ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

RSL Robotic Systems Lab
Prof. Dr. Marco Hutter

Master Thesis

Interaction Force Control for a Series Elastic Actuated Exoskeleton

Spring Term 2019

Supervised by:
Yves Zimmermann
Dr. Farbod Farshidian

Author:
Emek Barış Küçüktabak

Declaration of Originality

I hereby declare that the written work I have submitted entitled

Interaction Force Control for a Series Elastic Actuated Exoskeleton

is original work which I alone have authored and which is written in my own words.[1]

Author(s)

Emek Barış	Küçüktabak

Student supervisor(s)

Yves	Zimmermann
Farbod	Farshidian

Supervising lecturer

Marco	Hutter

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on 'Citation etiquette' (`https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiarism-citationetiquette.pdf`). The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Zürich, 07.07.2019
Place and date

Signature

[1] Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

Intellectual Property Agreement

The student acted under the supervision of Prof. Hutter and contributed to research of his group. Research results of students outside the scope of an employment contract with ETH Zurich belong to the students themselves. The results of the student within the present thesis shall be exploited by ETH Zurich, possibly together with results of other contributors in the same field. To facilitate and to enable a common exploitation of all combined research results, the student hereby assigns his rights to the research results to ETH Zurich. In exchange, the student shall be treated like an employee of ETH Zurich with respect to any income generated due to the research results.
This agreement regulates the rights to the created research results.

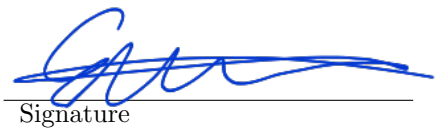
1. Intellectual Property Rights

1. The student assigns his/her rights to the research results, including inventions and works protected by copyright, but not including his moral rights ("Urheberpersönlichkeitsrechte"), to ETH Zurich. Herewith, he cedes, in particular, all rights for commercial exploitations of research results to ETH Zurich. He is doing this voluntarily and with full awareness, in order to facilitate the commercial exploitation of the created Research Results. The student's moral rights ("Urheberpersönlichkeitsrechte") shall not be affected by this assignment.
2. In exchange, the student will be compensated by ETH Zurich in the case of income through the commercial exploitation of research results. Compensation will be made as if the student was an employee of ETH Zurich and according to the guidelines "Richtlinien für die wirtschaftliche Verwertung von Forschungsergebnissen der ETH Zürich".
3. The student agrees to keep all research results confidential. This obligation to confidentiality shall persist until he or she is informed by ETH Zurich that the intellectual property rights to the research results have been protected through patent applications or other adequate measures or that no protection is sought, but not longer than 12 months after the collaborator has signed this agreement.
4. If a patent application is filed for an invention based on the research results, the student will duly provide all necessary signatures. He/she also agrees to be available whenever his aid is necessary in the course of the patent application process, e.g. to respond to questions of patent examiners or the like.

2. Settlement of Disagreements

Should disagreements arise out between the parties, the parties will make an effort to settle them between them in good faith. In case of failure of these agreements, Swiss Law shall be applied and the Courts of Zurich shall have exclusive jurisdiction.

Zürich, 07.07.2019
Place and date


Signature

Contents

Abstract

People with the damaged central system often have problems in their motor functions. With movement therapies, by applying the appropriate forces to the human parts, partial or full recoveries are possible. Conventionally this is done by a therapist; however, there are also many studies focusing on robot-assisted therapy. ANYexo [1] is a torque controllable upper limb exoskeleton as a research platform for robot-assisted therapy. The goal of this project is to develop and investigate force control algorithms for an upper-limb exoskeleton and test them on ANYexo. Due to the safety concerns; first, a simulation environment is created. By using anthropometric tables, human arm properties are parametrized with respect to human height and weight. Interaction between the human and the ANYexo is modeled with spring and damper elements, where spring constant is calculated from the Young's moduli of the fat and muscle. In the simulation environment, the feedforward controller, virtual mass controller, PID controller, single disturbance observer, and cascaded disturbance observer controllers are tested. With cascaded disturbance observer structure, best force control is achieved. On the hardware tests, due to the incomplete sensory system of the ANYexo, cascaded disturbance observer is not tested. Virtual mass controller and single disturbance observer controller show similar performances on the transparency task, by achieving a felt inertia that is 22% and 24% of the felt inertia when feedforward controller is active.

Symbols

Symbols

$\boldsymbol{q}$	joint positions
$\boldsymbol{\tau}$	joint torques
$\boldsymbol{M}$	mass matrix
$\boldsymbol{b}$	centrifugal and Coriolis generalized forces
$\boldsymbol{g}$	gravitational generalized forces
$\boldsymbol{J_c}$	contact Jacobian
$\boldsymbol{\lambda_c}$	contact force
p	plant
$\hat{p}$	model of the plant
q	controller
d	disturbance
$\hat{d}$	disturbance estimation
u	controller output (plant input)
r	reference command
m	mass of the simplified version of the exoskeleton
m_h	mass of the simplified version of the human
m_{virt}	virtual mass
k	spring constant
c	damping constant
F_i	interaction force in the simplified interaction model
p_A	all-pass portion of the plant
p_M	minimum phase portion of the plant
d_A	all-pass portion of the disturbance
d_M	minimum phase portion of the disturbance
f	filter
λ	disturbance observer tuning paramter
T_s	sampling time
σ	singular value
t	time
R	rotation matrix
$\boldsymbol{n}$	rotation direction
θ	amount of rotation
E	Young's modulus
G	Shear modulus

A	contact area
l	thickness of fat or muscle

Indices

x	x axis
y	y axis
z	y axis

Acronyms and Abbreviations

ETH	Eidgenössische Technische Hochschule
RSL	Robotic Systems Lab
ROS	Robot Operating System
HO	hierarchical optimizer
GH	glenohumeral
ED	elevation-depression
PR	protraction-retraction

Chapter 1

Introduction

Loss of motor functions in adults is mostly caused by the damage in the central nervous system. With movement therapy, it is possible to partially or fully recover impaired motor functions [2]. Conventionally, this therapy is provided by a therapist in one-to-one training sessions. However, in the last 20 years, there are many studies focusing on the robot-assisted therapy([1], [3], [4], [5], [6], [7]) and some of them have already shown significant improvement in the therapy outcome ([5], [6]).

During therapy, an exoskeleton can influence human motion in 3 different ways. These are guiding, assisting and correcting. Guiding(Figure 1.1) is controlling the position or orientation of one on the body parts, e.g. arm. This can be achieved by a robot with position control ability. Assisting(Figure 1.2) is applying force to help the human to follow the desired maneuver. When exoskeleton assists, the human requires less effort to move his/her body part. Depending on the need of the patient, the amount of the assistance changes. Another way to influence human motion is correcting(Figure 1.3). To correct, the error of the human body part from the desired configuration is estimated, and a force is applied to decrease this error. For guiding, controlling the positions of the contact points is enough. However, for assisting and correcting, the interaction force between the human and the robot should be controlled.

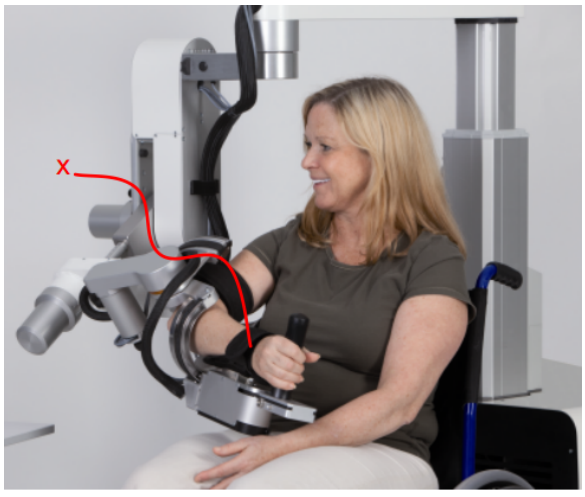


Figure 1.1: Representative figure of guiding. Red curve shows the desired position over time.

Figure 1.2: Representative figure of assisting. White dots shows the desired position over time. Red line represents the assistance by the exoskeleton

Figure 1.3: Representative figure of correcting. White dots shows the desired position over time. Red line represents the correcting force applied by the exoskeleton

About controlling the interaction force between a robot and its environment, there are many studies in the literature. The most basic one is to use a feed-forward controller as in [8]. In this method, only system dynamics is used to compensate for the gravitational and Coriolis effects. Another common way to control the interaction force is by using the admittance control method. This is achieved by using force feedback and controlling the position of the actuators with the purpose of decreasing the force error [9]. There are also more complicated studies which use the disturbance observer structure with an estimated interaction model [10]. Details of this method are extensively discussed in Chapter 3.

In this project, the main focus is to develop and implement methods to control the interaction force between a series elastic actuated exoskeleton and the human. The exoskeleton in concern is briefly introduced in Chapter 2. Different force and position control methods are explained in Chapters 3 and 4, respectively. Then, in Chapter 5, the simulation environment that is used to test the developed control

algorithms is introduced, and the test results of these methods are presented and compared in Chapter 6.

Chapter 2

Related Work

In this chapter, studies that were conducted under Robotic System Lab(RSL) which are related to this thesis are presented. Previously obtained knowledge, tools and the hardware of the lab is used to cumulatively improve the knowledge in this area.

2.1 A Versatile and Dynamic Upper-Limb Rehabilitation Robot: ANYexo

In [1], ANYexo is introduced as follows: "*ANYexo is a versatile upper-limb exoskeleton based on low-impedance torque controllable series elastic actuators. It is designed as an experimental platform to validate novel algorithms and hardware concepts that aim towards more autonomy of robots for therapy of moderately and severely affected patients with neural impairment causing disability in the shoulder and arm movements.*" ANYexo with a user can be seen in Figure 2.1.

Figure 2.1: Exoskeleton arm of ANYexo with a user [1].

The human shoulder has 12 DoFs [11]; however it can be approximated reasonably well with 5 DoFs; these degrees of freedoms are elevation/depression and protraction/retraction of the shoulder girdle, and 3 rotational freedoms of the humerus.

The elbow motion is a simple 1 degree-of-motion. Therefore, ANYexo has 6 degrees-of-freedoms to be able to fully manipulate the human arm; where the first 5 joints to move shoulder girdle and the humerus, and the 6 th joint is for the elbow movement. The overall kinematic structure of the ANYexo is shown in Figure 2.2.

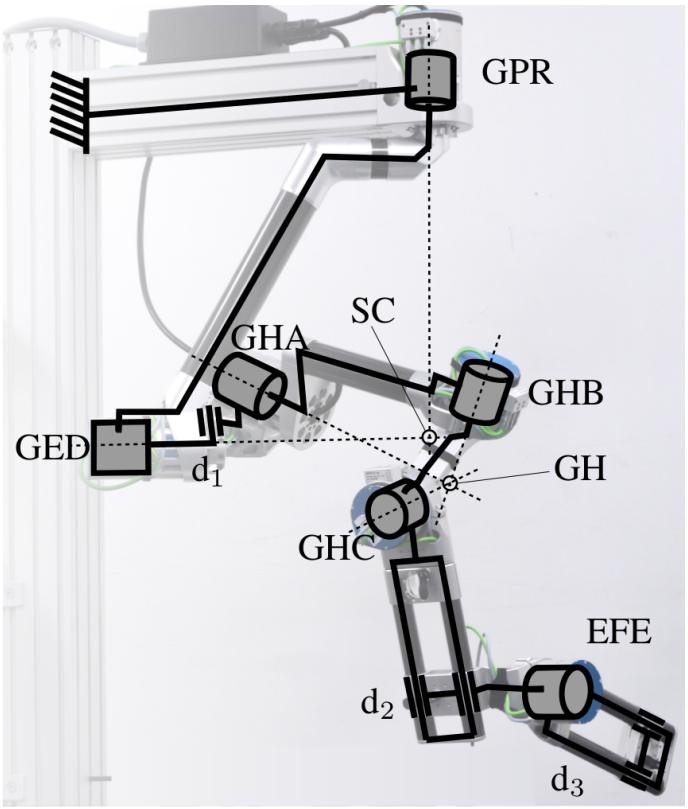


Figure 2.2: The kinematic structure of ANYexo with the shoulder girdle joints (SG): protraction/retraction (GPR) and elevation/depression (GED); the glenohumeral joints (GH): GHA, GHB, and GHC; elbow joint EFE. [1].

The joints of the ANYexo are actuated by the series elastic actuators called ANYdrive [12]. Accurate torque controllability of the ANYdrives allows controlling the interaction force between ANYexo and the human. Moreover, all link elements of ANYexo are developed for high stiffness, low weight and are strong enough to withstand the stress caused by the maximum torque applied by the ANYdrives. The connection between the human and the exoskeleton are obtained with a pair of cuffs placed at the upper arm and forearm as shown in Figure 2.3. At the connection points, 6-DoF F/T-sensors Rokubi Mini 1.1 are placed between the exoskeleton and the cuffs. They measure force and torque range of 1000 N and 8 Nm, respectively in the z-direction and 500 N and 5 Nm, respectively in the xy-plane with noise level less than % 0.02 [1].

The overall structure of the software framework, its interaction with hardware and human is shown in Figure 2.4. The code is written in C++ as nodes of the Robot Operating System(ROS)[1]. It is largely adapted from other robots in RSL(ANYmal[12] and ANYpulator[14]). The templated and modular style of the code requires mod-

[1]Robot Operating System: www.ros.org

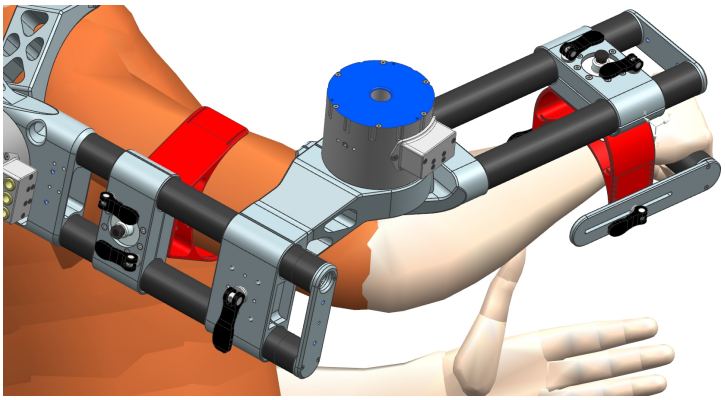


Figure 2.3: Human, robot connection. Cuffs are shown in red [13].

ification of only a few classes to adapt it to a new robot. Thus, new methods can easily be transferred between robots [1].

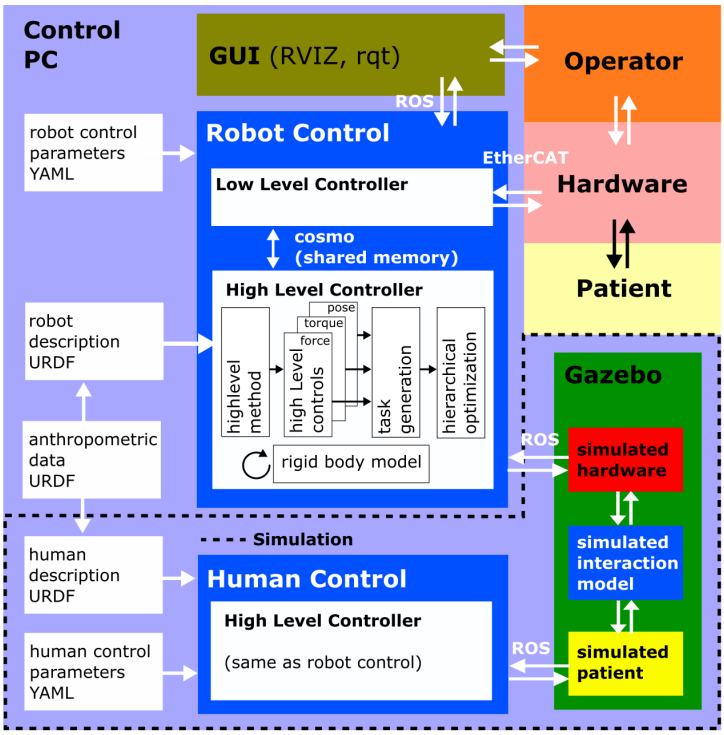


Figure 2.4: Flow chart of the control structure including the simulation [1].

Due to the interaction between the robot and the human, safety is one of the main concerns in the development of the new methods. Therefore, a simulation environment is created for the sake of safety and easiness of the test. Gazebo[2] physics engine is used to simulate the multi-body dynamics and controllers are implemented in a ROS framework as explained above. A screenshot from the simulator is shown in Figure 2.5.

