

Conveying information through non-verbal communication in robotics

Danny M.J. Hameeteman

Master of Science Thesis



Conveying information through non-verbal communication in robotics

MASTER OF SCIENCE THESIS

For the degree of Master of Science in Systems and Control at Delft
University of Technology

Danny M.J. Hameeteman

April 20, 2016

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University
of Technology



Copyright © Delft Center for Systems and Control (DCSC)
All rights reserved.



DELFT UNIVERSITY OF TECHNOLOGY
DEPARTMENT OF
DELFT CENTER FOR SYSTEMS AND CONTROL (DCSC)

The undersigned hereby certify that they have read and recommend to the Faculty of
Mechanical, Maritime and Materials Engineering (3mE) for acceptance a thesis
entitled

CONVEYING INFORMATION THROUGH NON-VERBAL COMMUNICATION IN ROBOTICS

by

DANNY M.J. HAMEETEMAN

in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE SYSTEMS AND CONTROL

Dated: April 20, 2016

Supervisor(s):

dr.ir. G.A. Delgado Lopes

Reader(s):

prof.dr.ir. P.P. Jonker

dr.ir. E.I. Barakova

ir. E. Najafi

Abstract

Robots are becoming more and more involved in our daily life. In order to increase the acceptance of modern robotics in society even further, quite some improvements in human-robot interaction have to be made. Non-verbal communication is one such way of conveying information between robots and humans. In spite of its great potential, the current state of research is fairly limited.

The goal of this research is to create a control framework that could generate movements with the flexibility to convey information by adapting the movement trajectories. Other studies have mainly researched non-verbal interaction using an optimized feed-forward controller. Humans are however able to move without learning all movement sequences in advance, which is the motivation to solely focus on feedback controlled motions in current research. Considering the motion metrics traveled path, electrical power, gravitational torque and end-effector jerk, the proportional-derivative controller in the workspace proves to be best capable of producing the most human-like motions.

The control framework for communicating non-verbally consists of an Operational Space Control hierarchy, combining the aforementioned controller and a novel method minimizing the gravitational torque during movement. Furthermore, human-likeness is even further increased by imposing joint limits based on the human extremes. These limits cause potential local minima, which prevent the controller from reaching the desired final position. Introducing an extra, unusual coordinate helps overcoming these minima taking into consideration the angle between current and final position. With these components included, the framework is able to reach any desired location taking into consideration human-likeness and maximum joint angle constraints.

The Labanotation is used to describe the different motions to be conducted, the same way as is done with human dancing motions. Combining this representation with the

Laban Time, Space and Weight Effort, it is now possible to conduct different movements. Professional dancers were asked to participate in an experiment, resulting in the relation between the end-effector trajectories and the Laban Basic Efforts: dab, flick, float and glide. These trajectories were used to optimize the gains within the control framework. The procedure was run for different initial and final desired positions, resulting in four sets of control gains corresponding to each of the four Efforts.

Different motions are obtained using the four different control sets. Simulation shows that direct and flexible motions (Space Effort) are distinguishable by the shortest trajectories opposed to longer traveling distances respectively. High gravitational torque gains and low spring and damping gains result in more flexible motions. Furthermore, sustained and sudden motions (Time Effort) differ by the time the end-effector needs to reach the desired position. Sudden motions show to have higher spring-damping ratios with respect to sustained motions. In summary, the general control framework proposed by this thesis has the flexibility to generate motions either using the dabbing, flicking, floating or gliding Effort and therefore contributes to the acceptance of robots in society.

Table of Contents

Acknowledgements	xiii
1 Introduction	1
1.1 Background	1
1.2 Literature	2
1.3 Research goal	4
1.4 Outline and scope	5
2 Robotic framework	9
2.1 Model creation	9
2.2 Model verification	12
3 Human-like motion analysis	17
3.1 Feedback controllers	17
3.1.1 Introduction	17
3.1.2 Comparison	20
3.2 Metrics analysis	22
3.2.1 Electrical power	22
3.2.2 Gravitational torque	23
3.2.3 End-effector jerk	29
3.3 Discussion	30
3.4 Synopsis	34
4 General control framework	35
4.1 Operation space control with gravity	35
4.1.1 Introduction to OSC	35
4.1.2 Implementation of OSC	36

4.2	Joint limits	40
4.3	Polar angle	41
4.4	Lyapunov stability	46
4.5	Synopsis	53
5	Convey information	55
5.1	Labanotation input	55
5.2	Laban Efforts	58
5.3	Verification method	61
5.3.1	Experiment	61
5.3.2	Optimization	64
5.4	Results and analysis	67
5.5	Discussion	76
5.6	Synopsis	80
6	Conclusions and recommendations	83
6.1	Conclusions	83
6.2	Recommendations	85
A	Results general movements	87
	Bibliography	95
	Glossary	99
	List of Acronyms	99
	List of Symbols	100

