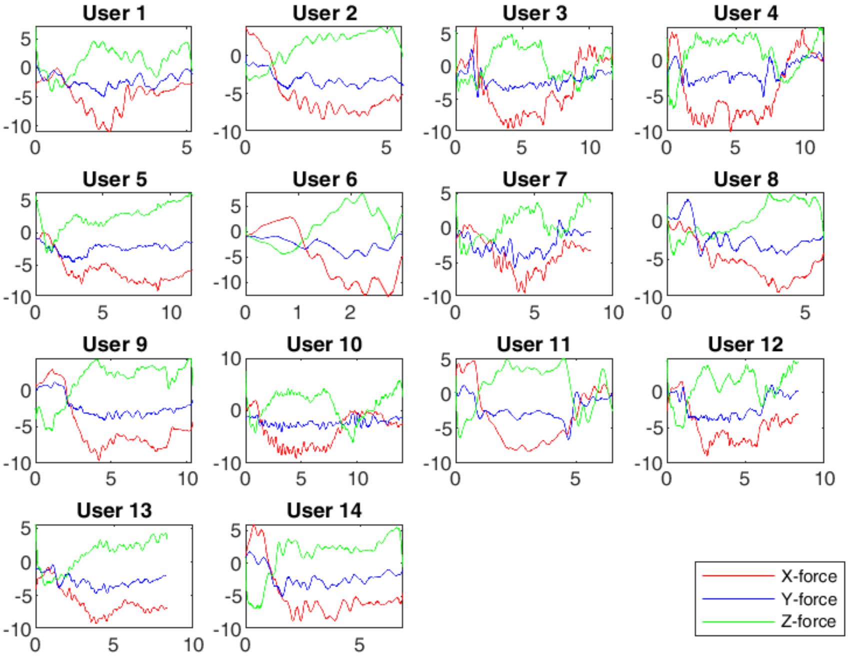


Figure I.4: End-effector force applied to the Franka Emika Panda arm during the task of pushing button 1 once started in the middle. The results are obtained during the use of the single muscle pair co-contraction controller.

J Pushing heavy block: results for the posture co-contraction controller

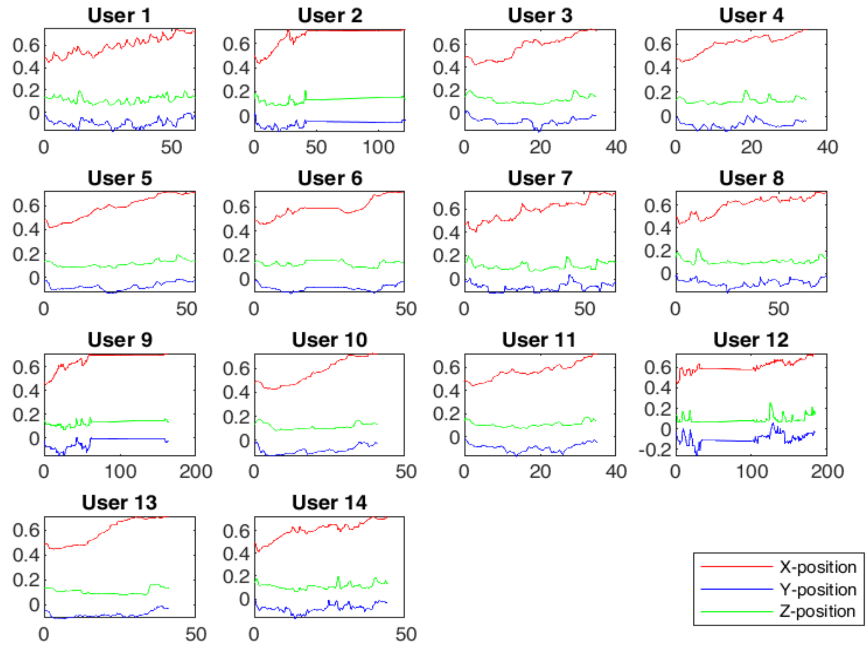


Figure J.1: End-effector position of the Franka Emika Panda arm during the pushing of the heavy block. The results are obtained during the use of the posture co-contraction controller.

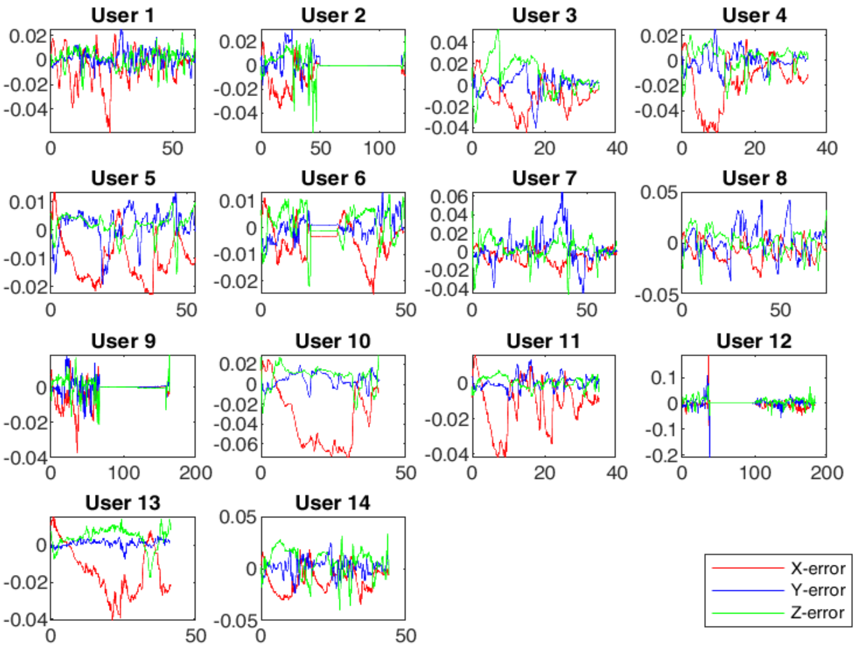


Figure J.2: End-effector error between the Franka Emika Panda arm and the Virtuose haptic device during the pushing of the heavy block. The results are obtained during the use of the posture co-contraction controller.

