Electromyographic Control of an Upper Body Exoskeleton

CA1035

AALBORG UNIVERSITY
DENMARK

Master Thesis
Faculty of Engineering and Science
School of Information and
Communication Technologies
Control and Automation
Fredrik Bajers Vej 7C
9220 Aalborg East
webinfo@es.aau.dk

Title:

Electromyographical Control of an Upper Body Exoskeleton

Project period:
Fall 2016 + spring 2017
2016-09-02 - 2017-06-08

Group number:
17gr1035

Group members:
Bjarke Roe-Poulsen
Lasse Bromose
Morten Rechter

Supervisor:
Christoffer Eg Sloth

Pages: 109

Appendices: 18

Concluded: 2017-08-06

Abstract:

This master thesis describes the initial design and implementation of a controller for an assisting upper body exoskeleton with two degrees of freedom; elbow- and shoulder flexion/extension. This project limits itself to only implementing control on the elbow-joint.

The thesis includes the derivation, parameterization and validation of a dynamic model describing the exoskeleton arm. Further a model of the human torque output in the elbow joint is derived. The joint torque is modeled using the Hill Muscle Model, in conjunction with electromyography and joint angle measurements.

After an experimental comparison of different controller structures, a controller is implemented using a hierarchical structure that separates exoskeleton controller design into three layers. One that directly controls the actuators, one that handles the interaction between user and exoskeleton, and one that tracks the intention of the user. The three layers are implemented as computed torque control, admittance control and joint torque estimation, respectively.

Quantitative tests have not been performed, but qualitative tests indicate that the implemented controller feels natural, and the user reported that the controller was assisting when lifting a dumbbell disc.

This thesis can be used as basis for further research intro electromyography-based upper body exoskeleton control, as the methods presented also are applicable in estimating and controlling other joints as well.

Resumé

Projektet involverer design af et kontrolsystem til et exoskelet til overkroppen. Projektet har begrænset sig til udvikling af et kontrolsystem til albueleddet.

Projektet baserer sig på et elektromekanisk system udviklet på AAU's Institut for Mekanik og Produktion, som en del af det EU-understøttede AXO-SUIT projekt.

Udvikling af exoskelet-kontrol er et forskningsområde i udvikling, og der er derfor ikke nogen endelige standardiserede løsninger på hvordan man bedst styrer et exoskelet. Projektet præsenterer to forskellige kontrolsystemer: et hvor et estimat af momentet i det menneskelige led bliver sat som reference direkte til motoren, og et der bygger på en hierarkisk struktur med tre lag. De tre lag består af en *task-level* controller, som estimerer hvad brugerens bevægelses-intention er, en *high-level* controller som håndterer interaktionen mellem bruger og exoskelet og en *low-level* controller der sørger for at aktuatorerne udfører det nødvendige arbejde.

Til brug i *task-level* controller inkluderer specialet en implementering og parametrisering af et estimeringssystem som ud fra brugerens elektromyografi og led-vinkel kan estimere momentet i albueleddet. Estimeringssystemet baserer sig på Hills muskel-model og SENIAM's metoder til filtrering af elektromyografiske signaler. Parametrene til modellen er fundet eksperimentelt og tilpasset til den enkelte bruger.

High-level controlleren er implementeret som en admittans controller, der tager det estimerede moment som input og genererer en hastighedsreference som *low-level* controlleren skal følge. Forskellige metoder er evalueret eksperimentelt, og den som brugeren sagde var mest komfortabel blev valgt.

Low-level controlleren er implementeret som en PID *computed torque* controller. Denne type af controller bruger *feedback-linearisering* for at fjerne al ulineær dynamik i systemet, og kræver derfor en god model af systemdynamikken. En dynamisk model af exoskelettet er derfor udledt, parametriseret og valideret, til brug i *computed torque* controlleren.

Til projektet er der udviklet både en hardware og software implementering af kontrol-systemet. Hardware implementeringen bygger på brushless-controllere fra MAXON, samt et passende interface til dem. Til det formål er der udviklet og produceret to PCB'er. Softwaren til controlleren er udviklet til at køre på en indlejret Linux platform, som også har mulighed for at køre i realtid.

Den samlede 3-niveau controller er testet på albueleddet, med kvalitative evalueringsparametre. Brugeren rapporterede at bevægelser føltes kontrollerbare og responsive, og der var god understøttelse under løft af en håndvægtsskive.

Projektet er derfor et godt bidrag til Aalborg Universitets forskning i exoskeletter og danner et solidt grundlag for videre forskning i elektromyografi som basis for kontrol af exoskeletter.

Preface

This master thesis is written by group 17GR1035 at Aalborg University and concludes the masters program in Automation and Control at the Institute of Electronic Systems.

A thanks is extended to the two Ph.D. students at Aalborg University Muhammad Raza-Ul Islam and Simon Christensen for building and maintaining the mechanical exoskeleton platform.

Citation is done using a scheme based on the IEEE citation standard. A citation before a full stop regards the sentence and a citation after full stop regards the entire paragraph.

This report is handed in electronically. A `.tar.gz` is attached containing the software that is developed for the exoskeleton, which is not released as open source.

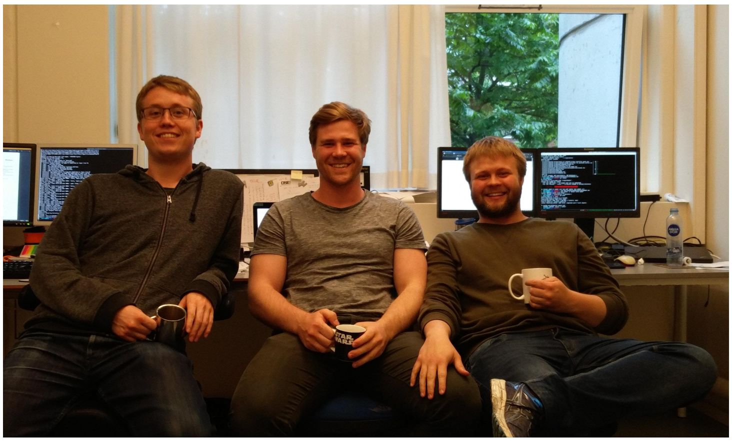


Figure 1: *The members of group 17GR1035 from left to right: Morten Rechter, Bjarke Nørskov Roe - Poulsen, Lasse Bromose.*

Abbreviations

AAU Aalborg University.

BLE Bluetooth Low Energy.

CE Contractile Element.

CoM Center of Mass.

DC Direct Current.

DOF Degrees of Freedom.

EC Electric Cummutator.

EMD Electromechanical Delay.

EMG Electromyography.

FSR Force Sensitive Resistor.

GA Genetic Algorithm.

IMU Inertial Measurement Unit.

MQX Message Queue eXecutive.

MSE Mean Squared Error.

NMSE Normalized Mean Squared Error.

PaGMO Parallel Global Multiobjective Optimizer.

PE Parallel Element.

PSO Particle Swarm Optimization.

PWM Pulse Width Modulation.

SE Serial Element.

List of Symbols and Notation

Notation

Δ	The prefix Δ indicates a change or difference	
$\dot{x}$	Time derivative is denoted with a dot	
$\boldsymbol{M}$	Matrices are denoted with bold capital letters	
$\text{sign}\,(a)$	Function to get the sign of a. Returns 1 or -1	
$\bar{A} \times \bar{B}$	Cross-product is indicated with a $\times$	
$\bar{v}$	Vectors are denoted with a line above the symbol	
$\bar{v}^{\mathrm{T}}$	The transpose of a matrix or vector is indicated with a raised T	
$a\,b$	Multiplication symbols are omitted	
x_d	References trajectories are indicated by subscript d for *desired*	

Symbol Names

α	Filter parameter	
β	Filter parameter	
δ	Electromechanical Delay	[s]
ϵ	Efficiency	[%]
η	EMG signal	
λ	Eigenvalue	
$\boldsymbol{B}$	Moment of inertia	$[\mathrm{kg\,m^2}]$
$\boldsymbol{I}_C$	Moment of inertia	$[\mathrm{kg\,m^2}]$
$\boldsymbol{I}_m$	Moment of inertia	$[\mathrm{kg\,m^2}]$
$\boldsymbol{M}$	Moment of inertia	$[\mathrm{kg\,m^2}]$
$\mathscr{L}$	The Lagrangian	[J]
ω	Angular velocity	[rad/s]
ω_n	Natural Frequency	[rad/s]
ϕ	Gaussian tuning parameter	
ψ	Filtered Electromyography signal	
ρ	Density	[kg/L]
σ	Slope parameter	
τ	Torque	[N m]
θ	Joint angle	[rad]
υ	Electromyography envelope	
$\bar{\psi}$	Model uncertainty	
$\bar{e}$	Angular position error	[rad]
$\bar{F}$	Friction torque	[N m]
$\bar{G}$	Gravitational torque	[N m]
$\bar{g}$	Gravitational force vector	$[\mathrm{m/s^2}]$
$\bar{n}$	Feedback linearization torque	[N m]
$\bar{u}$	Input signal	
$\bar{V}$	Centrifugal and Coriolis torque	[N m]
$\bar{y}$	Measurement	
ξ	Percentage of fast muscle fibers	[%]
ζ	Damping factor	
A	Degree of nonlinearity	
a	Activation Signal	
b	Viscous friction coefficient	
C	Filter parameter	
d	length	[m]
E	Normalized mean square error	
F	Force	[N]
i	Current	[A]

K	Kinetic energy	[J]
K_i	Integral gain	
K_p	Proportional gain	
k_t	Motor torque constant	[N m/A]
K_v	Differential gain	
k_v	Motor velocity constant	[V/(rad/s)]
L	Constant length	[m]
l	Length	[m]
L_a	Armature inductance	[H]
m	Mass	[kg]
N	Gear ratio	
r	Length	[m]
R_a	Armature resistance	[Ω]
S	Shape parameter	
t	Time	[s]
T_s	Sampling time	[s]
U	Potential energy	[J]
U_m	Motor voltage	[V]
U_{emf}	Back EMF voltage	[V]
v	Linear velocity	[m/s]
V_{CE_0}	Contractile velocity	[m/s]

Contents

1 Introduction

The concept of assistive robotic suits goes far back in the science fiction literature. Be it Marvel Comics' *Iron Man* or Lucasfilm's *Darth Vader*, the idea of using electromechanics to improve human motion has been around for decades. Innovations made in battery technologies, micro processors and electrical actuators in recent years have accelerated the development of real-world assistive robotic suits, or powered exoskeletons, significantly.

Early development within the field of powered exoskeletons was heavily focused on military applications. The first attempt of developing an exoskeleton started in 1965, and the suit was named Hardiman. It was co-developed by the United States military and General Electric. Development of Hardiman went on until 1969, and within those years, they created a prototype. The prototype proved somewhat successful, but had a number of issues (including stability, weight and power consumption) which led to the end of the project [1].

Since then some of the focus has shifted - while there is still research and development that focuses on military applications, rehabilitation and motion enhancement is gaining increasing amounts of attention. Recently, *SuitX* has gained a lot of attention, because of a story about a Steven Sanchez, who had a BMX accident in 2005, and has been paralyzed from the belly and down, who, with the help of *SuitX* is able to walk again [2].

Development of exoskeletons for rehabilitation is a growing business. Companies all over the world are developing exoskeleton systems to enhance human movement. Several, like *ReWalk*, *Ekso Bionics* and, *Parker Hannifin* from the US, as well as *REX Bionics* from New Zealand focus on walking, and assisting patients with spinal chord damage or other types of lower body paralysis. Other systems, like the French *Hercule* by *RB3D*, the Korean *DSME*, or Japanese *CYBERDYNE*, focus on supporting an able-bodied wearer during different work tasks, such as lifting goods, using heavy power tools, or support during stressful working positions.

While only one of the mentioned companies is based in the EU, this is likely to change in the future. The European Commission decided in 2014 to fund the largest civilian-funded robotics innovation programme, SPARC, with €700.000.000 [3]. SPARC is divided into several topic groups, including a healthcare group [4]. The healthcare group is divided into three topics; *Clinical Robotic*, which aims to support the clinical staff and assist in e.g diagnosis and surgical intervention. *Rehabilitation*, where patients have a direct interaction with a robot, that will enhance recovery. *Assistive Robotics* are robots which primary function is to assist either patients or their carer.

One of the current research projects within assistive robotics is the AXO-SUIT, which Aalborg University (AAU) is part of. It focuses on the development and construction of a full body assistive exoskeleton. The purpose of AXO-SUIT is to increase the endurance of wearers with weak muscles, such as elderly people. This will reduce the fatigue of the wearer and allow him or her to continue working on low-intensity tasks for extended periods of time. [5]

The AXO-SUIT research team at AAU is working on the upper body with focus on the arms, and have already built a prototype called the AXO-arm, shown in **Figure 1.1**, which this project is based on.

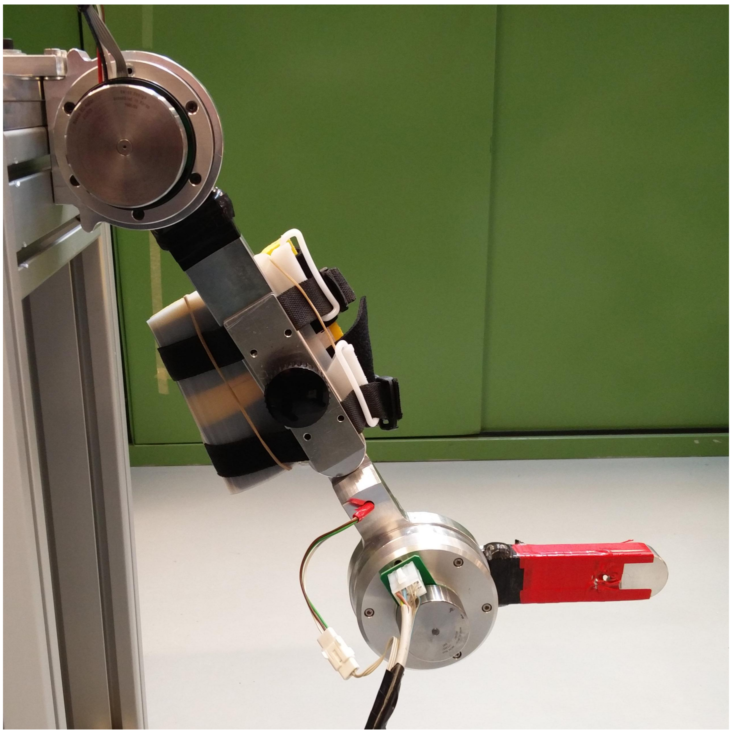


Figure 1.1: *The AXO-arm, developed at AAU Department of Mechanical and Manufacturing Engineering.*

In recent years, several upper body exoskeleton designs have been proposed. In [6] the authors describe a 7-Degrees of Freedom (DOF) exoskeleton arm where the joints are actuated by motors pulling wires connected to the mechanical joint. The motors are at the base of the arm, and reel in the wire to follow muscle contractions and let it go when the muscle relaxes. Arms of this type can be relatively light, but of course the base of the arm gets quite heavy and complex, because the motors still have to be somewhere in the system.

In [7] a similar wire system is used. However, instead of reeling the wire in and out using motors, this paper suggests pneumatic muscle actuators. They consist of a rubber tube connected to pressurized air. Increasing the pressure in the tubes makes them contract and releasing the pressure makes them relax. Actuators of this type have many advantages, including a low weight, cost effective production, and mechanical properties that are similar to real muscles.

On the prototype of the AXO-arm, the joint actuators are placed directly in the joints of the arm and it only has 2 DOF, with a motor controlling each joint. This design reduces the complexity of the mechanical system at the base of the arm, and also of the arm itself, since there is no need for running steel cable from one joint to the other. However, this design choice has the disadvantage of increasing the weight of the arm itself, since the wearer is carrying the weight of both the mechanical construction and the motors.

A vital part of controlling an assisting exoskeleton is measuring the intent of the wearer. This intention measurement should act as a control reference for the exoskeleton. If the exoskeleton were to simply mimic the motion of the wearer, the use of one or several Inertial Measurement Unit (IMU)'s, attached to the wearer, might be sufficient. However, the exoskeleton needs to assist the wearer, for example in case a lift is to be performed that the wearer is not strong enough to do. In that case, a measurement of muscle activity or neural muscle activation is desired.

A popular sensor choice for exoskeleton control is the surface Electromyography (EMG) sensor. It is used both in [6], [8], and in the commercial lower limb exoskeleton from CYBERDYNE. The EMG sensor measures the electrical potential in a muscle, with respect to some reference. Only surface EMG is considered here, as an intramuscular EMG is deemed too invasive for the purposes of the

AXO-arm.

A muscle that is relaxed does not show any electrical activity [9], when the muscle is contracting it is possible to measure the electric impulse that is sent from the brain. The measurement of this phenomenon is called electromyography.

The use of electromyographic sensors in [6] is of particular interest as the authors use the EMG signal and the user's arm joint angles to estimate the intended joint torque. They do so using a system that they call a *Myoprocessor* which is capable of online torque calculations based on the level of neural activation and a dynamic muscle model.

Available at the Section of Automation and Control at AAU is the Myo Band, shown on **Figure 1.2**, that is developed by Thalmic Labs. It is a commercial EMG-based sensor band for estimating hand gestures. It provides high quality EMG measurements with an 8 bit resolution[10], and a sample rate of 200 Hz[11]. It consists of 8 pods that are placed evenly around the wearers forearm, each containing an EMG sensor. The pods are connected to one another with an expandable band, assuring a comfortable and secure grip around the arm.

Figure 1.2: *The Thalmic Labs Myo Band [12]*

The finish and ease of use of the Myo Band makes it a compelling choice for this project. It also has a built in gyroscope and accelerometer which might prove useful in position estimation, this can be sampled at 50 Hz. However, it is made for use for the forearm, which might make it difficult to detect shoulder movement, which is is desirable in this project. It is, however, possible to mount the Myo Band on the upper arm instead, possibly assisting in shoulder movement detection.

The AXO-SUIT development team at AAU is in the process of developing a new type of sensor. Like EMG it seeks to measure the level of muscle activation. It is based on force sensitive resistors (FSR), which is a variable resistor where the resistance decreases when force is applied to it. A series of FSRs are mounted on the inside of a non-expanding nylon armband, and then strapped around the arm. When the muscles are activated they tense up and push outwards onto the FSRs reducing the resistance in the FSRs.

FSRs are relatively cheap components and unlike the Myo Band, there are no limitations on the circumference of the arm that it mounts around. However, there are no commercial solutions readily available, so a potential solution requires design, implementation, and testing before use.

As the research in the FSR band is still ongoing, with no published results regarding its intention estimation capabilities, the Myo Band is the superior choice. Its ease of use, and the repeated use of EMG technology in the litterature also supports this choice.

Project Summary

The focus of this master thesis will be on controlling the 2 DOF AXO-arm, using EMG measurements obtained using the Myo Band. It will work from the problem statement:

> *How can an EMG based control system be implemented on a 2 DOF upper body exoskeleton, to allow a natural and supporting motion of the wearer?*

The project contains a detailed description and model of the AXO-arm, and implementation of a human joint torque estimation technique. These two form the basis for design of a controller that is implemented and tested.

System Description 2

The current setup of the AXO-SUIT consists of two joints; an elbow joint and a shoulder joint. For this project a Myo Band has been added, which includes EMG sensors. A microprocessor for control has also been added. The setup is shown in **Figure 2.1**, and the components are described in the following sections.

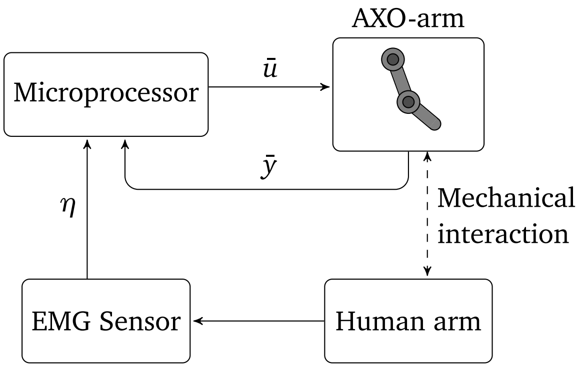


Figure 2.1: *Overview of the AXO-arm system.*

The vector $\bar{u}$ contains the input signals to the two joints. Vector $\bar{y}$ contains the sensor measurements from the joints. The EMG sensor block measures muscle activation levels, and transmits them to the microprocessor in the vector η.

Of course there is an interaction between the AXO-arm and the human arm in **Figure 2.1**, in that the two are strapped together.

2.1 AXO-arm

The AXO-arm block in **Figure 2.1** consists of several subsystems. There is the electromechanical system with the motors, gears, and the physical arm. Then there is the electronics, with motor drivers and sensors. Each joint has a motor driver, that measures the motor velocity using a built in hall sensor, and measures the applied current. There is also an absolute encoder in both joints to measure the angle. A diagram of the AXO-arm setup is shown in **Figure 2.2**, where θ, $\dot{\theta}$, and i_a correspond to the vector $\bar{y}$ in **Figure 2.1**. The angle and angular velocity, θ and $\dot{\theta}$, are measured, as well as the armature current, i_a.

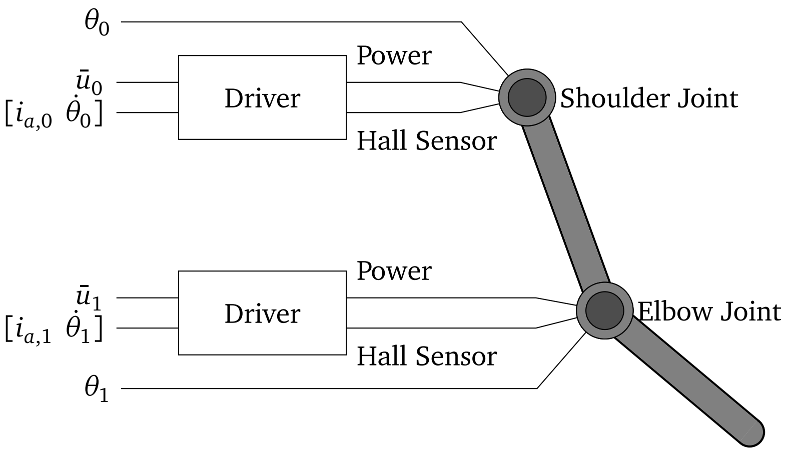


Figure 2.2: *The electrical wiring of the AXO-arm.*

2.1.1 Elbow Joint

The elbow joint consists of a MAXON EC45 motor with a 50:1 Harmonic Drive gear[1]. The MAXON motor is controlled by an ESCON 50/5 Servo Controller. There is a NOVOHALL RFD4000 position encoder located at the joint, providing measurements of the joint angle. The position encoder has a measuring range of 0° to 360°, with a resolution of 12 bit. The measurement is output as an analog voltage between 0.25 V to 4.75 V.

As a safety precaution to protect the wearer of the AXO-SUIT, the movement of the elbow joint is limited to angles between 0° (fully stretched) and 147° (fully flexed), as shown in **Figure** 2.3a. The red line indicates the maximum flexion angle and the black line indicates the joint at 0° which is also protected by a mechanical stop.

2.1.2 Shoulder Joint

The shoulder joint is very much like the elbow joint but with a bigger motor. A MAXON EC60 motor with a 50:1 Harmonic Drive gear, and it is also controlled via a ESCON 50/5 controller. The angle of the shoulder joint is also measured using the NOVOHALL RFD4000. The shoulder joint also has limited movement to protect the wearer. It can move between −10° and 190° as shown on **Figure** 2.3b.

(a) *Limits of the elbow joint.* **(b)** *Limits of the shoulder joint.*

Figure 2.3: *The mechanically allowed movement ranges of the two joints of the AXO-arm. The red lines indicate limits and the black lines indicate 0°. On the elbow joint, 0° and maximum extension coincide.*

[1]The model CSD-25-50-2A-GR is used in both the shoulder- and elbow joint.

2.1.3 Motor Driver

The motors are controlled using ESCON controllers. ESCON 50-5 drivers are chosen for controlling the motors. This motor driver is a rather advanced system that takes care of several motor control tasks. The key features are [13]:

- It generates the current signals to each of the three phases of the Electric Cummutator (EC) motor.
- It measures the motor velocity using built in hall sensors, and is able to provide its measured value as an analog voltage.
- It measures the current drawn by the motor. It sums the current of the three phases to get a single value in stead of three. It provides an analog voltage that represents the current.
- There is an on-board control system that allows control with either a velocity- or current reference. It is also possible to operate the motor in open-loop but only with a velocity reference. The references to these controllers can be set either using an analog voltage or a Pulse Width Modulation (PWM) signal.
- The reference signal can only be positive, so the driver also needs a direction signal. It is set with a digital signal that is either logical high or low. Logical high is between 2.4 and 36 V
- Finally, it is possible to enable and disable the motor with a digital pin.

2.1.4 Hardware Implementation

While the AXO-arm setup currently consists of two Maxon EC motors, a near-future upgrade will add one more motor to increase the DOF. The ESCON 50-5 drivers should for ease of use be connected to a motherboard PCB. For this purpose, a motherboard with room for three ESCON 50-5 drivers has been developed and produced. It is shown on **Figure 2.4**, and a diagram can be found in **Appendix D**.

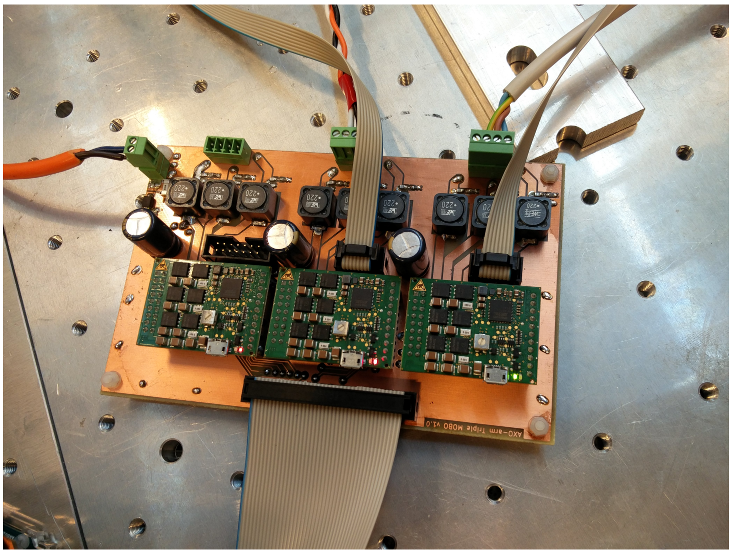


Figure 2.4: *The triple ESCON motherboard. The small ribbon cables are for the HALL sensor and encoder on each motor. The wide ribbon cable is for interfacing with a micro-controller.*

The motherboard is designed according to the guidelines in [13]. It suggests a 220 µF electrolytic capacitor on the supply voltage for each driver, along with a TVS[2] diode and a fuse. It also recommends

[2]Transient voltage suppressor.

adding a motor choke to reduce the emission of electromagnetic interference [13]. It has been chosen to apply a motor choke of 22 µH, along with an appropriate RC-filter on each phase. The component values are found in [13] and shown on **Figure 2.5**.

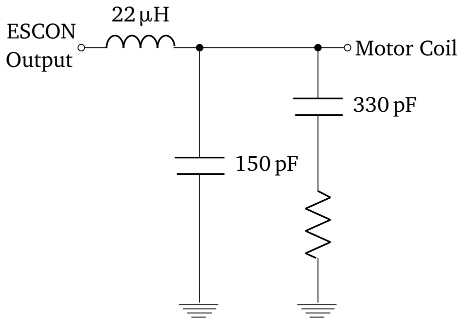


Figure 2.5: *The motor choke. A circuit like this is attached to every phase of the motor.*

The 40-pin ribbon cable contains all the control inputs and analog outputs from the three ESCON drivers, as well as the analog output voltages of the encoders. The encoder signals are divided on the motherboard to move them from 0 V to 5 V to 0 V to 3.3 V. This setup makes it possible to attach any micro controller to the other end of the ribbon cable, and interface with the AXO-arm.

The chosen micro controller for implementing the controller is a UDOO Neo. It is a single board computer that seeks to combine what is great about Arduino and what is great about the Raspberry Pi. It has a Freescale i.MX6 SoloX processor, that combines two Arm cores in a single chip; a Cortex A9 and an M4. The UDOO has an on-board wireless card with WiFi and Bluetooth. For connecting the micro controller with the ribbon cable, a micro controller shield has been made, shown in **Figure 2.6**. Light emitting diodes show if the motors are enabled and which direction they are moving.

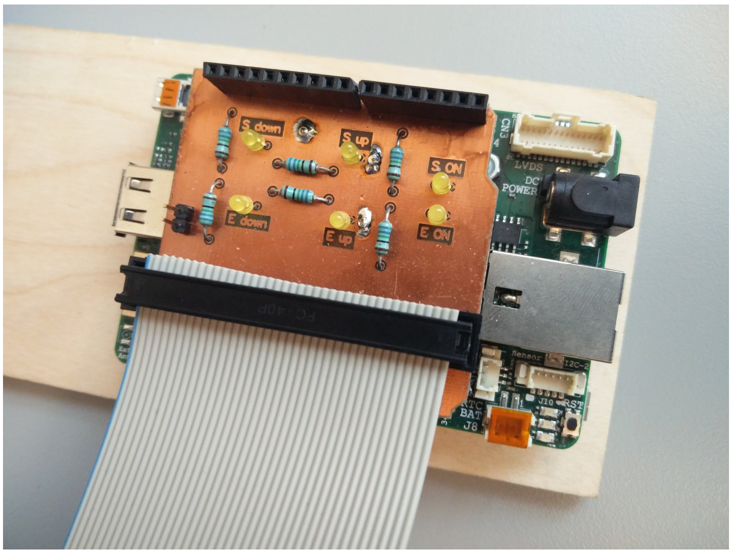


Figure 2.6: *This shield functions as the link between the UDOO Neo and the motherboard ribbon cable. It is here attached to the micro controller.*

A diagram of the shield can also be found in **Appendix D**.

2.2 Muscle Sensors

To estimate muscle movement and activity a Myo Band has been purchased. The Myo Band contains 8 different medical grade stainless steel EMG sensors and a 9-axis IMU. The Myo Band provides a Bluetooth interface.

The developers behind the Myo Band, Thalmic Labs, provide a library to interact with the Myo Band, unfortunately the library is only available for Microsoft Windows and Apple OS X. They did however, also release a description of the protocol used to communicate with the Myo Band, described in two blog posts on their developer blog, [14] and [11]. They also created a GitHub repository with related datatype definitions [15]. With this information, it is possible to communicate with the Myo band without using their official library, and interact with the Myo band from e.g. a Linux system.

For this project an OS independent interface module has been implemented. It is build around the Bluetooth Low Energy (BLE) interface specified by the aforementioned blog posts and data types. A detailed description of the implementation can be found in **Appendix C**.

Requirements 3

Specifying requirements for exoskeletons is not a well-researched area. Since the mechanical part of the system directly interfaces with a human being, the majority of the requirements are functional, and thus difficult to distill into mathematically verifiable answers. In this chapter we describe the functionalities of the AXO-arm, and formulate a set of system- and performance requirements that support the functionalities.

3.1 Functional Requirements

As mentioned in **Chapter 1**, the focus of the AXO-SUIT project is to strengthen users who have weak muscles. A common example of a weak-muscled individual, that is also used in the graphics on the AXO-SUIT website [5], is an elderly person trying to get by on his or her own. The focus of the AXO-arm controller is to assist the user in his or her movements, by transferring some of the load on the muscles onto the AXO-arm construction.

It is of course important that the operation of the AXO-arm is not dangerous to the user. The possible range of motion of the AXO-arm should therefore not extend beyond the range of motion of a human user, i.e. it should not be possible to bend the elbow backwards. Ideally the AXO-arm should be able to operate in the entire human range of motion, but the system that is available at AAU at the time of writing only has 2 DOF, resulting in a system that only allows movement in a vertical plane.

The following list contains a breakdown of the functional requirements.

1. The AXO-arm must accurately follow the motion of the user during free motion, and not impose extra resistance when moving around. Operation has to feel natural.

2. The AXO-arm must support the user when a load is applied, for instance if something is picked up.

3. The AXO-arm should be safe to use and not injure the user by forcing the joints into positions that are not allowed by normal human motion.

3.2 Technical Requirements

3.2.1 Motion Control

Motion tracking is an essential part of the exoskeleton design. The motion controller should be able to track trajectories, generated by the user, in both angle and angular velocity domain, and should do so in a sufficiently fast manner. A power spectral analysis of movement in the elbow joint was performed in [16] and found a maximum voluntary movement frequency of 4 Hz to 6 Hz. Since the AXO-arm will be mounted on a human arm, having zero steady-state error on the position is not critical, as the feedback loop that consists of the user will compensate for this. However, in the torque domain, zero steady state error is critical since an exoskeleton that attempts movement when the user is not attempting to move is undesirable. Motor torque control is an inner loop relative to the position controller. A rule of thumb when designing cascade controllers, is that the inner loops should be at least 5 times faster than the outer loops [17]. The requirements for the motion tracking controller are:

- A closed loop bandwidth of 6 Hz or more on the position controller.
- A closed loop bandwidth of 30 Hz or more on the torque controller.
- No torque steady-state error.
- A sample rate of at least 10 times the desired closed loop bandwidth, on both torque and position controller. [18]
- No overshoot on the position control.
- Gain margin of at least 10 dB, and a phase margin of at least 45° on the position control.

3.2.2 Model

A standard way of controlling serial manipulators is using feedback linearization which is a model based control technique where all nonlinear terms are canceled by the input signal. For this purpose, a dynamic model of the system needs to be developed, and it needs to be sufficiently precise to allow for a well performing feedback linearized controller. The model must have a precision that results in a fitness, measured using Normalized Mean Squared Error (NMSE), of at least 0.8, when it is applied to a measurement series. This precision should allow the model to track the mechanical system with good precision when used inside a feedback loop. NMSE is a normalized measure of fitness that can take on values in the range $E \in]-\infty, 1]$, where 1 is a perfect fit. If the NMSE is zero, then the fit is no better than that of an affine approximation. The calculation of NMSE is shown in **Equation (3.1)**, where x is the measurement data, $\hat{x}$ is the estimate from the model.

$$E = 1 - \left|\left|\frac{x-\hat{x}}{x-\text{mean}(x)}\right|\right|^2 \tag{3.1}$$

3.2.3 Intention Tracking

The primary input signal to the AXO-arm is measurements from EMG sensors. This measurement will be used to estimate the force exerted by the user and in extension thereof, the desired motion. Some amount of learning is to be expected from the user, who will adapt his or her behavior while using the AXO-arm. However, when it comes to estimating the load on the user's muscles, the estimator does need to be close to reality. It is thus chosen to demand a precision of the estimated joint torques that results in a NMSE of at least 0.75.

3.2.4 Load Transfer

The purpose of the AXO-arm is to increase the stamina of the user when working under load. The AXO-arm must therefore be capable of transferring the load from the muscles and onto the electromechanical system. It is chosen to transfer 50 % of the load onto the electromechanical system.

3.2.5 Safety

The AXO-arm needs to be safe to operate. The joints must therefore not be able to move into positions that impossible to reach for humans. Since the AXO-arm for now only has two degrees of freedom: elbow flexion/extension and shoulder flexion/extension, these are the two joints to consider. According to [19] the elbow joint can move between 0° (fully stretched) and 150° (fully flexed), and the shoulder can move between −60° and 180° if we place 0° where the arm is hanging down the side of the body. The movement of the AXO-arm should thus not exceed these angles.

There is no ISO standard on exoskeletons yet, so the technical standard on collaborative robots, TS15066 [20], is used in this project. It states that the quasi static physical contact force between an operator and a machine can at most be 160 N on the forearm, and 150 N on the upper arm, before it results in a minor injury. A minor injury is specified a severity of 1 on the Abbreviated Injury Scale. For a transient contact force the maximal force is doubled. The interaction force between the user and the AXO-arm must thus not exceed these values. This force can be translated into maximum joint torques using the length of the links, the gear ratio, and the gear efficiency. The two links are 227 mm and 330 mm long, so the maximum allowed output torque on the elbow and shoulder joint, respectively, are

$$\tau_{\text{max,elbow}} = 0.227\,\text{m}\;160\,\text{N} = 36.32 \qquad [\text{N m}] \tag{3.2}$$

$$\tau_{\text{max,shoulder}} = 0.330\,\text{m}\;150\,\text{N} = 49.5 \qquad [\text{N m}] \tag{3.3}$$

TS15066 further states that the operator of a collaborative robot should at all times have the means to at any time and by a single action stop the robot. The *means to stop the robot* could be an enabling device or an emergency stop button.

Part I

Mechanical Modeling

An important part of controller design involves knowledge of the system that is to be controlled. This part contains a kinematic and dynamic model of the AXO-arm, including both electrical and mechanical parts. It also describes how the models are parameterized and finally there is a verification to asses the model validity.

Forward Kinematics 4

Kinematics is a branch of mathematics that describes the motion of objects relative to each other. It is therefore a useful tool when modeling serial manipulators.

Forward kinematics is a method for relating the Cartesian coordinate and orientation of the end effector to the base of the serial manipulator. This means that the tool frame can be described with relation to the base frame. In most serial manipulator cases, the known variables are the joint configurations. Forward kinematics is a method for relating the joint variables to the position and orientation of the end effector. A frame in this context is a coordinate system. The base frame is where the AXO-arm has a fixed base and the tool frame is the most outer point of the robot arm, where a tool is usually placed. To understand forward kinematics, first the idea behind transformation matrices is explained since these are the cornerstones of forward kinematics and relates the coordinate systems of two neighboring joints. This is followed by an explanation of a standard way to construct these transformations matrices called The Denavit-Hartenberg Convention. With this tool, the transformation matrices for the current setup are found and by combination of these, the forward kinematics is constructed. [21]

4.1 Terminology

To ease the description of the forward kinematics, here is a small section on terminology and notation.

Frame

A coordinate system that is attached to a point on the serial manipulator. Coordinate frame A is denoted $\{A\}$.

Transformation matrix

A matrix, ${}^{b}_{a}\boldsymbol{T} \in \mathbb{R}^{4\times4}$, that transforms points from one frame to another. The use of sub- and superscripts indicates from- and to which frame the transformation is done. ${}^{0}_{1}\boldsymbol{T}$ is thus the transformation matrix from frame $\{1\}$ to $\{0\}$.

Translation vector

A vector, ${}^{b}\bar{p}_{\text{a,ORG}} \in \mathbb{R}^{3\times1}$, that is part of the transformation matrix and describes the translation of one frame relative to another. ${}^{0}\bar{p}_{1,\text{ORG}}$ is the translation from the origin of frame $\{1\}$ to $\{0\}$.

Rotation matrix

A matrix, ${}^{b}_{a}\boldsymbol{R} \in \mathbb{R}^{3\times3}$, that is part of the transformation matrix and describes the rotation of one frame relative to another. ${}^{0}_{1}\boldsymbol{R}$ is the rotation from frame $\{1\}$ to $\{0\}$.

Point Position

A vector, ${}^{a}\bar{p} \in \mathbb{R}^{3\times1}$, that describes the position of a point relative to frame $\{\text{a}\}$. ${}^{1}\bar{p}$ is a point position relative to frame $\{1\}$.

Origin of a Frame

${}^{a}\bar{O} \in \mathbb{R}^{3\times1}$ describes the origin of a frame. ${}^{1}\bar{O}$ is the origin of frame $\{1\}$.

4.2 Transformation Matrices

The purpose of transformation matrices is visualized in **Figure 4.1**, where a point ${}^{1}\bar{p}$ is defined with respect to the coordinate system frame {1}. If the point needs to be defined in the other coordinate system, frame {0}, a rotation and a translation needs to be done, to get the representation of the point as ${}^{0}\bar{p}$.

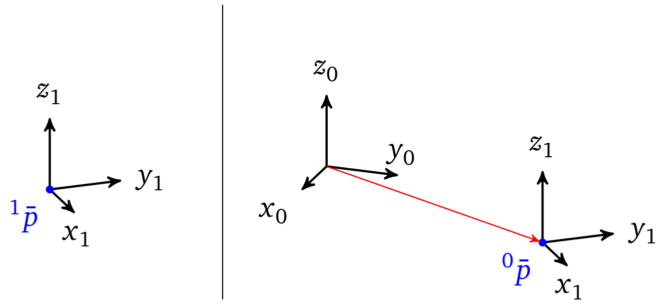


Figure 4.1: *The figure illustrates how a point ${}^{1}\bar{p}$ that is attached to the frame {1}, needs to be rotated and translated for the point to belong to the frame {0} instead.*

To describe the rotation and translation mathematically, **Equation (4.1)** is used. Here the transformation matrix ${}^{0}_{1}\boldsymbol{T}$ consists of a rotation matrix ${}^{0}_{1}\boldsymbol{R}$, a translation vector ${}^{0}\bar{p}_{1,\mathrm{ORG}}(x, y, z)$, a $0^{1\times3}$ zero row and a 1, which equals the mentioned size 4×4. The last row was added to avoid having a singular transformation matrix. To make multiplying with a position vector possible, a 1 should be added to it before multiplication.

$$\begin{bmatrix} {}^{0}\bar{p} \\ 1 \end{bmatrix} = {}^{0}_{1}\boldsymbol{T} \begin{bmatrix} {}^{1}\bar{p} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{0}_{1}\boldsymbol{R} & {}^{0}\bar{p}_{1,\mathrm{ORG}} \\ 0^{1\times3} & 1 \end{bmatrix} \begin{bmatrix} {}^{1}\bar{p} \\ 1 \end{bmatrix} \tag{4.1}$$

4.3 Denavit Hartenberg Convention

The Denavit Hartenberg convention is used as a standard way of describing the transformation from one coordinate system to another. To make such a transformation, first the coordinate systems (one per joint in the setup) have to be placed in a specific manner. Next, parameters describing translation and rotation are determined. Finally by inserting the parameters in a standard matrix the transformation is constructed. The Denavit Hartenberg convention used is the classical one, also described in [22]. The Denavit Hartenberg rules for placing the coordinate systems of the setup are shortly stated below (where only the rules relevant for the current setup are listed):

Step 1 Place the joint z axes, z_0, z_1, z_2... z_n, so that the joints rotate around their associated z axis.

Step 2 Place the base frame (origin ${}^{0}O$) on the z_0 axis and place the x_0 and y_0 axis to make right-handed coordinate system.

Step 3 Place the origin ${}^{i}O$ where the axis z_i and z_{i-1} common normal intersects z_i.

Step 4 Place x_i where the common normal between z_{i-1} and z_i goes through ${}^{i}O$ and also in that direction.

Step 5 Place y_i so that the right-hand frame is completed.

With those rules, coordinate systems are placed on top of the robot arm as shown in **Figure 4.2**. Since there are two revolute joints and one tool frame in the setup, three z axes are placed (z_0, z_1

and z_2). Since the tool frame placement is not specified by the rules above and has no function (no rotation or translation), the associated z_2 axis can be placed in a desired manner, which is why it is simply placed parallel to the other z axis, so that the x_2 axis can still obey step 4. For the base frame there are no rules for placing the x axis, and is therefore chosen to point downwards in the direction of gravity.

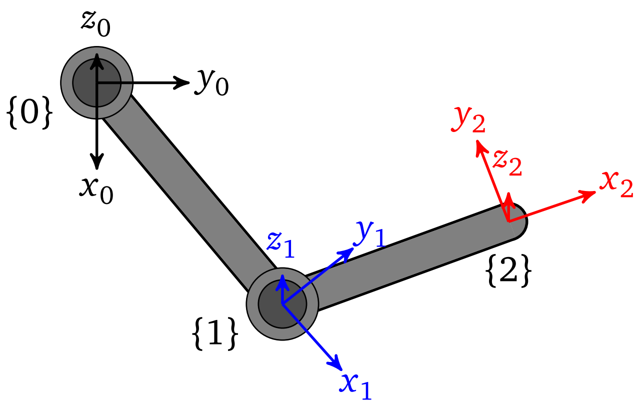


Figure 4.2: *The coordinate systems are placed on the setup with the use of Denavit Hartenberg approach.*

The Denavit Hartenberg notation uses following parameters to construct the transformation matrix.

- a_i: Distance from the intersection of the x_i and z_{i-1} axis to the origin ${}^{i}O$ along x_i.
- d_i: Distance from the intersection of the x_i and z_{i-1} axis to the origin ${}^{i-1}O$ along z_{i-1}.
- α_i: The angle between z_{i-1} and z_i measured around the x_i axis.
- θ_i: The angle between x_{i-1} and x_i measured around the z_{i-1} axis.

When using the stated rules, it is seen in **Figure 4.3** that the lengths a_1, a_2 and angles θ_1 and θ_2 are present and of relevance, while the lengths and angels d_i and α_i are zero.

