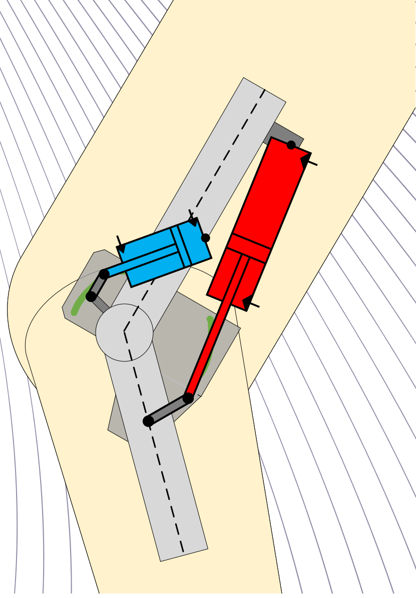


A Comparative Simulation Study of Reactive Control and Model Predictive Control for Discrete Hydraulic Actuation of Knee Exoskeleton

Master's Thesis

10th Semester Project

Emil Plovmand Munk

AALBORG UNIVERSITY
STUDENT REPORT

Department of Energy Technology
Aalborg University
http://www.aau.dk

Title:
A Comparative Simulation Study of Reactive Control and Model Predictive Control for Discrete Hydraulic Actuation of Knee Exoskeleton

Theme:
Master Thesis

Project Period:
Spring semester 2022

Project Group:
MCE4

Participant:
Emil Plovmand Munk

Supervisor:
Torben Ole Andersen

Copies: 1

Page Numbers: 73

Date of Completion:
May 27, 2022

Abstract:

This project seeks to develop a reactive- and model predictive control strategy to control a discrete hydraulically actuated knee exoskeleton with the goal of computational- and energy efficiency, when controlling the knee angle during a gait cycle. For developing the controller designs, models for the exoskeleton, human leg and ground reaction force are derived. The reactive controller is implemented as a PD-controller with torque compensation and torque feed-forward, in combination with a force switching algorithm. Torque compensation is also used with the model predictive controller (MPC), which implements a linearized prediction model to find the optimal inputs who minimize the value of a cost function designed to penalize tracking error and energy consumption. Simulative results show that the MPC achieves approximately twice as good tracking precision as the reactive controller, with position and speed rms errors of 0.63° and 0.19 rad/s over a whole gait cycle. The MPC is also shown to be more energy efficient and less noise sensitive.

Preface

Since no prototype of the system was available within the given time frame for writing this thesis, no experimental work could be included. Therefore, the project focuses solely on simulative controller evaluation with the purpose of delivering two controller algorithm suggestions that can be implemented and tested on the prototype by the researchers of Linz University when the prototype is ready for testing.

Aalborg University, May 27, 2022

Instructions for reading

The report is written in LaTeX, and each chapter is marked with a certain number, and is divided into sections. All the references used throughout the report are indicated by the method referred to as the Institute of Electrical and Electronics Engineers (IEEE). The bibliography is made in Mendeley and BibTeX, and the citations used throughout the sections are noted in the text either at the beginning of a section, or as each individual statement is made. Citations of figures and tables are mentioned in the caption.

Emil Munk

Emil Plovmand Munk

Summary

In the context of exoskeletons, hydraulics is a method of actuation that has not been subject to a lot of research. The high power density of hydraulic actuators means that low peripheral mass can be achieved when used for exoskeleton actuation. Researchers at Linz University and Linz Center of Mechatronics have designed a novel concept for a hydraulically actuated knee exoskeleton, for which this Master's thesis seeks to develop energy efficient controller algorithms that are fit for real-time implementation.

The exoskeleton system is actuated by two conventional dual-chamber hydraulic cylinders, configured mechanically in a way that makes the actuation system equivalent to a multi-chamber cylinder. The four cylinder chambers are connected to a high and low pressure rail, which gives 16 discrete actuator force levels, serving as input to the system. In the project, a model of the system is presented, consisting of the hydraulic actuator model, and a mechanical model of the human leg. Furthermore, a model for the ground reaction force is derived, in order to simulate the controlled response during a gait cycle, with the goal of tracking the knee angle while delivering the required knee torque.

Two controller approaches are developed: a reactive controller and a model predictive controller. The project goal is to evaluate the controllers through benchmarking. The reactive controller is implemented as a PD-controller with torque compensation and torque feed-forward. For converting the continuous controller output signal to an appropriate actuator force level index, a force switching algorithm (FSA) is chosen through a comparative analysis of two designs. The model predictive controller (MPC) uses a linearized prediction model in combination with an optimization algorithm to find the optimal inputs on a prediction horizon who minimize the value of a cost function designed to penalize tracking error and energy consumption. Both controller approaches assume a modification of the knee exoskeleton system, which allows for measuring the knee torque in order to implement torque compensation when the disturbance of the ground reaction force is acting.

For the simulative controller evaluation, noise is implemented on the controller signals. The MPC achieves approximately twice as good tracking precision as the reactive controller, with position and speed rms errors of 0.63° and 0.19 rad/s over the whole gait cycle. Less energy is also consumed by the MPC, which also utilizes significantly fewer force level switches during the gait cycle. Furthermore, analysis indicates that the MPC is less noise sensitive. Thereby, it is concluded that the MPC hast the best performance of the two controller structures, with the only draw-back of a higher processing time than for the reactive controller structure. The processing time of the MPC is recorded to be 0.3 ms, but whether this is fast enough for real-time implementation is left for testing on the microprocessor chosen for controlling the prototype. Finally, an analysis shows that neglecting the torque compensation has a severely negative impact on both controller performances, why it is concluded that large benefit is to be drawn from having access to measurements of the knee torque.

Contents

Nomenclature

Latin Symbols

J_1	Inertia of foot	[kg m^2]
J_2	Inertia of shank	[kg m^2]
l_1	Length of foot	[m]
l_2	Length of shank	[m]
m_1	Mass of foot	[kg]
m_2	Mass of shank	[kg]
$\frac{\mathrm{d}z_1}{\mathrm{d}\psi}$	Transmission ratio of AB-cylinder	[m/rad]
$\frac{\mathrm{d}z_2}{\mathrm{d}\psi}$	Transmission ratio of CD-cylinder	[m/rad]
$\underline{\dot{x}}_{\mathrm{CM1}}$	Velocity vector for CM_1 in global reference frame	[m]
$\underline{\dot{x}}_{\mathrm{CM2}}$	Velocity vector for CM_2 in global reference frame	[m]
$\underline{\mathrm{F}}$	Sorted vector of discrete actuator force levels	[N]
$\underline{\underline{\mathcal{J}}}$	Inertia matrix	[kg m^2]
$\underline{\underline{\mathrm{A}}}$	State space system matrix	[-]
$\underline{\underline{\mathrm{C}}}$	State space output matrix	[-]
$\underline{\underline{\mathrm{p}}}_{\mathrm{comb}}$	Sorted matrix of chamber pressures corresponding to discrete actuator force levels	[Pa]
$\underline{\underline{\mathrm{R}}}$	Rotation matrix	[-]
$\underline{\underline{\mathrm{u}}}$	Sorted matrix of binary chamber pressure values corresponding to discrete actuator force levels	[-]
$\underline{\underline{B}}_{\mathrm{F}}$	Ground reaction force matrix	[m]
$\underline{\underline{B}}_{\mathrm{M}}$	Joint torque sign matrix	[-]
$\underline{F}_{\mathrm{GRF}}$	Ground reaction force vector	[N]
$\underline{M}_{\mathrm{G}}$	Joint torque vector	[Nm]
$\underline{r}_{\mathrm{CM1}}$	Position vector for CM_1 in local reference frame	[m]
$\underline{r}_{\mathrm{CM2}}$	Position vector for CM_2 in local reference frame	[m]

$\underline{x}_{\mathrm{k}}$	Position vector for knee in global reference frame	[m]
$\underline{x}_{\mathrm{G1}}$	Position vector for G_1 in global reference frame	[m]
$\underline{x}_{\mathrm{G2}}$	Position vector for G_2 in global reference frame	[m]
$\underline{x}_{\mathrm{G3}}$	Position vector for G_3 in global reference frame	[m]
S	Cost function value	[-]
$\underline{\mathrm{B}}$	State space input vector	[-]
$\underline{V}_{\mathrm{f0}}$	Zero vector field	[Nm]
$\underline{u}$	Row input vector of $\underline{\underline{\mathrm{u}}}$	[-]
A	Area	[m^2]
CM1	Foot's center of mass	[-]
CM2	Shank's center of mass	[-]
CR	Cross-over coefficient for the DE-algorithm	[-]
E_{in}	Supply energy	[J]
E_{kin}	Kinetic energy	[J]
E_{pot}	Potential energy	[J]
$e_{\dot{\phi}2,\mathrm{rms}}$	Rms speed error over one gait cycle	[rad/s]
$e_{\dot{\phi}2}$	Speed error	[rad/s]
$e_{\phi2,\mathrm{rms}}$	Rms position error over one gait cycle	[rad]
$e_{\phi2}$	Position error	[rad]
E_{sh}	Energy loss at valve switch	[J]
$e_{M\mathrm{GRF,rms}}$	Rms ground reaction torque error over one gait cycle	[Nm]
F	Mutation coefficient for the DE-algorithm	[-]
F_{AB}	Hydraulic actuator force of AB-cylinder	[N]
F_{CD}	Hydraulic actuator force of CD-cylinder	[N]
F_{hyd}	Hydraulic actuator force	[N]
F_{xG}	x-component of ground reaction force	[N]
F_{yG}	y-component of ground reaction force	[N]
F_{b}	FSA2 force range constant	[N]

g	Gravitational acceleration	[m/s^2]
$G_F(s)$	Force transfer function	[-]
$G_M(s)$	Torque transfer function	[-]
$G_{PD}(s)$	PD-controller transfer function	[-]
$G_s(s)$	System transfer function	[-]
$G_{\phi 2}(s)$	Position transfer function	[-]
G1	Foot joint	[-]
G2	Shank joint	[-]
G3	Knee joint	[-]
K_D	PD-controller derivative gain	[-]
K_{ext}	Actuator torque linearization constant	[1/(kg m^2)]
K_{hyd}	Actuator torque linearization constant	[1/(kg m^2)]
$k_{idx,prev}$	Row index of $\underline{\underline{u}}$ at previous time step	[-]
k_{idx}	Row index of $\underline{\underline{u}}$	[-]
K_P	PD-controller proportional gain	[-]
M_{ext}	External knee torque	[Nm]
M_{G1}	Foot torque	[Nm]
M_{G2}	Shank torque	[Nm]
M_{G3}	Knee torque	[Nm]
M_{GRF}	Ground reaction knee torque	[Nm]
M_{hyd}	Hydraulic actuator knee torque	[Nm]
$n_{F,sw}$	Number of force level switches	[J]
NH	Number of time steps on the prediction horizon	[-]
NP	Population number for the DE-algorithm	[-]
p	Pressure	[Pa]
p_{nom}	Nominal valve pressure drop	[m^3/s]
Q	Flow	[m^3/s]
Q_{nom}	Nominal valve flow	[m^3/s]

t	Time	[s]
T_s	Sample period	[s]
$v_{x,CM1}$	x-direction speed of CM_1 in global reference frame	[m]
$v_{x,CM2}$	x-direction speed of CM_2 in global reference frame	[m]
$v_{y,CM1}$	y-direction speed of CM_1 in global reference frame	[m]
$v_{y,CM2}$	y-direction speed of CM_2 in global reference frame	[m]
W	FSA1 force shift penalty constant	[N]
W_1	Position error weighing factor in S	[-]
W_2	Speed error weighing factor in S	[-]
W_3	Force level switching energy consumption weighing factor in S	[-]
$W_{G1,angular}$	Angular work on G_1	[J]
$W_{G1,linear}$	Linear work on G_1	[J]
W_{G2+G3}	Work on G_2 and G_3	[J]
x_K	x-position of knee	[m]
x_{CM1}	x-position of CM_1 in global reference frame	[m]
x_{CM2}	x-position of CM_2 in global reference frame	[m]
y_K	y-position of knee	[m]
y_{CM1}	y-position of CM_1 in global reference frame	[m]
y_{CM2}	y-position of CM_2 in global reference frame	[m]
z_1	Piston position of AB-cylinder	[m]
z_2	Piston position of CD-cylinder	[m]
$z_{1,max}$	Maximum piston position of AB-cylinder	[m]
$z_{2,max}$	Maximum piston position of CD-cylinder	[m]

Greek Symbols

α_{oil}	Air content in oil	[%]
β	Bulk modulus	[Pa]
β_0	Constant bulk modulus	[Pa]
β_{eq}	Total viscous friction constant	[Nms]

$\ddot{\phi}_1$	Foot angular acceleration	[rad/s^2]
$\ddot{\phi}_2$	Foot angular acceleration	[rad/s^2]
$\dot{\phi}_1$	Foot angular velocity	[rad/s]
$\dot{\phi}_2$	Shank angular velocity	[rad/s]
η_{CM1}	y-location of CM_1 in local reference frame	[m]
η_{CM2}	y-location of CM_2 in local reference frame	[m]
κ	Polytropic constant of air	[-]
ϕ_1	Foot angle	$[degree]$
ϕ_2	Shank angle	$[degree]$
ψ	Knee angle	$[degree]$
τ_{lp}	PD-controller low pass filter time constant	[-]
ζ_{CM1}	x-location of CM_1 in local reference frame	[m]
ζ_{CM2}	x-location of CM_2 in local reference frame	[m]
s_β	Pressure related slope of bulk modulus	[-]

Superscripts

$*$	Reference

Subscripts

k	Discrete time step k
A	Ankle
H	Hip
K	Knee
T	Toe

Most common acronyms

DE	Differential Evolution
FSA	Force Switching Algorithm
GRF	Ground Reaction Force
GRFM	Ground Reaction Force Model
MPC	Model Predictive Control

1 Introduction

1.1 Motivation

As a result of the aging society, the need for wearable robotics is increasing. The most important requirement for these devices, also called exoskeletons, is wearing comfort which is achieved through the design of the mechanical structure which should be light weight, compact and emulate human motion closely [1]. In order to enable this, an efficient drive system is an important factor. This project concerns itself with the control of a digital hydraulic actuated knee exoskeleton designed by a research group at Johannes Kepler University Linz and Linz Center of Mechatronics. The scope of the project is to implement and evaluate computationally- and energy efficient control algorithms, who can mimic the gait cycle of a human knee.

1.2 Exoskeletons

It is expected that there is a significant market for exoskeletons but right now there are not any widely available commercial solutions, although there are many examples of exoskeletons for both rehabilitation, military and everyday use [1]. An example of a knee exoskeleton device is the BoostX developed by SuitX, which can be seen in Figure 1.1 [2].

Figure 1.1: BoostX for support of the knee during motion (Source: [2]).

The BoostX is electromechanically actuated, but there lies an alternative in hydraulic actuation. Historically, hydraulically actuated exoskeletons have not been commercially successful. An example of this is the HULC exoskeleton from Lockheed Martin, which was a commercial failure due to the inability of the control system to closely mimic the human gait - studies even showed that the exoskeleton tired the user more than when walking without it [1]. Lockheed Martin has since then focused on electromechanical actuation resulting in their newest exoskeleton, the ONYX [3], which is based on a concept similar to the one seen in [4]. The advantages of hydraulics in the context of exoskeletons is the high force density of hydraulic actuators, and low peripheral mass allowing for more easily achieving the needed dynamic properties of the system [1].

The general requirements for a the actuation system of an exoskeleton are [1]:

- Wearing comfort.
- System efficiency. Relates to wearing comfort, since this will result in lesser weight and size.
- Back-drivability.

It is important to have a light weight and energy dense power source which can be converted to hydraulic power. Currently there is a lack of technology which can live up to these requirements. Therefore, the best solution is currently to use batteries rather than fuel supplied combustion engines [1]. In this regard, the multi-chamber cylinder is an approach to actuation that can prove beneficial if it is controlled as a digital fluid power system [5]. Some benefits of digital fluid power are: energy efficiency, simple and reliable components, high degree of flexibility in programming of control algorithms and unification of hydraulic components potentially allowing for cheap components of high quality [6]. Some key challenges regarding the technology are: noise, pressure spikes, price and the need for complicated/non-conventional control [6].

Supplying the four chambers of the multi-chamber cylinder with discrete pressure levels supplied through several digital flow control units (DFCUs) comprised of ON/OFF valves, the actuator force can be changed by varying the combination of which valves are turned on and off. For a digitally controlled multi-chamber cylinder, there will be a number of force levels, $\mathrm{N_F}$, according to: $\mathrm{N_F} = \mathrm{N_P^{N_C}}$, where $\mathrm{N_P}$ is the number of pressure levels and $\mathrm{N_C}$ is the number of cylinder chambers [6]. For a four chamber cylinder supplied with two pressure levels, this gives 16 different valve combinations, which results in 16 different force levels. This allows for force control (also called secondary control) as an alternative to primary control which is the conventional control approach used most commonly in industry. When using primary control, throttling losses are significant when compared to secondary control since no throttling is needed to achieve the control in the case of secondary control [5], [7], [8]. In this report, the secondary control approach is what is going to be utilised for controlling the exoskeleton system.

1.3 Control of Exoskeletons: State of the Art

There are generally two approaches to the digital control of multi-chamber cylinders, that being direct force control (DFC) or model predictive control (MPC). The article [7] from

2009 was the first time a study concerning discrete control of ON/OFF valves connected to the four chambers of a multi-chamber cylinder was published. Here DFC was implemented utilising a cost term which chooses the force level closest to the force reference. A penalty term is added to this cost function to solve the problem of high frequency valve switching. This algorithm is the basis of all digital control of multi-chamber cylinders. In [9], a modified version of this algorithm is used where the penalty term in the cost function is replaced by a sleep period which ensures that no switching occurs in a small time interval after each switching. This paper also explores an energy efficient DFC which chooses between three possible force levels within a defined band around the force reference according to the least energy consumption and then chooses the one closest to the force reference.

Alternatively, model predictive control can be used to take into account that it at some time steps is more reasonable to create a larger force error to get better control in the following time steps [5]. This controller approach has been studied in [5], [10]–[15]. Although better control performance can be observed with MPC rather than with DFC [5] it is a challenge that the MPC-algorithm is more computationally demanding than the DFC [14].

Naturally, it is also desirable to implement user intention detection in the control algorithm in order to enable the control to be in alignment with what the user desires. [4] proposes a method for this using a torque transducer measuring the knee torque as well as three pressure sensors placed under the sole of the user's foot, measuring the force on the heel and toe. But in general, it can be said that designing a knee exoskeleton that works in symbiosis with how the human brain plans and follows a gait trajectory, is a challenging control task that has not seen a lot of research.

1.4 Human Gait Cycle

In order to control the exoskeleton it is important to understand the human gait cycle. An introduction to the human gait cycle will therefore be presented in this section. This report is based on human gait data from the HuMoD database [16], which is a collection of position and force data recorded from a healthy male of 32 years of age, 82 kg mass and 179 cm height. The data used for the examination of the exoskeleton in this report is in the case of straight slow walking at a velocity of 1 m/s. At this velocity, the gait cycle takes ≈1.1s. The movement of the leg during a gait cycle is plotted at a frequency of 50 Hz and shown in Figure 1.2.

Figure 1.3 shows the positioning of the foot, shank and thigh during one gait cycle at seven different time stamps. It can be noted that the figures from t(1)=0 ms → t(4) = 600 ms show the stance phase, and the ones from t(5) = 800 ms → t(7) = 1100 ms depict the swing phase. The stance phase is defined as the movement from when the ankle first touches the ground to when the toe releases the ground. The swing phase is defined as

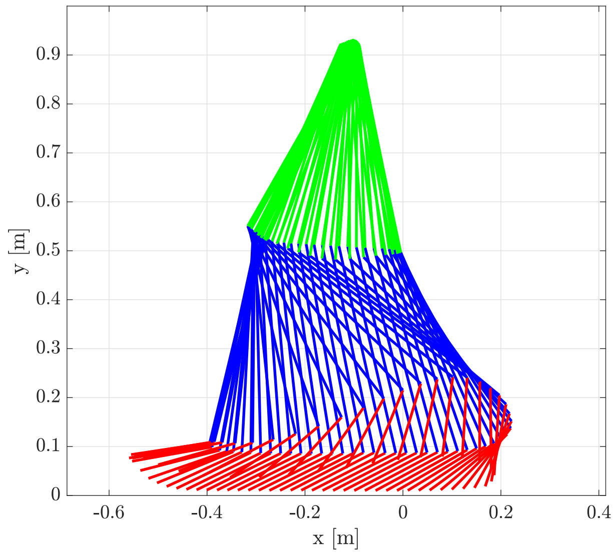


Figure 1.2: Leg movement during a gait cycle. Colors indicate: the foot (red line), the shank (blue line) and the thigh (green line).

the movement in which the leg travels through the air without ground contact, until the ankle again touches the ground. The stance phase constitutes 62% of the gait cycle [17].

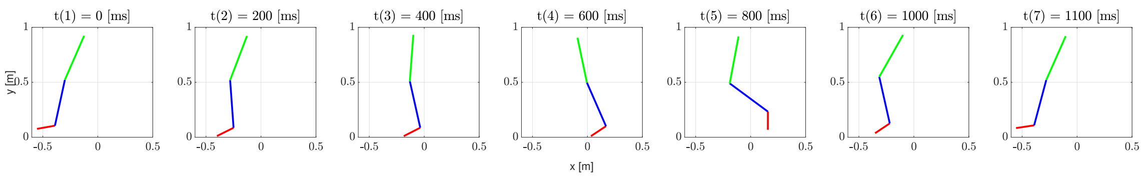


Figure 1.3: Positioning of the leg during one gait cycle depicted in seven instances. Colors indicate: the foot (red line), the shank (blue line) and the thigh (green line).

The knee torque during a gait cycle can be found using a mechanical model of the human leg (see Section 3.2) along with the HuMoD data. In Figure 1.4 the knee torque, M_{G3}, is for one gait cycle plotted against the angular position of the knee, ψ. On the figure, the color mapping indicates the time during the gait cycle which is $t \in \{0...1.1s\}$. It is worth noting that the maximum knee torque is required in the span where the knee is almost fully stretched: $\psi \in \{155°...180s\}$. This can be used for manipulating the transmission ratios of the actuators[1] to achieve maximum torque in this range [18]. This will be elaborated on in the next section.

Figure 1.5 shows an alternative plot of the data in Figure 1.4, where the knee torque and knee angle are plotted simultaneously on the y-axis against time on the x-axis. Here it can be noted that the red area from 0 ms to 682 ms constitutes the stance phase, and the

[1]Defined as the ratio between linear displacement of the actuators and the angular displacement of the knee: $\frac{dz}{d\psi}$

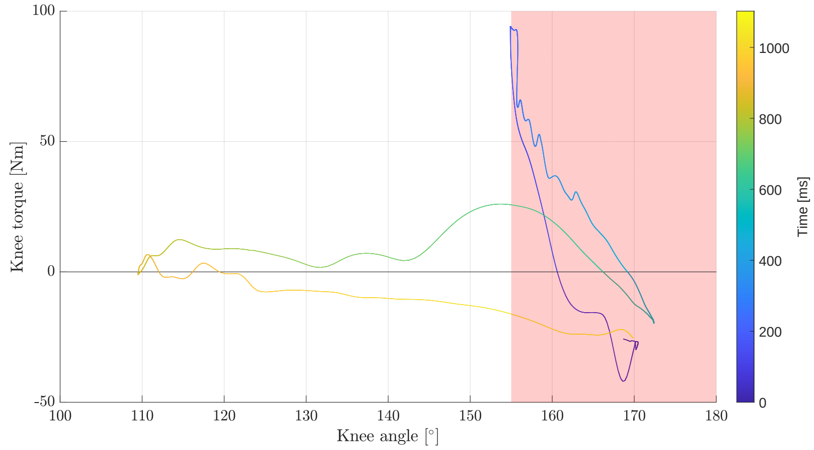


Figure 1.4: Knee torque, M_{G3}, on the y-axis versus knee angle, ψ, on the x-axis. The color bar indicates the time during the gait cycle which runs through $t \in \{0 \ldots 1.1s\}$

blue area from 682 ms to 1100 ms is the swing phase. The dark area from 150 ms to 250 ms marks the peak knee torque. In general it can be noted that the maximum knee torque is required during the stance phase.

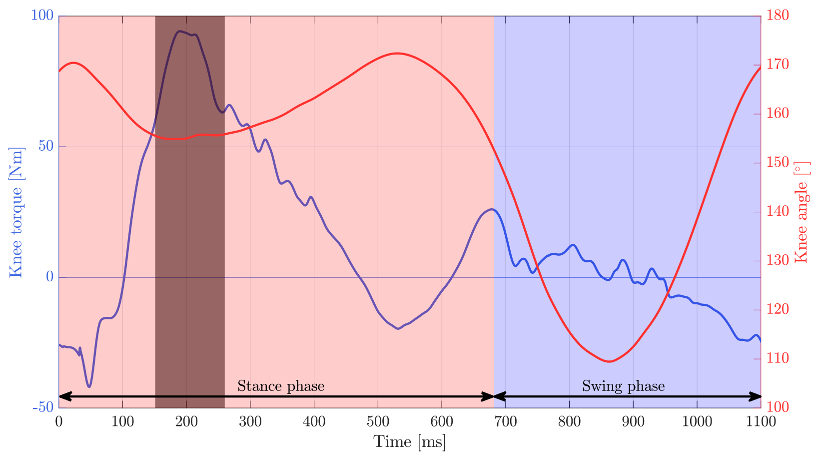


Figure 1.5: On the y-axis: Knee torque, M_{G3} and knee angle, ψ. On the x-axis: time.

1.5 System Description

The system examined in this project is the exoskeleton depicted in Figure 1.6. It consists of a mounting brace on which two identical hydraulic actuators are fixated. As seen to the left in the figure, this brace is attached to the human leg and thereby assists the user during motion.

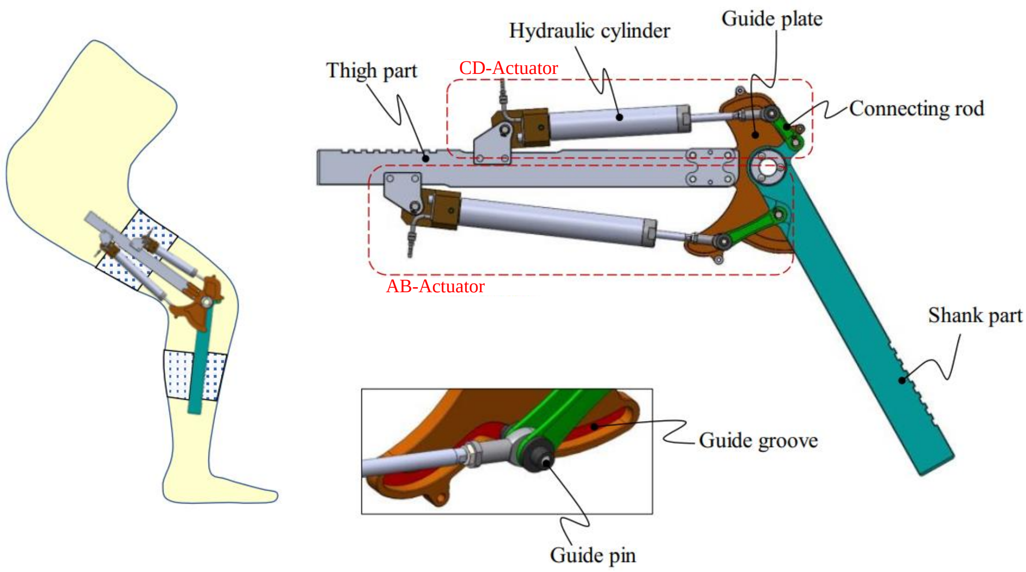


Figure 1.6: Diagram of the system (Source: [18]).

The actuation is done by two conventional hydraulic cylinders fixated in a way that lets them operate oppositely. This is seen in the AB-actuator and CD-actuator depicted in Figure 1.6. Thereby, this configuration is equivalent to a multi chamber cylinder. This configuration is desirable because it is cheaper than if a multi-chamber cylinder was used. The downside of choosing this configuration is larger size. Two carefully designed guiding grooves for the tool points of these actuators can also be seen in the bottom of the figure. These modify the transmission ratios of the actuators. The design constraint for these guiding grooves is: The transmission ratio of the AB-actuator is twice that of the CD-actuator. It is of utmost importance that the actuators are able to provide the necessary force needed by the knee during a gait cycle. Since the weight of the system is a concern, the size of the hydraulic components can be reduced by manipulating the transmission ratios. This is done by utilising the fact that the maximum torque is needed only when $\psi \in \{155°...180°\}$. [18] proposes an exoskeleton design that utilises two guide grooves, guiding the tool points of the actuators in a way that effectively ensures fulfillment of the second design criteria (the AB-actuator's transmission ratio should be twice the value for the CD actuator). Doing this, the torque procured by the actuators is thus manipulated as the knee angle changes. This can be seen in Figure 1.7. Thereby, when the transmission ratios are increased in the range $\psi \in \{155°...180°\}$, it means that the hydraulic forces will be reduced if the torque is kept constant [14]. This allows for a reduction in the size of the hydraulic components.

A technical drawing of the exoskeleton system can be seen in Figure 1.8. Each of the four chambers of the two cylinders are connected to two pressure lines using ON/OFF valves. One of the pressure lines constitutes the high pressure level, P_{HP}, and the other the low pressure level P_{LP}. For this project these two pressure levels attain the values: $P_{HP} = 200$

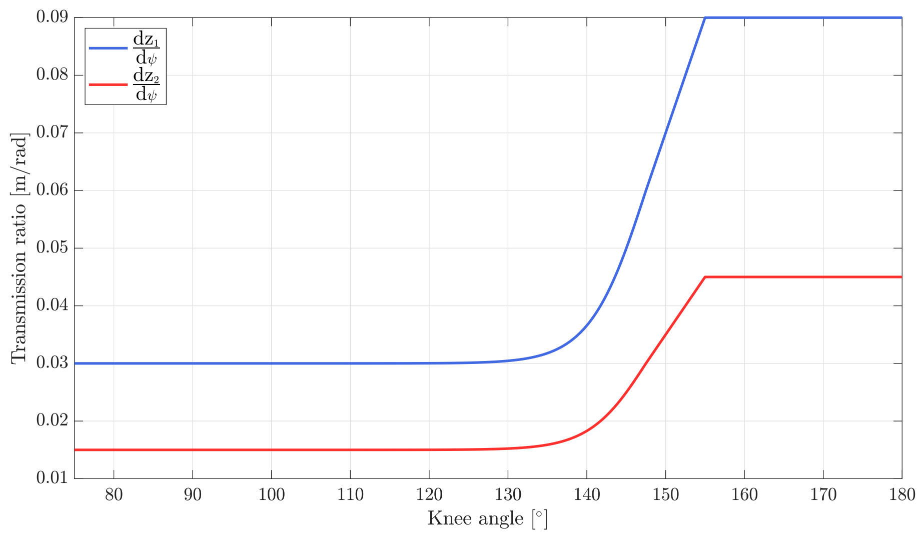


Figure 1.7: The relation between the transmission ratios plotted against the knee angle. Blue curve: for the AB-actuator, $\frac{dz_1}{d\psi}$ Red curve: for the CD actuator, $\frac{dz_2}{d\psi}$

bar and $P_{LP} = 1.01325$ bar

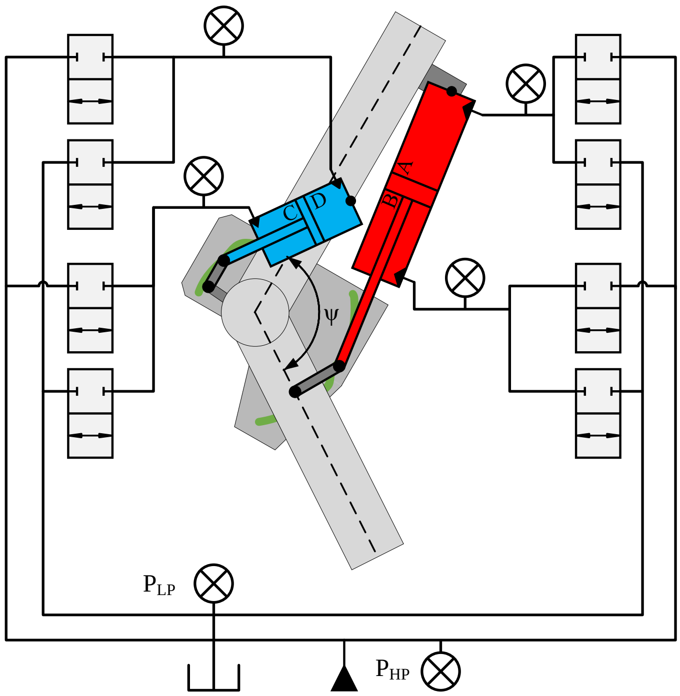


Figure 1.8: Hydraulic diagram of the exoskeleton system.