[2]http://gazebosim.org/

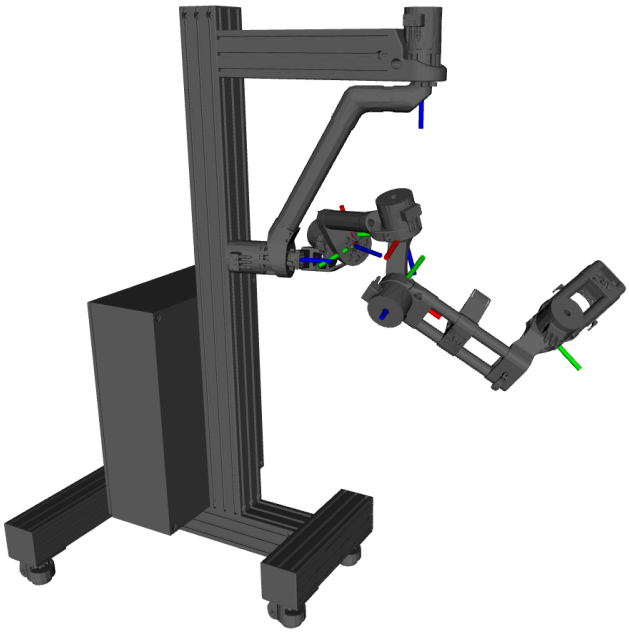


Figure 2.5: A screenshot of the ANYexo from the simulation environment.

2.2 Whole Body Control and Hierarchical Optimization

As explained in Chapter 1, the main focus of the thesis is to control the interaction force. However, while trying to track a reference force, it is very important to consider the safety of the user and the hardware limits of the system. Therefore, during the operation, there are several tasks to be achieved. In [15] and [16], it is shown that the whole body control task formulation coupled with the hierarchical optimization is a very convenient way to find the actuator requirements to satisfy tasks with different priorities as best as possible.

In [15], it is shown that a task $\mathbf{T}$ can be defined as a set of linear equality and/or inequality constraints on the solution vector $\mathbf{x} \in \mathbb{R}^n$:

$$\begin{cases} \mathbf{Ax - b = w} \\ \mathbf{Dx - f} \leq \mathbf{v} \end{cases} \tag{2.1}$$

where $\mathbf{w}$ and $\mathbf{v}$ are slack variables to be minimized. A set of tasks $\mathbf{T_1}, \cdots, \mathbf{T_p}$ can be either solved at the same time, optionally weighted against each other, or in a strict prioritized order. Solving p tasks yields an optimal solution $\mathbf{x}^*$. This vector is chosen as the combination of joint accelerations and joint torques as shown in Equation 2.2, where $\ddot{\boldsymbol{q}}$ and $\boldsymbol{\tau}$ are joint accelerations and joint torques vectors, respectively [15].

$$\mathbf{x}^* = \begin{bmatrix} \ddot{\boldsymbol{q}} \\ \boldsymbol{\tau} \end{bmatrix}, \tag{2.2}$$

Whole body control(WBC) problem is formulated as a quadratic optimization problem composed of linear equality and inequality tasks as in Equation 2.1. These tasks

are prioritized in accordance with their importance. The Hierarchical Optimization framework tries to find the optimal solution at the null-spaces of the more prior tasks. In this way, performances of the higher priority tasks are not affected by the less prior ones. After the optimization problem is solved, desired joint torques, τ, is fed into the ANYdrives.

Some of the common tasks that are used and their formulation are as follows:

Dynamic Consistency: It is vital that the solved desired joint accelerations, torques and the interactions forces are physically consistent. In other words, found parameters should satisfy the equation of motion for multi-body systems:

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{b}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{J_c^T}\boldsymbol{\lambda_c}, \tag{2.3}$$

where $\boldsymbol{\tau}$ is the joint torques, $\boldsymbol{q}$ is the generalized joint coordinates, $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{b}$ is the centrifugal and Coriolis generalized forces, $\boldsymbol{g}$ is the gravity forces, $\boldsymbol{\lambda_c}$ is the external forces that are applied on the robot from contact points and $\boldsymbol{J_c}$ is the Jacobian of contact points. Reformulating with respect to the equality constraint convention in Equation 2.1 leads the following coefficients.

$$\boldsymbol{A} = [-\boldsymbol{M} \quad \boldsymbol{I}], \tag{2.4}$$

$$\mathbf{b} = \boldsymbol{b} + \boldsymbol{g} - \boldsymbol{J_c^T}\boldsymbol{\lambda_c} \tag{2.5}$$

Joint Torque Limits: Optimal solution of the joint torques should not exceed the physical limits of the actuator.

$$\boldsymbol{\tau} \leq \boldsymbol{\tau}_{\max} \tag{2.6}$$

Reformulating to the inequality convention in Equation 2.1:

$$\boldsymbol{D} = [\mathbf{0} \quad \boldsymbol{I}], \tag{2.7}$$

$$\mathbf{f} = \boldsymbol{\tau}_{\max} \tag{2.8}$$

Joint Acceleration Limits: Optimal solution of the joint accelerations should not exceed the physical limits of the actuator.

$$\ddot{\boldsymbol{q}} \leq \ddot{\boldsymbol{q}}_{\max} \tag{2.9}$$

Reformulating to the inequality convention in Equation 2.1:

$$\boldsymbol{D} = [\boldsymbol{I} \quad \mathbf{0}], \tag{2.10}$$

$$\mathbf{f} = \ddot{\boldsymbol{q}}_{\max} \tag{2.11}$$

Other limits such as joint position and velocity can be converted to an acceleration limit by taking once or twice time derivative, respectively.

Task Space Acceleration: In Chapter 3 it is explained that the interaction force control is achieved by controlling the task space accelerations along the desired force control directions. The relation between the task space velocity, $\dot{\boldsymbol{x}}$, and the joint velocities is expressed in Equation 2.12.

$$\dot{\boldsymbol{x}} = \boldsymbol{J_c}\dot{\boldsymbol{q}} \tag{2.12}$$

Taking time derivative:

$$\ddot{\boldsymbol{x}} = \boldsymbol{J_c}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\boldsymbol{c}}\dot{\boldsymbol{q}} \tag{2.13}$$

Reformulating with respect to the equality constraint convention in Equation 2.1 leads the following coefficients.

$$\boldsymbol{A} = [\boldsymbol{J_c} \quad \boldsymbol{0}], \tag{2.14}$$

$$\mathbf{b} = \ddot{\boldsymbol{x}} - \dot{\boldsymbol{J}}_{\boldsymbol{c}}\dot{\boldsymbol{q}}. \tag{2.15}$$

Some modifications are proposed on this task in Section 3.5.4.

Collision Avoidance: Depending on the configuration, the GHB joint might interfere with the head of the user. Therefore, the position of this joint with respect to the base should be controlled. This is done by converting the distance between the joint and the head of the user to a task space acceleration limit. The details of this approach are explained in [1] and in this video[3]

Minimizing Joint Limits: If the defined tasks do not occupy space that is larger or equal to 6 in the task space, there would be infinitely many solutions, where some of them might be arbitrarily large. Therefore, it is a good practice to add a task that would minimize joint accelerations. This is achieved with a task where all of the desired joint accelerations are set to 0.

$$\ddot{\boldsymbol{q}} = 0. \tag{2.16}$$

With the convention in Equation 2.1, this leads to a task with the following coefficients:

$$\boldsymbol{A} = [\boldsymbol{I} \quad \boldsymbol{0}], \tag{2.17}$$

$$\mathbf{b} = \boldsymbol{0}. \tag{2.18}$$

These tasks are defined with the associated priorities that are presented in Table 2.1.

[3] https://youtu.be/F1zXnPDfTgM

Table 2.1: Tasks used during force tracking with the associated priorities

Priority	Task
1	Equation of Motion
2	Joint Torque Limits
2	Joint Position Limits
2	Joint Velocity Limits
2	Collision Avoidance
3	Task Space Acceleration
4	Minimize Contact Forces

Chapter 3

Interaction Force Control

Interaction between the human and the exoskeleton occurs due to the connection cuffs that are located on the upper arm and on the wrist as shown in Figure 2.3. Controlling the interaction force exerted to the exoskeleton from these cuffs, which is equal to the force exerted to the human, decouples the human from the robot. This allows to separate the correcting and guiding from the state of the exoskeleton. In other words, guiding and correcting tasks can take only human into account and output the desired interaction force to be applied to the human.

5 different methods to control the interaction force is explained in the following sections in detail with the increasing complexity. This chapter only covers the theoretical reasoning of these methods; their simulation and hardware results are presented in Chapter 6.

3.1 Feedforward Control

The simplest approach to control the interaction force is to use an open loop feedforward control by using the model and multi-body dynamics of the exoskeleton. Generalized equation of motion(EOM) for a multi-body system is expressed in Equation 2.3.

Without using any sensor, the most can be done is to compensate for the Coriolis, centrifugal and gravitational effects. This can be done by applying torques equal to

$$\boldsymbol{\tau} = \hat{\boldsymbol{b}}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \hat{\boldsymbol{g}}(\boldsymbol{q}) - {\boldsymbol{J_c}}^T \boldsymbol{\lambda_c^*}, \tag{3.1}$$

where $\hat{\boldsymbol{b}}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, $\hat{\boldsymbol{g}}(\boldsymbol{q})$ is our knowledge of the nonlinear and gravitational terms and $\boldsymbol{\lambda_c^*}$ is the desired interaction force. To find the theoretical limit of this method, it is assumed that we have the exact knowledge of the model and Equation 3.1 is substituted into Equation 2.3 to find the resulting interaction force error. This leads to:

$$\boldsymbol{\lambda_{err}} = \boldsymbol{\lambda_c^*} - \boldsymbol{\lambda_c} = -({\boldsymbol{J_c}}^T)^{-1} \boldsymbol{M} \ddot{\boldsymbol{q}}. \tag{3.2}$$

From Equation 3.2, it is clear that the error increases as the acceleration of the robot parts increases. Moreover, due to its open loop structure, it is not possible to compensate for the model mismatches. This method would be feasible to use if the human arm moves slowly and if there is a small mismatch between the robot and its model.

3.2 Virtual Mass Control

In section 3.1, it is shown that the interaction force error is proportional to the mass and the acceleration of the robot. Acceleration is due to the human motion which can not be controlled. However, if the sensed inertia of the robot can be reduced, this will lead to a smaller interaction force error. Felt inertia can be decreased by using the sensed force and setting generalized acceleration to be

$$\ddot{\boldsymbol{q}} = {\boldsymbol{M}_{\text{virt}}}^{-1} {\boldsymbol{J}_{\boldsymbol{c}}}^{T} (\boldsymbol{\lambda_c} - \boldsymbol{\lambda_c^*}), \tag{3.3}$$

where $\boldsymbol{M}_{\text{virt}}$ is the desired virtual(felt) inertia of the robot. Substituting Equation 3.3 into Equation 2.3 leads the generalized torques to be

$$\boldsymbol{\tau} = \boldsymbol{M}\,{\boldsymbol{M}_{\text{virt}}}^{-1} {\boldsymbol{J}_{\boldsymbol{c}}}^{T} (\boldsymbol{\lambda_c} - \boldsymbol{\lambda_c^*}) + \boldsymbol{b} + \boldsymbol{g} - {\boldsymbol{J}_{\boldsymbol{c}}}^{T} \boldsymbol{\lambda_c}. \tag{3.4}$$

Assuming the model of the robot is perfect and substituting Equation 3.4 into Equation 2.3, interaction force error is found as

$$\boldsymbol{\lambda_{err}} = \boldsymbol{\lambda_c^*} - \boldsymbol{\lambda_c} = -({\boldsymbol{J}_{\boldsymbol{c}}}^{T})^{-1} \boldsymbol{M}_{\text{virt}} \ddot{\boldsymbol{q}}. \tag{3.5}$$

This result is very similar to Equation 3.2. The only difference is, this time force error can be reduced by setting a lower virtual mass. Fundamentally, this method resembles controlling generalized acceleration to track a reference interaction force with a P controller, where P equals the inverse of the virtual mass matrix, $\boldsymbol{M}_{\text{virt}}$. One limit for decreasing the virtual mass(increasing the P) is the actuator saturation. The other is the sampling time. As the sampling time increases, even if no actuator saturation occurs, the minimum virtual mass that leads to stable response increases. If actuators allow high generalized accelerations and if high sampling rate can be obtained, this method could give very satisfying performances.

3.3 PID Control

The method explained in Section 3.2 only uses the instantaneous force error and does not use its values over time or its change of rate. These pieces of information can be used by the integrative and derivative action of a PID controller to improve the performance.

Our aim is to control the task space acceleration for the eventual purpose of controlling the interaction force. Therefore, a PID controller is designed such that it takes the interaction force error as an input and outputs the desired task space acceleration, $\ddot{x}^*$ as shown in Equation 3.6. Desired task space acceleration is fed into the Hierarchical Optimizer as explained in Section 2.2.

$$\ddot{x}^* = PID(\boldsymbol{\lambda_c^*} - \boldsymbol{\lambda_c}) \tag{3.6}$$

Same limits due to actuator saturation and sampling time also occur in this method as in the virtual mass control. Moreover, due to the integrative action, saturation of the actuators or not being able to apply the commanded acceleration would lead to wind up problems and unstable response. Another limit for this method comes from the number of parameters to tune. To control force along total of 6 directions, there would be 18 parameters to tune. If parameters for each task space directions could be tuned alone, this would not be a problem. However, due to the serial structure of ANYexo, acceleration of one direction depends on another. Thus, tuning of the parameters should be done in a holistic way rather than by tuning based on their individual responses, which makes it quite complicated and very hard to tune. This

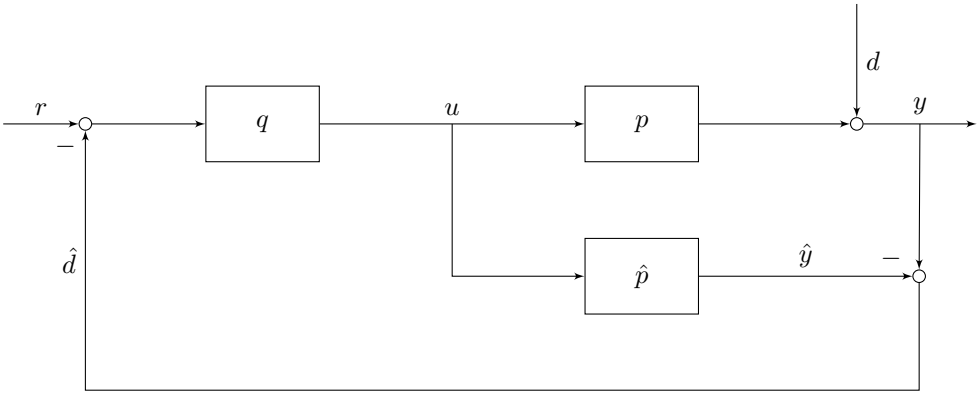


Figure 3.1: Block diagram of a typical IMC structure. q: controller, p: plant, $\hat{p}$: knowledge of the plant, r: reference y: output, $\hat{y}$: estimated output, d: disturbance and $\hat{d}$: estimated disturbance.

method makes sense to use if the interaction force is desired to be controlled along one or two axes.

3.4 H_2 Optimal Control with Internal Model Control Structure

In Section 3.3, it is explained that PID control can obtain good performance; however for task spaces with high dimensions, it is quite hard to tune, and its closed-loop stability is not guaranteed when there are model mismatches and actuator saturation. Moreover, the interaction model between the human and the robot is not explicitly used in that controller. To overcome these problems, the Internal Model Control(IMC) structure and its design are presented in this section. For the sake of simplicity and ease of implementation, control of each direction of the interaction forces are handled separately. This allows us to implement the Single Input Single Output(SISO) control structure rather than the much more complicated Multi Input Multi Output(MIMO) control structure.

3.4.1 SISO IMC Structure

As explained in [17], the fundamental difference of the IMC structure from the classical feedback structure is the fact that the controller takes the estimated disturbance as input rather than the measured error. Therefore, in literature, this structure is also called as the disturbance observer structure. Block diagram of a typical IMC structure is presented in Figure 3.1.

The logic behind the IMC structure is quite straight-forward. The purpose is to estimate the disturbances that can not be directly observed and to compensate for them. First, by using the plant model and the output of the controller, the expected output is calculated. Then, the difference between the measured output and the estimated output gives the disturbance that needs to be compensated. Robustness of this structure comes from the fact that model mismatches can also be considered as a disturbance which is compensated by the controller.