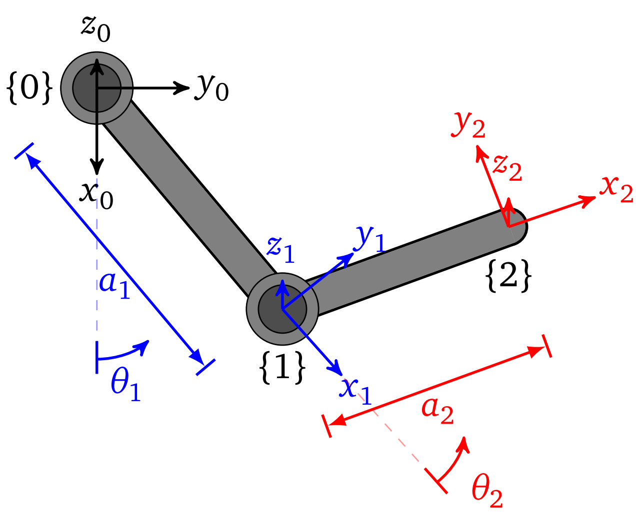


Figure 4.3: *The variable lengths a_1, a_2 and angles θ_1 and θ_2 are placed with the use of the Denavit Hartenberg approach.*

The Denavit Hartenberg approach is smart and simple because the method limits itself to only deal with rotations around the x and z axis as well as only the translations in the direction of the x and z axis. By placing the coordinate systems according to the specified rules, all rotation and translation about the y axis is avoided. The way of setting up the coordinate systems fits with a standard transformation matrix which can be used in all cases as long as the rules are followed. With the parameters and the standard transformation matrix, a transformation describing the translation and rotation between frame $\{2\}$ and $\{1\}$ can be created as well as one for frame $\{1\}$ and $\{0\}$. The Denavit Hartenberg standard matrix is shown in **Equation (4.2)** where one inputs the found parameters a_i, d_i, α_i and the joint variable, θ_i, to get the transformation between frame $\{i\}$ and $\{i-1\}$.

$$
{}^{i-1}_{i}\boldsymbol{T} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\alpha_i) & \sin(\theta_i)\sin(\alpha_i) & a_i\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i)\cos(\alpha_i) & -\cos(\theta_i)\sin(\alpha_i) & a_i\sin(\theta_i) \\ 0 & \sin(\alpha_i) & \cos(\alpha_i) & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.2}
$$

4.4 Forward Kinematics of the Robot Arm

The parameters of importance for the current setup are also shown in **Table 4.1** for overview:

joint i	a_i	α_i	d_i	θ_i
1	330 mm	0	0	θ_1
2	227 mm	0	0	θ_2

Table 4.1: *The table gives an overview of the parameters that are used for the Denavit Hartenberg matrix.*

With **Equation (4.2)** and **Table 4.1** the following transformation matrices are derived:

$$
{}^{0}_{1}\boldsymbol{T} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & a_1\cos(\theta_1) \\ \sin(\theta_1) & \cos(\theta_1) & 0 & a_1\sin(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.3}
$$

$$
{}^{1}_{2}\boldsymbol{T} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & a_2\cos(\theta_2) \\ \sin(\theta_2) & \cos(\theta_2) & 0 & a_2\sin(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}
$$

The transformation of the end frame relative to the base frame can now be derived as shown in **Equation (4.5)** which concludes the forward kinematics.

$$
{}^{0}_{2}\boldsymbol{T} = {}^{0}_{1}\boldsymbol{T}\,{}^{1}_{2}\boldsymbol{T} = \begin{bmatrix} \cos(\theta_1+\theta_2) & -\sin(\theta_1+\theta_2) & 0 & a_1\cos(\theta_1)+a_2\cos(\theta_1+\theta_2) \\ \sin(\theta_1+\theta_2) & \cos(\theta_1+\theta_2) & 0 & a_1\sin(\theta_1)+a_2\sin(\theta_1+\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.5}
$$

Dynamic Model 5

This chapter explains the dynamics of the AXO-arm. The dynamic model is based on the Euler Lagrange Formulation, which is a method for modeling a serial manipulator [21][23][22]. The chapter will be structured according to following:

- First the Euler Lagrange equation is described.
- Second the potential and kinetic energy of the links are derived.
- Then the Lagrange D'Alambert Principle is used to add actuator- and friction torques to the model.
- Lastly, a model of the electric motors and their attached gears is derived.

5.1 The Euler Lagrange Equation

The Euler Lagrange Formulation is an energy based way of modeling a mechanical system, and in this case the serial manipulator. A central part of the formulation is to make use of the Euler Lagrange equation, since that is a tool for constructing a dynamic model of a mechanical system when the energy of it is known. The Euler Lagrange equation is explained in this section, while the energy of the serial manipulator is described in **Section 5.2** and **Section 5.3**. To understand the Euler Lagrange equation, first generalized coordinates are explained, which are used in the calculations.

Generalized Coordinates

To understand generalized coordinates, consider first a group of particles that are connected and together form an object. In the case of a serial manipulator, these particles could be links. If the motion of every particle is a result of the force applied, and all particles are to be considered when describing the motion of the object, it would be very comprehensive. The reason is that the connection between the particles would make up constraints in the calculation of the system configuration at any given moment which is computationally complex.

To simplify the calculations, we can describe the configuration of the system in *generalized coordinates*, which is a smaller variable set (specific for the system). A popular choice of generalized coordinates when describing serial manipulators is the set of joint parameters, $\bar{\theta}$. The AXO-arm has only revolute joints, and the joint variables are therefore angles. This reformulation allows us to describe the equations of motion of the system with respect to joint angles. [23]

The Euler Lagrange Equation

The model of the serial manipulator is based on the Euler Lagrange formulation. The Euler Lagrange equation is therefore explained in this section. To do that, first the Lagrangian, $\mathscr{L}$, is defined as shown in **Equation (5.1)**.

$$\mathscr{L}(\bar{\theta}(t), \dot{\bar{\theta}}(t)) = K(\bar{\theta}(t), \dot{\bar{\theta}}(t)) - U(\bar{\theta}(t)) \qquad [\mathrm{J}] \qquad (5.1)$$

Here U is the potential energy and K the kinetic energy. It should be noticed that the potential and kinetic energy are both described in generalized coordinates. A serial manipulator can contain several links. $\bar{\theta}(t)$ and $\dot{\bar{\theta}}(t)$ are therefore vectors.

$$\bar{\theta}(t) = \begin{bmatrix} \theta_1(t) \\ \theta_2(t) \\ \vdots \\ \theta_n(t) \end{bmatrix} \qquad [\mathrm{rad}] \tag{5.2}$$

$$\dot{\bar{\theta}}(t) = \frac{\mathrm{d}\bar{\theta}(t)}{\mathrm{d}t} = \begin{bmatrix} \dot{\theta}_1(t) \\ \dot{\theta}_2(t) \\ \vdots \\ \dot{\theta}_n(t) \end{bmatrix} \qquad [\mathrm{rad/s}] \tag{5.3}$$

The key concept of the Euler Lagrange equation is that a mechanical system will always move along the path, that consumes the least amount of energy. The Euler Lagrange equation is derived by minimizing the Lagrangian. The Euler Lagrange equation is stated in **Equation (5.4)**.[24] [23]

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\partial \mathscr{L}(t,\bar{\theta}(t),\dot{\bar{\theta}}(t))}{\partial \dot{\bar{\theta}}(t)} - \frac{\partial \mathscr{L}(t,\bar{\theta}(t),\dot{\bar{\theta}}(t))}{\partial \bar{\theta}(t)} = 0 \qquad [\mathrm{J}] \tag{5.4}$$

To make use of **Equation (5.4)** to describe the dynamics of the system, first the kinetic and potential energy of the links has to be derived. It should be noted that the Euler Lagrange equation stated in **Equation (5.4)** is only valid for conservative systems, but the equation will in **Section 5.4** be expanded to include non-conservative forces.

5.2 Potential Energy

To describe the potential energy of the AXO-arm, first a single link is considered, followed by a description of the whole arm. Potential energy is only related to gravity. In Denmark the gravitational acceleration is $g = 9.816\,\mathrm{m/s^2}$, resulting a gravitational force vector,

$${}^{0}\bar{g} = \begin{bmatrix} 9.816 \\ 0 \\ 0 \end{bmatrix}, \qquad [\mathrm{m/s^2}] \tag{5.5}$$

which is shown with brown on **Figure 5.1**.

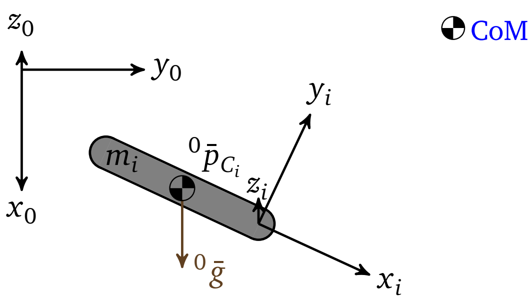


Figure 5.1: *The gravity vector $\bar{g}$ affects the body in its Center of Mass (CoM).*

The link, with mass m_i and position ${}^0\bar{p}_{C_i}$ described in base frame, is affected by gravity in the x direction. The vector ${}^0\bar{p}_{C_i}$ is the position of the CoM of the link. An equation of the link potential energy is formulated as:

$$U_i = -\, m_i\ {}^0\bar{g}^{\mathrm{T}}\ {}^0\bar{p}_{C_i} \qquad [\mathrm{J}] \qquad (5.6)$$

The point ${}^0\bar{p}_{C_i}$ is a function of the joint angles of the serial manipulator. For deriving ${}^0\bar{p}_{C_i}$ **Equation (4.1)** is used. For clarity **Equation (4.1)** is shown below in a simpler form:

$${}^0\bar{p}_{C_i} = {}^0_i\boldsymbol{R}\ {}^i\bar{p}_{C_i} + {}^0\bar{p}_{\mathrm{i,ORG}} \qquad (5.7)$$

Here the rotation matrix ${}^0_i\boldsymbol{R}$ is dependent on the joint angles. With the potential energy of a single link derived, the total potential energy of the AXO-arm related to the base frame can be calculated using **Equation (5.8)**. The energies can be summed because they are all defined in the same coordinate frame.

$$U = \sum_{i=1}^{N} U_i = \sum_{i=1}^{N} m_i\ {}^0\bar{g}^{\mathrm{T}}\ {}^0\bar{p}_{C_i} \qquad [\mathrm{J}] \qquad (5.8)$$

5.3 Kinetic Energy

The second half of **Equation (5.1)** is the kinetic energy in the serial manipulator. To describe it, a single link is considered at first, followed by a description of the whole arm. Kinetic energy depends on both linear (${}^0\bar{v}_{C_i}$) and rotational velocity (${}^0\dot{\bar{\theta}}_i$); here **Figure 5.2** shows these two types of velocity. Both velocities are related to the link CoM.

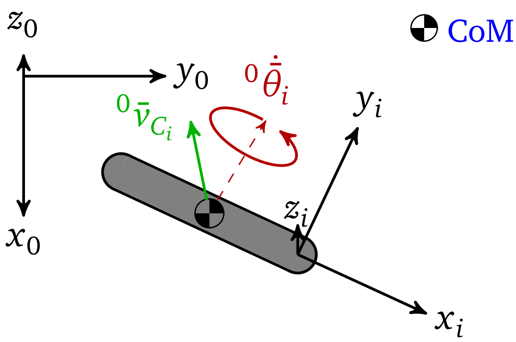


Figure 5.2: *The velocity of the link is both related to a linear (${}^0\bar{v}_{C_i}$) and angular velocity (${}^0\dot{\bar{\theta}}_i$) about the center of mass.*

As described in [23] the kinetic energy is expressed as in **Equation (5.9)**. Here ${}^0\boldsymbol{I}_{C_i}(\bar{\theta}(t))$ is the inertia of the link about its CoM related to the base frame. The link mass is m_i.

$$K_i = \frac{1}{2} m_i\ {}^0\bar{v}_{C_i}^{\mathrm{T}}\ {}^0\bar{v}_{C_i} + \frac{1}{2}\ {}^0\dot{\bar{\theta}}_i^{\mathrm{T}}\ {}^0\boldsymbol{I}_{C_i}\ {}^0\dot{\bar{\theta}}_i \qquad [\mathrm{J}] \qquad (5.9)$$

To obtain the linear velocity of the CoM with respect to the base frame, ${}^0\bar{v}_{C_i}$, **Equation (5.7)** is used again since ${}^0\bar{v}_{C_i} = \frac{\mathrm{d}}{\mathrm{d}t}({}^0\bar{p}_{C_i})$:

$${}^0\bar{v}_{C_i} = \frac{\mathrm{d}}{\mathrm{d}t}({}^0\bar{p}_{C_i}) = \frac{\mathrm{d}}{\mathrm{d}t}\left({}^0\bar{p}_i + {}^0_i\boldsymbol{R}\ {}^i\bar{p}_{C_i}\right) \qquad [\mathrm{m/s}] \qquad (5.10)$$

As given in **Equation (5.11)** The rotational velocity, ${}^{0}\dot{\bar{\theta}}_i$, of joint {i} with respect to the base frame, is calculated as the sum of all previous joint velocities j and the local angular velocity itself i. Every rotational velocity has to be related to the base frame which is done using the rotational matrix $\boldsymbol{R}$.

$$ {}^{0}\dot{\bar{\theta}}_i = {}^{0}\dot{\bar{\theta}}_{i-1} + {}^{0}_{i-1}\boldsymbol{R}\, {}^{i-1}\dot{\bar{\theta}}_i = \sum_{j=1}^{i} {}^{0}_{j-1}\boldsymbol{R}\, {}^{j-1}\dot{\bar{\theta}}_j \qquad [\mathrm{rad/s}] \qquad (5.11) $$

Note that the local angular velocity ${}^{i-1}\dot{\bar{\theta}}_i$ is not defined with relation to the link CoM, but the link frame {i}. This is because the orientation of the link CoM is the same as the link frame and angular velocity is only dependent on orientation. Like the angular velocity, the link inertia is also only dependent on the orientation. Therefore in order for it to be described with respect to the baseframe, only a rotation of the inertia tensor is needed as shown in **Equation (5.12)**. Here ${}^{i}\boldsymbol{I}_{C_i}$ is the inertia tensor of link i, around the CoM defined in frame {i}. It should be noted that the inertia tensor is constant when described in its local frame.

$$ {}^{0}\boldsymbol{I}_{C_i}(\bar{\theta}) = {}^{0}_{i}\boldsymbol{R}(\bar{\theta})\, {}^{i}\boldsymbol{I}_{C_i}\, {}^{0}_{i}\boldsymbol{R}^{\mathrm{T}}(\bar{\theta}) \qquad [\mathrm{N\,m}] \qquad (5.12) $$

With an expression of the link kinetic energy, the Kinetic energy of the AXO-arm can be calculated as the sum of all the link kinetic energies:

$$ K = \sum_{i=1}^{N} K_i = \sum_{i=1}^{N} \left(\frac{1}{2}\, m_i\, {}^{0}\bar{v}_{C_i}^{\mathrm{T}}\, {}^{0}\bar{v}_{C_i} + \frac{1}{2}\, {}^{0}\dot{\bar{\theta}}_i^{\mathrm{T}}\, {}^{0}\boldsymbol{I}_{C_i}\, {}^{0}\dot{\bar{\theta}}_i \right) \qquad [\mathrm{J}] \qquad (5.13) $$

5.4 The Euler Lagrange Formulation Including Non-conservative Torques

In **Section 5.1** the Euler Lagrange equation with the conservative torques was shown as **Equation (5.4)**. This is only a model of the conservative forces in the system, created by the potential- and kinetic energy. A conservative force is one where energy does not enter or leave the system, e.g. when potential energy is added by elevating an object, the same amount of energy can be released by dropping it. The aim of this section is to include non-conservative torques such as friction and actuator torque in the model. The Lagrange-D'Alembert Principle is used to include non-conservative torques in the Euler Lagrange equation. By letting $\bar{Q}$ contain all the non-conservative torques, the Euler Lagrange equation can be modified into:

$$ \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathscr{L}(\bar{\theta}, \dot{\bar{\theta}})}{\partial \dot{\bar{\theta}}} - \frac{\partial \mathscr{L}(\bar{\theta}, \dot{\bar{\theta}})}{\partial \bar{\theta}} = \bar{Q} \qquad [\mathrm{N\,m}] \qquad (5.14) $$

Now by use of the results from **Section ??**, the Lagrangian can be described in terms of the kinetic and potential energy:

$$ \frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial (K(\bar{\theta}, \dot{\bar{\theta}}) - U(\bar{\theta}))}{\partial \dot{\bar{\theta}}} - \frac{\partial K(\bar{\theta}, \dot{\bar{\theta}}) - U(\bar{\theta})}{\partial \bar{\theta}} = \bar{Q} \qquad [\mathrm{N\,m}] \qquad (5.15) $$

By calculating the derivatives of **Equation (5.15)**, we get:

$$ \boldsymbol{M}(\theta)\ddot{\bar{\theta}} + \bar{V}(\bar{\theta}, \dot{\bar{\theta}}) + \bar{G}(\bar{\theta}) = \bar{Q} \qquad [\mathrm{N\,m}] \qquad (5.16) $$

The elements derived from solving the derivative are divided between $\boldsymbol{M}$, $\bar{V}$, and $\bar{G}$ where:

- The $\boldsymbol{M}$ matrix contains all the terms from the derivation that is multiplied by the rotational acceleration $\ddot{\theta}$. The terms contain parts of the link inertia tensors and link masses and as a whole $\boldsymbol{M}$ describes the inertia of the system.
- The $\bar{V}$ vector contains all the elements that the angular velocity is a part of. It describes how centrifugal forces and Coriolis fictive forces are influencing the AXO-arm as torques.
- The $\bar{G}$ vector contains the elements that are only dependent on the angular position. The $\bar{G}$ vector is derived purely from the potential energy and describes how the gravitational force is effecting the AXO-arm as torques.

Next the non-conservative torques(external torques) contained in $\bar{Q}$ are shown in **Equation (5.17)**, where the $\bar{F}$ vector describes the friction of the system and $\bar{\tau}_{\text{in}}$ is a vector describing the actuator torques.

$$\bar{Q} = \bar{\tau}_{\text{in}} - \bar{F}(\dot{\bar{\theta}}) \tag{5.17}$$

Finally the derived result of the Euler Lagrange equation with non-conservative torques included is written in **Equation (5.18)**.

$$\bar{\tau}_{\text{in}} = \boldsymbol{M}(\theta)\ddot{\bar{\theta}} + \bar{V}(\bar{\theta}, \dot{\bar{\theta}}) + \bar{G}(\bar{\theta}) + \bar{F}(\dot{\bar{\theta}}) \tag{5.18}$$

It should be noted that the motor internal dynamics are not included in the equation yet, which is the aim of the following section.

5.5 Addition of the Motor Internal Dynamics to the AXO-arm dynamics

This section contains a model of the motors that are used in the AXO-arm. It is structured in the following way:

- First the motors for the AXO-arm setup are described and it is explained how these can be approximated with a DC motor model.
- It is presented how a current controller can benefit the AXO-arm model and how it influences the motor dynamics.
- The effect of the gear on the motor dynamics is accounted for.
- It is studied how the motor dynamics can be implemented in the AXO-arm dynamics to express a final equation of the robot arm dynamics.

5.5.1 Motor Modeling

This section presents the motor model that is used in this project. In each of the actuated joints reside a brushless Direct Current (DC) motor, or equivalently EC motor. The brushless motors are of the type Maxon 412825 and 40268. The main structural difference between brushed and brushless DC motors is that in the brushed case, the coil is on the rotor and the magnet is on the stator, and in the brushless case the magnet is on the rotor and the coils are on the stator. Having the coils on the stator gives a lower friction because there is no need for electrical connection between stationary and moving parts. This also reduces the wear on the motor, and improves the efficiency. [25]

Controlling the EC motor takes more effort than a DC motor. Typically, as is also the case in this project, EC motors have three phases, each connected to a set of coils. The principle of operation relies on the three phases being connected to AC power where each line is 120° separated in phase. This allows the EC motor to move the magnetic field from one coil to the next. The turn direction is determined by the relative phase displacement between adjacent coils.

The dynamic model of an EC motor is similar to that of a regular DC motor. A regular DC motor model is based on the equivalent electrical and mechanical system shown in **Figure 5.3**. The following

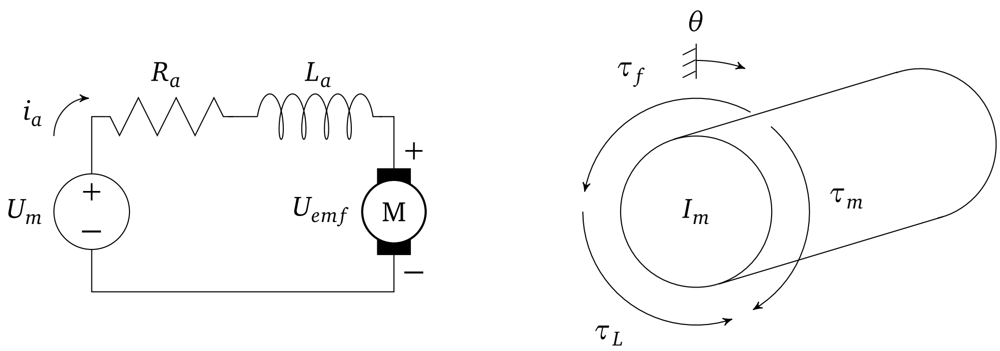


Figure 5.3: *Electrical model of a DC motor, and a rotational free body diagram of the motor axle.*

equations define the dynamics.

$$U_m(t) = i_a(t)R_a + L_a \frac{\mathrm{d}}{\mathrm{d}t} i_a(t) + U_{emf}(t) \qquad [\mathrm{V}] \qquad (5.19)$$

$$I_m(t)\,\ddot{\theta} = \tau_m(t) - \tau_L(t) - \tau_f(t) \qquad [\mathrm{N}] \qquad (5.20)$$

$$U_{emf}(t) = k_v\,\dot{\theta}(t) \qquad [\mathrm{V}] \qquad (5.21)$$

$$\tau_m(t) = k_t\,i_a(t) \qquad [\mathrm{N\,m}] \qquad (5.22)$$

$$\tau_f(t) = b\,\dot{\theta}(t) + \mathrm{sign}\big(\dot{\theta}(t)\big)\;\tau_c(t) \qquad [\mathrm{N\,m}] \qquad (5.23)$$

The motor back emf, U_{emf}, is modeled in **Equation (5.21)** as being linear in motor velocity, $\dot{\theta}$. The motor torque, τ_m, is linear in the coil current, and the friction is assumed to consist of viscous friction and coulomb friction, as shown in **Equation (5.23)**. The torque from the attached load is denoted τ_L. **Equation (5.19)-(5.20)** show the total model of both the electrical and the mechanical system of a DC motor.

The expansion of the DC motor model to make it fit that of an EC motor, includes taking into account the three phases in the EC motor. Thus, there are three electrical diagrams equivalent to that of **Figure 5.3**, but each with an i_a and U_m that is phase shifted 120°. An example of the three i_a signals is shown in **Figure 5.4**. In the EC motor case, the model equations are:

$$U_{m_n}(t) = i_{a_n}(t)R_a + L_a \frac{\mathrm{d}}{\mathrm{d}t} i_{a_n}(t) + U_{emf}(t), \quad \text{where} \;\; n \in \{1,2,3\} \qquad [\mathrm{V}] \qquad (5.24)$$

$$I_m(t)\,\ddot{\theta} = \sum_{n=1}^{3}\left(\tau_{m_n}\right) - \tau_L(t) - \tau_f(t) \qquad [\mathrm{N\,m}] \qquad (5.25)$$

$$U_{emf}(t) = k_v\,\dot{\theta}(t) \qquad [\mathrm{V}] \qquad (5.26)$$

$$\tau_{m_n}(t) = k_t\,i_{a_n}(t), \;\text{where}\; n \in \{1,2,3\} \qquad [\mathrm{N\,m}] \qquad (5.27)$$

$$\tau_f = b\dot{\theta} + \mathrm{sign}\big(\dot{\theta}\big)\;\tau_c \qquad [\mathrm{N\,m}] \qquad (5.28)$$

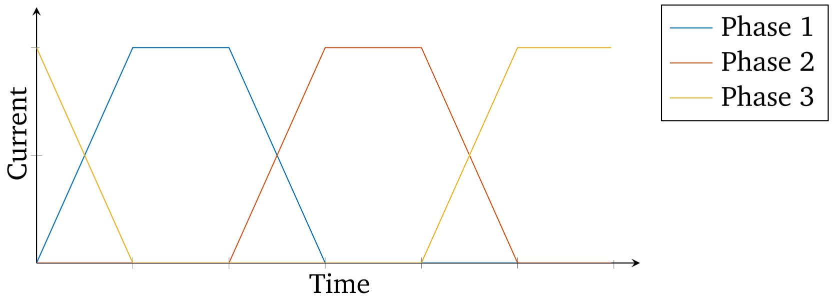


Figure 5.4: *The shape of the current signals in the different phases.*

As **Figure 5.4** indicates, the sum of the three i_a signals is constant. Therefore, once the generation of the three different drive signals to the coils is in place, the model can be simplified to that of a regular DC motor.

5.5.2 Mode of Operation

The motors are controlled using ESCON 50-5 drivers. This type of driver allows operation with either velocity or current reference signals. For this project, current control is desired since the current is proportional to the torque that the motors apply, as seen in **Equation (5.22)**. Therefore, by controlling the current, the torque is also controlled.

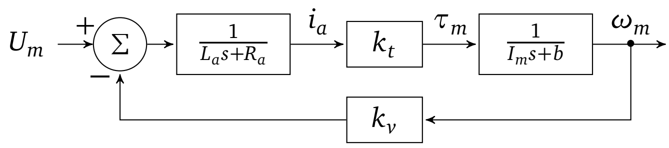
As mentioned in **Section 2.1.3**, the ESCON drivers do not support open loop current control. It is therefore necessary to implement a closed loop current controller. A block diagram of the DC motor is shown in **Figure 5.5a**, and it is expanded to include the current controller in **Figure 5.5b**. For simplicity, the friction is kept as a linear term $\tau_f = b\,\omega_m$, but in the next section, it again takes the form of **Equation (5.28)**. Note that in this section ω_m is the rotational velocity of the motor axle ($\dot{\theta}(t)$).

The block CC, in **Figure 5.5**, is the current controller, and in the present case with the two motors, the auto tuned controller parameters are shown in **Table 5.1**. The controllers are auto tuned with Maxon Studio and for both motors a PI controller is used.

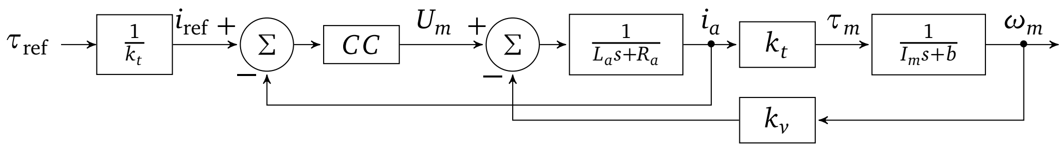
	Time Constant	Proportional part
Maxon 412825	631 µs	260
Maxon 402687	658 µs	1260

Table 5.1: *PI Controller Parameters found by autotuning in Maxon Studio*

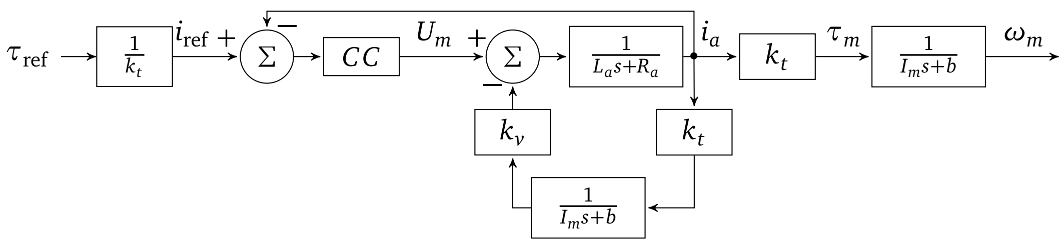
With the current controller and its parameters known, it is next studied how the current controller effects the dynamics of the motors. In figure **Figure 5.5c** the DC motor block diagram including the current controller is rearranged. By doing that, the system can be seen as two parts in series. The first part is from a torque reference τ_{ref} to a motor torque τ_m and the second part is the pure mechanics from τ_m to ω_m. Since the current controller time constants are very fast and the electrical time constants of the motors are also assumed to be very fast, a hypothesis is that the first part of the block diagram from τ_{ref} to τ_m can be approximated as a gain of 1, if k_t is known. This of course has to be justified since the assumption about the fast time constants is uncertain and also because it is seen that the mechanics of the motor is also a part of the transfer function from τ_{ref} to τ_m. To justify the hypothesis, measurements of the motor current are captured in order to find time constants of the transfer function from τ_{ref} to τ_m. The current reference is also saved in order to define the initial time. Additionally the motor velocity is measured to find a time constant of the whole motor dynamics from τ_{ref} to ω_m. By comparing the two time constants it can be justified whether the time



(a) *DC motor blockdiagram*



(b) *DC motor blockdiagram including current control*



(c) *Rearranged blockdiagram showing where to measure*

Figure 5.5

constant of the transfer function from τ_{ref} to τ_m is sufficiently fast and insignificant to approximate the transfer function from τ_{ref} to τ_m as a gain of 1.

From the measurement report in **Appendix A** it is seen that the electrical time constant is around 50 to 60 ms, while the mechanical time constant is at least 500 to 600 ms (for both joints). The time constant of the transfer function from τ_{ref} to τ_m is thus deemed insignificant when compared to the mechanical time constant, which reduces the block diagram in **Figure 5.5c** to only consist of the motor mechanics as shown in **Equation (5.29)**. To summarize, by using current control, the electrical part of the motor can be neglected in the model, so that the focus is only on the mechanics.

$$\frac{\omega_m}{\tau_{ref}} = \frac{1}{k_t} k_t \frac{1}{I_m s + b} = \frac{1}{I_m s + b} \tag{5.29}$$

The mechanical equation given in **Equation (5.29)** is of course not the true one because of the simplified friction, but is simply stated to give an idea of why the electrical part vanishes from the equation. The true mechanics of the motor are given in **Section 5.5.1** where the equation used further is **Equation (5.20)**. Additionally it should be noted that in this section, the motor was without any load for simplicity, but since the AXO-arm is a load, it is added in the following section.
From **Appendix A**, also notice that there is no steady state error from τ_{ref} to τ_m, which means that the steady state requirement described in **Section 3.2.1** is fulfilled.

5.5.3 The Gears' Effect on the Motor Dynamics

In this section the gears are included in the mechanical model, to couple the motor dynamics and the AXO-arm dynamics. On each of the motors, a gear is attached. It is harmonic drive gear with a ratio of 50/1. It is modeled merely as a gain on the motor angle, velocity, acceleration and torque.

$$\theta = \frac{1}{50}\,\theta_m \qquad [\mathrm{rad}] \tag{5.30}$$

$$\dot{\theta} = \frac{1}{50}\,\dot{\theta}_m \qquad [\mathrm{rad/s}] \tag{5.31}$$

$$\ddot{\theta} = \frac{1}{50}\,\dot{\theta}_m \qquad \left[\mathrm{rad/s^2}\right] \tag{5.32}$$

$$\tau_{\mathrm{in}} = 50\,\tau_m \qquad [\mathrm{N\,m}] \tag{5.33}$$

Where θ is the angle of a joint on the AXO-arm, or equivalently the output angle of the gear.

With basis in **Equation (5.20)**, τ_L is studied. It is the torque applied from the load and since the load is the AXO-arm, it must include some inertia. Therefore the AXO-arm loads the motor with a torque that is equal the product of inertia, joint angular acceleration and the gear ratio. By expanding τ_L, **Equation (5.20)** can thereby be reformulated as:

$$I_m\,\ddot{\theta}_m + \ddot{\theta}\,I_L\,\frac{1}{N} = \tau_m - \tau_{f_m}(\dot{\theta}_m) \qquad [\mathrm{N\,m}] \tag{5.34}$$

Where I_m and I_L are motor- and load inertia, N is the gear ratio, $\dot{\theta}$ and $\ddot{\theta}$ are the angular velocity and -acceleration of the AXO-arm and $\dot{\theta}_m$ and $\ddot{\theta}_m$ are the angular velocity and -acceleration of the joint motor. As seen in **Equation (5.34)**, $\ddot{\theta}\,I_L\,\frac{1}{N}$ is the torque from the inertia part of the load. The friction torque of the motor $\tau_{f_m}(\dot{\theta}_m)$ is kept as a function for simplicity and because it is modeled as nonlinear. For the load inertia torque, the inverse gear factor $\frac{1}{N}$ is multiplied to transform it into torque on the motor axle. Using **Equation 5.30–5.33** we get:

$$\tau_m = \ddot{\theta}\,I_m\,N + \ddot{\theta}\,I_L\,\frac{1}{N} + \tau_{f_m}(\dot{\theta}_m) \qquad [\mathrm{N\,m}] \tag{5.35}$$

$$\tau_m = \ddot{\theta}\,\frac{1}{N}\,(I_m\,N^2 + I_L) + \frac{1}{N}\,\tau_{f_m}(\dot{\theta}\,N)\,N \qquad [\mathrm{N\,m}] \tag{5.36}$$

$$\tau_m\,N = \ddot{\theta}\,(I_m\,N^2 + I_L) + \tau_{f_m}(\dot{\theta}\,N)\,N \qquad [\mathrm{N\,m}] \tag{5.37}$$

Where **Equation (5.37)** is the final result of this section, since the gear is now implemented in the motor dynamics.

5.5.4 Implementing the Motor Dynamics in the AXO-arm Dynamics

To include the motor dynamics given in **Equation (5.37)** into the AXO-arm dynamics given in **Equation (5.18)**, these equations are at first compared. For clarity both equation are restated below:

$$\tau_m\,N = \ddot{\theta}\,(I_m\,N^2 + I_L) + \tau_{f_m}(\dot{\theta}\,N)\,N \qquad [\mathrm{N\,m}] \tag{5.38}$$

$$\bar{\tau}_{\mathrm{in}} = \boldsymbol{M}(\bar{\theta})\ddot{\bar{\theta}} + \bar{V}(\bar{\theta},\dot{\bar{\theta}}) + \bar{G}(\bar{\theta}) + \bar{F}(\dot{\bar{\theta}}) \qquad [\mathrm{N\,m}] \tag{5.39}$$

It is important to note that **Equation (5.38)** is the mechanical model for a single motor, while **Equation (5.39)** is the vector valued model of the entire AXO-arm system without motor dynamics. Parts of the system are therefore described by both equations. The load term, I_L, corresponds to a row of the $\boldsymbol{M}$ matrix, and the friction torque, τ_{f_m}, corresponds to a row of $\bar{F}(\dot{\bar{\theta}})$. The motor inertia, I_m, can easily be added to the total system inertia matrix, $\boldsymbol{M}$, due to the motor being placed in the origin of the joint coordinate system. **Equation (5.38)-(5.39)** can thus be combined to form the complete model of both the serial manipulator and the attached motor dynamics, where $\boldsymbol{N}$ is a diagonal matrix with gear ratios on the diagonal and $\boldsymbol{I}_m$ is a diagonal matrix with motor inertias on the diagonal:

$$\bar{\tau}_{\text{in}} = \bar{\tau}_m \boldsymbol{N} \qquad [\text{N m}] \qquad (5.40)$$

$$\bar{\tau}_m \boldsymbol{N} = (\boldsymbol{I}_m \boldsymbol{N}^2 + \boldsymbol{M}(\bar{\theta}))\ddot{\bar{\theta}} + \bar{V}(\bar{\theta}, \dot{\bar{\theta}}) + \bar{G}(\bar{\theta}) + \bar{F}(\dot{\bar{\theta}}\, \boldsymbol{N})\, \boldsymbol{N} \qquad [\text{N m}] \qquad (5.41)$$

which can also be written as **Equation (5.42)**.

$$\ddot{\bar{\theta}} = (\boldsymbol{I}_m \boldsymbol{N}^2 + \boldsymbol{M}(\bar{\theta}))^{-1}(\bar{\tau}_m \boldsymbol{N} - \bar{V}(\bar{\theta}, \dot{\bar{\theta}}) - \bar{G}(\bar{\theta}) - \bar{F}(\dot{\bar{\theta}}\, \boldsymbol{N})\, \boldsymbol{N}) \qquad \left[\text{rad/s}^2\right] \qquad (5.42)$$

To make **Equation (5.42)** more intuitive, it is shown as a block diagram in **Figure 5.6**. **Equation (5.42)** is the final result of this chapter since it describes the entire AXO-arm dynamics.

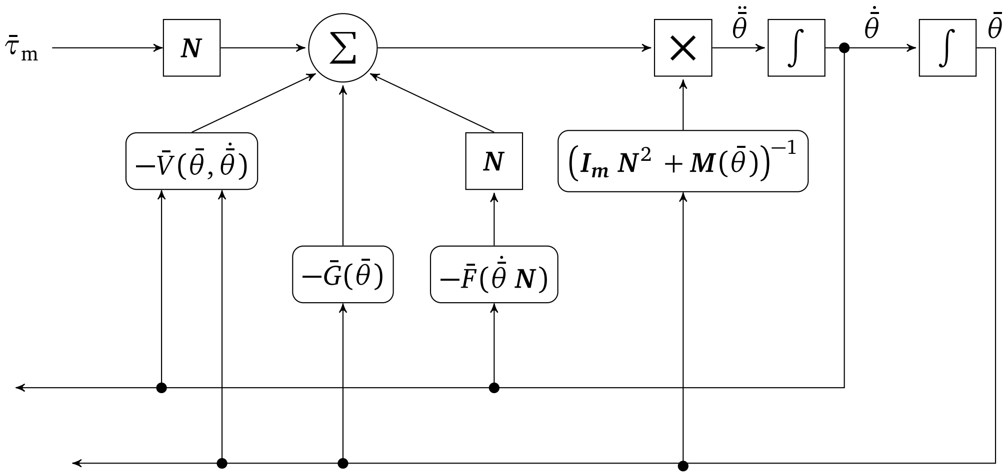


Figure 5.6: *Block of the AXO-arm model. Rounded edges indicates that the block is a function of the input signal*

Parameter Estimation of the Dynamic Model 6

Having found a dynamic model of the exoskeleton arm, it is now necessary to parameterize it. This chapter describes which parameters that are known and which we need to estimate using measured data. **Section 6.1** presents an overview of the model parameters, and how they each are determined. Some of the model parameters will be estimated using white-box parameter estimation and others will be calculated from production drawings. The model we use for the white-box parameter estimation is summarized in **Section 6.1.1**. After that, **Section 6.2**, accounts for the parameter estimation tests that are conducted and the parameters are calculated. Lastly, a validation of the model and parameters is done.

6.1 Model Parameters

The inputs to the dynamic model in **Chapter 5** are torques. The motors and their drivers do not allow torque input, as written in **Section 2.1.3**. However, they do allow current control. In **Chapter 5** we state that the torque of a DC-motor is proportional to the current with the torque constant k_t. It is thus, indirectly, possible to set torque inputs to the system if k_t is known, and the controller on the ESCON driver is properly tuned. The torque constant for the motors can be found in the datasheet, but due to tolerances, it is deemed best to determine these experimentally. It is thus one of the parameters that will be estimated. [26]

The parameters that describe the physical dimensions of the system matrix, $\boldsymbol{M}$, and the vector, $\bar{V}$, are calculated from the CAD drawings. In the CAD drawings we find the CoM, and tensor matrices, which can be transformed to the global reference frame, using kinematics. We assume that the parts are machined sufficiently precise for the mass, center of mass, and inertia to fit with the CAD models.

Gravitational force is assumed constant and equal to $9.816\,\mathrm{m/s^2}$ pointing downwards in the global reference frame.

The friction force in the system is unknown. **Chapter 5** suggests a friction model in the motor that consists of viscous friction and coloumb friction. It turns out, that this type of model fits the system quite well, as shown in **Section 6.2**. To summarize, it is necessary to estimate the following parameters:

- Motor torque constant, k_t for each of the two motors.
- Viscous friction coefficient b for each of the two joints.
- Coloumb friction τ_c for each of the two joints.

6.1.1 Model for White-box Parameter Estimation

Serial manipulators are a well-researched subject, with a known closed for model structure. It is therefore chosen to estimate the parameters using a white-box method, which is a system identification technique where unknown parameters are fitted to a known model.

To simplify the white-box model used for experimental estimation of torque constant and friction, the AXO-arm has been disassembled into its two joints and corresponding links. This allows us to neglect any effect of one joint on the other, and since we are not estimating any inertias with this experiment, disassembling the arm will not pose any problems. Furthermore, to remove the effect of gravity on the measurements, the two joints are placed on the side so that they move in a horizontal plane. In that configuration, the joints have vertical axes of rotation, and it is therefore reasonable to assume no influence from gravity. A photo of the configuration is shown in **Figure ??**. The motion of a single joint can now be described by following differential equations.

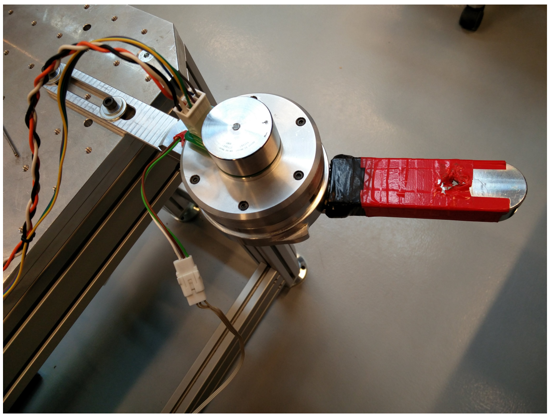


Figure 6.1: *Foto of the elbow joint during testing. It is mounted so it moves in a horizontal plane.*

$$I_m\,\ddot{\theta}_m(t) = \tau_m(t) - \tau_f(\dot{\theta}_m, t) \qquad [\mathrm{N\,m}] \qquad (6.1)$$

$$\tau_m(t) = k_t\, i_a(t) \qquad [\mathrm{N\,m}] \qquad (6.2)$$

$$\tau_f(t) = b\,\dot{\theta}_m(t) + \mathrm{sign}(\dot{\theta}_m(t))\,\tau_c \qquad [\mathrm{N\,m}] \qquad (6.3)$$

The sign function is implemented as the sigmoid function. The makes the model continuous, and thus improves simulation speed. It also solves the problem of defining $\mathrm{sign}\left(\dot{\theta}\right)_m$ when $\dot{\theta}_m = 0$.

$$\mathrm{sgm}(\dot{\theta}_m, \sigma) = \frac{2}{1+\exp(-\sigma\,\dot{\theta}_m)} - 1 \qquad (6.4)$$

shown in **Figure 6.2**.

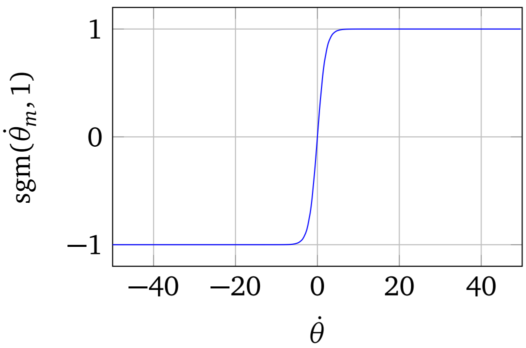


Figure 6.2: *The sigmoid function used in stead of sign, to keep differentiability. The parameter σ determines the slope, and here $\sigma = 1$.*

Using the sigmoid function results in the need to estimate one more parameter: σ. Once measurements of i_a and $\dot{\theta}_m$ are made, it is possible to obtain $\ddot{\theta}_m$ by numeric differentiation and the parameters k_t, b, τ_c, and σ can be found by fitting the model to the measurements. To reduce measurement noise, which numeric differentiation is quite sensitive to, the velocity measurements are filtered before the differentiation takes place. Firstly, the ESCON 50-5 drivers that measure the velocity are configured to output not the raw velocity measurements, but what Maxon Motor calls Averaged Speed. This is the velocity measurements, filtered by a first order low-pass filter with a cut-off frequency of 5 Hz [27]. Then, after the measurements are saved to the computer, the velocity signal is filtered using a moving average filter with a window size of 5.

6.2 Estimation Experiment and Results

The parameter estimation test is described in **Section 6.2.2**. The joints are individually actuated, and parameters are estimated one joint at a time. Before the experiments can be conducted, considerations have to be made with regards to the choice of input signals.

6.2.1 Choice of Input Signal

Two types of input signal have been chosen for the test. It is stated in [28] that sine wave inputs and square pulse inputs are good and popular choices. Due to the physical angle limitations on the AXO-arm, described in **Section 2.1**, it is desirable to choose a signal that is symmetric around zero, to make sure that the movement stays within the possible operating region. The recommended type of square pulse train in [28] is one with random period, but constant amplitude. To generate this signal, it is necessary to set a maximum period that does not allow movement all the way from the center of the valid region to the edge of the valid region. This signal period is found experimentally. For the generation of a sine-signal, a frequency sweep is suggested, to evaluate the system response at several input frequencies. Examples of the input signals are shown in **Figure 6.3**.