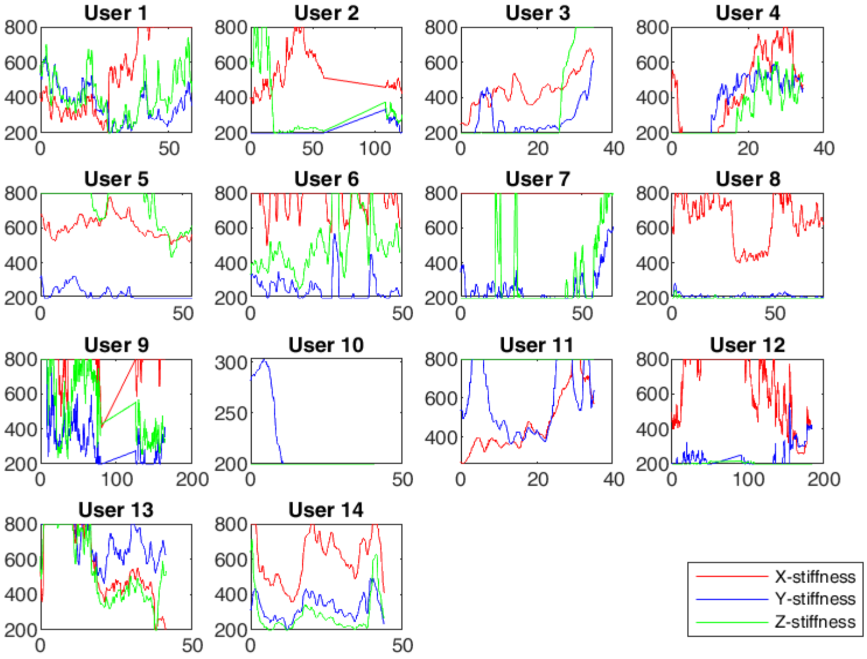


Figure J.3: End-effector stiffness of the Franka Emika Panda arm during the pushing of the heavy block. The results are obtained during the use of the posture co-contraction controller.

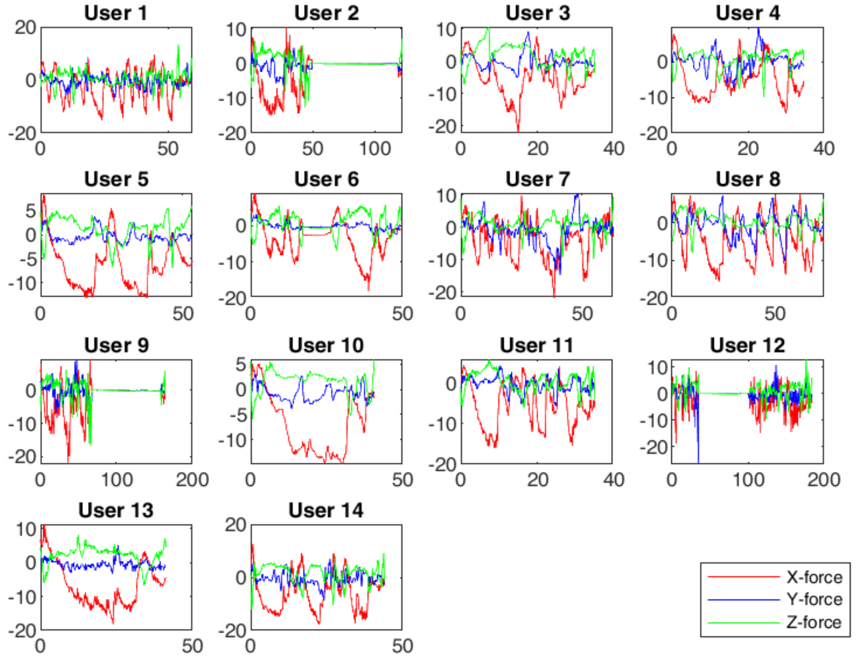


Figure J.4: End-effector force of the Franka Emika Panda arm during the pushing of the heavy block. The results are obtained during the use of the posture co-contraction controller.

K Pushing heavy block: results for the single muscle pair co-contraction controller