There are 16 unique configurations of high and low pressure in the four different cylinder chambers, allowing 16 different force levels. The force spectrum, along with the corresponding chamber pressures can be seen in Figure 1.9. The uniform torque steps occur due to these two criterias:

- A 4:1 ratio between bore and rod side of the actuators: $A_A = A_D = 4A_B = 4A_C$
- The transmission ratio of the AB-actuator is twice that of the CD-actuator.

The direct relationship between chamber pressures and force makes it possible to directly implement secondary control.

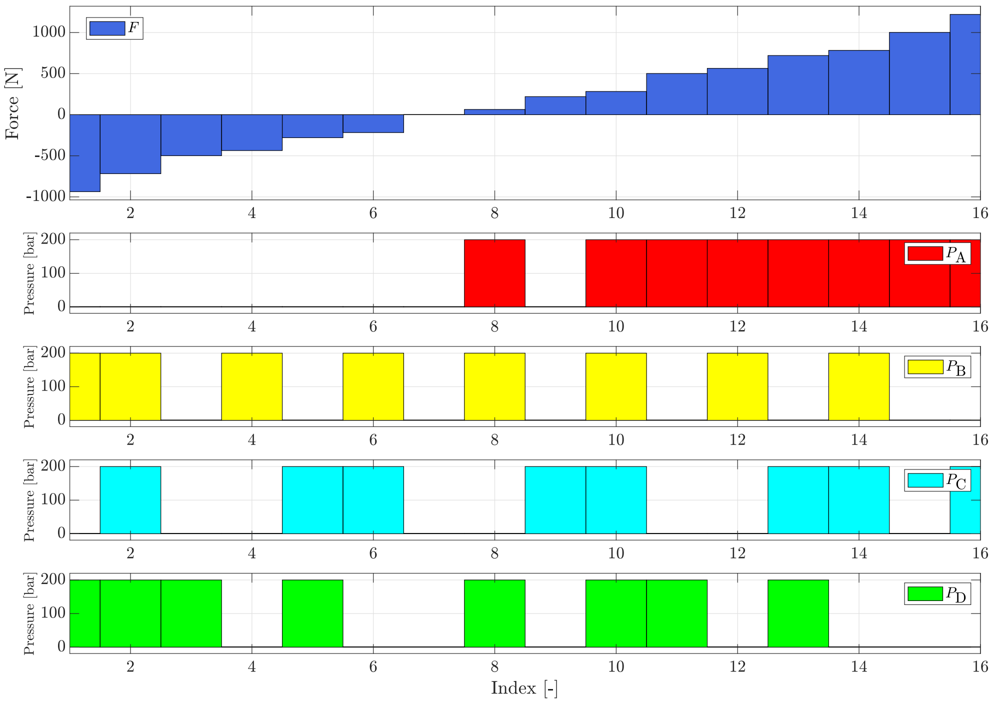


Figure 1.9: Force spectrum and corresponding pressure levels in the A, B, C and D chambers

This design makes it possible to achieve the desired torque during the gait cycle. The task is to implement a control algorithm that switches between the different force levels depicted in Figure 1.9 according to a reference force trajectory, while keeping the correct angle of the knee during the gait cycle.

2 Scope of Project

2.1 Study Objective

Design and benchmark a reactive- and model predictive control strategy with the purpose of implementation on a hydraulically actuated knee exoskeleton.

2.1.1 Project Goals

- Implement a model of the exoskeleton system and human leg.
- Develop a reactive controller and a model predictive controller.
- Evaluate the controllers' performances through simulative analysis.
- Benchmark the two controllers.

2.1.2 Design Specifications

- The case for which the control is implemented is slow walking (1 m/s) for a healthy male (82 kg and 182 cm), based on the HuMoD database.
- The control goal is, for one gait cycle, to track the knee angle with a maximum position error of $\pm 3°$, while supplying the required knee torque.
- The controllers should be computationally efficient in order to make them real-time implementable.
- The controllers are desired to be energy efficient, since this will allow the user of the exoskeleton to either go further or carry less weight.

2.1.3 Assumptions

- It is assumed that the human leg is an ideal 3-DOF planar mechanical mechanism with the ankle and knee constituting ideal revolute joints. This is in reality not true, since the movement the leg is more complex, which means that the exoskeleton might exert unwanted force on the knee joint [19], [20].
- Friction is modelled in the knee joint as being purely viscous with an equivalent friction coefficient accounting for both the friction from the hydraulic cylinders, the internal workings of the human leg and exoskeleton.
- It is assumed that measurements of the knee torque is available.
- The supply side of the system is not in the scope of the project, which means that the supply pressure is assumed to be a constant 200 bar, and the tank pressure is assumed to be a constant 1 bar.

2.1.4 Limitations

The data for the human gait cycle is taken from the HuMoD database. This data is essential for doing simulations using the mechanical model. The following database parameters are used when doing simulations: foot angle, foot speed, foot acceleration, knee position, thigh angle and hip position.

This means that when control is implemented on the system, the knee angle might deviate from the HuMoD measurement value, which means that there will be another foot position, thus resulting in a different ground reaction force than what is given by HuMoD. Therefore, it is needed to implement a ground reaction force model in order to increase the fidelity of the controlled model.

The researchers at Linz University and Linz Center of Mechatronics are currently working on building a prototype of the exoskeleton, on which experimental validation of the model as well as the designed controllers, can be performed. But due to the prototype being unavailable, it was impossible to conduct any experimental evaluation within the time range given for writing this thesis. Therefore, the work in this project focuses solely on theory and simulations. All specification data for the system is based on information given by the staff in Linz.

3 Modelling

Chapter Summary

This chapter firstly presents the hydraulic model describing the pressure and valve dynamics. Secondly, the mechanical model of the human leg is presented, which gives the exoskeleton model. And in the final part of the chapter, the system model is linearized in order to be used for controller design.

3.1 Hydraulic Model of Actuators

The exoskeleton is actuated by two cylinders in a configuration that allows them to work equivalently to a multi chamber cylinder. This is shown in Figure 3.1, with notations for the piston positions of the two actuators, z_1 and z_2, and the knee torque, M_{G3}. Specifications for the hydraulic actuators are given in Table 3.1.

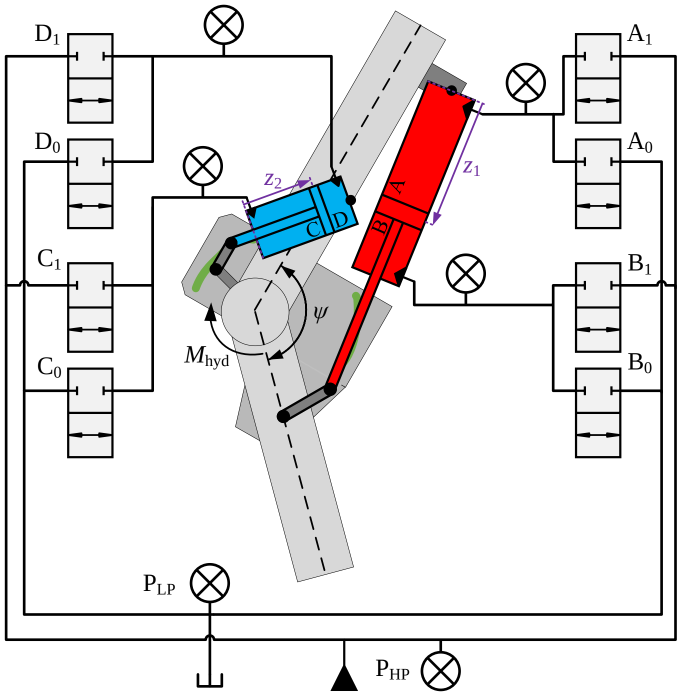


Figure 3.1: Diagram of exoskeleton.

Constant	Description	Value	Unit
$z_{1,max}$	Maximum stroke length of AB-cylinder	0.0896	[m]
$z_{2,max}$	Maximum stroke length of CD-cylinder	0.0448	[m]
$A_A = A_D$	Bore side area	$5.03 \cdot 10^{-5}$	[m^2]
$A_B = A_C$	Rod side area	$2.20 \cdot 10^{-5}$	[m^2]

Table 3.1: Hydraulic actuator parameters.

The pressure dynamics for the four chambers are modelled by the flow continuity equation:

$$\boxed{\dot{p}_\mathrm{i} = \frac{\beta_\mathrm{i}}{V_\mathrm{i}}(Q_\mathrm{i} - A_\mathrm{i}\dot{z}_\mathrm{j})} \tag{3.1}$$

Here, $\dot{p}$ is the pressure gradient, β is the bulk modulus of the fluid, V is the volume, Q is the chamber flow, A is the piston area and $\dot{z}_\mathrm{j}$ is the speed of the piston where j denotes the AB- or CD-cylinder piston head. The index i denotes the i'th chamber $\{A, B, C, D\}$. Note that the speed of the i'th chamber's piston, $\dot{z}_\mathrm{j}$, is the piston velocity corresponding to the AB and CD actuator, where the sign depends on the piston movement direction (positive sign for expanding chamber, negative sign for contracting chamber).

The chamber flows are modelled with the summation of the flows over both valves connecting each chamber, which is modelled by the orifice equation:

$$\boxed{Q_\mathrm{i} = Q_{\mathrm{i}1} + Q_{\mathrm{i}0} = \sum_{j=0}^{1} y_{\mathrm{i,j}} \mathrm{Q}_{\mathrm{nom}} \sqrt{\frac{\left|\mathrm{p_j} - p_\mathrm{i}\right|}{\mathrm{p_{nom}}}} \mathrm{sign}\left(\mathrm{p_j} - p_\mathrm{i}\right)} \tag{3.2}$$

Here, $y_{\mathrm{i,j}}$ is the normalized valve opening, $\mathrm{Q_{nom}}$ is the nominal valve flow, $\mathrm{p_{nom}}$ is the nominal valve pressure drop and $\mathrm{p_j}$ is the pressure depending on which line is connected ($\mathrm{p_{HP}}$ for j = 1 and $\mathrm{p_{LP}}$ for j = 0 and). The constants used in the valve flow model are given in Table 3.2.

Constant	Description	Value	Unit
$\mathrm{Q_{nom}}$	Nominal valve flow	1.5	[L/min]
$\mathrm{p_{nom}}$	Nominal pressure drop	35	[bar]
$\mathrm{p_{HP}}$	Low pressure side of pressure rail	200	[bar]
$\mathrm{p_{LP}}$	Low pressure side of pressure rail	1	[atm]

Table 3.2: Valve flow model parameters.

Based on information given by the staff at Linz University, the valve dynamics are constituted by a 2 ms dead time and a 1 ms ramp period from when a valve command u_i is given. As an example of this, Figure Figure 3.2 shows how the valve command signal translates to valve openings. In the figure, u_A, is set to 1 at t = 0.5 ms. This means that the high pressure line is to be connected to chamber A. Then, after a 2 ms dead time, $y_{A,1}$ begins ramping up and $y_{A,0}$ begins ramping down. The ramp period lasts for 1 ms after which the A chamber is connected solely to the high pressure line.

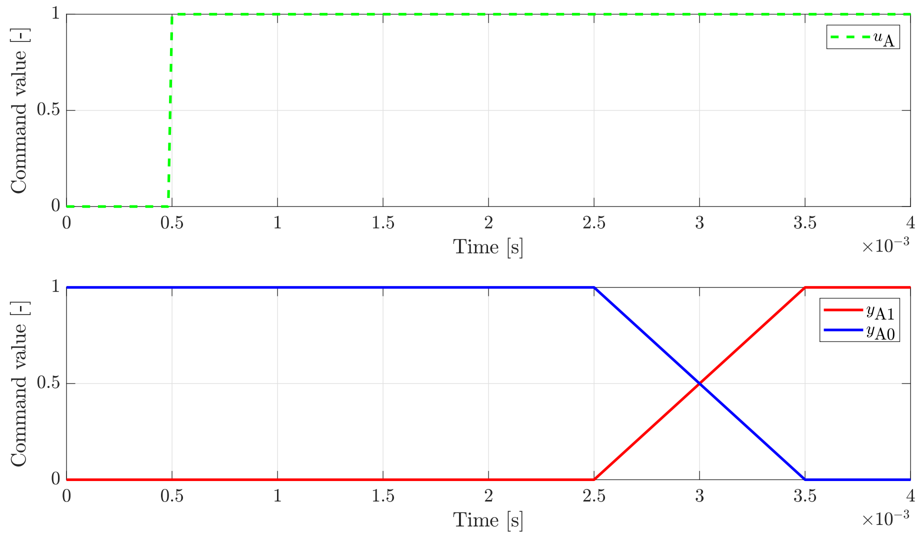


Figure 3.2: Valve dynamics. Top: A-chamber valve command value. Bottom: A-chamber valve openings.

The effective bulk modulus is modelled by [21]:

$$\boxed{\beta_i = \frac{(1-\alpha_{oil})(1+s_\beta \frac{p_i - p_0}{\beta_0})^{-\frac{1}{s_\beta}} + \alpha_{oil}(\frac{p_0}{p_i})^{\frac{1}{\kappa}}}{\frac{1}{\beta_0}(1-\alpha_{oil})(1+s_\beta \frac{p_i - p_0}{\beta_0})^{-\frac{s_\beta+1}{s_\beta}} + \frac{\alpha_{oil}}{\kappa p_0}(\frac{p_0}{p_i})^{\frac{\kappa+1}{\kappa}}}} \tag{3.3}$$

Here, the constant α_{oil} describes the volumetric content of air in the fluid at atmospheric pressure, κ denotes the polytropic constant of air, s_β is the constant related to the slope of the oil's bulk modulus relating to the pressure, β_0 is the oil compression modulus at the initial reference pressure p_0 and p_i is the pressure in chamber i. The constants for the bulk modulus model are given in Table 3.3.

Constant	Description	Value	Unit
α_{oil}	Air content in oil	0.2	[%]
κ	Polytropic constant of air	1.4	[-]
s_β	Pressure related slope of bulk modulus	11.4	[-]
β_0	Constant bulk modulus	14000	[bar]
p_0	Initial reference pressure	1	[bar]

Table 3.3: Bulk modulus model parameters.

The input energy from the supply is modelled by the integration of the flow from the high pressure rail multiplied by the high pressure level:

$$\boxed{E_{\text{in}} = \sum_{i=1}^{4} \int \mathrm{p}_{\text{HP}} \cdot Q_{\text{i1}}} \tag{3.4}$$

Using the concept of virtual work [22], the transmission ratios for the two actuators can be derived [14]:

$$\begin{aligned} M_{\text{hyd}}\mathrm{d}\psi &= F_{\text{AB}}\mathrm{d}z_1 + F_{\text{CD}}\mathrm{d}z_2 \\ \Longrightarrow \quad M_{\text{hyd}} &= F_{\text{AB}}\frac{\mathrm{d}z_1}{\mathrm{d}\psi} + F_{\text{CD}}\frac{\mathrm{d}z_2}{\mathrm{d}\psi} \end{aligned} \tag{3.5}$$

Recalling from Section 1.5 that in order to achieve uniform steps in the force resolution, the transmission ratio of the AB-actuator is twice that of the CD-actuator [14]:

$$\frac{\mathrm{d}z_1}{\mathrm{d}\psi} = 2\frac{\mathrm{d}z_2}{\mathrm{d}\psi} \tag{3.6}$$

Then, using Equations (3.5) and (3.6), the hydraulic knee torque, M_{hyd}, and the cylinder force, F_{hyd}, are derived:

$$\boxed{M_{\text{hyd}} = (F_{\text{AB}} + 0.5F_{\text{CD}})\frac{\mathrm{d}z_1}{\mathrm{d}\psi}} \tag{3.7}$$

$$\boxed{F_{\text{hyd}} = F_{\text{AB}} + 0.5F_{\text{CD}} = p_{\text{A}}\mathrm{A}_{\text{A}} - p_{\text{B}}\mathrm{A}_{\text{B}} + 0.5p_{\text{C}}\mathrm{A}_{\text{C}} - 0.5p_{\text{D}}\mathrm{A}_{\text{D}}} \tag{3.8}$$

There are 16 force levels for the system. These can be organised as seen in the following equation, where $\underline{\mathrm{F}}$ is the force vector, $\underline{\underline{\mathrm{p}}}_{\text{comb}}$ is a matrix with the 16 unique pressure combinations possible for the system and $\underline{\mathrm{A}}$ is a vector containing the cylinder chamber's equivalent piston areas when taking the transmission ratios into account:

$$\underline{\mathrm{F}} = \begin{bmatrix} \mathrm{F}_{\text{min}} \\ \vdots \\ \mathrm{F}_{\text{max}} \end{bmatrix} = \underline{\underline{\mathrm{p}}}_{\text{comb}}\,\underline{\mathrm{A}} = \begin{bmatrix} \mathrm{p}_{\text{LP}} & \mathrm{p}_{\text{HP}} & \mathrm{p}_{\text{LP}} & \mathrm{p}_{\text{HP}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{p}_{\text{HP}} & \mathrm{p}_{\text{LP}} & \mathrm{p}_{\text{HP}} & \mathrm{p}_{\text{LP}} \end{bmatrix} \begin{bmatrix} \mathrm{A}_{\text{A}} \\ -\mathrm{A}_{\text{B}} \\ 0.5\mathrm{A}_{\text{C}} \\ -0.5\mathrm{A}_{\text{D}} \end{bmatrix} \tag{3.9}$$

$\underline{\mathrm{F}}$ and $\underline{\underline{\mathrm{p}}}_{\text{comb}}$ are sorted for the pressure combinations resulting in the lowest force level beginning from the top and rising downwards. Then, $\underline{\underline{\mathrm{p}}}_{\text{comb}}$ is converted to the control input matrix, $\underline{\underline{u}}$, which is a matrix of binary values where $\mathrm{p}_{\text{HP}} = 1$ and $\mathrm{p}_{\text{LP}} = 0$:

$$\underline{\underline{\mathrm{p}}}_{\text{comb}} = \begin{bmatrix} \mathrm{p}_{\text{LP}} & \mathrm{p}_{\text{HP}} & \mathrm{p}_{\text{LP}} & \mathrm{p}_{\text{HP}} \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{p}_{\text{HP}} & \mathrm{p}_{\text{LP}} & \mathrm{p}_{\text{HP}} & \mathrm{p}_{\text{LP}} \end{bmatrix} \Longrightarrow \underline{\underline{u}} = \begin{bmatrix} 0 & 1 & 0 & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 1 & 0 \end{bmatrix} \tag{3.10}$$

This matrix contains 16 rows of control input vectors, who are accessed by the matrix row index, k_{idx}:

$$\underline{\underline{u}}(k_{\text{idx}},:) = \underline{u} = \begin{bmatrix} u_{\text{A}} & u_{\text{B}} & u_{\text{C}} & u_{\text{D}} \end{bmatrix} \tag{3.11}$$

Here, the vector $\underline{u}$ is the input vector which the controllers to be developed will use for commanding the valve block in order to control the system's position and torque.

3.2 Mechanical Model of Human Leg[1]

A diagram of the human leg is shown in Figure 3.3. This serves as the foundation for the mechanical model derivation.

To the left, the leg is shown with the following notations:

- Foot angle ϕ_1, shank angle ϕ_2, hip angle ϕ_3 and knee angle ψ
- Foot joint G_1, ankle joint G_2 and knee joint G_3
- Torque around the three joints: M_{G1}, M_{G2} and M_{G3}
- Center of mass for the foot CM1 and center of mass for the shank CM2
- Ground reaction force components F_{xG} and F_{yG}

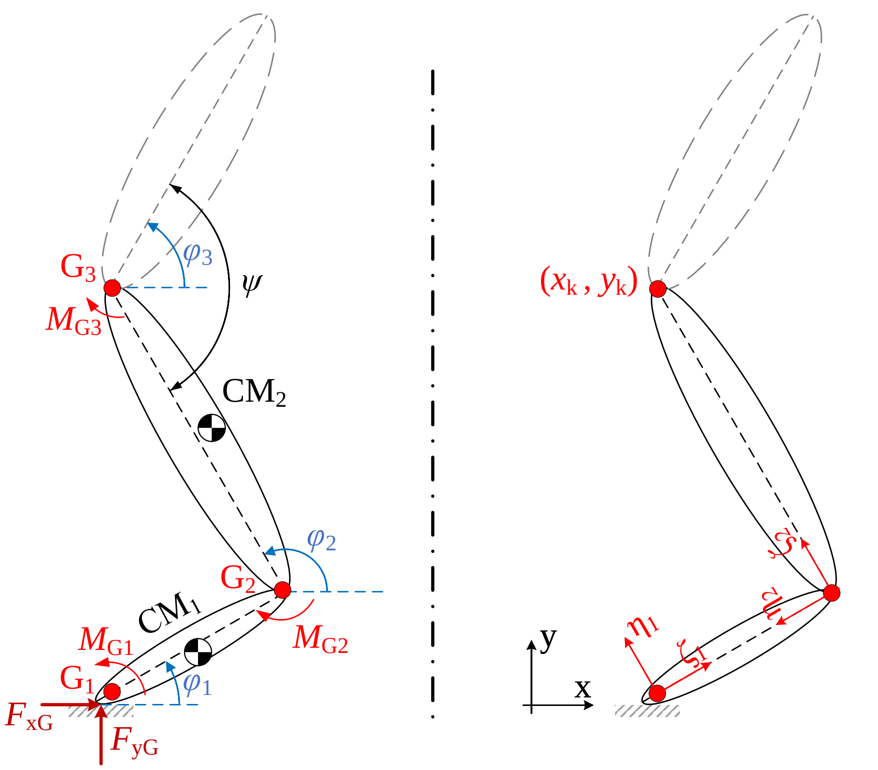


Figure 3.3: Diagram of human leg with notations used for the model derivation.

[1]The model derivation in this section is based on [14].

To the right on Figure 3.3, the leg is illustrated with the knee coordinates being (x_K,y_K) and the three reference frames:

- The global reference frame (x,y)
- The foot's limb fixed local reference frame (ζ_1,η_1)
- The shank's limb fixed local reference frame (ζ_2,η_2)

The knee angle is given by:

$$\psi = \pi + \phi_3 - \phi_2 \tag{3.12}$$

The mechanical model constants are given in Table 3.4:

Constant	Description	Value	Unit
m_1	Mass of foot	1.0058	[kg]
m_2	Mass of shank	4.0370	[kg]
l_1	Length of foot	0.1607	[m]
l_2	Length of shank	0.4344	[m]
J_1	Foot's moment of inertia	0.006	[kg$\cdot$ m^2]
J_2	Shank's moment of inertia	0.0582	[kg$\cdot$ m^2]
ζ_{CM1}	x-location of CM_1 in local reference frame	0.0607	[m]
ζ_{CM2}	x-location of CM_2 in local reference frame	0.253	[m]
η_{CM1}	y-location of CM_1 in local reference frame	-0.032	[m]
η_{CM2}	y-location of CM_2 in local reference frame	-0.0206	[m]

Table 3.4: Mechanical model constants.

Kinetic and Potential Energy

The kinetic energy of the system is described with the following equation:

$$E_{\text{kin}} = \frac{1}{2}\mathrm{m}_1(v^2_{\text{x,CM1}} + v^2_{\text{y,CM1}}) + \frac{1}{2}\mathrm{m}_2(v^2_{\text{x,CM2}} + v^2_{\text{y,CM2}}) + \frac{1}{2}\,\dot{\phi}_1^2\,\mathrm{J}_1 + \frac{1}{2}\,\dot{\phi}_2^2\,\mathrm{J}_2 \tag{3.13}$$

Here, m_1 and m_2 are the masses of the foot and shank respectively. J_1 and J_2 are the foot and shank's moment of inertia around the z-axis in each of their own limb fixed coordinate systems. $v_{x,CM1}$ and $v_{y,CM1}$ is the x- and y-component of the speed for the foot's center of mass (CM1). The same is the case for the shank's center of mass (CM2); denoted by $v_{x,CM2}$ and $v_{y,CM2}$. Lastly, $\dot{\phi}_1$ and $\dot{\phi}_2$ denote the angular velocity of the foot and shank.

The potential energy is described by:

$$E_{\text{pot}} = \mathrm{m}_1\,\mathrm{g}\,y_{\text{CM1}} + \mathrm{m}_2\,\mathrm{g}\,y_{\text{CM2}} \tag{3.14}$$

Where g is the gravitational acceleration and y_{CM1} and y_{CM2} are the y-coordinates of the two centres of mass in the global reference frame.

Kinematic Model

The knee joint is described in the global reference frame by:

$$\underline{x}_{\mathrm{K}} = \begin{bmatrix} x_{\mathrm{K}} \\ y_{\mathrm{K}} \end{bmatrix} \tag{3.15}$$

The three joints are described with respect to the the toe joint:

$$\underline{x}_{\mathrm{G1}} = \begin{bmatrix} x_{\mathrm{G1}} \\ y_{\mathrm{G1}} \end{bmatrix} \tag{3.16}$$

$$\underline{x}_{\mathrm{G2}} = \begin{bmatrix} x_{\mathrm{G1}} + l_1 \cos(\phi_1) \\ y_{\mathrm{G1}} + l_1 \sin(\phi_1) \end{bmatrix} \tag{3.17}$$

$$\underline{x}_{\mathrm{G3}} = \begin{bmatrix} x_{\mathrm{G1}} + l_1 \cos(\phi_1) + l_2 \cos(\phi_2) \\ y_{\mathrm{G1}} + l_1 \sin(\phi_1) + l_2 \sin(\phi_2) \end{bmatrix} \tag{3.18}$$

The knee coordinates of Equation (3.15) are then related to the toe coordinates by:

$$\underline{x}_{\mathrm{K}} = \underline{x}_{\mathrm{G3}} \implies \begin{bmatrix} x_{\mathrm{G1}} \\ y_{\mathrm{G1}} \end{bmatrix} = \begin{bmatrix} -l_1 \cos(\phi_1) - l_2 \cos(\phi_2) + x_{\mathrm{K}} \\ -l_1 \sin(\phi_1) - l_2 \sin(\phi_2) + y_{\mathrm{K}} \end{bmatrix} \tag{3.19}$$

Using this, the joint positions of Equations (3.16) to (3.18) are reformulated to:

$$\underline{x}_{\mathrm{G1}} = \begin{bmatrix} -l_1 \cos(\phi_1) - l_2 \cos(\phi_2) + x_{\mathrm{K}} \\ -l_1 \sin(\phi_1) - l_2 \sin(\phi_2) + y_{\mathrm{K}} \end{bmatrix} \tag{3.20}$$

$$\underline{x}_{\mathrm{G2}} = \begin{bmatrix} -l_2 \cos(\phi_2) + x_{\mathrm{K}} \\ -l_2 \sin(\phi_2) + y_{\mathrm{K}} \end{bmatrix} \tag{3.21}$$

$$\underline{x}_{\mathrm{G3}} = \begin{bmatrix} x_{\mathrm{K}} \\ y_{\mathrm{K}} \end{bmatrix} \tag{3.22}$$

The two centres of mass are in the foot's and shank's limb fixed coordinate systems defined to be:

$$\underline{r}_{\mathrm{CM1}} = \begin{bmatrix} \zeta_{\mathrm{CM1}} \\ \eta_{\mathrm{CM1}} \end{bmatrix} \tag{3.23}$$

$$\underline{r}_{\mathrm{CM2}} = \begin{bmatrix} \zeta_{\mathrm{CM2}} \\ \eta_{\mathrm{CM2}} \end{bmatrix} \tag{3.24}$$

The rotation matrix is:

$$\underline{\underline{R}} = \begin{bmatrix} \cos(\phi) & -\sin(\phi) \\ \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{3.25}$$

Using the joint positions of Equations (3.20) to (3.22) along with the center of mass positions of Equations (3.23) and (3.24) as well as the rotation matrix of Equation (3.25), the centres of mass are found in the global reference frame:

$$\underline{x}_{\mathrm{CM1}} = \underline{x}_{\mathrm{G1}} + \underline{\underline{R}}\,\underline{r}_{\mathrm{CM1}} = \begin{bmatrix} x_{\mathrm{CM1}} \\ y_{\mathrm{CM1}} \end{bmatrix}$$

$$\begin{bmatrix} x_{\mathrm{CM1}} \\ y_{\mathrm{CM1}} \end{bmatrix} = \begin{bmatrix} -l_1 \cos(\phi_1) - l_2 \cos(\phi_2) + x_{\mathrm{K}} + \cos(\phi_1)\,\zeta_{\mathrm{CM1}} - \sin(\phi_1)\,\eta_{\mathrm{CM1}} \\ -l_1 \sin(\phi_1) - l_2 \sin(\phi_2) + y_{\mathrm{K}} + \sin(\phi_1)\,\zeta_{\mathrm{CM1}} + \cos(\phi_1)\,\eta_{\mathrm{CM1}} \end{bmatrix} \tag{3.26}$$

$$\underline{x}_{\mathrm{CM2}} = \underline{x}_{\mathrm{G2}} + \underline{\underline{R}}\,\underline{r}_{\mathrm{CM2}} = \begin{bmatrix} x_{\mathrm{CM2}} \\ y_{\mathrm{CM2}} \end{bmatrix}$$

$$\begin{bmatrix} x_{\mathrm{CM2}} \\ y_{\mathrm{CM2}} \end{bmatrix} = \begin{bmatrix} -l_2 \cos(\phi_2) + x_{\mathrm{K}} + \cos(\phi_2)\,\zeta_{\mathrm{CM2}} - \sin(\phi_2)\,\eta_{\mathrm{CM2}} \\ -l_2 \sin(\phi_2) + y_{\mathrm{K}} + \sin(\phi_2)\,\zeta_{\mathrm{CM2}} + \cos(\phi_2)\,\eta_{\mathrm{CM2}} \end{bmatrix} \tag{3.27}$$

Using the y-components of Equations (3.26) and (3.27), the potential energy of the system can be found according to Equation (3.14).