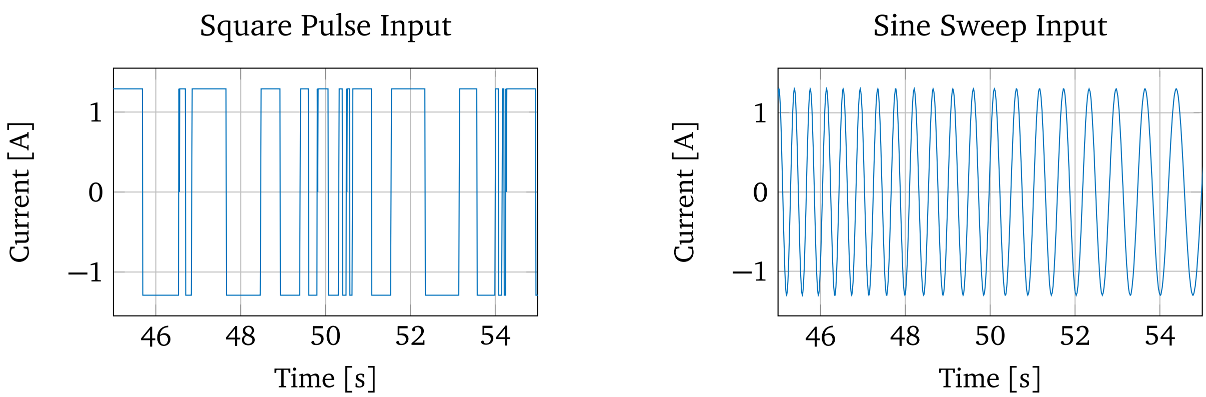


Figure 6.3: *Examples of the two input signals for parameter estimation tests. Both types of signals were* 60 s *long, these plots only show an excerpt.*

The amplitude and period limitations of the input signals are chosen to avoid hitting the mechanical end stops of the joints. The square pulses therefore have a maximum width of 1 s for the shoulder joint and 0.5 s for the elbow joint. To verify the linearity of the system, the test is conducted at three different amplitudes, for both joints. The sine sweeps are chosen to have linearly increasing frequency in range 5 Hz to 0.02 Hz for the shoulder motor and 10 Hz to 1 Hz for the elbow motor. The elbow motor needs higher frequencies because it accelerates faster due to the smaller load, and its operating region is smaller. The sine sweeps are applied at two different amplitudes. A summary of the input signals is shown in **Table 6.1**.

6.2.2 Experiments

A series of experiments have been conducted to obtain measurements that allow finding parameters. The AXO-arm has been disassembled so that parameters can be calculated for one joint at a time. The

	Shoulder		Elbow	
	Frequency Range [Hz]	Amplitudes [A]	Frequency Range [Hz]	Amplitudes [A]
Square Pulse	0 to 0.5	1.1, 1.3, 1.5	0 to 1	0.5,0.6,0.7
Sine Sweep	0.02 to 5	1.3, 1.5	1 to 10	0.6,0.7

Table 6.1: *Signal frequencies and amplitudes of the parameter estimation tests.*

test setup is shown in **Figure 6.4** and is the same for both shoulder- and elbow joint measurements. The PC initiates motion and logs the system response. The vector $\bar{u}$ contains current reference signals and $\bar{y}$ are measurements of joint angle, angular velocity and motor armature current.

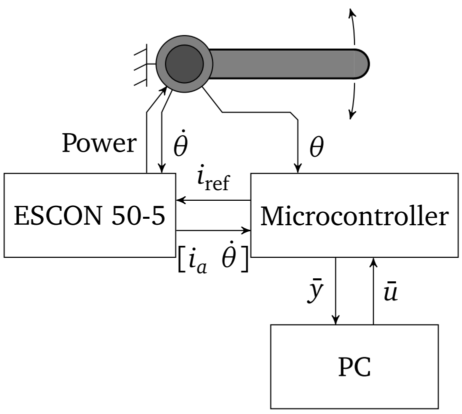


Figure 6.4: *The test setup when estimating parameters for a single joint.*

The two joints and links that make up the AXO-arm are individually mounted on a table. The signals described in the previous section are applied, and measurements of current, angle, and angular velocity are obtained. Each signal is 60 s long.

6.2.3 Friction Parameters

To get the parameters for the friction model,

$$\tau_f = b\,\dot{\theta}_m + \tau_c\,\mathrm{sgm}(\dot{\theta}_m, \sigma), \qquad [\mathrm{N\,m}] \qquad (6.5)$$

the current and velocity is measured when the acceleration is zero. This dataset is extracted from the data obtained by applying the input signals described in **Section 6.2.1**. The measured current and velocity is extracted whenever the acceleration is zero. In this case, the dynamics of the motor is governed by

$$\tau_m = \tau_f \qquad [\mathrm{N\,m}] \qquad (6.6)$$

$$k_t\,i_a(t) = b\,\dot{\theta}_m(t) + \tau_c\,\mathrm{sgm}(\dot{\theta}_m(t), \sigma) \qquad [\mathrm{N\,m}] \qquad (6.7)$$

For this part of the parameter estimation, we choose to not find k_t, but instead find the parameters: $\frac{b}{k_t}$, $\frac{\tau_c}{k_t}$, and σ, because in this section, we only seek to parameterize the friction model. It is also not possible to find k_t, because **Equation (6.7)** is over-parameterized. The MATLAB function `lsqcurvefit()` is used to find the parameters that best fit the model to the measured data of i_a and $\dot{\theta}_m$. The pseudo-inverse could also have been used to find the least squares solution to the curvefitting problem, however, `lsqcurvefit()` allows constraints on the parameters. This lets us make sure that they are positive, and thus keep their physical meaning.

The function `lsqcurvefit()` is a least squares data-fitting function. When supplied with a model, $F(\bar{u}, \text{par})$, and a measured input-output sequence, $(\bar{u}, \bar{y})$, and an initial guess of the parameters, par, it will find the parameters that minimize the Mean Squared Error (MSE):

$$E(\text{par}) = \frac{1}{N} \sum^{N} (F(\bar{u}, \text{par}) - \bar{y})^2 \tag{6.8}$$

The default solver options are used.

Figure 6.5 shows how the friction model compares to the measurements on both motors. The uneven sample density comes from the fact that the points are an extract of a larger set of measurements. Recall that we only use the current and velocity at times when the acceleration is zero. This method is used to get as many samples as possible. Simply measuring the current at different constant velocities, is a problem because the joint movement of the AXO-arm is limited, so the motor may not reach a constant velocity before hitting a mechanical end-stop. **Table** 6.2 shows the parameters that describe the friction model.

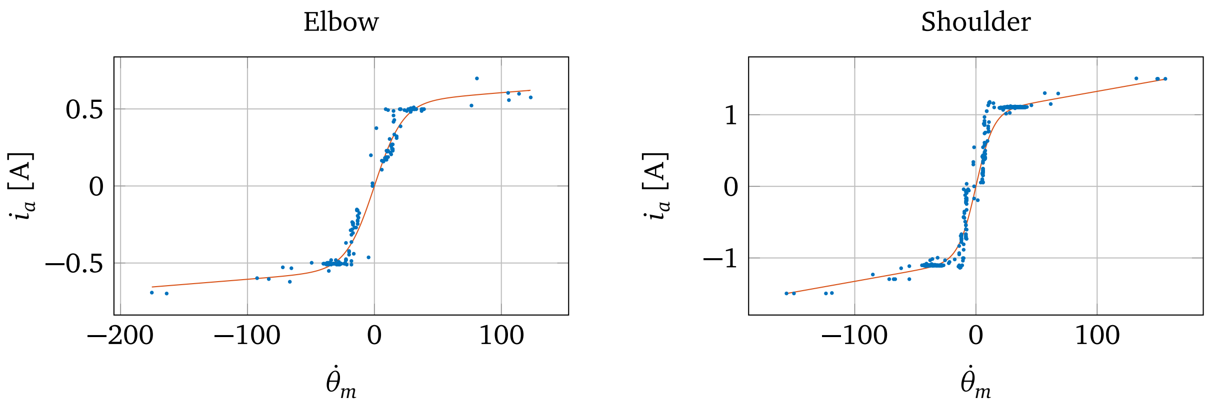


Figure 6.5: *Relationship between armature current and velocity, when the acceleration is zero. This data is used to fit a friction model consisting of coloumb friction and viscous friction.*

	$\frac{b}{k_t}\left[\frac{\text{A}}{\text{rad/s}}\right]$	$\frac{\tau_c}{k_t}$[A]	σ [·]
Shoulder	0.0030	1.0245	0.1383
Elbow	0.0007	0.5394	0.0889

Table 6.2: *Friction parameters for the two joints on the AXO-arm.*

6.2.4 Motor Constant

Having derived a suitable model of the friction in the joints, this section describes how the motor torque constant, k_t, is estimated. In the previous section, we found a model for the current consumed by friction, this is denoted $i_f(\dot{\theta}_m)$. The model is now reduced to

$$I_m\, \ddot{\theta}_m(t) = \tau_m(t) - \tau_f(\dot{\theta}_m) \qquad [\text{N m}] \tag{6.9}$$

$$\tau_m(t) = k_t\, i_a(t) \qquad [\text{N m}] \tag{6.10}$$

$$\tau_f(t) = k_t\, i_f(\dot{\theta}_m) \qquad [\text{N m}] \tag{6.11}$$

Since i_f is known, it is straightforward to fit the model to the data, to obtain the motor torque constant, k_t. Once again, the MATLAB function `lsqcurvefit()` is used to fit the model to the measurements. The fitness between the model and the data, measured as NMSE, is $E_s = 0.901$ and $E_e = 0.804$ for the shoulder- and elbow joint respectively. Changing the model slightly, to allow a different torque constant for $i_f(\dot{\theta}_m)$ and $i_a(t)$ improves the model substantially. The model is now changed into

$$I_m \ddot{\theta}_m(t) = k_t\, i_a(t) - k_f\, i_f(\dot{\theta}_m) \qquad [\mathrm{N\,m}] \qquad (6.12)$$

This method is unsupported by literature but it merely amounts to a scaling of the friction model, to also fit it to the data where the acceleration is nonzero. The NMSE is now $E_s = 0.971$ and $E_e = 0.927$, which is a significant enough improvement that the modification is accepted.

Table 6.3 shows the parameters. Interestingly, the motor torque constants specified in the data sheets of the motors are 0.114 N m/A and 0.131 N m/A, so the estimated values are reasonable. The

	k_t [N m/A]	$b\left[\frac{\mathrm{N\,m}}{\mathrm{rad/s}}\right]$	τ_c [N m]	σ [·]
Shoulder	0.0708	194.69×10^{-6}	0.066	0.1383
Elbow	0.0764	44.05×10^{-6}	0.0359	0.0889

Table 6.3: *Motor and friction parameters for the two joints on the AXO-arm.*

data used for these parameter estimations is 5 min long. It is thus infeasible to show all of it here. However, we have chosen a few excerpts where we compare simulated and measured data. This is shown for the elbow joint in **Figure 6.6** and the shoulder joint in **Figure 6.7**. On both of the figures, the two top graphs is with a random period square pulse train and the two bottom graphs are with a sine sweep input. The motion of the tests correspond to realistic human movement. The velocities are in the range −100 rad/s to 100 rad/s on the input side of the gears, resulting in an actual joint velocities of −2 rad/s to 2 rad/s.

On the plots in **Figure 6.6c** and **Figure 6.7c** something strange is going on. Some of the periods have a much smaller amplitude than the others. It is not clear what creates this phenomenon, but maybe the friction is not completely time invariant and operation in open loop is therefore not completely predictable. This phenomenon is not investigated further.

Overall, though, **Figure 6.6** and **Figure 6.7** show that the model fits generally well with the data, and the estimation error is rarely above 10 rad/s to 15 rad/s, which on the output of the gear corresponds to an actual joint velocity error of 0.2 rad/s to 0.3 rad/s. The data shown in **Figure 6.6** and **Figure 6.7** is a comparison between the model and the data that the parameters have been fitted to. In the following section, a validation using new data is made.

6.3 Parameter Validation

Before the model of the complete AXO-arm is validated, this section seeks to validate the parameters of the motor model. This is done by comparing simulations with measurements, but using a data set that has not been used for parameter fitting. Extracts from these data sets are shown in **Figure 6.8**. The model still fits the measurements well. The validation gives a NMSE fitness between the measurements and the simulation of $E_e = 0.968$ and $E_s = 0.953$ for the elbow- and shoulder joint respectively, thus meeting the requirement of an NMSE of at least 0.8 on the mechanical model. The motor model parameters are therefore accepted for use in the following chapter that validates the model of the entire AXO-arm.

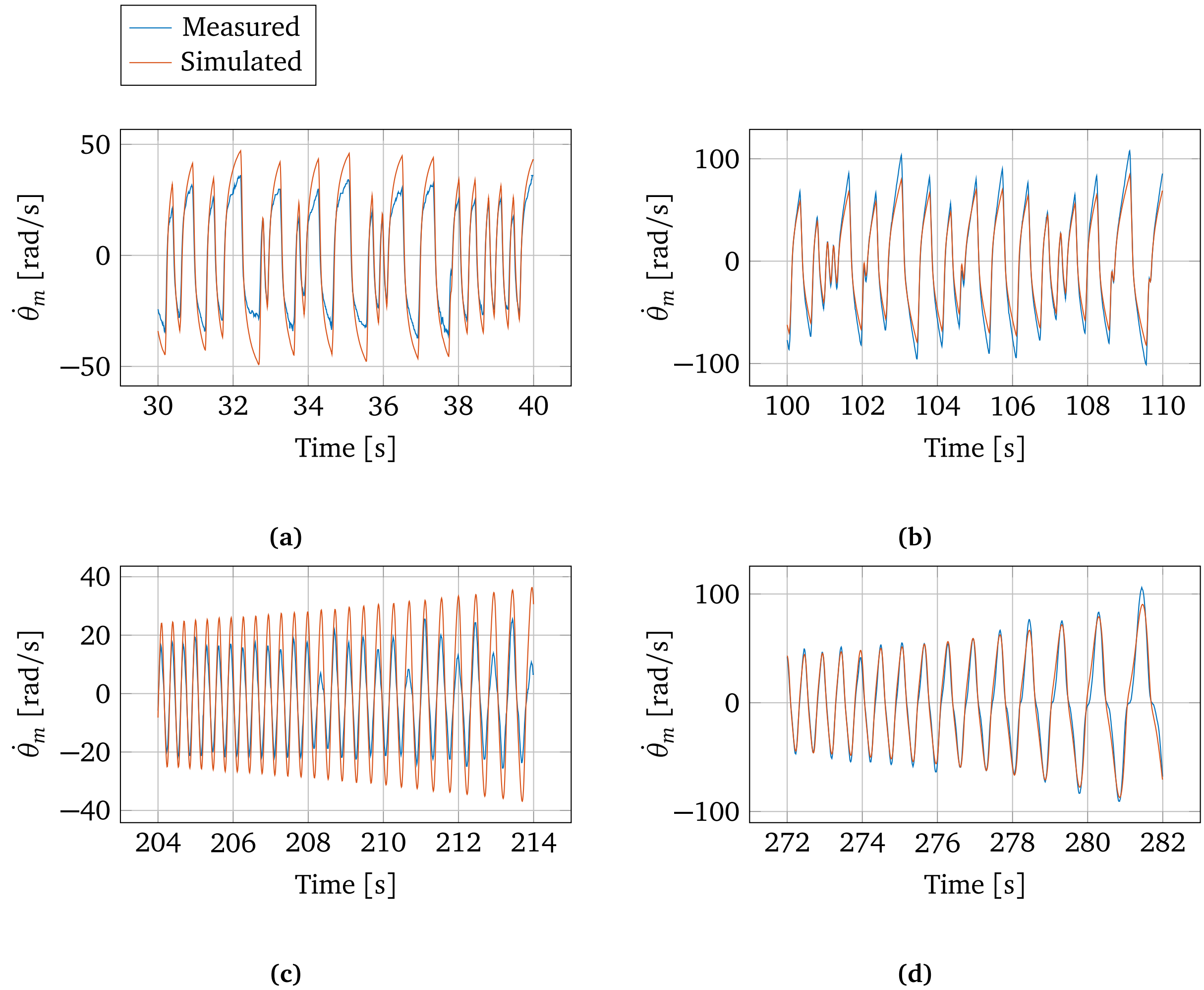
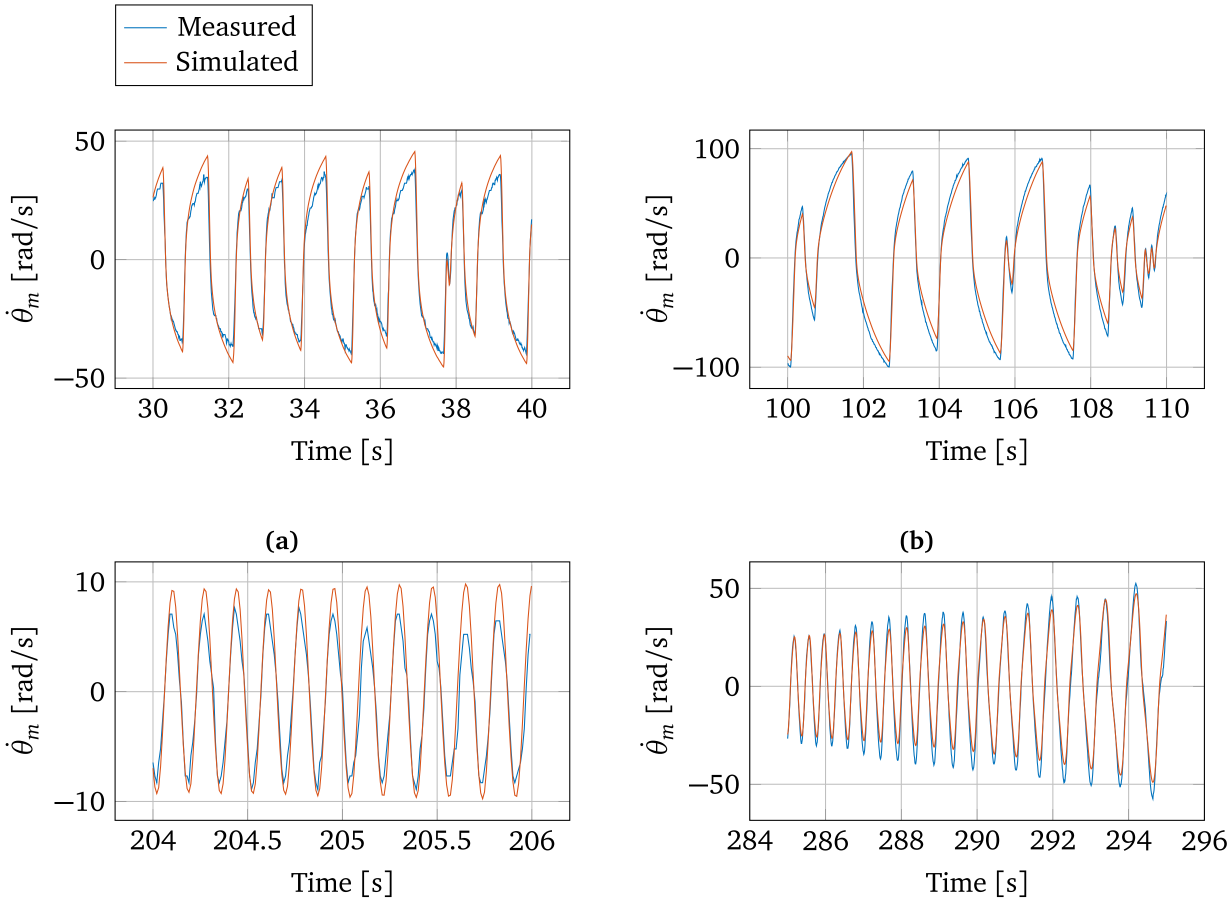


Figure 6.6: *Comparison of simulated and measured velocity of the elbow motor. Blue is measurement, red is simulation.*



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

Figure 6.7: *Comparison of simulated and measured velocity of the shoulder motor. Blue is measurement, red is simulation.*

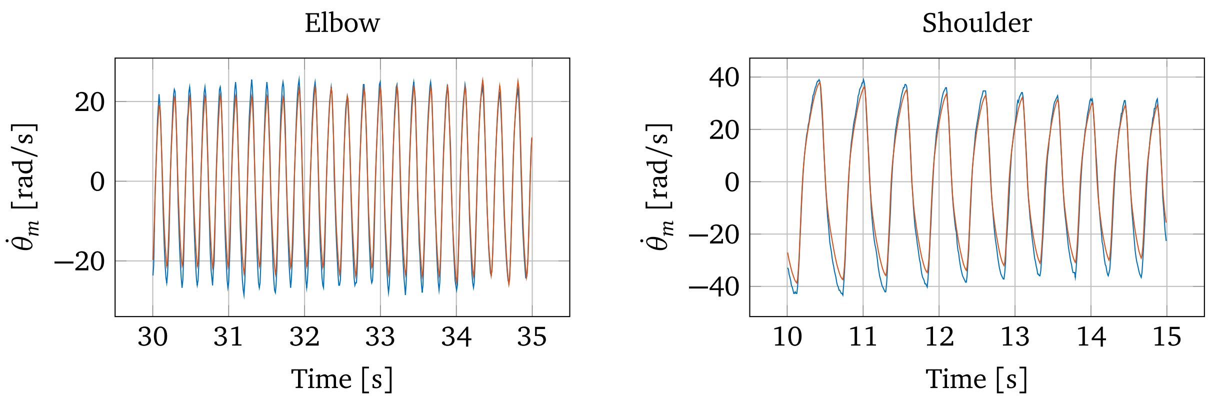


Figure 6.8: *Comparison of measurements and simulations using a data-set that has not been used for parameter fitting.*

Verification of the Dynamic Model 7

In **Chapter 6**, actuator parameters are found that match the model when the motors are modeled individually. To check that the found parameters also fit well with the full mechanical model found in **Chapter 5**, this chapter aims to verify the model by comparing real measurements with model simulations obtained with the found parameters. In **Section 7.1** and **Section 7.2** the measurements and simulations are explained. In **Section 7.3** the aim is to verify the model by comparing measurements and simulations. Lastly **Section 7.4** introduces final improvements of the model by small parameter adjustments.

7.1 Measurement of Exoskeleton Planar Movement

In **Chapter 6**, the AXO-arm was disassembled to remove as much of the dynamics as possible. For the model verification in this chapter, it is put back together. The AXO-arm is attached to a platform in such a way, that the joints' coordinate systems match the arrangement given in **Chapter 4**. Both joints rotate around horizontal axes and the serial manipulator therefore moves around in a vertical planar space. The AXO-arm attached to the platform is shown in **Figure 7.1**.

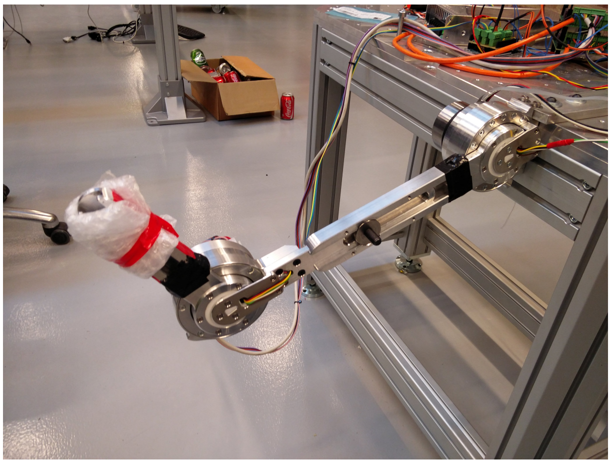


Figure 7.1: *The serial manipulator is tightly attached to the platform to avoid interfering oscillations when measuring. The forearm link is bubble-wrapped to avoid damaging it, if it hits the table.*

With the mechanical setup in place, the electrical setup is shown in **Figure 7.2**. By using this setup, the motors can be actuated simultaneously, while measuring current, angle, and angular velocity of each motor. The inputs from the PC are given by the vector $\bar{u}$, containing current reference signals for each motor. The micro controller then applies the current reference signals, i_{ref} to the ESCON drivers, and measures the actual applied current, i_a, the velocity, $\dot{\theta}$, and the position, θ, in each joint. These measurements are then returned to the computer in the vector $\bar{y}$. The inputs and outputs with a subscript (1) belongs to the shoulder motor and driver, and those with (2) the elbow motor and driver. Since the model parameters are known, the measured current can be converted to torque, $\tau = i_a \cdot k_t$, for use in the mechanical model simulations.

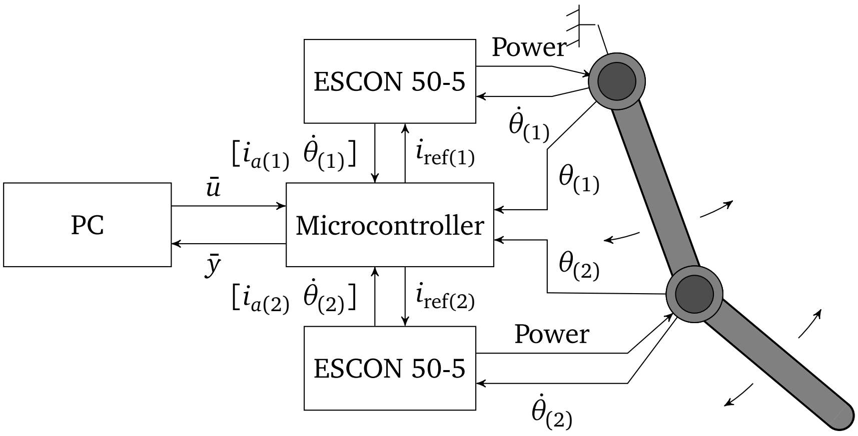


Figure 7.2: *Schematic showing the configuration between PC, micro controller, ESCON drivers and serial manipulator.*

7.1.1 Choice of Input

As concluded in **Chapter 6**, sine wave and square pulse inputs are good choices for inputs. These types of signals are therefore used again, but in simpler forms because of the mechanical limitations of the AXO-arm. It is observed that when applying a square pulse or sinusoidal input, the arm tends to drift in joint positions towards the mechanical end stops, since gravity is now effecting the setup and because the serial manipulator is actuated in open loop with no control of position. To get an acceptable measurement, drifting is almost avoided by choosing constant signal frequencies and difference in amplitude depending on direction, which unfortunately neglects sine sweep and random period square pulse signals as options. Constant period signals are therefore chosen for both the square pulse and sinusoidal input and their period is chosen appropriately (found by experiments) to avoid the mechanical end stops. The chosen inputs are described in **Table 7.1**, and shown in **Figure 7.3**. The sampling period is 10 ms for both inputs and outputs.

	Shoulder		Elbow	
	Frequency [Hz]	Amplitude [Nm]	Frequency [Hz]	Amplitude [Nm]
Square Pulse	0.5	-0.8 to 6	1	-1.9 to 2.1
Sinusoid	0.25	-2.3 to 6	1	-2.4 to 2.6

Table 7.1: *Shape and amplitude of the two different input signals to the joint actuators of the AXO-arm.*

7.2 Simulation of Exoskeleton

With a description of how to obtain the verification measurements, the simulation part of the verification is described in this section. To be able to verify the model by comparing measurements and simulations, the simulations in this section have been made with the same inputs as in **Section 7.1.1**. To be able to simulate the joint angles and angular velocities of the exoskeleton, the model derived in **Chapter 5 Equation (5.42)** is used and written here again for clarity:

$$\ddot{\bar{\theta}} = (\boldsymbol{I}_m \boldsymbol{N}^2 + \boldsymbol{M}(\bar{\theta}))^{-1}(\bar{\tau}_m \boldsymbol{N} - \bar{V}(\bar{\theta}, \dot{\bar{\theta}}) - \bar{G}(\bar{\theta}) - \bar{F}(\dot{\bar{\theta}}\, \boldsymbol{N})\, \boldsymbol{N}) \quad \left[\text{rad/s}^2\right] \tag{7.1}$$

$$= f(\bar{\theta}, \dot{\bar{\theta}}, \bar{\tau}_m) \quad \left[\text{rad/s}^2\right] \tag{7.2}$$

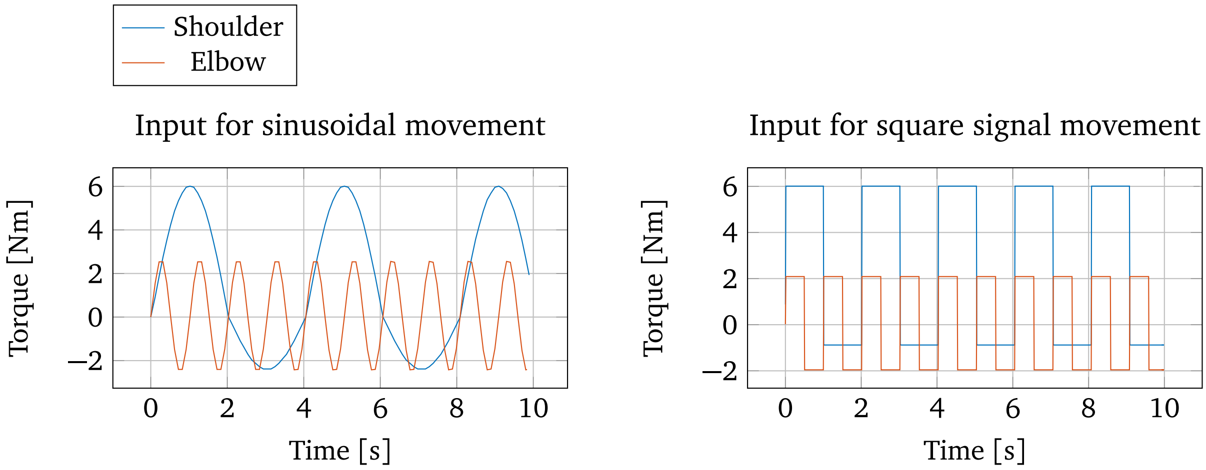


Figure 7.3: *This figure shows the two inputs for the measurements. The blue curve is motor 1(shoulder) and the red curve is motor 2(elbow). The current reference signals are converted to torque using the torque constants.*

The inertia tensor matrices of the two links, ${}^{1}\boldsymbol{I}_{C_1}$ and ${}^{2}\boldsymbol{I}_{C_2}$, that are part of $\boldsymbol{M}$, are below:

$$ {}^{1}\boldsymbol{I}_{C_1} = \begin{bmatrix} 1.237\times10^{-3} & -5.512\times10^{-6} & 2.358\times10^{-3} \\ -5.512\times10^{-6} & 2.105\times10^{-2} & 5.844\times10^{-7} \\ 2.358\times10^{-3} & 5.844\times10^{-7} & 2.090\times10^{-2} \end{bmatrix} \quad \left[\mathrm{kg/m^2}\right] \tag{7.3} $$

$$ {}^{2}\boldsymbol{I}_{C_2} = \begin{bmatrix} 8.178\times10^{-5} & 0.001\times10^{-9} & -2.650\times10^{-5} \\ 0.001\times10^{-9} & 1.069\times10^{-3} & 0 \\ -2.650\times10^{-5} & 0 & 1.118\times10^{-3} \end{bmatrix} \quad \left[\mathrm{kg/m^2}\right] \tag{7.4} $$

Since the links of the serial manipulator are only rotating around the joints' z axes the model only uses entry 3,3 (also written as the I_{zz} element of ${}^{i}\boldsymbol{I}_{C_i}$). These values will not change since they are found from CAD drawings that are assumed to be sufficiently precise in the real setup.
Regarding the actuator-inertias $\boldsymbol{I}_m$ these are found partly from CAD drawings and the data sheets [29] and [30], since they consists of both the motor inertia and a part of the gearbox. The $\boldsymbol{I}_m$ values are stated below:

$$ \boldsymbol{I}_m = \begin{bmatrix} 1.492\times10^{-4} & 0 \\ 0 & 0.463\times10^{-4} \end{bmatrix} \quad \left[\mathrm{kg/m^2}\right] \tag{7.5} $$

Where entry 1,1 describes the inertia in the shoulder, and entry 2,2 the elbow. With the inertias stated the parameter list is expanded as shown in **Table 7.2**. In **Chapter 6** the model is parameterized using the velocities on the motor axle, but in this chapter we use the joint velocity on the output of the gears. The viscous friction in **Table 7.2** is therefore scaled using the gear ratio, $N = 50$.

	k_t [N m/A]	b $\left[\frac{\mathrm{N\,m}}{\mathrm{rad/s}}\right]$	τ_c [N m]	σ [·]	I_{zz} [kg m^2]	I_m [kg m^2]
Shoulder	0.0708	9.73×10^{-3}	0.066	0.1383	2.090×10^{-2}	1.492×10^{-4}
Elbow	0.0764	2.20×10^{-3}	0.0359	0.0889	1.118×10^{-3}	0.463×10^{-4}

Table 7.2: *Motor, friction and inertia parameters for the two joints on the AXO-arm.*

For simulation purposes the model is rewritten into State Space form:

$$\begin{bmatrix} \dot{\bar{\theta}} \\ \ddot{\bar{\theta}} \end{bmatrix} = \begin{bmatrix} \dot{\bar{\theta}} \\ f(\bar{\theta}, \dot{\bar{\theta}}, \bar{\tau}_m) \end{bmatrix} \tag{7.6}$$

To be able to simulate with input sequences that are identical to the measurements, it is chosen to simulate using the Forward Euler method which is advanced according to:

$$\begin{bmatrix} \bar{\theta}(k+1) \\ \dot{\bar{\theta}}(k+1) \end{bmatrix} = \begin{bmatrix} \bar{\theta}(k) \\ \dot{\bar{\theta}}(k) \end{bmatrix} + T_s \begin{bmatrix} \dot{\bar{\theta}}(k) \\ f(\bar{\theta}(k), \dot{\bar{\theta}}(k), \bar{\tau}_m(k)) \end{bmatrix} \tag{7.7}$$

There T_s is the sampling time (10 ms) which is equal to the sampling time in **Section** 7.1.1.

7.3 Comparison of Measurement and Simulation Results

This section shows the comparison between measurements and simulation. In **Figure** 7.4 the angles and angular velocities of the motors, when applying sinusoids, are shown. The red curves represent simulations, while the blue curves are measurements. As shown, the frequencies of simulations and measurements are similar, while the amplitudes are not. In the same manner, the results when applying square pulse signals are shown in **Figure** 7.5. Here the observations are the same, namely that the frequencies fit but the amplitudes do not. Another observation is that the joint angle measurements and simulations tend to drift apart over time. This is not surprising since small model uncertainties turn into large deviations when they undergo the double integration that happens between the input signal and the joint position. The performance of the model will thus be evaluated by comparing measured and simulated velocity. The quality of the fit is evaluated using MATLAB's implementation of NMSE.

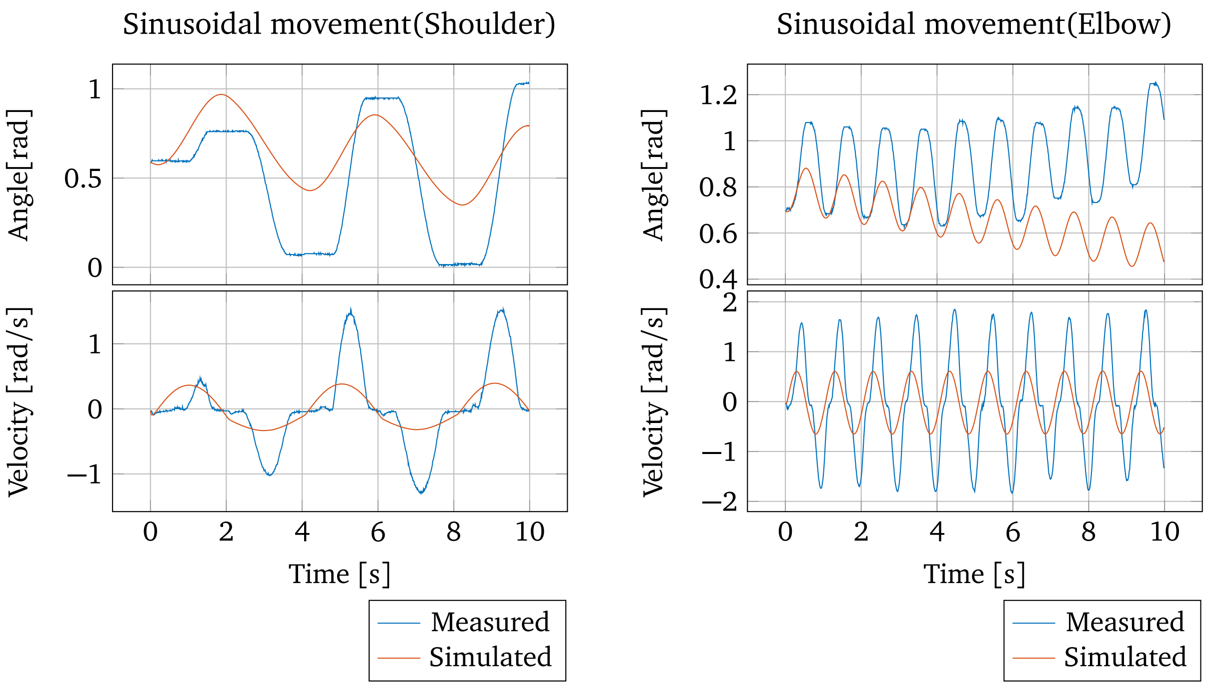


Figure 7.4: *Angles and angular velocities of the motors, when applying the sinusoids described in* ***Section*** *7.1.1.*

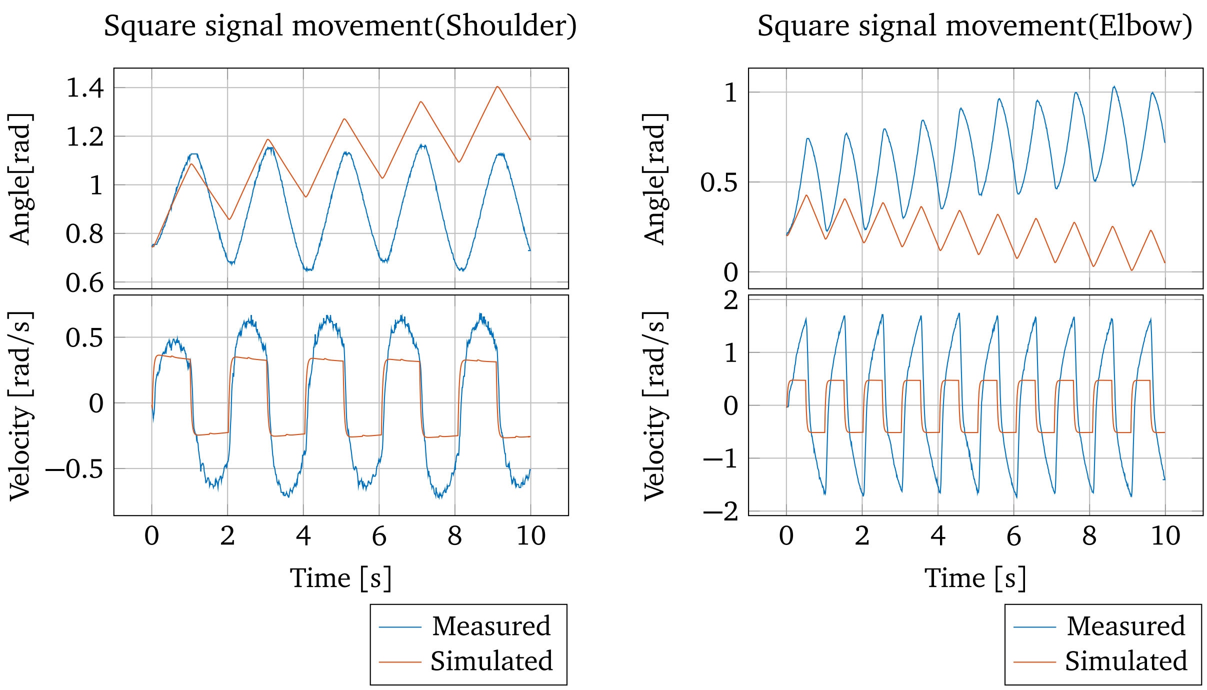


Figure 7.5: *Angles and angular velocities of the motors, when applying the square pulse signals described in* ***Section*** *7.1.1.*

The NMSE for the velocity data presented in **Figure** 7.4 and **Figure** 7.5 is $E = 0.533$. This statistic is not satisfactory, so an additional parameter fitting of the motor model is performed in the following section.

7.4 Improvements of the Mechanical Model

The comparison of the simulation and measurements make it clear that the model needs to be improved. In **Chapter** 6 the AXO-arm was disassembled and the joints were mounted with a vertical axis of rotation. This could result in a different friction torque, than when the gear is mounted with a horizontal axis of rotation since this loads the gear differently. It is therefore attempted to recalculate the parameters of the friction model. Several combinations of friction parameter updates were attempted, but the lowest NMSE was obtained by adjusting the Coulomb friction parameters, τ_c and σ. The NMSE is now improved to $E = 0.845$, which meets the model precision requirement from **Chapter** 3.

Like in **Chapter** 6 we use the MATLAB function `lsqcurvefit()` for parameter estimation. A comparison of the updated model output and measurements are shown in **Figure** 7.6 and **Figure** 7.7. The amplitudes of the measurements and simulations are now much more similar, yet joint angles are still drifting. It also seems that some of the Coulomb friction dynamics is not correctly captured by the model. However, the monotony of the model is correct, and much of the dynamic behavior is captured well.