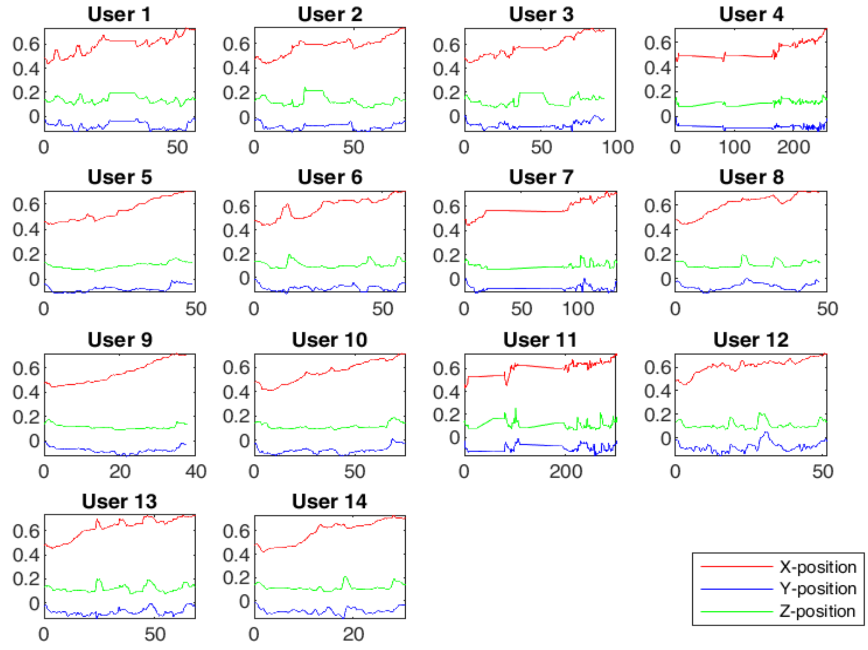


Figure K.1: End-effector position of the Franka Emika Panda arm during the pushing of the heavy block. The results are obtained during the use of the single muscle pair co-contraction controller.

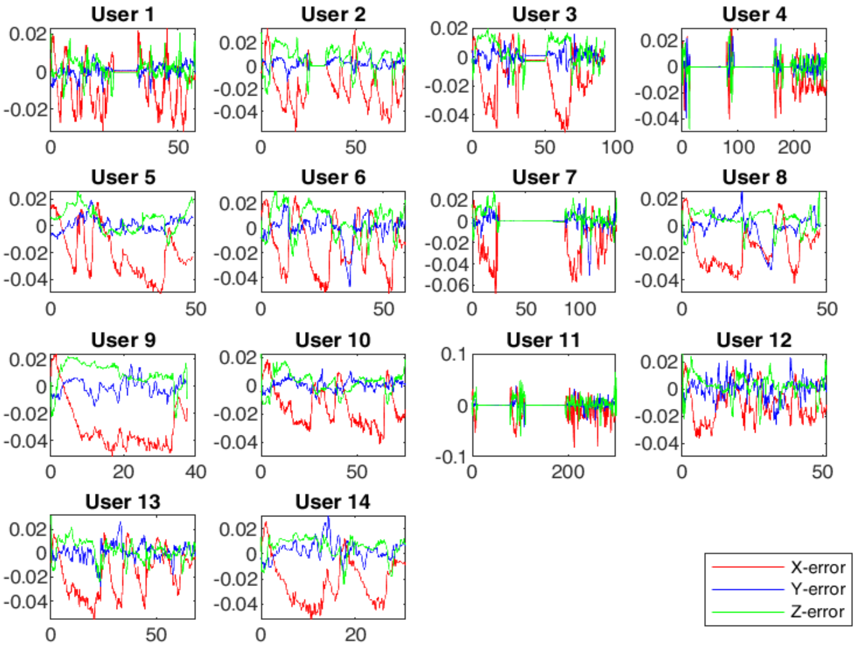


Figure K.2: End-effector error between the Franka Emika Panda arm and the Virtuose haptic device during the pushing of the heavy block. The results are obtained during the use of the single muscle pair co-contraction controller.

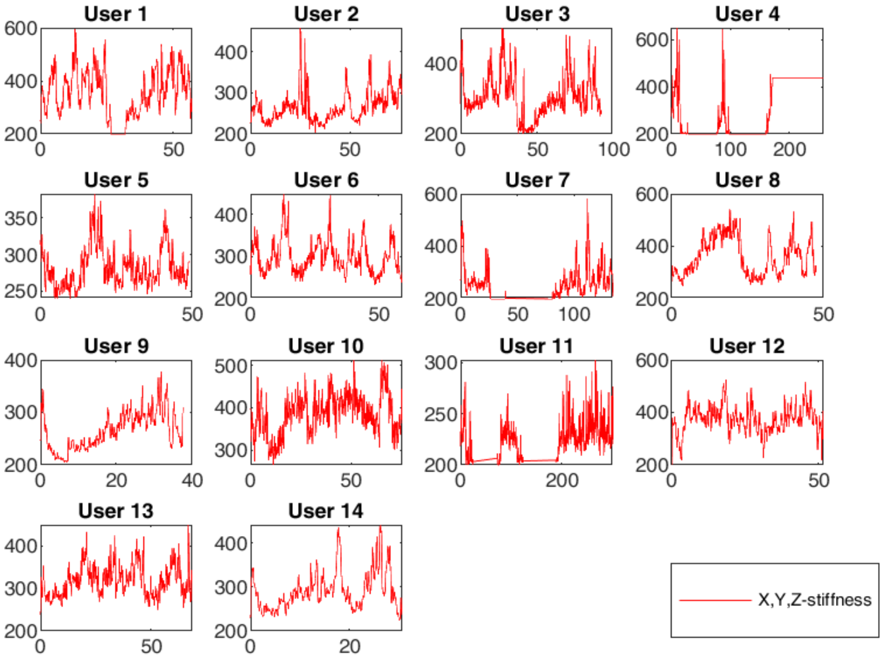


Figure K.3: End-effector stiffness of the Franka Emika Panda arm during the pushing of the heavy block. The results are obtained during the use of the single muscle pair co-contraction controller.