3.4.2 Plant

As explained in Section 3.3, it is desired to control the task space interaction forces by controlling the accelerations along those directions. Therefore, the output of the controller gives the desired task space acceleration, $\ddot{x}$ which is to be fed into the Hierarchical Optimizer(Section 2.2). Moreover, the plant of the system is described by the interaction model between the human and the robot. It defines the relationship between the task space acceleration and the interaction force along that direction.

Interaction model between the human and the exoskeleton is modeled as a spring-damper system similar to the previous works of [18], [19], [20], [21], [22] and [23].

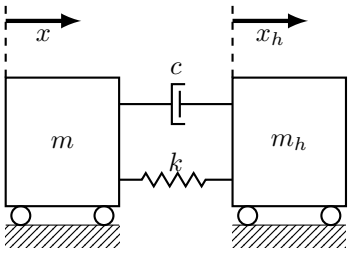


Figure 3.2: Simplified visualisation of the interaction model between human and the exoskeleton. Subscript h represents the human; where as parameters with no subscripts belong to the exoskeleton. m: mass, x: displacement, k: spring constant, c: damping constant.

To describe the plant, the relation between the interaction force and the motion of the exoskeleton is investigated. The interaction force between the human and the exoskeleton can be expressed as:

$$F_i = k(x_h - x) + c(\dot{x}_h - \dot{x}), \tag{3.7}$$

where F_i is the interaction force. Taking time derivative twice and rearranging leads to the following equation

$$\ddot{F}_i = -ka - c\dot{a} + ka_h + c\dot{a}_h, \tag{3.8}$$

where a is the task space acceleration, $\ddot{x}$. Converting Equation 3.8 from time domain to s domain:

$$s^2F_i(s) = -ka(s) - sca(s) + ka_h(s) + sca_h(s). \tag{3.9}$$

This leads the plant to be:

$$P(s) = \frac{F_i(s)}{a(s)} = \frac{-k - sc}{s^2}. \tag{3.10}$$

Because human is independent and there are no direct measurements of its motion, the last 2 terms in Equation 3.8 is considered as a disturbance.

Even though the interaction is modeled as a spring-damper system, in reality, there would also be a delay due to the static friction and the gap at the connection between the human and the robot. Therefore, damping and the other mentioned factors are approximated as a pure delay. The amount of the delay(θ) can be tuned by examining the real hardware performance. As an initial step, the amount of

delay can be approximated by c/k. A system with high damping leads to more delay; whereas, high spring constant leads to a faster response.

$$P(s) \simeq \frac{-k}{s^2} e^{-\theta s} \tag{3.11}$$

Considering the Hierarchical Optimizer and the approximated plant, equivalent of the structure in Figure 3.1 is presented in Figure 3.3

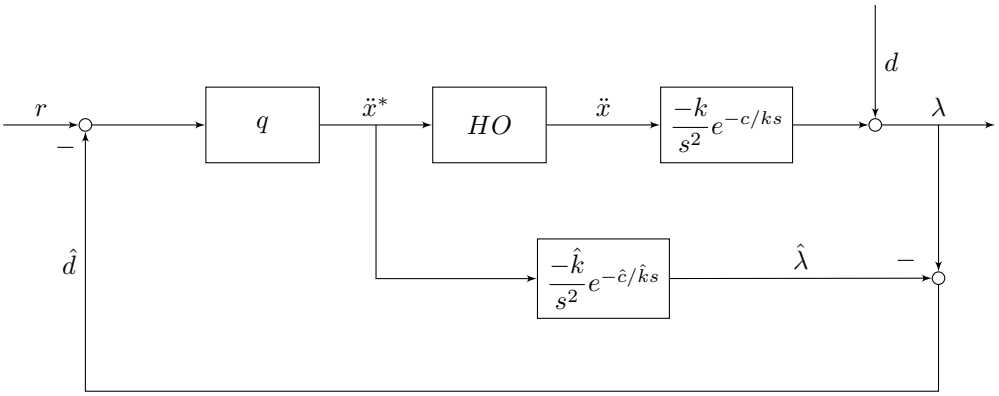


Figure 3.3: SISO IMC implementation for interaction force control. λ: interaction force, $\hat{\lambda}$: estimation of the interaction force, $\hat{k}$: knowledge of the spring constant, $\hat{c}$: knowledge of the damping constant

In the structure presented in Figure 3.3, the model block($\hat{p}$) takes the desired acceleration as input rather than the actual acceleration. This allows observing the effect of the acceleration tracking error as a disturbance. Moreover, if the model block takes the actual acceleration as input, this would lead to an unstable response. This can be seen by examining the transfer function from u to y in Figure 3.1. This transfer function is equal to p, which is unstable due to its double poles at the origin.

3.4.3 Controller Design

After the plant is found, and the necessary approximations are done, the design of the controller can be started. During the design of the Internal Model Controller, procedures explained in Chapter 4 and Chapter 5 of [17] are taken as references.

The controller is designed in 2 main steps. First, H_2 optimal controller is found for the nominal model. Then, to make the controller causal and to ensure robustness for model mismatches, a low-pass filter is implemented.

H_2 Optimal Controller for the Nominal Model

H_2 optimal controller is the controller that minimizes the Euclidean norm norm of the error over time as expressed in Equation 3.12

$$\tilde{q} = \min_{\tilde{q}} ||e||_2^2 = \int_0^\infty e^2(t)dt, \tag{3.12}$$

where e is the interaction force error λ_{err} in our case.

Plant and the disturbance can be factorized into its minimum-phase and all pass portions as in Equations 3.13 and 3.14

$$\tilde{p} = \tilde{p}_A \tilde{p}_M \tag{3.13}$$

$$\tilde{d} = d_A d_M \tag{3.14}$$

where subscript M means the minimum phase portion and subscript A describes the all-pass portion.

It is explained and proved in [17] that H_2 optimal controller for a plant $\tilde{p}$ and a disturbance d equals to:

$$\tilde{q} = (\tilde{p}_M d_M)^{-1} \{\tilde{p}_A^{-1} d_M\}_*, \tag{3.15}$$

where the operator $\{\bullet\}_*$ denotes that after a partial fraction expansion of the operand all terms involving the poles of $\tilde{p}_A^{-1}$ are omitted.

One should be careful about the convention of this formulation, where the disturbance d is added after the plant. Therefore, if one wants to take disturbances that are before the plant into account, its effect after the plant should be used as d. As a design criterion, it is decided that the controller should be able to compensate for the step errors on the acceleration tracking. Because Hierarchical Optimizer is before the plant, $\frac{1}{s}$ should be multiplied by $\frac{1}{s^2}$ and $\frac{1}{s^3}$ should be used as the disturbance in the Equation 3.15.

Factorizing $\tilde{p}$ to its portions is quite straight-forward. Its minimum phase portion equals to:

$$\tilde{p}_M = \frac{-k}{s^2}, \tag{3.16}$$

whereas, its all pass portion equals to:

$$\tilde{p}_A = e^{-\theta s} \simeq \frac{-s + 2/\theta}{s + 2/\theta}. \tag{3.17}$$

Substituting $\tilde{p}_M$, $\tilde{p}_A$ and d_M into Equation 3.15:

$$\tilde{q} = (\frac{-k}{s^2}\frac{1}{s^3})^{-1} \{\frac{s + 2/\theta}{-s + 2/\theta}\frac{1}{s^3}\}_*, \tag{3.18}$$

Solving the $\{\bullet\}_*$ operation in Equation 3.18:

$$\tilde{q} = (\frac{-k}{s^2}\frac{1}{s^3})^{-1} (\frac{1}{s^3} + \frac{\theta}{s^2} + \frac{0.5\theta^2}{s}), \tag{3.19}$$

Rearranging the terms in Equation 3.19:

$$\tilde{q} = \frac{-s^2(0.5\theta^2 s^2 + \theta s + 1)}{k} \tag{3.20}$$

It is clear that the calculated H_2 optimal controller is not causal and can not be implemented. Therefore, a low-pass filter that would make the controller causal is implemented. This filter not only makes the controller causal but also gives robustness against the mismatch between the plant and the model.

Filter design

To have an implementable controller that asymptotically rejects a specified input, the filter should satisfy the following requirements:

(i) The controller $q = \tilde{q}f$ must be proper.

(ii) $(1 - p\tilde{q}f)d$ has to be stable for the asymptotic rejection of the disturbances.

As in [17], a filter of the form

$$f(s) = \frac{a_{k-1}s^{k-1} + \ldots + a_1 s + a_0}{(\lambda s + 1)^{m+k-1}} \tag{3.21}$$

where λ is the time constant of the filter, is used. In Equation 3.20, it can be seen that zero-pole excess of the nominal controller is 4. Therefore m should be at least 4 to satisfy the requirement (i). For unstable disturbances (in our case: $\frac{1}{s^3}$) with a pole of multiplicity l at π_i, the filter has to satisfy

$$f(\pi_i) = 1 \tag{3.22}$$

and

$$\frac{d^j f(s)}{ds^j} = 0 \qquad j = 1, \ldots, l-1 \tag{3.23}$$

for the requirement (ii). Because there is a multiplicity of 3, k value should also be 3. Inserting m and k into the Equation 3.21 leads the filter to be in the form of

$$f(s) = \frac{a_2 s^2 + a_1 s + a_0}{(\lambda s + 1)^6}. \tag{3.24}$$

a_0, a_1 and a_2 are calculated from the 3 conditions coming from Equations 3.22 and 3.23. The final version of the filter that satisfies all of the requirements is as follows:

$$f(s) = \frac{15\lambda^2 s^2 + 6\lambda s + 1}{(\lambda s + 1)^6}. \tag{3.25}$$

The only parameter to tune is the time constant, λ of the filter. As λ is decreased, the controller acts faster and its performance gets better. For the nominal model, stability is guaranteed for every λ that is greater than zero. However, when there are discrepancies between the plant and the model, there is a minimum λ value that will yield a stable response.

To sum up H_2 optimal controller at Equation 3.20, and the low pass filter at Equation 3.25 are designed for the structure presented in Figure 3.3.

Implementation

In theory, the structure presented in Figure 3.3 is stable as long as the model of the plant takes the desired acceleration as the input. However, in practice, it is not the case. Because of the unstable structure of the plant model, small numerical errors grow in time and make the response unstable. To overcome this issue, an equivalent system of the classical feedback structure is used as in Figure 3.4.
If

$$c = \frac{q}{1 - \tilde{p}\tilde{q}f}, \tag{3.26}$$

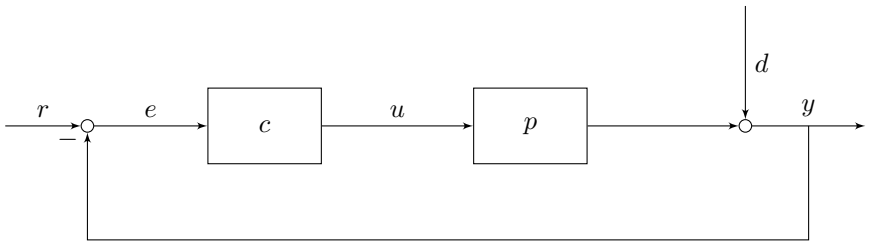


Figure 3.4: Block diagram of a classical feedback structure. c: controller, e: error.

systems at Figure 3.1 and 3.4 are identical.

To sum up, the controller is designed by using the IMC structure; however, the equivalent classical feedback structure is implemented for practical reasons.

3.5 H_2 Optimal Control with Cascaded IMC Structure

In Section 3.4, the design of the H_2 optimal control by using the IMC structure is presented. Recall that the interaction force is controlled by setting the desired acceleration and feeding this information to the Hierarchical Optimization as an input. Even though the interaction force controller is designed to overcome step errors on acceleration tracking, if the commanded accelerations are controlled, interaction tracking performance can be improved. Therefore, another IMC structure is implemented as an inner loop to track the desired task space acceleration.

Because there is no information on the behavior of the reference interaction force, dynamics of the reference and disturbance are assumed to be similar in section 3.4. In other words, the interaction force controller is designed to track step references and reject step disturbances with the same time constant. However, in the cascaded structure, the reference of the inner loop(acceleration) controller is the output of the outer(interaction force) controller. Therefore, 2 separate controllers to track acceleration reference and to reject disturbances on acceleration tracking are implemented.

3.5.1 Two-Degree-of-Freedom IMC structure

If it is known that the dynamics of the reference signal and the disturbances are not similar, better performance can be obtained by implementing the 2 degree-of-freedom controller structure which is presented in Figure 3.5.

Having different controllers for disturbance rejection and reference tracking allows us to design and tune them separately. While a single controller used for the interaction force control, a 2-degree-of-freedom structure is used for the acceleration control. The overall cascaded structure can be seen in Figure 3.6.

3.5.2 Plant

The plant of the outer loop to control the interaction force is explained in Section 3.4.2. For the inner loop, the input of the plant is the desired task space acceleration to the hierarchal optimization and the output is the obtained task space acceleration. Moreover, what is expected from the hierarchal optimization is that

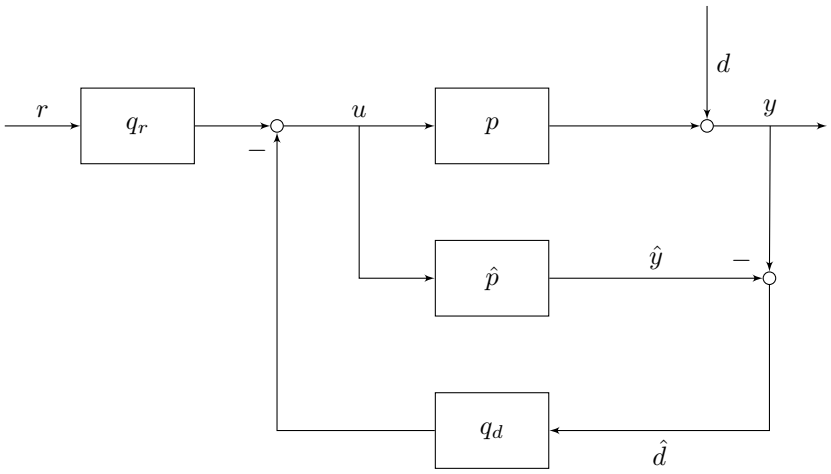


Figure 3.5: IMC structure with separate controllers for reference tracking and disturbance rejection. q_d: controller for disturbance rejection, q_r: controller for reference tracking.

the desired acceleration is realized in a one-time step. Therefore, the plant model for the acceleration loop equals to one-time step pure delay as shown in Equation 3.27.

$$\hat{p}_{\ddot{x}} = e^{-T_s s}, \tag{3.27}$$

where T_s is the time step of the controller in seconds.

3.5.3 Controller Design

At the outer loop, the same controller which is designed for the single disturbance observer structure is used. For the inner loop, the controller is designed by using the same 2 step approach which is explained in Section 3.4.3 in detail.

H_2 Optimal Controller for the Nominal Model

Recall that the H_2 optimal controller is the controller that minimizes the 2-norm of the error over time as expressed in Equation 3.12. For a nominal plant $\tilde{p}$ and a disturbance d, the optimal controller is expressed in Equation 3.15.