Using the verification data for parameter fitting means that a different validation dataset needs to be obtained to avoid over-fitting. Therefore, a final validation of the model is made by comparing the response of the model, to measurements that have not been used for parameter fitting. This new data-set is made with an input-signal that is similar to what we have previously used. A plot comparing the model and the measurements are shown in **Figure** 7.8, and show that the model is performing just as well as with the data that it is fitted after. This is also backed by the NMSE,

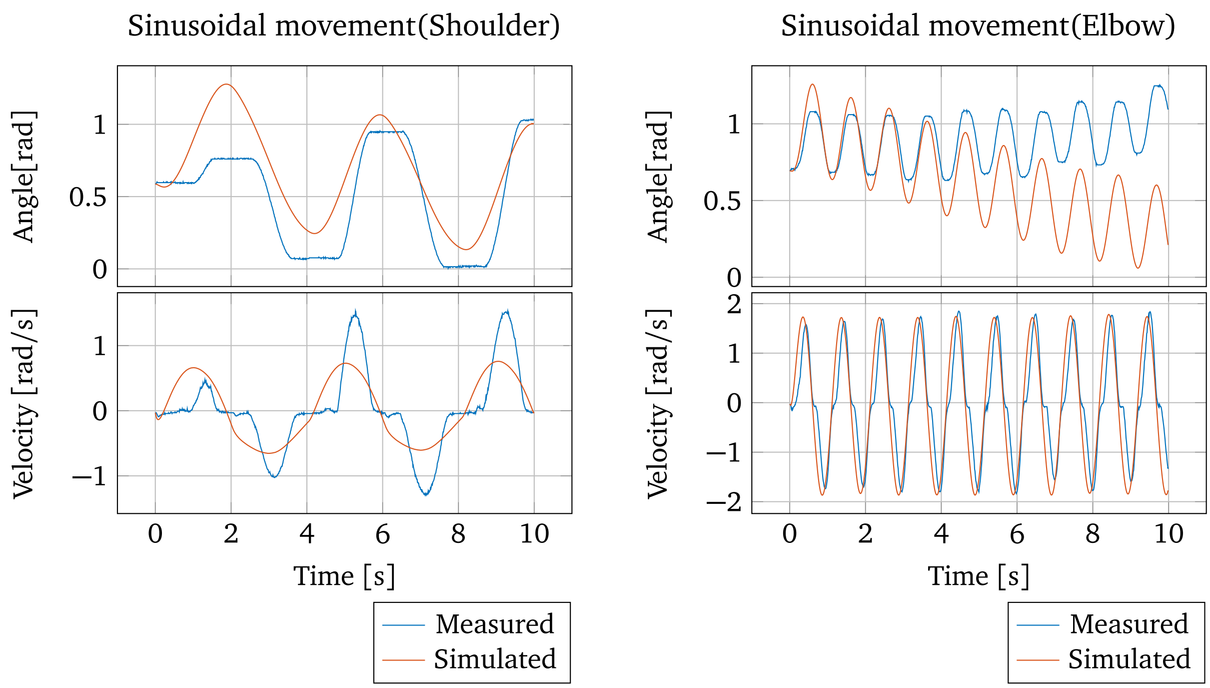


Figure 7.6: *Angles and angular velocities of the motors when the parameters τ_c and σ are adjusted. The input for both simulation and measurement are given in 7.3*

$E = 0.826$. The final parameters for the mechanical model are shown in **Table 7.3**. While there is significant drifting in the position modeling, the model should still be sufficient when used as part of a feedback loop and it is therefore verified.

	k_t [N m/A]	$b\left[\frac{\text{N m}}{\text{rad/s}}\right]$	τ_c [N m]	σ [·]	I_{zz} [kg m^2]	I_m [kg m^2]
Shoulder	0.0708	9.73×10^{-3}	4.82	0.0475	2.090×10^{-2}	1.492×10^{-4}
Elbow	0.0764	2.20×10^{-3}	0.36	0.1299	1.118×10^{-3}	0.463×10^{-4}

Table 7.3: *Final motor, friction and inertia parameters for the two joints on the AXO-arm.*

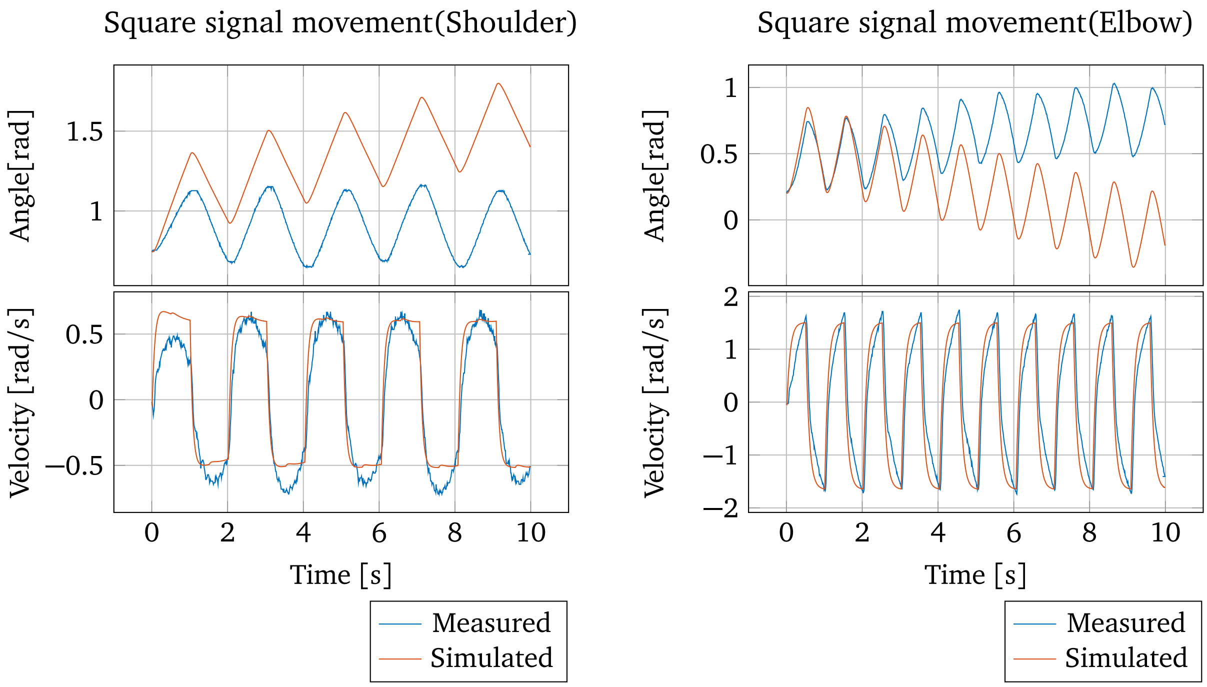


Figure 7.7: *Angles and angular velocities of the motors when the parameters τ_c and σ are adjusted. The input for both simulation and measurement are given in 7.3*

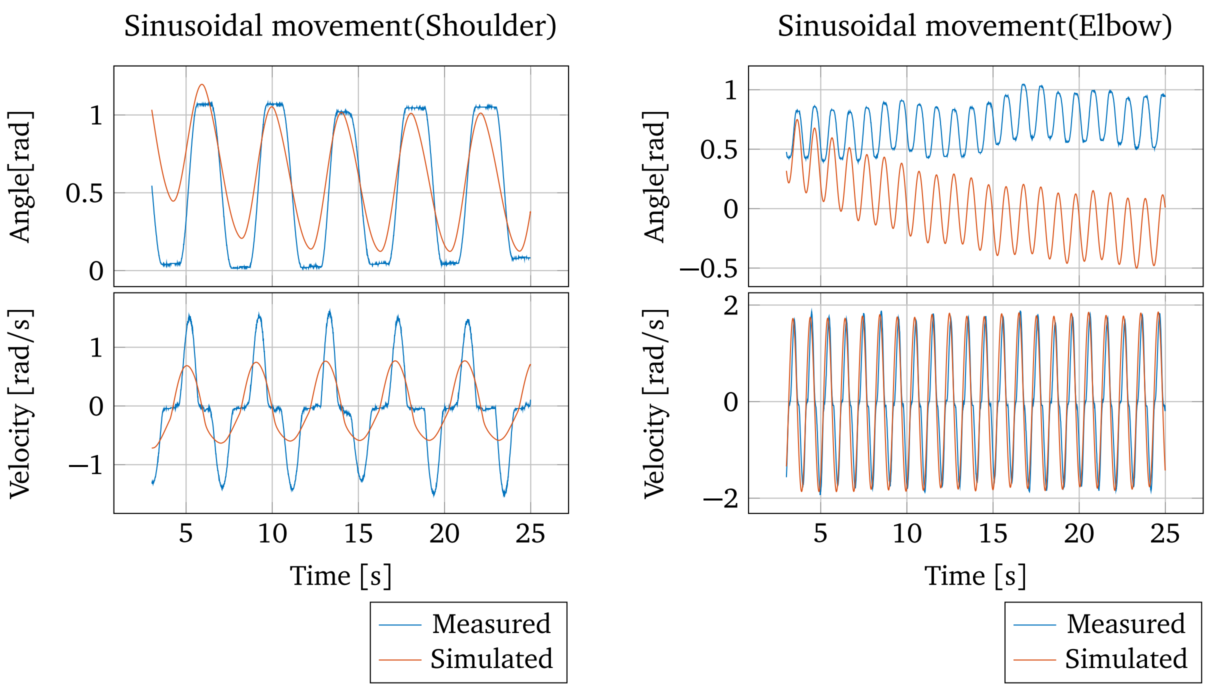


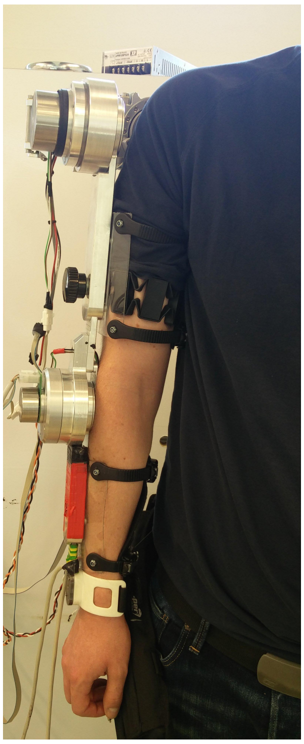
Figure 7.8: *Comparison between measurements and simulations. The measurements have not been used for parameter estimation.*

Human-Exoskeleton Interaction 8

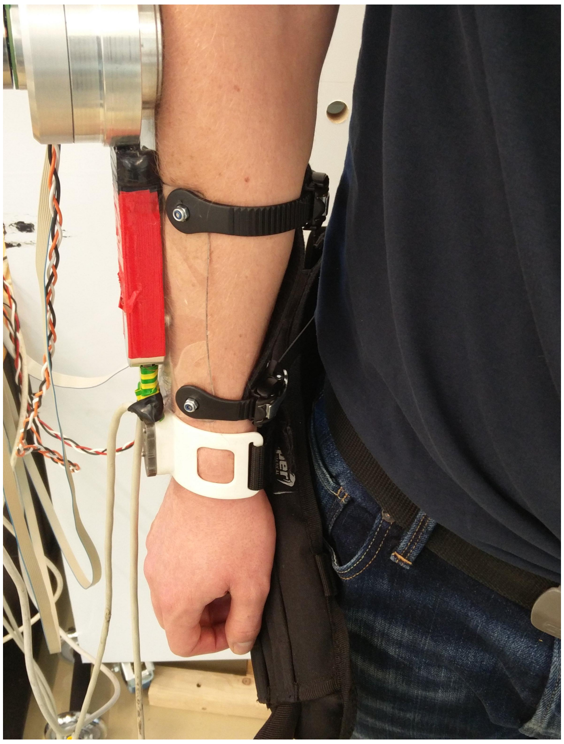
An important and difficult part of designing an exoskeleton lies in the human-robot interaction. This project will not develop any mechanical solution to this problem, however, it will contain a description of the solution that is currently in use at AAU and it will also describe how the attachment of a human arm will influence the dynamics of the AXO-arm.

8.1 The Mechanical Connection Between the AXO-arm and its User

The AXO-arm is not yet mobile. That means that the shoulder joint is rigidly attached to a mechanical reference. At the time of writing, that reference is a wall. The AXO-arm has a two links that correspond to a forearm and an upper arm. Using a plastic shell with plastic tighteners, the AXO-arm can be attached to a user. The mount is shown attached to a user in **Figure 8.1**. It is made of $\approx 3\,\text{mm}$ plastic, so we will assume a rigid connection between the user and the AXO-arm. We will thus treat any force that is exerted by the user as an external force applied to the AXO-arm.



(a)



(b)

Figure 8.1: *The AXO-arm mounted on a user.*

8.2 Modeling the Effect of the Arm on the System Dynamics

When modeling the human arm in interaction with the AXO-arm, two different approaches are immediately apparent. The first assumes that the exoskeleton should control the motion of the user, with no effort from the user. Such a control would imply that the AXO-arm can find the exact desired trajectory, without the user moving, and then the AXO-arm moves the user. This would be the case if the exoskeleton was to be used by a paralyzed person, and would require the inertial, gravitational, and frictional terms of the human arm to be included in the dynamic model.

The other approach recognizes the feedback control that a human is exercising on its limbs. In this scenario, the user is capable of moving his arms by himself, and during regular motion, the AXO-arm is following the user more than it is carrying him. Modeling the human arm is thus not as crucial to the operation, since the dynamics are compensated for by the user. It might still prove useful to estimate the weight of a potential potential payload that the user is carrying.

The purpose of the AXO-arm is assisting weak muscled users during light weight tasks. The latter approach to modeling the human is therefore chosen, where the user is assumed to compensate for most of his own weight, and the electromechanics are thus modeled alone. The present dynamic model is therefore not expanded to include the dynamics of the human arm.

Part II

Muscle Modeling

This part of the project presents a way to describe the motion of an arm, from EMG signals. It relies on modeling the force that a muscle exerts and the connection between the muscle and the joints. After describing the model, we parameterize it and verify it experimentally.

Muscle Model 9

In this chapter the biomechanics of a general human skeletal muscle is studied. The aim is a model that estimates the torque produced in a joint by some muscle, given a joint angle and a measured EMG signal. The muscle model will be used as part of the AXO-arm control loop. The user's estimated joint torque will be used to generate references trajectories for the AXO-arm.

In this chapter the model is general and is therefore a basis for the next chapter, which aims to identify relevant parameters to make a specific model for each muscle of relevance in the current setup. The general muscle model that needs to be constructed is shown in **Figure 9.1** and it is based on the works of [6]. Two parts need to be studied, namely muscle force and muscle shape. Each section will have a summary of the unknown parameters that need to be estimated to make the model work and make it muscle specific. Furthermore this chapter also includes an explanation of which specific arm muscles are of interest for the current setup.

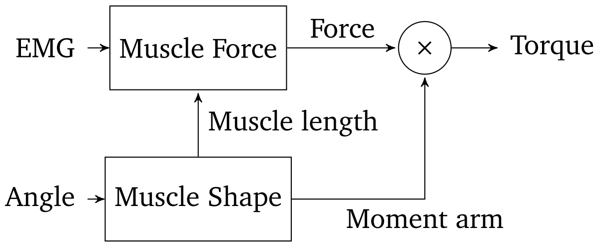


Figure 9.1: *General Muscle Model. [6]*

The contents of this chapter are therefore:

- General description of skeletal muscles.
- Modeling of muscle force and emphasizing of relevant parameters.
- Modeling of muscle shape and emphasizing of relevant parameters.
- Choice of muscles, relevant for the setup.

9.1 Biomechanics of a Skeletal Muscle

Skeletal muscles refer to the type of muscle that is responsible for human movement. The ends of the muscle are attached to bones, allowing the muscles to manipulate the human body. Movement is created when a muscle contracts, thereby manipulating the bones that it is attached to, resulting in a rotation of the adjacent joints. Functionally, the muscle consists of two parts: the muscle belly and the tendon that binds it to the bone. [31]

Skeletal muscles are divided into two groups: parallel and pennate muscles. The difference between the two is the structure of the muscle fibers and the tendons. In a parallel muscle, the fibers run mostly parallel, and connect to a tendon in each end. A muscle fiber can shorten roughly 50 % to 60 %, so the parallel muscles are capable of significant shortening, compared to pennate muscles. The fibers of parallel muscles can be up to 90 % of the total muscle length. [31]

A pennate muscle consist of several tendons that extend along the length of the muscle. The muscle fibers that attach to the tendons are thus shorter and pull in several different directions. Due to the fibers of pennate muscles being shorter, this type of muscle has a smaller range of motion than a parallel muscle. [31]

Muscles are capable of activating the skeletal joints by contracting. When the muscle-belly contracts a pull is created on the bones, resulting in a torque in adjacent joints. The distance between the attachment of the muscle to the joint determines the range of motion that the muscle controls and the torque it is able to provide. The approach that is chosen for modeling muscles in this project is based on [6], and divides the model into two parts, as indicated on **Figure 9.1**. One part that seeks to model the force that a muscle exerts, and one part that models the shape of the muscle and how it interacts with the joints.

9.2 Modeling Muscle Force

This section describes the muscle force part of **Figure 9.1**. To model the force generated by the muscles, the Hill Muscle Model is used. This type of model describes the force generated by the muscles, based on the length (i.e contraction) of the muscle, and the contractile velocity [31]. It takes into account the viscoelastic properties of the muscle-fiber and the tendons.

The Hill model consists of three parts, Contractile Element (CE), and two elastic elements, Serial Element (SE) and Parallel Element (PE). The combination of CE and PE makes up the dynamics of the muscle belly, while SE accounts for the dynamics of the tendons at each end of the muscle. The force generated in the muscle can be expressed as:

$$F = F_{SE} \quad [\mathrm{N}] \quad (9.1)$$

$$= F_{CE} + F_{PE} \quad [\mathrm{N}] \quad (9.2)$$

where F_{CE}, F_{PE} and F_{SE} are the force generated in each element, respectively.

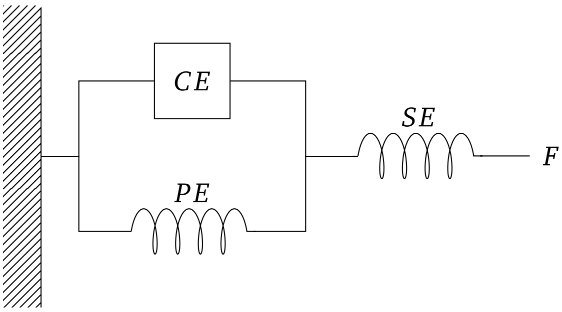


Figure 9.2: *Hill's Muscle Model*

The following subsections describe a method for modeling these three elements.

9.2.1 Model of the Contractile Element

The force exerted by CE is the active generation of force in the muscle. It is modeled as a fraction of the maximal contractile force, $F_{CE_{max}}$, as proposed in [6]. Three factors determine the size of the fraction: the normalized force-length relationship, $f(\Delta l_{CE})$, the normalized force-velocity relationship, $g(v_{CE})$, and the muscle activation signal, $a(t)$.

$$F_{CE} = a(t)\, f(\Delta l_{CE})\, g(v_{CE})\, F_{CE_{max}} \quad [\mathrm{N}] \quad (9.3)$$

Force-length Relationship

The force-length relationship is a positive function that has a maximum in some optimal muscle fiber length. In [6] it is suggested to approximate it as the Gaussian function in **Equation (9.4)**.

$$f(\Delta l_{CE}) = \exp\left(-\frac{1}{2}\left(\frac{\frac{\Delta l_{CE}}{L_{CE_0}} - \phi_m}{\phi_v}\right)^2\right) \qquad [\mathrm{N}] \qquad (9.4)$$

$$\Delta l_{CE} = l_{CE} - L_{f_s} \qquad [\mathrm{mm}] \qquad (9.5)$$

The force-length relationship thus becomes a function the of length of the muscle fibers, Δl_{CE}, the slack length of the fibers, L_{f_s}, and the optimal length of the muscle fibers, L_{CE_0}. The parameters ϕ_m and ϕ_v have no physiological interpretation, they are simply tuning parameters to fit the mean and variance of the Gaussian. An example of $f(\Delta l_{CE})$ is shown in **Figure 9.3a**.

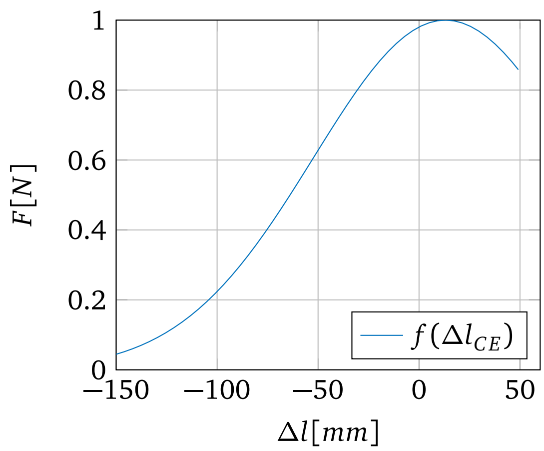
Force-velocity Relationship

The force a muscle can exert decreases with the contraction velocity. The force-velocity relationship is therefore a decreasing function when the magnitude of the velocity grows. The active movement that a muscle can do is to contract, so the velocity of the contractile element is always negative. It is approximated by **Equation (9.6)** which is a normalized function that is minimal when the contractile velocity, v_{CE}, grows in negative direction. [6]

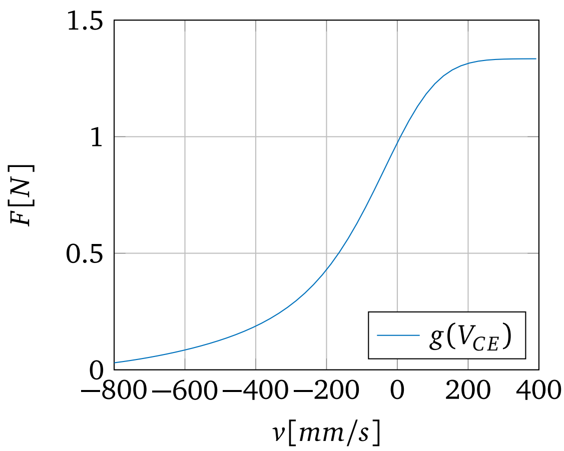
$$g(v_{CE}) = \frac{0.1433}{0.1074 + \exp\left(-1.3\sinh\left(2.8\,\frac{v_{CE}}{V_{CE_0}} + 1.64\right)\right)} \qquad [\mathrm{N}] \qquad (9.6)$$

$$V_{CE_0} = \frac{1}{2}(a(t)+1)\left(2\,L_{CE_0} + 8\,L_{CE_0}\,\alpha\right) \qquad (9.7)$$

The force-velocity relationship depends on the contraction velocity, v_{CE}, the maximal contraction velocity, V_{CE_0}, the optimal length of the muscle fibers, L_{CE_0}, the percentage of fast fibers in the muscle, α, and the activation of the muscle, $a(t)$. An example of $g(v_{CE})$ is shown in **Figure 9.3b**.



(a) *Example of $f(\Delta l_{CE})$*



(b) *Example of $g(v_{CE})$ with $a(t) = 1$*

Figure 9.3: *Example of the force-length and force-velocity functions*

The Activation Signal

The muscle activation signal, $0 \leq a(t) \leq 1$, is a function of the EMG measurements and is a measure of the degree of recruitment of the muscle when neurally activated. The EMG signal undergoes some processing before it can be used as a muscle activation level. The processing is shown graphically in **Figure 9.4**.

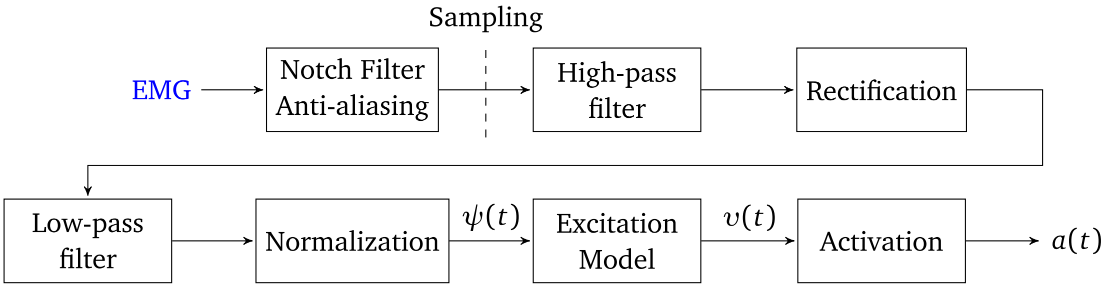


Figure 9.4: *The different processing steps necessary for converting EMG signals into a muscle activation level.*

The notch filter removes 50 Hz noise which is present anywhere in the vicinity of electrical installations. This, along with an anti-aliasing filter is implemented by Thalmic Labs on the Myo Band. The high-pass filter removes any DC offset. The rectification along with the low-pass filter functions as an implementation of a linear envelope. The high-pass and low pass filters have been designed according to [32]. Specifically the high-pass filter is a forth order Butterworth filter with a cut-off frequency at 30 Hz and the low-pass is also a fourth order Butterworth with a cut-off frequency of 6 Hz. The low-pass filtered signal is normalized with respect to the maximum values the Myo Band can measure which is 128, the highest absolute value of a signed 8-bit integer. According to [32] it is recommended to add a second low-pass filter, or as they call it a *muscle excitation model*, as a difference equation of the form **Equation (9.8)**. It is a second order integrator, that results in the signal $\upsilon(t)$. This signal reflects the way that muscles react to neural excitation, where the nerve signal builds up its effect on the muscle when repeated, and it takes into account the electromechanical delay from the EMG-signal to muscle movement. The muscle excitation model is implemented as the difference equation:

$$\upsilon(t) = \xi\psi(t-\delta) - \beta_1\upsilon(t-1) - \beta_2\upsilon(t-2) \tag{9.8}$$

where δ is the electromechanical delay from stimulation to muscle response. The variables ξ, β_1, and β_2 are filter parameters. The filter, **Equation (9.8)**, should be near critically damped and with unit gain to approximate the nature of the skeletal muscles. Imposing the constraints **Equation (9.9)-(9.10)**, ensures that the filter is damped if the design parameters, C_1 and C_2, are chosen as either negative or with opposite signs with $|C_1| > |C_2|$. The constraint **Equation (9.11)** ensures that the filter has unit gain.

$$\beta_1 = C_1 + C_2 \quad \text{where} \quad |C_1| \leq 1 \quad \text{and} \quad |C_2| \leq 1 \tag{9.9}$$

$$\beta_2 = C_1\, C_2 \tag{9.10}$$

$$\alpha = 1 + \beta_1 + \beta_2 \tag{9.11}$$

The activation function shown in **Equation (9.12)** is a widely accepted estimate of the activation of the muscle [32][33].

$$a(t) = \frac{e^{A\,\upsilon(t)} - 1}{e^{A} - 1} \tag{9.12}$$

Here A is the degree of nonlinearity[**moosavi_ehsani**], and $0 \leq \upsilon(t) \leq 1$ is the EMG signal after integration [32]. Examples of the relation between $a(t)$ and $\upsilon(t)$ are shown in **Figure 9.5**.

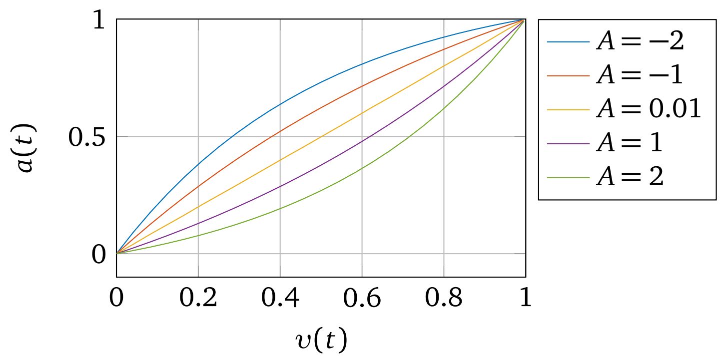


Figure 9.5: *Example of the relation between $a(t)$ and $\upsilon(t)$. [32]*

9.2.2 Model of the Serial Element & Parallel Element

The Hill Muscle Model attempts to relate muscle models to a spring, mass and damper system, where SE and PE act as nonlinear springs. In [6] they use a different approach to model this type of spring. They drop the connection to mechanical springs, as their model is expressed in terms closer to the biology of the arm. It approximates the relation shown in **Figure 9.2** by **Equation (9.13)**, using the relevant muscle parameters. The force of the passive spring elements of the hill model becomes a function of the change in muscle or tendon length, $\Delta l_{PE,SE}$, the maximum passive force, $F_{PE,SE_{max}}$, the shaping parameter S, the maximum length and change in length of the muscle, L_{max} and ΔL_{max}, the slack length of the muscle and tendon, L_{m_s}, L_{t_s}. The equations below describe the implementation of this non-linear spring model, and how it depends on the muscle parameters.

$$F_{PE,SE} = \frac{F_{PE,SE_{max}}}{e^{S_{PE,SE}} - 1} \left(e^{\frac{S_{PE,SE}}{\Delta L_{PE,SE_{max}}} \Delta l_{PE,SE}} - 1 \right) \quad [\mathrm{N}] \quad (9.13)$$

$$F_{PE_{max}} = 0.05\, F_{CE_{max}} \quad [\mathrm{N}] \quad (9.14)$$

$$\Delta l_{PE} = l_{PE} - L_{m_s} \quad [\mathrm{mm}] \quad (9.15)$$

$$\Delta L_{PE_{max}} = L_{max} - (L_{CE_0} + L_{t_s}) \quad [\mathrm{mm}] \quad (9.16)$$

$$F_{SE_{max}} = 1.3\, F_{CE_{max}} \quad [\mathrm{N}] \quad (9.17)$$

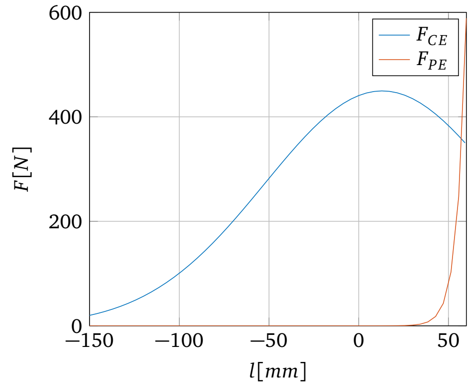
$$\Delta L_{SE_{max}} = 0.03\, L_{t_s} \quad [\mathrm{mm}] \quad (9.18)$$

$$\Delta l_{SE} = l_{SE} - L_{t_s} \quad [\mathrm{mm}] \quad (9.19)$$

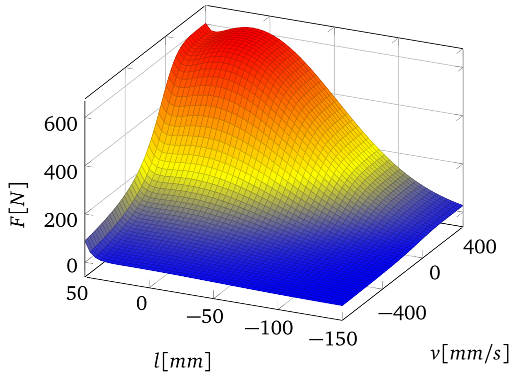
This concludes the model of the *Muscle Force* part of **Figure 9.1**. Following is a summary of the main conclusions.

The different parts of **Equation (9.3)** are accounted for, using approximations for the force-length and force-velocity relationships. A model for the level of activation, $a(t)$, is found as a function of the EMG signal, $\eta(t)$.

The passive elastic parts of the muscle are also modeled, using **Equation (9.13)**, which models the passive spring force as a function of the length of the muscle. The resulting force of the muscle can therefore be modeled as the sum of the two curves on **Figure 9.6a**, and when that curve is evaluated over the force-velocity relationship, we get the surface plot in **Figure 9.6b**. This plot shows that the force a muscle can exert, depends not only on how well trained the subject is, but also on the type of movement and the pose of the joints.



(a) *$F_{CE}(\Delta l, v)$ and $F_{PE}(\Delta l)$ with $v = 0$, and $a(t) = 1$.*

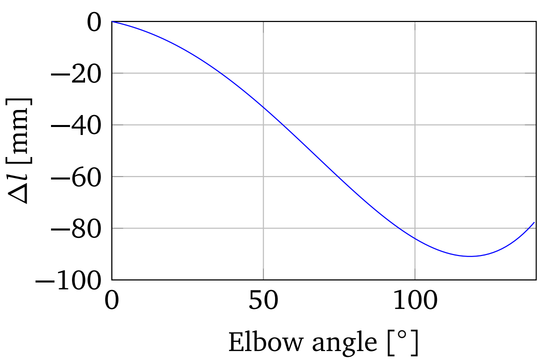


(b) *Surface plot of the force, $F(\Delta l, v)$, with $a(t) = 1$*

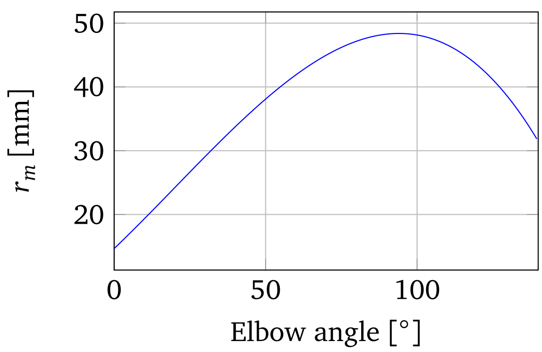
Figure 9.6

9.3 Dynamics of Muscle Length and Moment Arm

The calculation of muscle force includes an estimated relationship between the muscle length and the force it can exert. Therefore, it is necessary to estimate the length of a given muscle. The length of the muscles in the arms can be estimated using the arm joint angles. In [34], the length of 13 arm muscles are determined as polynomial functions of joint angles. For example, the length of biceps brachii can be approximated by a fourth order polynomial of the elbow joint angle plus a first order polynomial of the shoulder angle. In **Figure 9.7a** the change in length of biceps brachii is plotted as a function of the elbow angle where the shoulder angle is assumed to be 0°. Similar functions are presented in [34] for shoulder flexion/extension, elbow flexion/extension, wrist flexion/extension, and wrist radio-ulnar motion. These have been implemented for use in the muscle estimator. Naturally, the length of a given muscle mainly depends on the joint that the muscle can manipulate.



(a) *The change in length of biceps brachii as a function of the elbow angle. The shoulder extension/flexion is assumed to be 0°.*



(b) *The Moment arm of biceps brachii as a function of the elbow angle. The shoulder extension/flexion is assumed to be 0°.*

Figure 9.7

Similarly, [34] presents the length of the moment arms in the joints as polynomial functions of joint angles. This is useful for converting the force generated by the muscles to torque in the adjacent joints. The moment arm, r_m, is defined as the perpendicular distance between the muscle and the joint it exerts force upon. Its estimation is necessary to convert the force vector of a muscle to a joint torque. The moment arm, r_m, depends on the angle of activation, θ_a of the muscle force, $\bar{F}$. A visualization of the moment-arm is shown in **Figure 9.8**. In **Figure 9.7b** are the relationship between the angle of the elbow and length of the moment arm of biceps brachii with relation to the elbow shown, but [34] also contains polynomial estimations of the moment arms of the other 12 arm muscles it describes, thus providing all the necessary knowledge to implement the block *Muscle Shape* in **Figure 9.1**.

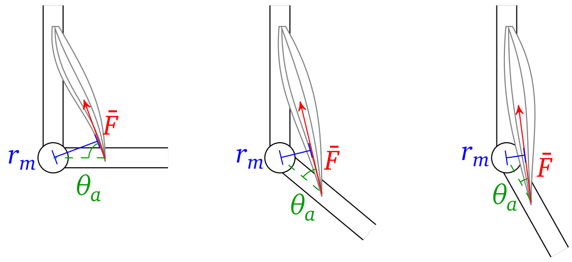


Figure 9.8: *The moment arm is defined as the perpendicular distance between the muscle force vector and the joint. As the figure shows, it changes with the joint angle.*

9.3.1 Finding the Length of the Muscle Fibers

In **Section 9.3** it is shown, that the length of the muscles can be estimated by the joint angles. The muscle model needs the length of the muscle, and either the length of the muscle fibers, or the length of the tendon. Hence, it is investigated how the length of the muscle fibers and tendon can be found.

The structure of the Hill Model shown in **Figure 9.2** is equivalent to the structure that is used in [6], where the SE is in series with the CE, as shown in **Figure 9.9**. Looking at this model structure it follows that the force generated by the CE and SE are the same (**Equation (9.20)**). Furthermore, the length of the muscle is the sum of its parts, that is, the length of the muscle is the length of the muscle fibers and the length of the tendon (**Equation (9.21)**).

The contractile velocity is also used in the model in the force-velocity function (**Equation (9.6)**), it is impossible to measure the contractile velocity in a non intrusive manner, hence the velocity is approximated with a differential approximation using the length of the fibers (**Equation (9.22)**).

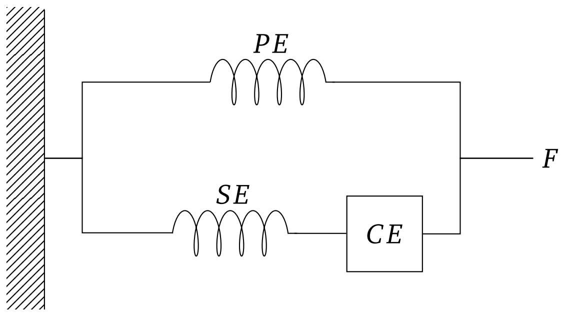


Figure 9.9: *Another representation of Hill's Muscle Model [6].*

$$F = F_{SE} + F_{PE} = F_{CE} + F_{PE} \implies F_{SE} = F_{CE} \tag{9.20}$$

$$l_{PE} = l_{SE} + l_{CE} \implies \Delta l_{PE} = \Delta l_{SE} + \Delta l_{CE} \tag{9.21}$$

$$v_{CE} = \frac{\Delta l_{CE}[n] - \Delta l_{CE}[n-1]}{t_s} \tag{9.22}$$

These relations with forces and lengths, as well as the differential approximation, can be used to numerically estimate Δl_{CE} by using a root finding method on **Equation (9.23)**.

$$0 = F_{SE}(\Delta l_{PE} - \Delta l_{CE}[n]) - F_{CE}(\Delta l_{CE}[n], \Delta l_{CE}[n-1]) \tag{9.23}$$

The bisection method is selected as it doesn't require computing the derivative, and guarantees convergence provided it is possible to straddle the root.

With Δl_{PE} found via the joint angles, and Δl_{CE} numerically found, all the non-constant variables needed in the model are found. Before the model can be used in conjunction with the AXO-arm, the muscles that carry the most information about the movements of interest need to be identified.

9.4 Choice of Muscles

Before the muscle model can be implemented to calculate joint torques on the user of the AXO-arm, it should be investigated which muscles are of interest. The two DOF on the AXO-arm allow flexion/extension of the elbow- and shoulder joint.

9.4.1 Elbow Joint Muscles

The primary elbow flexors are *biceps brachii*, *brachialis* and *brachioradialis* (**Figure 9.10**) and the primary extensor is the *triceps brachii*[31]. The flexor *pronator teres* and the extensor *aneconeus* could also provide useful information, however these are located near the elbow joint, and will be hard to measure with the selected Myo Band sensor, they are therefore disregarded. A brief description of the chosen muscles follow below.

Biceps brachii is a large muscle attached to the *humeral* and the *radioulnar* articulations. As the muscle is connected to these articulations contraction of the *biceps brachii* will affect the corresponding joints, namely the *humeroulnar* and *humeroradial* joints [31].

Brachialis is a monoarticular muscle, i.e. a “elbow only” muscle, and does not have any influence in forearm pronation and supination. It can only be used for elbow flexion [31].

Brachioradialis is located near the wrist extensors, and even shares an *innervation* with them, yet it is still considered an elbow flexor, and there are ongoing discussions about its role regarding elbow supination and pronation [31].

Triceps brachii is the biggest muscle related to extension of the elbow joint, and has no influence on supination or pronation of the arm. It is attached to two places on the *humerus* with the *lateral head*, and the medial head. It is also attached on the *scapula* with the long head. All three heads meet up, and are attached to a common place on the *ulna* [31].

9.4.2 Shoulder Joint Muscles

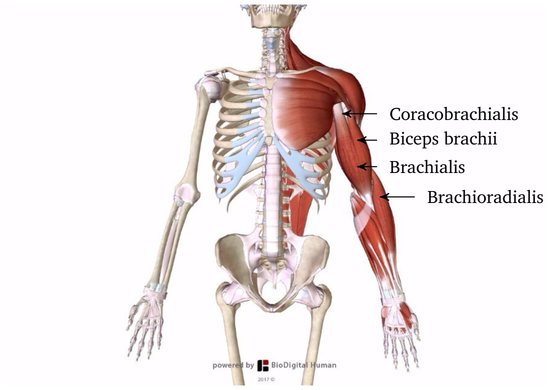
According to [35] the muscles responsible for shoulder extension are *latissimus dorsi, teres major* and *deltoid* and for flexion it is *coracobrachialis* and *deltoid*, shown on **Figure 9.10**.

Latissimus dorsi is a large back muscle, attached to the *spine, pelvis* and *humerus*. It has a supportive role in shoulder flexion [31].

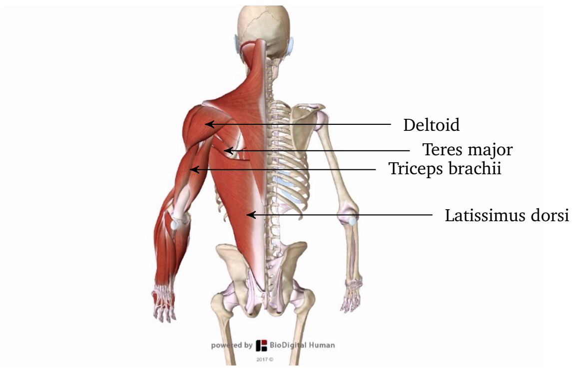
Teres major is attached to the *scapula* and *humerus*. It supports shoulder extension [31].

Deltoid is a group of muscles that is attached to *scapula* and *clavicle* in one end and to *humerus* in the other. Deltoid has a supporting role during shoulder flexion [31].

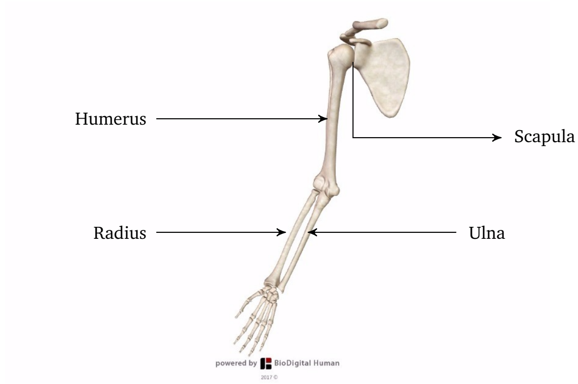
Coracobrachialis is a shoulder flexor that is hidden behind *biceps brachii* in **Figure 9.10a**. Its purpose is not well studied, but it is thought to be a shoulder flexor [31].



(a) *Front of a human body showing the muscles in the arm.*



(b) *Back of a human body showing the muscles in the arm.*



(c) *Front of a humans right arm, showing the bones.*

Figure 9.10:

The project will be limited to focus on control of the elbow joint. The muscles responsible for shoulder joint movement, such is deltoid and latissimus dorsi are not measurable using the Myo Band, because it is strapped around the arm. The scope of the project is therefore narrowed to focus on elbow joint movement. To measure the EMG activity of *biceps brachii*, *brachialis* and *triceps brachii* are a Myo Band placed on the upper arm of the test subject. The Myo Band is attached to the subject's upper arm such that pod 4 is placed on *biceps brachii*. As *brachialis* is placed directly under *biceps brachii* are the same pod used for *brachialis*. With the placement of pod 4 is it natural that pod 8 is used for *triceps brachii*, as it is placed directly on that muscle. To measure *brachioradialis* are a Myo Band placed on the lower arm, with pod 4 on-top of *brachioradialis*, with these two Myo Bands it is possible to estimate the torque generated in the elbow joint, as the sum of torques of the selected muscles, M, once the muscle model have been parameterized (**Equation (9.24)**).

$$\bar{\tau}_{\text{joint}} = \sum_{m \in M} F_m \, r_m \qquad [\mathrm{N\,m}] \qquad (9.24)$$

where F is the muscle-force and r is the perpendicular distance from the muscle-force vector to the joint.

Summary

The main takeaway of this chapter is that muscle force estimation is inherently difficult. Even in the presented method, there are many obstacles. The parameters are not easily obtainable, and the variables, such as muscle length or contraction velocity are not easy to measure. Many of the models are heuristic estimations of the biomechanic dynamics that needs optimization and fitting to be of any use. Before moving on to finding the parameters, this last section of the chapter gives an overview of the entire model that allows calculation of joint torques based on EMG measurements.

As shown in **Figure 9.11** the muscle model has four main parts:

- EMG Processing
- Muscle length calculation
- Joint moment arm calculation
- Muscle force calculation

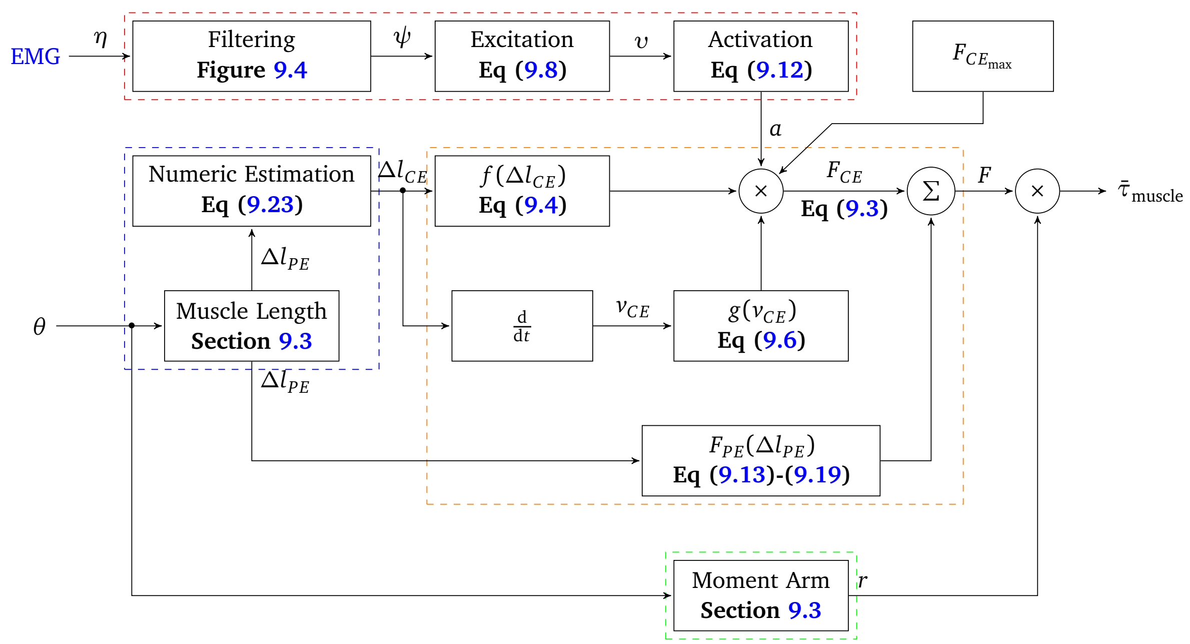


Figure 9.11: *The calculation model of the joint torque contribution of a single muscle.*

When the total joint torque is calculated, the torque contributions from all the flexors are added, and the torque contribution of the extensors are subtracted, as shown in **Figure 9.12**.

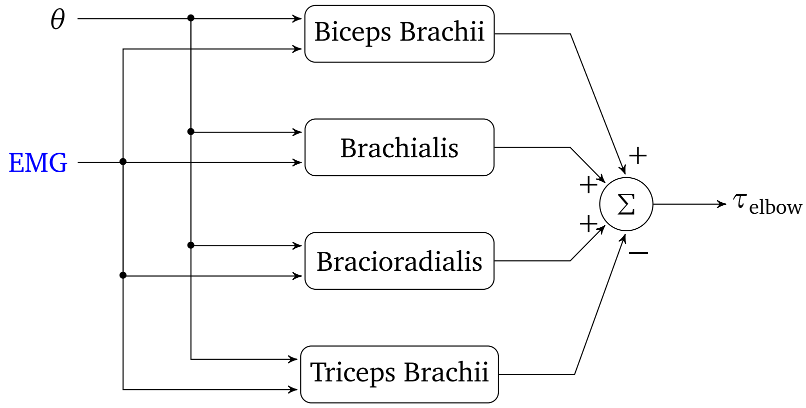


Figure 9.12: *How the total joint torque is calculated from the individial muscles. The EMG signal comes from differnet sensors for different muscles, but are lumped together here for simplicity.*

This concludes the chapter on modelling the biomechanics that generate torque in the elbow joint. In the following chapter, a method for estimating all the unknown paramaters of the model is presented, and after the parameters are found, the validity of the model is evaluated experimentally.

10 Parameter Estimation of the Muscle Models

In the muscle model there is a number of unknown parameters. This chapter describes the process of estimating these parameters. **Section 10.1** summarizes which parameters need to be estimated and **Section 10.2** presents a white-box estimation method for finding the parameters of the muscle model. Lastly, **Section 10.4** shows the parameters that were found, and validates the result for further use in the controller.