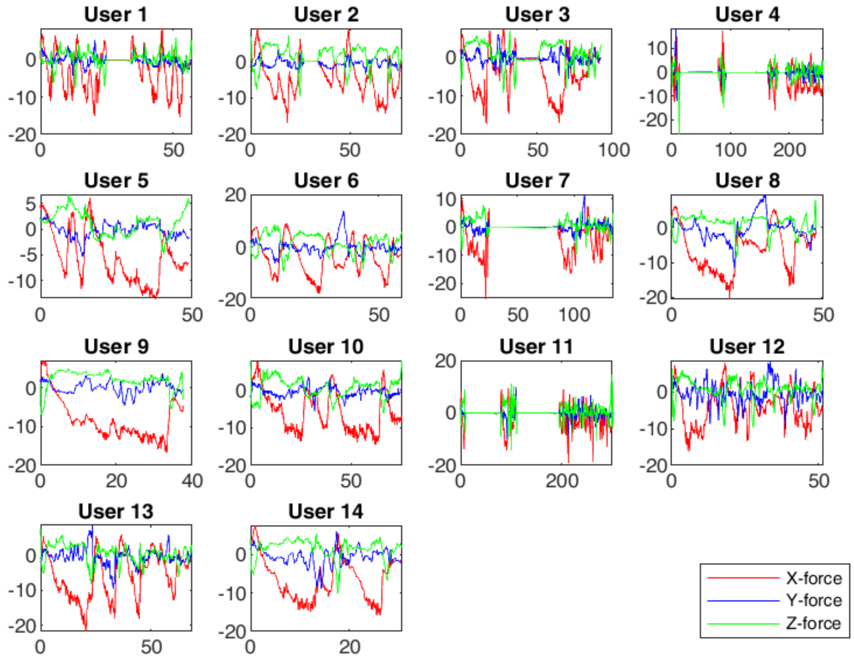


Figure K.4: End-effector force of the Franka Emika Panda arm during the pushing of the heavy block. The results are obtained during the use of the single muscle pair co-contraction controller.

L Pushing light block: results for the posture co-contraction controller

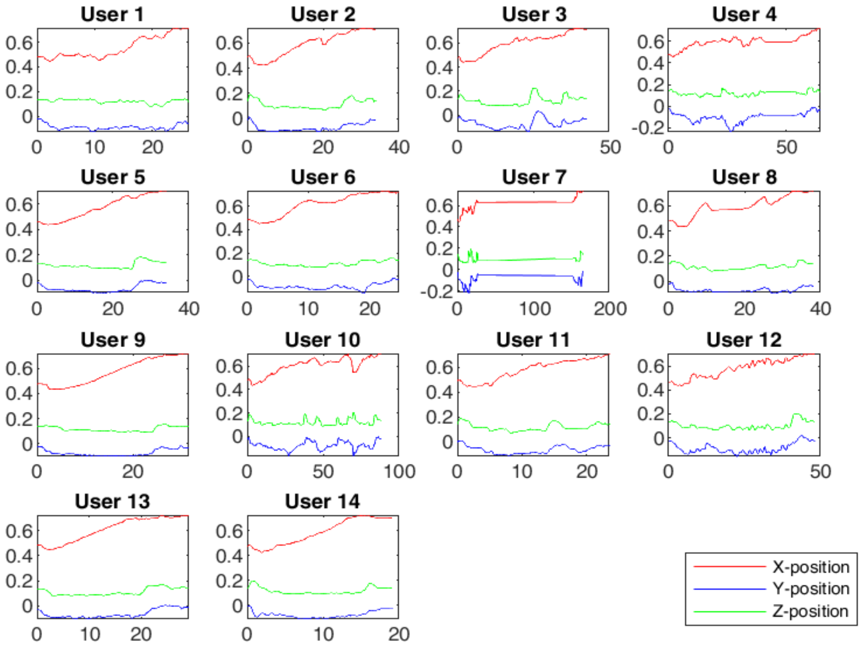


Figure L.1: End-effector position of the Franka Emika Panda arm during the pushing of the light block. The results are obtained during the use of the posture co-contraction controller.

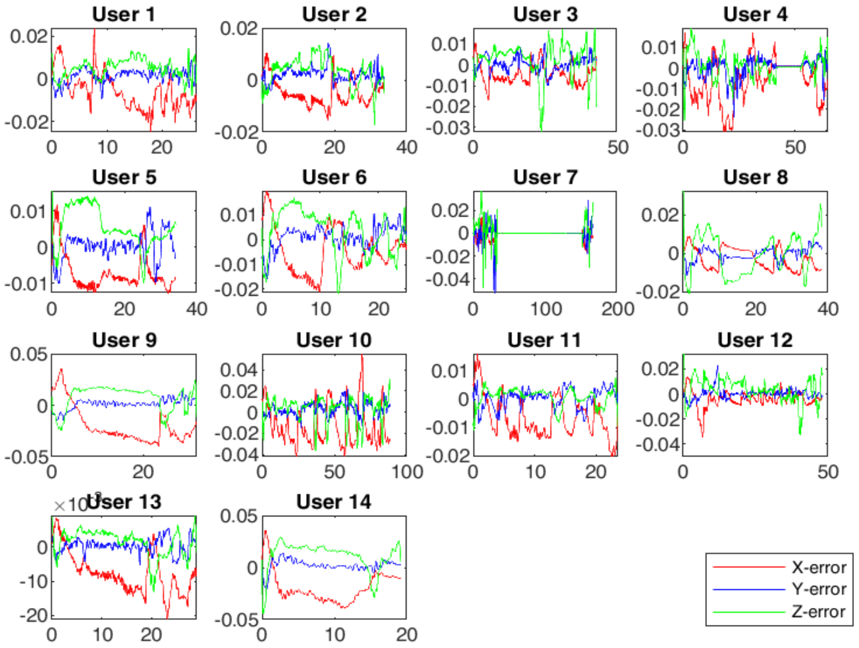


Figure L.2: End-effector error between the Franka Emika Panda arm and the Virtuose haptic device during the pushing of the light block. The results are obtained during the use of the posture co-contraction controller.