Because the acceleration plant only consists of a delay element, it has no minimum-phase portion as shown in Equation 3.28 and 3.29.

$$\tilde{p}_M = 1, \tag{3.28}$$

$$\tilde{p}_A = e^{-T_s s} \simeq \frac{-s + 2/T_s}{s + 2/T_s}., \tag{3.29}$$

As a design criterion, it is decided to design the controller such that it is H_2 optimal to ramp references and disturbances. Therefore, $d_M = 1/s^2$, $\tilde{p}_M$ and $\tilde{p}_A$ in Equations 3.28, 3.29 are inserted in Equation 3.15 to find the H_2 optimal controller. This leads,

$$\tilde{q} = T_s s + 1. \tag{3.30}$$

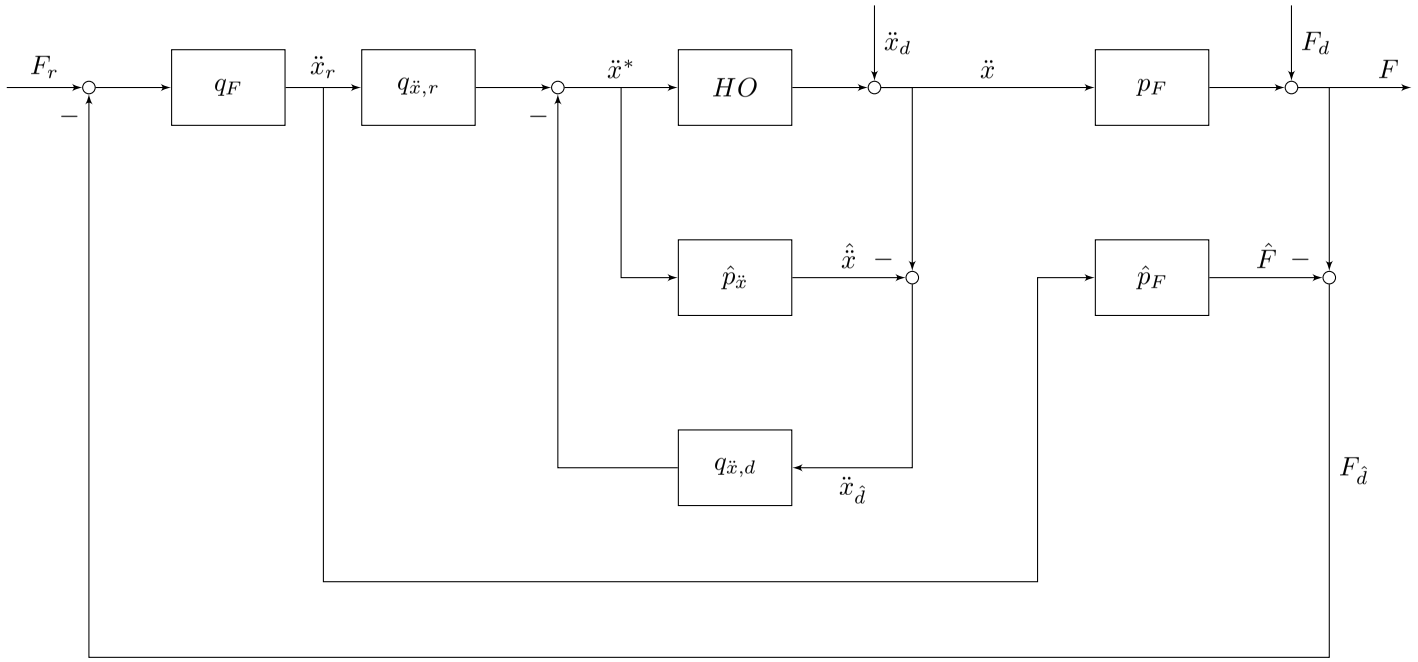


Figure 3.6: Cascaded IMC structure. The inner loop controls the acceleration; where as, the outer loop controls the interaction force. q_F: force controller for both disturbance rejection and reference tracking, $q_{\ddot{x},r}$: acceleration controller for reference tracking, $q_{\ddot{x},d}$: acceleration controller for disturbance rejection, HO: hierarchical optimizer, takes desired acceleration as input, $\hat{p}_{\ddot{x}}$: model of the hierarchical optimizer, p_F: plant describing the relation between the acceleration and the interaction force, $\hat{p}_F$: model of the plant p_F. For the signals, F and $\ddot{x}$ means interaction force and acceleration, respectively. Subscript r and d indicates reference and disturbance, respectively. Hat means the knowledge of a signal. For example, $\ddot{x}_{\hat{d}}$ indicates our knowledge of the disturbance on the acceleration.

It is clear that the found optimal controller in Equation 3.30 is non-causal; thus, not implementable. Therefore, as in Section 3.4.3, a low-pass filter is implemented.

Filter design

Recall from Section 3.4.3 that the controller with the filter should satisfy the following conditions:

(i) The controller $q = \tilde{q}f$ must be proper,

(ii) $(1 - p\tilde{q}f)d$ has to be stable for the asymptotic rejection of the disturbances,

and the filter has the form shown in Equation 3.21.

Because the H_2 optimal controller has a zero-pole excess of 1 and the ramp disturbance in consideration has double poles at the origin, m and k values in Equation 3.21 are set to 1 and 2, respectively. By solving the Equations 3.22 and 3.23 simultaneously, the filter to be used is found as

$$f = \frac{2\lambda s + 1}{(\lambda s + 1)^2}, \tag{3.31}$$

which leads to

$$q_{\ddot{x},d} = q_{\ddot{x},r} = \tilde{q}f. \tag{3.32}$$

Even though the controllers for the disturbance rejection and reference tracking are equivalent, they are implemented as different blocks to be able to tune them separately.

3.5.4 Jacobian Reduction

The controllers that are introduced in Section 3.1 - 3.5 works in Single-Input-Single-Output(SISO) manner. In other words, control loops only use force and accelerations measurements along one direction, and their output does not depend on the measurements along other directions. This approach simplifies the controller design procedure; however, if the robot has less controllability than the number of control loops running, this would decrease the controller performance. This is due to the fact that the effect of less controllability would also be seen as a disturbance and excessive outputs would be tried to achieve, which is not desired. To overcome this issue, inside the hierarchical optimization block, a strategy that we call as a *Jacobian reduction* is used. This method allows creating extra nullspace for the lower priority tasks in addition to not trying tasks along low controllability.

The singular value decomposition of the contact Jacobian gives valuable information regarding the controllability of an exoskeleton.

$$J_{m\times n} = U_{m\times m}\Sigma_{m\times n}V^T_{n\times n} \tag{3.33}$$

where m is the number of directions that we try to control and n is the number of actuators of the exoskeleton. U and V are orthogonal and Σ is a diagonal matrix. The structure of those matrices are shown in Equation 3.34, 3.35 and 3.36.

$$U = [\mathbf{u_1} \quad \mathbf{u_2} \quad \dots \quad \mathbf{u_m}] \tag{3.34}$$

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & \dots & 0 & \\ 0 & \sigma_2 & \dots & 0 & \\ \vdots & \vdots & \ddots & \vdots & 0^{m \times \max(n-m,0)} \\ 0 & 0 & \dots & \sigma_{\min(m,n)} & \\ & & 0^{\max(m-n,0) \times n} & & \end{bmatrix} \tag{3.35}$$

$$V = [\mathbf{v_1} \quad \mathbf{v_2} \quad \dots \quad \mathbf{v_n}] \tag{3.36}$$

The largest magnitude output is obtained along $\mathbf{v_1}$. The output is then along $\mathbf{u_1}$ with magnitude $\sigma\mathbf{u_1}$, which is the direction with the largest controllability of the robot. On the other hand, the direction of $\mathbf{u_m}$ requires the most effort form the actuators; thus, is the direction with the lowest controllability. Therefore, the matrix U is actually a rotation matrix between the control space and the singularity space whose first and last element represent the most and the least controllable direction, respectively.

The equality constraint for desired task space acceleration in body frame can be written as in Equation 3.37

$$_B J\ddot{q} = {}_{\mathbf{B}}\ddot{\mathbf{x}} - {}_B\, \dot{J}\dot{q} := b \tag{3.37}$$

It is desired to convert this task to singularity space from body space. Moreover, in the singularity frame, components with low controllability should either be smoothed or completely eliminated. The smoothing can be done by smoothing the σ values in the matrix Σ. The smoothed $\tilde{\Sigma}$ matrix is shown in Equation 3.38

$$\tilde{\Sigma} = \begin{bmatrix} c_1\sigma_1 & 0 & \dots & 0 & \\ 0 & c_2\sigma_2 & \dots & 0 & \\ \vdots & \vdots & \ddots & \vdots & 0^{\min(m,n) \times n - \min(m,n)} \\ 0 & 0 & \dots & c_{\min(m,n)}\sigma_{\min(m,n)} & \end{bmatrix}, \tag{3.38}$$

where c_i is the smoothing factor and $c_i(\sigma_i)$ satisfies the following conditions at threshold values of t_1 and t_2:

(i) $c_i(\sigma_i <= t_1) = 0$,

(ii) $c_i(\sigma_i >= t_2) = 1$,

(iii) $c_i^{'}(\sigma_i = t_2) = 0$.

When singular values are between the two threshold values, the smoothing factor can be found by finding the 2$^{\text{nd}}$ order polynomial that satisfies these 3 conditions. This is shown in Equation 3.39 and in Figure 3.7.

$$c_i = \frac{-1}{(t_1 - t_2)^2}\sigma_i^2 + \frac{2t_2}{(t_1 - t_2)^2}\sigma_i + \frac{t_1(t_1 - 2t_2)}{(t_1 - t_2)^2}, \quad t_1 < \sigma_i < t_2 \tag{3.39}$$

By using the modified $\tilde{\Sigma}$ matrix, smoothed body space Jacobian is calculated in Equation 3.40

$$_B\tilde{J} = U\tilde{\Sigma}V^T \tag{3.40}$$

Recall that U is a rotation matrix from singularity frame to body frame. Therefore, body space Jacobian can be converted to a Jacobian in singularity space by

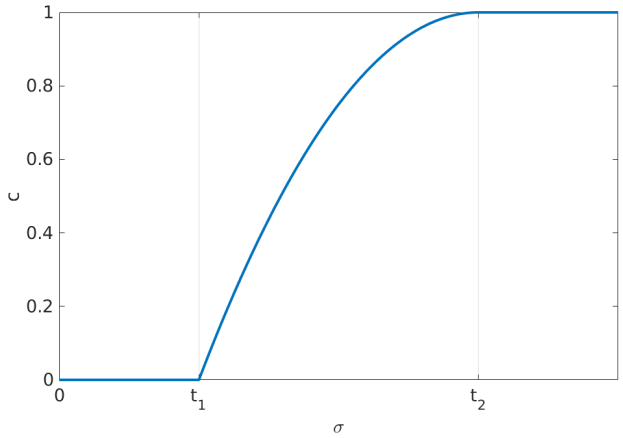


Figure 3.7: Smoothing Factor vs Singular Value

multiplying it with U^{-1}. After the conversion, the task expressed in Equation 3.37 would be equivalent to

$$\tilde{\Sigma} V^T = U^{-1}({}_{\mathbf{B}}\ddot{\mathbf{x}} - {}_B\, \dot{J}\dot{q}). \tag{3.41}$$

By defining the tasks in the singularity frame and smoothing, it is not tried to achieve impossible tasks; moreover, null-space of the task becomes larger, which increases the performance of the lower priority tasks.

Chapter 4

Position Control

In Chapter 3, interaction force control strategies are explained. By controlling the force between the human and ANYexo, assisting or correcting tasks can be achieved. Even though the main focus of the project is on the force control, some position control methods are also developed to guide the human arm.

4.1 Trajectory Generation

When position commands are given, one must be sure that the commanded values are continues. Non-continues position commands would lead to very fast motions which is undesired and not safe. These commands can be in joint space, task space or ISB space which is explained detailly in Section 4.2. The purpose of the trajectory generator is to create a smooth and continues trajectory from the current position to the target position in any control space.

For a trajectory to be smooth, in addition to the position; velocity and acceleration should also be continues. Assuming the initial and target velocity and accelerations are zero, smooth trajectory requirements brings the following boundary conditions for the desired position with respect to time, $r(t)$, in any control space(joint, task, ISB).

(i) $\boldsymbol{r}(0) = \boldsymbol{r_o}$,

(ii) $\dot{\boldsymbol{r}}(0) = 0$,

(iii) $\ddot{\boldsymbol{r}}(0) = 0$,

(iv) $\boldsymbol{r}(t_{\text{final}}) = \boldsymbol{r}_{\text{target}}$,

(v) $\dot{\boldsymbol{r}}(t_{\text{final}}) = 0$,

(vi) $\ddot{\boldsymbol{r}}(t_{\text{final}}) = 0$,

where $\boldsymbol{r_o}$ is the initial position. A normalized interpolation factor with respect to time, $s(t)$, is used to determine the desired instantaneous position. Value of s changes from 0 to 1 as time passes. The normalization is done such that when s is 0, the desired position is the initial position, and when it is 1, the desired position is the target position. This can be seen in Equation 4.1

$$\boldsymbol{r}(t) = \boldsymbol{r_o} + s(t)(\boldsymbol{r}_{\text{target}} - \boldsymbol{r_o}). \tag{4.1}$$

$s(t)$ should be designed such that Equation 4.1 satisfies all the conditions from (i) to (vi). Because there are 6 constraints, a 5^{th} order polynomial with respect to the normalized time is used as shown in Equation 4.2.

$$s(t) = a\hat{t}^5 + b\hat{t}^4 + c\hat{t}^3 + d\hat{t}^2 + e\hat{t} + f, \tag{4.2}$$

where, $\hat{t}$ is the normalized time expressed in Equation 4.3

$$\hat{t} = \frac{t}{t_{\text{final}}}, \tag{4.3}$$

and t_{final} is the maneuver time set by the user. By using the 6 equations from (i) to (vi), 6 unknown coefficients from a to f are solved simultaneously and the solution is presented in Equation 4.4. Note that, using the normalized time in the design of s, simplifies the problem significantly, and allows the coefficients to become constant.

$$s(t) = 6\hat{t}^5 - 15\hat{t}^4 + 10\hat{t}^3 \tag{4.4}$$

Depending on the application, desired velocity or accelerations can be also used. These can be obtained simply by taking the time derivative of Equation 4.1.

$$\dot{\boldsymbol{r}}(t) = \dot{s}(t)(\boldsymbol{r}_{\text{target}} - \boldsymbol{r}_{\mathbf{o}}). \tag{4.5}$$

$$\ddot{\boldsymbol{r}}(t) = \ddot{s}(t)(\boldsymbol{r}_{\text{target}} - \boldsymbol{r}_{\mathbf{o}}). \tag{4.6}$$

where $\dot{s}$ and $\ddot{s}$ are expressed in Equation 4.7 and 4.8

$$\dot{s}(t) = s(t) = 30\hat{t}^4 - 60\hat{t}^3 + 30\hat{t}^2 \tag{4.7}$$

$$\ddot{s}(t) = s(t) = 120\hat{t}^3 - 180\hat{t}^2 + 60\hat{t} \tag{4.8}$$

To sum up, by using the presented trajectory generator, smooth and continues maneuvers can be obtained in any control space.

4.2 Control in ISB Space

The human arm has a mechanically quite complex structure. It can be thought that there are two links: shoulder combined with the upper arm and forearm. The shoulder alone has 12 DoFs [11]; however, it can be approximated reasonably well with 5 DoFs. These degrees of freedom are elevation/depression and protraction/retraction of the shoulder girdle; and 3 rotational freedoms of the humerus. This approximation is explained in [13] in detail and it is shown in Figure. 4.1.

Elevation/depression and protraction/retraction movements can be mapped directly to the motion of GPR and GED joints. However, the placement of the GHA, GHB and GHC joints are not intuitive to describe the orientation of the humerus. To describe these 3 rotational DoFs, in literature and clinics mostly ISB coordinate system is used [25]. This convention is also more intuitive for the users and is shown in Figure 4.2.

To control the position of the ANYexo in ISB Coordinate space, conversion between ISB coordinates and joint coordinates are done. This is achieved by first symbolically calculating the orientation of the humerus with respect to the world frame with ISB angles and joint angles by forward kinematics. Then, these two rotations are equalized to find the unknown parameters.

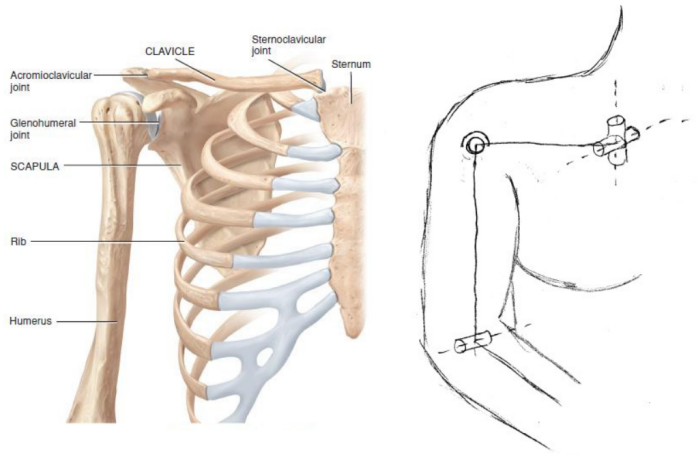


Figure 4.1: The human shoulder [24] and its approximation [13]

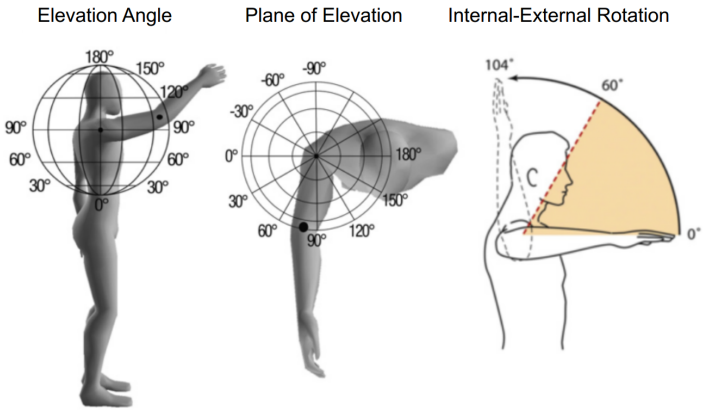


Figure 4.2: ISB coordinate convention

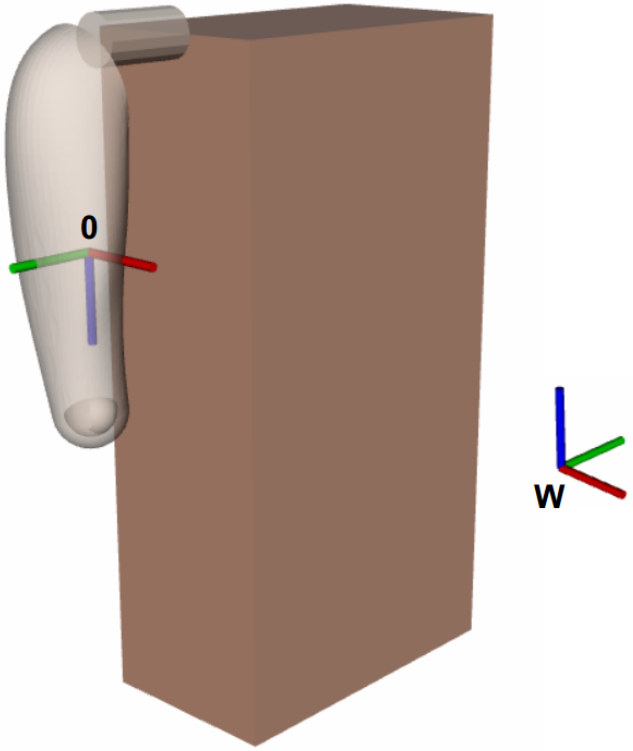


Figure 4.3: Zero configuration in ISB convention. Red: x, Green: y and Blue: z.