Taking the time derivative of the position vector for the first center of mass, given by Equation (3.26), results in the velocity:

$$\dot{\underline{x}}_{\mathrm{CM1}} = \underline{\mathrm{v}}_{\mathrm{CM1}} = \begin{bmatrix} v_{\mathrm{x,CM1}} \\ v_{\mathrm{y,CM1}} \end{bmatrix} \Longrightarrow$$

$$v_{\mathrm{x,CM1}} = l_1 \dot{\phi}_1 \sin(\phi_1(t)) + l_2 \dot{\phi}_2 \sin(\phi_2) + \dot{x}_{\mathrm{K}} - \dot{\phi}_1 \sin(\phi_1)\,\zeta_{\mathrm{CM1}} - \dot{\phi}_1 \cos(\phi_1)\,\eta_{\mathrm{CM1}} \tag{3.28}$$

$$v_{\mathrm{y,CM1}} = -l_1 \dot{\phi}_1 \cos(\phi_1) - l_2 \dot{\phi}_2 \cos(\phi_2) + \dot{y}_{\mathrm{K}} + \dot{\phi}_1 \cos(\phi_1)\,\zeta_{\mathrm{CM1}} - \dot{\phi}_1 \sin(\phi_1)\,\eta_{\mathrm{CM1}} \tag{3.29}$$

And likewise, the velocity of the second center of mass is obtained by taking the time derivative of Equation (3.27):

$$\dot{\underline{x}}_{\mathrm{CM2}} = \underline{\mathrm{v}}_{\mathrm{CM2}} = \begin{bmatrix} v_{\mathrm{x,CM2}} \\ v_{\mathrm{y,CM2}} \end{bmatrix} \Longrightarrow$$

$$v_{\mathrm{x,CM2}} = l_2 \dot{\phi}_2 \sin(\phi_2(t)) + \dot{x}_{\mathrm{K}} - \dot{\phi}_2 \sin(\phi_2)\,\zeta_{\mathrm{CM2}} - \dot{\phi}_2 \cos(\phi_2)\,\eta_{\mathrm{CM2}} \tag{3.30}$$

$$v_{\mathrm{y,CM2}} = -l_2 \dot{\phi}_2 \cos(\phi_2) + \dot{y}_{\mathrm{K}} + \dot{\phi}_2 \cos(\phi_2)\,\zeta_{\mathrm{CM2}} - \dot{\phi}_2 \sin(\phi_2)\,\eta_{\mathrm{CM2}} \tag{3.31}$$

Using Equations (3.28) to (3.31), the kinetic energy equation can be calculated using Equation (3.13).

3.2.1 Virtual Work

Now, the concept of virtual work is used to derive the interaction of the forces and couples on the leg.

Virtual Work on Ankle and Knee Joints

Figure 3.4 shows an illustration of the rotational virtual displacements for the knee and ankle joints. To the left, the shank angle, ϕ_2 and the hip angle, ϕ_3, are changed by the virtual angular displacements $\delta\phi_2$ and $\delta\phi_3$. This is seen by the green lines, denoting how the limbs change to a new position. Here the difference, $\delta\phi_3 - \delta\phi_2$ denotes the resulting angular displacement of the knee.

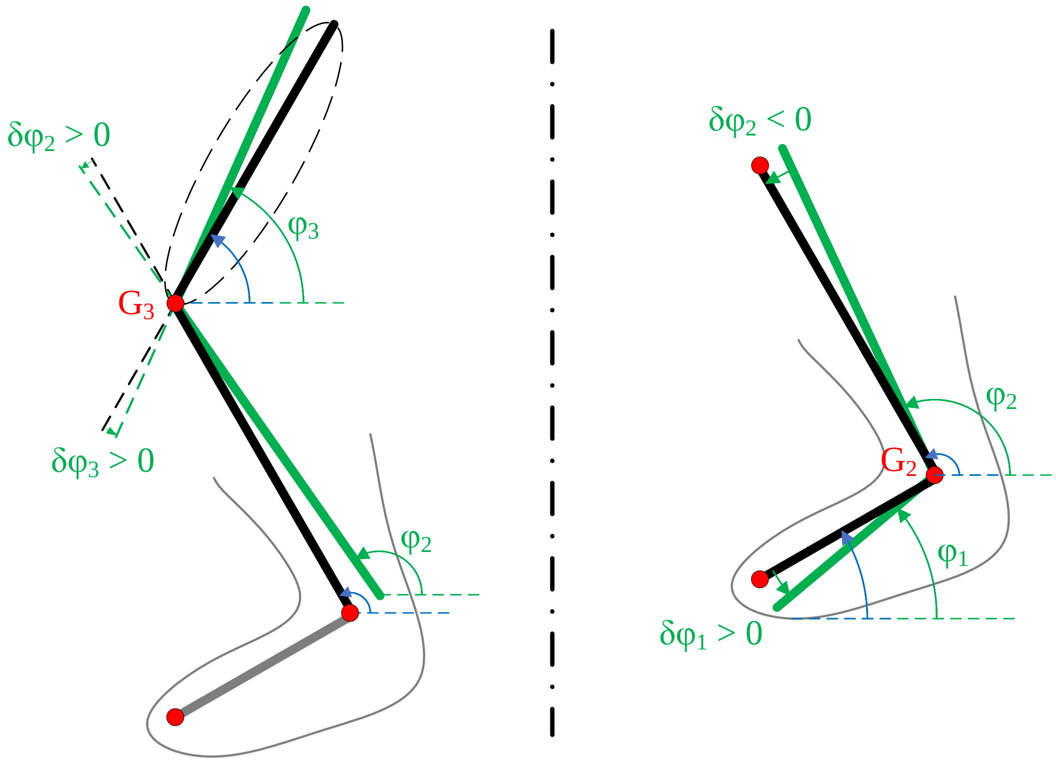


Figure 3.4: Illustration of angular virtual displacements for the knee (left) and the foot (right).

To the right of Figure 3.4, it is shown how the virtual displacements of the foot angle, $\delta\phi_1$ and shank angle, $\delta\phi_2$, influences the angular displacement of the G_2 angle: $\delta\phi_2 - \delta\phi_1$. This means that the virtual work due to the couples on joint 2 and joint 3, are:

$$\delta W_{\text{G2+G3}} = M_{\text{G2}}(\delta\phi_2 - \delta\phi_1) + M_{\text{G3}}(\delta\phi_3 - \delta\phi_2) \tag{3.32}$$

The data for the thigh movement is predetermined based on the HuMoD database, which means that the knee torque, M_{G3}, will only affect the shank angle, $\delta\phi_2$. Therefore, Equation (3.32) simplifies to:

$$\delta W_{\text{G2+G3}} = M_{\text{G2}}(\delta\phi_2 - \delta\phi_1) - M_{\text{G3}}\delta\phi_2 \tag{3.33}$$

Virtual Work on Toe Joint

Now, the virtual work on the toe joint is found. Here, there will be both linear and angular virtual displacements due to the ground reaction force (GRF). Firstly, the linear virtual displacements of the toe joint are found using the position vector of Equation (3.20):

$$\delta \underline{x}_{G1} = \begin{bmatrix} \delta x_{G1} \\ \delta y_{G1} \end{bmatrix} = \begin{bmatrix} l_1 \sin(\phi_1)\,\delta\phi_1 + l_2 \sin(\phi_2)\,\delta\phi_2 \\ -l_1 \cos(\phi_1)\,\delta\phi_1 - l_2 \cos(\phi_2)\,\delta\phi_2 \end{bmatrix} \tag{3.34}$$

The virtual work due to the virtual linear displacement of the toe joint is then given by the scalar product of the GRF vector, $\underline{F}_{GRF}$, and the virtual linear displacement vector of Equation (3.34):

$$\delta W_{G1,\text{linear}} = \underline{F}_{GRF}\,\delta \underline{x}_{G1} \qquad ; \qquad \underline{F}_{GRF} = \begin{bmatrix} F_{xG} & F_{yG} \end{bmatrix} \tag{3.35}$$

The angular virtual displacement of the toe joint is simply $\delta\phi_1$, since this is only dependent on the foot link orientation. Figure 3.5 shows an illustration of the foot in two positions. To the left, the ankle has contact to the ground in the point (x_C , y_C), and to the right the ground contact point is at the toe.

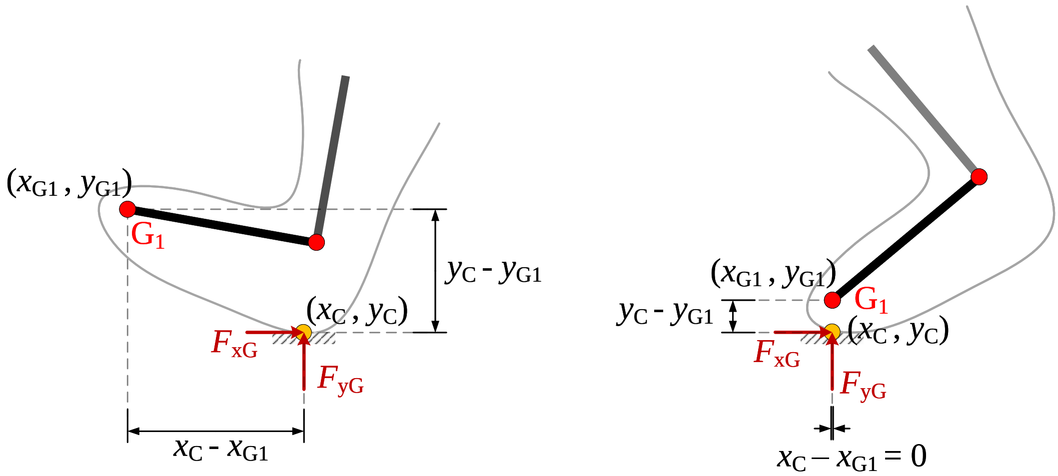


Figure 3.5: Illustration of how the GRF generates a torque around the toe joint G1.

Two lever arms are denoted, r_x and r_y, on which the GRF components, F_{xG} and F_{yG}, work and thereby create a torque around the toe joint. The lever arms are defined to be the horizontal and vertical distances between the contact point for the GRF (x_C , y_C) and the toe joint position (x_{G1},y_{G1}). This is also shown in Figure 3.5. The lever arms, r_x and r_y, are then found using Equations (3.20) and (3.21):

$$\underline{r}_{GRF} = \begin{bmatrix} r_y \\ r_x \end{bmatrix} = \begin{bmatrix} y_C - y_{G1} \\ x_C - x_{G1} \end{bmatrix} = \begin{bmatrix} y_C + l_1 \sin(\phi_1) + l_2 \sin(\phi_2) - y_K \\ x_C + l_1 \cos(\phi_1) + l_2 \cos(\phi_2) - x_K \end{bmatrix} \tag{3.36}$$

Here, F_{xG} works on r_{y}, and F_{yG} works on r_{x}, resulting in the following torque contributions:

$$M_{\text{G1}} = \underline{F}_{\text{GRF}} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \underline{r}_{\text{GRF}} = -F_{\text{xG}} r_{\text{y}} + F_{\text{yG}} r_{\text{x}} \tag{3.37}$$

In this equation, the diagonal matrix switches the sign of the lever arm, r_{y}, in order to keep the torque contributions consistent with the definition of the positive direction being counter-clockwise. As an example, looking at the right part of Figure 3.5, it can be seen that there is no difference between x_{C} and x_{G1}, meaning that F_{yG} will not give any torque contribution around the toe joint in this specific case.

Then, the virtual work resulting from the torque components on the toe joint, G1, becomes:

$$\delta W_{\text{G1,angular}} = \delta\phi_1 M_{\text{G1}} \tag{3.38}$$

The work on the toe joint is then given as:

$$\delta W_{\text{G1}} = \delta W_{\text{G1,linear}} + \delta W_{\text{G1,angular}} \tag{3.39}$$

Total Virtual Work

The total virtual work is found by summing the virtual work on all three joints as derived in Equations (3.33) and (3.39):

$$\begin{aligned} \delta W = \delta W_{\text{G1}} + \delta W_{\text{G2+G3}} &= \left(\left(-l_2 \sin(\phi_2) + y_{\text{K}} - y_{\text{C}} \right) \delta\phi_1 + l_2 \sin(\phi_2)\, \delta\phi_2 \right) F_{\text{xG}} \\ &+ \left(\left(l_2 \cos(\phi_2) - x_{\text{K}} + x_{\text{C}} \right) \delta\phi_1 - l_2 \cos(\phi_2)\, \delta\phi_2 \right) F_{\text{yG}} + M_{\text{G2}} \left(\delta\phi_2 - \delta\phi_1 \right) - M_{\text{G3}} \delta\phi_2 \end{aligned} \tag{3.40}$$

3.2.2 Lagrangian Mechanics

The Lagrangian is the difference between the kinetic of Equation (3.13) and the potential energy of Equation (3.14):

$$\mathcal{L} = E_{\text{kin}} - E_{\text{pot}} \tag{3.41}$$

The Lagrangian equation is formulated using Equations (3.40) and (3.41):

$$\boxed{\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathcal{L}}{\partial \dot{\underline{q}}} - \frac{\partial \mathcal{L}}{\partial \underline{q}} = \frac{\partial \delta W}{\partial \delta \underline{q}}} \tag{3.42}$$

$$\frac{\mathrm{d}}{\mathrm{d}t} \frac{\partial \mathcal{L}}{\partial \dot{\underline{q}}} = \begin{bmatrix} \kappa_1 \\ \kappa_2 \end{bmatrix} \quad ; \quad \frac{\partial \mathcal{L}}{\partial \underline{q}} = \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix} \quad ; \quad \frac{\partial \delta W}{\partial \delta \underline{q}} = \begin{bmatrix} \omega_1 \\ \omega_2 \end{bmatrix}$$

Here, the equation for virtual work constitutes a constraint equation describing the influence of the external effects done at the joints. The vectors $\underline{q}$, $\dot{\underline{q}}$ and $\delta\underline{q}$ denote the variables for position, velocity and virtual displacement corresponding to the system's degrees of freedom. Since data concerning the dynamics of the thigh is taken from the HuMoD database, the system has two degrees of freedom resulting in:

$$\underline{q} = \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} \quad ; \quad \dot{\underline{q}} = \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} \quad ; \quad \delta\underline{q} = \begin{bmatrix} \delta\phi_1 \\ \delta\phi_2 \end{bmatrix}$$

The three terms of Equation (3.42) are given in the following equations.

$$\begin{aligned}
\kappa_1 = \mathrm{m}_1 \Big[& ((l_1 - \zeta_{\mathrm{CM1}}) \cos(\phi_2) - \eta_{\mathrm{CM1}} \sin(\phi_2)) \cos(\phi_1) \\
& + (\eta_{\mathrm{CM1}} \cos(\phi_2) + \sin(\phi_2)(l_1 - \zeta_{\mathrm{CM1}})) \sin(\phi_1) \Big] l_2 \ddot{\phi}_2 \\
& + \left(\left({\eta_{\mathrm{CM1}}}^2 + l_1^2 - 2 l_1 \zeta_{\mathrm{CM1}} + {\zeta_{\mathrm{CM1}}}^2 \right) \mathrm{m}_1 + \mathrm{J}_1 \right) \ddot{\phi}_1 \\
& - \mathrm{m}_1 \Big[(\cos(\phi_1) \eta_{\mathrm{CM1}} - \sin(\phi_1)(l_1 - \zeta_{\mathrm{CM1}})) \ddot{x}_{\mathrm{K}} \\
& + ((l_1 - \zeta_{\mathrm{CM1}}) \cos(\phi_1) + \sin(\phi_1) \eta_{\mathrm{CM1}}) \ddot{y}_{\mathrm{K}} \\
& + \Big((\eta_{\mathrm{CM1}} \cos(\phi_2) + \sin(\phi_2)(l_1 - \zeta_{\mathrm{CM1}})) \cos(\phi_1) \\
& - \sin(\phi_1)((l_1 - \zeta_{\mathrm{CM1}}) \cos(\phi_2) - \eta_{\mathrm{CM1}} \sin(\phi_2)) \Big) \dot{\phi}_2^2 l_2 \Big]
\end{aligned} \tag{3.43}$$

$$\begin{aligned}
\kappa_2 = \mathrm{m}_1 \Big[& ((l_1 - \zeta_{\mathrm{CM1}}) \cos(\phi_1) + \sin(\phi_1) \eta_{\mathrm{CM1}}) \cos(\phi_2) \\
& - \sin(\phi_2)(\cos(\phi_1) \eta_{\mathrm{CM1}} - \sin(\phi_1)(l_1 - \zeta_{\mathrm{CM1}})) \Big] l_2 \ddot{\phi}_1 \\
& + \left((\mathrm{m}_1 + \mathrm{m}_2) l_2^2 - 2 \zeta_{\mathrm{CM2}} l_2 \mathrm{m}_2 + \left({\eta_{\mathrm{CM2}}}^2 + {\zeta_{\mathrm{CM2}}}^2 \right) \mathrm{m}_2 + \mathrm{J}_2 \right) \ddot{\phi}_2 \\
& + (-\eta_{\mathrm{CM2}} \cos(\phi_2) \mathrm{m}_2 + ((\mathrm{m}_1 + \mathrm{m}_2) l_2 - \zeta_{\mathrm{CM2}} \mathrm{m}_2) \sin(\phi_2)) \ddot{x}_{\mathrm{K}} \\
& + (((-\mathrm{m}_1 - \mathrm{m}_2) l_2 + \zeta_{\mathrm{CM2}} \mathrm{m}_2) \cos(\phi_2) - \eta_{\mathrm{CM2}} \sin(\phi_2) \mathrm{m}_2) \ddot{y}_{\mathrm{K}} \\
& + \mathrm{m}_1 \Big[(\cos(\phi_1) \eta_{\mathrm{CM1}} - \sin(\phi_1)(l_1 - \zeta_{\mathrm{CM1}})) \cos(\phi_2) \\
& + ((l_1 - \zeta_{\mathrm{CM1}}) \cos(\phi_1) + \sin(\phi_1) \eta_{\mathrm{CM1}}) \sin(\phi_2) \Big] \dot{\phi}_1^2 l_2
\end{aligned} \tag{3.44}$$

$$\epsilon_1 = \mathrm{m}_1 \mathrm{g} (-l_1 \cos(\phi_1) + \cos(\phi_1) \zeta_{\mathrm{CM1}} - \sin(\phi_1) \eta_{\mathrm{CM1}}) \tag{3.45}$$

$$\epsilon_2 = -\mathrm{m}_1 \mathrm{g} l_2 \cos(\phi_2) + \mathrm{m}_2 \mathrm{g} (-l_2 \cos(\phi_2) + \cos(\phi_2) \zeta_{\mathrm{CM2}} - \sin(\phi_2) \eta_{\mathrm{CM2}}) \tag{3.46}$$

$$\omega_1 = \cos(\phi_2) F_{\mathrm{yG}} l_2 - \sin(\phi_2) F_{\mathrm{xG}} l_2 - x_{\mathrm{K}} F_{\mathrm{yG}} + y_{\mathrm{K}} F_{\mathrm{xG}} - F_{\mathrm{xG}} y_{\mathrm{C}} + F_{\mathrm{yG}} x_{\mathrm{C}} - M_{\mathrm{G2}} \tag{3.47}$$

$$\omega_2 = -\cos(\phi_2) F_{\mathrm{yG}} l_2 + \sin(\phi_2) F_{\mathrm{xG}} l_2 + M_{\mathrm{G2}} - M_{\mathrm{G3}} \tag{3.48}$$

The system of equations given by the Lagrange equation is now organized into the following form:

$$\boxed{\underline{\underline{\mathcal{J}}}\,\underline{\ddot{\phi}} = \underline{\underline{B}}_{\mathrm{M}}\,\underline{M}_{\mathrm{G}} + \underline{\underline{B}}_{\mathrm{F}}\,\underline{\mathrm{F}}_{\mathrm{GRF}} + \underline{V}_{\mathrm{f0}}} \tag{3.49}$$

An elaboration on each of the terms in Equation (3.49) follows here:

Firstly, $\underline{\underline{\mathcal{J}}}$ is the inertia matrix which relates to the acceleration vector $\underline{\ddot{\phi}}$. These are given by:

$$\underline{\underline{\mathcal{J}}} = \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \qquad ; \qquad \underline{\ddot{\phi}} = \begin{bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{bmatrix}$$

The terms in the inertia matrix are found extracting the coefficients for the acceleration variables in Equations (3.43) and (3.44):

$$\mathrm{J}_{11} = \left(\eta_{\mathrm{CM1}}{}^2 + l_1^2 - 2l_1\zeta_{\mathrm{CM1}} + \zeta_{\mathrm{CM1}}{}^2\right)\mathrm{m}_1 + \mathrm{J}_1 \tag{3.50}$$

$$J_{12} = l_2\mathrm{m}_1\left(l_1\cos(-\phi_2 + \phi_1) - \zeta_{\mathrm{CM1}}\cos(-\phi_2 + \phi_1) + \eta_{\mathrm{CM1}}\sin(-\phi_2 + \phi_1)\right) \tag{3.51}$$

$$J_{21} = J_{12} \tag{3.52}$$

$$\mathrm{J}_{22} = \left(\eta_{\mathrm{CM2}}{}^2 + l_2^2 - 2l_2\zeta_{\mathrm{CM2}} + \zeta_{\mathrm{CM2}}{}^2\right)\mathrm{m}_2 + l_2^2\mathrm{m}_1 + \mathrm{J}_2 \tag{3.53}$$

$\underline{\underline{B}}_{\mathrm{M}}$ is derived from Equations (3.47) and (3.48) and denotes the influence of the joint torques vector, $\underline{M}_{\mathrm{G}}$, on the foot and shank angular accelerations:

$$\underline{\underline{B}}_{\mathrm{M}} = \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix} \qquad ; \qquad \underline{M}_{\mathrm{G}} = \begin{bmatrix} M_{\mathrm{G2}} \\ M_{\mathrm{G3}} \end{bmatrix} \tag{3.54}$$

$\underline{\underline{B}}_{\mathrm{F}}$ is derived from Equations (3.47) and (3.48) and denotes the terms relating to the ground reaction forces, F_{xG} and F_{yG}:

$$\underline{\underline{B}}_{\mathrm{F}} = \begin{bmatrix} B_{\mathrm{F},11} & B_{\mathrm{F},12} \\ B_{\mathrm{F},21} & B_{\mathrm{F},22} \end{bmatrix} = \begin{bmatrix} -l_2\sin(\phi_2) + y_{\mathrm{K}} - y_{\mathrm{C}} & l_2\cos(\phi_2) - x_{\mathrm{K}} + x_{\mathrm{C}} \\ l_2\sin(\phi_2) & -l_2\cos(\phi_2) \end{bmatrix} \tag{3.55}$$

Lastly, $\underline{V}_{\mathrm{f0}}$ is all the remaining terms of Equations (3.43) to (3.48) who have no relation to the accelerations, torques or ground reaction forces. This vector is given as:

$$\underline{V}_{\mathrm{f0}} = \begin{bmatrix} V_{\mathrm{f0,1}} \\ V_{\mathrm{f0,2}} \end{bmatrix} \implies$$

$$V_{f0,1} = \Big[-\dot{\phi}_2^2 l_2 \,(l_1 - \zeta_{CM1}) \sin(-\phi_2 + \phi_1) + \dot{\phi}_2^2 \cos(-\phi_2 + \phi_1)\, \eta_{CM1} l_2 \\ + (\cos(\phi_1)\, \eta_{CM1} - \sin(\phi_1)\,(l_1 - \zeta_{CM1}))\, \ddot{x}_K + (\mathrm{g} + \ddot{y}_K)\,((l_1 - \zeta_{CM1}) \cos(\phi_1) + \sin(\phi_1)\, \eta_{CM1}) \Big] \mathrm{m}_1 \tag{3.56}$$

$$V_{f0,2} = \dot{\phi}_1^2 l_2 \mathrm{m}_1 \,(l_1 - \zeta_{CM1}) \sin(-\phi_2 + \phi_1) - \cos(-\phi_2 + \phi_1)\, \dot{\phi}_1^2 \eta_{CM1} l_2 \mathrm{m}_1 \\ + \Big(\eta_{CM2} \cos(\phi_2)\, \mathrm{m}_2 - ((\mathrm{m}_1 + \mathrm{m}_2)\, l_2 - \zeta_{CM2} \mathrm{m}_2) \sin(\phi_2) \Big) \ddot{x}_K \\ + (\mathrm{g} + \ddot{y}_K)\,(((\mathrm{m}_1 + \mathrm{m}_2)\, l_2 - \zeta_{CM2} \mathrm{m}_2) \cos(\phi_2) + \eta_{CM2} \sin(\phi_2)\, \mathrm{m}_2) \tag{3.57}$$

3.2.3 Simplifying the Model

Taking a closer look at the torque term of Equation (3.54), it can be seen that it is possible to completely remove the influence of the ankle torque on the shank angle dynamics, by multiplying the system of equations with the vector $\underline{k} = \begin{bmatrix} 1 & 1 \end{bmatrix}$:

$$\underline{k}\, \underline{\underline{B}}_M \, \underline{M}_G = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} M_{G2} \\ M_{G3} \end{bmatrix} = \begin{bmatrix} 0 & -1 \end{bmatrix} \begin{bmatrix} M_{G2} \\ M_{G3} \end{bmatrix} = -M_{G3} \tag{3.58}$$

This vector is multiplied onto the whole system of equations given by Equation (3.49) in order to obtain the simplified Equation (3.59) for the knee torque:

$$\underline{k}\, \underline{\underline{\mathcal{J}}}\, \underline{\ddot{\phi}} = \underline{k}\, \underline{\underline{B}}_M \, \underline{M}_G + \underline{k}\, \underline{\underline{B}}_F \, \underline{F}_{GRF} + \underline{k}\, \underline{V}_{f0} \implies$$

$$\boxed{M_{G3} = -\underline{k}\, \underline{\underline{\mathcal{J}}}\, \underline{\ddot{\phi}} + \underline{k}\, \underline{\underline{B}}_F \, \underline{F}_{GRF} + \underline{k}\, \underline{V}_{f0}} \tag{3.59}$$

Here, the terms are:

$$\underline{k}\, \underline{\underline{\mathcal{J}}}\, \underline{\ddot{\phi}} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} J_{11} & J_{12} \\ J_{21} & J_{22} \end{bmatrix} \begin{bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{bmatrix} = (J_{11} + J_{21})\ddot{\phi}_1 + (J_{12} + J_{22})\ddot{\phi}_2 \tag{3.60}$$

$$\underline{k}\, \underline{\underline{B}}_F \, \underline{F}_{GRF} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} B_{F,11} & B_{F,12} \\ B_{F,21} & B_{F,22} \end{bmatrix} \begin{bmatrix} F_{xG} \\ F_{yG} \end{bmatrix} = (B_{F,11} + B_{F,21})F_{xG} + (B_{F,12} + B_{F,22})F_{yG} \tag{3.61}$$

$$\underline{k}\, \underline{V}_{f0} = \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} V_{f0,1} \\ V_{f0,2} \end{bmatrix} = V_{f0,1} + V_{f0,2} \tag{3.62}$$

Equation (3.59) is used to calculate the knee torque during the gait cycle. This is done using the HuMoD data for ϕ_1, $\dot{\phi}_1$, $\ddot{\phi}_1$, x_K, $\ddot{x}_K$, y_K, $\ddot{y}_K$, F_{xG} and F_{yG}. The torque is plotted during a gait cycle in Figure 3.6, and this is what serves as the torque reference trajectory.

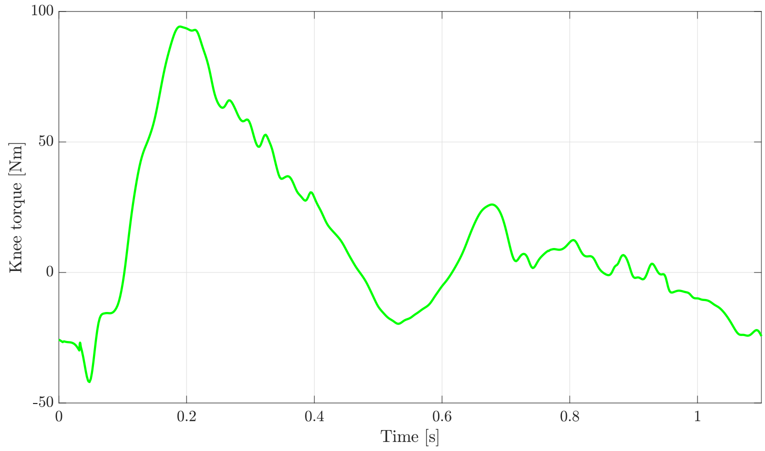


Figure 3.6: Knee torque, M_{G3}, plotted for one gait cycle using Equation (3.59).

The knee torque plotted in Figure 3.6 is the same as was shown in Figure 1.5 from Section 1.5, which is corresponding with the results obtained in [14], [18]. This verifies the implementation of the simulation model.

In order to obtain the dynamic model for the exoskeleton, Equation (3.59) is rewritten whilst equating $M_{G3} = M_{hyd}$ from Equation (3.7):

$$\ddot{\phi}_2 = \frac{-M_{hyd} + M_{fric} - (J_{11} + J_{21})\ddot{\phi}_1 + (B_{F,11} + B_{F,21})F_{xG} + (B_{F,12} + B_{F,22})F_{yG} + V_{f0,1} + V_{f0,2}}{(J_{12} + J_{22})} \tag{3.63}$$

In Equation (3.63), the friction is added by regarding the friction as consisting solely of a viscous term according to:

$$M_{fric} = \beta_{eq}\dot{\psi} = \beta_{eq}(\dot{\phi}_3 - \dot{\phi}_2) \tag{3.64}$$

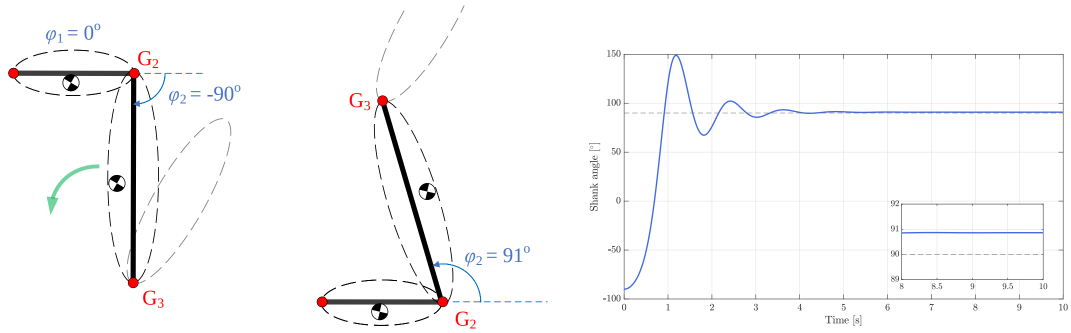
The viscous friction coefficient β_{eq} is an equivalent term that collects the friction in the hydraulic actuators, the exoskeleton and the human knee joint. Since no prototype is available at the time of writing, a value has to be assumed for β_{eq}. Analysing the angular velocity of the knee joint during the HuMoD gait cycle, it is noted that the maximum absolute speed is $\approx$ 7 rad/s. Setting $\beta_{eq} = 1$ Nms means that the maximum friction torque the system will be $\approx$ 7 % of the peak knee torque during the stance phase of the gait cycle. This is assumed to be a valid order of magnitude for the constant, but as mentioned before, it must be stressed that an experimental parameter determination is necessary in order to achieve a validated friction model that is true to the physical system.