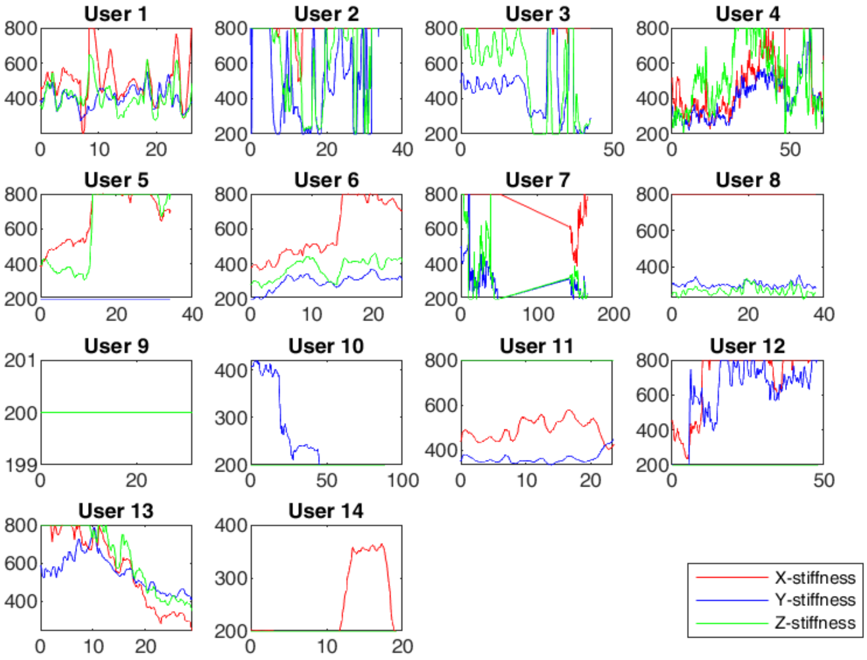


Figure L.3: End-effector stiffness of the Franka Emika Panda arm during the pushing of the light block. The results are obtained during the use of the posture co-contraction controller.

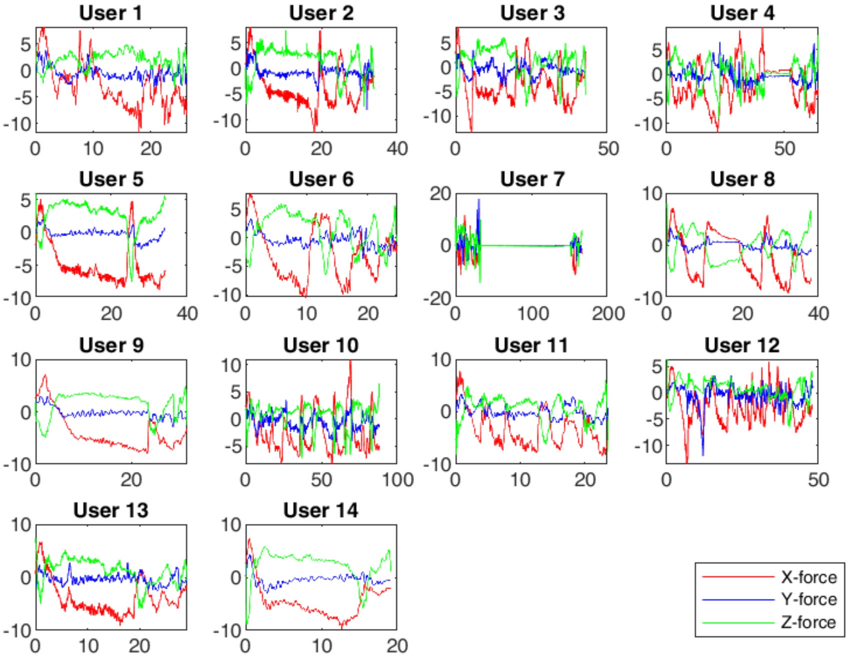


Figure L.4: End-effector force of the Franka Emika Panda arm during the pushing of the light block. The results are obtained during the use of the posture co-contraction controller.

M Pushing light block: results for the single muscle pair co-contraction controller

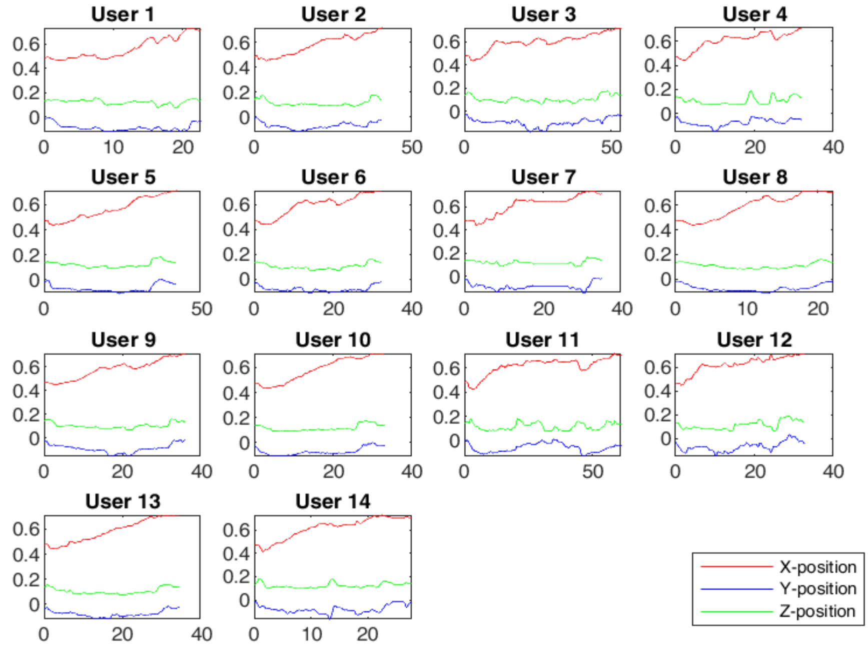


Figure M.1: End-effector position of the Franka Emika Panda arm during the pushing of the light block. The results are obtained during the use of the single muscle pair co-contraction controller.