Resulting orientation of the upper arm with respect to ISB coordinates is found by consecutive rotations. The orientation of the humerus when all ISB angles are set to 0 can be seen in Figure 4.3. This is due to the zero definition in Figure 4.2. The initial rotation is from world to zero configuration and expressed in Equation 4.9.

$$R_{W0} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{4.9}$$

The frame names are numbered from 1 to 3, after the consecutive rotations: plane of elevation, elevation angle and internal-external rotation. The purpose is to calculate the rotation from world to frame 3, R_{W3} in terms of ISB angles.

After initial rotation matrix is found, the other rotations are done by using the angle axis convention which is shown in Equation 4.10, where θ is the amount of rotation in the unit direction of $\boldsymbol{n}$ defined in the previous frame.

$$\chi = \begin{pmatrix} \theta \\ \boldsymbol{n} \end{pmatrix} \tag{4.10}$$

Changing plane of elevation, from frame 0 to frame 1:

$$\boldsymbol{n}_{01} = \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \quad \theta_{01} = \theta_{\text{plane of elevation}} \tag{4.11}$$

Elevation Angle, from frame 1 to frame 2:

$$\boldsymbol{n}_{12} = \begin{bmatrix} -1 \\ 0 \\ 0 \end{bmatrix} \quad \theta_{12} = \theta_{\text{elevation angle}} \tag{4.12}$$

Internal-external rotation, from frame 2 to frame 3:

$$\boldsymbol{n}_{23} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad \theta_{23} = \theta_{\text{internal-external rotation}} \tag{4.13}$$

Rotations represented in angle axis convention are converted to rotation matrix to easily calculate the overall rotation as shown in Equation 4.14, where subscript H means the frame of the humerus.

$$R_{WH,\text{ISB}} = R_{W0}R_{01}R_{12}R_{23} \tag{4.14}$$

Calculating the rotation of the humerus by joint angles is quite straight forward due to the serial linkage structure of the ANYexo. It can be calculated as in Equation 4.15

$$R_{WH,\text{joint}} = R_{z,1}R_{1,2}R_{z,2}R_{2,3}R_{z,3}R_{3,4}R_{z,4}R_{4,5}R_{z,5} \tag{4.15}$$

where $R_{i,i+1}$ is constant and is due to the initial orientation of the joints. $R_{z,i}$ is due to the joint movements as expressed in Equation 4.16.

$$R_{z,i} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad i = 1, 2 \cdots, 6 \tag{4.16}$$

and θ_i is the angle of the i$^{\text{th}}$ joint. Recall that, protraction/retraction and elevation/depression is directly mapped to the position of GPR and GED joints. Thus, our aim is to find the relation between the positions of GHA, GHB and GHC, and the plane of elevation, elevation angle, internal-external rotation. After the orientation of the humerus is calculated for both ISB and joint coordinates in terms of knowns and unknowns, unknown parameters can be found. It is easier to equalize the two orientations if it is expressed in quaternion than in rotation matrix, because of the less number of elements in a quaternion.

$$q_{\text{ISB}} = \begin{pmatrix} w_{\text{ISB}}(\text{plane of elev., elev. angle, int-ext rot.}) \\ x_{\text{ISB}}(\text{plane of elev., elev. angle, int-ext rot.}) \\ y_{\text{ISB}}(\text{plane of elev., elev. angle, int-ext rot.}) \\ z_{\text{ISB}}(\text{plane of elev., elev. angle, int-ext rot.}) \end{pmatrix} = \begin{pmatrix} w_{\text{joint}}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5,) \\ x_{\text{joint}}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5,) \\ y_{\text{joint}}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5,) \\ z_{\text{joint}}(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5,) \end{pmatrix} = q_{\text{joint}} \tag{4.17}$$

By using the 4 constraints in Equation 4.17, unknown parameters, either ISB angles or joint angles, are found by using the Multidimensional Root Finding function in GNU Scientific Library [26].

Chapter 5

Simulation Environment

In applications where there is an interaction between the robot and its environment, safety is one of the main concerns. For an exoskeleton, the environment itself is the human, which makes the safety a very important factor during designing and testing of the controllers.

As explained in Chapter 2, Gazebo[1] physics engine is used to simulate the multi-body dynamics and controllers are implemented in a ROS[2] framework. This allows relatively faster testing and a safe environment to test. ANYexo was already in the simulation environment. Therefore, as a first step, the human model is generated and implemented into the simulation.

5.1 Human

The human is modeled as a combination of serial links connected with revolute joints. A kinematic structure that is very similar to the ANYexo is used [1]. This structure can be seen in Figure 2.2.

In Figure 2.2, it can also be seen that the axes of GPR and GED joints coincide at a point. This causes the radii of protraction/retraction and elevation/depression to be the same (Visual representation of these motions can be seen in Figure 5.1). However, it is known from [4] that the radii of these 2 motions are different as shown in Figure 5.2. Therefore, in the kinematic structure of the human, these 2 axes do not coincide. There is a relative offset on the transverse plane(see Figure 5.3 and 5.4) equals to the difference between the radii of elevation/depression and protraction/retraction. This is the only difference between the kinematic structure of ANYexo and the implemented human. Human in the simulation environment can be seen in Figure 5.5.

ANYexo is designed to be adjustable to the size of the 5^{th} percentile female to 95^{th} percentile male [1]. Therefore, human in the simulation environment should have adjustable link lengths to be able to implement different sized people. Only height, mass and the gender of the human are given as inputs to the simulation; then, link parameters shown in Figure 5.4 such as length, mass, and moment of inertia are estimated by using anthropometric tables [27], [28].

[1] http://gazebosim.org/
[2] Robot Operating System: www.ros.org

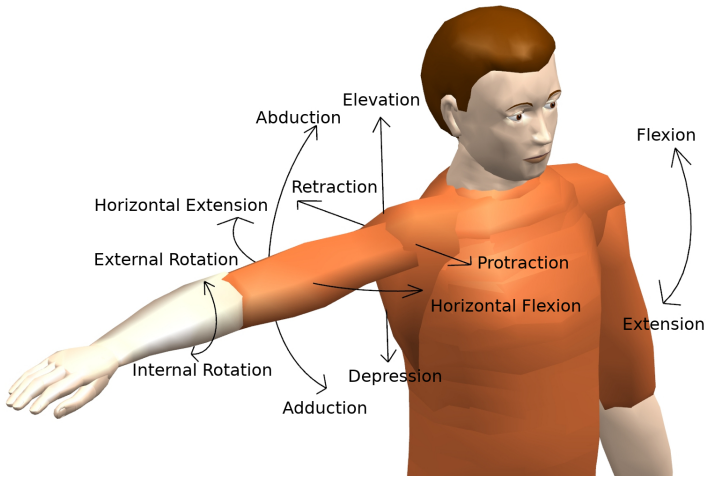


Figure 5.1: Human shoulder movements[13]

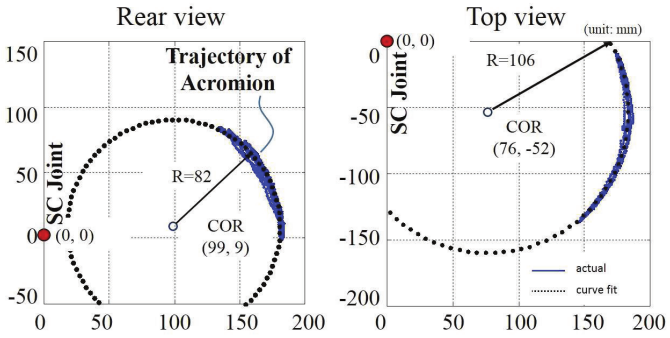


Figure 5.2: The trajectory of the acromion during the right-hand shoulder girdle motion of a healthy subject: (a) elevation– depression; (b) protraction–retraction.[4]

Anthropometric tables show the statistics of the dimensions and weights of human body parts. An example can be seen in Figure 5.6. In our estimation, it is assumed that a person which is at X $^{\text{th}}$ percentile body height, is also at X $^{\text{th}}$ percentile for other body part dimension. Similarly, a person which is at Y $^{\text{th}}$ percentile weight is also at Y $^{\text{th}}$ percentile for the masses of other body parts. In reality, this is not always the case; for example, some people might have longer arms compared to their height; however, to quickly implement different sized people in the simulation, this is a very useful approximation. For all of the parameters expressed in Figure 5.4, parametrization is done with respect to the overall height or mass. As an example, parametrization for the female upper arm length is presented in Figure 5.7. By using the equations of the fitted lines, dimensional parameters are calculated from the stature and mass parameters are found from the human weight.

5.2 Interaction Model

After human is implemented into the simulation, the next step would be to connect the exoskeleton to the human. For this, it is required to model the interaction between them. In Figure 2.3, two connection cuffs at the upper arm and forearm can be seen. Connection at those points is modeled by using spring and damper

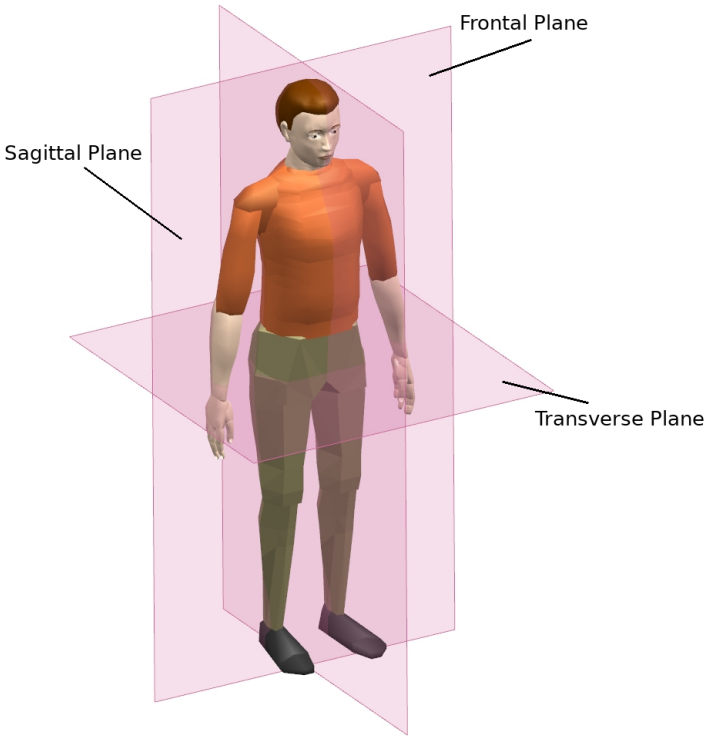


Figure 5.3: Human body planes [13]

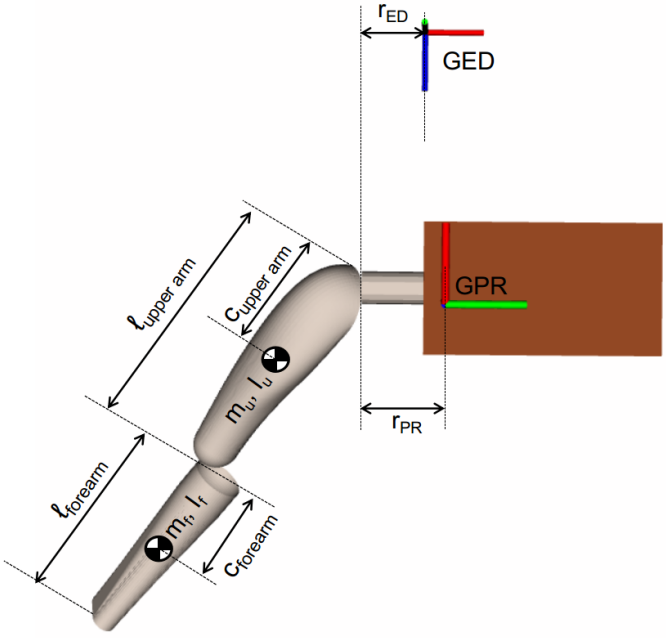


Figure 5.4: Human parameters to estimate. l: length of the link, c: proximal distance to the center of mass, m: mass, I: mass moment of inertia, r_{ED}: radius of elevation/depression, r_{PR}: radius of protraction/retraction

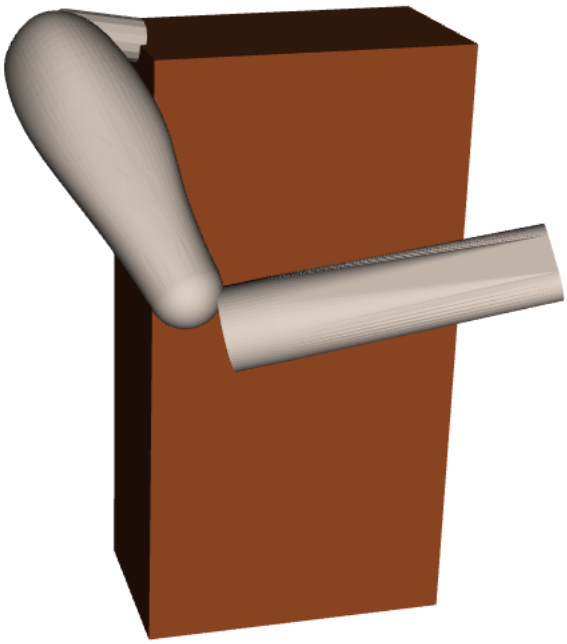


Figure 5.5: Human in the simulation environment

elements similar to the works of [18], [19], [20], [21], [22] and [23].

When a force is applied to human flesh, elastic deformation occurs on the tissue. Therefore, Young's and shear moduli should be used to find the spring parameters of the interaction. In [29], it is presented that the tissue at the human arm consists of bone, muscle, and fat. Bone can be safely approximated as a rigid body because its Young's modulus is in the magnitude of 10 GPa [30]. Therefore, the human arm can be modeled as a composite material which consists of fat and muscle as shown in Figure 5.8.

In [31], Young's moduli of the muscle and fat at the human forearm are investigated by using an indentation test setup shown in Figure 5.9. It is presented that the Young's moduli of fat and muscle are approximately 21 kPa and 87 kPa, respectively. The relation between the Young's modulus and the spring constant is expressed in Equation 5.1, where E is the Young's modulus, A is the contact area and l is the thickness of the material. The contact area, A, depends on the cuff width and the diameter of the upper arm or forearm. Cuff widths of ANYexo at the upper arm and forearm are 90mm and 45mm, respectively. The diameter of the human arm is also parametrized with respect to the human height as explained in 5.1. In [32], it is shown that the thickness of the fat is approximately 3mm, and does not change significantly for different subjects. Therefore, the thickness of the fat is taken as 3mm and the thickness of the muscle is taken as the distance between the fat and the bone. This thickness is calculated by using the bone thickness values in [33] and human fore/upper arm diameter values in [27] and [28].