A simplified form of the system dynamics of Equation (3.63) are given in Equation (3.65):

$$\boxed{\ddot{\phi}_2 = \frac{-M_{hyd} + M_{ext}}{(J_{12} + J_{22})}} \tag{3.65}$$

In this notation, the term M_{ext} describes the external torque contributions on the knee torque who come from the friction, the centripetal force for the foot and shank, the ground reaction force and the gravity. Equation (3.65) gives the shank angle dynamics as a function of the hydraulic actuator torque exerted on the knee. Thereby, a model for the knee movement of the exoskeleton has been derived, since the shank angle directly relates to the knee angle of Equation (3.12) because the thigh angle, ϕ_3, is predetermined from the HuMoD data.

A sanity check of the dynamic equation for shank movement is conducted as shown in Figure 3.7. Here, Equation (3.65) is initialized where the shank is placed in the upright position of $\phi_2 = -90°$. Throughout the test, the foot angle is kept constant at $\phi_1 = 0°$. The simulation is run where there is no knee movement, ground reaction force and actuator torque meaning that $\ddot{x}_{\text{K}} = \ddot{y}_{\text{K}} = M_{\text{hyd}} = F_{\text{xG}} = F_{\text{yG}} = 0$. Furthermore, the movement of the foot is also set to zero: $\dot{\phi}_1 = \ddot{\phi}_1 = 0$. As seen in Figure 3.7b, the angle of the shank begins rising until it settles at a value of $\phi_2 = -91°$. This is also illustrated in Figure 3.7a, where the initial and final position of the leg during the sanity check is shown. This behaviour is in accordance with expectations, and thus indicates that the model is correct. Furthermore, the response also gives an impression of how the chosen friction coefficient, β_{eq}, affects the system; which is by inducing a response that is neither very underdamped nor overdamped.



(a) Initial and final position of the leg.

(b) Shank angle, ϕ_2, during the test.

Figure 3.7: Sanity check of Equation (3.65).

3.2.4 Ground Reaction Force Model

When simulating the system, it is possible to implement the measured ground reaction force from the HuMoD database. But this is not desired when the control algorithm is to be implemented on the system, since deviations from the measured gait cycle can occur, and if this happens, then the GRF will no longer be representative for the simulated case. Therefore, a model is derived for the GRF, which enables the GRF to be influenced by the deviations between the shank angle, ϕ_2 and the HuMoD data, thus allowing for controller evaluation through simulations. The GRF components due to the elastic compression as well as damping at the toe and ankle joints can be modelled as [14]:

$$F_{\text{y,Gi}} = \text{k}_{\text{y,Gi}}(\text{y}_{\text{Gi,thres}} - y_{\text{Gi}}) - \text{c}_{\text{y,Gi}} \cdot \min(0, v_{\text{y,Gi}}) \tag{3.66}$$

$$F_{\text{x,Gi}} = \text{k}_{\text{x,Gi}}(x_{\text{C,Gi}} - x_{\text{Gi}}) - \text{c}_{\text{x,Gi}}\, v_{\text{x,Gi}} \tag{3.67}$$

Here, the x- and y-direction speed of the respective joints are given by $v_{\text{x,Gi}}$ and $v_{\text{y,Gi}}$, and the x- and y-position of the joints are x_{Gi} and y_{Gi}. Gi denotes either the toe joint, G1, or the ankle joint, G2. In Equation (3.66), $\text{k}_{\text{y,Gi}}$ is the foot's spring constant for the vertical compression of the foot and $\text{c}_{\text{y,Gi}}$ is the damping constant relating to the vertical speed of the foot. This only occurs when the foot moves downwards, and is accounted for by the minimum function, $\min(0, v_{\text{y,Gi}})$, defining the speed to only be included when it is negative. In Equation (3.67), $\text{k}_{\text{x,Gi}}$ is the spring constant for the foot's horizontal compression, depending on the difference between the calculated displaced ground contact point of $x_{\text{C,Gi}}$ (see Equation (3.70)) and the measured joint x-position, x_{Gi}, from the HuMoD data. Under the assumption of static friction, there will be damping of the foot in the x-direction as well, where $\text{c}_{\text{x,Gi}}$ is the damping constant working in both the positive and negative direction.

Furthermore, the following conditions are implemented in the code for both the toe and ankle joint:

$$F_{\text{y,G1}} = F_{\text{x,G1}} = 0 \qquad \text{if} \qquad \text{y}_{\text{G1,thres}} < y_{\text{G1}} \tag{3.68}$$

$$F_{\text{y,G2}} = F_{\text{x,G2}} = 0 \qquad \text{if} \qquad \text{y}_{\text{G2,thres}} < y_{\text{G2}} \tag{3.69}$$

The conditions of Equations (3.68) and (3.69) define when the foot comes into contact with the ground. The threshold values $\text{y}_{\text{G1,thres}}$ and $\text{y}_{\text{G2,thres}}$ are determined based on the HuMoD data. The upper part of Figure 3.8 shows the ground reaction torque, M_{GRF}, which is calculated from Equation (3.61) using HuMoD data. The lower part of Figure 3.8 shows the y-positions of the toe and the ankle joints.

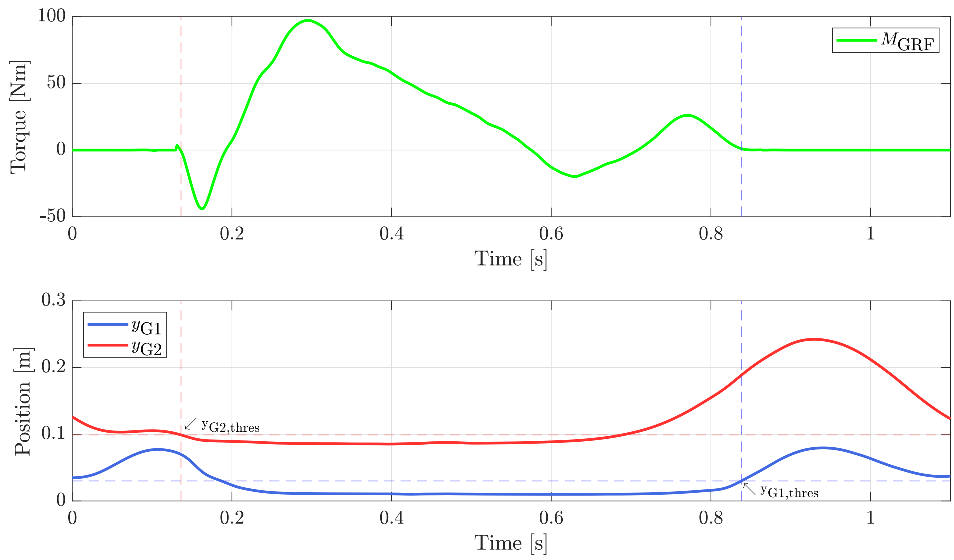


Figure 3.8: Upper figure: the ground reaction torque calculated from HuMoD data. Lower figure: the y-position of the toe and ankle joints. The red and blue dashed lines represent the points at which the threshold values are chosen.

Since the ankle is the first joint to come into contact with the ground during the gait cycle, the threshold value $y_{G2,thres}$ is given by y_{G2} at the time when M_{GRF} first becomes non-zero. This is shown with the red dashed lines in Figure 3.8. Likewise, the toe is the last joint to leave the ground during the gait cycle, which means that $y_{G1,thres}$ is determined as the value for y_{G1} at the time M_{GRF} becomes zero again. This is shown with the blue dashed lines in Figure 3.8.

Equations (3.66) to (3.69) are the basis of the GRF and implemented in the code as illustrated in the flowchart of Figure 3.9.

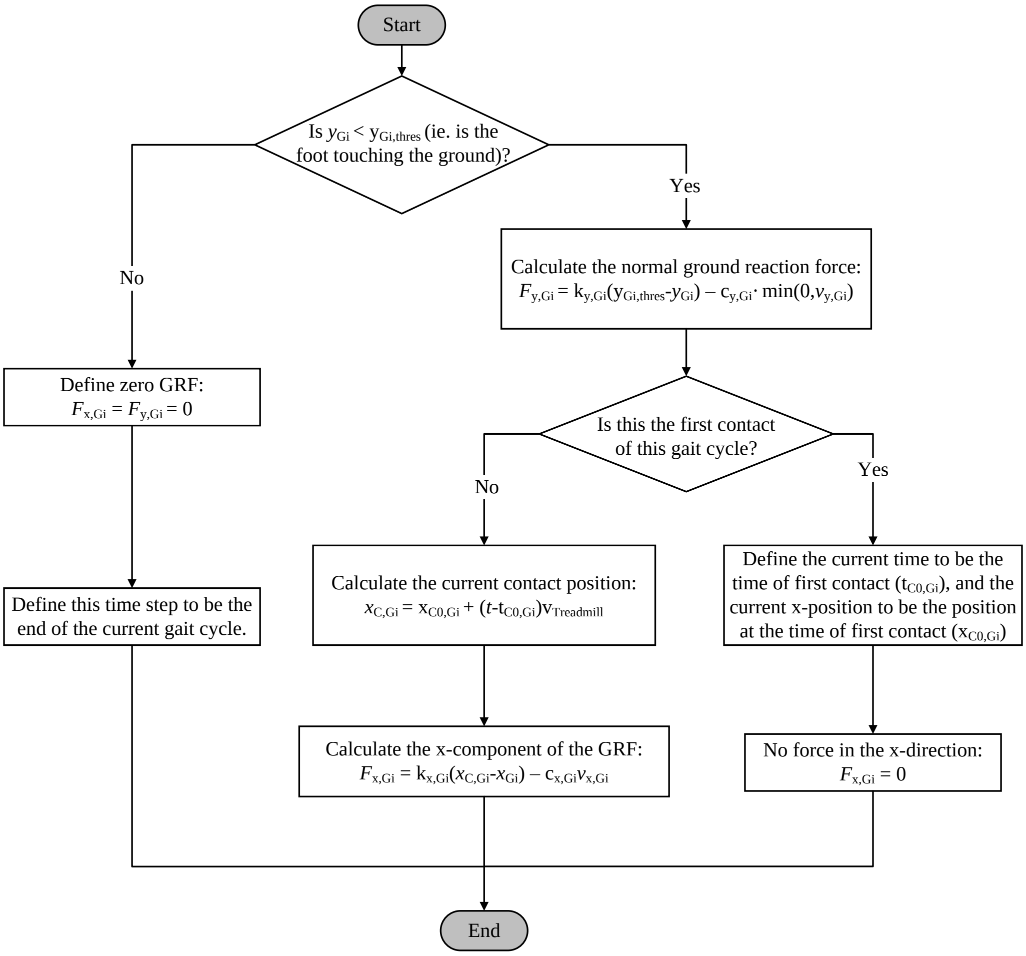


Figure 3.9: General flowchart for the code implementation of the GRF on the joint, Gi.

This figure shows the flow diagram for the ground reaction forces on the toe joint, G1, and the ankle joint, G2 (described by the general indicator, Gi). The implementation is based on several if-else statements. Firstly, it is checked whether or not the joint is actually touching the ground by comparing the y-value of the joint position, y_{Gi}, to the threshold value, $y_{Gi,thres}$, defining the hard boundary between no ground contact and full ground contact. If there is ground contact, then the normal force, $F_{y,Gi}$, is calculated. Afterwards, it is checked whether or not it is the first time during the gait cycle that the joint touches the ground. If yes, then the current time is set to be the time of first contact, $\mathrm{t}_{\mathrm{C0,Gi}}$, and the x-position at this time is defined to be the x-position at the time of first contact, $\mathrm{t}_{\mathrm{C0,Gi}}$. Furthermore, the tangential force, $F_{\mathrm{x,Gi}}$, is set to zero, since this force is based on the movement of the foot after the first contact. The two parameters, $\mathrm{t}_{\mathrm{C0,Gi}}$ and $\mathrm{x}_{\mathrm{C0,Gi}}$, are used for the next calculations when the joint still has ground contact after the time of first contact. The displaced ground contact point, $x_{\mathrm{C,Gi}}$, is given by:

$$x_{\mathrm{C,Gi}} = \mathrm{x}_{\mathrm{C0,Gi}} + (t - \mathrm{t}_{\mathrm{C0,Gi}})\mathrm{v}_{\mathrm{Treadmill}} \tag{3.70}$$

Here, $\mathrm{v}_{\mathrm{Treadmill}}$ is the speed of the treadmill used for collecting the HuMoD data , and t is the current time. Then, $x_{\mathrm{C,Gi}}$ is used to calculate the tangential force, $F_{\mathrm{x,Gi}}$.

Going back to the start of the flow diagram, if a situation happens where there is no ground contact of the joint, then both ground reaction forces are set to zero for this joint. Furthermore, the current gait cycle is defined to be finished, thus allowing for new first contact values to be defined when the joint once again comes into contact with the ground in the next gait cycle.

Finally, it is desired to find a function describing how the contact point alternates between the ankle and toe during a gait cycle. Looking at the HuMoD data plotted in Figure 3.10, it can for one gait cycle be seen that the x-position of the ground contact point, x_{C}, switches between the x-positions of the ankle, x_{G2} and toe, x_{C} when these three parameters are plotted on the y-axis against the difference between the x-positions of the ankle and hip, $x_{\mathrm{G2}} - \mathrm{x}_{\mathrm{H}}$.

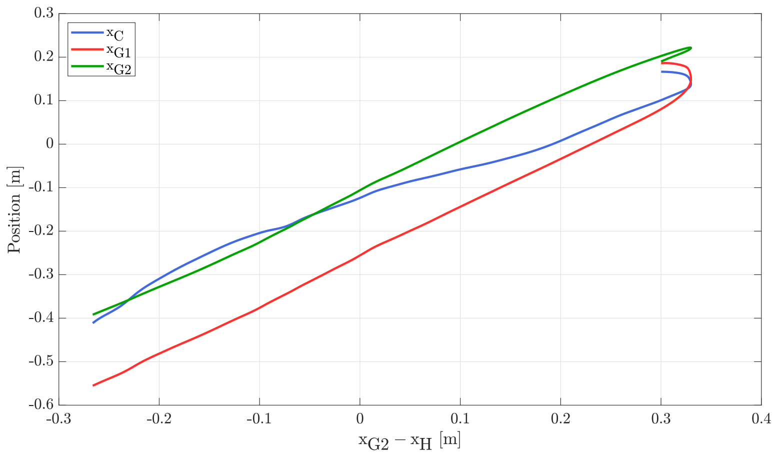


Figure 3.10: Plot showing the contact point, x_{C}, switching between the ankle position, x_{G2}, and the toe position, x_{G1}, when plotted against the difference between the ankle and hip, $x_{\mathrm{G2}} - \mathrm{x}_{\mathrm{H}}$ during one gait cycle.

A function describing how the contact point alternates between the ankle and toe during a gait cycle is:

$$\alpha = 0.5 + 0.5\tanh\big(\mathrm{a}_1(x_{\mathrm{G2}} - \mathrm{x}_{\mathrm{H}} + \mathrm{a}_2)\big) \tag{3.71}$$

Here, a_1 affects how rapid the curve changes from 0 to 1, and a_2 affects the horizontal displacement of the curve. These parameters are chosen to be: $\mathrm{a}_1 = 10$ and $\mathrm{a}_2 = 0.1$.

Using Equation (3.71), the x-location of the GRF can be found:

$$\mathrm{x}_{\mathrm{GRF}} = x_{\mathrm{G2}}(1-\alpha) + x_{\mathrm{G1}}\alpha \tag{3.72}$$

This expression states that when $\alpha = 0$, the GRF works solely on the ankle, and when $\alpha = 1$, the GRF works solely on the toe. Likewise, the GRF will work equally on the toe and ankle when $\alpha = 0.5$. The total GRF in the x- and y-direction are then found using Equations (3.66), (3.67) and (3.71) and the same principle as in Equation (3.72):

$$F_{\mathrm{x}} = F_{\mathrm{x,G2}}(1-\alpha) + F_{\mathrm{x,G1}}\alpha \tag{3.73}$$

$$F_{\mathrm{y}} = F_{\mathrm{y,G2}}(1-\alpha) + F_{\mathrm{y,G1}}\alpha \tag{3.74}$$

The torque components induced by the GRF around the knee joint, x_{G3}, are then:

$$M_{\mathrm{x}} = F_{\mathrm{x}}(\mathrm{x}_{\mathrm{GRF}} - x_{\mathrm{G3}}) \tag{3.75}$$

$$M_{\mathrm{y}} = F_{\mathrm{y}}(\mathrm{x}_{\mathrm{GRF}} - x_{\mathrm{G3}}) \tag{3.76}$$

And the total ground reaction torque:

$$M_{\mathrm{GRF}} = M_{\mathrm{x}} + M_{\mathrm{y}} \tag{3.77}$$

Equation (3.77) replaces the ground reaction torque of Equation (3.61) which is based on HuMoD data. In order to verify the fidelity of the ground reaction force model, the ideal ground reaction torque presented in the upper part of Figure 3.8 is compared to the result obtained with the model in Figure 3.11. The GRF model constants are shown in Table 3.5.

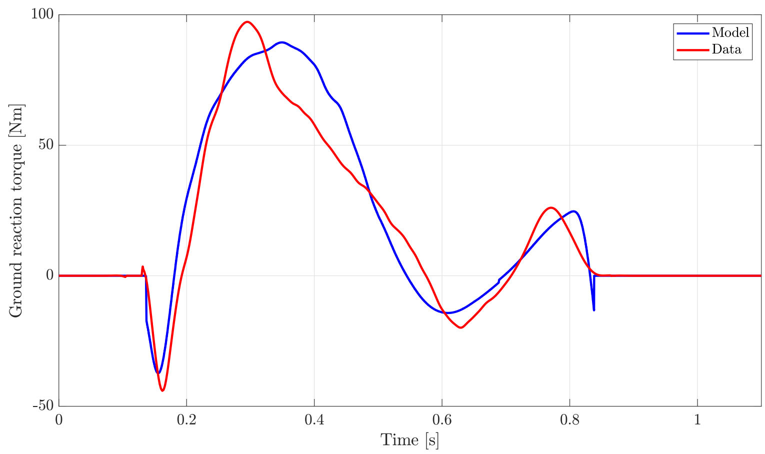


Figure 3.11: Plot of M_{GRF} calculated using HuMoD data (red curve), and calculated using the GRF model presented in this section (blue curve).

In Figure 3.11, the deviation between the model curve (blue curve) and the data curve (red curve) is most significant at the peak torque, where it the model undershoots the data initially, and then sees a slower decrease than the data. Furthermore, there is a deviation between the curves in the negative spike for the model torque that occurs just as the toe leaves the ground in the final part of the gait cycle. But generally, the data curve is closely

Constant	Description	Value	Unit
$y_{G1,thres}$	Toe threshold value	0.03	[m]
$y_{G2,thres}$	Ankle threshold value	0.099	[m]
$k_{x,G1}$	Toe x-axis spring constant	694.9	[N/m]
$k_{y,G1}$	Toe y-axis spring constant	13782	[N/m]
$c_{x,G1}$	Toe x-axis damping constant	$1.021 \cdot 10^{-4}$	[N·s/m]
$c_{y,G1}$	Toe y-axis damping constant	997.1	[N· s/m]
$k_{x,G2}$	Ankle x-axis spring constant	8634	[N/m]
$k_{y,G2}$	Ankle y-axis spring constant	64851	[N/m]
$c_{x,G2}$	Ankle x-axis damping constant	53.30	[N· s/m]
$c_{y,G2}$	Ankle y-axis damping constant	$1.044 \cdot 10^{-4}$	[N·s/m]

Table 3.5: Constant values used in the GRF model.

represented by the model curve, and therefore it is assumed to be of high enough fidelity to be used for controller evaluation through simulations which is done by substituting Equation (3.77) with Equation (3.61). It is found that setting the length of the shank and foot as constants in the simulation gives bad correspondence between the model and the HuMoD data. This can be explained by the method used for collecting the HuMoD data, where reflective markers were placed on the body of the test subjects. Since the human skin moves slightly during motion, this movement will also be present in the recorded data. When using the GRFM, these small deviations actually prove immensely significant, since the model is based on the compression of the foot tissue using spring coefficients. As seen in Table 3.5, these coefficients are of vary large values, meaning that even small displacements result in large force reactions. This is discussed in further detail in Chapter 6. Through simulative evaluation it was found that the GRFM most closely resembling the HuMoD data was achieved with the length of the foot and shank being calculated based on the predetermined HuMoD data. This in turn means that the GRFM does not achieve full state dependency. Despite this flaw, the model is still used in the future parts of the report for simulative evaluation.

3.3 Model Linearization

The shank acceleration of Equation (3.78) which relates the knee torque to the change in knee acceleration is used in order to implement a prediction model that can be used for model predictive control (MPC). If this approach is to be implemented in real time, it is necessary to linearize the differential equation, since it would take too long time to process an optimization algorithm when the equation is in its non-linear form. The goal is to obtain a model that can be used for predicting the future states of the system according to the chosen inputs. The shank acceleration is a function dependent on two variables.

$$\ddot{\phi}_2 = f(M_{\text{hyd}}, M_{\text{ext}}) \tag{3.78}$$

When linearizing the system, the term M_{ext} causes a challenge, since this term is highly non-linear as well as discontinuous as seen in Section 3.2.4 where the GRFM is presented. Therefore, this term is accounted for by assuming access to measurements of the total knee torque, M_{total}, with a torque transducer or strain gauge. Thereby, the external torque can be found using Equation (3.65):

$$M_{\text{total}} = \ddot{\phi}_2(J_{12} + \mathrm{J}_{22}) = -M_{\text{hyd}} + M_{\text{ext}} \implies M_{\text{ext}} = M_{\text{total}} + M_{\text{hyd}} \tag{3.79}$$

The non-linear terms can be taken into account in the prediction model by using the measured external torque, M_{ext}. Thereby the model's prediction power can be increased. Equation (3.78) is linearized by using the first order Taylor series:

$$\begin{aligned}\Delta\ddot{\phi}_2 = \left.\frac{\partial\ddot{\phi}_2}{\partial M_{\text{hyd}}}\right|_{\mathrm{x}_0} \Delta M_{\text{hyd}} + \left.\frac{\partial\ddot{\phi}_2}{\partial M_{\text{ext}}}\right|_{\mathrm{x}_0} \Delta M_{\text{ext}} &= \mathrm{K}_{\text{hyd}}\Delta M_{\text{hyd}} + \mathrm{K}_{\text{ext}}\Delta M_{\text{ext}} \\ &= \mathrm{K}_{\text{hyd}} \tfrac{\mathrm{d}z_1}{\mathrm{d}\psi}_0 \Delta F_{\text{hyd}} + \mathrm{K}_{\text{ext}}\Delta M_{\text{ext}}\end{aligned} \tag{3.80}$$

Where the linearization point is:

$$\mathrm{x}_0 = \left[\phi_{1,0}\ \phi_{2,0}\ \tfrac{\mathrm{d}z_1}{\mathrm{d}\psi}_0\right] \tag{3.81}$$

And the linearization constants are:

$$\begin{aligned}\mathrm{K}_{\text{hyd}}^{-1} = -\Big(&\mathrm{l}_2\mathrm{m}_1(\mathrm{l}_1 - \zeta_{\text{CM1}})\cos(-\phi_{2,0} + \phi_{1,0}) + \sin(-\phi_{2,0} + \phi_{1,0})\eta_{\text{CM1}}\mathrm{l}_2\mathrm{m}_1 \\ &+ (\mathrm{m}_1 + \mathrm{m}_2)\mathrm{l}_2{}^2 - 2\,\mathrm{l}_2\mathrm{m}_2\zeta_{\text{CM2}} + (\eta_{\text{CM2}}{}^2 + \zeta_{\text{CM2}}{}^2)\mathrm{m}_2 + J_2\Big)\end{aligned} \tag{3.82}$$

$$\mathrm{K}_{\text{ext}} = -\mathrm{K}_{\text{hyd}} \tag{3.83}$$

This concludes the system modelling which serves as the foundation for the controller development presented in the next chapter.

4 Controller Designs

Chapter Summary

This chapter presents the controllers developed for tracking the position of the knee angle during a gait cycle. Firstly, a reactive approach is presented, consisting of a conventional PD-controller with torque compensation and torque feed-forward. This is followed by a presentation of two force switching algorithms, used for converting the continuous controller input to the discrete input values for the system's valve commands. Lastly, the model predictive controller is presented along with the optimization algorithm used for its implementation.

4.1 Reactive Control

As a benchmark for evaluating the controller performance of the MPC, conventional reactive control is implemented. A block diagram of the controller structure is shown in Figure 4.1. Here, feedforward from the shank acceleration is added using the linearized system model of Equation (3.80) neglecting the external disturbance. Furthermore, torque compensation is added to the controller output under the assumption of having access to measurements of the external torque of M_{ext}. Adding the compensation and feed-forward terms to the controller output serves as a baseline for the hydraulic actuators, which means that the controller in turn has to do less work.

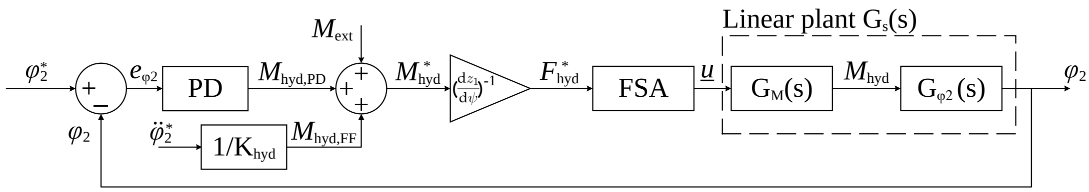


Figure 4.1: Block diagram of reactive controller structure.

The transfer function for the motion dynamics, $G_{\phi 2}(s)$, is found by Laplace transforming Equation (3.80). For the linear system analysis, the external torque term, M_{ext}, is assumed to consist only of the viscous friction term. Furthermore, the angular velocity of the thigh is assumed to be zero, yielding $M_{\text{ext}} = M_{\text{fric}} = \beta_{\text{eq}}(\dot{\phi}_3 - \dot{\phi}_2) = -\beta_{\text{eq}}\dot{\phi}_2$. The transfer function is then derived:

$$\phi_2 s^2 = \mathrm{K}_{\text{hyd}} M_{\text{hyd}} - \beta_{\text{eq}} \mathrm{K}_{\text{ext}} \phi_2 s \tag{4.1}$$

$$\implies G_{\phi 2}(s) = \frac{\phi_2}{M_{\text{hyd}}} = \frac{\mathrm{K}_{\text{hyd}}}{s^2 + \mathrm{K}_{\text{ext}} \beta_{\text{eq}} s} \tag{4.2}$$

The actuator torque is dependent on the actuator force, which in turn is dependent on the chamber pressures. But there is a delay in between a set of valve commands are given until the desired chamber pressure levels are achieved. Keeping in mind that the chamber pressures are directly related to the actuator force, the transfer from valve commands to actuator force, is modelled as the response of a first order transfer function, which resembles the pressure build up in a constant volume through a fixed orifice:

$$G_{\mathrm{F}}(s) = \frac{F_{\mathrm{hyd}}}{M^{*}_{\mathrm{hyd}}} = \frac{1}{\tau_{\mathrm{F}} s + 1} \tag{4.3}$$

Please note that the FSA's conversion from reference value to force level index is assumed to be a unit gain. Equation (4.3) is converted to actuator torque with the transmission ratio to get the delay transfer function $G_{\mathrm{M}}(s)$:

$$\implies G_{\mathrm{M}}(s) = \frac{M_{\mathrm{hyd}}}{M^{*}_{\mathrm{hyd}}} = \frac{F_{\mathrm{hyd}} \frac{\mathrm{d}z_1}{\mathrm{d}\psi}_0}{M^{*}_{\mathrm{hyd}}} = G_{\mathrm{F}}(s) \frac{\mathrm{d}z_1}{\mathrm{d}\psi}_0 = \frac{\frac{\mathrm{d}z_1}{\mathrm{d}\psi}_0}{\tau_{\mathrm{F}} s + 1} \tag{4.4}$$

From the analysis presented in Appendix A, it is found that an approximation of the slowest delay time between a valve command is given and the pressure level has settled, is 6.6 ms. Using this, and defining the settling time to be four times the time constant, then the first order time constant is $\tau_{\mathrm{F}} = 1.65$ ms.

The linearization point for the linear model is chosen to be based on the HuMoD data during the gait cycle when the time is t=0.23 s. This point is during the stance phase when the most torque is required during the gait cycle, and therefore it is deemed a suitable working point for the controller design:

$$\mathrm{x}_0(\mathrm{t} = 0.23s) = \begin{bmatrix} \phi_{1,0} & \phi_{2,0} & \frac{\mathrm{d}z_1}{\mathrm{d}\psi}_0 \end{bmatrix} = \begin{bmatrix} 27^\circ & 89^\circ & 0.09 \frac{\mathrm{m}}{\mathrm{rad}} \end{bmatrix} \tag{4.5}$$

Using this linearization point, the linearization constants of Equations (3.82) and (3.83) become: $\mathrm{K}_{\mathrm{hyd}} = -2.4$ and $\mathrm{K}_{\mathrm{ext}} = 2.4$. The system's open loop transfer function given from Equations (4.1) and (4.4) is:

$$\boxed{G_{\mathrm{s}}(s) = \frac{\phi_2}{M^{*}_{\mathrm{hyd}}} = G_{\phi 2}(s) \cdot G_{\mathrm{M}}(s) = \frac{\mathrm{K}_{\mathrm{hyd}} \frac{\mathrm{d}z_1}{\mathrm{d}\psi}_0}{s(s + \mathrm{K}_{\mathrm{ext}} \beta_{\mathrm{eq}})(\tau_{\mathrm{F}} s + 1)}} \tag{4.6}$$

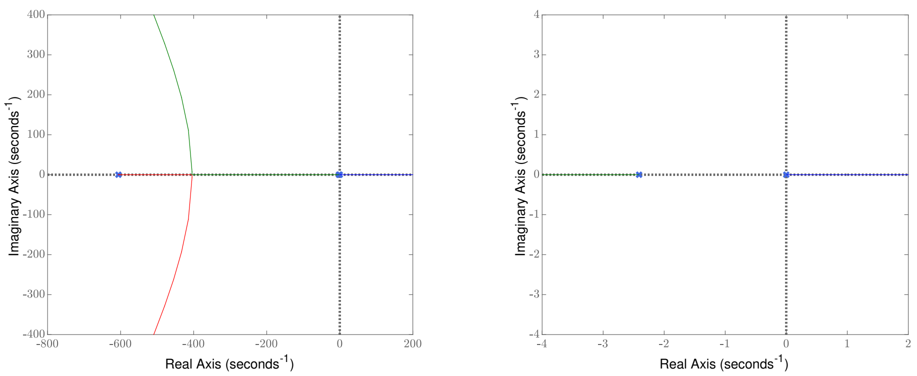


Figure 4.2: Root locus plots. Left: the open loop system transfer function given by Equation (4.6). Right: zoomed plot of the open loop system transfer function.