List of Figures

1.1	Police officer controlling the traffic on a busy crossing, using non-verbal communication only [6].	2
1.2	Affective communication models.	3
1.3	Overview of the thesis structure, including reference to the chapters and their relations.	7
2.1	The 3-link manipulator expressing the three configuration coordinates. The subsequent angle is calculated with respect to the previous link's configuration.	10
2.2	Block diagram representing the equation of motion using the system matrices $M(q)$, $C(\dot{q}, q)$ and $G(q)$ and damping term b . The angular velocity and position are found by integrating the acceleration, so the subsequent acceleration could be calculated.	12
2.3	The robot simulated with no torque input and no damping, starting at an arbitrary position in the workspace. The angular position and velocity of each link fluctuate freely in time.	13
2.4	The kinetic co-energy and potential energy and the total energy, as the sum of these two, for the system with no input torque and no dissipation while the damping term equals zero.	14
2.5	The robot simulated with no torque input and some damping, starting at an arbitrary position in the workspace. The velocity decreases to zero as the angular position reaches the lowest potential energy level over time.	15
2.6	The kinetic co-energy and potential energy and the total energy, as the sum of these two, for the system with no input torque and some dissipation while the damping term is greater than zero.	16

3.1	The simulation results of all five introduced controllers of the end-effector. The trajectories in the workspace, configuration velocities and coordinates and input torques differ for the different controllers. The end-effector starts at $[-0.20, -0.70]$ and moves to $[0, 0.72]$	21
3.2	Electrical power curves corresponding to the three motors attached to the links using five different controllers. The integrated total electrical power is included in the title of each plot. The end-effector starts at $[-0.20, -0.70]$ and moves to $[0, 0.72]$	24
3.3	The gravity torque landscape calculated using the vector $G(q)$ for varying q_1 and q_2 , where $q_3 = 0$ rad. The peaks correspond to quite some torque on the motors, while zero represents no gravity pulling at all.	25
3.4	The gravity torque landscape calculated using the vector $G(q)$, with q_1 kept at a constant angle. The influence of varying q_3 seems less significant on the overall gravity torque.	26
3.5	Contour map of the gravity torque for varying q_1 and q_2 values. The blue and green circle are the initial and final position respectively and the red crosses represent the traveled path in terms of configuration coordinates. .	27
3.6	The gravity torque curves for the five different controllers over time. The line at $t = 7$ s represents the moment where the robot gets very close to the desired position.	28
3.7	The jerk of the end-effector in both x- and y-direction together with the overall jerk for all five controllers.	31
4.1	The control structure of the Operational Space Controller with different priorities among the different goals. The task controller moves the end-effector to the desired position and the human-like controller takes the robot's path and configuration into account as a second priority.	37
4.2	Smooth sign function of $G(q)$, such that multiplication of this sign function with $G(q)$ will always result in positive value.	38
4.3	Comparing the motion starting at $[0, -0.72]$ and moving towards $[0.5, 0.5]$ without gravity torque control, with OSC and with OSC including the extra constraint $\ X - X_{\text{des}}\ $. Both the trajectories in the workspace and the gravity torque trajectories are given.	39
4.4	The joint limit functions for each link depending on its angular position. Values equal to zero represent configurations where the limits are not reached. The robot is bounded by $q_1 : [-\pi, \frac{3}{4}\pi]$, $q_2 : [-\frac{1}{12}\pi, \frac{11}{12}\pi]$ and $q_3 : [-\frac{1}{4}\pi, \frac{1}{2}\pi]$	41
4.5	Comparing the same motion as before starting at $[0, -0.72]$ and moving towards $[0.5, 0.5]$, but now without and with joint limit control included. The trajectories in the workspace are given together with the angular position of each link for both situations.	42