Forearm-Hand Length

FEMALE N = 2208			MALE N = 1774		
Centimeters		Inches	Centimeters		Inches
44.29	Mean	17.44	48.40	Mean	19.06
2.34	Std Dev	.92	2.33	Std Dev	.92
54.60	Maximum	21.50	57.80	Maximum	22.76
32.40	Minimum	12.76	38.60	Minimum	15.20
	Percentiles			Percentiles	
39.14	1st	15.41	43.43	1st	17.10
39.74	2nd	15.65	43.98	2nd	17.31
40.12	3rd	15.79	44.32	3rd	17.45
40.62	5th	15.99	44.79	5th	17.63
41.38	10th	16.29	45.52	10th	17.92
41.91	15th	16.50	46.02	15th	18.12
42.32	20th	16.66	46.42	20th	18.28
42.69	25th	16.81	46.78	25th	18.42
43.02	30th	16.94	47.10	30th	18.54
43.33	35th	17.06	47.41	35th	18.66
43.63	40th	17.18	47.70	40th	18.78
43.92	45th	17.29	47.99	45th	18.89
44.21	50th	17.41	48.28	50th	19.01
44.51	55th	17.52	48.58	55th	19.12
44.81	60th	17.64	48.88	60th	19.24
45.13	65th	17.77	49.20	65th	19.37
45.47	70th	17.90	49.53	70th	19.50
45.84	75th	18.05	49.91	75th	19.65
46.26	80th	18.21	50.33	80th	19.82
46.74	85th	18.40	50.83	85th	20.01
47.35	90th	18.64	51.46	90th	20.26
48.25	95th	18.99	52.42	95th	20.64
48.81	97th	19.22	53.04	97th	20.88
49.21	98th	19.38	53.49	98th	21.06
49.81	99th	19.61	54.20	99th	21.34

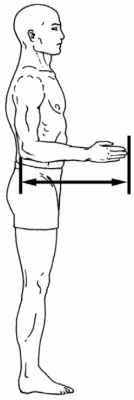


Figure 5.6: Anthropometric table of forearm-hand length

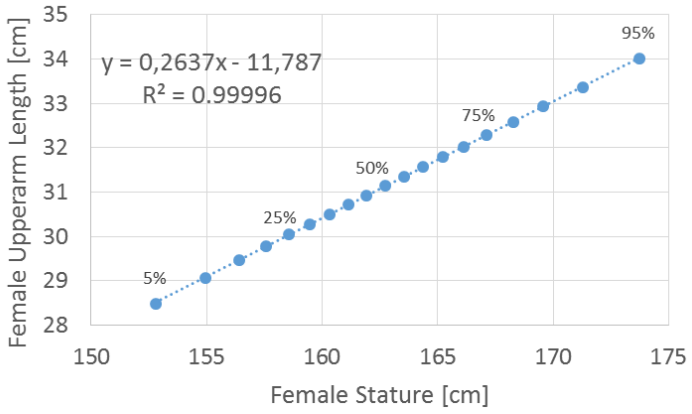


Figure 5.7: Female upper arm length vs stature. Each point represents the percentiles on the anthropometric table

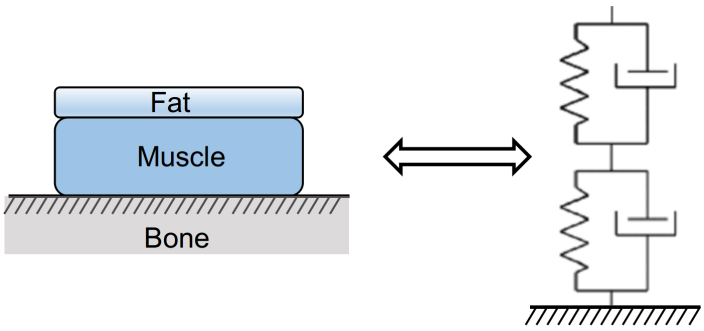


Figure 5.8: Simplified human arm interaction model

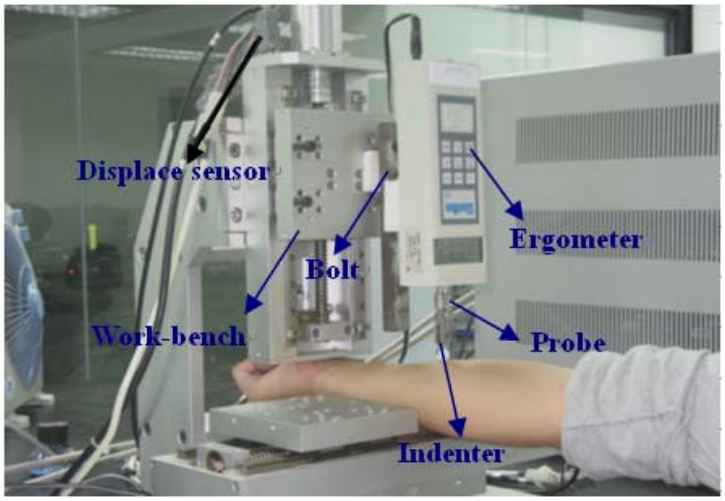


Figure 5.9: The measurement system of indentation [31]

$$k = \frac{EA}{l} \tag{5.1}$$

It should be noted that the Young's modulus describes the property of material on normal stress. Therefore, by using Equation 5.1, spring constants only for the directions that are perpendicular to the contact area can be found. (x and y axes in Figure 5.10. To calculate the spring constants along shear(z) directions, it is necessary to use the shear modulus. In [34], it is explained that the soft tissue mostly contains water and almost incompressible. Therefore, Poisson's ratio, ν can be approximated as 0.5. When Poisson's ratio is known, shear modulus, G and the equivalent spring constant can be easily found as in Equation 5.2 and 5.3, respectively.

$$G = \frac{E}{2(1+\nu)} \tag{5.2}$$

$$k = \frac{GA}{l} \tag{5.3}$$

After spring constants for fat and muscle are calculated, the equivalent spring constant along each direction can be found simply as in Equation 5.4. Calculated spring

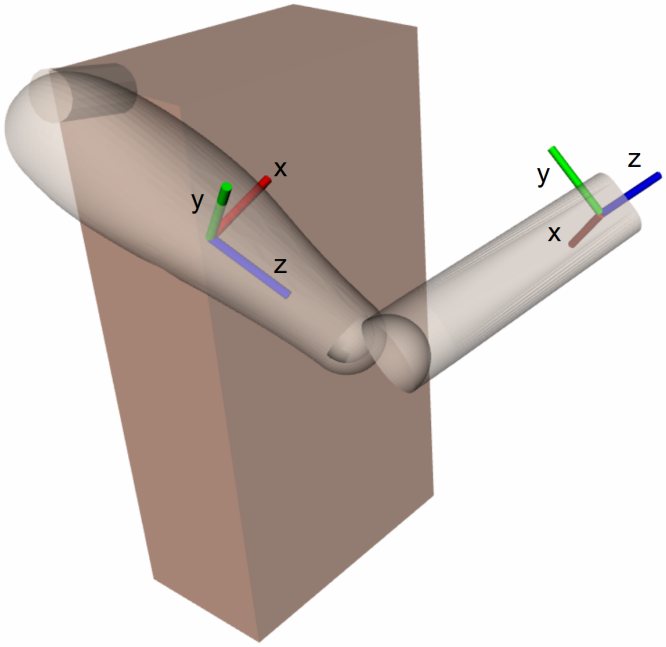


Figure 5.10: Contact frames

coefficients for a male with 50^{th} percentile dimensions are presented in Table 5.1.

$$k_{\text{eq}} = \frac{1}{\frac{1}{k_{\text{fat}}} + \frac{1}{k_{\text{muscle}}}} \tag{5.4}$$

Unfortunately, any work in the literature related to the damping constant at the human arm that can be directly used is not found. However, in [35] and [36] experiments were done on the human legs. From these 2 studies, damping coefficient at upper arm and forearm is taken as 250 Ns/m as an approximation.

5.3 ANYexo Link Adjustment

In Sections 5.1 and 5.2, the human model and its interaction with the exoskeleton is explained. It is also required that the link dimensions of the exoskeleton match with the human size. For this, ANYexo has passive prismatic joints which allow adjusting the lengths of the necessary links before the operation. Those joints are indicated by d_i in Figure 2.2. d_1 is to adjust the width of the shoulder; whereas, d_2 and d_3 are for the length of the upper arm and forearm, respectively. Even though the length of the human forearm and upper arm can be directly mapped to d_2 and d_3, adjusting shoulder width, d_1 is slightly more complicated. As explained in Section 5.1 kinematic structures of ANYexo and human are not the same. This makes it impossible to have a perfect fit between the glenohumeral joints of ANYexo and human in all configurations. The value of d_1 is calculated for the default configuration, where all joint positions are at 0°, which is shown in Figure 5.11. The Optimal value of d_1 perfectly brings the glenohumeral(GH) joints of the human and ANYexo to the same position. It is found by simply expressing the position of the

Table 5.1: Spring coefficients for a 50^{th} percentile male subject. Superscripts; +: coefficient along positive direction, −: coefficient along negative direction

Location	Direction	Spring constant(k)	Unit
Upper arm	x^+	7400	N/m
Upper arm	x^-	7400	N/m
Upper arm	y^+	7400	N/m
Upper arm	y^-	7400	N/m
Upper arm	z^+	1800	N/m
Upper arm	z^-	1800	N/m
Forearm	x^+	11900	N/m
Forearm	x^-	9000	N/m
Forearm	y^+	12800	N/m
Forearm	y^-	14500	N/m
Forearm	z^+	3500	N/m
Forearm	z^-	3500	N/m

GH joint of ANYexo in terms of d_1 from the forward kinematics and equating it to the GH position of human at zero configuration.

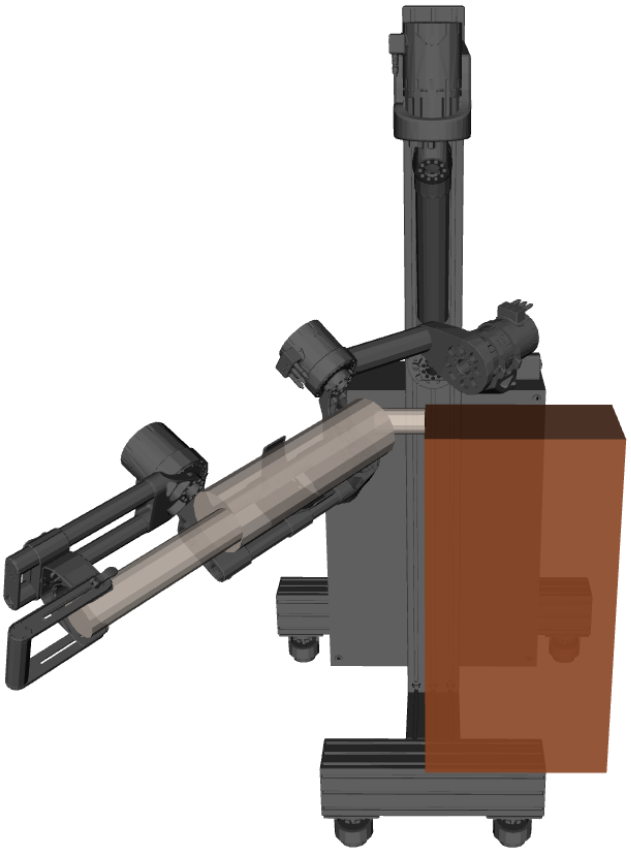


Figure 5.11: Zero configuration, where d_1 is adjusted

5.4 Sensor Implementation

In Section 3.1, it is explained that the controllers use force and torque measurements. Performance of these controllers might differ significantly depending on whether perfect or noisy measurements are used. To create a realistic testing environment, the performance of the sensor measurements in the actual setup is investigated and implemented into the simulation.

Force measurements are recorded when there is no external disturbance; thus, readings are around 0N. In Figure 5.12, it can be seen that the readings oscillate from the real value of 0N. To understand the behavior of these oscillations, the power spectrum of the signal is investigated and shown in Figure 5.13. It can be seen that the signal has almost equal contributions from each frequency, which is a property of the white noise. Therefore, along each direction, the variance of the white noise is found and implemented in the simulation environment. These values are presented in Table 5.2.

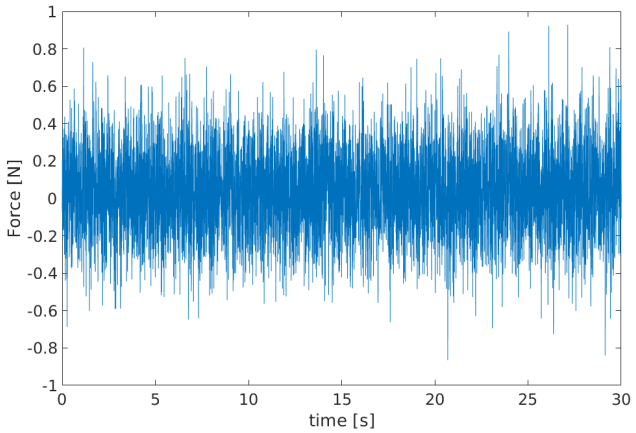


Figure 5.12: Force measurements along the x axis of upper arm when there is no external force is applied

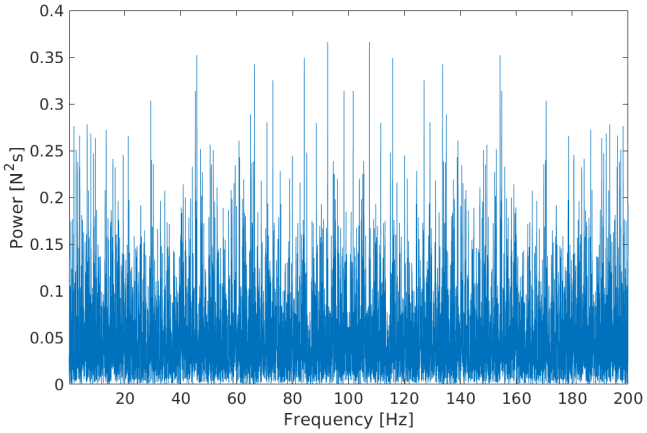


Figure 5.13: Power spectrum of the force measurements along the x axis of upper arm when there is no external force is applied

Table 5.2: White noise variance values that are used in the force sensor implementation

Location	Direction	Variance	Unit
Upper arm	x	0.052389	N^2
Upper arm	y	0.041720	N^2
Upper arm	z	0.020127	N^2
Forearm	x	0.053866	N^2
Forearm	y	0.055601	N^2
Forearm	z	0.030625	N^2

Chapter 6

Results

Several interaction force control methods are discussed in Chapter 3. One measure to test the force control is to test the transparency; in other words, setting the desired interaction forces and torques to zero. This also makes the comparison of different methods simpler. Therefore, results of these methods are presented for transparency both in simulation and in hardware.

6.1 Simulation Results

Due to the interaction with the user, safety has great importance. Moreover, force control operations lead to unexpected behavior more often compared to position control operations. Therefore, at first, the interaction force control methods are tested in the simulation environment. This also allows us to objectively compare the performances of different controllers.

6.1.1 Step Response

In control theory, one of the common ways to test the performance of a controller is to measure its step response. In our case, this is equivalent to have a non-zero interaction force when the controller is activated, and human stays still while the controller tries to reduce the interaction force. The test is done in the configuration shown in Figure 6.1.

Because the behavior of the responses along any direction at the upper arm or forearm is very similar, only the responses along the x direction at the upper arm are presented in Figure 6.2. Even though it is possible to roughly compare the performances by only visual inspection, it is better to use an objective metric. Therefore, settling times of the controllers for an error band of $\pm$ 0.2 N are found and presented in Table 6.1.

Table 6.1: Settling time of the proposed methods

Method	Settling Time [s]
Feed forward controller	33.1
Virtual mass controller($m_v = 0.5m$)	27.1
Single disturbance observer	2.6
Cascaded disturbance observer	1.0
PID	N/A

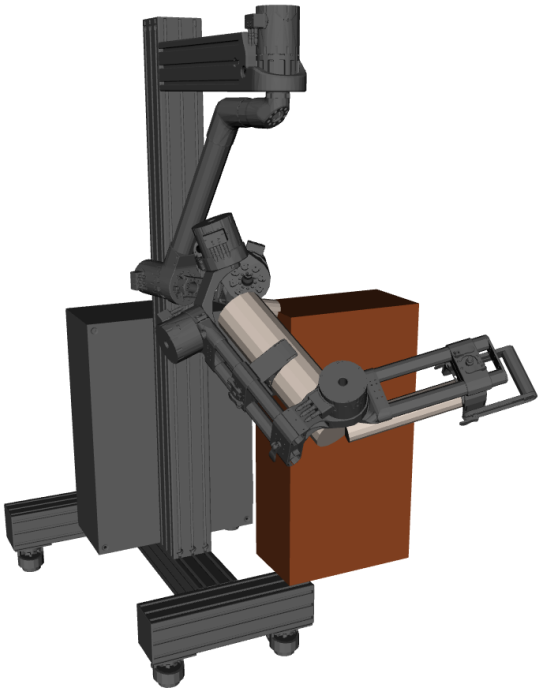


Figure 6.1: Configuration of the human and the exoskeleton during the step response tests.