Looking at Figure 4.2, it can be seen that one closed loop pole always will be located in the right half plane. Therefore it is desired to stabilize the system in the closed loop and achieve as high a bandwidth as possible without having the controller amplify the measurement noise that might be present in the physical system. This is done by implementing a PD-controller structure given by Equation (4.7), since both the proportional and derivative gain increases the bandwidth of the controller.

$$\boxed{G_{\mathrm{PD}}(s) = \frac{M_{\mathrm{hyd,PD}}}{e_{\phi 2}} = -(\mathrm{K_P} + \mathrm{K_D} s)\frac{1}{\tau_{\mathrm{lp}} s + 1}} \tag{4.7}$$

In order to make the transfer function proper, a low pass filter is added to the controller structure in order to filter the error input.
The controller's gain parameters are chosen to be:

$\mathrm{K_P}$ = 106.2	$\mathrm{K_D}$ = 29.5	τ_{lp} = 0.0458

Table 4.1: Overview of PD-controller parameters.

These parameters are found from an initial tuning based on Equation (4.6) using the Matlab extension 'Control System Designer', followed by an iterative tuning through simulations. The controller transfer function implements a pole in $-\frac{1}{\tau_{\mathrm{lp}}} = -21.82$ rad/s and a zero in $-\frac{\mathrm{K_P}}{\mathrm{K_D}} = -3.65$ rad/s. Placing the pole and zero here stabilizes the system as can be seen in Figure 4.3, which shows the the root locus plot for the PD-controller applied to the open loop transfer function. Please note that the figure is zoomed in, so that the fast pole of Equation (4.4) at s=-606 rad/s can not be seen.

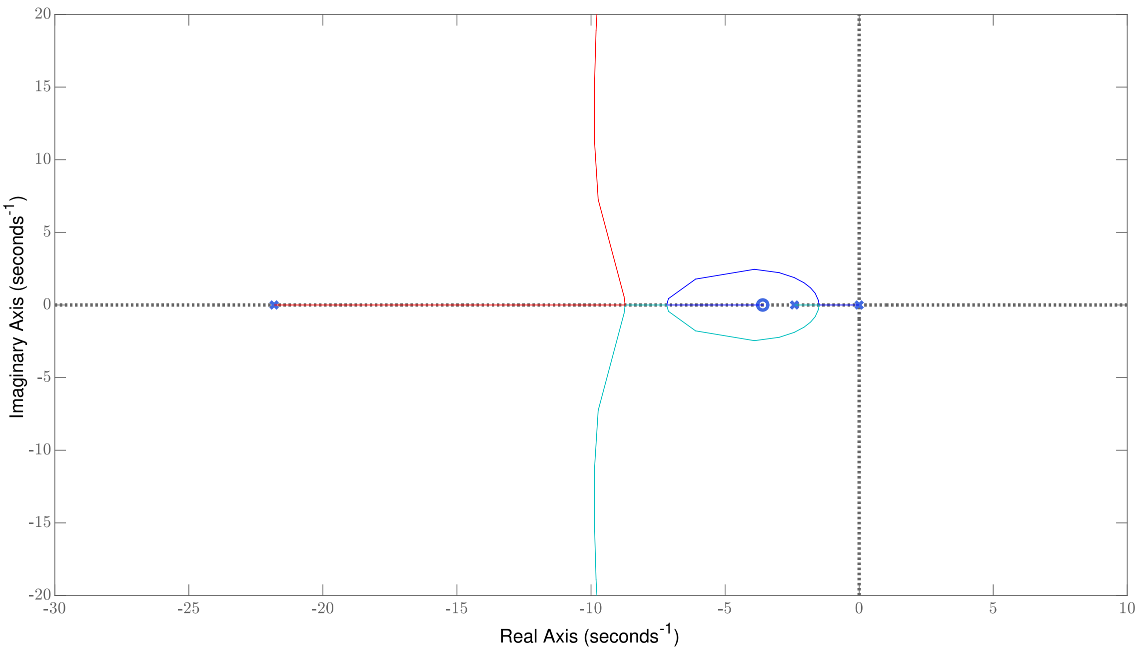


Figure 4.3: Root locus for controller applied to system: $G_{\mathrm{s}}(s)G_{\mathrm{PD}}(s)$.

Applying the PD-controller to the system transfer function and closing the loop gives the bode plot and step response seen in Figure 4.4.

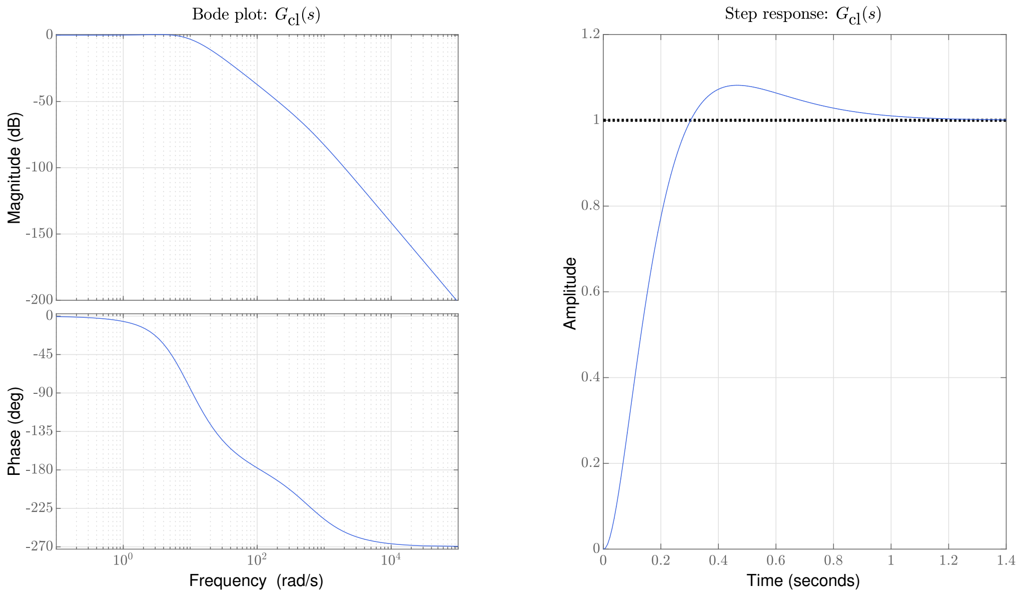


Figure 4.4: Left: closed loop bode plot. Right: closed loop step response.

Settling time	0.87 s
Rise time	0.2 s
Overshoot	8.2%

Table 4.2: Step response data.

Data for the step response is shown in Table 4.2. The response is slow, which the closed loop bandwidth of 10 rad/s (seen from Figure 4.4) also tells. Through iterative tuning on the gain parameters for K_P and K_D it was attempted to further increase the bandwidth, but it was found that this causes poor controlled response with large position oscillations in simulations when the controller was applied to the non-linear system. In the simulative controller evaluations (shown in Chapter 5), noise is added on the measured position signal used in the controller in order to test the controller under conditions resembling that on a real test setup. This is important because the derivative term of the controller is sensitive towards noise disturbances.

4.1.1 Force Switching Algorithm[1]

In order to be able to use the reactive controller on the system, it is necessary to implement a force switching algorithm (FSA) which converts the continuous force reference given by the PD-controller to a discrete row index of the control input matrix $\underline{\underline{u}}$ of Equation (3.10).

[1]The FSA2 derived in this section is based on the work outlined in [9].

FSA1

The most simple choice of FSA-structure would be to pick the input vector corresponding to the force level, F (Equation (3.9)), which is closest to the force reference F^* [7]. This structure is denoted FSA1:

$$\boxed{F_{\text{hyd}} = \underline{\text{F}}[k_{\text{idx}}] \quad ; \quad k_{\text{idx}} = \arg\min(|F^* - \underline{\text{F}}[k_{\text{idx}}]| + \text{W} \cdot k_{\text{idx,change}})} \tag{4.8}$$

Here, the term W is a weighting term that penalizes changing the force index. The coefficient $k_{\text{idx,change}}$ is a binary variable, which is defined to be 1 when the new force level index considered is different from the value currently applied to the system:

$$u_{\text{change}} = \begin{cases} 1 & \text{if} \quad k_{\text{idx}} \neq k_{\text{idx,prev}} \\ 0 & \text{if} \quad k_{\text{idx}} = k_{\text{idx,prev}} \end{cases} \tag{4.9}$$

In order to avoid high frequency switching of the valves, the FSA is sampled in 10 ms intervals. This prohibits premature switching before the chamber pressures have been allowed due time to rise to the desired levels (see Appendix A for an analysis of the system's slowest pressure transient time).

FSA2

A second structure for the FSA is also implemented. This is denoted FSA2. This approach takes into account the energy losses associated with changing between pressure levels. This is because each time a shift in a chamber's pressure level occurs, energy will be lost. Figure 4.5 shows an example of the A-chamber pressure, p_{A}, and flow, Q_{A}, during a shift from being connected to the low pressure rail to the high pressure rail. This data is from the simulations of the system controlled for a gait cycle, and as it can be seen, there is a non-zero flow when the pressure is in steady state which is because of the chamber volume changing.

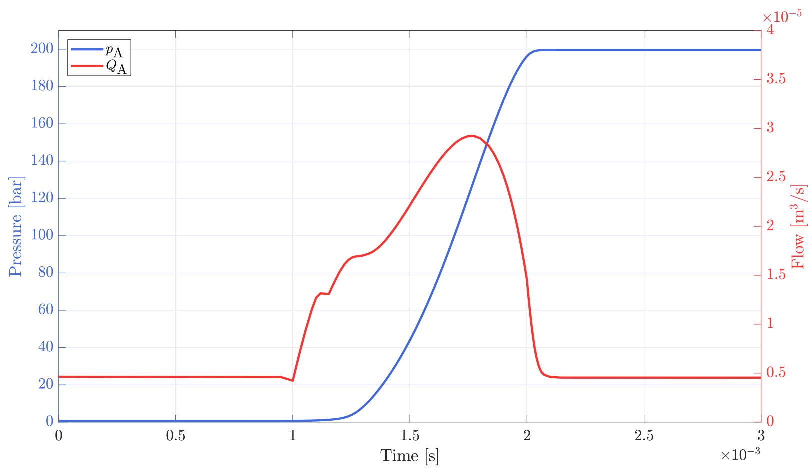


Figure 4.5: Pressure and flow for the A-chamber during a shift from low to high pressure.

Hydraulic energy is given by:

$$E_{\text{sh}} = \int_{t_0}^{t_1} p_{\text{A}} Q_{\text{A}} \text{d}t \tag{4.10}$$

Where t_0 and t_1 denote the start and end of the switching event respectively. Assuming a constant bulk modulus, β, and a constant chamber volume, V, the energy loss for when a volume is connected to a pressure rail with constant pressure can be derived. Firstly, the flow, Q_C, required for a pressure change in the chamber pressure, p_C, from p_0 to p_1, is described by [9]:

$$\dot{p}_{\text{C}} = \frac{\beta}{\text{V}} Q_{\text{C}} \implies p_{\text{C}} = \frac{\beta}{\text{V}} \int Q_{\text{C}} \text{d}t + p_0 \tag{4.11}$$

As time goes towards infinity, the pressure in the chamber will go towards that of the high pressure rail: $\lim_{t \to \infty} p_{\text{C}} = p_1$. This means that the fluid volume needed for the pressure change will be given by [9]:

$$V_{\text{C}} = \int_0^{\infty} Q_{\text{C}} \text{d}t = \frac{V}{\beta}(p_1 - p_0) \tag{4.12}$$

The energy loss can then be defined as the difference between the energy supplied by the source, E_{S}, and the energy stored in the chamber, E_{C} [9]:

$$E_{\text{sh}} = E_{\text{S}} - E_{\text{C}} = \int_0^{\infty} p_1 Q_{\text{C}} \text{d}t - \int_0^{\infty} p_{\text{C}} Q_{\text{C}} \text{d}t \tag{4.13}$$

Equation (4.11) is rewritten to:

$$p_{\text{C}} = \frac{\beta}{\text{V}} \int Q_{\text{C}} \text{d}t + p_0 = \frac{\beta}{\text{V}} V_{\text{Q}} + p_0 \tag{4.14}$$

$$V_{\text{Q}} = \int Q_{\text{C}} \text{d}t \qquad \implies \qquad \dot{V}_{\text{Q}} = Q_{\text{C}} \tag{4.15}$$

Where V_{Q} is the chamber flow volume for an indefinite time range. Then, Equation (4.13) is manipulated using Equations (4.12), (4.14) and (4.15):

$$\begin{aligned} E_{\text{sh}} &= \int_0^{\infty} p_1 Q_{\text{C}} \text{d}t - \int_0^{\infty} p_{\text{C}} Q_{\text{C}} \text{d}t = p_1 V_{\text{C}} - \int_0^{\infty} \left(\frac{\beta}{\text{V}} V_{\text{Q}} + p_0 \right) Q_{\text{C}} \text{d}t \\ &= p_1 V_{\text{C}} - p_0 \int_0^{\infty} Q_{\text{C}} \text{d}t - \frac{\beta}{\text{V}} \int_0^{\infty} V_{\text{Q}} \dot{V}_{\text{Q}} \text{d}t = (p_1 - p_0) V_{\text{C}} - \frac{1}{2} \frac{\beta}{\text{V}} \left[V_{\text{Q}}^2 \right]_0^{\infty} \\ &= (p_1 - p_0) V_{\text{C}} - \frac{1}{2} \frac{\beta}{\text{V}} V_{\text{C}}^2 = \frac{V}{\beta}(p_1 - p_0)^2 - \frac{1}{2} \frac{V}{\beta}(p_1 - p_0)^2 = \frac{1}{2} \frac{V}{\beta}(p_1 - p_0)^2 \end{aligned} \tag{4.16}$$

Equation (4.16) describes the losses associated with switching the pressure from one level to another. This pressure loss is independent of valve area and dynamics, and therefore, this is an inevitable loss that will occur when compressing fluid [9]. The total energy loss for all four cylinder chambers when switching the force level, is then given by the summation of the energy losses for all four chambers:

$$\boxed{E_{\text{sh}}(x, y) = \sum_{i=1}^{4} \frac{1}{2} \frac{V_i}{\beta} (p[u_y(i)] - p[u_x(i)])^2} \tag{4.17}$$

In this equation, i denotes the i'th cylinder chamber. Equation (4.17) is used to design a more energy efficient force switching algorithm than the one shown in Equation (4.8):

$$\boxed{F_{\text{hyd}} = \underline{\mathrm{F}}[k_{\text{idx}}] \quad ; \quad k_{\text{idx}} = \underset{k_{\text{idx}} \in \{k_{\text{idx-}}\ ,\ k_{\text{idx,prev}}\ ,\ k_{\text{idx+}}\}}{\text{arg min}} \left(F^* - \underline{\mathrm{F}}[k_{\text{idx}}]\right)} \tag{4.18}$$

Here, the row index of the sorted force output matrix, k$_idx$, is chosen among three different candidates $k_{\text{idx-}}, k_{\text{idx,prev}}, k_{\text{idx+}}$. These are chosen based on the energy loss equation according to the following conditions:

$$\boxed{\begin{aligned} k_{\text{idx-}} &= \text{arg} \min_{k_{\text{idx}} \in S_{-}} E_{\text{sh}}(k_{\text{idx,prev}}, k_{\text{idx}}) \quad ; \quad S_{-} = \{F^* - \mathrm{F_b} < \underline{\mathrm{F}}[k_{\text{idx}}] < F^*\} \\ k_{\text{idx+}} &= \text{arg} \min_{k_{\text{idx}} \in S_{+}} E_{\text{sh}}(k_{\text{idx,prev}}, k_{\text{idx}}) \quad ; \quad S_{+} = \{F^* < \underline{\mathrm{F}}[k_{\text{idx}}] < F^* + \mathrm{F_b}\} \end{aligned}} \tag{4.19}$$

Here, $\mathrm{F_b}$ denotes the force band value defining a region around the force reference, $F^* \pm \mathrm{F_b}$. Within this region, the row indices are found for which the least amount of energy is used whilst still keeping the force output within the defined reference band. $k_{\text{idx,prev}}$ is the same index as at the previous time step. $k_{\text{idx-}}$ is the index for which the actuator force lies within the band below the reference, and $k_{\text{idx+}}$ is the index for which the actuator force lies within the band above the reference.

The code implementation for the force switching algorithm of Equations (4.18) and (4.19) is outlined in the flowchart of Figure 4.6 on Page 40. The FSA2 is also sampled in 10 ms intervals, as is the case for FSA1. In the initial case where there is no previous time step, the index is defined to be $k_{\text{idx}} = 7$ which corresponds to $\mathrm{F_{hyd}} \approx 0\,\mathrm{N}$. First, all possible force level candidates within the band around the force reference are found and stored in the vectors $\underline{k}_{\text{idx-,c}}$ and $\underline{k}_{\text{idx+,c}}$. Then the code implements checks to see if there are a non-zero number of possibilities. If this is the case, then the switching energy losses, E_{sh}, are found for each of these candidates and stored in the vectors $\underline{E}_{\text{sh-}}$ and $\underline{E}_{\text{sh+}}$. If there are no candidates, then the relevant index is defined to be that of the previous time step. Then, another check is conducted to see if the energy loss vectors are empty. This is necessary in the code implementation in the case when there are no candidates in the reference band, since this will also mean that there will be no calculated values for $\underline{E}_{\text{sh-}}$ and $\underline{E}_{\text{sh+}}$. If this is the case, the indices are defined to be the previously chosen index. If $\underline{E}_{\text{sh-}}$ or $\underline{E}_{\text{sh+}}$ are not empty, then the indices corresponding to the minimum energy losses are found. Finally, force levels resulting from these indices, along with that of the previously chosen index, are then compared to the force reference, and then the choice giving the minimum force error is chosen. This index is defined to be $k_{\text{idx,prev}}$, preparing the algorithm for the next iteration. The algorithm is configured to also include the force level index corresponding to the nearest force reference, and storing it in the vector $\underline{k}_{\text{idx-,c}}$. This is done in order to always take this value into account even though no candidates are within the reference band. This means that in the case where $\mathrm{F_b} = 0$, the FSA2 is equivalent to FSA1.

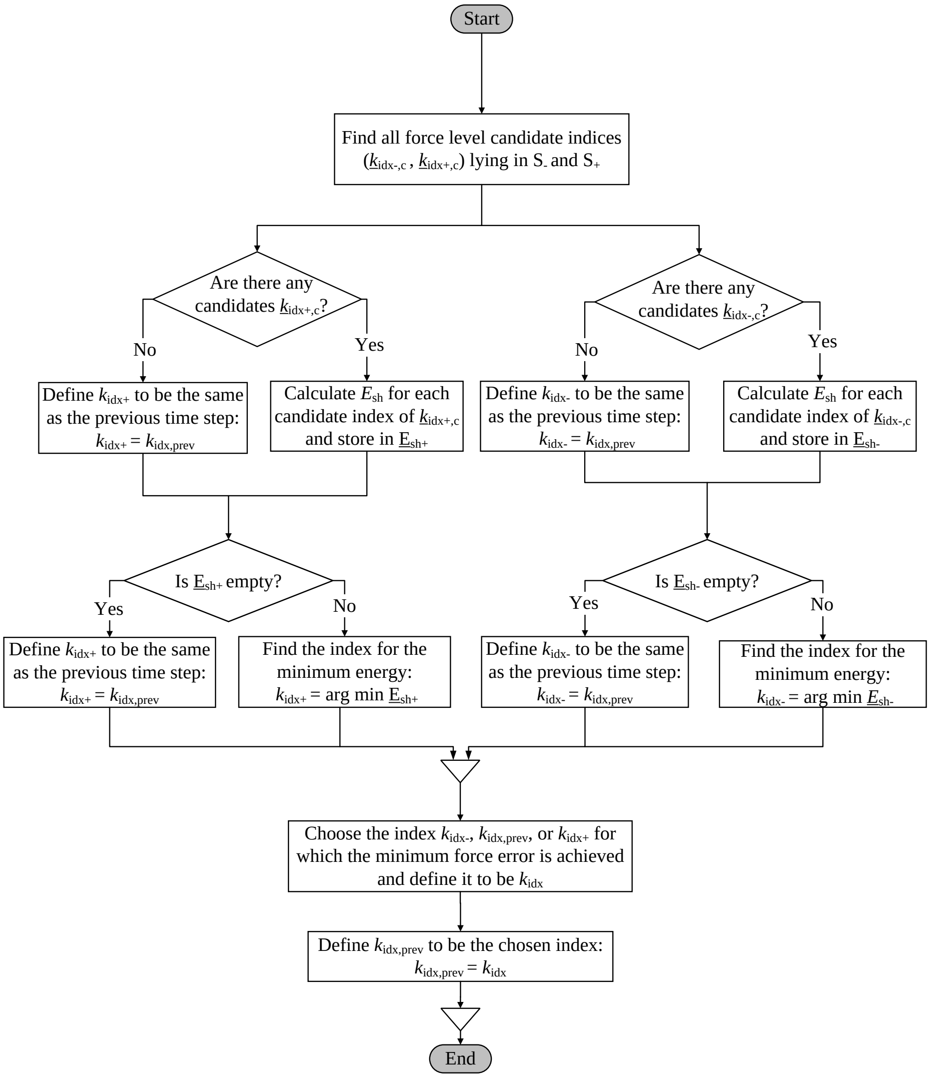


Figure 4.6: Flowchart of the code implementation for FSA2.

Comparative Analysis of FSA1 and FSA2

The two FSA structures are compared against each other in order to examine their tracking precision and energy consumption at different values for their respective tuning constants. The simulations are run with the reactive controller structure presented in Section 4.1, and no ground reaction force is included which is in order to neglect its influence on the response. This makes the results more easily comparable. Furthermore, noise is implemented on the measured position signal used in the controller. Further elaboration on this noise is presented in the beginning of the simulation result chapter on Page 55.

Figure 4.7 shows the energy consumption when simulating the system with the FSA1 and Figure 4.8 shows results found with the FSA2. For both figures, it can on the leftmost plot be seen that the final energy consumption is reduced as the tuning constants are increased. This is a result of the number of pressure switches occurring, which is seen in the plots in the middle, where fewer switches means less energy consumed. On the rightmost plots, the number of switches for each individual chamber is shown. The chambers with the largest volumes are the A and D chambers, meaning that pressurizing these will cost the most energy. Looking at the FSA2 specifically, it can be seen that the approach indeed penalizes changing the pressure levels in these chambers, since the number of switches for these chambers are lowest as the force range constant, F_b, is increased. The lowest energy consumption is achieved by the FSA2 at the highest value for F_b. Interestingly, the FSA2 achieves a lower energy consumption with a higher number of force level switches, which indicates that the algorithm works as intended, with the ability to choose the least energy consuming possibility within the defined reference band.

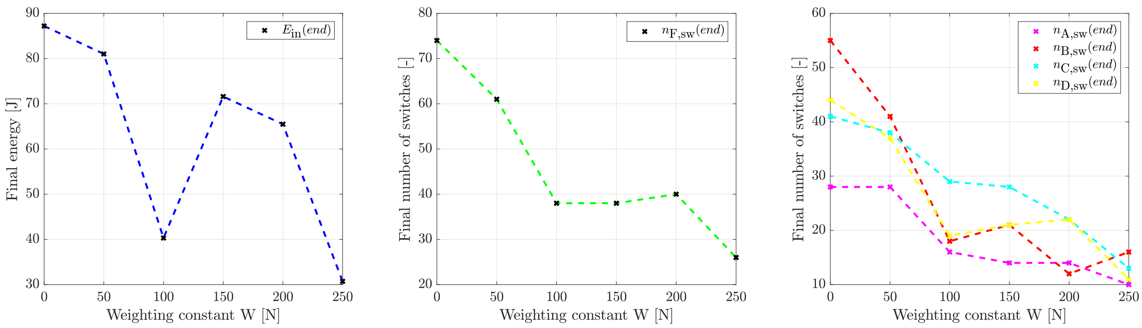


Figure 4.7: Energy and switching with FSA1 at varying weighting constants. Left: energy consumption. Middle: total number of force level switches. Right: individual number of cylinder chamber valve switches.

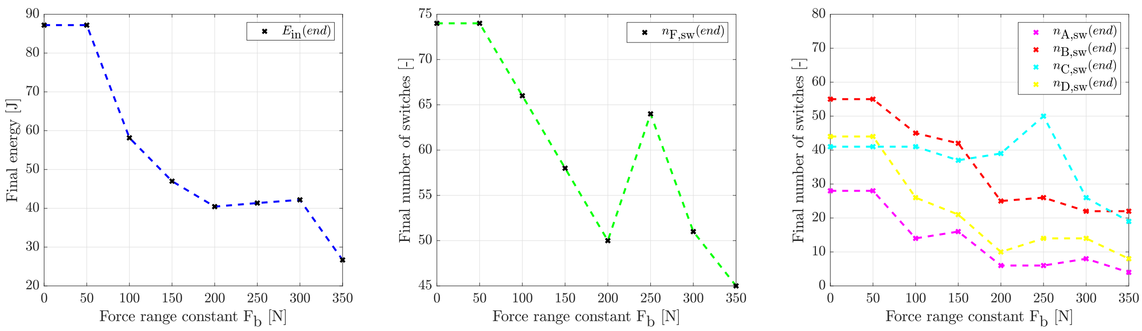


Figure 4.8: Energy and switching with FSA2 at varying weighting constants. Left: energy consumption. Middle: total number of force level switches. Right: individual number of cylinder chamber valve switches.

The tracking precision is also analyzed by looking at the rms position and speed error over the whole gait cycle. Figure 4.9 shows the results for FSA1 and Figure 4.10 shows results for FSA2. It can be noted that the results obtained are the same when $F_b = W = 0$ N, which is expected since they should be equivalent in this case. For the FSA1 it can be said in general, that when the weighting constant is high, the rms tracking errors increase, while the energy consumption decreases. For the FSA2, no clear relationship can be seen for the position rms error, since the error both increases and decreases as the weighting constant is increased. But looking at the speed rms error, it seems like there is an increase in error as the weighting constant increases. Comparing the results from Figure 4.9 with the results from Figure 4.10, the FSA2 gives the best tracking precision. Based on the results, it can be concluded that the FSA2 performs better than the FSA1, since the energy consumption is lower while the tracking performance is better. Therefore, the FSA2 is chosen for future simulative analysis.

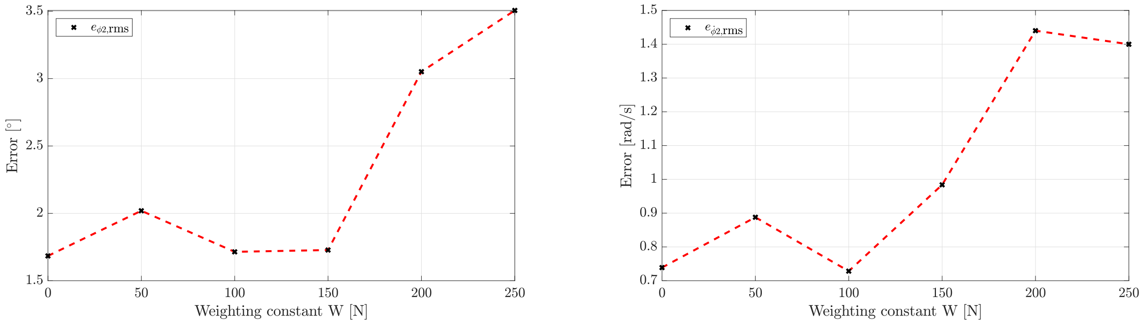


Figure 4.9: Tracking with FSA1 at varying weighting constants. Left: position rms error. Right: speed rms error.

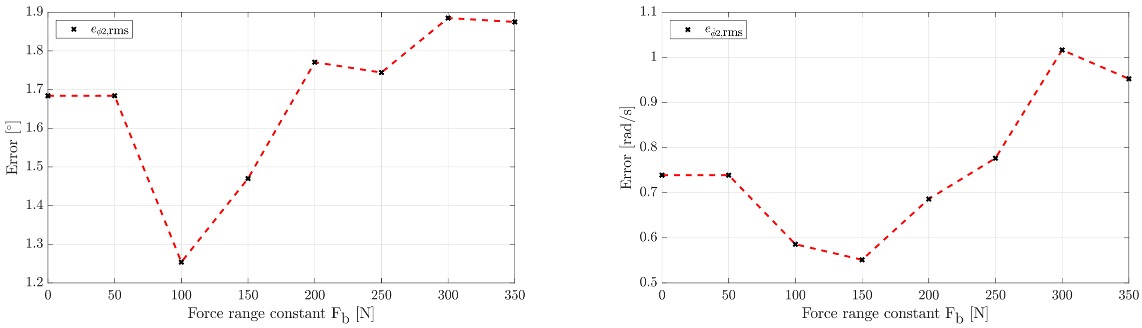


Figure 4.10: Tracking with FSA2 at varying weighting constants. Left: position rms error. Right: speed rms error.

In order to choose F_b, a tradeoff has to be made, since lowering the energy consumption is at the cost of decreasing the tracking precision. The constant is chosen to be $F_b = 250$ N, since both the energy consumption is low and good tracking performance is achieved.

4.2 Model Predictive Control

The model predictive controller predicts the future inputs at a series of time steps in order to achieve the response minimizing a customizable cost function. This cost function is defined to consist of terms that penalize tracking error as well as a term that penalizes energy consumption. Figure 4.11 shows a block diagram of the MPC-structure. The approach uses measurements of the position, speed, transmission ratio and external torque, as well as a known motion trajectory in order to find the valve command vector $\underline{u}$. Please note that stability of the MPC will not be proven in this report.

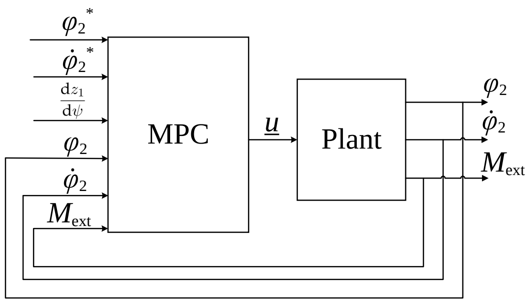


Figure 4.11: Block diagram of MPC structure.

The MPC-algorithm works in three steps [23]:

1. Measure the position, ϕ_2, speed, $\dot{\phi}_2$, transmission ratio, $\frac{\mathrm{d}z_1}{\mathrm{d}\psi}$ and external torque, M_{ext}, at the current time step, t(k).
2. Find the optimal inputs at each time step on the prediction horizon using the measured values from step 1 along with the optimization algorithm (presented in Section 4.2.3) with the prediction model (presented in Equations (4.23) to (4.25)). The optimization algorithm minimizes the cost function defined in Equation (4.27) in order to do this.
3. Apply the input at the current time step, t(k), to the system and repeat from step 1 at the next sample instant.

Figure 4.12 gives an example of the optimized input values found with MPC-principle. The figure shows the force index input values found with the MPC-algorithm for a prediction horizon of NH=5 at t= 0.9 s and t= 0.91 s with a sample time of $T_s = 10$ ms. The blue line shows the system input values found through optimisation at time step t=k, and the red line shows how the algorithm gives a new set of input values for the optimization performed at time step t=k+1. Since only the input at the first time step is applied to the system, the MPC will correct the errors that occur due to deviations in the prediction, because the feedback from the measured states are taken into account for the new optimization at the next time step.