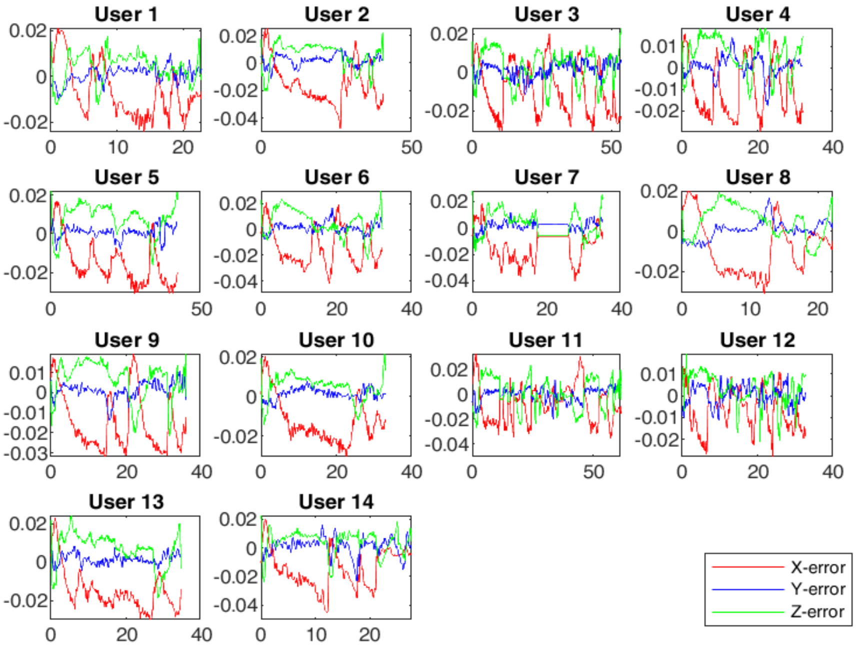


Figure M.2: End-effector error between the Franka Emika Panda arm and the Virtuose haptic device during the pushing of the light block. The results are obtained during the use of the single muscle pair co-contraction controller.

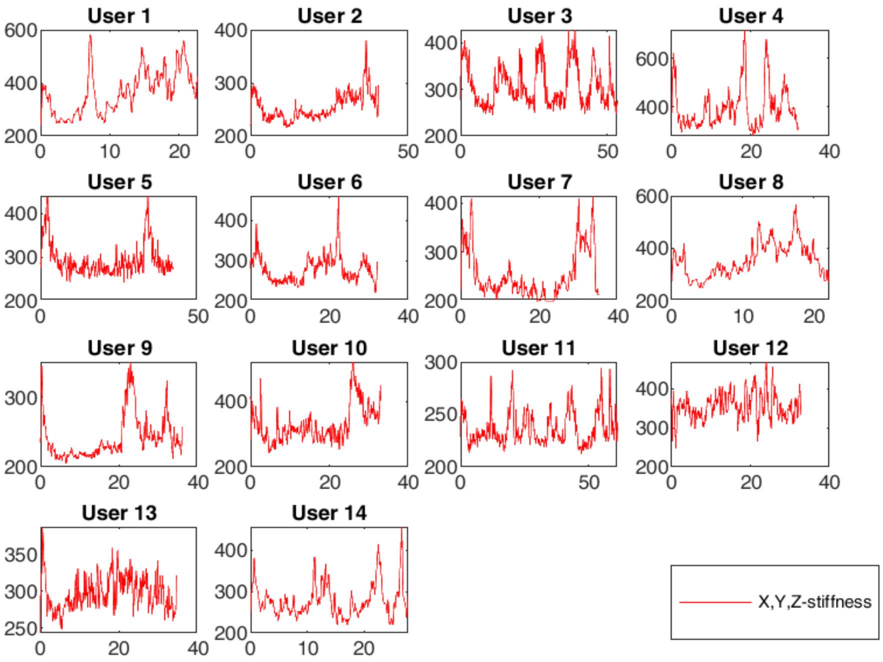


Figure M.3: End-effector stiffness of the Franka Emika Panda arm during the pushing of the light block. The results are obtained during the use of the single muscle pair co-contraction controller.