10.1 Model Parameters

The muscle model consists of three parts:

- Generation of the activation signal (**Section 9.2**).
- Force estimation based on the activation signal (**Section 9.2.1**).
- Conversion from force to torque (**Section 9.3**)

A list of all the unknown muscle parameters is shown in **Table 10.1**.

Name	Symbol	Unit	Equation
Maximum contractile force of CE	$F_{CE_{max}}$	N	9.3
Maximum muscle length	L_{max}	m	9.16
Optimal length of CE	L_{CE_0}	m	9.4, 9.6
Tendon slack length	L_{t_s}	m	9.16, 9.19
Percentage of fast muscle fibers	ξ	%	9.7
Shape parameter of SE or PE	S_{SE}, S_{PE}	·	9.13
Gaussian tuning parameters	ϕ_m, ϕ_v	·	9.4
Electromechanical Delay	ψ	s	9.8
Degree of nonlinearity of activation function	A	·	9.12
Filter parameters	C_1, C_2	·	9.9–9.11

Table 10.1: *Table of all muscle parameters that need to be estimated. There is one of each parameter per muscle.*

Most of the parameters are person dependent, and reasonable initial guesses are difficult to find. However we do know something about the Electromechanical Delay (EMD).

In [36] the EMD of *biceps brachii* and *triceps brachii* are found as 41 ± 13ms and 26 ± 11ms, respectively. These values are used as initial guesses in the parameter estimation. The EMD of *brachialis* and *brachioradialis* are given the same initial value as *biceps brachii*.

The estimation of parameters for the muscle model consists of a minimization of the MSE between the estimated human joint torque and the measured joint torque. The measurement procedure is described in **Appendix B**. The muscle model estimates a joint torque from the joint angle and the EMG. The optimization problem is thus shown graphically in **Figure 10.1**.

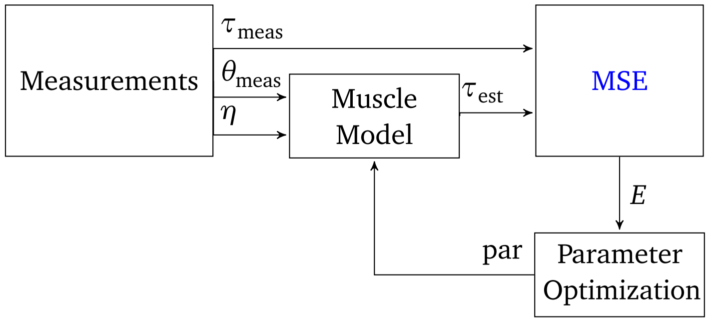


Figure 10.1: *The structure of the parameter optimization of the muscle model.*

The parameters are optimized for two different models. The one shown in **Figure 9.12**, and a reduced model with only the primary extensor and flexor; (*biceps bracii* and *triceps brachii*). This is partly to reduce the dimensions of the estimation (26 vs 52 parameters), and partly to test if we can get a good enough estimate of the joint torque in the elbow with only one Myo Band.

These two models are compared and it is evaluated whether this simplification results in an acceptable trade-off.

10.2 Choice of Estimation Algorithm

It is in [6] suggested to estimate the optimal parameters using a Genetic Algorithm (GA). This method handles large search spaces well and it has a small risk of converging to only locally optimal solutions. In this project, a different population-based optimization algorithm called Particle Swarm Optimization (PSO) is used. This is chosen on the basis of the findings of [37], and [38].

In [37], the performance of an implementation PSO and GA is evaluated. They found that for most problems, the solutions were equally good. Generally PSO was found to produce a more stable solution (smaller variance in the result over multiple runs) with faster convergence. Similar results were found in [38], where their PSO and GA estimate reached very similar parameters, but PSO converged after only 10 generations where GA converged after 25 generations. Based on this, PSO is chosen to estimate the muscle parameters.

A detailed description of the PSO algorithm is presented in the following section.

10.2.1 Particle Swarm Optimization

The PSO algorithm started as an attempt to simulate the flocking behavior of bird flocks and shoals of fish, which was later modified to an optimization algorithm. A key aspect within the swarm of birds and fish is the flow of information. If a predator approaches, the swarm is moving in a cohesive way, almost as if it was a single individual. [39]

PSO is built around these concepts, and each particle within the swarm moves around based on it's own knowledge, as well as the knowledge of the swarm (in the simplest implementation). Each particle can be viewed as a member of a flock. The location of each particle represents a possible solution to a given problem, and to evaluate the performance of the location a fitness function is needed.

Each particle saves their personal best location, and communicates it with the rest of the swarm. Within the swarm the best location off all the current and previous locations are saved. The particles move around independently with this information available, and adjusts their courses based on it. To avoid early convergence the course adjustments have random perturbations added to them.

Equation (10.1)-(10.2) show the updating step of the particles' position and velocity for an N-dimension optimization problem with M particles.

$$\bar{v}^i[k+1] = w\,\bar{v}^i[k] + \phi^p\,\bar{r}^p\,(\bar{p}^i - \bar{x}^i[k]) + \phi^\gamma\,\bar{r}^\gamma\,(\bar{\gamma} - \bar{x}^i[k]) \tag{10.1}$$

$$\bar{x}^i[k+1] = \bar{x}^i[k] + \bar{v}^i[k+1] \tag{10.2}$$

$$\bar{x}^i, \bar{v}^i, \bar{p}^i, \bar{\gamma} \in \mathbb{R}^N \tag{10.3}$$

$$\bar{r}^p, \bar{r}^\gamma \in \mathscr{U}^N(0,1) \tag{10.4}$$

$$i \in 1, 2, \ldots M \tag{10.5}$$

Where $\bar{x}^i[k]$ and $\bar{v}^i[k]$ are the location and velocity of the ith particle at time k, w is an inertia factor. The vector $\bar{p}^i$ is the "personal best location" of particle i and ϕ^p is the influence factor of the personal best. Similarly $\bar{\gamma}$ is the best location in the swarm and ϕ^γ is an influence factor of the global best. The random vectors $\bar{r}^p$ and $\bar{r}^\gamma$ ensure pseudo-random movement of the particles, to avoid early convergence. The algorithm is shown in **Algorithm 1** [40]. The `fitness()` function is minimal when the fit is optimal.

Algorithm 1 The standard Particle Swarm Optimization algorithm

```
Initialize a population of particles with random positions and velocities
while Termination condition not reached do
    for Each particle i do
        for Each dimension n do
            Update position and velocity (Equation (10.1)-(10.2))
        end for
        if fitness(x̄^i) < fitness(p̄^i) then
            p̄^i ← x̄^i
        end if
        if fitness(x̄^i) < fitness(γ̄) then
            γ̄ ← x̄^i
        end if
    end for
end while
```

10.3 Method

Before the parameters can be estimated, a set of input-output measurements needs to be obtained. In this case, the input is the EMG signal and joint angle, and the output is joint torque. At the health department at AAU a Kin Com machine is located. The machine can be used to measure the force (or torque) generated in a joint. Two Myo Bands have been used to measure the EMG signal; one on the forearm and one on the upper arm. The test setup and procedures are described in **Appendix B**.

Once the input-output data is collected, the parameter estimation has been performed using a library called Parallel Global Multiobjective Optimizer (PaGMO) [41]. It is maintained by the European Space Agency, and they also provide Python-bindings for the optimization algorithms. The library can run several solvers in parallel, and has several implementations of optimization algorithms, including PSO.

The library is based around inheritance, and it includes an abstract base class for optimization problems. To estimate parameters for the activation filters and muscle models a concrete problem has been implemented. The implementation overrides the objective function which will be minimized, and a function that informs the solver about constraints. The objective function instantiates the muscle models and activation filters, and runs a full simulation with the current parameter guess. < As described in **Appendix B** the Myo Bands start logging before the Kin-Com. This is because it takes a while to connect to them, but this has the added effect of allowing the EMG filters to settle before logging the data from the Kin-Com. The EMG signal is passed through a series of filters that are implemented as difference equations. The data that is used for parameter fitting thus has meaningful filter outputs right from the start.

The torque estimates from all six exercises (3 flexion and 3 extension) are appended to each other to form a single vector, allowing us to calculate the MSE between the simulated and measured torque.

Both the torque estimates from the simulations are appended together to form a single time series, and the torque measurements are as well. The MSE between them is then computed, and the MSE between the simulations and the measured are used as the result of the fitness function, that is minimized by the PSO.

In **Section 9.2.1** two constraints are mentioned, which of course should be included in the optimization problem, it has been done by overwriting the constraint function with:

- A constraint for C_1 and C_2 that ensures that either are both negative, or they have opposite signs with $|C_1| > |C_2|$.
- A constraint that ensures that the max length (L_{max}) is bigger than the sum of the optimal fiber length (L_{CE_0}) and the tensor slack L_{t_s} length.

The two different models (the full 4 muscle mode, and the reduced 2 muscle model) have been estimated using this setup. Each model has been optimized with 200 different initial guesses with 25 particles, and 500 iterations, as this configuration provided a good trade-off between the quality of the solution and the computation time. The default parameters of the PSO algorithm in PaGMO were used, they can be found in [41].

10.4 Estimation Experiment and Results

In this section the overall results of the parameter estimation are discussed, and the performance of the reduced muscle model (with only 2 muscles) is compared to the full muscle model. The estimation is based on 3 extension and 3 flexion exercises, with 10 repetitions each.

10.4.1 Activation Signal

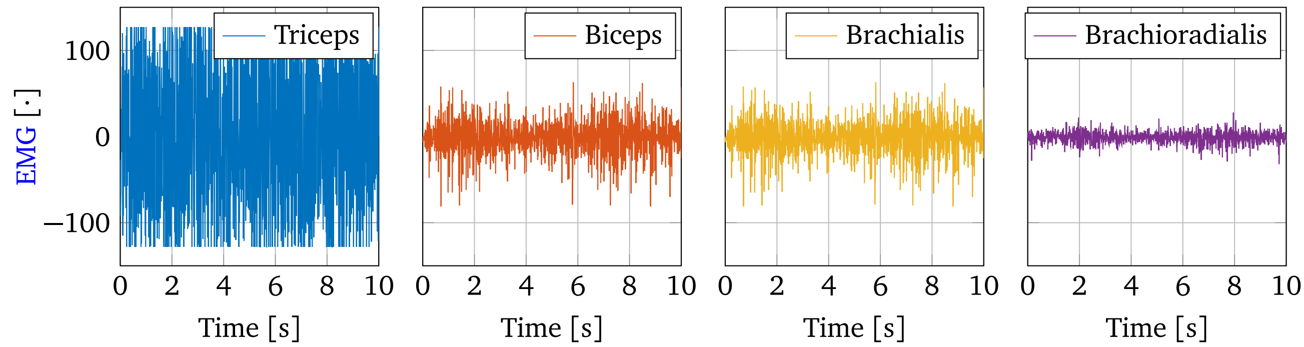
A slight modification to the filter processing described in **Section 9.2.1** – the EMG signal is re-sampled using the Fourier method with a rectangular window. It is re-sampled to a 100 Hz signal before the high-pass filter. This is to easier align the EMG signal with the angles and torques measured on the Kin-Com, which samples at 100 Hz. For each activation signal, four parameters are estimated: the EMD, d, the two design parameters, C_1 and C_2, and finally the degree of non-linearity, A. The estimated values for the two different models are shown in **Table 10.2**[1].

On **Figure 10.2** the glsEMG signal from the Myo Band and the estimated activation signal from the first 10 s of an extension are shown. **Figure 10.2a** shows the EMG signal and **Figure 10.2b** the corresponding activation signals It is evident on **Figure 10.2a** that the Myo Band is saturating. On **Figure 10.2b** are the 4 muscle model on the left side, and the 2 muscle on the right side. It can be seen that the EMG and activation signal is significantly higher for triceps than the others, which is expected as triceps is the primary extensor.

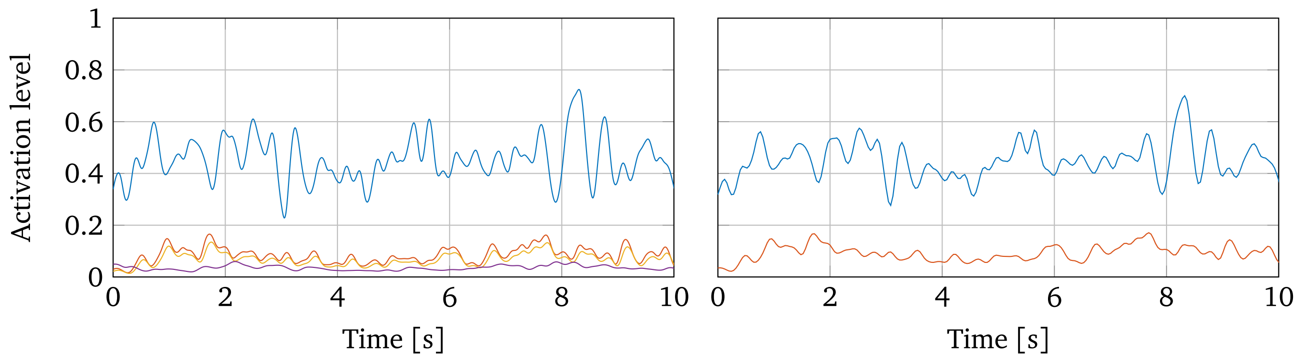
[1]Due to an implementaion error, C_1 and C_2 in the two muscle model do not keep the constraints from **Section 10.3**. A better result might be obtained from a new parameter estimation.

	Four Muscle Model				Two Muscle Model	
	Triceps	*Biceps*	*Brachialis*	*Brachioradialis*	*Triceps*	*Biceps*
$\mathbf{C_1}$	-0.62	-0.06	-0.71	-0.92	-0.87	0.42
$\mathbf{C_2}$	-0.55	-0.59	-0.55	-0.16	0.24	-0.87
A	0.24	0.56	0.97	0.05	0.26	0.34
δ [s]	0.03	0.07	0.08	0.01	0.02	0.05

Table 10.2: *Estimated parameters for the activation filters in the muscle models.*



(a) *Excerpt of the raw EMG signal from the Myo Band during an extension exercise.*

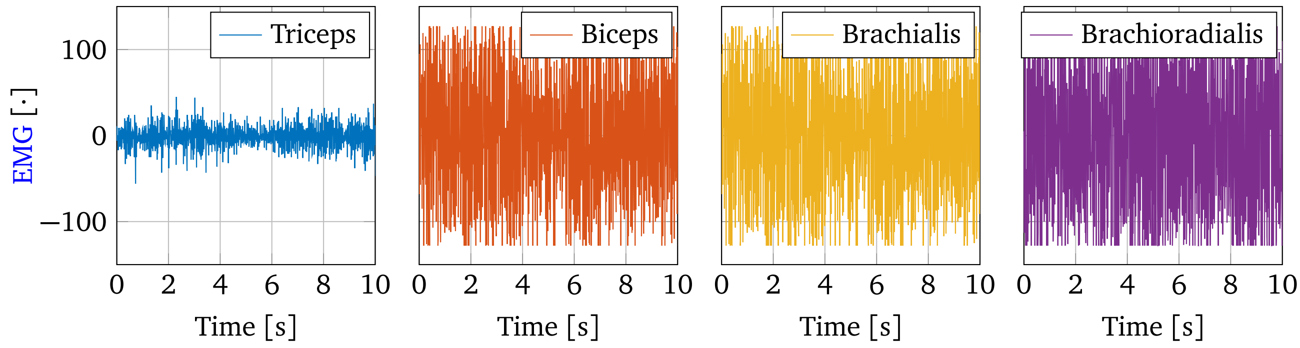


(b) *Excerpt of the estimated activation signal based on* ***Figure 10.2a***

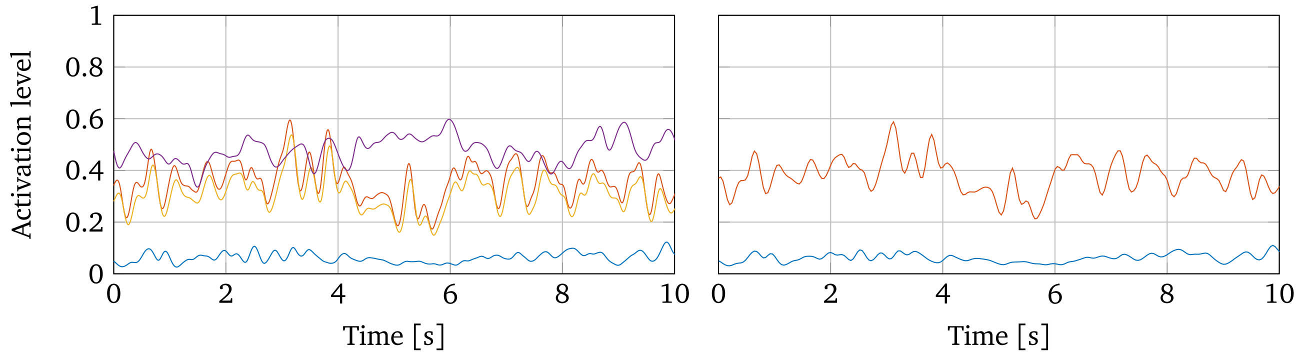
Figure 10.2

Similarly the first 10 seconds of an flexion exercise are shown in **Figure 10.3**. The EMG signal from the Myo Band is shown on **Figure 10.3a**, with the related activation signals in **Figure 10.3b** – again with the 4 muscle model on the right side. It is apparent that the information about *brachioradialis* is lost in the 2 muscle model, further is it apparent that the activation signal to *brachialis* is very close to the activation signal of *biceps*. This is likely due to the placement of *brachialis*, right under biceps.

The activation level seems generally low considering that the test subject was working as hard as he could against the Kin-Com machine. This could be caused by the low EMG signal range of the Myo Band (shown in **Appendix ??**). This is an undesired trait of the Myo Band but no attempts to overcome this limitation has been made, as the overall result of the parameter estimation are satisfactory.



(a) *Excerpt of the raw EMG signal from the Myo Band during an flexion exercise.*



(b) *Excerpt of the estimated activation signal based on* ***Figure 10.3a***

Figure 10.3

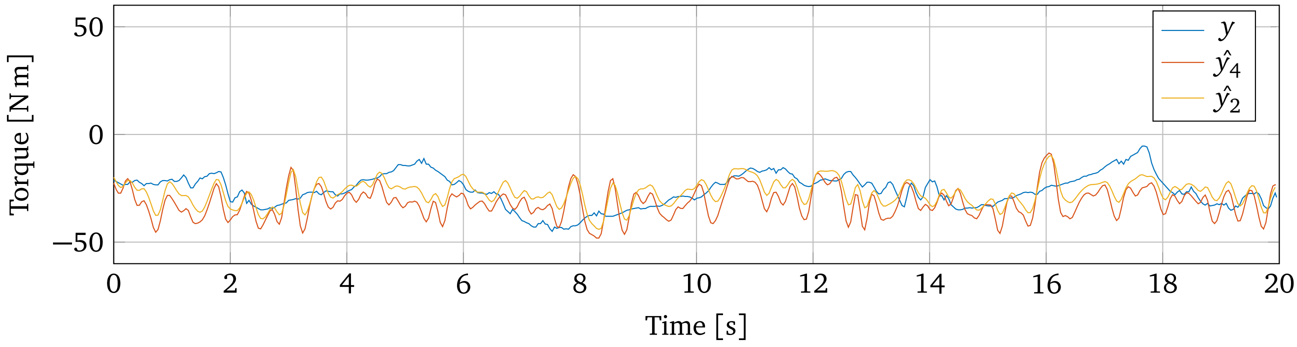
10.4.2 Torque Estimation

For the Hill Muscle Model nine parameters were estimated per muscle. The estimated values are shown in **Table 10.3**. Some of the parameters are a lot higher than expected, and unrealistic from a biological point of view. For the normalized force-length ($f(l_{CE})$) and force-velocity ($g(v_{CE})$) function, this is likely due to the freedom the optimization algorithm had, where no bounds were added, and the only constraints it had was those mentioned in **Section 8.1b**. The large values for $F_{CE_{max}}$ could be due to the low activation signal, recall that the model of the contractile element is $F_{CE} = a(t)f(l_{CE})g(v_{CE})F_{CE_{MAX}}$

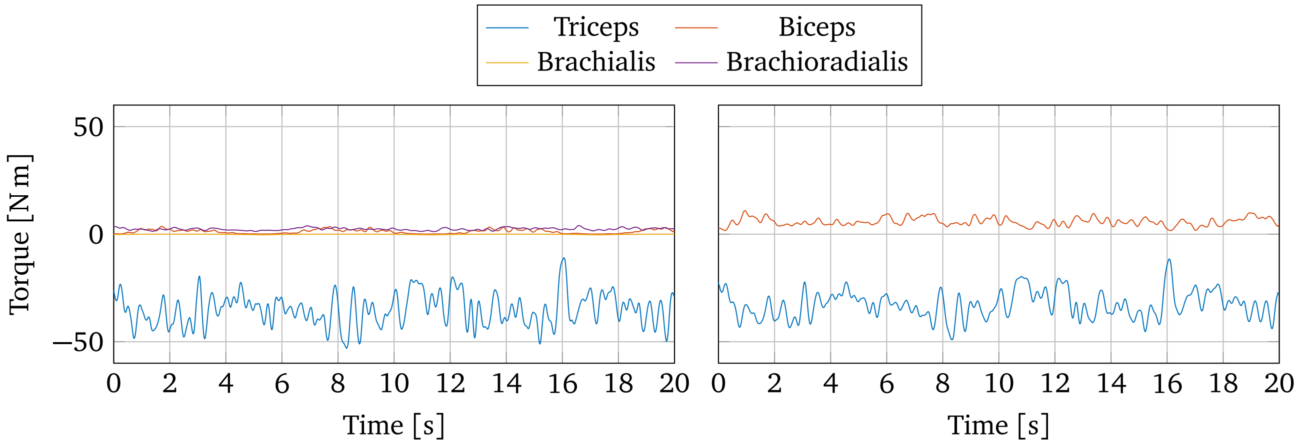
	Four Muscle Model				**Two Muscle Model**	
	Triceps	*Biceps*	*Brachialis*	*Brachioradialis*	*Triceps*	*Biceps*
$\mathbf{S_{PE}}$	2.91	14.12	1.32	16.61	9.19	8.71
$\mathbf{S_{SE}}$	1.61	9.92	9.34	15.52	12.24	6.25
α	0.95	0.93	0.99	0.74	0.89	0.84
$\phi_{\mathbf{m}}$	0.54	0.11	0.94	0.93	0.15	0.84
$\phi_{\mathbf{v}}$	0.75	0.20	0.14	0.92	0.36	0.75
$\mathbf{F_{CE_{max}}}$ [N]	4293	1511	3121	4692	3485	5158
$\mathbf{L_{max}}$ [mm]	581	864	790	306	630	812
$\mathbf{L_{CE_0}}$ [mm]	191	148	233	124	220	199
$\mathbf{L_{t_s}}$ [mm]	323	203	280	102	226	156

Table 10.3: *Parameters for the Hill Muscle Model*

On **Figure 10.4** it is evident that the two different models estimate roughly the same, y is the measured torque, $\hat{y}_4$ and $\hat{y}_2$ are the model responses to the EMG. This is expected as only one extensor is included in each model, namely *triceps brachii*. The effect of reducing the model is more



(a) *Total torque of the a extension exercise series*



(b) *Individual torque estimations for the 4 muscle model of* ***Figure 10.4***

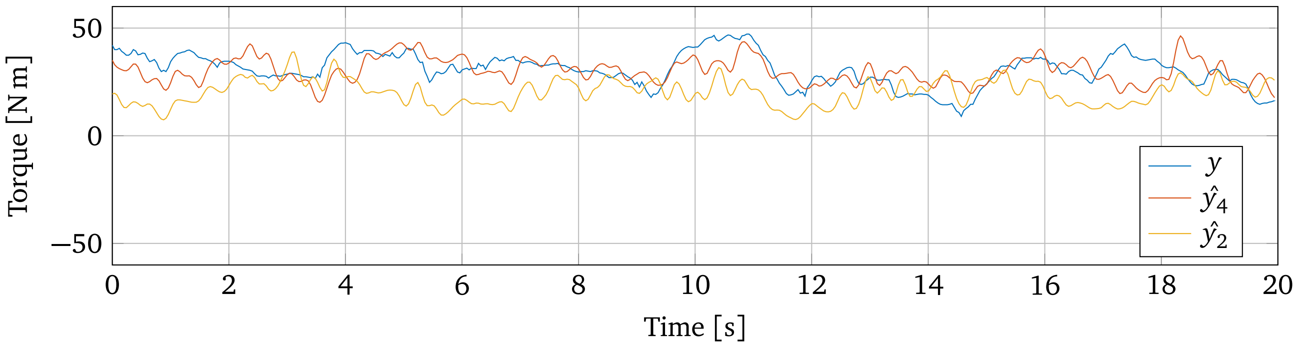
Figure 10.4

evident in **Figure 10.5**, where the 2 muscle model doesn't fit the measured data as well as the 4 muscle model does. Looking at the different models at **Figure 10.5b**, with the 4 muscle model on the right and the 2 muscle model on the left, it is clear that this is due to the reduction of the model, and the loss of *brachioradialis*. Even though the performance of the 2 muscle model is clearly worse then the 4 muscle model, it is still performing quite well with a fitness of $E = 0.78$ compared to a fitness of $E = 0.83$. With both models performing better than required (as described in **Section 3.2.3**), both are accepted for further testing against a verification data set.

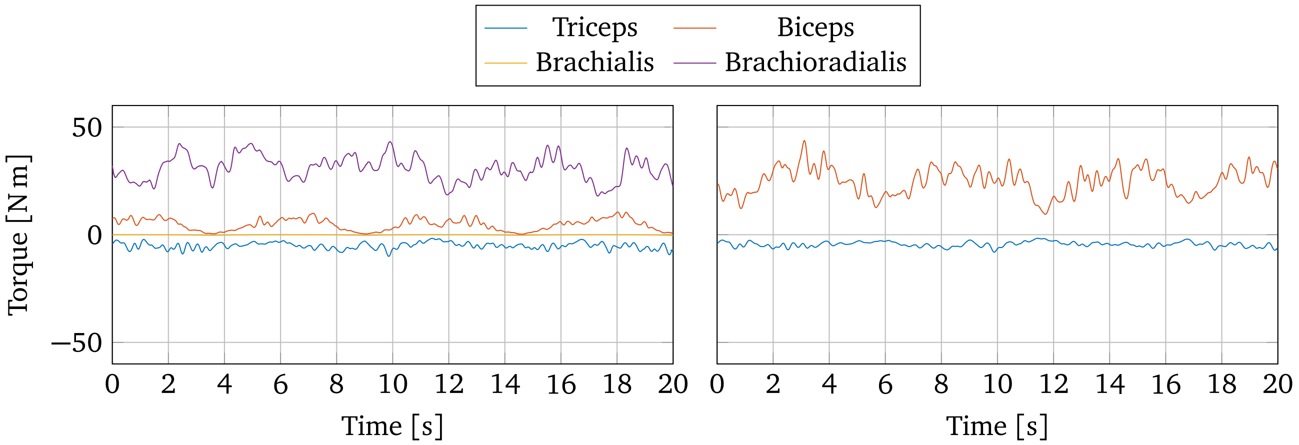
10.5 Parameter Validation

To verify the parameters a verification data set has been captured. The set contains measurements from two extension and two flexion exercises. An excerpt of the data is shown in **Figure 10.6**. The overall performance is very satisfactory, of both the 4 muscle model, and 2 muscle model, with a fitness of $E = 0.88$ and $E = 0.85$ respectively, measured using NMSE.

As the performance of the 4 muscle is not significantly better then the 2 muscle model, will the 2 muscle model be used, as it is both computationally cheaper, and doesn't require an additional Myo Band.



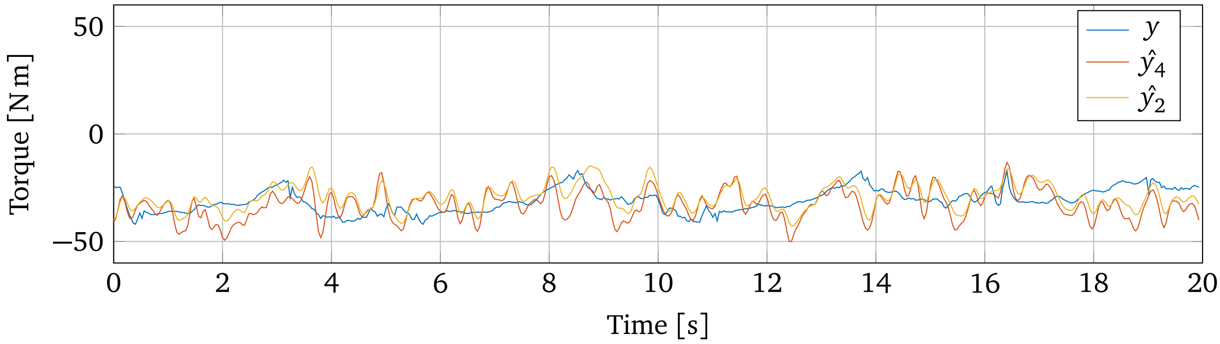
(a) *Total torque of the a flexion exercise series*



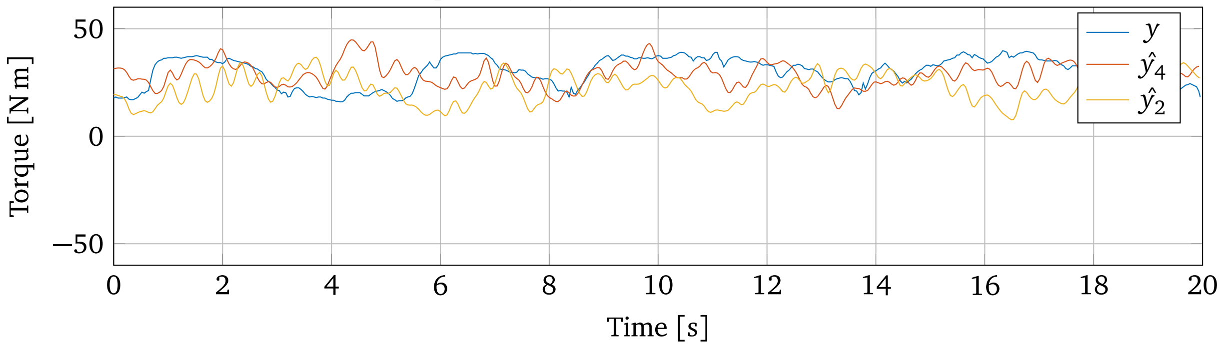
(b) *Individual torque estimations for the 4 muscle model of* ***Figure*** *10.5. Uses the same coloring scheme as* ***Figure*** *10.4b*

Figure 10.5

With the muscle model and mechanical model parameterized and evaluated we are now ready to evaluate appropriate control strategies.



(a) *Total torque of the beginning of an extension exercise*



(b) *Total torque of the beginning of an flexion exercise*

Figure 10.6

Part III

Controller Design

Having created a model of not only the AXO-arm, but also of the human arm, this section will describe the design and implementation of a control system on the AXO-arm.

Control Strategy 11

This chapter accounts for the structure of the chosen controller. A brief study of the state of the art of exoskeleton control is done after which the controller structure is shown. In the following chapters, the different parts of the controller are designed.

Several control topologies for exoskeletons exist. In [42] a hierarchical structure is suggested, which splits exoskeleton control into three layers: task-level control, high-level control, and low-level control. The task-level controller concerns itself with the generation of reference trajectories when the user is performing a specific task. The high-level controller controls the interaction between the user and the exoskeleton, and the low-level controller is the system that directly interacts with the exoskeleton and controls the actuators to accomplish the orders of the high-level controller.

Two different approaches are presented in this project, both using a hierarchical structure. Since the muscle model that is described in **Chapter 9** outputs a joint torque, it seems logical to apply this directly to the motors, in a *Human Joint Torque Controller*. To remove as much of the physical load of the AXO-arm on the user, it is chosen to use feedback linearization for this controller, and simply apply the estimated joint torque as the input to this. This structure has two hierarchical layers, the muscle-estimator and the feedback linearization. It is implemented and tested in **Chapter 13**.

A controller that follows the method in [42] is also implemented and tested. In the case of exoskeletons for rehabilitation and training, where repetition of different predefined exercises is desired, [42] suggests a structure with a trajectory generator as the task-level controller, impedance control as the high-level controller and force or torque control as the low-level controller. This system would create a response that seeks to follow the trajectory demanded by the task-level controller, but allows the user to create some deviation using the impedance controller. Impedance control creates a force reference from a position error, and thus makes the system act as a spring that tries to pull the user towards the reference trajectory. The inner loop would be a force controller that makes the exoskeleton apply the force that is demanded by the high-level controller.

If, on the other hand, a trajectory is not known beforehand, [42] recommends a control scheme based on admittance control. In this configuration, the task-level controller is the user, the high-level controller is an admittance controller and the low-level controller is a position controller. Admittance control is, in a sense, the *inverse* of impedance control. Instead of generating a force reference from a position error, it generates a position trajectory from a force input. Admittance control is a popular choice in assistive exoskeleton control, such as in [43], [44], and [45]. The structure with position control as the low-level, admittance control as the high-level and the operator as the task-level is chosen for use in this project. Using the EMG-based model of joint torque from **Chapter 10**, the output of the operator becomes a torque. Since the motors are placed in the joints, it makes sense to keep the variables in joint space. The structure is shown in **Figure 11.1**.

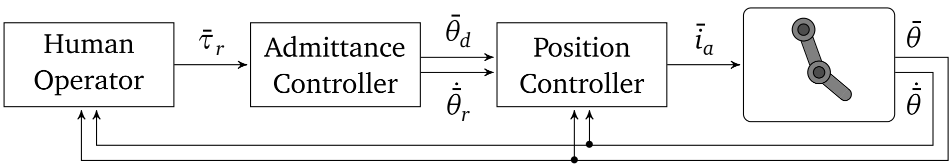


Figure 11.1: *The three levels of control on the AXO-arm controller.*

The three level control structure has been implemented on the UDOO Neo platform. This is a convenient choice because there is a powerful Arm Cortex M4 (200MHz) core that is capable of running software in real time. It also has an Arm Cortex A9 core (1GHz) that runs a Linux distribution.

The Linux system can interface with the Myo Band and log data for performance analysis.

It is chosen to implement the feedback linearization on the real time core, and the joint torque estimator and admittance controller on the Linux core. The distribution of the controller is shown in **Figure 11.2**.

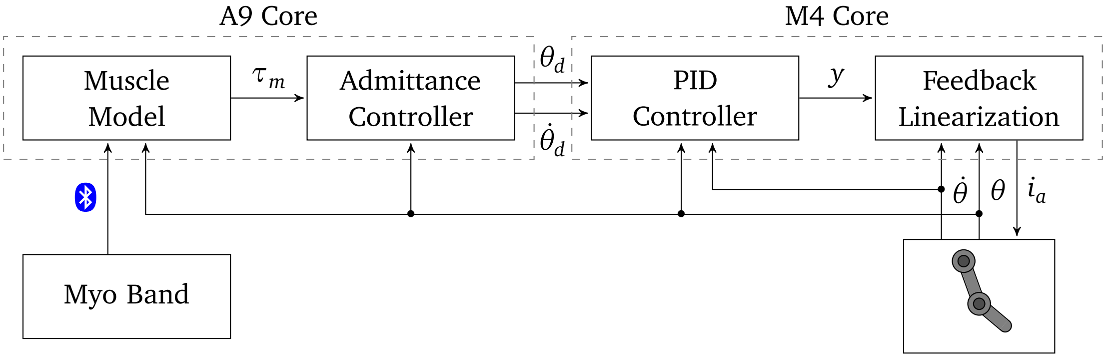


Figure 11.2: *The structure of the control loops implemented on the UDOO Neo.*

With the *Human Joint Torque Controller*, the software implementation is similar, but the signal τ_d is connected directly to y. A detailed description of the software implementation can be found in **Appendix E**.

The following chapter describes the design and test of the two suggested controllers from the inside out. First a position controller is designed as a computed torque PID controller with feedback linearization. After that, the controller that applies the torque from the muscle model directly to the motors is described in greater detail and tested. Finally, the admittance controller is designed to create angle and angular velocity references from the input torque to the system.

Position Control 12

In the previous chapter, the overall controller structure was described. In this chapter, the position control is designed, implemented, and verified, followed by the high-level controllers in the following chapters.

First, the position control strategy is presented and its stability is analyzed. After that follows a section on model uncertainties and how to handle them in the design, and finally the controller is given values and verified using a trajectory that mimics human movement.

12.1 Computed Torque Controller for Position Control

A standard method of implementing trajectory following on a serial manipulator is *computed torque control* [21][22]. It is a model-based control scheme that relies heavily on feedback linearization. The feedback linearized system is shown in **Figure 12.1**.

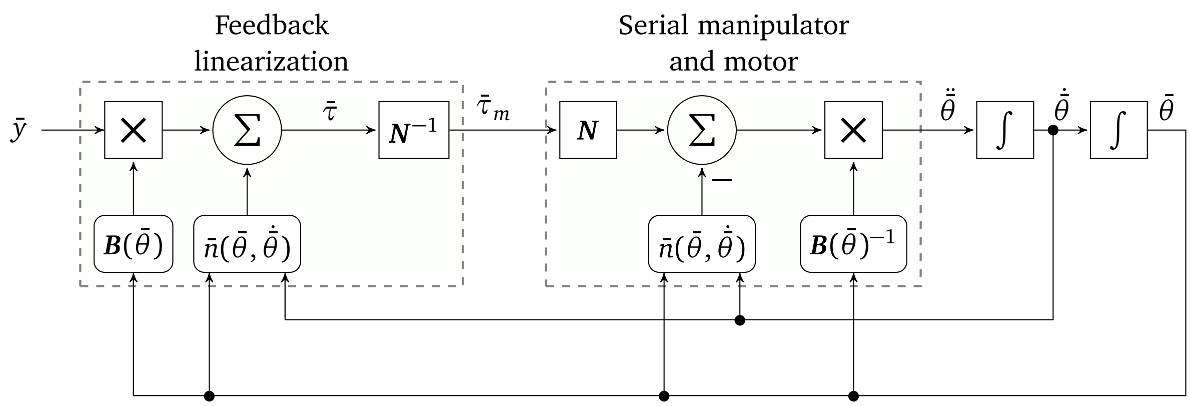


Figure 12.1

The main idea is to apply an input that cancels all the nonlinear dynamics of the system such that it becomes linear and thus easier to control using well known solutions. Recall that the dynamics of the entire serial manipulator are described by,

$$\ddot{\bar{\theta}} = \boldsymbol{B}(\bar{\theta})^{-1}(\bar{\tau} - \bar{n}(\bar{\theta}, \dot{\bar{\theta}})) \qquad [\mathrm{rad/s^2}] \qquad (12.1)$$

$$\boldsymbol{B}(\bar{\theta}) = \boldsymbol{I}_m \boldsymbol{N}^2 + \boldsymbol{M}(\bar{\theta}) \qquad [\mathrm{N\,m}] \qquad (12.2)$$

$$\bar{n}(\bar{\theta}, \dot{\bar{\theta}}) = V(\bar{\theta}, \dot{\bar{\theta}}) + \bar{G}(\bar{\theta}) + \bar{F}(\dot{\bar{\theta}} \cdot \boldsymbol{N})\,\boldsymbol{N} \qquad [\mathrm{N\,m}] \qquad (12.3)$$

which is also stated in **Equation (5.42)**. Using the control law,

$$\bar{\tau} = \boldsymbol{B}(\bar{\theta})\bar{y} + \bar{n}(\bar{\theta}, \dot{\bar{\theta}}) \qquad [\mathrm{N\,m}] \qquad (12.4)$$

yields a system that from the reference, $\bar{y}$, to joint position, $\bar{\theta}$, is merely a double integration so $\bar{y} = \ddot{\bar{\theta}}$. Obviously this is only the case when all the dynamics of the system is modeled perfectly.

Computed torque does not only contain feedback linearization. A double integrator in itself is only marginally stable. Computed torque control uses PD-control to track position and velocity trajectories. If we apply PD-control on the joint positions of the AXO-arm, a diagram of the resulting controller is shown in **Figure 12.2**, where $\bar{\theta}_d, \dot{\bar{\theta}}_d, \ddot{\bar{\theta}}_d$ make up the desired trajectory and $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ are symmetric positive definite matrices. [46]

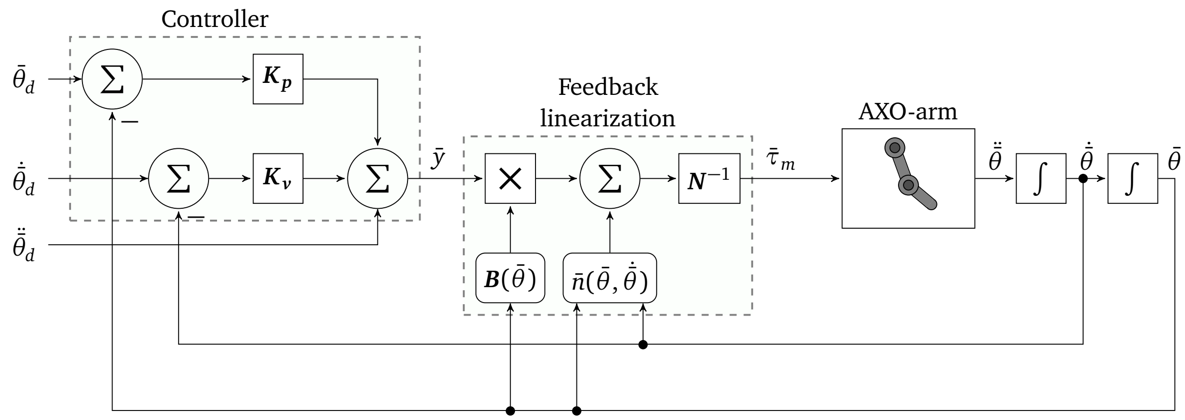


Figure 12.2: *Computed torque control of the AXO-arm. The section from* ***Figure 12.1*** *called* Serial manipulator and motor *is for simplicity reduced to the block called* AXO-arm.

The closed-loop form of the computed torque system is defined as

$$\bar{y} = \boldsymbol{K}_p\,(\bar{\theta}_d - \bar{\theta}) + \boldsymbol{K}_v\,(\dot{\bar{\theta}}_d - \dot{\bar{\theta}}) + \ddot{\bar{\theta}}_d \qquad \left[\mathrm{rad/s^2}\right] \tag{12.5}$$

Due to the feedback linearization, the dynamics from $\bar{y}$ to $\ddot{\bar{\theta}}$ correspond to a gain of one, so $\bar{y} = \ddot{\bar{\theta}}$. We can therefore write

$$0 = \boldsymbol{K}_p\,(\bar{\theta}_d - \bar{\theta}) + \boldsymbol{K}_v\,(\dot{\bar{\theta}}_d - \dot{\bar{\theta}}) + (\ddot{\bar{\theta}}_d - \ddot{\bar{\theta}}) \qquad \left[\mathrm{rad/s^2}\right] \tag{12.6}$$

If we define the angle tracking error, e, as

$$\bar{e} = \bar{\theta}_d - \bar{\theta} \qquad [\mathrm{rad}] \tag{12.7}$$

$$\dot{\bar{e}} = \dot{\bar{\theta}}_d - \dot{\bar{\theta}} \qquad [\mathrm{rad/s}] \tag{12.8}$$

$$\ddot{\bar{e}} = \ddot{\bar{\theta}}_d - \ddot{\bar{\theta}} \qquad \left[\mathrm{rad/s^2}\right] \tag{12.9}$$

then the closed-loop system becomes,

$$0 = \boldsymbol{K}_p \bar{e} + \boldsymbol{K}_v \dot{\bar{e}} + \ddot{\bar{e}} \qquad \left[\mathrm{rad/s^2}\right] \tag{12.10}$$

$$\begin{bmatrix} \dot{\bar{e}} \\ \ddot{\bar{e}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -\boldsymbol{K}_p & -\boldsymbol{K}_v \end{bmatrix} \begin{bmatrix} \bar{e} \\ \dot{\bar{e}} \end{bmatrix} \tag{12.11}$$

where the poles, or eigenvalues, of the system can be found from the roots of the characteristic equation:

$$\det\left(\begin{bmatrix} 0 & I \\ -\boldsymbol{K}_p & -\boldsymbol{K}_v \end{bmatrix} - \lambda I\right) = \lambda^2 + \lambda \boldsymbol{K}_v + \boldsymbol{K}_p \qquad [\cdot] \tag{12.12}$$

$$= 0 \tag{12.13}$$

It is seen that by choosing $\boldsymbol{K}_p$ and $\boldsymbol{K}_v$ positive definite, the poles of the system will be in the left half of the complex plane resulting in a stable system.

12.1.1 Uncertainty of the Model

The computed torque scheme explained above, builds upon the assumption of the AXO-arm being perfectly modeled. However the model is of course not perfect, which can also be seen in **Chapter 7**. Therefore, the uncertainties of the model have to be taken into account to guarantee that the AXO-arm position control works as intended.