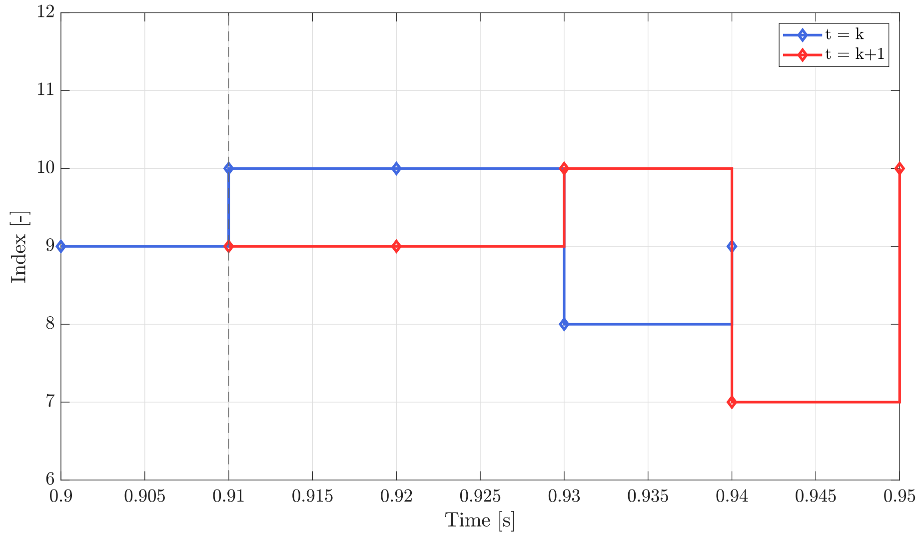


Figure 4.12: Example of the optimized force level indices, k_{idx}, found using the DE-algorithm at t = k = 0.9 s and t = k+1 = 0.91 s, when NH=5 and $\mathrm{T_s} = 10$ ms.

4.2.1 Prediction Model

In order to obtain a prediction model, the state space formulation is used:

$$\frac{d}{dt}\underline{x} = \underline{\underline{\mathrm{A}}}\,\underline{x} + \underline{\mathrm{B}}\,\underline{u}$$

$$\mathrm{y} = \underline{\underline{\mathrm{C}}}\,\underline{x}$$

Which can be written in its discrete form as:

$$\underline{x}(k+1) = \underline{\underline{\mathrm{A}}}\,\underline{x}(k) + \underline{\mathrm{B}}\,\underline{u}(k)$$

$$\mathrm{y}(k) = \underline{\underline{\mathrm{C}}}\,\underline{x}(k)$$

In order to use the discrete time state space formulation, the position and speed at the next time step (k+1) are needed to be found. First, the next time step position is found by using the definition of speed, which is change in position over time:

$$\dot{\phi}_2(k) = \frac{\phi_2(k+1) - \phi_2(k)}{\mathrm{T_s}} \implies \phi_2(k+1) = \dot{\phi}_2(k)\mathrm{T_s} + \phi_2(k) \tag{4.20}$$

The speed at the next time step is found by using the definition of acceleration, which is change in speed over time:

$$\ddot{\phi}_2(k) = \frac{\dot{\phi}_2(k+1) - \dot{\phi}_2(k)}{\mathrm{T_s}} \implies \dot{\phi}_2(k+1) = \ddot{\phi}_2(k)\mathrm{T_s} + \dot{\phi}_2(k) \tag{4.21}$$

Here, the acceleration at the current time step, $\ddot{\phi}_2(k)$, is found from the linearized shank acceleration of Equation (3.80) and the equation relating the actuator forces to the knee torque given by Equation (3.7):

$$\ddot{\phi}_2(k) = \mathrm{K_{hyd}} \left.\frac{\mathrm{d}z_1}{\mathrm{d}\psi}\right._0 F_{\mathrm{hyd}}(k) + \mathrm{K_{ext}} M_{\mathrm{ext}}(k) \tag{4.22}$$

With the actuator torque given by $M_{\mathrm{hyd}} = \frac{\mathrm{d}z_1}{\mathrm{d}\psi}_0 F_{\mathrm{hyd}}(k)$. Instead of using the transmission ratio linearization constant, $\frac{\mathrm{d}z_1}{\mathrm{d}\psi}_0$, this parameter is instead chosen to be updated at each time step based on the measurement of the knee angle. This is assessed to be vital since the transmission ratio is very important as to which actuator force will realize the desired knee torque. The draw-back of this approach is that the controller will demand a larger processing time, since the prediction model will need to be re-configured at each time step.

Using Equations (4.20) to (4.22), the system can then be represented in its discrete state space form as [5]:

$$\begin{bmatrix}\phi_2(k+1)\\ \dot{\phi}_2(k+1)\end{bmatrix} = \begin{bmatrix}1 & \mathrm{T_s}\\ 0 & 1\end{bmatrix}\begin{bmatrix}\phi_2(k)\\ \dot{\phi}_2(k)\end{bmatrix} + \begin{bmatrix}0\\ \mathrm{K_{hyd}}\frac{\mathrm{d}z_1}{\mathrm{d}\psi}\mathrm{T_s}\end{bmatrix} F_{\mathrm{hyd}}(k) + \underline{\underline{I}}\begin{bmatrix}0\\ \mathrm{K_{ext}T_s}\end{bmatrix} M_{\mathrm{ext}}(k)$$

This formulation assumes that the pressure dynamics in relation to the sampling time are fast enough that when a valve command is given, the pressure at the next time step will be that of the pressure reservoir. This puts an upper limit on the controller update frequency of 100 Hz (see Appendix A for elaboration on how this value is determined).

Recursive evaluation of the discrete state space formulation yields future states on a prediction horizon defined by NH [5], [23]:

$$\begin{bmatrix}\underline{x}(k+1)\\ \underline{x}(k+2)\\ \vdots\\ \underline{x}(k+\mathrm{NH})\end{bmatrix} = \begin{bmatrix}\underline{\underline{A}}\\ \underline{\underline{A}}^2\\ \vdots\\ \underline{\underline{A}}^{\mathrm{NH}}\end{bmatrix}\underline{x}(k) + \begin{bmatrix}\underline{B} & 0 & \cdots & 0\\ \underline{\underline{A}}\,\underline{B} & \underline{B} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \underline{\underline{A}}^{\mathrm{NH}-1}\underline{B} & \underline{\underline{A}}^{\mathrm{NH}-2}\underline{B} & \cdots & \underline{B}\end{bmatrix}\begin{bmatrix}F_{\mathrm{hyd}}(k)\\ F_{\mathrm{hyd}}(k+1)\\ \vdots\\ F_{\mathrm{hyd}}(k+\mathrm{NH}-1)\end{bmatrix}$$
$$+\begin{bmatrix}\underline{\underline{I}} & 0 & \cdots & 0\\ \underline{\underline{A}} & \underline{\underline{I}} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ \underline{\underline{A}}^{\mathrm{NH}-1} & \underline{\underline{A}}^{\mathrm{NH}-2} & \cdots & \underline{\underline{I}}\end{bmatrix}\begin{bmatrix}M_{\mathrm{ext}}(k)\\ M_{\mathrm{ext}}(k+1)\\ \vdots\\ M_{\mathrm{ext}}(k+\mathrm{NH}-1)\end{bmatrix} \tag{4.23}$$

The future position and speed outputs are given by:

$$\underline{y}_{\phi 2,k} = \begin{bmatrix}\underline{y}(k+1)\\ \underline{y}(k+2)\\ \vdots\\ \underline{y}(k+\mathrm{NH})\end{bmatrix} = \begin{bmatrix}\underline{C}_{\phi 2} & 0 & \cdots & 0\\ 0 & \underline{C}_{\phi 2} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \underline{C}_{\phi 2}\end{bmatrix}\begin{bmatrix}\underline{x}(k+1)\\ \underline{x}(k+2)\\ \vdots\\ \underline{x}(k+\mathrm{NH})\end{bmatrix} \tag{4.24}$$

$$\underline{y}_{\dot{\phi} 2,k} = \begin{bmatrix}\underline{y}(k+1)\\ \underline{y}(k+2)\\ \vdots\\ \underline{y}(k+\mathrm{NH})\end{bmatrix} = \begin{bmatrix}\underline{C}_{\dot{\phi} 2} & 0 & \cdots & 0\\ 0 & \underline{C}_{\dot{\phi} 2} & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \underline{C}_{\dot{\phi} 2}\end{bmatrix}\begin{bmatrix}\underline{x}(k+1)\\ \underline{x}(k+2)\\ \vdots\\ \underline{x}(k+\mathrm{NH})\end{bmatrix} \tag{4.25}$$

Where:

$$\underline{C}_{\phi 2} = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad , \quad \underline{C}_{\dot{\phi} 2} = \begin{bmatrix} 0 & 1 \end{bmatrix} \tag{4.26}$$

The prediction model shown in Equations (4.23) to (4.25), is used to get the future states through optimization on the input torque resulting from the actuator force, F_{hyd}, from which the force level index k_{idx} can be directly derived and used as input to control the system.

4.2.2 Cost Function

The cost function is defined to be:

$$\boxed{S = \sum_{n=1}^{NH} \left(\mathrm{W}_1 (\underline{y}^*_{\phi 2,k} - \underline{y}_{\phi 2,k})^2 + \mathrm{W}_2 (\underline{y}^*_{\dot{\phi} 2,k} - \underline{y}_{\dot{\phi} 2,k})^2 \right) + \mathrm{W}_3 \, E_{\text{sh}}(k_{\text{idx}}, k_{\text{idx,prev}})} \tag{4.27}$$

Here, $\underline{y}^*_{\phi 2,k}$ and $\underline{y}^*_{\dot{\phi} 2,k}$ are vectors that contain the reference values for the shank position and speed at the future NH time steps. W1 and W2 are weighting constants that penalize the tracking error for position and speed respectively at all time steps on the prediction horizon. The errors at each time step on the prediction horizon are summed. In the last term, W3 penalizes the energy losses associated with switching between pressure levels in the four chambers at time step t = k, which corresponds to the time step the controller output is applied to the valve block. The energy losses associated with pressurizing or de-pressurizing a cylinder chamber are calculated with Equation (4.17). In the equation, k_{idx} denotes the new force index candidate and $k_{\text{idx,prev}}$ denotes the force index applied by the controller at the previous time step.

4.2.3 Differential Evolution Algorithm

As the prediction horizon increases, the number of possible valve configurations increases exponentially. Therefore an optimization algorithm is needed in order to find the best input configuration within a time that is realizable in real time on an experimental test setup. For this purpose, the differential evolution algorithm presented in [24] is used. It should be noted that this genetic algorithm does not guarantee finding a global minimum, but has the benefit of being able to work with discrete inputs. Therefore no FSA is needed, since the output given by the algorithm will be the directly implementable valve commands. A flowchart of the DE-algorithm implemented for MPC is illustrated by Figure 4.13 on Page 49. When the optimization loop is run the first time, the vector $\underline{k}_{\text{idx,prev}}$ (of length equal to NH) is defined to be $\underline{k}_{\text{idx,prev}} = [8 \cdots 8]$, but at all other time steps this vector will contain the optimal input values from the previous time steps. This vector is used to construct the population matrix, $\underline{\underline{k}}_{\text{idx,target}}$ of Equation (4.28).

Secondly, the prediction model is constructed. This makes the algorithm more computationally demanding, but the approach was chosen in order to be able to implement the varying transmission ratio instead of a linearized value, since this value is vital for the resulting controller output.

The population matrix $\underline{\underline{k}}_{\text{idx,target}}$ is then defined. This is constituted of NP vectors called the target vectors, $\underline{k}_{\text{idx,target}}$. The population size is chosen to be NP=4, which gives the target matrix four rows. The four target vectors are chosen to span the range of the 16 possible force indices meaning that when the optimization algorithm is initialized, the vectors will be:

$$\underline{\underline{k}}_{\text{idx,target}} = \begin{bmatrix} 16 & \cdots & 16 \\ 12 & \cdots & 12 \\ & \underline{k}_{\text{idx,prev}} & \\ 4 & \cdots & 4 \end{bmatrix} \tag{4.28}$$

$\underline{k}_{\text{idx,prev}}$ is updated at each optimization based on the assumption that the optimal input parameters will be in the vicinity of the previous value, and thereby, the algorithm should converge in fewer iterations. Then, a calculation is performed in order to find the population matrix row index, $k_{\text{idx,min}}$, which achieves the minimum value of the cost function. Afterwards, the while loop is run, where the population is looped through at each iteration. This is done a fixed number of times. Alternatively, it would have been a possibility to implement some convergence condition and use that as a stopping criterion. But for this project, due to the simplicity, a constant for how many iterations the optimization loop is run is defined to be *run_throughs* = 1000. For each iteration of the population loop, the cost function is evaluated with the corresponding target vector at the current iteration in order to get the first cost function value of S_1. This is followed by the calculation of the mutation vector, $\underline{k}_{\text{idx,mutate}}$, which is found according to the following mutation algorithm [5]:

$$\underline{k}_{\text{idx,mutate}} = \underline{k}_{\text{idx,target,1}} + \mathrm{F}(\underline{k}_{\text{idx,target,min}} - \underline{k}_{\text{idx,target,1}}) + \mathrm{F}(\underline{k}_{\text{idx,target,2}} - \underline{k}_{\text{idx,target,3}}) \tag{4.29}$$

This mutation is based on randomly picking three of the target vectors other than the one for the current population loop. These random three vectors are denoted, $\underline{k}_{\text{idx,target,1}}$, $\underline{k}_{\text{idx,target,2}}$ and $\underline{k}_{\text{idx,target,3}}$. The vector $\underline{k}_{\text{idx,target,min}}$ is the target vector for which the minimum cost function value is achieved, which is: $\underline{k}_{\text{idx,target,min}} = \underline{k}_{\text{idx,target}}(k_{\text{idx,min}}, :)$. The mutation constant, F, is chosen to be F = 0.7, which is the same value used in [5] and [23]. The first part of the mutation equation, $\underline{k}_{\text{idx,target,1}} + \mathrm{F}(\underline{k}_{\text{idx,target,min}} - \underline{k}_{\text{idx,target,1}})$, moves the solution towards the optimal value from the previous solution, whereas the final term, $\mathrm{F}(\underline{k}_{\text{idx,target,2}} - \underline{k}_{\text{idx,target,3}})$, prevents premature convergence [5].

Because of the discrete number of input values, the mutated vector is imposed a constraint, which dictates that the force indices are integer values ranging from 1 to 16. This is done with the following Matlab-code, where the index value is truncated by rounding off:

```
for i = 1:NH
   k_idx_mutate(i) = round(k_idx_mutate(i));
   if k_idx_mutate(i) < 1
       k_idx_mutate(i) = 1;
   end
```

```
    if k_idx_mutate(i)> 16
        k_idx_mutate(i) = 16;
    end
end
```

Here, the mutated vector is looped through, rounded off, and then subjected to the upper and lower bound. After this, crossover from the mutation vector, $\underline{k}_{\text{idx,mutate}}$, to the trial vector, $\underline{k}_{\text{idx,trial}}$, is performed. This is done with the following Matlab-code:

```
rand_i = randi(NH,1,1);
for j = 1:NH
    rand_j = rand;
    if rand_j ≤ CR || j == rand_i
        k_idx_trial(i,j) = k_idx_mutate(j);
    elseif rand_j > CR && j ≠ rand_i
        k_idx_trial(i,j) = k_idx_target(i,j);
    end
end
```

This code states that if the random value *rand_j* ∈ [0 ... 1] is below or equal to the crossover probability constant, CR, then the trial vector entry at the current crossover loop iteration will become the entry of the mutation vector. This will also be true if the current crossover loop iteration counter *j* is equal to *rand_i* ∈ [1,2,...,NH]. This is to ensure that at least one value from the mutation vector passes on to the trial vector. If *rand_j* > CR, the entry from the mutation vector "dies", and the target vector entry passes on to the trial vector. The crossover probability constant is defined to be CR = 0.9, also the same value used in [5] and [23].

The cost function is evaluated with the trial vector, $\underline{k}_{\text{idx,trial}}$, in order to get the second cost function value, S_2. When the population loop is completed, the four rows of S_2 are compared to the four rows of S_1 in order to see if any of the trial vectors result in a lower value of the cost function. If $S_2 < S_1$ the trial vector is passed on to the the target vector to be used for next population loop. Then, the row index, $k_{\text{idx,min}}$, corresponding to target vector giving the minimum value of the cost function is found, and the *iteration_count* is incremented by one. This is repeated until the while loop meets its stopping criterion of *iteration_count* = *run_throughs*.

The optimization is finalized by defining $\underline{k}_{\text{idx,prev}}$ to be the target vector giving the minimum cost function value and storing this for the next time step. And finally, the valve command vector is found from the input matrix of Equation (3.10) and the first entry of the minimum cost target vector: $\underline{u} = \underline{\underline{\text{u}}}(\underline{k}_{\text{idx,target,min}}(1), :)$. This is the controller output, which is commanded to the system's valve block.

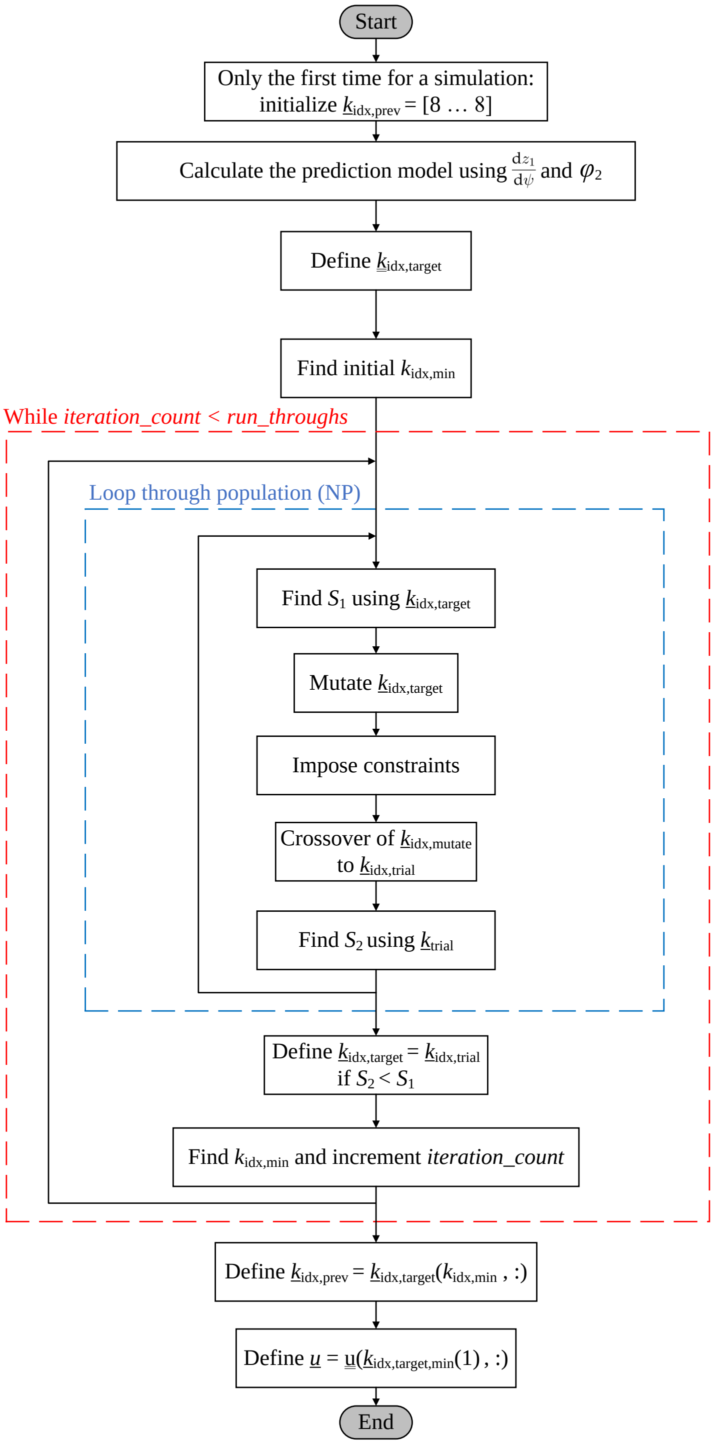


Figure 4.13: Flow chart of MPC implementation using the DE-algorithm.

As an example of how the DE-algorithm works, Figure 4.14 shows an optimization run under the conditions during the gait cycle at t=0.9 s with a sample time of $T_s = 10$ ms and NH=5. The left figure shows how the value of the cost function decreases as the iteration count increases. This plot has been zoomed in on the x-axis, since the largest decrease in cost function value happens in the first 10 iterations, but in reality 1000 iterations are run.

The middle and the right figures show blue crosses for the predicted outputs for shank position and shank speed found through the optimization. These are compared to the red curves for the HuMoD reference data, and as it can be seen, there is close relation even though the predicted values are not spot on the curves. This is due to the integrality constraint on the input values.

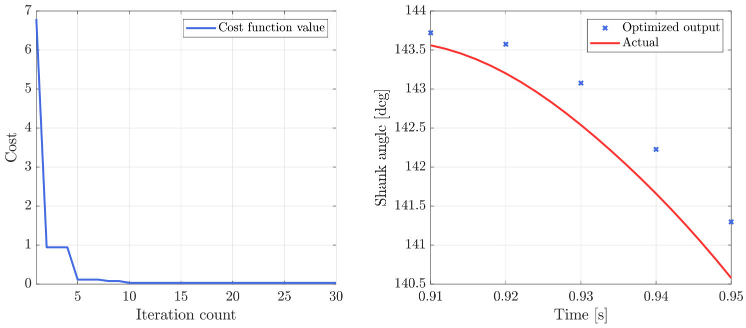


Figure 4.14: Example of DE-algorithm optimizing at t=0.9 s (@ $T_s = 10$ ms, W1=10, W2=1, W3=0 and NH=5). Left: the cost function value as the iteration count increases. Middle: the predicted and actual shank position. Right: the predicted and actual shank speed.

4.2.4 MPC Tuning

For tuning the MPC, the ground reaction force is omitted in order to neglect its influence to make the results more easily comparable.

Initially, the cost function (of Equation (4.27)) is used with W3 = 0 meaning that only the tracking terms are included. Thereby, this cost function solely penalizes the tracking errors. Figure 4.15 shows the rms errors for the shank's position and speed during a gait cycle at different prediction horizons. When NH > 1, the measured external torque M_{ext} is only included in the prediction model of Equation (4.23) for the first time step, since the external torque at the future time steps cannot be known with the linearized model. It can be seen from the results that increasing the prediction horizon does not cause improvement of the position and speed tracking. Even though the prediction horizon of NH=3 gives the smallest position rms error, the prediction horizon of NH=2 is chosen, since it allows for brute forcing the optimization algorithm. Brute forcing the algorithm means that the cost function is evaluated for every possible input configuration, and thereby the global minimum is found. An analysis comparing the processing time with the DE-algorithm and the brute force algorithm will be presented in Section 4.2.5. From Figure 4.15, it can also be seen that the speed rms error is almost the same for NH=2 and NH=3, suggesting that the position error occurs due to a constant offset. Using a brute force optimization algorithm would also be possible for NH=1, but this case has the draw-back of not being able to control the position directly, since the discrete nature of the prediction model (Equation (4.23)) means that the predicted position can not change before the second time step. This means that when NH=1, the MPC only tracks the speed, which in turn can cause drifting of the position error when the system is subjected to the influence of external disturbances.

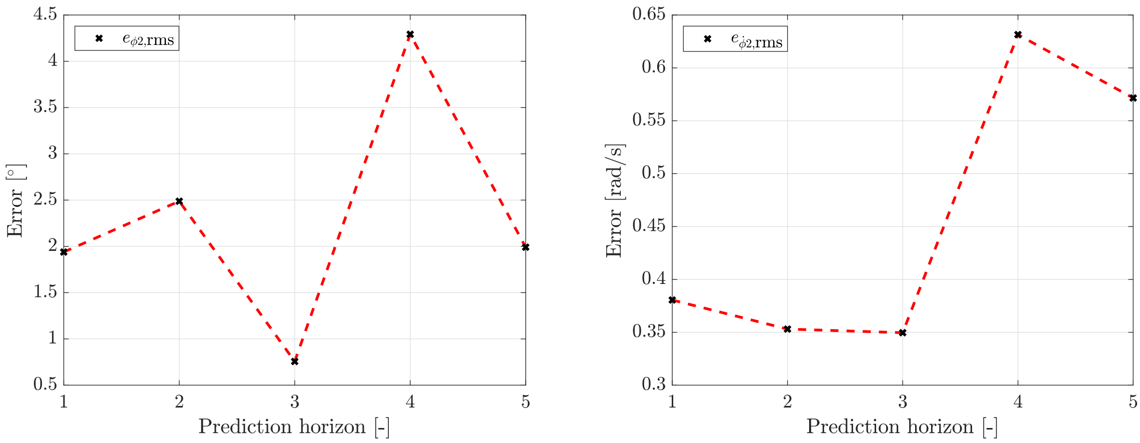


Figure 4.15: Tracking precision at varying prediction horizons (@: $T_s = 10$ ms, W1 = W2 = 1, W3 = 0). Left: position rms error. Right: speed rms error.

Using the prediction horizon defined to be NH=2, Figure 4.16 shows an examination of the influence of changing the valve update time along with the prediction model's prediction time. Again, no ground reaction force is implemented in these simulations. The figure shows that increasing the sample time decreases the tracking precision severely. It can also be seen that the energy consumption decreases as the sample time increases, which corresponds with the fact that fewer switches of the valves occur. The sample time of $T_s = 5$ ms is also tested, but in order to choose a conservative value ensuring that the pressure levels can settle in the given time period (according to Appendix A), the sample time of $T_s = 10$ ms is chosen for future analysis.

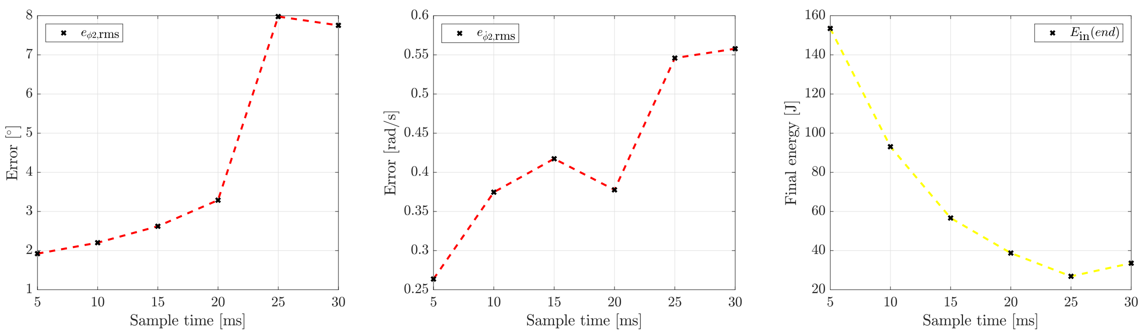


Figure 4.16: Tracking precision and energy consumption at varying the sample times (@: NH = 2, W1 = W2 = 1, W3 = 0). Left: position rms error. Middle: speed rms error. Right: energy consumption.

In Figure 4.17 the tracking precision and energy consumption is plotted as the ratio between the two weighting factors W1 and W2 is varied. The results show no major impact on the tracking precision or energy consumption when the ratio is changed when looking at the absolute values. It is although worth pointing out that as the weighting ratio is low, the smallest error is achieved for the speed tracking, and oppositely, when the weighting ratio is high, the position error is lowered. This corresponds with the definition of the cost function in Equation (4.27), where a low ratio should penalize the speed error the most, and a high ratio should penalize the position error the most. In conclusion, even though no major difference can be noted on the influence of the weighting ratios, a higher ratio gives better position tracking along with lower energy consumption. Therefore, the weighting parameters are chosen to be W1 = 10 and W2 = 1.

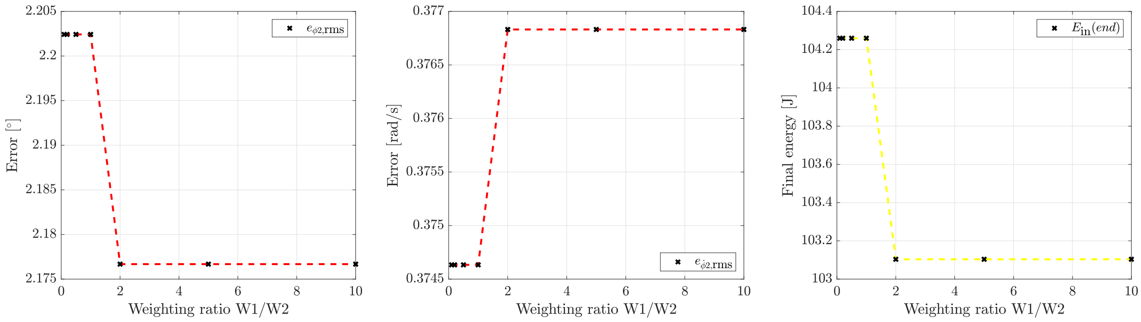


Figure 4.17: Tracking precision and energy consumption for varying weighting ratios (@: NH = 2, T_s = 10 ms, W3 = 0). Left: position rms error. Middle: speed rms error. Right: energy consumption.

Finally, the energy weighting constant, W3, is examined in order to find a value for the final tuning parameter W3, which is the weighting constant that determines the influence of the energy losses associated with switching the valves.

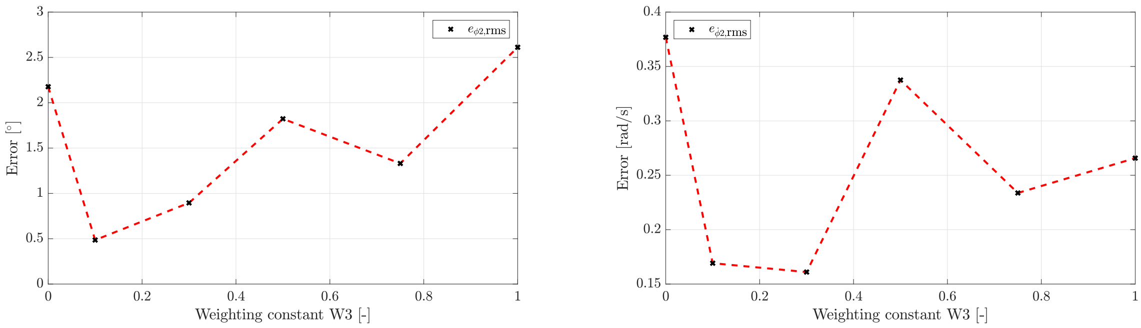


Figure 4.18: Tracking precision and energy consumption for varying weighting constants W3 (@: NH = 2, T_s = 10 ms, W1 = 10, W2 = 1). Left: position rms error. Right: speed rms error.