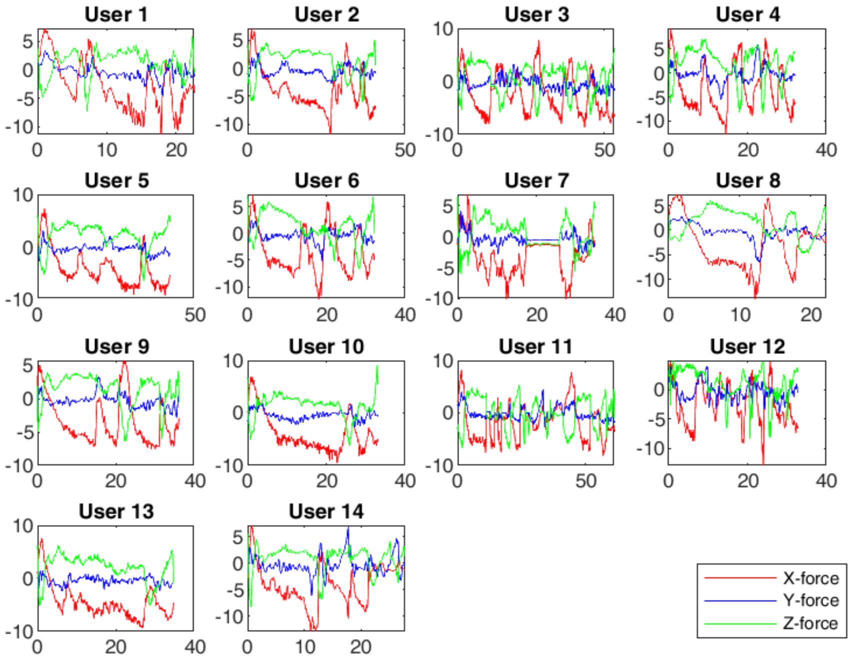


Figure M.4: End-effector force of the Franka Emika Panda arm during the pushing of the light block. The results are obtained during the use of the single muscle pair co-contraction controller.

Bibliography

[1] P. F. Hokayem and M. W. Spong, “Bilateral teleoperation: An historical survey,” *Automatica*, vol. 42, no. 12, pp. 2035 – 2057, 2006.

[2] K. L. Moore, A. F. Dalley, and A. M. Agur, “Clinically oriented anatomy, 7th edition,” *Clinical Anatomy*, vol. 27, no. 2, pp. 508–812, 2014.

[3] M. T.A. Drixler, “Illustratieve anatomie, schouder & armen.” http://www.med-info.nl/anatomie_arm.html. Retrieved on 04-Feb-2021.

[4] H. Ohashi, M. Al-Naser, S. Ahmed, T. Akiyama, T. Sato, P. Nguyen, K. Nakamura, and A. Dengel, “Augmenting wearable sensor data with physical constraint for dnn-based human-action recognition,” 08 2017.

[5] Haption, “Virtuose 6d, the large workspace haptic device".” https://www.haption.com/en/products-en/virtuose-6d-en.html. Retrieved on April. 02, 2021.

[6] WiredWorkers, “Franka emika panda.” https://wiredworkers.io/nl/onze-cobots/cobot-franka-emika-panda/#about. Retrieved on May. 28, 2021.

[7] C. Rand, “The ins and outs of muscle contractions.” https://aurorascientific.com/the-ins-and-outs-of-muscle-contractions/, Aug 2019. Retrieved on 20-May-2021.

[8] S. Lichiardopol, *A survey on teleoperation*. DCT rapporten, Technische Universiteit Eindhoven, 2007. DCT 2007.155.

[9] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikawa, “Ground-space bilateral teleoperation of ets-vii robot arm by direct bilateral coupling under 7-s time delay condition,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp. 499–511, 2004.

[10] G. Niemeyer and J.-J. E. Slotine, “Telemanipulation with time delays,” *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 873–890, 2004.

[11] K. Hashtrudi-Zaad and S. E. Salcudean, “Transparency in time-delayed systems and the effect of local force feedback for transparent teleoperation,” *IEEE Transactions on Robotics and Automation*, vol. 18, no. 1, pp. 108–114, 2002.

[12] E. Estrada, W. Yu, and X. Li, “Stability and transparency of delayed bilateral teleoperation with haptic feedback,” *International Journal of Applied Mathematics and Computer Science*, vol. 29, no. 4, pp. 681 – 692, 01 Dec. 2019.

[13] R. V. Dubey, Tan Fung Chan, and S. E. Everett, “Variable damping impedance control of a bilateral telerobotic system,” *IEEE Control Systems Magazine*, vol. 17, no. 1, pp. 37–45, 1997.

[14] K. Van Teeffelen, D. Dresscher, W. Van Dijk, and S. Stramigioli, “Intuitive impedance modulation in haptic control using electromyography,” in *BIOROB 2018 - 7th IEEE International Conference on Biomedical Robotics and Biomechatronics*, (United States), pp. 1211–1217, IEEE Computer Society, 10 2018.

[15] R. Lieftink, “Bi-directional impedance reflection for cartesian space telemanipulation control to reduce the effect of time delay,” August 2020.

[16] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli, “Bilateral telemanipulation with time delays: A two-layer approach combining passivity and transparency,” *IEEE Transactions on Robotics*, vol. 27, no. 4, pp. 741–756, 2011.

[17] A. Ajoudani, N. G. Tsagarakis, and A. Bicchi, “Tele-impedance: Towards transferring human impedance regulation skills to robots,” in *2012 IEEE International Conference on Robotics and Automation*, pp. 382–388, 2012.

[18] Thalmic Labs, "Myo gesture control armband." https://web.archive.org/web/20170701185826/https://www.myo.com/techspecs. Retrieved on 12-Feb-2021.

[19] C. Fang, A. Ajoudani, A. Bicchi, and N. G. Tsagarakis, "Online model based estimation of complete joint stiffness of human arm," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 84–91, 2018.

[20] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, ch. 4, pp. 115–128. USA: CRC Press, Inc., 1st ed., 1994.

[21] G. Singh, A. Singla, and G. Virk, "Average lengths of human body parts," in *Modeling and Simulation of a Passive Lower-Body Mechanism for Rehabilitation*, 01 2016.

[22] R. W. Brockett, "Robotic manipulators and the product of exponentials formula," in *Mathematical Theory of Networks and Systems* (P. A. Fuhrmann, ed.), (Berlin, Heidelberg), pp. 120–129, Springer Berlin Heidelberg, 1984.