The model is thus expanded to also include uncertainties:

$$\hat{\boldsymbol{B}}(\bar{\theta}) = \tilde{\boldsymbol{B}}(\bar{\theta}) + \boldsymbol{B}(\bar{\theta}) \qquad \left[\mathrm{kg/m^2}\right] \qquad (12.14)$$

$$\hat{\bar{n}}(\bar{\theta},\dot{\bar{\theta}}) = \tilde{\bar{n}}(\bar{\theta},\dot{\bar{\theta}}) + \bar{n}(\bar{\theta},\dot{\bar{\theta}}) \qquad [\mathrm{N\,m}] \qquad (12.15)$$

Where elements with ˆ are parts of the estimated model used in the feedback linearization and elements with ˜ are the deviations from the true model. The feedback linearization control law including uncertainties is formulated as:

$$\bar{\tau} = \hat{\boldsymbol{B}}(\bar{\theta})\bar{y} + \hat{\bar{n}}(\bar{\theta},\dot{\bar{\theta}}) \qquad [\mathrm{N\,m}] \qquad (12.16)$$

Next the model equation given in **Equation (12.1)** is combined with the uncertain feedback linearization:

$$\ddot{\bar{\theta}} = \boldsymbol{B}\left(\bar{\theta}\right)^{-1}(\hat{\boldsymbol{B}}(\bar{\theta})\bar{y} + \hat{\bar{n}}(\bar{\theta},\dot{\bar{\theta}}) - n(\bar{\theta},\dot{\bar{\theta}})\Big) \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.17)$$

$$\ddot{\bar{\theta}} = \boldsymbol{B}(\bar{\theta})^{-1}\hat{\boldsymbol{B}}(\bar{\theta})\bar{y} + \boldsymbol{B}(\bar{\theta})^{-1}\tilde{\bar{n}}(\bar{\theta},\dot{\bar{\theta}}) \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.18)$$

$$\ddot{\bar{\theta}} = \bar{y} + \left(\boldsymbol{B}(\bar{\theta})^{-1}\hat{\boldsymbol{B}}(\bar{\theta}) - \boldsymbol{I}\right)\bar{y} + \boldsymbol{B}(\bar{\theta})^{-1}\tilde{\bar{n}}(\bar{\theta},\dot{\bar{\theta}}) \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.19)$$

$$\ddot{\bar{\theta}} = \bar{y} - \bar{\psi}(\bar{\theta},\dot{\bar{\theta}}) \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.20)$$

The uncertainty of the system is expressed by $\bar{\psi}(\bar{\theta},\dot{\bar{\theta}}) = (\boldsymbol{I} - \boldsymbol{B}(\bar{\theta})^{-1}\hat{\boldsymbol{B}}(\bar{\theta}))\bar{y} - \boldsymbol{B}(\bar{\theta})^{-1}\tilde{\bar{n}}(\bar{\theta},\dot{\bar{\theta}})$. By inserting the control law given in **Equation (12.5)**, a new system equation is derived:

$$\ddot{\bar{\theta}} = \boldsymbol{K_p}\,(\bar{\theta}_d - \bar{\theta}) + \boldsymbol{K_v}\,(\dot{\bar{\theta}}_d - \dot{\bar{\theta}}) + \ddot{\bar{\theta}}_d - \bar{\psi}(\bar{\theta},\dot{\bar{\theta}}) \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.21)$$

$$\bar{\psi}(\bar{\theta},\dot{\bar{\theta}}) = \boldsymbol{K_p}\,(\bar{\theta}_d - \bar{\theta}) + \boldsymbol{K_v}\,(\dot{\bar{\theta}}_d - \dot{\bar{\theta}}) + \ddot{\bar{\theta}}_d - \ddot{\bar{\theta}} \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.22)$$

$$\bar{\psi}(\bar{\theta},\dot{\bar{\theta}}) = \boldsymbol{K_p}\bar{e} + \boldsymbol{K_v}\dot{\bar{e}} + \ddot{\bar{e}} \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.23)$$

It is seen that while the ideal system in **Equation (12.6)** is not influenced by uncertainties, the system in **Equation (12.23)** is directly influenced by the uncertainty $\bar{\psi}(\bar{\theta},\dot{\bar{\theta}})$.

To check how the computed torque controller handles this uncertainty, the steady state error is studied. The uncertainty behaves non-linearly and is dependent on $\bar{\theta}$ and $\dot{\bar{\theta}}$, but for the next argument assume that the AXO-arm has settled and the uncertainty therefore is constant (since $\bar{\theta}$ and $\dot{\bar{\theta}}$ are fixed). The system in **Equation (12.23)** is first Laplace transformed:

$$\frac{\bar{e}(s)}{\bar{\psi}(s)} = \frac{1}{\boldsymbol{K_p} + \boldsymbol{K_v}s + s^2} \qquad [\cdot] \qquad (12.24)$$

When the uncertainty is constant it can be seen as a step input to **Equation (12.24)**. A step has the transfer function $\frac{\bar{\psi}}{s}$. By the final value theorem, the steady state error is then:

$$\lim_{t\to\infty} \bar{e}(t) = \lim_{s\to 0}\, s\; \bar{e}(s) = \lim_{s\to 0}\, \frac{\bar{\psi}}{s}\, \frac{s}{\boldsymbol{K_p} + \boldsymbol{K_v}s + s^2} = \frac{\bar{\psi}}{\boldsymbol{K_p}} \qquad [\mathrm{rad}] \qquad (12.25)$$

As seen, the computed torque controller suffers from a constant steady state position error, under the assumption of a constant uncertainty. This error can be made arbitrarily small by using a large proportional gain in the controller, but at some point, the actuator will be saturated. During movement of the AXO-arm, the uncertainty is not constant but dynamically changing with the system and therefore the position error is only expected to be worse under this condition.

With the system and its uncertainty analyzed, next appropriate gains for the controller are chosen and the system should is tested to study how the uncertainty affects the system. Before that, however, it is relevant to study which references are most suitable for the system, when its performance is evaluated.

12.2 Reference Trajectory

A common method for quickly showing performance of a feedback controller is its ability to track a step input. However, a step response is in this case not very meaningful as the purpose of the controller is to follow smooth trajectories, and it is therefore expected that the system error is always small. The performance of the controller will thus be evaluated by looking at how it tracks a trajectory that mimics human movement. The findings of [47] show that planar movement of the human arm follows bell shaped velocity trajectories. They estimate with good accuracy the trajectory of human motion using only knowledge of the start position, end position, and the duration of the movement. They do this by minimizing the L_2 norm of the jerk, the derivative of acceleration, over the trajectory. An expansion on the work of [47] is the method presented in [48]. They propose a bang-bang control approach to the jerk reference, in an attempt at minimizing the L_∞ norm of the jerk instead of the L_2 norm. Their model is not only simpler than that of [47] but it also predicts the trajectory of the human arm with greater accuracy.

The test trajectory chosen for visualizing and evaluating controller performance in this chapter is position and velocity references generated using a bang-bang jerk reference. Specifically, we choose a position reference signal that starts in 0 rad and ends in 1.57 rad with a duration of 2 s. According to [48] the required jerk reference is

$$\dddot{\theta}(t) = \begin{cases} J & 0 \le t \le \frac{T}{4} \\ -J & \frac{T}{4} \le t \le \frac{3T}{4} \\ J & \frac{3T}{4} \le t \le T \end{cases} \tag{12.26}$$

where J is the magnitude of a square jerk reference and T is the time duration of the trajectory. The magnitude of the jerk is found using the initial and final position and the duration of the trajectory.

$$J = 32\,\frac{\theta_f - \theta_i}{T^3} \tag{12.27}$$

The position, velocity, acceleration and jerk of the reference trajectory is shown in **Figure 12.3**. And this reference trajectory will be used to evaluate the performance of the controller. Not only does it mimic human behavior better than an step response, but it also allows larger control gains without saturation on the input signal because there are no jumps in the position and velocity reference signals.

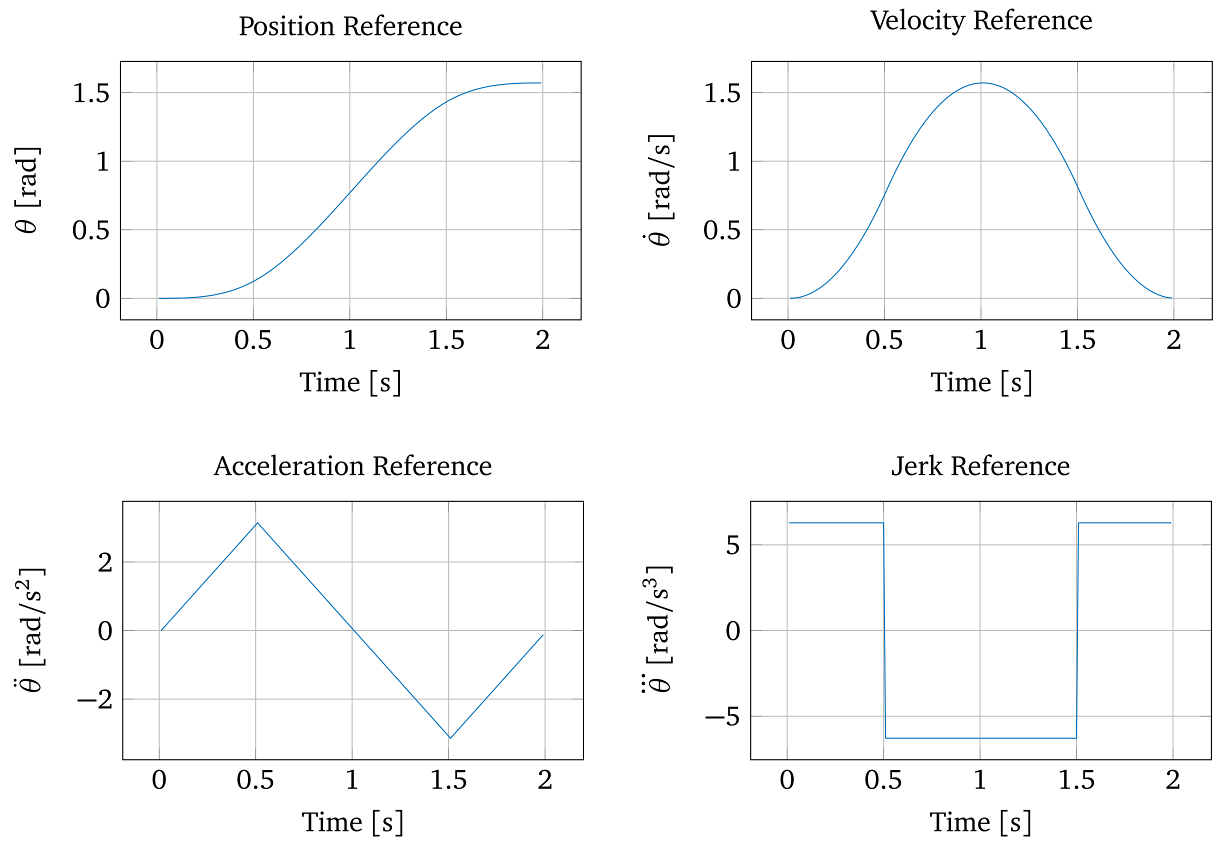


Figure 12.3: *The position and velocity reference for evaluating controller performance. The references for position, velocity, and acceleration are generated by integrating the jerk reference signal.*

12.2.1 Choice of Computed Torque Controller Gains

With the trajectories found, next, to check if the model uncertainty is actually affecting the real system, the controller gains $\boldsymbol{K_p}$ and $\boldsymbol{K_v}$ are chosen. The performance is evaluated by its ability to track the minimum jerk trajectory described in **Section** 12.2. With these measurements, the steady state errors are studied to see the impact of the real uncertainties and to clarify if any controller strategy improvements are necessary.

For choosing the controller gains, the characteristic equation shown in **Equation (**12.12**)**, is studied. It can be compared with the general form:

$$\lambda^2 + \lambda \boldsymbol{K_v} + \boldsymbol{K_p} = \omega_n^2 + 2\,\zeta\,\omega_n \lambda + \lambda^2 \qquad [\cdot] \qquad (12.28)$$

By choosing $\boldsymbol{K_p} = \omega_n^2$ and $\boldsymbol{K_v} = 2\,\zeta\,\omega_n$ the equations are similar. The natural frequency is ω_n and ζ is the damping factor. With $\zeta > 1$ the system becomes overdamped, which means that it settles slowly but with no oscillations or overshoot. The opposite case is by choosing $\zeta < 1$ (underdamped); here the system has complex conjugate poles which induce oscillations but a fast settling time. The compromise between over and underdamped is to choose $\zeta = 1$, which makes the system critically damped and thereby induce no oscillations and a faster settling time compared to the overdamped system. [49]

Oscillations are not wanted in the configuration but a fast settling time is also attractive, which is why a damping factor of 1 is chosen. This choice changes **Equation (**12.28**)**, to:

$$\lambda^2 + \lambda \boldsymbol{K_v} + \boldsymbol{K_p} = \omega_n^2 + 2\,\omega_n s + s^2 \qquad [\cdot] \qquad (12.29)$$

If all elements of **Equation (12.29)** are kept positive (and therefore also $\boldsymbol{K_p}$ and $\boldsymbol{K_v}$), the poles of the system are always located in the left half plane. An expression of the poles are:

$$\omega_n^2 + 2\ \omega_n s + s^2 = (s + \omega_n)^2 \qquad [\cdot] \qquad (12.30)$$

With $\boldsymbol{K_p} = \omega_n^2$ and $\boldsymbol{K_v} = 2\ \omega_n$ a relation between the gains are:

$$\boldsymbol{K_v} = 2\ \sqrt{\boldsymbol{K_p}} \qquad [\cdot] \qquad (12.31)$$

To summarize; by choosing the controller gains positive and by using the relation **Equation (12.31)**, the system is stable and critically damped. Since it is found in **Section 3.2.1** that the maximum voluntary movement frequency should be at least 6 Hz, an immediate choice of gains would be some that allow the system to follow this 6 Hz movement. This however caused the system to behave too violently. Experimentally some more appropriate gains were found, which unfortunately cannot meet the requirement in **Section 3.2.1**:

	K_p	K_v	ω_n
Shoulder	100	20	10
Elbow	100	20	10

Table 12.1

Here the natural frequency corresponding to the found gains, is also given. The natural frequency of 10 corresponds to a bandwidth of $\frac{10}{2\pi} = 1.6$ Hz. With the chosen controller gains, the measurements shown in **Figure 12.6** were obtained. Since the position references were 1.57 rad for both shoulder and elbow, **Figure 12.6** clearly shows that steady state errors are present. The system can therefore be improved.

12.2.2 PID Computed Torque Control

To remove the steady state error of the position control, PID control is implemented. In this section, it is therefore studied how this controller handles the uncertainty of the AXO-arm model. Additionally, for choosing controller gains and testing the system, the characteristic equation of the system is found. With the integral part, the control law changes to:

$$\bar{y} = \boldsymbol{K_i} \int (\bar{\theta}_d - \bar{\theta})\ \mathrm{d}t + \boldsymbol{K_p}\ (\bar{\theta}_d - \bar{\theta}) + \boldsymbol{K_v}\ (\dot{\bar{\theta}}_d - \dot{\bar{\theta}}) + \ddot{\bar{\theta}}_d \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.32)$$

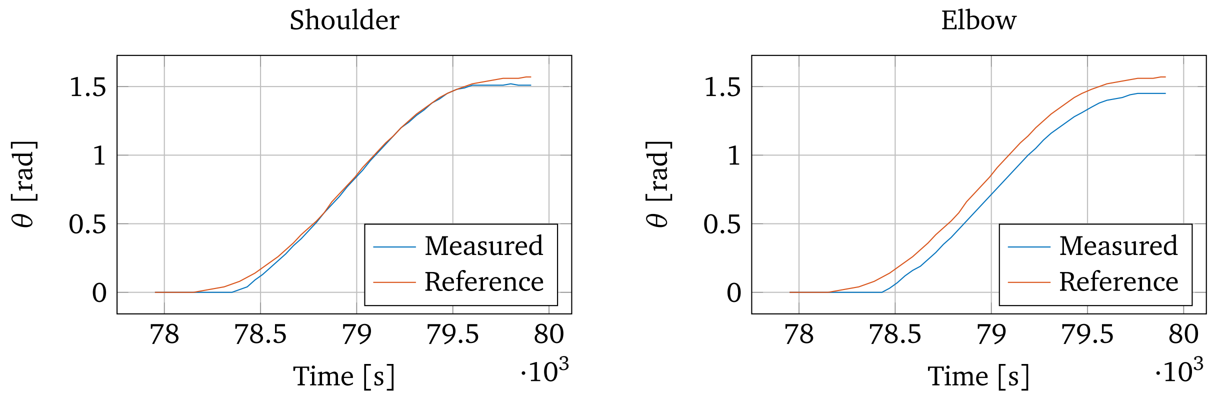


Figure 12.4: *Both joints experience steady state error on the position control.*

This control law combined with the model and feedback linearization with uncertainty given in **Equation (12.20)**, gives the following closed loop system:

$$\ddot{\theta} = \boldsymbol{K_i} \int (\bar{\theta}_d - \bar{\theta})\,\mathrm{d}t + \boldsymbol{K_p}\,(\bar{\theta}_d - \bar{\theta}) + \boldsymbol{K_v}\,(\dot{\bar{\theta}}_d - \dot{\bar{\theta}}) + \ddot{\bar{\theta}}_d - \bar{\psi}(\bar{\theta}, \dot{\bar{\theta}}) \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.33)$$

$$\bar{\psi}(\bar{\theta}, \dot{\bar{\theta}}) = \boldsymbol{K_i} \int (\bar{\theta}_d - \bar{\theta})\,\mathrm{d}t + \boldsymbol{K_p}\,(\bar{\theta}_d - \bar{\theta}) + \boldsymbol{K_v}\,(\dot{\bar{\theta}}_d - \dot{\bar{\theta}}) + \ddot{\bar{\theta}}_d - \ddot{\bar{\theta}} \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.34)$$

$$\bar{\psi}(\bar{\theta}, \dot{\bar{\theta}}) = \boldsymbol{K_i} \int \bar{e}\,\mathrm{d}t + \boldsymbol{K_p}\bar{e} + \boldsymbol{K_v}\dot{\bar{e}} + \ddot{\bar{e}} \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.35)$$

As seen, the system is identical to the computed torque scheme with PD controller (**Equation (12.23)**), with exception of the integral part. By studying the steady state error, it is proven that this controller removes the steady state position error caused by an uncertainty. The system is first Laplace transformed:

$$\frac{\bar{e}(s)}{\bar{\psi}(\bar{\theta}(s), \dot{\bar{\theta}}(s))} = \frac{s}{\boldsymbol{K_i} + \boldsymbol{K_p}s + \boldsymbol{K_v}s^2 + s^3} \qquad \left[\mathrm{s^2}\right] \qquad (12.36)$$

Next the steady state position error is found with the assumption of a constant uncertainty, $\frac{\bar{\psi}}{s}$:

$$\lim_{t\to\infty} \bar{e}(t) = \lim_{s\to 0}\, s\,\bar{e}(s) = \lim_{s\to 0} \frac{\bar{\psi}}{s} \frac{s^2}{\boldsymbol{K_i} + \boldsymbol{K_p}s + \boldsymbol{K_v}s^2 + s^3} = 0 \qquad [\mathrm{rad}] \qquad (12.37)$$

As seen, the steady state error is as expected 0. To check the controller in practice, gains are selected, and with measurements the PID controller is verified.

The system characteristic equation is also found by use of the control law given in **Equation (12.32)**. This time however, when combining the model and control law, the uncertainty is not considered. This results in the system equation:

$$0 = \boldsymbol{K_i} \int \bar{e}\,\mathrm{d}t + \boldsymbol{K_p}\bar{e} + \boldsymbol{K_v}\dot{\bar{e}} + \ddot{\bar{e}} \qquad \left[\mathrm{rad/s^2}\right] \qquad (12.38)$$

$$\begin{vmatrix} \dot{\bar{\epsilon}} \\ \dot{\bar{e}} \\ \ddot{\bar{e}} \end{vmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -\boldsymbol{K_i} & -\boldsymbol{K_p} & -\boldsymbol{K_v} \end{bmatrix} \begin{bmatrix} \bar{\epsilon} \\ \bar{e} \\ \dot{\bar{e}} \end{bmatrix} \qquad (12.39)$$

Here $\bar{\epsilon}$ is the integrated position error. The eigenvalues of the system is found by:

$$\det\left(\begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ -\boldsymbol{K_i} & -\boldsymbol{K_p} & -\boldsymbol{K_v} \end{bmatrix} - \lambda I\right) = \lambda^3 + \lambda^2\boldsymbol{K_v} + \lambda\boldsymbol{K_p} + \boldsymbol{K_i} = 0 \qquad [\cdot] \qquad (12.40)$$

The roots of **Equation (12.40)** are the poles of the system and can be compared with the general form [17]:

$$(s + \Gamma\,\omega_n)(\omega_n^2 + 2\,\xi\,\omega_n s + s^2) = \Gamma\,\omega_n^3 + (1 + 2\,\Gamma\,\xi)\omega_n^2\,s + (\Gamma + 2\,\xi)\omega_n\,s^2 + s^3 \qquad [\cdot] \qquad (12.41)$$

By choosing $\boldsymbol{K_i} = \Gamma\,\omega_n^3$, $\boldsymbol{K_p} = (1 + 2\,\Gamma\,\xi)\omega_n^2$ and $\boldsymbol{K_v} = (\Gamma + 2\,\xi)\omega_n$, the **Equation (12.41)** and **Equation (12.40)** are similar. Again by choosing the gains to be positive and the damping factor to be $\xi = 1$, the system is stable and overshoot is avoided since all the system poles become real. The system denominator changes to:

$$(s + \Gamma\ \omega_n)(\omega_n^2 + 2\ \omega_n s + s^2) = (s + \Gamma\ \omega_n)(\omega_n + s)^2 \qquad [\cdot] \qquad (12.42)$$

By use of **Equation (12.42)** a relation between the gains is formulated as:

$$\boldsymbol{K_i} = \Gamma\ \omega_n^3 \qquad [\cdot] \qquad (12.43)$$

$$\boldsymbol{K_v} = (\Gamma + 2)\left(\frac{\boldsymbol{K_i}}{\Gamma}\right)^{\frac{1}{3}} \qquad [\cdot] \qquad (12.44)$$

$$\boldsymbol{K_p} = (1 + 2\ \Gamma)\left(\frac{\boldsymbol{K_i}}{\Gamma}\right)^{\frac{2}{3}} \qquad [\cdot] \qquad (12.45)$$

By choosing $\Gamma = 1$, and following the requirement from **Chapter 3**, ω_n should have a bandwidth of $6 \cdot 2\pi = 37.7\,\mathrm{rad/s}$. This choice however result in very high gains which causes the system to behave too violently. Instead the controller gains in **Table 12.2** are found experimentally, which unfortunately do not fulfill the bandwidth requirement from **Chapter 3**.

	K_p	K_v	K_i	ω_n	Γ
Shoulder	108	18	216	6	1
Elbow	108	18	216	6	1

Table 12.2

The Bode plot of the feedback linearized system with PID control is shown in **Figure 12.5**. Since the controller is the same for both feedback linearized joints, the Bode plots of the joints are identical. We thus only show one. The gain margin is infinite, and the phase margin is around 135°, more than meeting the stability requirements of **Chapter 3**. The PID controller is implemented and its minimum jerk trajectory following capabilities are tested and shown on **Figure 12.6**. The performance is satisfactory, even though the bandwidth requirements are not met.

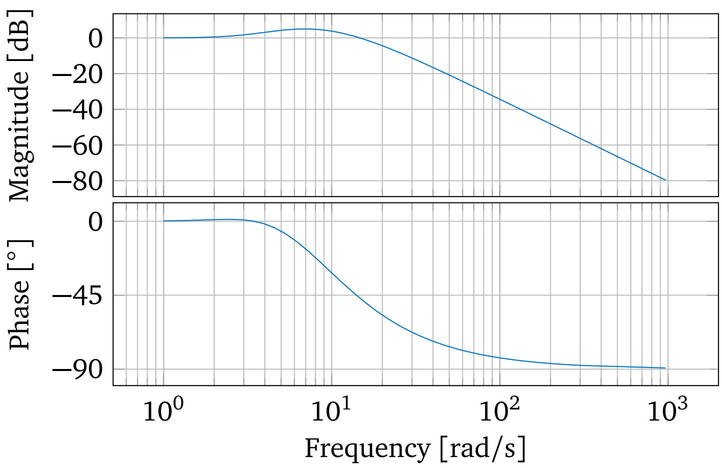


Figure 12.5: *Bode plot of the PID controlled and feedback linearized joints.*

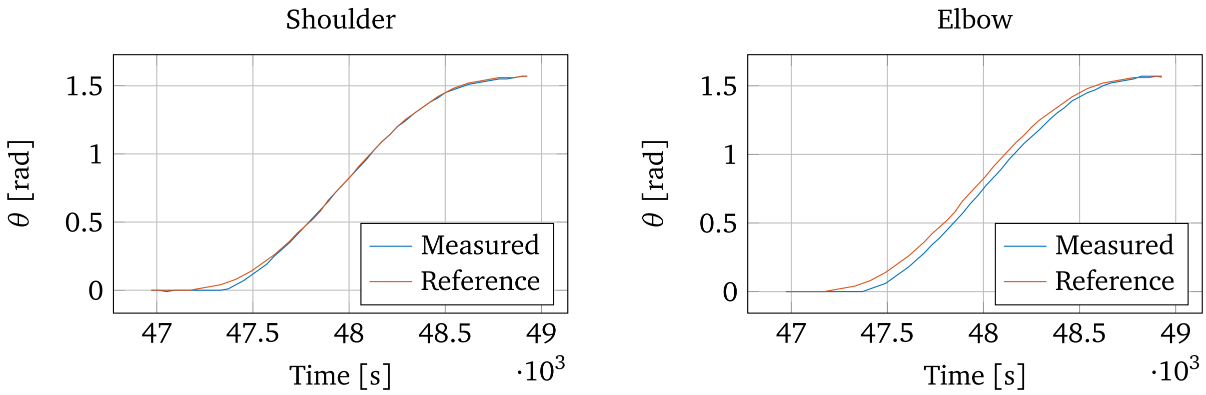


Figure 12.6: *With PID control, the error is guaranteed to go to zero, but it is at the cost of a slower settling time.*

13 Human Joint Torque Control

In **Chapter 9** a method of estimating the joint torque of the user is presented. The quality of this estimate is deemed good enough for use in exoskeleton control. A logical approach for exoskeleton control would be to apply this reference torque directly on motors of the feedback linearized AXO-arm. The proposed controller structure is shown in **Figure 13.1** where K_τ is a gain that allows tuning the controller. Before the proposed controller can be tested, some considerations are made with regard to the criteria for success in the tests.

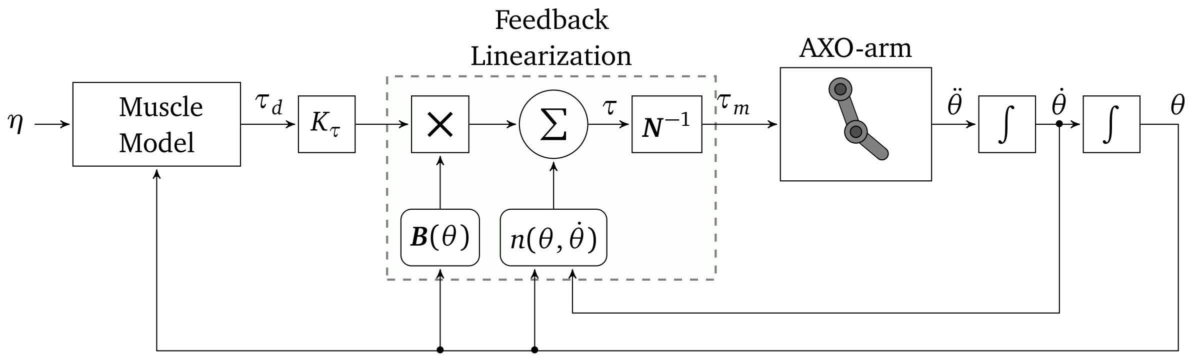


Figure 13.1: *The structure of the proposed feedback linearized torque control of the ELBOW joint of the AXO-arm*

13.1 Controller Testing Considerations

Testing the AXO-arm is a subjective process where the user is describing how the motion feels. Key aspects of the test are responsiveness, ease of control, and resistance.

Responsiveness is the measure of how quickly the AXO-arm responds to the user's deliberate motions. The user will feel it if the movement of the arm is delayed compared to his intention.

Ease of control is a subjective validation of the *correctness* of the AXO-arm's movement. Does it start and stop when intended? Or is it forcing involuntary movement of the user? Does the user feel as if he is in control or not?

Resistance relates to how the motions feel. Is the exoskeleton imposing extra resistance when the user is moving, or is it supporting the motion?

The controller is tested by having the user perform a controlled flexion-extension movement where the elbow joint is moved in the sequence 0-45-90-45-0°. Naturally, since the user is not a machine, we don't expect the angle to hit exactly 45° and 90°, but they are guidelines for the user. The measured position and velocity trajectories of such tests should match the way the user moves without the AXO-arm attached. A measurement of the elbow joint angle, measured using a potentiometer, is shown in **Figure 13.2**. Note that the user in this case makes no overshoot and settles on a 45° step in around 1 s. The position signal is s-shaped, as it is expected from **Figure 12.3**. There is significant steady state error, but that is due to the user not knowing the exact angle of his joint.

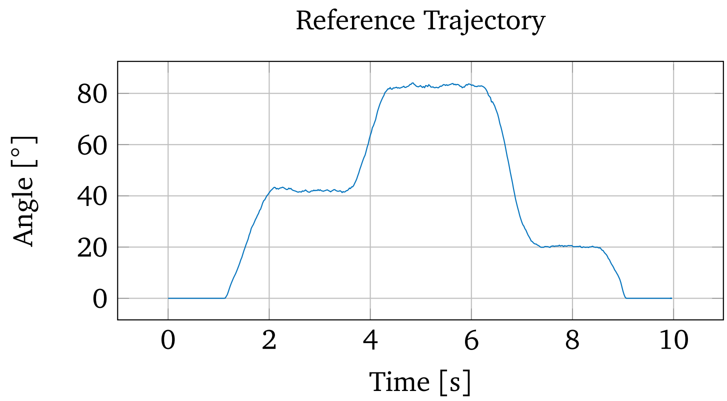


Figure 13.2: *Elbow joint motion without the exo-skeleton attached.*

13.2 Testing the Torque Controller

The torque controller has been implemented and tested on the AXO-arm. The controller K_τ was tuned to feel as comfortable as possible for the user. A gain of $K_\tau = 2$ proved to give the fastest possible response without being too fast and uncontrollable for the user. The position and velocity responses of an attempted motion like the one shown in **Figure 13.2**, as well as the estimated human joint torque are shown in **Figure 13.3**.

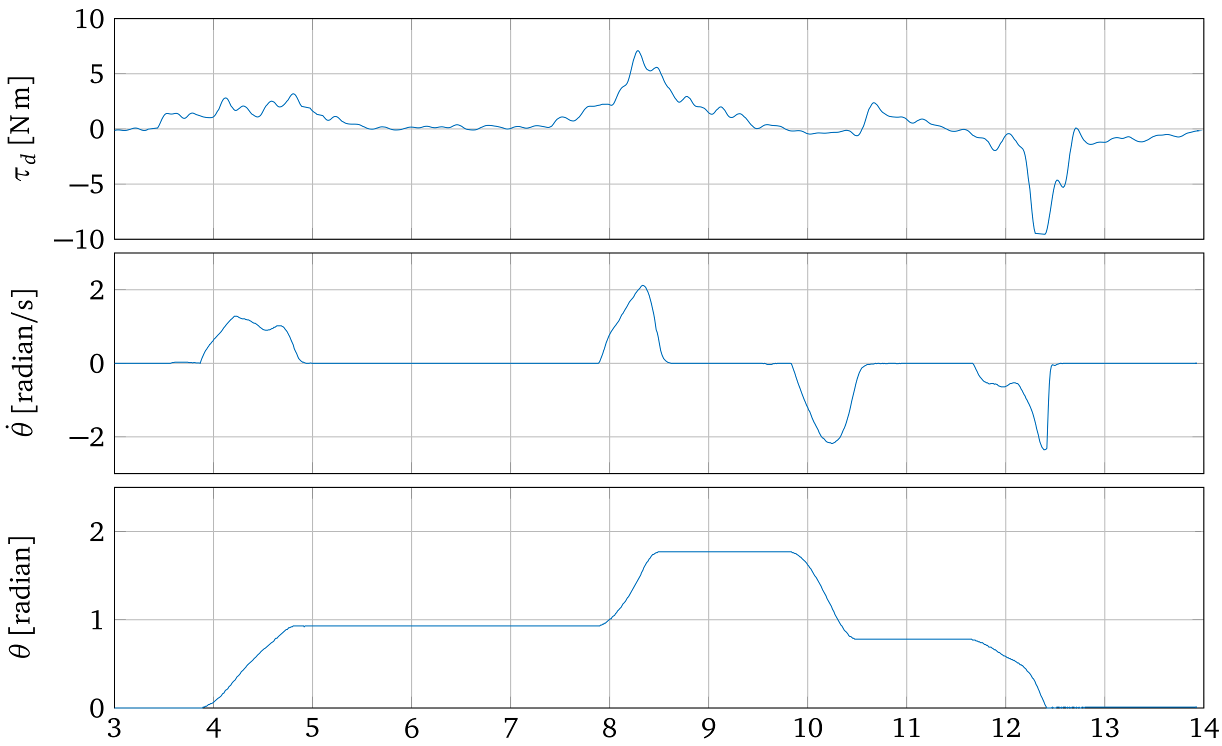


Figure 13.3: *The position and velocity responses using the torque controller $K_\tau = 2$. There is significant delay between the applied torque and the velocity and position response. The plots share the same time axis.*

The measurements show that there is a significant delay between the applied torque reference and the responses in velocity and position, visible at both 3.5 s and 7.5 s. So while the motion is quite well controlled, this delay is uncomfortable and it would vastly improve the user experience if it was reduced. To reduce the delay, admittance control is suggested. It is a popular choice in the literature on assisting robotics in general and exoskeletons in particular [43][50][44]. The admittance controller generates velocity- and, in some configurations, position-references from the measured interaction force between the user and the exoskeleton [43]. The effect of the admittance controller is thus comparable to that of a cascade controller, as the admittance controller requires inner loops for position and velocity tracking. The following chapter describes the design, implementation and test of admittance control.

Admittance Control 14

14.1 Admittance Control

The position control scheme described in **Section 12.1** and **Section ??** has for each serial manipulator joint the angular position, angular velocity and angular acceleration as references. While the position control aims at tracking these references, the references themselves have to represent human arm movement caused by a torque applied in the joints. This means that the references follow a trajectory that is dependent on the torque output from the muscle model described in **Chapter 9**. As described in the papers, [43], [50] and [44], admittance control is a good approach to generate references from an input force or torque. In all three papers, the admittance controller generates a velocity reference from a torque input. In [43] the velocity reference is integrated to generate a position reference in addition to the velocity reference. In [44] several different structures of the admittance controller is suggested, but the actual structure of the controller is not clear.

The literature agrees on the outer controller structure, namely that the interaction force is the input for the admittance controller while the angular velocity reference is the output. In this project, the human joint torque is used as the input to the admittance controller, since, if the human arm and the exoskeleton are perfectly coupled, that would be the interaction force.

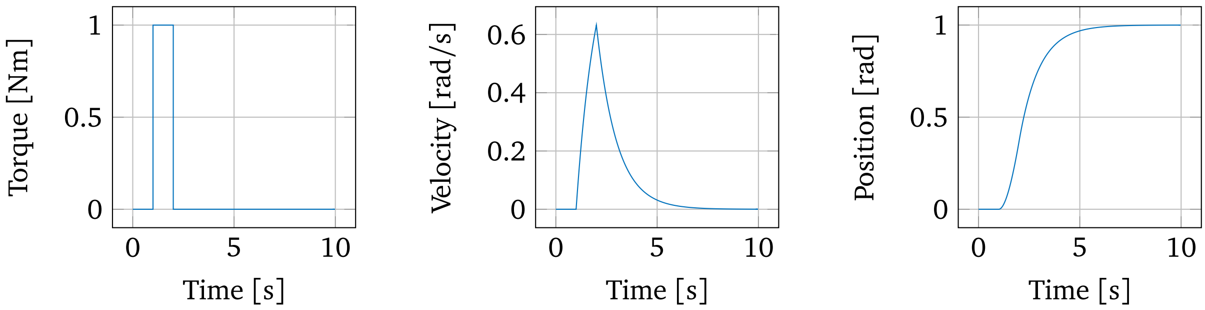
The clear difference between the papers lies in the internal controller structure of the admittance control and therefore the transfer function $A(s) = \frac{\dot{\theta}_d}{\tau}$. It was claimed in [44] that they used admittance control, but the actual implementation is not clear. The methods of [43] and [50] are briefly described below, along with their strengths and weaknesses.

14.1.1 Low-Pass Admittance

In [50] a controller with the transfer function shown in **Equation (14.1)**, is suggested. Here M represents the inertia of the human arm, while B is the friction.

$$A_1(s) = \frac{\dot{\theta}_d}{\tau} = \frac{1}{M\,s + B} \qquad [\mathrm{rad/(N\,m\,s)}] \tag{14.1}$$

Figure 14.1 shows the admittance filters response to a torque pulse, where the input amplitude and controller parameters are all chosen to be 1. The velocity reference rises, but converges back to zero. This admittance filter thus acts as a low-pass filter on the torque reference. It also means that the position of the AXO-arm remains at a constant position after the torque pulse is finished.



(a) *Torque pulse*

(b) *Velocity response to the torque pulse*

(c) *Position response to the torque pulse*

Figure 14.1

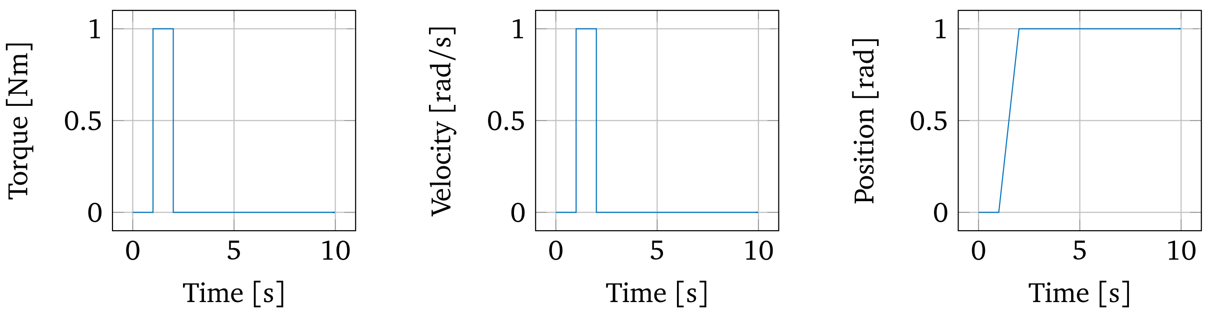
This controller compensates for gravity, since the wearer does not have to produce a constant torque to stay at a certain position. Reaching a specific position can still be a problem, but it might only be a matter of tuning the controller parameters to the user. Since this controller structure does not include the generation of a position reference, the motion controller developed in **Section** 12.2.2 needs to be modified to be pure velocity control while still meeting the bandwidth requirements of **Chapter** 3.

14.1.2 P Admittance

In [43] a simpler approach is presented. They suggest using a constant gain from joint torque to velocity reference.

$$A_2(s) = \frac{\dot{\theta}_d}{\tau} = K_c \qquad [\mathrm{rad/(N\,m\,s)}] \qquad (14.2)$$

When simulating the transfer function, the results shown in **Figure** 14.2 are obtained. Again the input amplitude and controller parameter are both chosen to be 1.



(a) *Torque pulse*

(b) *Velocity response to the torque pulse*

(c) *Position response to the torque pulse*

Figure 14.2

The results are similar to those shown in **Figure** 14.1 and therefore has most of the same pros and cons. This controller is more easily implemented, but might also introduce some discomfort since the transition from movement to standing still is very abrupt.

14.2 Choice of Admittance Control Method

The admittance controller topologies of **Section 14.1** are implemented and tested on the elbow joint of the AXO-arm. This section presents a review of the different methods, backed with measurements. It also presents a novel approach for implementing the admittance controller, that borrows from [50], but decreases the response time using the position tracking system developed in **Chapter 12**.

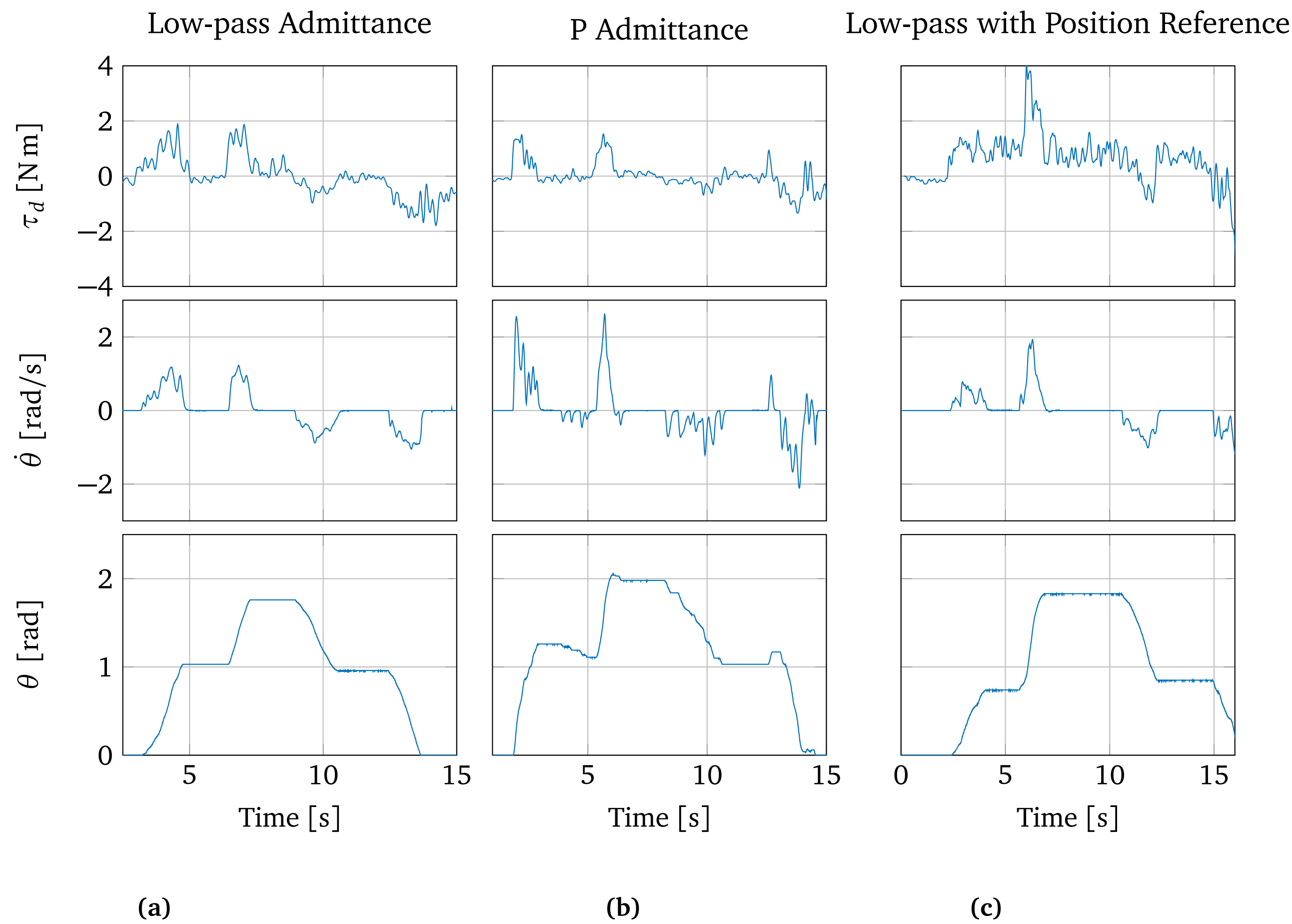


Figure 14.3: *Responses of two of the different suggested admittance control strategies, tuned to the user's satisfaction. The two lower rows, $\dot{\theta}$ and θ, are measurements, not reference signals. Strangely, the desired torque on the with the third method is significantly larger than the other two. This is likely due to slight differences in placement of the Myo Band.*

Before the low-pass admittance controller suggested by **Section 14.1.1** is implemented, a new computed torque structure is needed. In **Section 12.2.2**, the controller was left as a PID position controller. In the case of this controller, however, only a velocity controller is need. Using proportional control, the closed loop feedback linearized velocity control system becomes:

$$\frac{\theta}{\theta_d} = \frac{K_d}{K_d + s} \tag{14.3}$$

The controller has been designed to comply with the requirements listed in **Section 3.2.1**, namely that; the closed loop bandwidth should be $6\,\text{Hz} = 37.7\,\text{rad/s}$, and a gain margin of at least 10 dB. The proportional controller gain is thus chosen to be $K_d = 37$, resulting in a closed loop bandwidth of $36.9\,\text{rad/s}$. The gain margin is infinite, and the phase margin is 180°.