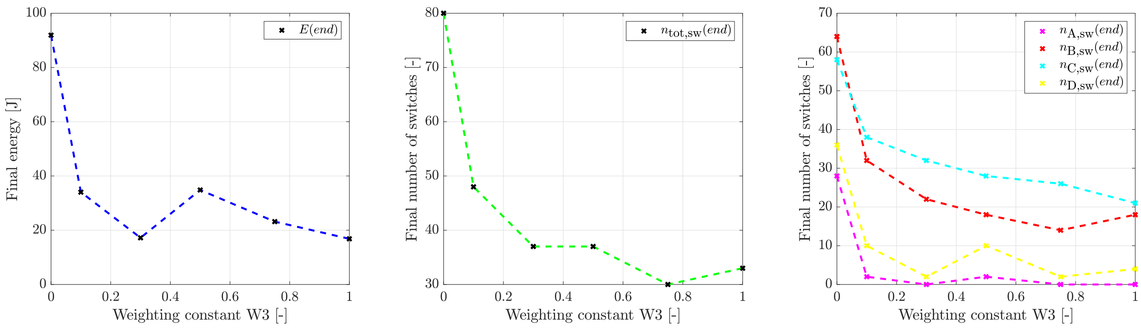


Figure 4.19: Energy consumption for varying weighting constants W3 (@: NH = 2, T_s = 10 ms, W1 = 10, W2 = 1). Left: energy consumption. Middle: number of force level switches. Right: number of valve switches for each chamber.

For varying values of W3, Figure 4.18 shows the tracking precision and Figure 4.19 shows the energy consumption. It can be seen that the increase of W3 decreases the tracking precision while lowering the energy consumption substantially. The value of W3 = 0.75

is chosen, since the lowest amount of force level switches occur at this value, while still keeping a relatively low tracking error.

4.2.5 Brute Force Optimization

As mentioned previously, the DE-algorithm is necessary if the control is to be implementable in real time when the prediction horizon is large. But since a prediction horizon of NH = 2 is chosen, there are only $16^2 = 256$ input combinations to the optimization problem, meaning that using a brute-force algorithm, where every possibility is evaluated, might be a viable alternative. Unlike the DE-algorithm, this approach will also guarantee finding a global minimum. In order to find if the brute force method is viable, a comparison of the processing time between the two optimization approaches is conducted with simulations where no ground reaction force is included. Firstly, the influence of changing the run-through number of the DE-algorithm is shown in Figure 4.20, where the rms position error is shown to the left, and the processing time of the MPC-algorithm is shown to the right. As the number of run-throughs is increased, the tracking precision is improved, and the processing time increases. The lowest procsssing time achieved is $\approx$ 1 ms.

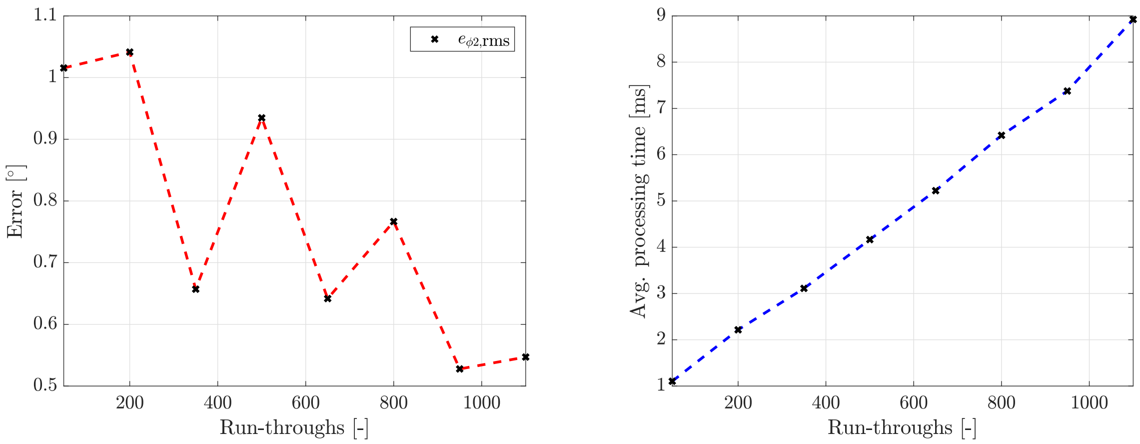


Figure 4.20: Tracking precision and processing time as the number of run-throughs for the DE-algorithm is increased (@: NH = 2, T_s = 10 ms, W1 = 10, W2 = 1, W3 = 0.75). Left: position rms error. Right: MPC-algorithm processing time.

Figure 4.21 shows the results for the brute force method, where the same simulation is run eight times. To the left, it can be seen that the rms error remains the same between simulations, as it should. To the right it can be seen that the calculated processing time differs from run to run, suggesting some variability in the way the processing time is calculated in Simulink. But from run 2-8, the calculated value settles at $\approx$ 0.35 ms. Comparing the results to the DE-algorithm results of Figure 4.20, the processing time with the brute force method is in every case equal to or better than the best case scenario for the DE-algorithm, meaning that relatively speaking, this approach is computationally viable in the case that the DE-algorithm is. And furthermore, the brute force method also adds the benefit of guaranteeing finding the global minimum, which is assumed to be the reason that the brute force algorithm achieves a lower position rms error than the best case gotten with the DE-algorithm.

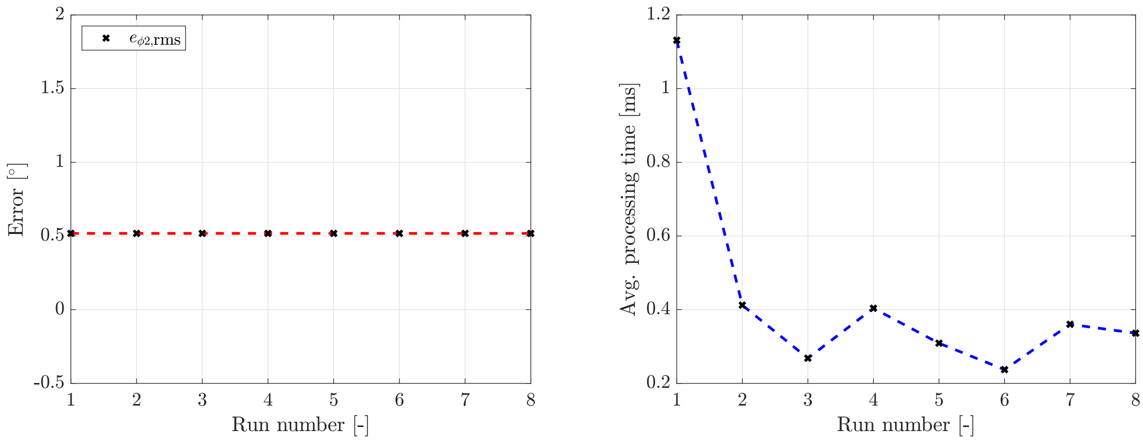


Figure 4.21: Multiple runs with the brute force method (@: NH = 2, T_s = 10 ms, W1 = 10, W2 = 1, W3 = 0.75).

The chosen tuning parameters for the MPC are given in Table 4.3. This concludes the controller development, for which simulation results will be shown in the next chapter.

MPC sample time	Cost function weighting values			DE-parameters			
T_s = 10 ms	W1 = 10	W2 = 1	W3 = 0.75	NP = 4	NH = 2	F = 0.7	CR = 0.9

Table 4.3: Overview of chosen MPC parameters.

5 Simulation Results

Chapter Summary

This chapter presents simulation results obtained with the reactive controller and the MPC presented in Chapter 4. Firstly, the results for the reactive controller structure are shown. Secondly, the results for the MPC are shown. And finally, a comparison between the two controllers is conducted.

For all the simulative evaluations, noise is implemented on the position and speed signals measured and used in the controllers. The noise is not based on any empirical data, but is assumed. The noise is chosen to have a frequency of 10 kHz and a variance of 0.0001. The same noise configurations are used for implementation on both the position and speed signal. An example of the noise is shown in Figure 5.1. Implementing this on the position signal results in peak values for the variation in the measured signal of $\approx \pm 1.2°$ and $\approx \pm 0.02$ rad/s.

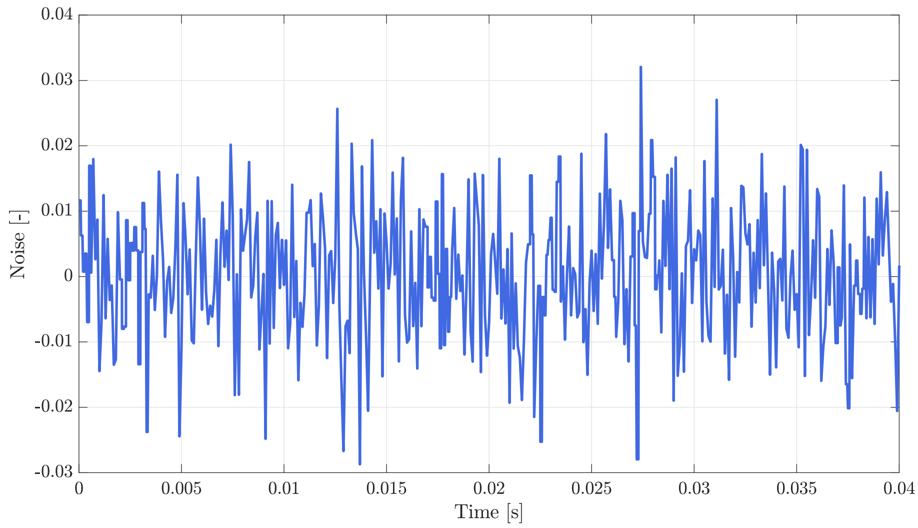


Figure 5.1: Noise implemented on position and speed measurements in the simulations.

5.1 Reactive Controller Results

The system is simulated with the the reactive controller design presented in Section 4.1. This constitutes a PD-controller with torque compensation and torque feed-forward along with the FSA2.

In Figure 5.2, the tracking results are shown. In the top of the figure is the position (left) and position error (right) plotted. In the bottom is the speed (left) and speed error (right) plotted. Decent speed and position tracking is achieved, with a maximum absolute position error of 2.6 °. Noise is only plotted on the error signals, where it is very prevalent in the position error.

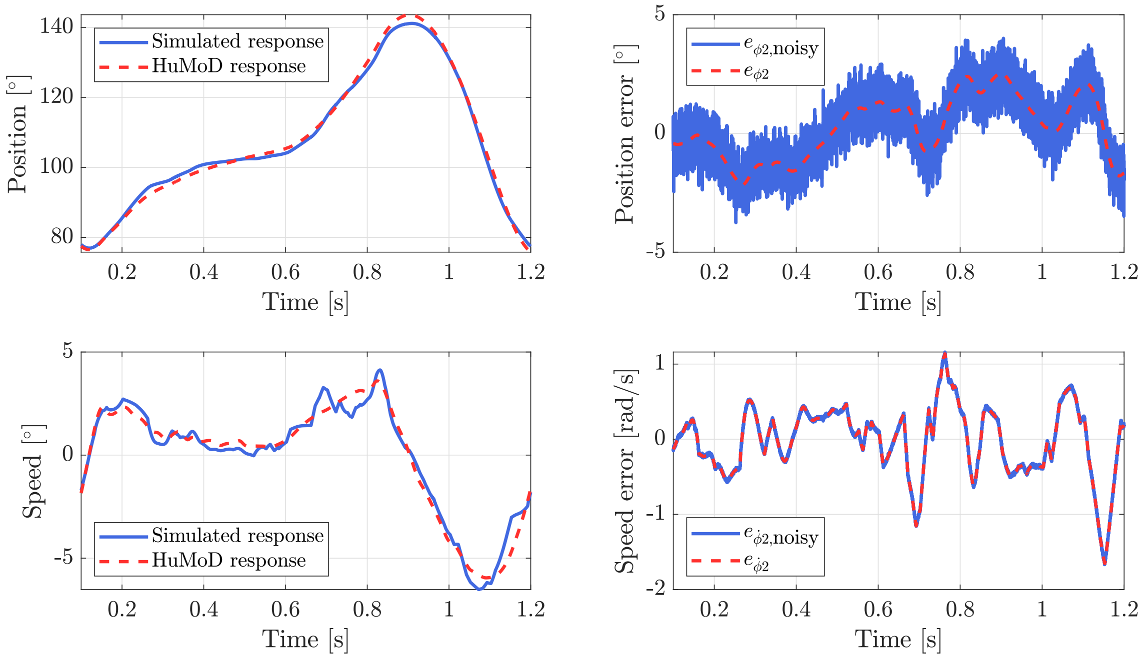


Figure 5.2: Top: position and position error. Bottom: speed and speed error.

The input energy, E_{in} , is plotted to the left in Figure 5.3 and as it can be seen, the maximum energy consumed is when the peak knee torque is delivered during the stance phase, which is 25 J. Energy recuperation can also be noted, as flow goes back to the pressure source, and the energy is lowered. Negative supply energy can even be noticed from t = 0.81 s to t = 1.1 s. In the middle of Figure 5.3, the number of force level switches, $n_{\text{F,sw}}$, is seen, and to the right of the figure, the switches in pressure level of the respective chambers are seen. It can be seen that at all times, the two rod side chambers (B-chamber and C-chamber) see the most switches during a gait cycle.

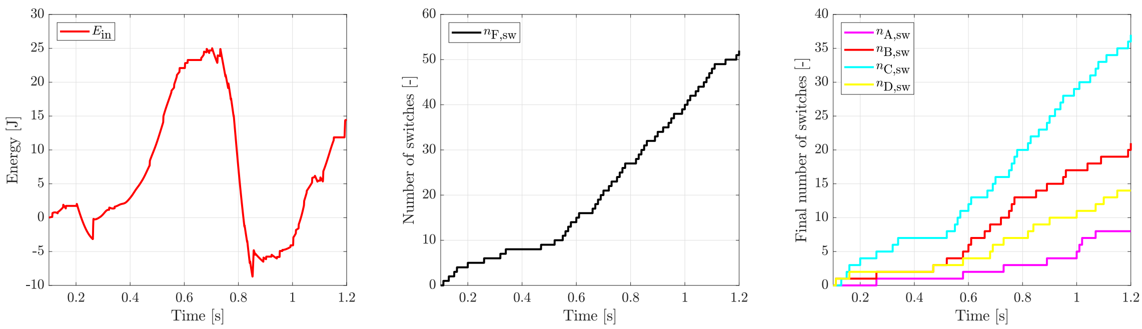


Figure 5.3: Left: input energy. Middle: number of force level switches. Right: individual number of chamber pressure switches.

Corresponding to the individual chamber pressure switches seen to the right in Figure 5.3, the chamber pressures are plotted in Figure 5.4.

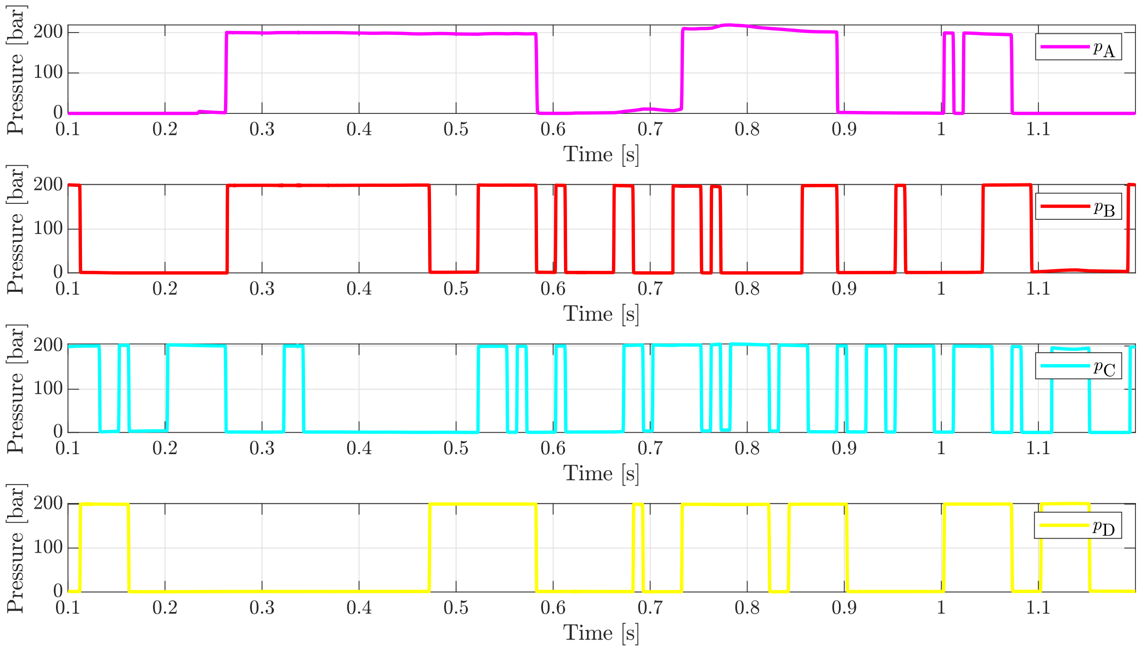


Figure 5.4: Chamber pressures.

In the top of Figure 5.5, the actuator torque reference, M^*_{hyd}, is shown with the red dashed line. It is clear that this signal is distorted by the signal measurement noise, which is amplified by the proportional and derivative gain of the PD-controller. The noisy reference signal does not translate directly to a noisy actuator torque, since the FSA provides filtering. This is seen by the blue line, which shows the actuator torque, M_{hyd}. It can be noticed that a downward ramping of M_{hyd} happens from t = 0.78 s to 0.82 s, which corresponds to the decrease of the transmission ratio that happens here, as it goes to its lowest value. Spikes in M_{hyd} can be noticed at different times, for instance at t = 0.26 s. This is due to the pressure dynamics, where the pressure takes longer time to settle in one chamber than the others. Furthermore, the green dashed line shows the actuator torque needed for an ideal gait cycle based on the HuMoD data. In the bottom of Figure 5.5, the force level index, k_{idx}, is shown, and this translates directly to the actuator torque.

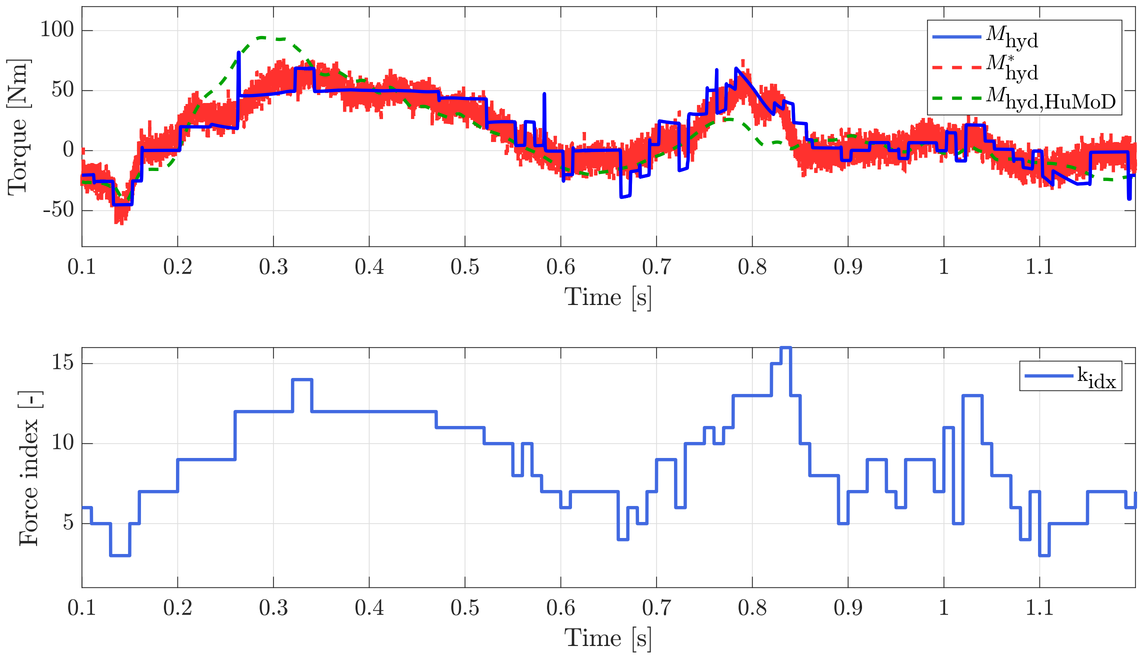


Figure 5.5: Top: actuator torque (blue), actuator torque reference (red) and ideal actuator torque based on HuMoD data (green). Bottom: force level index.

Figure 5.6 shows the three signals constituting the actuator torque reference. Here it can be seen that the torque compensation term, $M_{\mathrm{hyd,Mext}}$, and the torque feed-forward term, $M_{\mathrm{hyd,FF}}$ does most of the work, and the PD-controller reacts on the remaining position error with the smallest signal. The PD-controller's output, $M_{\mathrm{hyd,PD}}$, is the signal which is sensitive to noise, and there is indeed an amplification as can be seen in the top of the figure. A moving average of this signal is plotted to make it more easily to see the PD-controller's contribution.

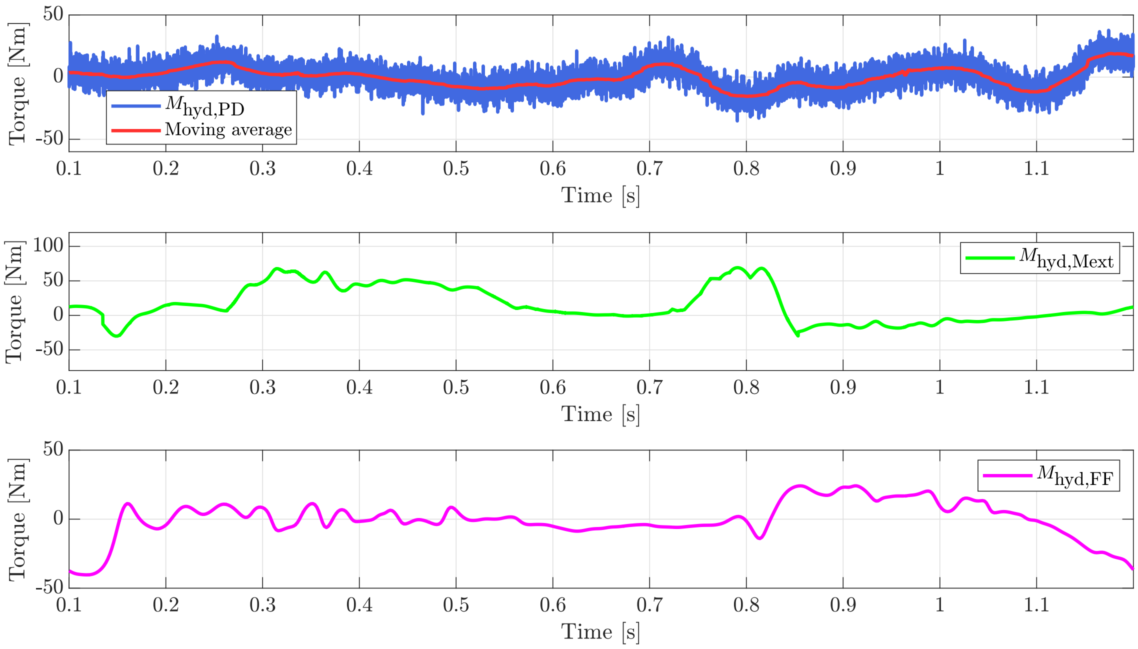


Figure 5.6: Top: PD-controller output. Middle: External torque compensation. Bottom: Torque feed-forward.

The ground reaction torque, M_{GRF}, is plotted in Figure 5.7. It can be seen that the correspondence between M_{GRF} of the simulation and $M_{\text{GRF,HuMoD}}$ of the ideal gait cycle is not very good. This is an indication that the knee angle is tracked poorly, since close tracking of the knee angle is needed in order to obtain a close relationship between M_{GRF} and $M_{\text{GRF,HuMoD}}$. The general tendencies of the ideal gait cycle are present, but especially at the final part of the stance phase from t=0.7s to t=0.85s, a large spike in M_{GRF} can be noted. This time range is where the system sees the largest position error as seen in Figure 5.2. Lastly, it can be seen that there is a close resemblance between M_{GRF} in Figure 5.7 and M_{hyd} in the top part of Figure 5.5, which shows that the ground reaction torque is the primary external influence that the actuator has to overcome.

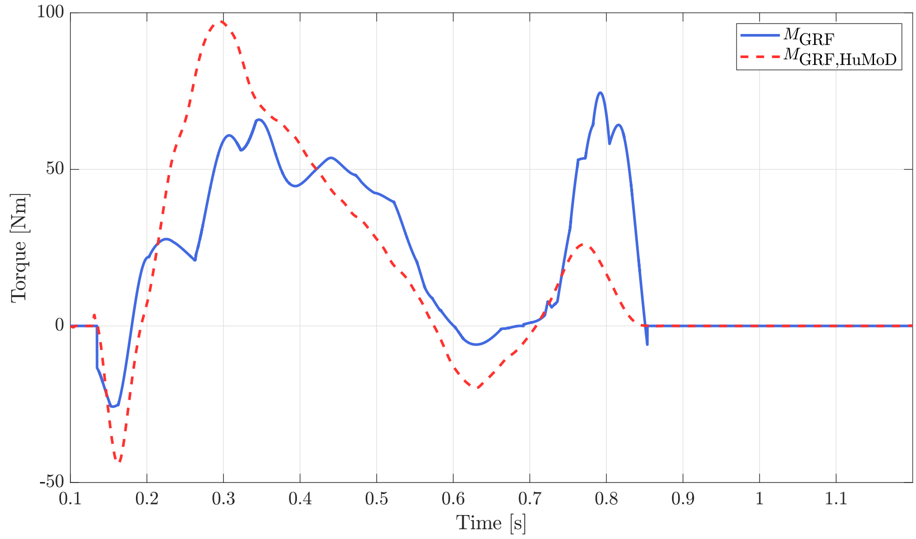


Figure 5.7: Ideal and simulated ground reaction torque, M_{GRF}.

5.2 MPC Results

The MPC used to get the simulation results in this section, is presented in Section 4.2. The tracking results of Figure 5.8 show that good tracking is achieved, with a maximum absolute position error of 1.4 °, which is smaller than for the reactive controller.

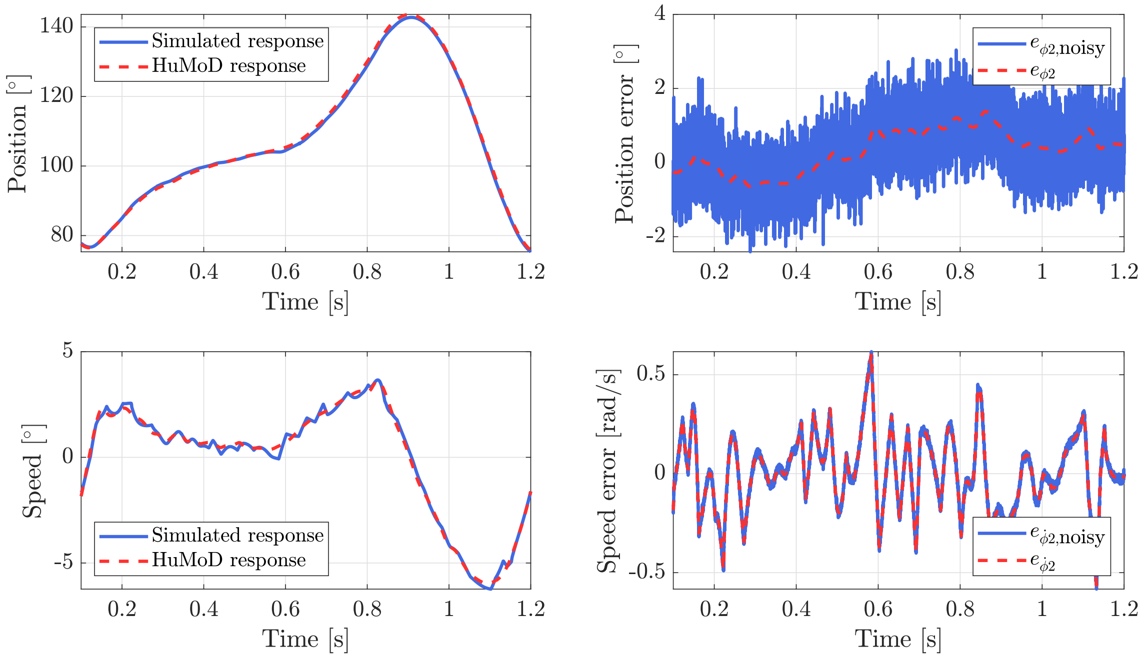


Figure 5.8: Top: position and position error. Bottom: speed and speed error.

Figure 5.9 and Figure 5.10 shows the input energy, number of force level and pressure level switches and chamber pressures. Clearly, there are fewer switches of the pressure in the A-chamber than was the case for the reactive controller, but the energy consumption seems to be very similar to the reactive controller, with a peak energy consumption of

25 J. Although no negative supply energy is seen during the gait cycle, the final energy consumption is still a bit lower than for the reactive controller. Interestingly, very few switches of the A and D chamber pressures occur, which suggests that the MPC penalizes changing these pressures in the high-volume chambers more than the reactive controller does.

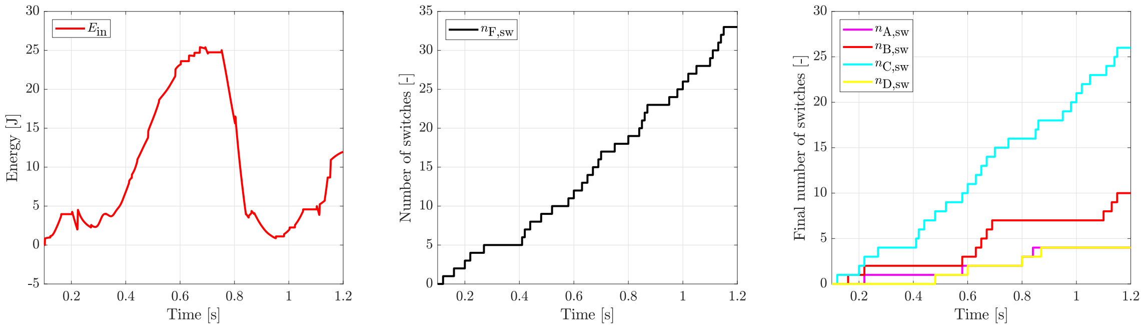


Figure 5.9: Left: input energy. Middle: number of force level switches. Right: individual number of chamber pressure switches.