[23] F. C. s. N. U. Park, *Modern robotics - mechanics, planning, and control*, ch. 3.3, pp. 89–104. Cambridge University Press, 2017.

[24] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*, pp. 27–31. USA: CRC Press, Inc., 1st ed., 1994.

[25] F. C. s. N. U. Park, *Modern robotics - mechanics, planning, and control*, ch. 5.1, pp. 178–190. Cambridge University Press, 2017.

[26] G. Folkertsma, *Modern Robotics practical Exercise 2*, p. 1–3. University of Twente, RaM, 2017.

[27] L. Selen, P. Beek, and J. van Dieen, "Can co-activation reduce kinematic variability? a simulation study.," *Biological Cybernetics*, vol. 93, pp. 373–81, 2005.

[28] F. E. Zajac, "Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control," *Crit Rev Biomed Eng*, vol. 17, no. 4, pp. 359–411, 1989.

[29] F. A. Mussa-Ivaldi, N. Hogan, and E. Bizzi, "J NeurosciNeural, mechanical, and geometric factors subserving arm posture in humans," *J Neurosci*, vol. 5, pp. 2732–2743, Oct 1985.

[30] A. Ajoudani, *Replicating Human Stiffness Profile with a Cartesian Impedance Controller in Realtime*, pp. 33–45. Cham: Springer International Publishing, 2016.

[31] M. D. Hill and G. Niemeyer, "Real-time estimation of human impedance for haptic interfaces," in *World Haptics Conference*, (Los Alamitos, CA, USA), pp. 440–445, IEEE Computer Society, mar 2009.

[32] E. Burdet, D. W. Franklin, and T. E. Milner, *Human robotics: neuromechanics and motor control*. MIT Press, 2013.

[33] V. Zatsiorsky, *Kinetics of Human Motion*. Human Kinetics, 2002.

[34] D. Shin, J. Kim, and Y. Koike, "A myokinetic arm model for estimating joint torque and stiffness from emg signals during maintained posture," *Journal of Neurophysiology*, vol. 101, no. 1, pp. 387–401, 2009. PMID: 19005007.

[35] K. R. S. Holzbaur, W. M. Murray, and S. L. Delp, "A model of the upper extremity for simulating musculoskeletal surgery and analyzing neuromuscular control," *Annals of Biomedical Engineering*, vol. 33, pp. 829–840, Jun 2005.

[36] F. Ficuciello, A. Romano, L. Villani, and B. Siciliano, "Cartesian impedance control of redundant manipulators for human-robot co-manipulation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2120–2125, 2014.

[37] F. C. s. N. U. Park, *Modern robotics - mechanics, planning, and control*, ch. 11.7, pp. 441–444. Cambridge University Press, 2017.

[38] C. Fang, A. Ajoudani, A. Bicchi, and N. G. Tsagarakis, "A real-time identification and tracking method for the musculoskeletal model of human arm," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 3472–3479, 2018.

[39] C. English and D. Russell, "Representations of multi-joint stiffness for prosthetic limb design," *Mechanism and Machine Theory*, vol. 43, pp. 297–309, 03 2008.

[40] A. L. Schwab, "Quaternions, finite rotation and euler parameters," 2002.

[41] J. J. Craig, *Introduction to Robotics: Mechanics and Control*. USA: Pearson Prentice Hall, 3nd ed., 2005.

[42] J. Schmidtler, K. Bengler, F. Dimeas, and A. Campeau-Lecours, "A questionnaire for the evaluation of physical assistive devices (quead): Testing usability and acceptance in physical human-robot interaction," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 876–881, 2017.

[43] C. Harris and D. Wolpert, "Signal-dependent noise determines motor planning.," *Nature*, vol. 394, pp. 780–4, 09 1998.

[44] A. H. Fagg, A. Shah, and A. G. Barto, "A computational model of muscle recruitment for wrist movements," *Journal of Neurophysiology*, vol. 88, no. 6, pp. 3348–3358, 2002. PMID: 12466451.

[45] E. Todorov and M. I. Jordan, "Optimal feedback control as a theory of motor coordination," *Nat Neurosci*, vol. 5, pp. 1226–1235, Nov 2002.

[46] D. W. Franklin, E. Burdet, K. Peng Tee, R. Osu, C.-M. Chew, T. E. Milner, and M. Kawato, "CNS learns stable, accurate, and efficient movements using a simple algorithm," *Journal of Neuroscience*, vol. 28, no. 44, pp. 11165–11173, 2008.

[47] J. J. Kutch, A. D. Kuo, A. M. Bloch, and W. Z. Rymer, "Endpoint force fluctuations reveal flexible rather than synergistic patterns of muscle cooperation," *Journal of Neurophysiology*, vol. 100, no. 5, pp. 2455–2471, 2008. PMID: 18799603.

[48] J. M. Winters, "An improved muscle-reflex actuator for use in large-scale neuromusculoskeletal models," *Annals of Biomedical Engineering*, vol. 23, pp. 359–374, Jul 1995.

[49] J. M. Winters and L. Stark, "Analysis of fundamental human movement patterns through the use of in-depth antagonistic muscle models," *IEEE Transactions on Biomedical Engineering*, vol. BME-32, no. 10, pp. 826–839, 1985.

[50] F. L. Lewis, D. Vrabie, and V. L. Syrmos, *DYNAMIC PROGRAMMING*, ch. 6, pp. 260–286. John Wiley & Sons, Ltd, 2012.

[51] S. Kostova, I. Ivanov, L. Imsland, and N. Georgieva, "Infinite horizon lqr problem of linear discrete time positive systems," *Proceeding of the Bulgarian Academy of Sciences*, vol. 66, 01 2013.