6.1.2 Response to cyclic human motion

Step responses of the controllers give a rough idea of the controller performances; however, it is important to test these controllers on a more realistic case such as moving arm in a cyclic manner. Sine commands are given to the joints of the human with the frequencies shown in Table 6.2. The screenshots of this motion are demonstrated in Figure 6.3.

Table 6.2: Joint movement frequencies for the cyclic human motion tests.

Joint	Frequency[Hz]
GPR	0.2
GED	0.3
GHA	0.8
GHB	1.1
GHC	1.2
EFE	1.0

Because the human arm moves in several directions simultaneously, instead of comparing the forces along one direction, magnitudes of the forces are used to be able to objectively compare the performances. Controller type is switched from one to another while the human is in motion. In Figure 6.4, controller performances of the feedforward and virtual mass controller are presented. The blue curve represents the interaction force magnitude at the upper arm when feed forward control is in use; whereas, the red curve shows the response of the virtual mass controller with virtual mass being half of the real mass. Recall that the desired interaction force

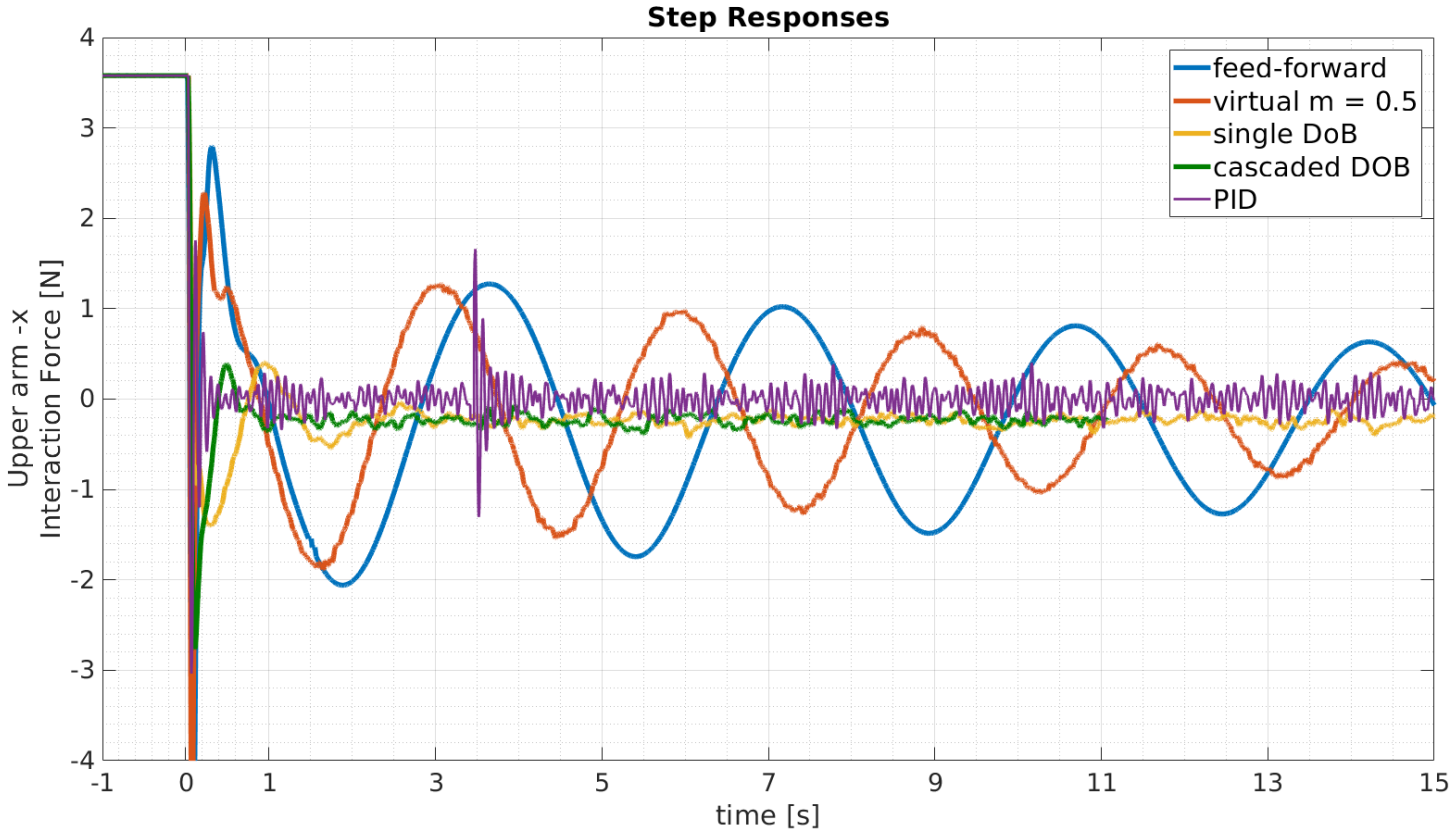


Figure 6.2: Step responses of the proposed methods at the upper arm along x direction. Controllers are activated at t = 0.

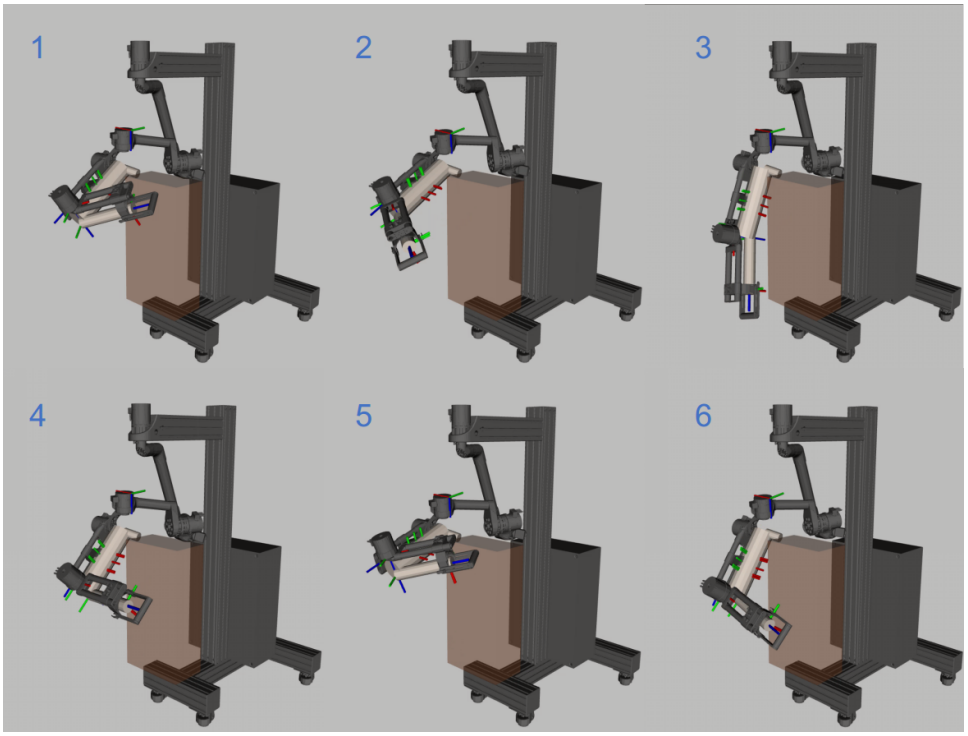


Figure 6.3: Screenshots of the cyclic motion with the commanded human joint frequencies given in Table 6.2.

is zero. Similarly, performances of the single and cascaded disturbance observer structured controllers are presented in Figure 6.5. Unfortunately, PID gains that lead to a stable response to the cyclic motion of the human could not be found; thus, force plots of the PID controller is not presented.

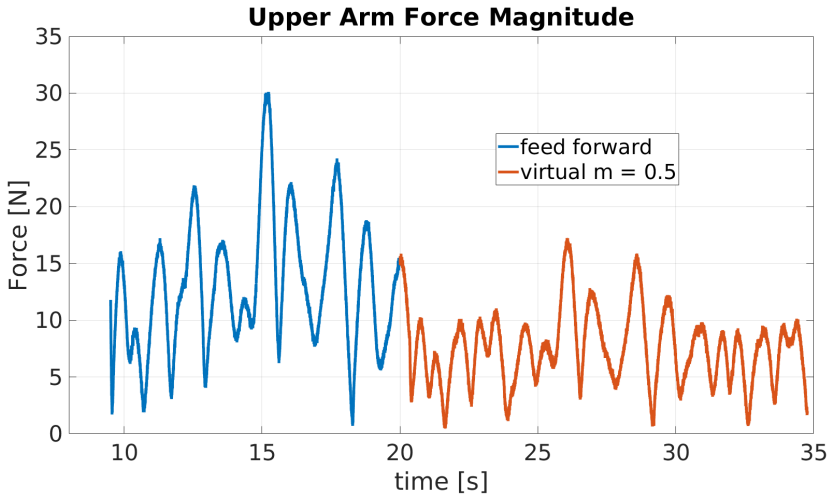


Figure 6.4: Interaction force magnitude at the upper arm. Controller is changed to virtual mass controller from the feed forward controller at t = 20s.

Because disturbance observer controllers try to control the task space acceleration,

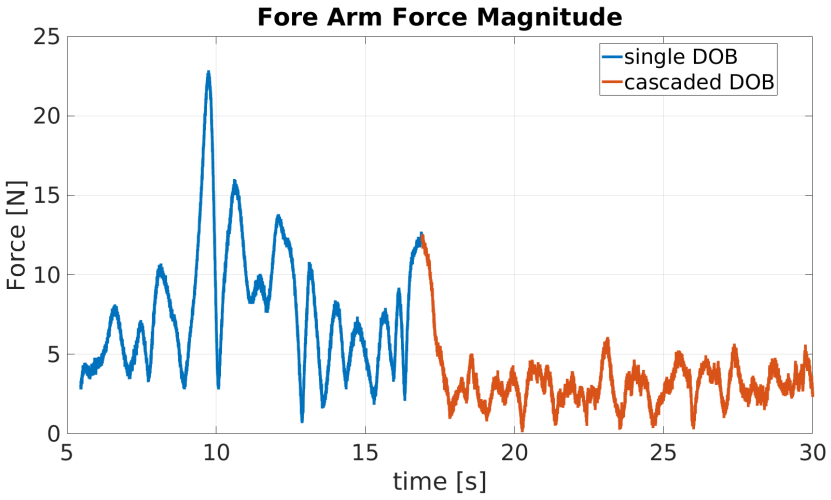


Figure 6.5: Interaction force magnitude at the forearm. Controller is changed to cascaded disturbance observer from the single disturbance observer at t = 17s. (DOB: disturbance observer)

it is also worth investigating their acceleration tracking performances. This is presented in Figure 6.6.

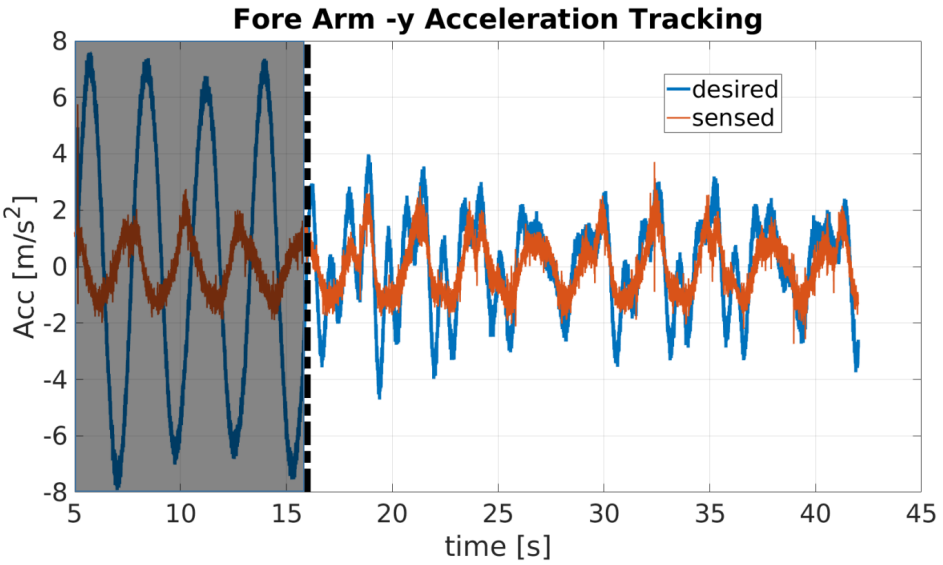


Figure 6.6: Acceleration tracking responses along the y axes of the fore arm. The shaded area on the left shows the response of the single disturbance observer controller, the non-shaded are on the right shows the response of the cascaded disturbance observer structure.

6.1.3 Robustness

In Chapter 3, it is explained that the disturbance observer controller uses the interaction model; more specifically, the spring and damping constant. Because we can not know the interaction parameters exactly in the real case, it is important

to test the disturbance observer controller under model mismatches. For the same human motion explained in Section 6.1.2, parameter knowledge of the controller is systematically changed, and the stability of the response is investigated. The results are presented in Table 6.3.

Table 6.3: Stability of the disturbance observer controller under model mismatch. k: spring constant, $\hat{k}$: controller knowledge of the spring constant, c: damping constant, $\hat{c}$: controller knowledge of the damping constant.

$\frac{\hat{k}}{k}$	$\frac{\hat{c}}{c}$	Result
0.75	1.00	Unstable
1.25	1.00	Stable
1.50	1.00	Stable
2.00	1.00	Stable
2.50	1.00	Unstable
1.00	1.25	Unstable
1.00	0.75	Stable
1.00	0.5	Stable
1.00	0.3	Stable
1.00	0.1	Unstable

6.2 Hardware Results

After testing the methods and obtaining stable responses in the simulation, controllers are tested on the hardware. Due to some problems on the force/torque sensor at the upper arm, only the interaction forces at the forearm is controlled. In the real system, it is not possible to disturb the exoskeleton in the exact same way and amount on different runs. The following experimental method is done to test the controllers as objectively as possible: The user is said to move his forearm along the direction shown in Figure 6.7 while trying to exert approximately same force to the exoskeleton. Then, the controller type is changed without informing the user. Force and acceleration measurements are recorded along the direction of motion and presented in Figure 6.8 and 6.9.

In Figure 6.8 and 6.9, it can be seen that the user applies force to the exoskeleton with approximately the same amplitude even if the controller is changed. However, when the virtual mass controller or disturbance observer controller is activated, much higher acceleration results are obtained. This means that the exoskeleton felt lighter to the user when those controllers are active. By using the acceleration and force measurements along a specific direction, felt inertia along that direction can be easily calculated with the formula given in Equation 6.1. Approximate felt inertias are calculated at the peak points, and their comparisons with the standard feed forward controller are presented in Table 6.4. Due to the incomplete sensor setup of the ANYexo, cascaded disturbance observe structure could not be tested on the hardware.

$$m_{\text{felt}} = \frac{F}{a} \tag{6.1}$$

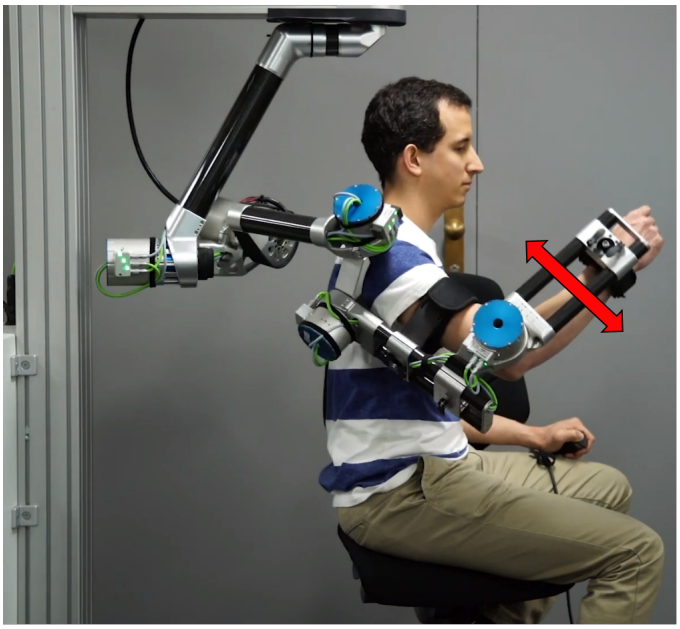


Figure 6.7: Configuration during the hardware test. Red arrow shows the direction that user moves his forearm.