The admittance controller has been tuned, and the best performance was achieved with **Equa-**

tion (14.4).

$$A_1(s) = \frac{10}{s+10} \tag{14.4}$$

That admittance controller results in the response shown in **Figure 14.3a**. The controller felt quite responsive. It stays in place when the user is not moving, as expected. The movement sequence of 0-45-90-45-0 ° is well preserved, with significant steady state errors, which is caused by the user. The user further reported that the AXO-arm felt controllable, and moved without large resistance. There is, however, a small delay between the torque reference and the velocity response. This effect was felt by the user, but is not visible in the plot.

The relatively good performance of this controller also makes logical sense, because torque is linear in acceleration, so creating a velocity reference by integrating torque is logical.

When using the P admittance controller described in **Section 14.1.2**, the arm moves faster than with the low-pass admittance controller. The best response was found with a $A_2(s) = 2$, but the integral action on the position reference creates a noticeable and undesired delay in the movement. Overshoot on the position is also clearly visible in **Figure 14.3b**, which is undesirable.

The position response shown in **Figure 14.3b** is not as good as the one created by the PI admittance controller. However, there is a slight, but noticeable delay between the joint torque reference and velocity response when using the low-pass admittance controller. It might not be visible on **Figure 14.3a**, but the test subject felt it during use. To reduce the delay between muscle activation and AXO-arm motion, a third and novel approach is suggested. In this approach, the admittance filter from **Equation (14.4)** is kept, but instead of using only velocity control in the computed torque controller, the PID position controller is used, and a reference trajectory generation system is implemented to get at position reference from the velocity. While generating position references by integrating the velocity reference might seem the most logical choice, **Figure 14.3b** shows that this method results in delays and overshoot on the motion, the delay is only expected to increase with the addition of a low pass filter in front of it. Instead, the position reference generator is implemented by moving the current measured position one time-step forward. The calculation of the position reference at time k is

$$\theta_d(k) = \theta_{\text{meas}}(k) + T_s\,\dot{\theta}_d(k) \qquad [\text{rad}] \tag{14.5}$$

This controller is compared to the two others in **Figure 14.3c**. The movement of the AXO-arm is now initiated immediately after the reference torque increased, and the user experienced an increase in responsiveness. The admittance control system is thus implemented using the third method, with the transfer function $A_1(s)$ in **Equation (14.4)**, and a position reference generated relative to the current position as shown in **Equation (14.5)**. The structure of the resulting admittance controller is shown in **Figure 14.4**

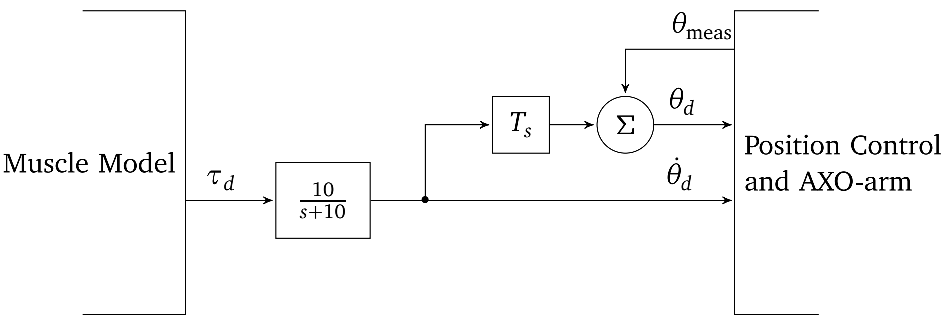


Figure 14.4: *Admittance controller scheme*

A position response similar to that of **Figure 14.3a** could have been achieved with the AXO-arm turned off, because the user can move the joint even without the help of the motors. The torque produced by the AXO-arm is therefore shown in **Figure 14.5**, along with the estimated human input torque. The applied motor torque is calculated from the current measurements, $\tau_m = k_t\, i_a\, N$. While

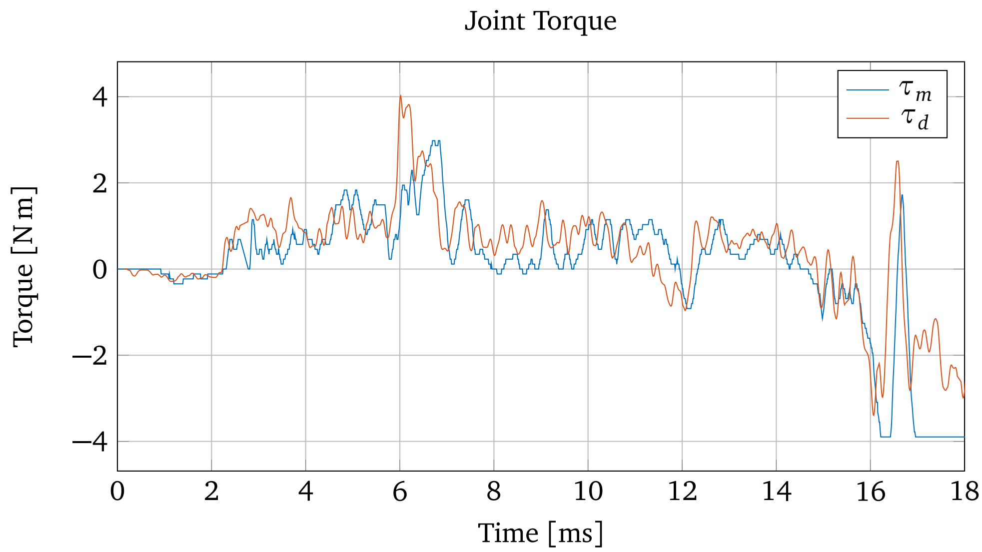


Figure 14.5: *The joint torque produced by the motor. The input torque has a sudden drop at* 11.5 s*, the point where the joint moves to* 0 rad *and hits the mechanical stop. The flat part of the negative torque is due to the maximum current limit on the elbow motor.*

there is some delay between the human joint torque and the motor torque, it is significantly smaller than when using pure torque control as shown in **Figure 13.3**. The admittance control system is thus accepted for use in the verification section, where its capability of supporting the user when under load is examined.

Part IV

Verification and Conclusions

This part tests and verifies the implemented control system and concludes on the project as a whole.

System Verification 15

The admittance controller that was developed in **Chapter 14** should be verified against the requirements of **Chapter 3**. The model-requirements were verified in their respective chapters, **Chapter 7** and **Chapter 10**. The position controller is verified during the initial tests in **Chapter 12**. This chapter thus evaluates the performance of the EMG-based control system, and its ability to not only follow the user and not be in the way, but also supporting the user when loaded. The chapter also comments on the safety related requirements of **Section 3.2.5**.

15.1 Assisting Control

The requirements specify that the AXO-arm system must carry 50 % of the load attached to the user. To formulate this more precisely, if the user is holding a load with some mass, the AXO-arm controller must apply an upwards force on the hand that is equal to half of that mass. To verify the supporting features of the controller, two trajectories are captured. One where there user follows the reference trajectory described in **Section 13.1** without a load, and one where the hand is loaded with a dumbbell disc weighing 560 g. These two motions are shown in **Figure 15.1**.

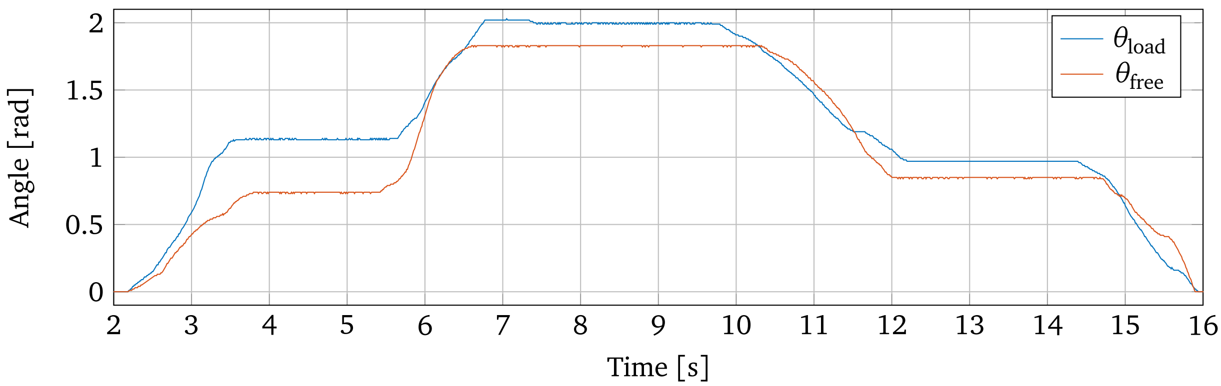


Figure 15.1: *The two movements used to test the support of the AXO-arm controller. While there is a different in angles and velocity, they are well aligned in time*

While the human joint movements differ slightly in angle and velocity, they are aligned in time. This allows us to compare the controller performance without having to tamper with the time axis.

To compare the two motions, we will look at the applied motor torque, τ_m, and at the estimated human joint torque, τ_d, of both motions. The difference between them should be equal to half the load that is added by the dumbbell disc.

The motor torque, τ_m is calculated from the current, $i_a(t)$ using the motor constant, k_t, and the gear ratio, N.

$$\tau_m = i_a(t)\, k_t\, N \qquad [\mathrm{N\,m}] \qquad (15.1)$$

The load torque is calculated using the weight, m_w, of the dumbbell disc, the length of the test subject's forearm, l, and the gravitational acceleration, g. The load also has an inertia term, calculated as a point mass that is translated the length of the subjects forearm away from the elbow joint. The dynamics of the load weight w.r.t. the elbow is described by

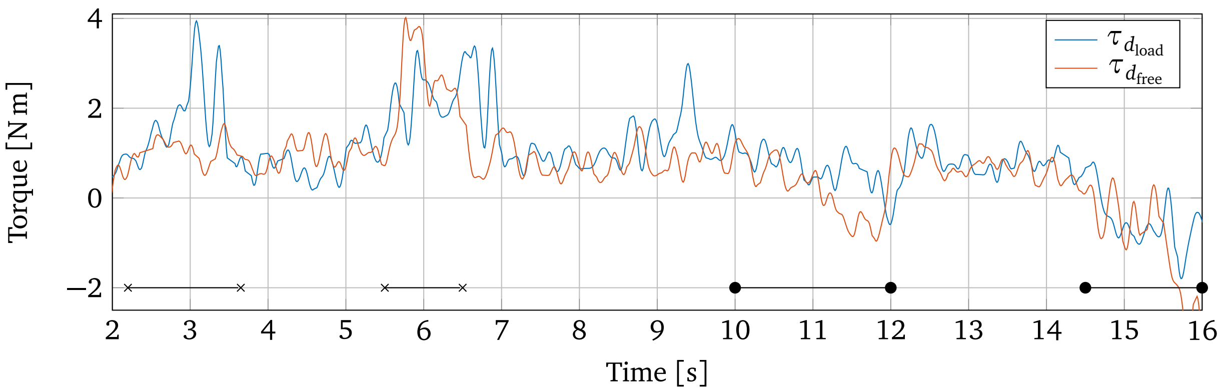
$$\tau_w = I_w \ddot{\theta} + \tau_g \qquad [\mathrm{N\,m}] \qquad (15.2)$$

$$\tau_g = \sin(\theta)\, m_w\, l\, g \qquad [\mathrm{N\,m}] \qquad (15.3)$$

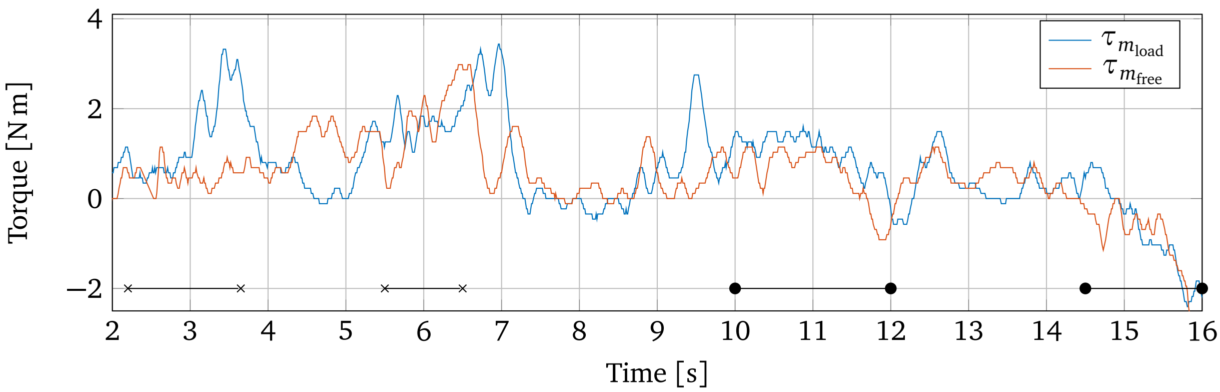
$$I_w = m_w\, l^2 \qquad \left[\mathrm{kg\,m^2}\right] \qquad (15.4)$$

Where τ_w is the resulting elbow joint torque from the added weight.

The human input torque, τ_d, during the two test motions is shown in **Figure 15.2a**, and in **Figure 15.2b** the applied motor torque is shown. Upwards motion is indicated with lines terminated by × and downwards motion is indicated with •.



(a) *The estimated human joint torque.*



(b) *The torque applied by the motor.*

Figure 15.2: *The motor- and human joint torque during the exercise. Lines terminated by × indicate upwards motion and •indicates downwards motion.*

By comparing the torques of **Figure 15.2**, and keeping in mind the trajectory shown in **Figure 15.1** several observations can be made.

The motor primarily outputs a torque when the load is lifted. It appears that the motor output torque drops when the AXO-arm is holding the load at constant angles. This could be due to the static joint friction not being properly compensated for. In fact, since the coulomb friction is implemented using a sigmoid function, the friction calculated from the feedback linearization at exactly 0 rad/s is zero. It thus seems that when the AXO-arm is stationary, the friction of the elbow joint is capable of holding, not only the mass of the user's arm, but also the dumbbell disc.

An strange phenomenon, visible at around 7 s to 10 s, on both **Figure 15.2a** and **Figure 15.2b** is that a nonzero τ_d, results in a τ_m close to zero. This could mean that something in the computed torque controller creates an offset. Recall that the computed torque controller consists of a PID controller and feedback linearization. Contributions from the proportional, integral, and derivative part of the PID are shown in **Figure ??**, along with the resulting acceleration reference, y.

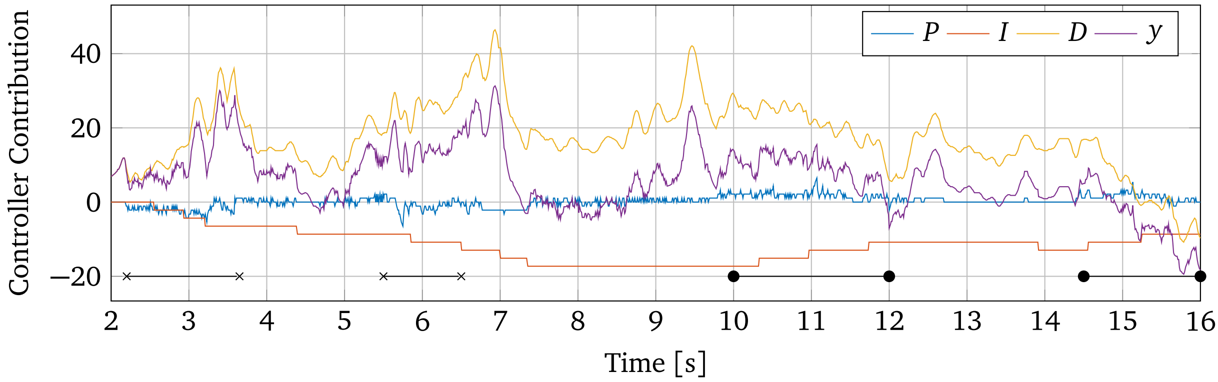


Figure 15.3: *The different contributions of the PID computed torque controller during movement with load. For some reason, the integral part gets negative, and stays negative, resulting in an acceleration reference $y \approx 0$ at 7 s to 9 s. The integral contribution looks like it has a lower resolution than the other signals. This is due to the error signal having a finite resolution and the I gain being the largest - thus resulting in a blocky graph.*

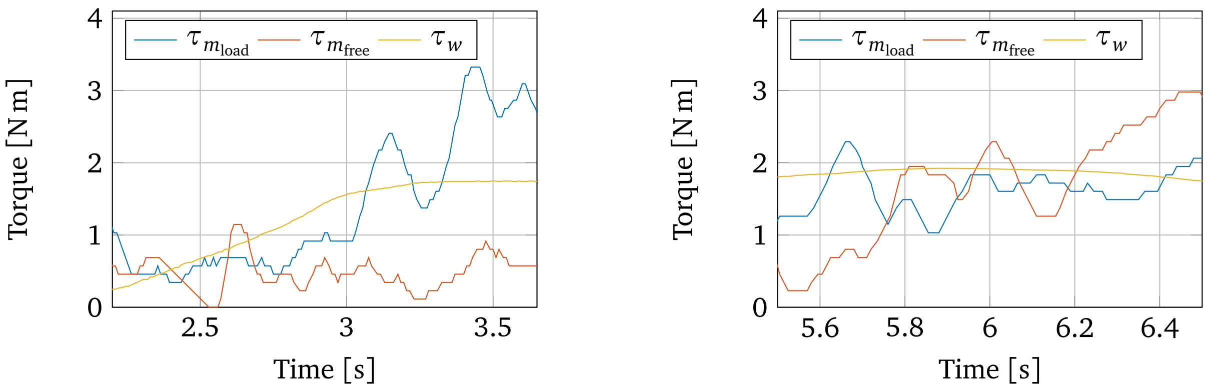
Strangely, the contribution from the integral part of the controller gets negative, and does not return to zero over the entirety of the measurement. This is likely due to an implementation error, but due to time constraints it has not been investigated further.

At the beginning of the first motion downwards, at 10 s to 12 s on **Figure 15.2**, it appears that the direction of the motion has the opposite sign of the human joint torque and motor torque. It appears to be the case for both movement with the dumbbell disc, and movement without it. This means that during this downwards movement, the user is actually working against the exoskeleton to move it downwards. The user did not report this as uncomfortable, which might be because gravity on the arm helps with downwards movement. The reason for this anomaly is not investigated further, but it could easily be the muscle model that estimates a joint torque that is incorrect.

When the joint angle becomes close to 0 rad the AXO-arm accelerates. This is visible on **Figure 15.1** at around 15.5 s. It seems that the muscle-model has a strange behavior around this angle which results in a large negative torque reference.

Looking at the motor torques during upwards movement, as in **Figure 15.4**, it is seen that, during the first lift, the motor applies more torque when the user is carrying the dumbbell disc than when he is not, but in the second, the two are more similar. In the first upwards movement, it looks like the difference in torque between movement with and without the load is close to the torque that is added by the load, τ_w. This is not the case for the second lift though.

The velocity of the loaded movement was highest during the first movement, and the velocity of the free movement was highest during the second movement, so the differences in applied motor torque could also be attributed to this difference in velocity.



(a) *The motor torque during the first upwards upwards motion*

(b) *The motor torque during the second upwards upwards motion*

Figure 15.4: *Motor torque during upwards movement, next to the load torque of the added weight.*

The data presented here is not sufficient to verify if the AXO-arm controller applies half the torque needed to lift the load. While the controller is definitely helping with the lift, as seen on **Figure 15.4**, at the time of writing there are still too many unknowns in the system to verify the control system.

15.2 Safety Features

The mechanical construction of the AXO-arm allows elbow flexion/extension in the range 0° to 147° and shoulder flexion/extension in the range −10° to 190°. This means that the AXO-arm is mechanically forced to stay within the the joint limits specified by [19], except of the case of maximal shoulder flexion where the AXO-arm is able to move 10° outside. No remedy has been implemented for this, but a future iteration of the mechanical construction should take this into consideration.

The requirements also specify a maximum output-torque on the joints to avoid injuring the wearer. It is based on the technical standard for collaborative robots [20]. The elbow and shoulder motors have maximum continuous torques of 0.319 N m and 0.134 N m, respectively [51][52]. The gears have a maximum efficiency of $\epsilon = 80\,\%$ [53]. The maximum continuous output torque of the two gears is thus

$$\tau_{\text{max,shoulder}} = 0.319\,\text{N m}\,\epsilon N = 12.76 \qquad [\text{N m}] \qquad (15.5)$$

$$\tau_{\text{max,elbow}} = 0.134\,\text{N m}\,\epsilon N = 5.36 \qquad [\text{N m}] \qquad (15.6)$$

which is well within the bounds. N is the gear ratio.

In accordance with [20] the user should at all times have the means of stopping the AXO-arm. This feature has not been implemented, but should be easily doable by adding an emergency stop button that disconnects the ON-signal of the ESCON drivers. It could also cut power to the system, but deactivating the motors is a more elegant solution as it allows the system to take further action.

Conclusions and Perspectives 16

In this chapter, the project is concluded with considerations on possible improvements, and an overall evaluation of the result of the project.

Perspectives

Several choices for implementation and methodology were made in the course of the project. This section summarizes what could have been done differently to improve on the project.

16.0.1 The Choice of EMG Sensor

The Myo Band is an easy, quick and handy tool for gathering EMG data but it has its limitations. The Myo band is meant for the lower arm, but for some people it is also possible to attach to the upper arm. However, since it is an armband, it is not possible for the Myo band to measure the EMG of shoulder muscles. Therefore another solution is needed, if other AXO-arm joints than the elbow, should be controlled. Additionally, the Myo Band is made for gesture recognition where the full characteristics of the EMG signal is not needed, but rather the difference in activity between the pods. A consequence of this is that the EMG signal is often saturated during use, and the resolution is only 8 bit. It is also clear that the Myo Band was not designed for real-time use. It has been observed that the we are only receiving EMG measurements with a somewhat irregular sample-rate over the BLE interface. The actual rate was found to vary between 75 Hz and 85 Hz, and not reaching the 100 Hz advertised in the data sheet. Clearly, the choice of EMG sensor can therefore be questioned.

16.0.2 The Muscle Model

The muscle model described in **Chapter 9** and **Chapter 10** is deemed satisfying, but some of the found parameters are unrealistic. An example is the maximum force of CE, $F_{CE_{max}}$, which is found to be more than 4000 N. The reason might be that the activation signal $a(t)$ is lower than expected, possibly due to the often saturated EMG measurements.

Regarding the muscle model parameters, these are individual for each person wearing the exoskeleton. New parameters therefore have to be found each time a new person wears the suit and the parameter optimization can take several days to complete. To ease this process a future improvement is to design and implement a calibration program that is used every time the suit is attached to a person or at least every time a new person is wearing it. The calibration program should quickly estimate the necessary model parameters and to do this a faster parameter estimation is needed, possibly with a reduced search space.

Other methods for muscle modeling could also be investigated, such as machine learning approaches or maybe commercial muscle simulation systems, such as the AnyBody Modeling System.

The dynamics of the EMG is dependent on many factors so the EMG processing could be expanded to take into account more signal properties than what is included in the present work. Since it is chosen to use non-invasive surface electrodes an immediate consequence is the crosstalk which occurs when signals from different muscles propagate to the skin [54]. It is not taken into account by the muscle model, but it is a potential source of error. A similar problem that is not taken into account is the choice of hand orientation during elbow movement. If the user's palm points upwards, the EMG signal of biceps brachii is different from when the user points his palm downwards.

Additionally the EMG signal varies dependent on fatigue in the muscle [55]. So, clearly, a very advanced model could be devised when using EMG.

A commonly used method for compliant control of robots that interact with humans is to measure the interaction force between the user and the robot with either FSRs as in [56] or with strain gauges. While the EMG sensor has the advantage of detecting human movement before it happens, the large amount of information in the signal makes it difficult to work with. Combining the EMG with direct interaction force measurements is an option that might produce better results than using the EMG alone.

Modeling of muscles and human motor control is an area that is heavily researched these years. The AXO-SUIT project is part of SPARC, which works towards improving health care robotics the European Union [5]. Modeling muscle dynamics and human motion for use in exoskeletons is thus part of the SPARC area of research[57].

16.0.3 The Mechanical Construction

For future improvements it is of course relevant to include control of more joints than only the elbow. Including more joints would increase the DOF and therefore be of better help to people with muscle disabilities. During the course of the project, a mechanical engineering team working on the AXO-arm has increased its DOF to 4. Two joints are thus added; an actuated joint capable of shoulder abduction and adduction, and a joint that is not actuated allowing medial and lateral rotation. The two added joints are shown on **Figure 16.1**.

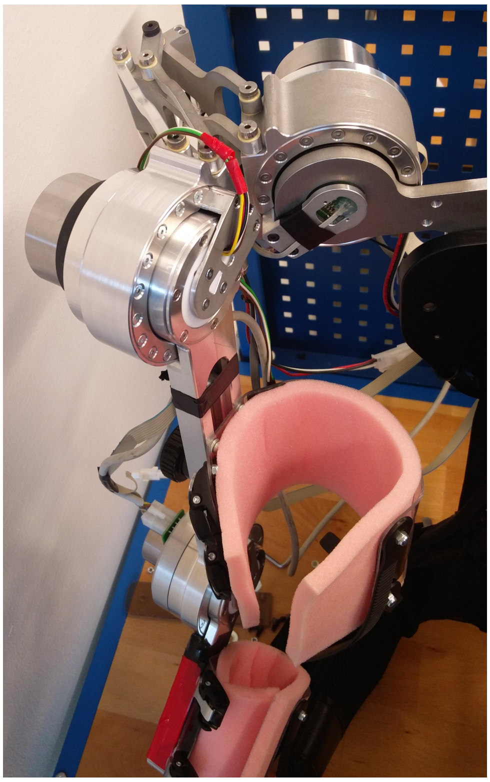


Figure 16.1: *The AXO-arm with the two additional joints that are added for increased mobility.*

Another possible improvement for the mechanical structure is to lower the coulomb friction in the motor gears. The high coulomb friction is an annoying factor when controlling low velocities since the movement is not as smooth as wanted. Therefore a possible solution is to study other types of gearboxes or completely change the mechanical setup such that the joints are actuated differently.

The motors actuating the mechanics can introduce violent movements, when not controlled correctly. Since some errors are first noticed when the controller is implemented on the real system, an emergency stop would be a good idea to prevent damaging the mechanics or injuring the user. An emergency stop is also required by the technical standard for collaborative robots [20].

16.0.4 Reflections on the chosen Microcontroller

Early in the project it was chosen to work with the UDOO Neo controller board. It was chosen in part to try working with a new technology, and in part because it is a nice and compact system that has both a proper operating system and the capability of real time software execution. In practice, however, the platform is too unstable to be recommendable to others. The UDOO communicates with a host computer using an emulated ethernet port over USB, which has proved extremely unreliable. The Arduino interface to the M4 core works fine, but it is difficult to debug as the interfacing capabilities are limited. During the course of the project, the AXO-arm has also been outfitted with two additional DOF, resulting in four additional analog signals to be measured. The UDOO Neo only has six ports on its ADC, so using it for control of the expanded AXO-arm is not possible without an external ADC. A superior choice would have been an industrial grade interface card, and a regular laptop or desktop computer.

Conclusions

This project set out to create a control system for the ongoing AXO-arm project at AAU. For this purpose, the electromechanical AXO-arm system is modeled, and the model is parameterized and verified against measurements. Using feedback linearization and PID control, a position controller for the AXO-arm is developed, allowing fast and precise tracking of trajectories. To connect the AXO-arm and its user, a general EMG-based model of the torque in a human arm joint, based on the works of [6], is developed. A model of the elbow joint using EMG-measurements obtained with a Myo Band is parameterized and verified against measurements. To make the Myo Band usable on Linux-based platforms, an open source python library for gathering EMG- and IMU-measurements from it has been developed.

Several ways of connecting the muscle model to the AXO-arm are attempted. Control by simply applying the estimated torque to the feedback linearized system is compared to two different implementations of admittance control. The admittance controller that proved most comfortable was one where the torque is low-pass filtered to generate a velocity reference for the PID controller, and a simple position trajectory generator creates the position reference. This structure results in a system that is easy to control, and supports the user when lifting a load.

While the problem statement in **Section 1** asks for a controller for both joints on the AXO-arm, the limited placement possibilities of the Myo Band only allows parameterization of a model of the elbow joint. However, the method presented in this work for obtaining joint torque and EMG-measurements is adaptable to also include shoulder movement. This upgrade requires EMG-sensors with more versatile placement options, but the works of [6] and [34], combined with the parameter estimation algorithm suggested in this thesis, makes the addition of shoulder joint control possible.

The test in **Chapter 15** proved somewhat successful. While the test lacks a quantitative measure of the support of the AXO-arm controller, the user did report that the controller was responsive and supportive during the lift of a dumbbell disc.

The findings of this thesis can thus function as a good basis for further work in exoskeleton controller design. The hierarchical controller based on [42] adds structure to the exoskeleton controller design process, while still remaining open to interpretation and individualized solutions. The AXO-arm development at AAU is with the addition of this master thesis, one step closer to increasing the quality of life of users with reduced muscle strength, reducing their dependence on help from the government, and possibly allowing them to remain a part of the workforce longer.

Bibliography

[1] Kellner Tomas. *The Story Behind the Real 'Iron Man' Suit*. URL: http://www.gereports.com/post/78574114995/the-story-behind-the-real-iron-man-suit/ (visited on 11/23/2010).

[2] US Bionics Inc. *Suitx, the Benchmark for Exoskeletons*. URL: http://www.suitx.com (visited on 01/23/2017).

[3] euRobotics. *What is SPARC? The Partnership for Robotics in Europe*. URL: https://www.eu-robotics.net/sparc/about/index.html (visited on 06/05/2017).

[4] SPARC. *TOPIC GROUPs and COORDINATORS - Status January 2017*. URL: https://www.eu-robotics.net/sparc/upload/topic_groups/list-Topic-groups-2017-01.pdf (visited on 06/05/2017).

[5] AXO-SUIT. *WELCOME TO AXO-SUIT*. URL: http://http://www.axo-suit.eu/ (visited on 01/23/2017).

[6] Ettore E. Cavallaro et al. "Myoprocessors for a Neural Controlled Powered Exoskeleton Arm". In: *IEEE Transactions on Biomedical Engineering* (2006).

[7] Darwin G. Caldwell et al. ""Soft" Exoskeletons for Upper and Lower Body Rehabilitation - Design, Control and Testing". In: *International Journal of Humanoid Robotics* (2007).

[8] Thunyanoot Prasertsakul, Teerapong Sookjit, and Warakorn Charoensuk. "Design of Exoskeleton Arm for Enhancing Human Limb Movement". In: (2011).

[9] Carlo J. Den Luca et al. "Filtering the surface EMG signal: Movement artifact and baseline noise contamination". In: *Jounal of Biomechanics* (43 2010).

[10] Thalmic Labs Inc. *myo::DeviceListener Class Reference*. URL: https://developer.thalmic.com/docs/api_reference/platform/classmyo_1_1_device_listener.html#a79f2abac09f0408c859cccf991d5294a.

[11] Paul Bernhardt. *#MyoCraft: EMG in the Bluetooth Protocol*. URL: http://developerblog.myo.com/myocraft-emg-in-the-bluetooth-protocol/ (visited on 10/24/2016).

[12] Thalmic Labs. *Myo Gesture Control Armband*. URL: https://www.myo.com (visited on 12/01/2016).

[13] *ESCON Module 50/5. Hardware Reference*. 438725. Maxon Motor. Nov. 2015.

[14] Paul Bernhardt. *Myo Bluetooth Protocol Released*. URL: http://developerblog.myo.com/myo-bluetooth-spec-released/ (visited on 10/24/2016).

[15] Thalmic Labs. *Myo-Blutooth*. URL: https://github.com/thalmiclabs/myo-bluetooth (visited on 10/24/2016).

[16] P. D. Neilson. "Speed of response or bandwidth of voluntary system controlling elbow position in intact man". In: *Medical and biological engineering* (4 1971).

[17] Karl Johann Åstrom. *Advanced PID control*. ISA- Instrumentation, Systems and Automation Society, 2006.

[18] Gene F. Franklin, J. David Powell, and Abbas Emami-Naeini. *Feedback Control of Dynamic Systems*. 7th. 2015. ISBN: 1-29-206890-6.

[19] Joumana Medlej. *Humana Anatomy Fundamentals: Flexibility and Joint Limitations*. Mar. 6, 2015. URL: https://design.tutsplus.com/articles/human-anatomy-fundamentals-flexibility-and-joint-limitations--vector-25401 (visited on 11/04/2017).

[20] *Robots and robotic devices - Collaborative robots*. ISO/TS. 2016.

[21] John J. Craig. *Introduction to Robotics Mechanics and Control*. 3rd. 2005. ISBN: 0-13-123629-6.

[22] Seth Hutchinson Mark W. Spong and M. Vidyasagar. *Robot Modeling and Control*. 1st. 2005. ISBN: 0-471-64990-2.

[23] S. Shankar Sastry Richard M. Murray Zexiang Li. *A Mathematical Introduction to Robotic Manipulation*. 1st. 1994. ISBN: 9780849379819.

[24] Rafael Wisniewski. *Mechanical Systems II Lagrange Mechanics*. URL: http://www.control.auc.dk/~raf/ModellingOfMech/MechanicalSystemsSection3.pdf. (visited on 03/25/2005).

[25] SL Montevideo Technology Inc. *Brushless vs Brushed Motors*. URL: http://www.dynetic.com/brushless%20vs%20brushed.htm (visited on 01/25/2017).

[26] *Motor data and operating ranges of maxon DC motors*. Maxon Motor. 2010.

[27] *ESCON Overview*. Maxon Motor. Apr. 2016.

[28] Lennart Ljung. "Practical Issues of System Identification". In: *Encyclopedia of LifeSupport Systems* (June 2007).

[29] Maxon Motor. *Maxon Motor 412825*. URL: http://www.maxonmotor.com/maxon/view/product/motor/ecmotor/ecflat/ecflat60/412825 (visited on 03/07/2017).

[30] Maxon Motor. *Maxon Motor 402687*. URL: http://www.maxonmotor.com/maxon/view/product/motor/ecmotor/ecflat/ecflat45/402687 (visited on 03/07/2017).

[31] Carol A. Oatis. *Kinesiology: The Mechanics and Pathomechanics of Human Movement*. 2nd. 2009. ISBN: 0-7817-7422-5.

[32] Thor F. Besier David G. Lloyd. "An EMG-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo". In: *Journal of Biomechanics* (Dec. 2002).

[33] Potvin Jim R, Norman Robert, and Mcgill Stuart M. "Mechanically corrected EMG for the continuous estimation of erector spinae muscle loading during repetitive lifting". In: *European Journal of Applied Physiology and Occupational Physiology* (Feb. 1996).

[34] Pascale Pigeon, L'Hocine Yahia, and Anatol G. Feldman. "Moment Arms and Lengths of Human Upper Limb Muscles as Functions of Joint Angles". In: *Journal of Biomechanics* (1996).

[35] *Kin-Com Clinical Reference*. Isokinetic International. 1994. URL: https://smiadm.hst.aau.dk/smiold/manuals/techman/kincom/.

[36] Paavo V. Komi Robert W. Norman. "Electromechanical delay in skeletal muscle under normal movement conditions". In: *Acta Physiologica* (1979).

[37] Olivier de Weck Raina Hassan Babak Cohanim and Gerhard Venter. "A Copmarison of Particle Swarm Optimization and the Genetic Algorithm". In: *AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* (2007).

[38] Sidhartha Panda and Narayana Prasad Padhy. "Comparison of particle swarm optimization and genetic algorithm for FACTS-based controller design". In: *Applied Soft Computing* (2005).

[39] Russel Eberhart James Kennedy. "Particle Swarm Optimization". In: *Proceedings of IEEE International Conference on Neural Networks* (1995).

[40] Maurice Clerc. "Standart Particle Swarm Optimization". In: *HAL* (2012).

[41] Marek Ruciński Francesco Biscani. *PaGMO (Parallel Global Multiobjective Optimizer)*. Mar. 6, 2015. URL: https://esa.github.io/pagmo/index.html.

[42] Khairul Anam and Adel Ali Al-Jumaily. "Active Exoskeleton Control Systems: State of the Art". In: *Procedia Engineering* (2012).

[43] Marc G. Carmichael and Dikai Liu. "Admittance Control Scheme for Implementing Model-based Assistance-As-Needed on a Robot". In: *35th Annual International Conference of the IEEE EMBS* (2013).

[44] Wen Yu, Jacob Rosen, and Xiaoou La. "PID Admittance Control for an Upper Limb Exoskeleton". In: *American Control Conference* (2011).

[45] Jian Huang et al. "Using a Human-Robot Interface Based on Motion Intention Recognition". In: *Transactions on Automation Science and Engineering* (2015).

[46] R Kelly, V. Sentibáñez, and A. Loría. *Control of Robot Manipulators in Joint Space*. 2005.

[47] Tamar Flash and Neville Hogan. "The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model". In: *The Journal of Neuroscience* (1985).

[48] Mehrdad Yazdani et al. "A Simple Control Policy for Achieving Minimum Jerk Trajectories". In: *Neural Networks* (2012).

[49] Bruno Siciliano et al. *Robotics Modelling, Planning and Control*. 1st. 2010. ISBN: 978-1-84628-641-4.

[50] Chi et al. "Power Assistance for Human Elbow Motion Support Using Minimal EMG Signals with Admittance Control". In: *International Conference on Mechatronics and Automation* (2011).

[51] *EC 45 Flat*. Maxon Motor. Apr. 2016. URL: http://www.maxonmotor.com/medias/sys_master/root/8821068005406/16-304-EN.pdf.

[52] *EC 60 Flat*. Maxon Motor. Apr. 2016. URL: http://www.maxonmotor.com/medias/sys_master/root/8821067808798/16-301-EN.pdf.

[53] *Cup Type Component Sets & Housed Units*. Harmonic Drive LLC. Mar. 2017. URL: http://www.harmonicdrive.net/_hd/content/documents/csd-shd.pdf.

[54] M. Solomonow et al. "Surface and Wire EMG Crosstalk in Neighbouring Muscles". In: *Journal of Electromyography and Kinesiology 4* (1994).

[55] N.A. Dimitrova and G.V. Dimitrov. "Interpretation of EMG changes with fatigue: facts, pitfalls, and fallacies". In: *Journal of Electromyography and Kinesiology 13* (2003).

[56] Jian Huang et al. "Control of Upper-Limb Power-Assist Exoskeleton Using a Human-Robot Interface Based on Motion Intention Recognition". In: *Transactions on Automation Science and Engineering* (2015).

[57] The Partnership for Robotics in Europe. *Robotics 2020 Multi-Annual Roadmap*. May 31, 2017. URL: https://www.eu-robotics.net/cms/upload/topic_groups/H2020_Robotics_Multi-Annual_Roadmap_ICT-2017B.pdf (visited on 12/02/2016).

Measurement Report - Motor Measurements with Current Control A

This measurement report is used for **Section 5.5.2**. The purpose is to apply (and save) a current step input to both of the motors Maxon 412825(for shoulder) and 40268(for elbow) and measure their current and velocity, since this is used for comparing the time constants of velocity and current. This is done with three different amplitudes of the step input for generality. The motors have attached a small load/arm but are running independently of each other and are not attached to each other in any way, which means individual measurements for each motor. The motors are also using the auto-tuned current controller given in **Section 5.5.2**.
The general test setup for each motor is shown in **Figure A.1**. For each of the two motors, the rotation of the load is limited mechanically by an end-stop. Therefore a problem while doing the measurements can be that the velocity might not have time enough to reach steady state and get the full step response. In order to capture as much of the step response as possible the load is therefore rotated from its minimum joint angle to its maximum and with a relatively low velocities, but not so low that the result of the measurements losses too much generality. The motors were fixed on a table in such a way, that only horizontal movements were possible, which neglects gravity as an interfering factor.

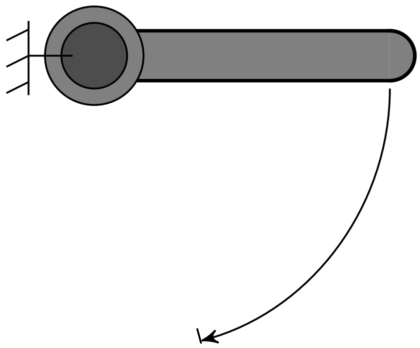
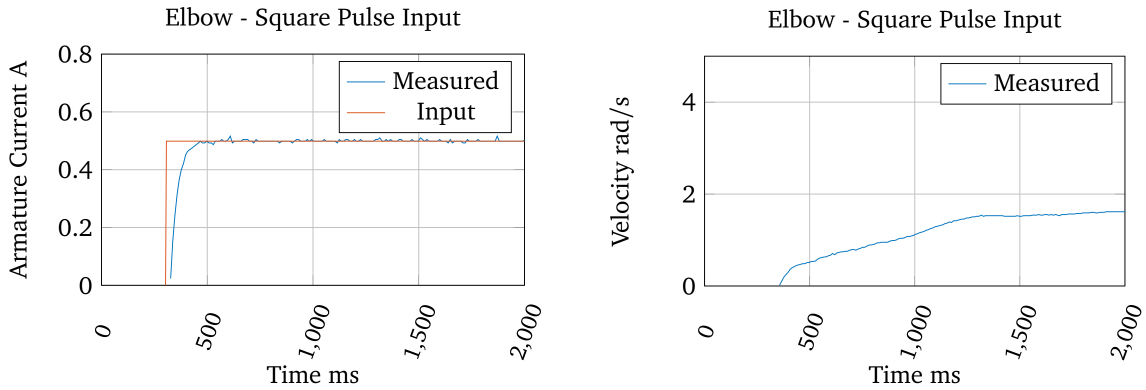


Figure A.1: *Test setup*

A.0.1 Results

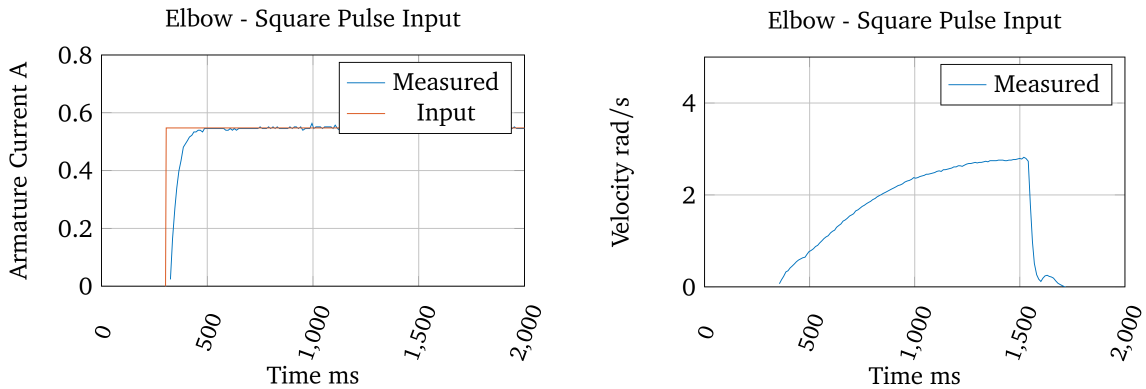
For the elbow joint motor the input steps are 0.5 A, 0.55 A and 0.6 A as shown in **Figure A.2a**, **Figure A.3a** and **Figure A.4a**. With armature currents below 0.45 A the static friction is too high for the motor to start moving, therefore, values above this are chosen. The maximum allowed current is 0.95 A.



(a) *At the time* 307 ms *the input step is applied. The measured current is settled at time* 458 ms. *The time constants of the current response is approximately* 367 − 307 = 60 ms.

(b) *The measured velocity is settled at time* ≈ 1500 ms. *The time constants of the velocity response is approximately* 912 − 307 = 605 ms.

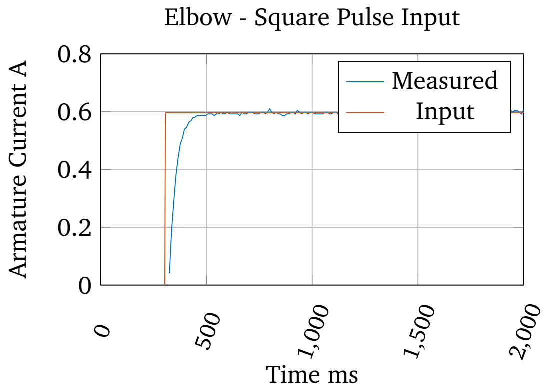
Figure A.2



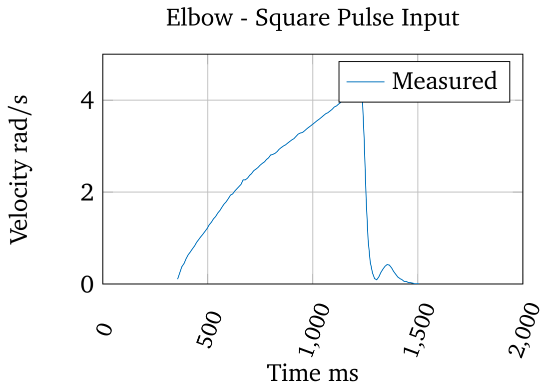
(a) *At the time* 306 ms *the input step is applied. The measured current is settled at time* 487 ms. *The time constants of the current response is approximately* 356 − 306 = 50 ms.