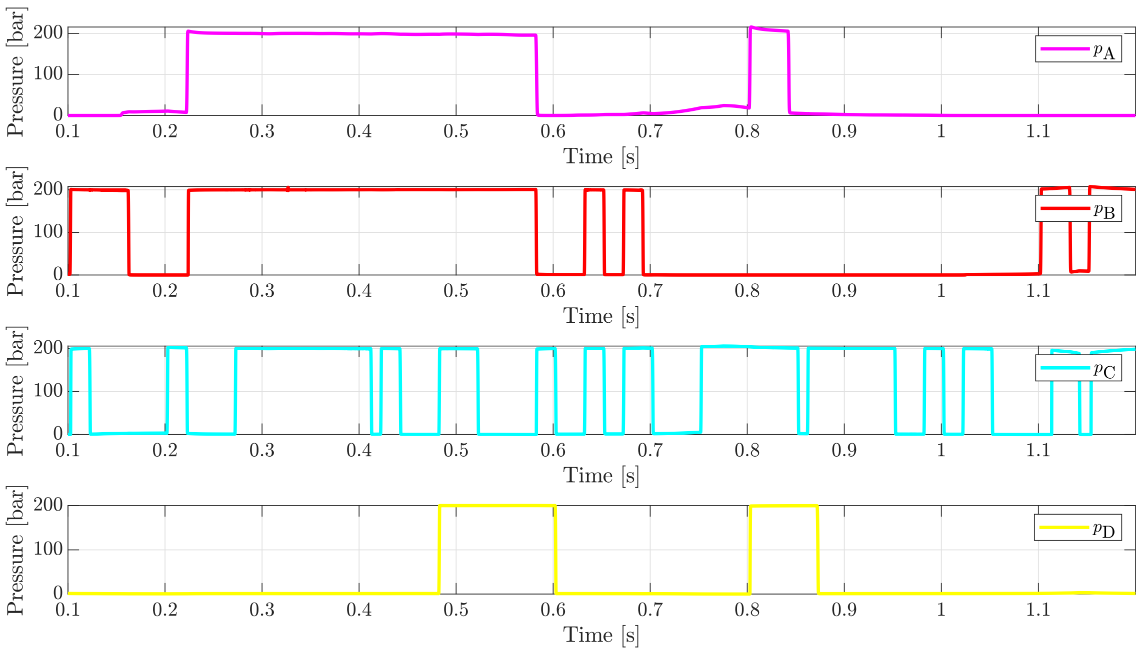


Figure 5.10: Chamber pressures.

Figure 5.11 shows the actuator torque reference, the simulated torque and the ideal HuMoD torque. The noise is not at all present on the reference signal, which suggests that the MPC is less susceptible to influence from noise than the PD-controller is. This is backed by an analysis presented in Chapter 6 showing that the MPC is able to track the reference trajectory better than the reactive controller when a larger noise is added on the measured position and speed signals. A sloped actuator torque, M_{hyd}, can be noticed from t = 0.8 s to t = 0.84 s, which is because the torque transmission ratio changes here at this exact instance to its lowest value.

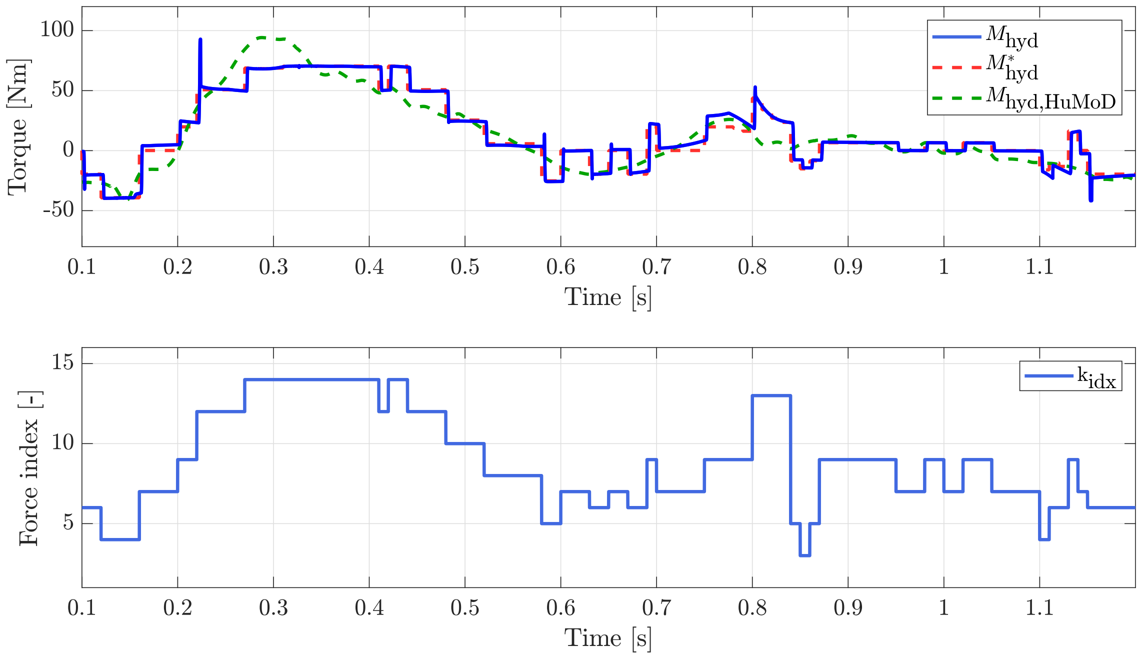


Figure 5.11: Top: actuator torque (blue), actuator torque reference (red) and ideal actuator torque based on HuMoD data (green). Bottom: force level index.

Finally, the ground reaction torque is plotted in Figure 5.12 where it is clear that the M_{GRF} more closely resembles the ideal HuMoD curve than was the case for the reactive controller. This is an indication that better tracking is achieved with the MPC.

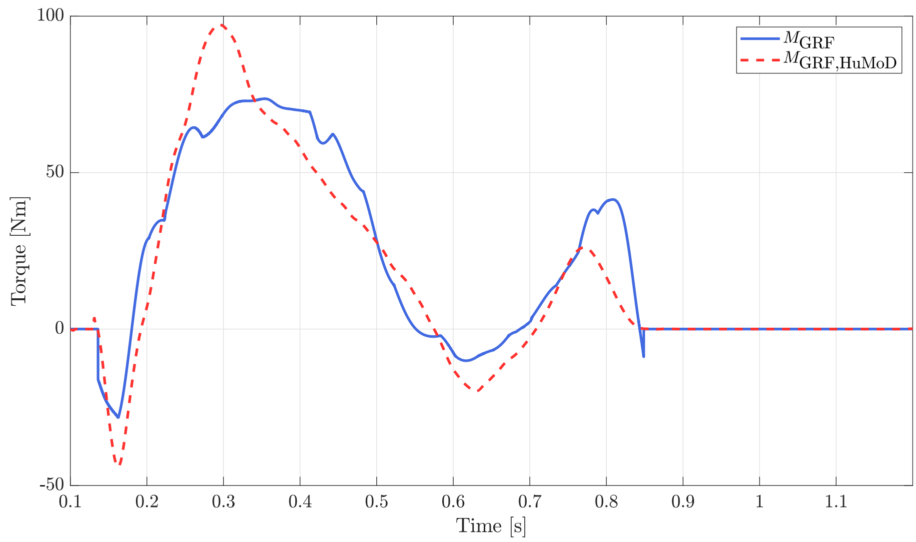


Figure 5.12: Ideal and simulated ground reaction torque, M_{GRF}.

5.3 Comparison of Controller Performances

In Table 5.1, a comparison of the controller performance of the reactive structure with the PD-controller and the MPC is made. The position rms error, $e_{\phi 2,\mathrm{rms}}$, shows that the smallest value is achieved by the MPC. Looking at the speed rms error, $e_{\dot{\phi} 2,\mathrm{rms}}$, it can be seen that the smallest value is also achieved by the MPC. The benchmarking parameter $e_{M\text{GRF,rms}}$ is introduced, and this denotes the rms error between the HuMoD ground reaction torque, $M_{\text{GRF,HuMoD}}$, and the simulated ground reaction torque, M_{GRF}. Based on the notion that a low value for $e_{M\text{GRF,rms}}$ means that good tracking is achieved, the MPC performs better tracking than the PD-controller. Furthermore, as can be seen in Figure 3.11, the model for the ground reaction torque, M_{GRF}, is under ideal conditions shown to not correspond exactly to the ground reaction torque obtained based on the HuMoD data. The rms error over the gait cycle in the ideal case is $e_{M\text{GRF,rms}} = 9.99$ N. This value can serve as a benchmark of how good the tracking is. Comparing this to the values obtained in Table 5.1, the MPC achieves the value closest, suggesting that the MPC approach provides tracking that more closely mimicks the human gait cycle than the PD-controller. The energy consumption at the final time of the gait cycle, E_{in}(end), shows that the least amount of energy is consumed when the MPC is employed. The energy consumption of the PD-controller is slightly larger than for the MPC. Lastly, it can also be seen that almost half as many force level switches occur for the MPC than for the PD-controller. On all parameters, the best performance is achieved by the MPC.

Parameter	Unit	PD	MPC
$e_{\phi 2,\text{rms}}$	[°]	1.32	0.63
$e_{\dot{\phi} 2,\text{rms}}$	[rad/s]	0.46	0.19
$e_{M\text{GRF,rms}}$	[N]	18.89	11.76
E_{in}(end)	[J]	14.46	11.97
$n_{\text{F,sw}}$(end)	[-]	52	33

Table 5.1: Comparison of the controller performance with PD-control and MPC.

This concludes the simulation results. As mentioned in the scope of the project of Chapter 2, the prototype of the system is at the time of writing still being built by the staff at Linz University and Linz Center of Mechatronics and will unfortunately not be ready in time for conducting an experimental controller evaluation of the controllers designed in this project. Therefore this concludes the presentation of the theoretical and simulative work done in this project, and the remaining chapters will focus on discussing the methods and results as well as drawing conclusions based on the results.

6 Discussion

6.1 Modelling Assumptions

The friction of the model is assumed to be purely viscous, and a value for the friction coefficient is assumed based on the knee torque and knee angular velocity during the gait cycle. This assumption most likely means that the numerical results obtained in the project are incoherent with the physical reality. Another modelling assumption lies within the model for the ground reaction torque, M_{GRF}. This is not fully state dependent, since the length of the foot and shank are not defined to be constant. In the top of Figure 6.1 it can by the blue curve be seen how M_{GRF} looks when using the constant length of the foot and shank in the GRFM. It is clear that not implementing the lengths of foot and shank to be variable according to the HuMoD data renders the model worthless. The reason for this is, as discussed in Section 3.2.4 on Page 31, that there will be small displacements of the reflective markers used for capturing the HuMoD data. This is seen clearly in the middle and bottom of Figure 6.1, where the length of the foot and shank are plotted based on calculations using the measured location of the knee, ankle and toe from the HuMoD database. Therefore, implementing the variable lengths when simulating the system is necessary to get a working GRFM, but this also means that M_{GRF} is not fully state dependent. But under the assumption that the position reference is closely tracked, the model should still be representative of the physical system.

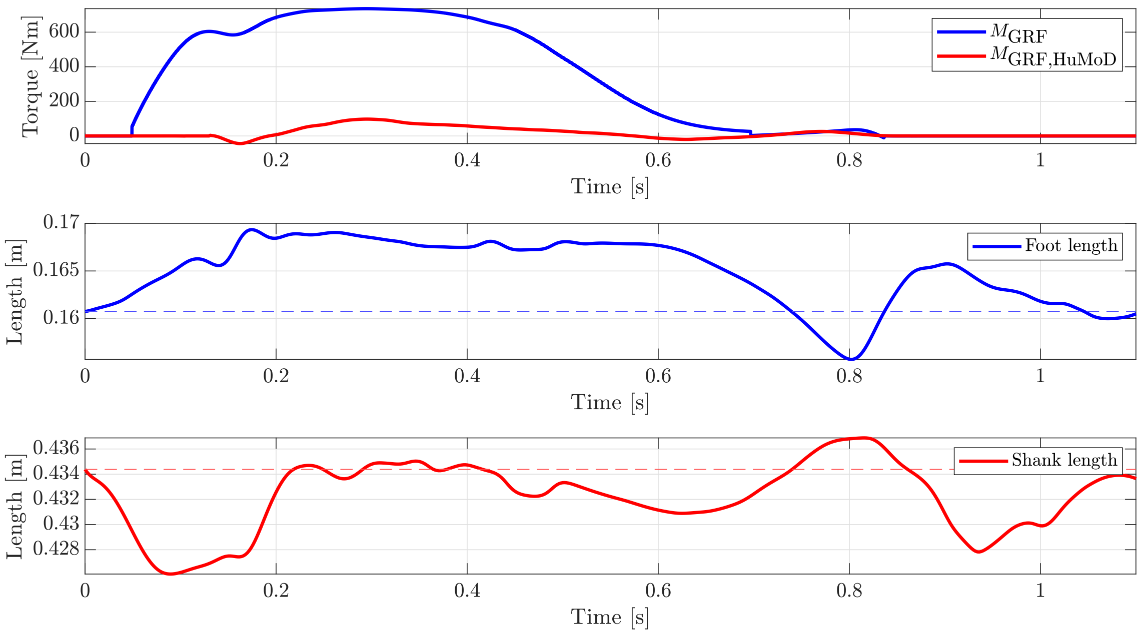


Figure 6.1: Top: Ground reaction torque calculated with constant l_1 and l_2. Middle: Dashed line is the constant foot length and the curve is the variable foot length. Bottom: Dashed line is the constant shank length and the curve is the variable shank length

Even though these model assumptions are made, it is expected that the general dynamics of the simulated system will correspond to the physical system, but only through experimental validation of the model and controllers can this be known for sure. When the model has been validated, the controllers will most likely have to be re-tuned, since the dynamics are expected to change significantly as a result of the friction and other possible unmodelled dynamics. But even if this is the case, the results obtained in this project are still valid for conducting a comparative study, since the two controllers are subjected to the same conditions.

6.2 Implementation of Torque Compensation

For the simulations shown in Chapter 5, access to measurements of the knee torque is assumed, even though this is not available in physical exoskeleton system currently. The importance of the torque compensation is therefore analysed, by running new simulations, where the torque compensation is neglected. This is shown by the results in Table 6.1, where the benchmarking results for the two controllers are shown with and without torque compensation included. The left side table is the results from Table 5.1, repeated here for ease of perusal. The table to the right is the results for the simulations where no torque compensation is included.

Parameter	Unit	PD	MPC
$e_{\phi 2,\text{rms}}$	[°]	1.32	0.63
$e_{\dot{\phi} 2,\text{rms}}$	[rad/s]	0.46	0.19
$e_{M\text{GRF},\text{rms}}$	[N]	18.89	11.76
$E_{\text{in}}(\text{end})$	[J]	14.46	11.97
$n_{\text{F,sw}}(\text{end})$	[-]	52	33

Parameter	Unit	PD	MPC
$e_{\phi 2,\text{rms}}$	[°]	3.25	3.21
$e_{\dot{\phi} 2,\text{rms}}$	[rad/s]	0.75	0.34
$e_{M\text{GRF},\text{rms}}$	[N]	34.95	35.25
$E_{\text{in}}(\text{end})$	[J]	45.34	21.94
$n_{\text{F,sw}}(\text{end})$	[-]	57	39

Table 6.1: Left: results with torque compensation. Right: results without torque compensation.

Comparing the two tables, it can be seen that both the rms tracking error for position and speed are increased. The increase in the values for $e_{M\text{GRF},\text{rms}}$ is a clear indication that the worse tracking means that the ground reaction torque does not resemble that from the ideal gait cycle. In general, the PD-controller performs worse when looking at the parameters in all cases, except for $e_{M\text{GRF},\text{rms}}$ where it is slightly better. It can be concluded, that the torque compensation is very important in order to achieve good tracking of the motion trajectory, and the possibilities for implementing the needed measurement equipment on the exoskeleton should be examined.

6.3 Noise Sensitivity

It is suspected that the MPC approach draws the benefit of being less sensitive towards noise disturbances than the PD-controller. To analyse if this is the case, more noise is implemented on the signal and new simulations are run. This is done by changing the variance to 0.001 on the position signal noise, corresponding to peak values for the variation in the measured signal of $\approx \pm 4°$. Extra noise is added on the speed signal (used only by the MPC), with a variance of 0.1 resulting in peak variations in the measured signal of $\approx \pm 0.7$ rad/s. Table 6.2 shows the benchmarking results with the smaller signal noise to the left (repeated for easy comparison) and to the right is shown the results with the larger noise implemented. It can immediately be seen that the performance of both the PD-controller and the MPC is degraded when extra noise is added.

Parameter	Unit	PD	MPC	Parameter	Unit	PD	MPC
$e_{\phi 2,\mathrm{rms}}$	[°]	1.32	0.63	$e_{\phi 2,\mathrm{rms}}$	[°]	2.01	1.57
$e_{\dot{\phi} 2,\mathrm{rms}}$	[rad/s]	0.46	0.19	$e_{\dot{\phi} 2,\mathrm{rms}}$	[rad/s]	0.83	0.32
$e_{M\mathrm{GRF,rms}}$	[N]	18.89	11.76	$e_{M\mathrm{GRF,rms}}$	[N]	39.14	29.94
$E_{\mathrm{in}}(\mathrm{end})$	[J]	14.46	11.97	$E_{\mathrm{in}}(\mathrm{end})$	[J]	44.22	37.92
$n_{\mathrm{F,sw}}(\mathrm{end})$	[-]	52	33	$n_{\mathrm{F,sw}}(\mathrm{end})$	[-]	81	64

Table 6.2: Left: results with small noise (position: $\approx \pm 1.2°$, speed: $\approx \pm 0.02$ rad/s). Right: results with large noise (position: $\approx \pm 4°$, speed: $\approx \pm 0.7$ rad/s).

When comparing the two results of Table 6.2, the most important indication of the quality of the tracking is $e_{M\mathrm{GRF,rms}}$. This is increased the most for the PD-controller. The position rms error increases a lot for both approaches, but interestingly, the rms speed error does not change very much for the MPC. This suggests that there is a constant offset in the position error, whilst the speed reference is tracked closely. These results indicate that the extra noise degrades the performance of the PD-controller more than for the MPC, which suggests that the MPC is less sensitive towards noise. It must be noted that these noise levels are not based on any empirical data, which means that depending on the level of noise in the laboratory test setup, the PD-controller will most likely have to be re-tuned, more or less aggressively.

6.4 Processing Time of the MPC-algorithm

The MPC-approach implemented in [14] had a processing time of $\approx$ 1-2 seconds at each time step, which was conducted with a genetic algorithm on the non-linear system of equations. This processing time is not realizable in real-time, since the gait cycle itself takes only 1.1 seconds. This serves as an indication of the $\approx$ 0.3 ms average processing time of the MPC with the brute force algorithm presented in Figure 4.21 on Page 54 being vastly better. It must be noted that these two processing times are based on calculations performed by two different computers, which makes the results not directly comparable. But still, with the processing time of the new controller being $\approx$ 3000 times faster, the difference is so large that the conclusion can be drawn that the MPC designed in this project is significantly faster. The question is now whether or not the algorithm is fast enough for real time implementation, which depends on the type of microprocessor used for controlling the prototype. Examination of this will be left for future work when the protoype has been built.

7 Conclusion

In order to fulfill the study objective of designing and benchmarking a reactive and model predictive control strategy against each other, a model of the system was firstly implemented. This consists of the hydraulic and mechanical model of the knee exoskeleton and a mechanical model of the human human leg, which uses motion capture data from the HuMoD database. Based on the model of the human leg supplied with the ideal HuMoD data, the knee motion reference is found. This serves as the data for which the controllers' tracking performances are evaluated. A model for the knee torque induced by the ground reaction force, is derived, and comparing this to the ground reaction torque calculated from the HuMoD data, it is found that the model deviates with a rms error of 10 N during a gait cycle, which means that the general dynamics of the ground reaction force are described by the model, but not to perfection.

For the reactive control scheme, a PD-controller is implemented for controlling the knee torque/force of the hydraulic actuators based on a position error signal. This is implemented along with torque compensation and torque feed-forward. Since the hydraulic actuators are controlled by discrete ON/OFF valves, there are 16 discrete force levels available when controlling the actuator force. In order to select the force level, two force switching algorithms (FSA) are implemented. The FSA1 utilizes chooses the force level closest to the force reference and a weighting term penalizing switching the force level. This penalty term is introduced because switching the force level causes energy losses. The FSA2 is designed to analyze the least energy consuming force level within a force range around the reference. A numerical simulative analysis shows that the least energy consumption and best tracking precision is achieved for the FSA2, which is why this is chosen for the comparative study with the MPC.

The MPC scheme uses a prediction model based on a linearized model of the system to find the optimal input values at a number of time steps on a prediction horizon in order to minimize a cost function. The optimization algorithm chosen for implementing the MPC is the differential evolution (DE) algorithm. Numerical analysis shows that increasing the prediction horizon does not yield better tracking performance. Therefore, a prediction horizon of two is chosen, since this allows using a brute force algorithm to optimize the problem; thereby guaranteeing finding the global minimum. A test is conducted to compare the processing times of the DE-algorithm with the brute force algorithm, and it is found that the brute force algorithm is three times faster (at 0.3 ms), meaning that this is a viable approach. The cost function is designed to penalize the position error, speed error and energy consumption. This is done with three weighting constant, who are chosen through at numerical tuning process.

The sampling time of both controllers is chosen to be 10 ms, which is done to allow time for the valve switching, the settling of the pressure levels as well as computation time.

Testing the reactive controller and the MPC shows that in both cases, successful tracking of the position during the gait cycle is achieved, along with supplying of the desired knee torque. Comparing the rms error of the position and speed over the whole gait cycle, the MPC is seen to outperform the reactive controller by having ≈ 50 % the error. The MPC also achieves very good resemblance with the ideal ground reaction torque; significantly better than for the reactive controller. And lastly, the MPC consumes less energy than the reactive controller and also utilizes significantly fewer force level switches. Furthermore, analysis with different noise levels suggest that the MPC is less sensitive towards noise on the measurement signals. These results are all based on the assumption of having access to the measurement of the knee torque, which allows for torque compensation. Removing the torque compensation has the result of degrading the performance substantially for both the reactive controller and MPC-controller, which suggests that there is significant benefit to be drawn from being able to compensate for the highly influential external disturbance of the ground reaction torque.

8 Future Work

8.1 Experimental Implementation on Prototype

The next step for this project is to validate the designed controllers. This work includes:

- Model validation.
- Validation of friction model and parameter determination.
- Implementation of the reactive controller and MPC-algorithm on the test bench's microprocessor.
- Experimental controller evaluation and validation.
- Benchmarking of the experimental controller results.

Please note, that the results in this report show degraded controller performance without the torque compensation, so if no torque transducer can be implemented on the setup, then the control schemes are expected to perform poorly during an experimental test with ground contact.

8.2 Further Developing the Controller Algorithm

Further development of the control algorithm can include:

- Alternatives to using the knee torque measurements if no measurement equipment can be installed.
- User intention with the purpose of trajectory planning.
- Controller performance at increased walking paces.
- Controller performance under loaded conditions. This could be walking up stairs, or picking up a load.
- Implementation of more pressure levels. If a third pressure level is implemented on the pressure rail and connected to each cylinder chamber, then the force resolution could be improved from 16 discrete levels to 81, which could possibly improve performance.

To summarize, a lot of work is still needed to be done in order to implement the knee exoskeleton successfully in conjunction with the experience of the human wearer.

Bibliography

[1] R. Scheidl, "Digital Fluid Power for Exoskeleton Actuation - Guidelines, Opportunities, Challenges", *The Ninth Workshop on Digital Fluid Power*, pp. 7–8, 2017.

[2] SuitX, *BoostX Knee*. [Online]. Available: `https://www.suitx.com/boostknee`.

[3] Lockheed Martin, *ONYX Exoskeleton Technologies: Military*. [Online]. Available: `https://www.lockheedmartin.com/en-us/products/exoskeleton-technologies/military.html`.

[4] J. H. Kim, M. Shim, D. H. Ahn, *et al.*, "Design of a Knee Exoskeleton Using Foot Pressure and Knee Torque Sensors", *International Journal of Advanced Robotic Systems*, vol. 12, no. 8, 2015.

[5] V. H. Donkov, "Secondary Control of Multi-chamber Cylinders for Low-speed, High-force Offshore Applications", Aalborg University, Tech. Rep., 2020.

[6] M. Linjama, "Digital Fluid Power – State of the Art", *Proceedings of the 12th Scandinavian International Fluid Power Conference Vol3*, pp. 331–xx, 2011.

[7] M. Linjama, H. P. Vihtanen, A. Sipola, and M. Vilenius, "Secondary Controlled Multi-Chamber Hydraulic Cylinder", *SICFP'09*, vol. 1, 2009.

[8] R. Scheidl, M. Linjama, and S. Schmidt, "Is the future of fluid power digital?", *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 226, no. 6, pp. 721–723, 2012.

[9] R. H. Hansen, T. O. Andersen, and H. C. Perdersen, "Analysis of discrete pressure level systems for Wave Energy Converters", *Proceedings of 2011 International Conference on Fluid Power and Mechatronics, FPM 2011*, pp. 552–558, 2011.

[10] A. H. Hansen and H. C. Pedersen, "Energy Cost of Avoiding Pressure Oscillations in a Discrete Fluid Power Force System", *Proceedings of the ASME/BATH 2015 Symposium on Fluid Power and Motion Control*, pp. 1–10, 2015.

[11] A. Hedegaard Hansen, M. F Asmussen, and M. M. Bech, "Energy optimal tracking control with discrete fluid power systems using model predictive control", *9th Workshop on Digital Fluid Power*, no. Ddc, 2017.

[12] M. Linjama, M. Huova, and K. Huhtala, "Model-based force and position tracking control of an asymmetric cylinder with a digital hydraulic valve", *International Journal of Fluid Power*, vol. 17, no. 3, pp. 163–172, 2016. [Online]. Available: `http://dx.doi.org/10.1080/14399776.2016.1185876`.

[13] M. Linjama and M. Huova, "Model-based force and position tracking control of a multi-pressure hydraulic cylinder", *Proceedings of the Institution of Mechanical Engineers. Part I: Journal of Systems and Control Engineering*, vol. 232, no. 3, pp. 324–335, 2018.

[14] R. Rituraj, R. Scheidl, P. Ladner, and Martin Lauber, “A Novel Design Concept of Digital Hydraulic Drive for Knee Exoskeleton”, in *Proceedings of ASME/BATH 2021 Symposium on Fluid Power and Motion Control, FPMC 2021*, 2021, pp. 1–9.

[15] E. Holl, R. Scheidl, and S. Eshkabilov, “Simulation study of a digital hydraulic drive for a knee joint exoskeleton”, *ASME/BATH 2017 Symposium on Fluid Power and Motion Control, FPMC 2017*, pp. 1–8, 2017.

[16] J. Wojtusch, *HuMoD Documentation*, 2017.

[17] P. Alves, F. Cruz, L. F. Silva, and P. Flores, “Synthesis of a Mechanism for Human Gait Rehabilitation: An Introductory Approach”, *Mechanisms and Machine Science*, vol. 24, no. September, 2015.

[18] R. Rituraj and R. Scheidl, “Investigation of an optimal design and control of digital hydraulic drive for knee exoskeleton”, *The 13th International Fluid Power Conference, 13. IFK*, 2022.

[19] M. Cenciarini and A. M. Dollar, “Biomechanical considerations in the design of lower limb exoskeletons”, *IEEE International Conference on Rehabilitation Robotics*, pp. 10–14, 2011.

[20] T. Lee, D. Lee, B. Song, and Y. Su Baek, “Design and control of a polycentric knee exoskeleton using an electro-hydraulic actuator”, *Sensors (Switzerland)*, vol. 20, no. 1, 2019.

[21] S. Kim and H. Murrenhoff, “Measurement of effective bulk modulus for hydraulic oil at low pressure”, *Journal of Fluids Engineering, Transactions of the ASME*, vol. 134, no. 2, 2012.

[22] M. J. L., K. L. G., and B. J. N., *Engineering Mechanics: Statics*, 8th. John Wiley & Sons, 2016.

[23] A. H. Hansen, M. F. Asmussen, and M. M. Bech, “Hardware-in-the-loop validation of model predictive control of a discrete fluid power power take-off system for wave energy converters”, *Energies*, vol. 12, no. 19, 2019.

[24] R. Storn and K. Price, “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces”, *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

A Limit on controller update frequency

Because of the delay time induced by the valve dynamics as well as the pressure transient time, there will be a limit on how fast switching of the valve commands can be done if it is desired to let the pressure levels settle at the desired levels before switching. For the valve dynamics the delay consists of a 2 ms dead time followed by a 1 ms ramp period. The slowest pressure transient time is approximated in the following derivation. Firstly, using the flow continuity equation:

$$\dot{p}_0 = \frac{\beta_0}{\mathrm{V}_0}(Q_0 + \mathrm{A_A}\dot{\mathrm{z}}_0) \tag{A.1}$$

Where the constants in the linearization point are chosen to be:

$$\beta_0(126\text{bar}) = 15340\,\text{bar} \tag{A.2}$$

$$\mathrm{V}_0(\mathrm{z}_{1,\max}) = 4.5 \cdot 10^{-6}\,\mathrm{m}^3 \tag{A.3}$$

$$\dot{\mathrm{z}}_0 = 0.62\,\frac{\mathrm{m}}{\mathrm{s}} \tag{A.4}$$

$$Q_0 = \mathrm{Q_{nom}}\sqrt{\frac{|0.63\mathrm{p_{HP}} - \mathrm{p_{LP}}|}{\mathrm{p_{nom}}}} = 4.72 \cdot 10^{-5}\frac{\mathrm{m}^3}{\mathrm{s}} \tag{A.5}$$

Here, the high pressure is decreased to 63% of the maximum value of 200 bar, which is to compute a case where the gradient is not maximized due to having the largest pressure difference over the valve. The bulk modulus is linearized at a pressure level of 126 which is 63% of the maximum pressure of 200 bar. This is plotted in Figure A.1. The volume is chosen to be the largest possible value, when the piston is at full stroke length. Furthermore, the linearization value for the piston speed is chosen to be the maximum speed according to the HuMoD data during a gait cycle. Lastly, the linearized flow is chosen to be when the pressure difference over the valve is maximal.
Using Equation (A.1), the slowest pressure gradient is then:

$$\dot{p}_0 = 54773\frac{\text{bar}}{\text{s}} \tag{A.6}$$

Which gives a maximum settling time of:

$$\mathrm{t_{max}} = \frac{\mathrm{p_{HP}} - \mathrm{p_{LP}}}{\dot{p}_0} = 3.6\text{ms} \tag{A.7}$$

Adding this value to the valve delay time gives 6.6 ms. Then a buffer is added as a safety margin for a total of 10 ms. This value will be used as the sample time for commanding the valve block, and ensures that the pressure levels will have settled at the desired levels within this time. This corresponds to a maximum valve update frequency of:

$$\mathrm{f_{max}} = \frac{1}{10\,\text{ms}} = 100\text{Hz} \tag{A.8}$$

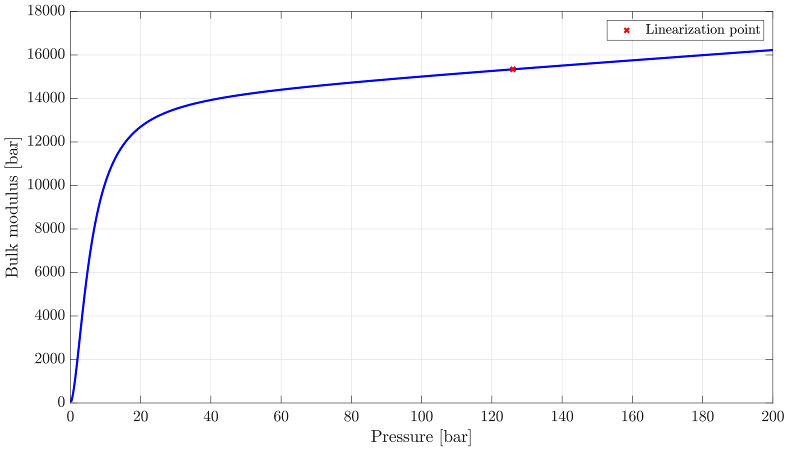


Figure A.1: Bulk modulus and the chosen linearization point.