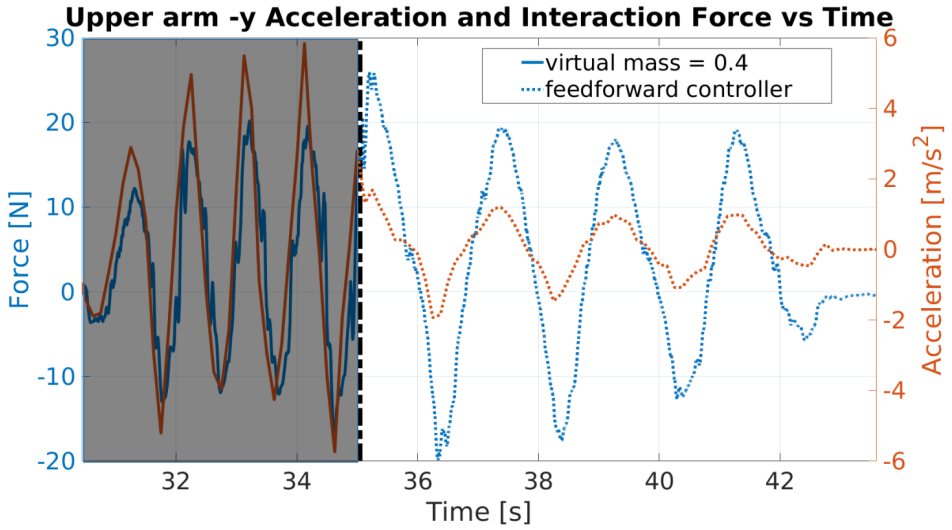


Figure 6.8: Acceleration and interaction force along the y direction at the upper arm. Shaded area with solid curves shows the motion when virtual mass($m_{\text{virt}} = 0.4m$) controller is running. Non shaded region with dotted lines represents the measurements when feed forward controller is active.

6.3 Discussion

Results shown in Sections 6.1 and 6.2 are discussed in this section to present reasoning and to draw conclusions about the performance of these controllers.

In Table 6.1, it can be seen that the feedforward controller has the slowest response with 33.1 seconds settling time to the step disturbance. This is expected

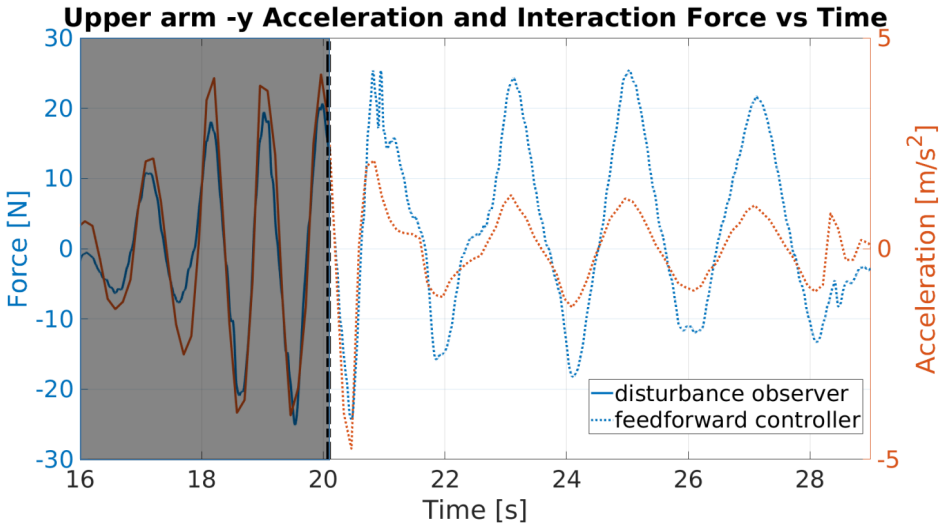


Figure 6.9: Acceleration and interaction force along the y direction at the upper arm. Shaded area with solid curves shows the motion when disturbance observer controller is running. Non shaded region with dotted lines represents the measurements when feed forward controller is active.

Table 6.4: Normalized felt inertias with the virtual mass and disturbance observer controllers. Normalization is done with respect to the felt inertia of the feed forward controller.

Controller	$\frac{m_{\text{felt}}}{m_{\text{feed forward}}}$
Virtual mass ($m_{\text{virt}} = 0.4m$)	0.22
Single disturbance observer	0.24

as it is the most basic approach with no use of force, torque and acceleration sensors(Section 3.1). Virtual mass controller($m_{\text{virt}} = 0.5m$) has the settling time of 27.1 seconds. At first glance, improvement brought by the virtual mass controller can be seen as insignificant. However, this is because of the fact that it tries to compensate for the inertial forces. Because human stays still, the exoskeleton has small acceleration; thus, small inertial forces. The single disturbance observer controller responds much faster to the step disturbance than the feedforward and virtual mass controller. Moreover, the cascaded disturbance observer structure responds even faster. This is because of the additional acceleration tracking control loop.

Even though the step response analysis gives an initial intuition, it is not enough to come conclusions on the controller performances. Step responses of the feedforward controller and virtual mass controller are quite similar. However, as it can be seen from Figure 6.4, the virtual mass controller performs much better than the feedforward controller when the human does the cyclic motion. Moreover, it can be seen in Figure 6.5 that the cascaded disturbance observer structure improves the performance significantly compared to the single disturbance observer structure. This is because of the additional acceleration tracking loop. This loop improves the acceleration control performance significantly(Figure 6.6), which leads to better interaction force tracking.

As it is explained in Section 6.2, sensory setup of the ANYexo was not complete at the times of the hardware experiments. Therefore, only force at the forearm is controlled. From Table 6.4, it can be seen that the felt mass with the virtual mass controller is 22% of the felt mass with the feedforward controller. However, as explained in 3.2, the expected value is 40%. The difference comes from the fact that, having force feedback automatically also compensates for the kinematic mismatches and the stick friction at the contact cuffs. It is seen that the single disturbance observer structure gives slightly worse result than the virtual mass controller; however, with more tuning and with the complete setup, it is believed that the disturbance observer structure would give better results.

Chapter 7

Conclusion

In this thesis, several interaction force control methods are investigated for a torque controllable series elastic actuated upper-limb exoskeleton, ANYexo. One reason to use upper-limb exoskeletons is to guide or assist the arm of the user. For this, appropriate forces should be exerted on the arm. Perfect force control decouples the human from the exoskeleton, and it allows to calculate assisting/correcting forces without considering the state of the exoskeleton. Therefore, the goal of this study is to be able to apply any force to the human arm through the contact points between the user and the exoskeleton.

Due to the interaction with the human, safety is one of the main concerns during the controller development and testing. Therefore, at first, a simulation environment is created. ANYexo itself was already in the simulator when the project has started. As a next step, a representative human model is developed. Similar but a more realistic kinematic structure is used on the human. By using anthropometric tables, human arm link properties such as, mass, length and moment of inertia are parametrized with respect to the height and mass of the human. To model the interaction between the human and the exoskeleton, spring damper elements are used. An approximate spring constant is calculated by using elastic and shear moduli of the fat and muscle.

After having a representative simulation environment, different force control algorithms are tested and their performances are compared. These are simple feed forward controller, virtual mass controller, PID controller, single disturbance observer, and cascaded disturbance observer controller. The tests are done on the transparency task; in other words, desired interaction forces and torques are set to zero. As expected, the feedforward controller with no use of force/torque/acceleration sensors gives the slowest and worst performance. The virtual mass controller shows similar performance to the feedforward controller when the human does not move. However, when the arm is moved in a fast manner, the virtual mass controller shows significantly better performance. Moreover, only having a single parameter to tune makes this approach a very strong candidate to use in the interaction force control tasks. On the other hand, due to the high amount of gains to tune and wind-up problems, a stable, satisfactory performance is not achieved with the PID controller. It is thought that using the interaction model in the controller design might improve the force control performance. Therefore, disturbance observer structure is implemented, where the interaction model is used to estimate the outside disturbance which is fed to the controller. The single disturbance observer structure gives slightly better performances than the virtual mass controller on the simulation. Moreover, the cascaded disturbance observer structure with the inner loop for the

acceleration tracking gives the best result by far in the simulation. On the hardware, tests are done by controlling the force only at the forearm, due to the incomplete sensory system of the ANYexo. On the hardware tests, virtual mass controller and the single disturbance observer structure performs similar transparency with felt inertia that is approximately 4 times less than the inertia felt when the feedforward controller is used.

Bibliography

[1] Y. Zimmermann, A. Forino, R. Riener, and M. Hutter, "Anyexo: A versatile and dynamic upper-limb rehabilitation robot," *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2019)*.

[2] L. H. Thomas, B. French, J. Coupe, N. McMahon, L. Connell, J. Harrison, C. J. Sutton, S. Tishkovskaya, and C. L. Watkins, "Repetitive task training for improving functional ability after stroke," *Stroke*, vol. 48, no. 4, pp. e102–e103, 2017.

[3] T. Nef and R. Riener, *Three-Dimensional Multi-Degree-of-Freedom Arm Therapy Robot (ARMin)*. London: Springer London, 2012, pp. 141–157.

[4] B. Kim and A. D. Deshpande, "An upper-body rehabilitation exoskeleton harmony with an anatomical shoulder mechanism: Design, modeling, control, and performance evaluation," *The International Journal of Robotics Research*, vol. 36, no. 4, pp. 414–435, 2017.

[5] V. Klamroth-Marganska, J. Blanco, K. Campen, A. Curt, V. Dietz, T. Ettlin, M. Felder, B. Fellinghauer, M. Guidali, A. Kollmar, A. Luft, T. Nef, C. Schuster-Amft, W. Stahel, and R. Riener, "Three-dimensional, task-specific robot therapy of the arm after stroke: a multicentre, parallel-group randomised trial," *The Lancet Neurology*, vol. 13, no. 2, pp. 159 – 166, 2014.

[6] J. M. Veerbeek, A. C. Langbroek-Amersfoort, E. E. H. van Wegen, C. G. M. Meskers, and G. Kwakkel, "Effects of robot-assisted therapy for the upper limb after stroke: A systematic review and meta-analysis," *Neurorehabilitation and Neural Repair*, vol. 31, no. 2, pp. 107–121, 2017, pMID: 27597165.

[7] S. J. Ball, I. E. Brown, and S. H. Scott, "Medarm: a rehabilitation robot with 5dof at the shoulder complex," in *2007 IEEE/ASME international conference on advanced intelligent mechatronics*, Sep. 2007, pp. 1–6.

[8] B. Kim and A. D. Deshpande, "Controls for the shoulder mechanism of an upper-body exoskeleton for promoting scapulohumeral rhythm," in *2015 IEEE International Conference on Rehabilitation Robotics (ICORR)*, Aug 2015, pp. 538–542.

[9] A. Gupta, M. K. O'Malley, V. Patoglu, and C. Burgar, "Design, control and performance of ricewrist: A force feedback wrist exoskeleton for rehabilitation and training," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 233–251, 2008.

[10] F. Farshidian, E. Jelavić, A. W. Winkler, and J. Buchli, "Robust whole-body motion control of legged robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 4589–4596.

[11] C. Högfors, G. Sigholm, and P. Herberts, "Biomechanical model of the human shoulder—i. elements," *Journal of Biomechanics*, vol. 20, no. 2, pp. 157 – 166, 1987.

[12] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, R. Diethelm, S. Bachmann, A. Melzer, and M. Hoepflinger, "Anymal - a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 38–44.

[13] A. Forino, "Anyexo - mechanical design of a robotic arm exoskeleton for stroke rehabilitation," *ETH Zürich(Master Thesis)*, July 2018.

[14] K. Bodie, C. D. Bellicoso, and M. Hutter, "Anypulator: Design and control of a safe robotic arm," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 1119–1125.

[15] C. D. Bellicoso, C. Gehring, J. Hwangbo, and M. Hutter, "Perception-less Terrain Adaptation through Whole Body Control and Hierarchical Optimization."

[16] C. Dario Bellicoso, F. Jenelten, P. Fankhauser, C. Gehring, J. Hwangbo, and M. Hutter, "Dynamic locomotion and whole-body control for quadrupedal robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 3359–3365.

[17] M. Morari and E. Zafiriou, *Robust Process Control*. Prentice Hall, Englewood Cliffs, 1989.

[18] R. Huang, H. Cheng, H. Guo, X. Lin, Q. Chen, and F. Sun, "Learning cooperative primitives with physical human-robot interaction for a human-powered lower exoskeleton," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct 2016, pp. 5355–5360.

[19] H.-T. Tran, H. Cheng, X. Lin, M.-K. Duong, and R. Huang, "The relationship between physical human-exoskeleton interaction and dynamic factors: using a learning approach for control applications," *Science China Information Sciences*, vol. 57, no. 12, pp. 1–13, Dec 2014.

[20] H. Kazerooni, R. Steger, and L. Huang, "Hybrid control of the berkeley lower extremity exoskeleton (bleex)," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 561–573, 2006.

[21] M. K. Duong, H. Cheng, H. T. Tran, and Q. Jing, "Minimizing human-exoskeleton interaction force using compensation for dynamic uncertainty error with adaptive rbf network," *Journal of Intelligent & Robotic Systems*, vol. 82, no. 3, pp. 413–433, Jun 2016.

[22] U. Nagarajan and A. Goswami, "Improved mobility with a neutral, motion-amplifying controller for an experimental exoskeleton," *SAE International Journal of Passenger Cars - Mechanical Systems*, vol. 8, 05 2015.

[23] M. K. Shepherd and E. J. Rouse, "Design and validation of a torque-controllable knee exoskeleton for sit-to-stand assistance," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 4, pp. 1695–1704, Aug 2017.

[24] D. Shier, *Hole's Human Anatomy & Physiology*, ser. Hole's Human Anatomy & Physiology. W.C. Brown Publishers, 1996.

[25] G. Wu, F. C. van der Helm, H. D. Veeger, M. Makhsous, P. V. Roy, C. Anglin, J. Nagels, A. R. Karduna, K. McQuade, X. Wang, F. W. Werner, and B. Buchholz, "Isb recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion—part ii: shoulder, elbow, wrist and hand," *Journal of Biomechanics*, vol. 38, no. 5, pp. 981 – 992, 2005.

[26] M. e. a. Galassi, "Gnu scientific library reference manual," 2018.

[27] NASA, "Anthropometric source book. volume 1: Anthropometry for designers," *NASA Technical Reports Center*, 1978.

[28] R. Huston, *Principles of Biomechanics*, ser. Mechanical Engineering. CRC Press, 2008.

[29] C. Pailler-Mattei, S. Bec, and H. Zahouani, "In vivo measurements of the elastic mechanical properties of human skin by indentation tests," *Medical Engineering and Physics*, vol. 30, no. 5, pp. 599 – 606, 2008.

[30] J. Y. Rho, R. B. Ashman, and C. H. Turner, "Young's modulus of trabecular and cortical bone material: Ultrasonic and microtensile measurements," *Journal of Biomechanics*, vol. 26, no. 2, pp. 111 – 119, 1993.

[31] J. Su, H. Zou, and T. Guo, "The study of mechanical properties on soft tissue of human forearm in vivo," in *2009 3rd International Conference on Bioinformatics and Biomedical Engineering*, June 2009, pp. 1–4.

[32] J. Iivarinen, "Diagnostics of human forearm soft tissues using indentation and suction measurements : experimental and modeling analysis," *Publications of the University of Eastern Finland. Dissertations in Forestry and Natural Sciences., no 141*, 2014.

[33] H. Haapasalo, H. Sievanen, P. Kannus, A. Heinonen, P. Oja, and I. Vuori, "Dimensions and estimated mechanical characteristics of the humerus after long-term tennis loading," *Journal of Bone and Mineral Research*, vol. 11, no. 6, pp. 864–872, 1996.

[34] V. R. Sherman, W. Yang, and M. A. Meyers, "The materials science of collagen," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 52, pp. 22 – 50, 2015, sI:Collagen mechanics.

[35] U. Nagarajan and A. Goswami, "Improved mobility with a neutral, motion-amplifying controller for an experimental exoskeleton," apr 2015.

[36] M. K. Shepherd and E. J. Rouse, "Design and validation of a torque-controllable knee exoskeleton for sit-to-stand assistance," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 4, pp. 1695–1704, Aug 2017.