(b) *The measured velocity is unfortunately not settled at the time that the load reaches the end angle, which in the plot can be seen as the velocity going to zero fast.*

Figure A.3



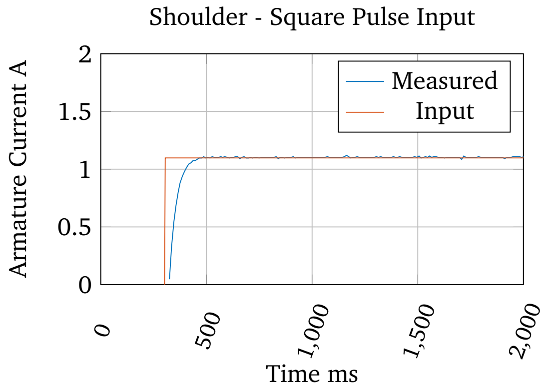
(a) *At the time* 306 ms *the input step is applied. The measured current is settled at time* 446 ms. *The time constants of the current response is approximately* $366 - 306 = 60$ ms.



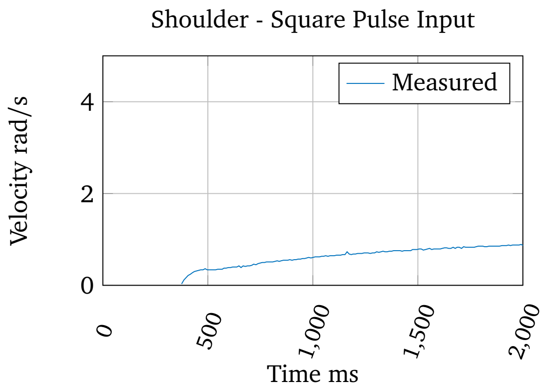
(b) *The measured velocity is unfortunately not settled at the time that the load reaches the end angle, which in the plot can be seen as the velocity going to zero fast.*

Figure A.4

For the shoulder joint motor the input steps are 1.1 A, 1.3 A and 1.5 A as seen in **Figure** A.5a, **Figure** A.6a and **Figure** A.7a. In Maxon studio a offset of 1 A is implemented. The maximum allowed current is 2 A

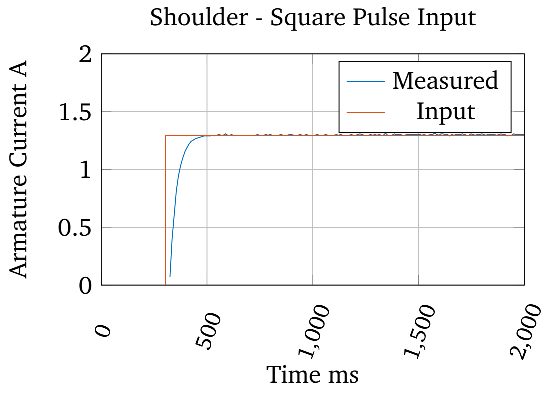


(a) *At the time* 305 ms *the input step is applied. The measured current is settled at time* 446 ms. *The time constants of the current response is approximately* $366 - 305 = 61$ ms.

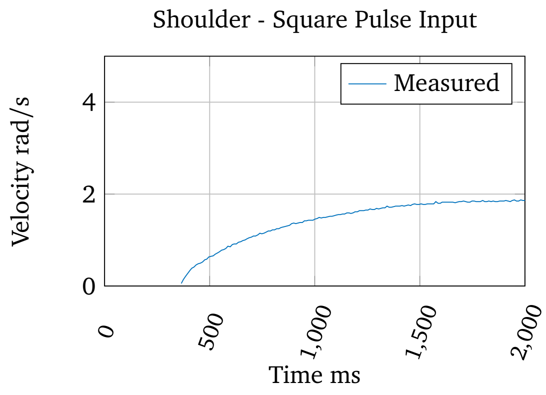


(b) *The measured velocity is unfortunately not settled at the time that the load reaches the end angle.*

Figure A.5

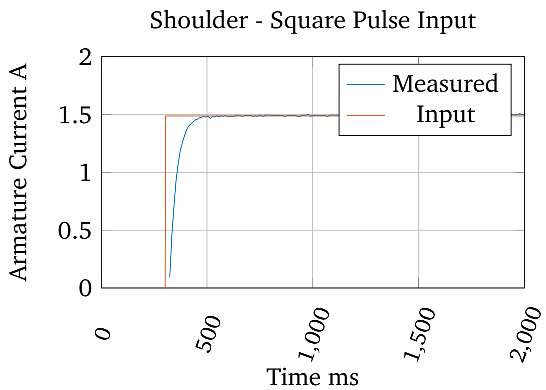


(a) *At the time* 305 ms *the input step is applied. The measured current is settled at time* 527 ms. *The time constants of the current response is approximately* $355 - 305 = 50$ ms.

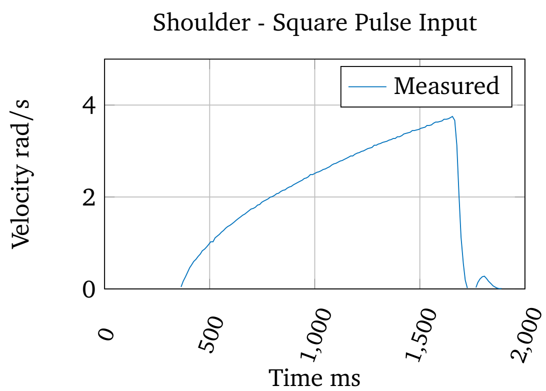


(b) *The measured velocity is unfortunately not settled at the time that the load reaches the end angle.*

Figure A.6



(a) *At the time* 304 ms *the input step is applied. The measured current is settled at time* 546 ms. *The time constants of the current response is approximately* $364 - 304 = 60$ ms.



(b) *The measured velocity is unfortunately not settled at the time that the load reaches the end angle, which in the plot can be seen as the velocity going to zero fast.*

Figure A.7

A.1 Conclusion of the Experiment

Regarding the motor Maxon 40268(for elbow), the results are shown in **Figure** A.2, **Figure** A.3 and **Figure** A.4. As observed besides the amplitude, all 3 measurements are very similar which indicate that the result is general. Even through the velocity responses are not all settled, it can clearly be observed that the time constants of the velocity responses are much larger than the time constants of the current responses. Also the input current are in all cases of different amplitude the same as the measured current, which implies unit gain from input current to measured current.

The results for the motor Maxon 412825(for shoulder) are very similar. The shoulder joint results are shown in **Figure** A.5, **Figure** A.6 and **Figure** A.7 and again these indicates that the results are general. Again even through the velocity responses are not settled, it can clearly be observed that the time constants of the velocity responses are much larger than the time constants of the current responses. Also the results showing input current and measured current imply unit gain with zero steady state error.

Measurement Report - Elbow Muscle Measurements for Parameter Estimation B

To be able to estimate the parameters of the different muscle models described in **Chapter 9**, data is needed. Here the inputs and output of the combined muscle model shown in **Figure B.1** needs to be measured and saved since these data vectors represent the data for parameter estimation. The inputs of the combined muscle model are joint angle and EMG signals, and the output is the elbow joint torque. As seen in the figure, the signal EMG_1 is input for two muscle models. The reason for

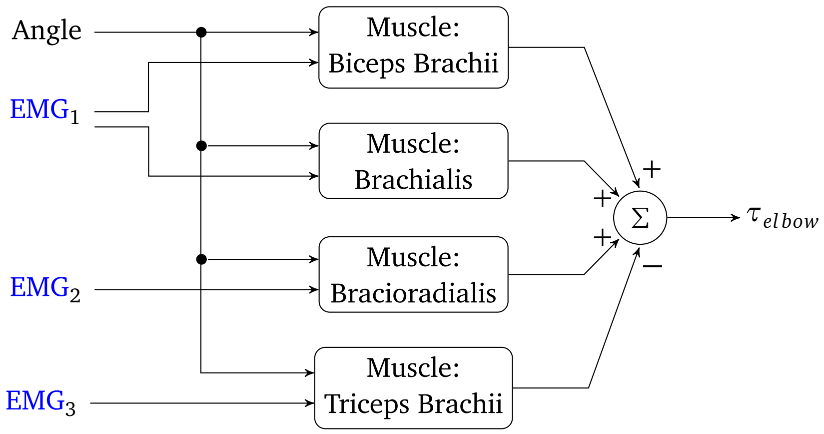


Figure B.1: *The figure shows an approach for combining the different muscle models that are related with elbow joint movement.*

doing this is that Biceps Brachii and Brachialis is located almost in the same place on the upper arm. This means that their muscle potential is also going to be measured with the same EMG sensor. The reason for the torque produced by the muscle model of Triceps Brachii to be subtracted from the other muscle models, is their difference in joint direction. With the objective clear an explanation of the material list, equipment setup, measurement procedure and results are presented in the following.

B.1 Material List

The materials used for the measurement are:

- 2 Myo Armbands (for measuring EMG)
- Kin-Com 125AP machine (for measuring angle and torque)
- USB Portable Diskette Drive (for extracting data from the Kin-Com)
- Diskette
- Computer (for extracting data from the Myo armbands)

B.2 Equipment Setup

In this section the setup is described, such that it is known how to attach the Myo armbands to an arm and how to position a person in the Kin-Com.

B.2.1 Myo Armbands

The Myo armbands are placed on the right arm in such a way that they can capture the EMG of all four muscles. The position of the muscles are shown in **Section** 9.4. One Myoband is placed on the lower right arm where the sensor pod number 4 is placed on top of the muscle Bracioradialis and with the USB connector pointing towards the shoulder, as shown in **Figure** B.2b. The numbering of the sensor pods are shown in **Figure** B.2a. The second Myoband is placed on the upper arm, again

(a) *Numbering of sensor pods. The USB connector is in bottom on the sensor pod number 4. [11]*

(b) *The two Myo Bands mounted on an arm*

with the USB connector pointing towards the shoulder. Sensor pod number 4 is placed on top of the muscle Biceps Brachii and thereby also covers the muscle Brachialis. This position should roughly make pod number 1 or 8 fit with the position of the muscle Triceps Brachii.

B.2.2 Kin-Com

The Kin-Com is a machine wherein a person is placed to do a certain exercise, with the options of measuring joint angle, joint angular velocity and limb force during the exercise. The Kin-Com is shown in **Figure** B.3 and as seen both the seat and dynamometer (measure device) has to be adjusted for the specific setup, which in this case is to measure elbow force (which will later be converted to torque) and angle. To obtain the position of seat and dynamometer the Kin-Com computer is used, since it can guide the user through the positioning for a certain exercise. After guidance from the computer, adjustments are done manually to make the machine setup more comfortable; an example is the maximum and minimum elbow angle during the exercise. After the positioning, the test person

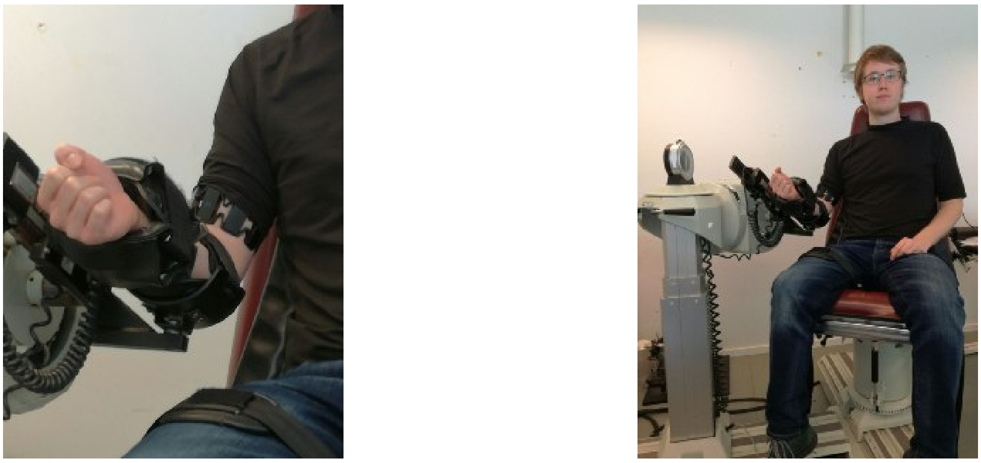


Figure B.3

is placed in the seat where his arm is fixed in the dynamometer as shown. Here it is important that the elbow is placed such that it rotates around the dynamometer rotor axis.

B.3 Measurement Procedure

In this section; first it is explained how to obtain measurement data using the Myo armbands, and secondly the Kin-Com. The Myo armband and Kin-Com measurements have to run simultaneously for the muscle parameter estimation to be valid. Unfortunately the Kin-Com machine does not put at time stamp on the saved measurement data, which induce the problem of how to compare the Kin-Com data with the Myo armband data. Fortunately since the Kin-Com offers joint angle measurements, and the Myo armbands offer accelerometer data(which is angle dependent), these data are used to align the Myo armband and Kin-Com measurements in the most optimal way.

For obtaining the Myo armband data the following steps are followed:

1 By use of the Myo armband setup described in **Appendix C**, the two armbands can communicate with a computer where IMU and EMG data are collected.

2 With the computer setup ready, the armbands are attached on the right arm of the testperson as described earlier.

3 On the computer the measurements are first initiated when it is clear that the Myo armbands are turned on and that contact has been established between the computer and armbands.

For obtaining the Kin-Com data the following steps are followed:

1 On the Kin-Com computer an isokinetic elbow flex/extend exercise is started.

2 Either a flex or extend exercise is chosen (the entire measurement has to be done for both).

3 With the testperson ready he either begins to flex or extend his arm corresponding to the choice of exercise. While the testperson constantly does this the Kin-Com dynamometer begins to steer the testpersons arm in a rotating movement up and down with constant velocity. 10 times the arm is steered up and down and the measurement ends.

4 The measurements are saved on the Kin-Com and extracted with an diskette.

To make sure that both Kin-Com and Myo armband data is present for all 10 arm movement repetitions, the Myo armbands should begin measuring earlier than the Kin-Com. With both procedures followed, such that measurement data is obtained simultaneously from the Myo armbands and Kin-Com, only more measurement data can improve the parameter estimation. Therefore three extend and three flex exercise measurements are obtained in total. Regarding to the Kin-Com force measurement, it can be converted to measured elbow torque by use of **Equation (B.1)**.

$$\tau = F \cdot l \tag{B.1}$$

Where F is the force measured, τ is the torque measured and l is the distance from the Kin-Com rotor the Kin-Com force sensor. The distance l is measured to 0.22 m.

B.4 Results

In the following the data obtained from measurements are shown.

B.4.1 Correlating the data

First the accelerometer data from the lower arm Myo armband is shown in **Figure B.4**. This data is used for correlating the Kin-Com and Myo Band data. Since the gravitational acceleration measured in the armband is angle dependent it looks very similar to the joint angle data shown in **Figure B.5a**. That is why the accelerometer data and the joint angle data is used to align the Kin-Com and Myo Band data. Notice the minor spaces between the data lumps; this is done purely for the reader to notice the data representing the three extend exercises followed by the three flex exercises.

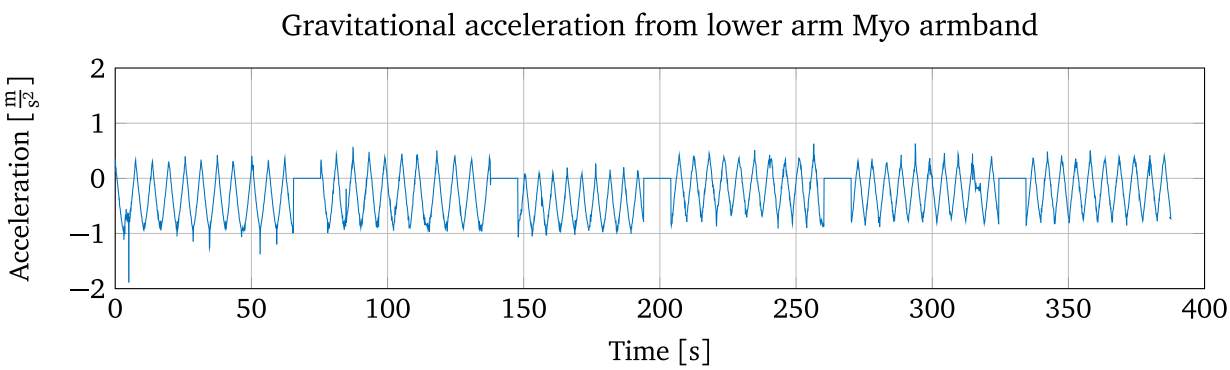
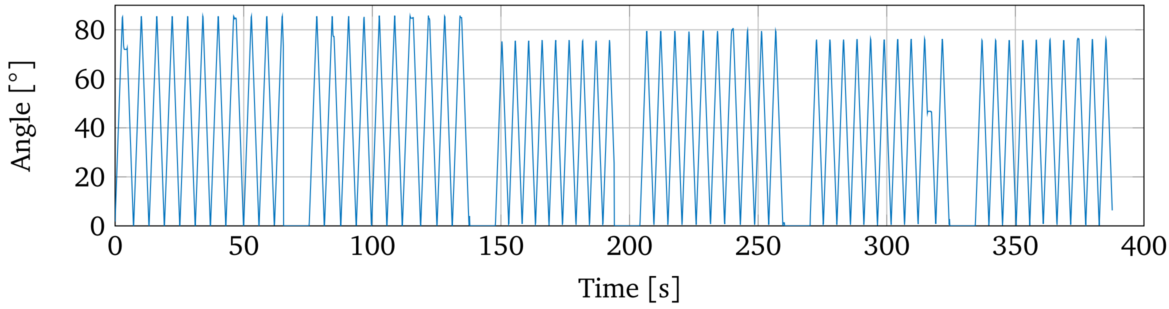


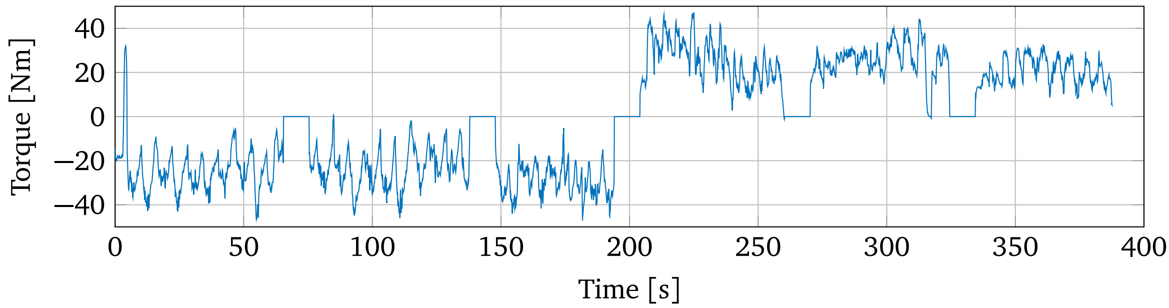
Figure B.4

B.4.2 Measurement Data for Parameter Estimation

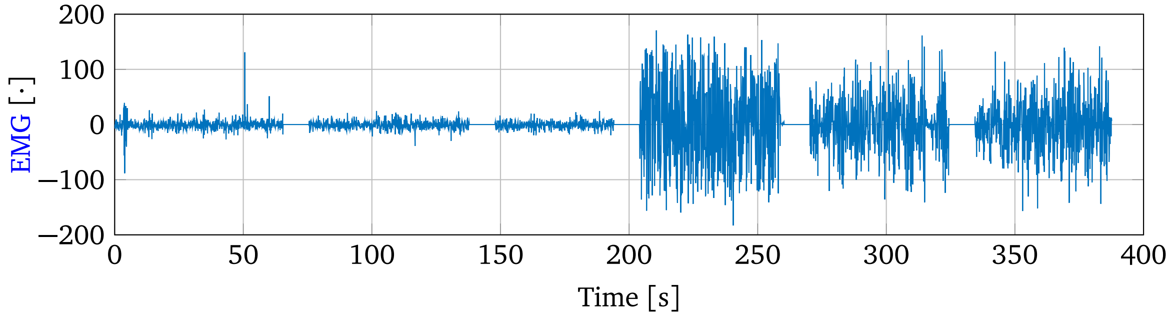
The Kin-Com and Myo armbands data for parameter estimation are shown in the following graphs **Figure B.5a**, **Figure B.5b**, **Figure B.5c**, **Figure B.5d**, **Figure B.6a** and **Figure B.6b**. Note that upper arm POD1 and POD8 EMG data are very similar and therefore both capture the Triceps Brachii EMG very well:



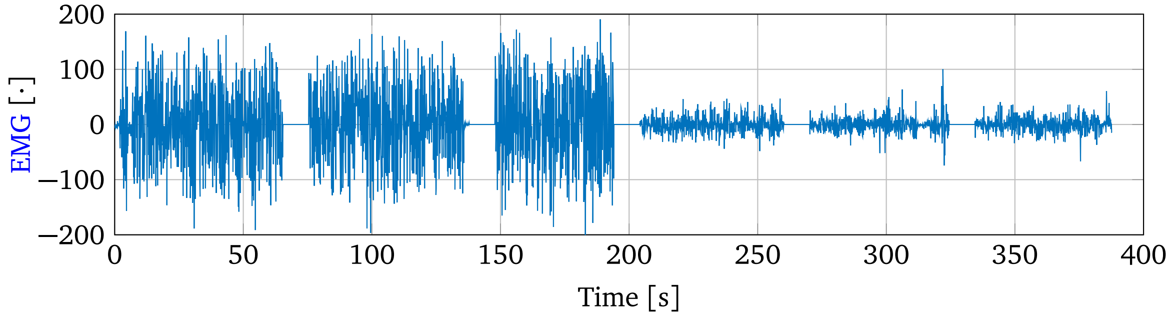
(a) *Elbow Angle measured by Kin-Com*



(b) *Elbow Torque measured by Kin-Com*

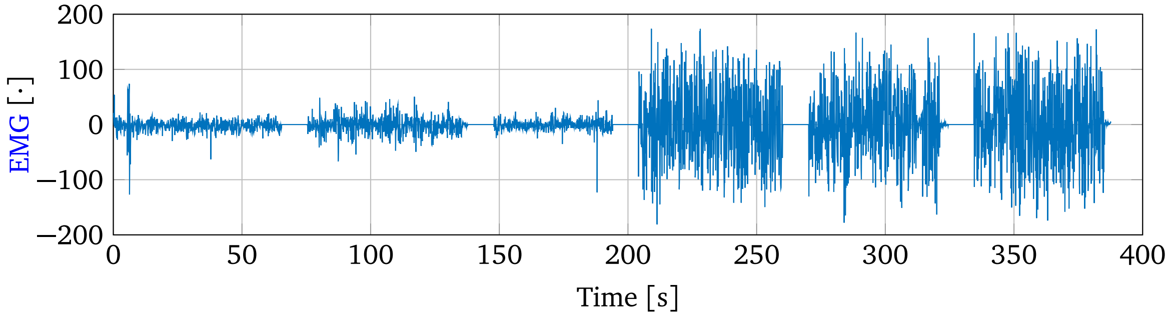


(c) *POD4 EMG measured by lower Arm Myo Armband*

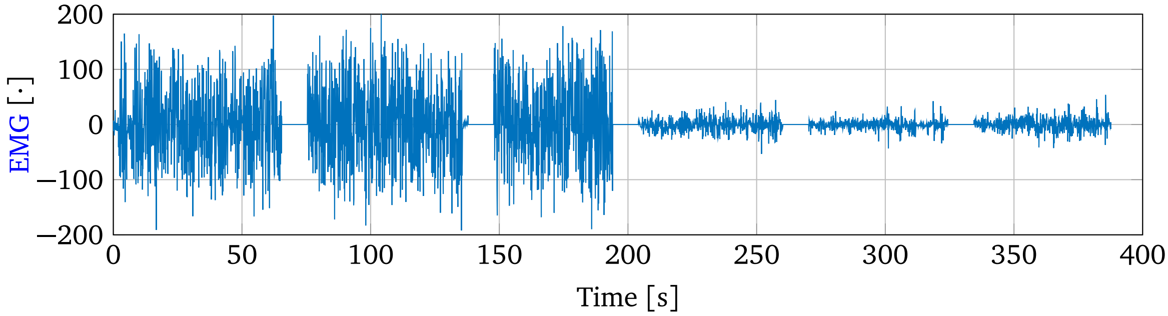


(d) *POD1 EMG measured by upper Arm Myo Armband*

Figure B.5



(a) *POD4 EMG measured by upper Arm Myo Armband*



(b)

Figure B.6: *POD8 EMG measured by upper Arm Myo Armband*

PyMyo C

For the purpose of this project a small *Python* module, named PyMyo, for the Myo Band has been implemented. It was implemented partially due to lack of official Linux support, and to be able to connect to multiple Myo Bands simultaneously.

Thalmic Labs has released a BLE specification [1], describing the protocol they use to communicate with the Myo Band in their official libraries. The specification consists of a *C*-header file, defining a number of structures and enums. To map these to a *Python* representation two different implementations of a Foreign Function Interface have been considered; the standard module *ctypes* and *CFFI* from the people behind *PyPy*.

CFFI was chosen as it is faster than *ctypes* (in runtime), and the developers of *CFFI* try to not impose limits on the user's interpreter. This is a part of the goal of the project: "Attempt to support both PyPy and CPython, with a reasonable path for other Python implementations like IronPython and Jython."[2], whereas *ctypes* only supports *CPython*.

To communicate with the Myo Band the BLE interface provided by bluepy [3] is used.

C.1 Implementation

When a new PyMyo object is created, it scans for BLE devices, and if a Myo Band is found, and attempts to connect to it. With a successfully connected Myo Band the method `enable_services(...)` can be called. It will send a BLE request to get a UUID for all available services, and enable services based on the arguments passed to the method. When a service is enabled all the characteristics of the service are enabled. Once services have been enabled a callback function is set, which will be invoked every time a new BLE packet is received from the Myo Band.

[1] https://github.com/thalmiclabs/myo-bluetooth
[2] https://cffi.readthedocs.io/en/latest/#goals
[3] http://ianharvey.github.io/bluepy-doc/

C.2 Usage

The API provided by PyMyo is described blow, with a small example following

- `constructor` - The constructor takes 3 optional arguments;
  - `iface` - The ID of the interface to use (`iface`). Defaults to to 0.
  - `on_emg` - A callback function that is invoked whenever a EMG measurement is received. Defaults to a small pretty print function.
  - `on_imu` - A callback function that is invoked whenever a IMU measurement is received. Defaults to a small pretty print function.
- `connect` - Scans for devices and connects to a Myo Band (provided at least one was found).
- `enable_services` - This method takes 3 arguments, and enables services on the Myo Band accordingly
  - `emg_mode` - Enum value from the BLE specification. Defaults to `emg_mode_send_emg`.
  - `imu_mode` - Enum value from the BLE specification. Defaults to `imu_mode_none`.
  - `classifier_mode` - Enum value from the BLE specification Defaults to `classifier_mode_disabled`. (Not supported)
- `set_sleep_mode` - Enables or disables automatic sleep the Myo Band.
- `waitForNotifications` - Inherited from bluepy. Yield the thread until a BLE notification is received.

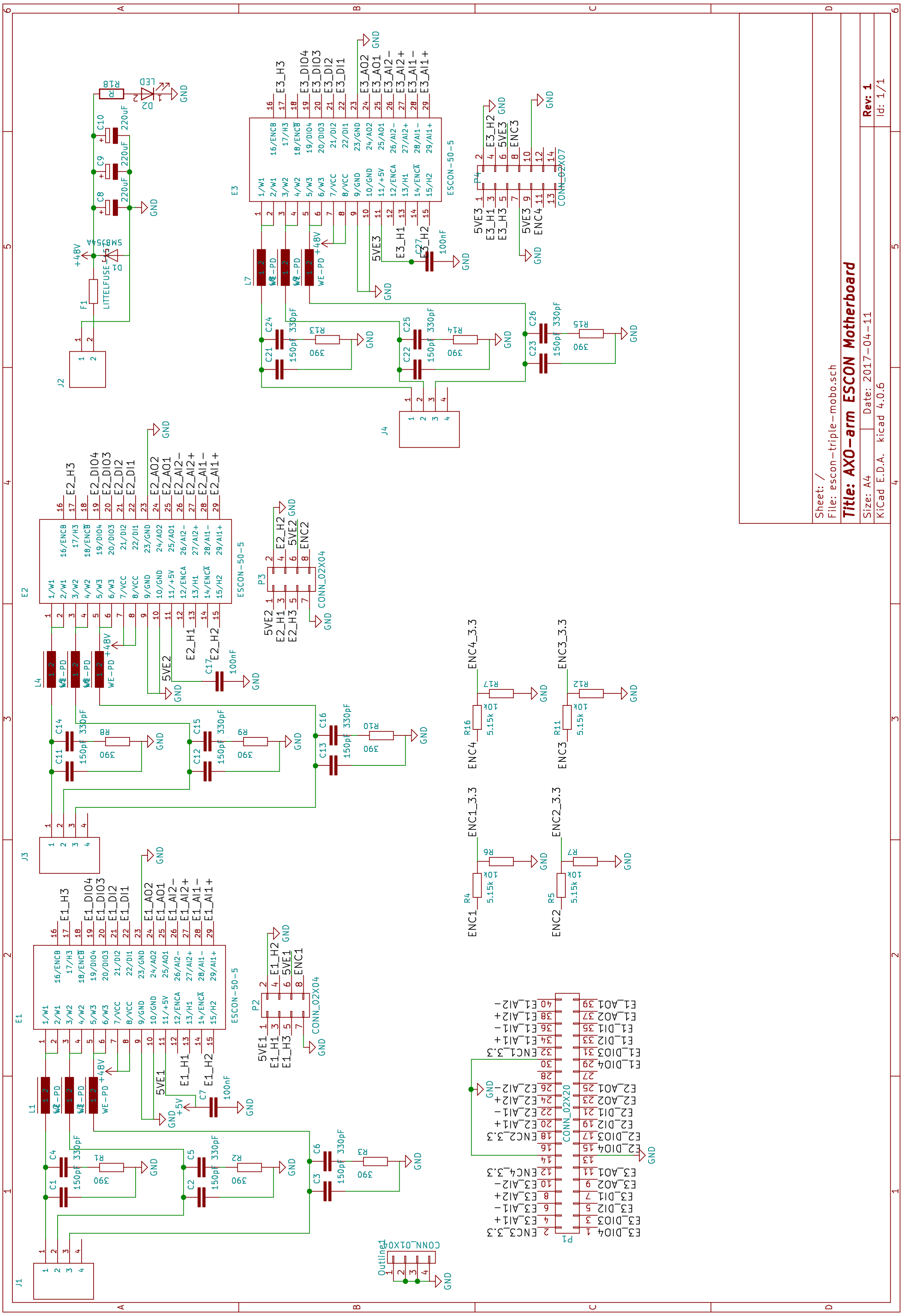
A small example with a simple print function as callback for both EMG and IMU is shown below.

```
on_measurement = lambda measurement: print(measurement)
m = PyMyo(iface='0', on_emg=on_measurement, on_imu=on_measurement)
m.connect()

m.enable_services()

while True:
  m.waitForNotifications()
```

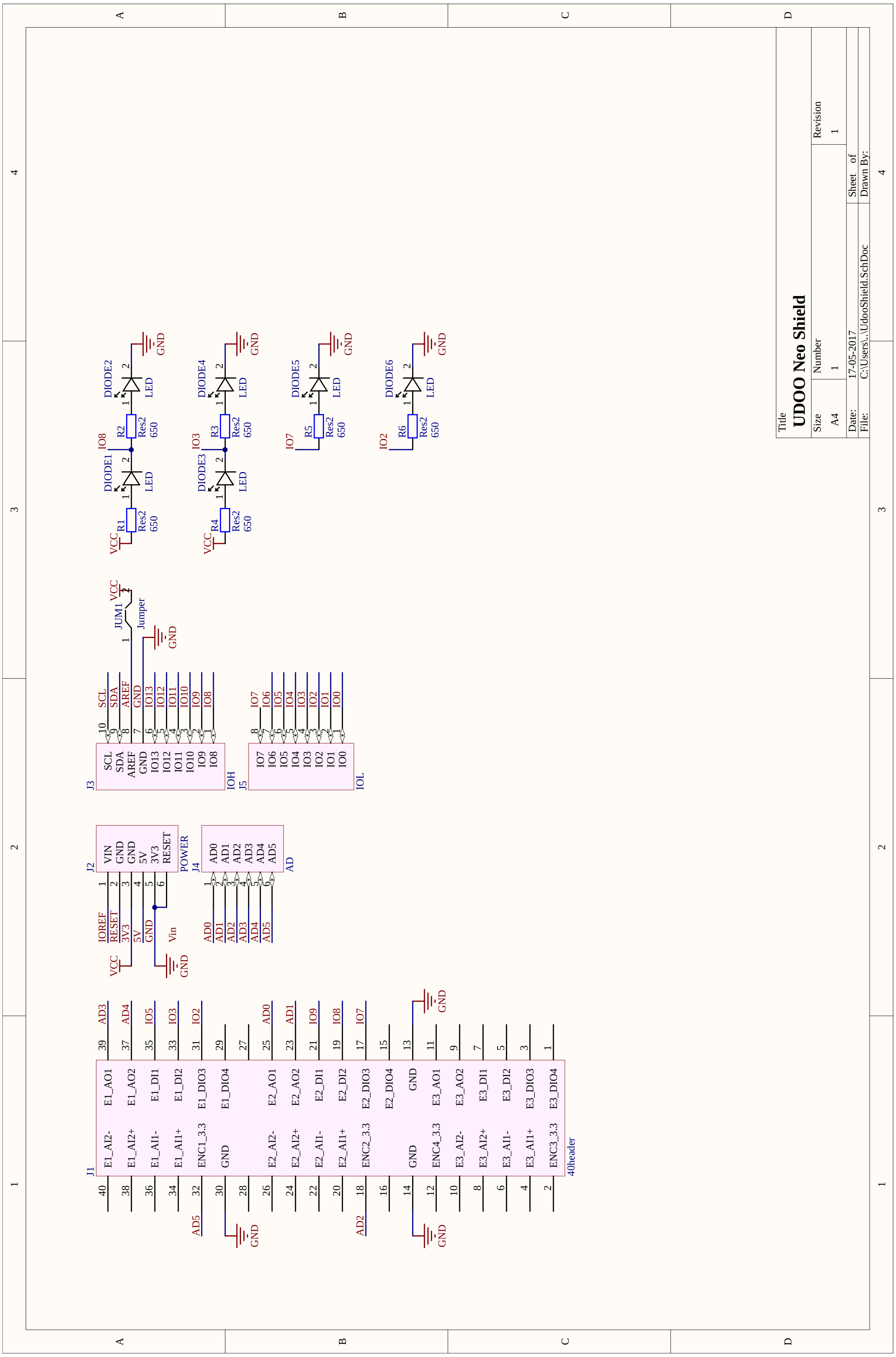
PyMyo is freely available as beerware (i.e. under beer license), and can be found on github, at https://github.com/lbromo/PyMyo.

PCB Diagrams D

Triple ESCON 50-5 Motherboard



UDOO Neo Shield



Software E

As mentioned the software is running on two different cores; a real-time Cortex M4, and an A9 application core. In the following sections the software on each core is described, and finally the inter-core communication protocol is designed to enable message passing between the cores.

E.1 Real Time Software

The software that runs on the real time (M4) core on the UDOO consists of 2 main tasks. One that handles communication with the Linux core, and one that runs the feedback linearized controller. Both the controller and communication task runs with a frequency of 200 Hz. They are setup using Message Queue eXecutive (MQX), which provides an interface for setting up hardware timers with μs resolution, that invoke a callback function.

The two cores communicate with each other using a hardware serial connection. The development team at UDOO did develop an internal multi-core communication module that runs in software, however, there is a known issue with it; it can become slow if the two cores are not synced up, and in some cases completely hangs. A workaround has been suggested; if the A9 core is actively reading when the M4 core sends a message, it should be faster. It does speed things up a bit, but there are still issues with the internal communication [1]. Therefore a hardware serial connection is chosen. Even though a hardware serial connection was chosen in the end, the message protocol between the two cores is defined with the limitations in mind, í.e. designed as a request-reply pattern. The details are are shown in **Section E.3**.

E.2 Linux Program

On the A9 cure the muscle model and admittance controller is running. A high-level overview of the implementation is shown on **Figure E.1**.

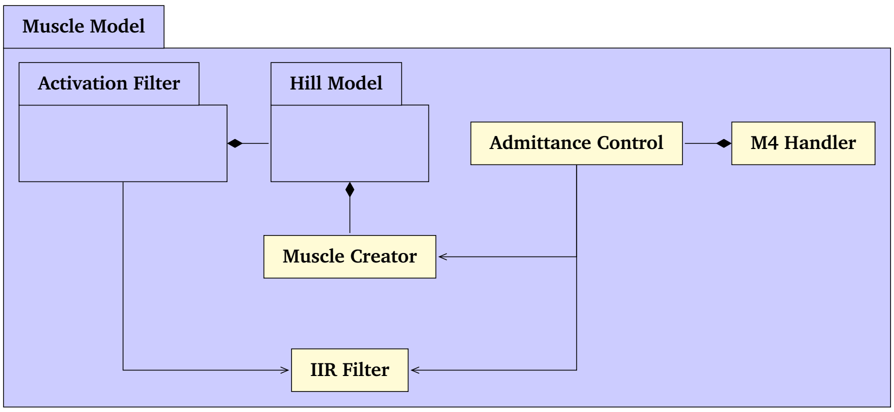


Figure E.1: *Overview of the implementation.*

[1]See issue 14: https://github.com/UDOOboard/linux_kernel/issues/14

The *Activation Filter* (**Figure E.2**) handles the communication with the Myo Band, through PyMyo (**Appendix C**). This is done by the observable *EMG* object, where each *Activation Filter* can subscribe to be notified whenever a new EMG measurement is received. The *Activation Filter* internally filters the EMG vector, first by selecting the appropriate Myo Band pod, using the *enum* class *EMGPOD*, and then running the filters (via *IIR filter*), adding new measurements, and filter outputs to the deques.

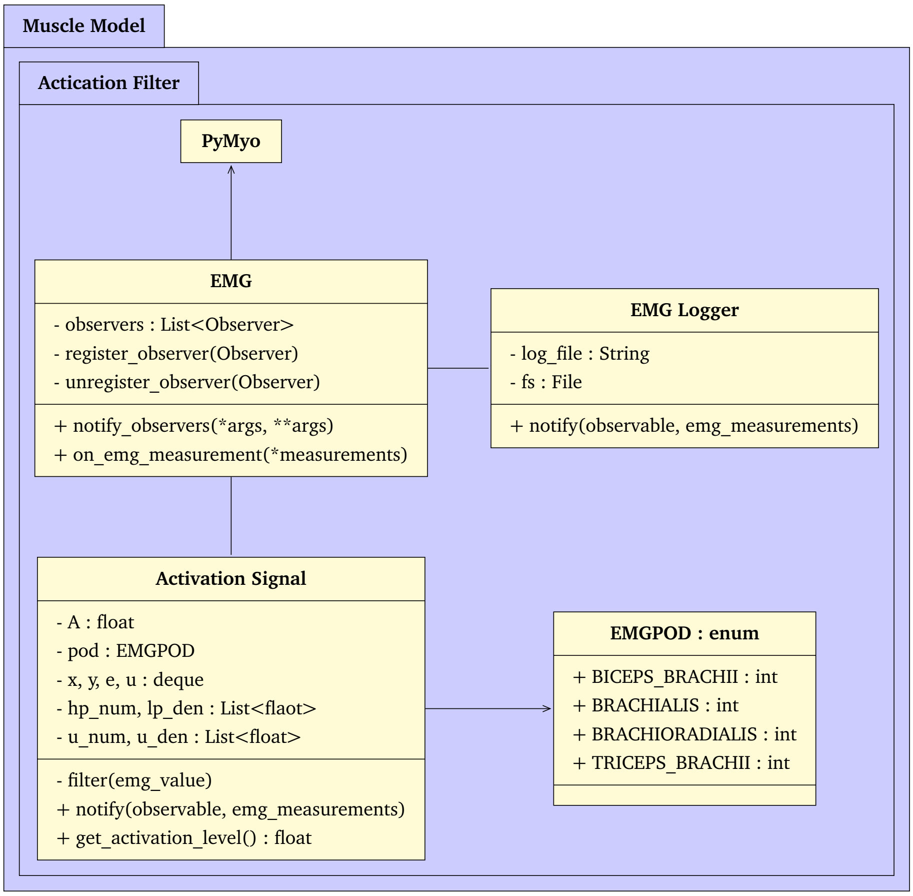


Figure E.2: *Implementation of the Activation Filter.*

The *Hill Muscle* (**Figure E.3**) contains an *Activation Signal*, a *MUSCLE NAME* together will the estimated parameter (for the given *MUSCLE NAME*). It can be used to estimate both the force and torque generated by the muscle; both tasks the current angles on the joints, and the torque estimate further requires the joint of which the torque should be estimated. From the angles and joint the muscle length and moment arm are estimated (via. *Muscle Utils*), and then the Hill Model is used to estimate the force. As multiple muscles are needed to be able to control the arm, a creational pattern is used. The *Muscle Creator* takes a file of estimated parameters, and creates the appropriate *Activation Filters* and *Hill Models*. This is used by *Admittance Control*.

The *Admittance Control* estimates the joint torque and translates those to velocity and position references (using the *IIR filter*), which are passed to the *M4 Handler*.

The *M4 Handler* sends references to the real-time core, and logs the measurements that are sent from the real-time core to a buffered file. A buffered file was chosen to reduce the time spent on disk I/O. It buffers the input until either of the following three events occurs; 4KB data is in the buffer, *flush* is called or *close* is called. On any of these three events the buffer is written to the disk, and cleared. The same buffer used to get the current angles of the AXO-arm, this is all done in a separate thread.

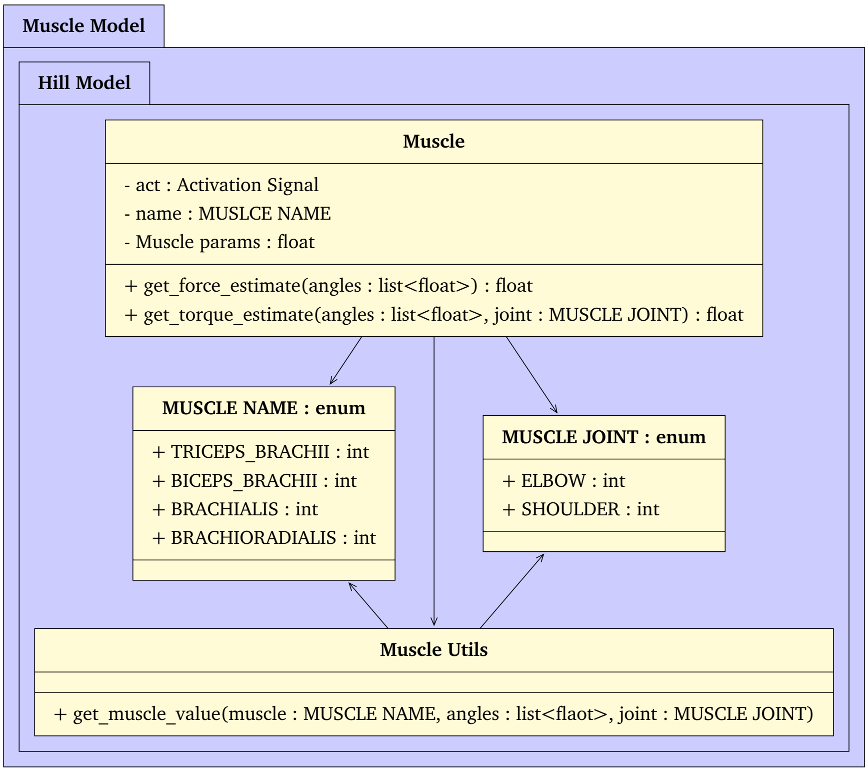


Figure E.3: *The implementation of the Hill Model*

E.3 Inter Core Communication Protocol

The inter core communication protocol have been designed with the limitations of the UDOO Neo in mind – it has been designed as a request-response pattern, where the M4 core doesn't transmit anything, before getting asked by the A9. This is due to issues with the implementation of the Multi Core Communication module.

The following messages have been defined:

New reference:
The A9 core can send a new reference message at any moment. No response is allowed from the M4 core. The message is defined as:
R<θ_{1_d}>,<θ_{2_d}>,<$\dot{\theta}_{1_d}$>,<$\dot{\theta}_{2_d}$>.

Measure:
The A9 core requests a new measurement, and the M4 send measurements.

A9 requests - M,

M4 response - S,<time>,<shoulder id>,<θ_{1_d}>,<$\dot{\theta}_{1_d}$>,<integrated error>, <θ_1>,<$\dot{\theta}_{1_d}$,<shoulder current>,<shoulder pwm value>,<shoulder pwm direction>,<elbow id>,<θ_{2_d}>,<$\dot{\theta}_{2_d}$>,<integrated error>, <θ_2>, <$\dot{\theta}_{2_d}$,<elbow current>,<elbow pwm value>,<elbow pwm direction>

Each floating point number, that is all except time, joint ids, pwm values and pwm directions are multiplied by 100 and cast to an int before sending the message. This have been chosen as transmitting floating point numbers sent over a serial port is not trivial.