4.6	The reachable workspace of robot arm introduced by the joint limits, where each location within the area could be reached. Local minima are hit when the shortest path goes straight through the unreachable area.	43
4.7	The new polar angle describes the angle between the current and desired x-, y-position, so the robot will always move through its reachable area. The polar angle is included as an extra coordinate in the task controller.	43
4.8	The activation function for polar angle control, because it should only influence the movement when there is a change of hitting a local minima. . . .	44
4.9	Comparing the trajectories starting at $[0, 0.72]$ and moving towards $[0, -0.72]$ in the workspace, without joint limits, with joint limits but without polar angle and with both.	45
4.10	LaSalle Invariance Principle, where \mathcal{E} represents the all points in the set Ω where $\dot{V} = 0$, and \mathcal{M} corresponds to the largest invariant set of \mathcal{E} , approached by all solutions in Ω , based on [32].	47
4.11	Partial integrals of the joint limit function H_{jl} called P_{jl} . Values equal to zero represent configurations where the limits are not reached. The robot's configuration is bounded by $q_1 : [-\pi, \frac{3}{4}\pi]$, $q_2 : [-\frac{1}{12}\pi, \frac{11}{12}\pi]$ and $q_3 : [-\frac{1}{4}\pi, \frac{1}{2}\pi]$	48
4.12	Partial derivative of the joint limit function H_{jl} , which is both lower and upper bound. Values equal to zero represent configurations where the limits are not reached. The robot's configuration is bounded by $q_1 : [-\pi, \frac{3}{4}\pi]$, $q_2 : [-\frac{1}{12}\pi, \frac{11}{12}\pi]$ and $q_3 : [-\frac{1}{4}\pi, \frac{1}{2}\pi]$	50
5.1	Example of the Labanotation, one way of representing motion sequences, especially used to describe dancing movements.	56
5.2	Labanotation symbols, a brief selection of the available alphabet [36]. . . .	57
5.3	Input sequence used to control robot, based on dancing exercise <i>Port de Bras</i> . . .	59
5.4	Laban's dynamosphere representing the Time, Space and Weight Effort along with their descriptive opposite elements [18]. The combination of these elements results in the Eight Basic Efforts.	60
5.5	The experimental set-up created on an online platform. The participants are asked to drag the bead towards to cross making use of the mouse. The experiment is found on www.dcsc.tudelft.nl/~glopes/Danny/	62
5.6	The subset of original Laban's dynamosphere, where the Basic Efforts with opposing Weight Effort are joined together.	63
5.7	Example of different trajectories in the workspace and the relative path trajectories over time. Values larger than 1 on the y-axis correspond to more indirect movements towards the desired position.	64
5.8	The experimental results in the workspace obtained from several participants that were asked to move the bead to the cross. Each Basic Effort is separately presented, distinguishable by their Time and Space Effort element. . .	68

5.9	The experimental results of the relative distance traveled over time, obtained from several participants that were asked to move the bead to the cross. Each Basic Effort is separately presented, distinguishable by their Time and Space Effort element.	69
5.10	Generalization of the experimental results used as optimization input. Sudden motions reach the desired position faster and the distance traveled is 1.3 times the shortest path for flexible movements.	70
5.11	The resulting clusters after running the optimization procedure over 12 motions for each Effort. The o represent the optimized control gains and the X shows the mean values for each cluster.	72
5.12	Movement 2 using the generalized set of control gains, from $[0, -0.72]$ to $[0.5, 0.5]$	73
5.13	Movement 5 using the generalized set of control gains, from $[0.5, 0.5]$ to $[0, -0.72]$	73
5.14	Movement 7 using the generalized set of control gains, from $[0.5, -0.5]$ to $[0.5, 0.5]$	74
5.15	Movement 9 using the generalized set of control gains, from $[0.5, 0.5]$ to $[0.5, -0.5]$	74
5.16	Movement sequence using the Labanotation input for the four different Basic Efforts. The robot starts at $[0, -0.72]$ and stops its motion at $[0.5, 0.5]$. The dotted line in the lowest figure corresponds to the shortest distance between each intermediate initial and desired position.	75
1	Movement 1 using the generalized set of control gains, from $[0, -0.72]$ to $[0.72, 0]$	87
2	Movement 2 using the generalized set of control gains, from $[0, -0.72]$ to $[0.5, 0.5]$	88
3	Movement 3 using the generalized set of control gains, from $[0, -0.72]$ to $[0, 0.72]$	88
4	Movement 4 using the generalized set of control gains, from $[0.72, 0]$ to $[0, -0.72]$	89
5	Movement 5 using the generalized set of control gains, from $[0.5, 0.5]$ to $[0, -0.72]$	89
6	Movement 6 using the generalized set of control gains, from $[0, 0.72]$ to $[0, -0.72]$	90
7	Movement 7 using the generalized set of control gains, from $[0.5, -0.5]$ to $[0.5, 0.5]$	90
8	Movement 8 using the generalized set of control gains, from $[0.5, -0.5]$ to $[0, 0.72]$	91
9	Movement 9 using the generalized set of control gains, from $[0.5, 0.5]$ to $[0.5, -0.5]$	91

10	Movement 10 using the generalized set of control gains, from $[0, 0.72]$ to $[0.5, -0.5]$	92
11	Movement 11 using the generalized set of control gains, from $[0.72, 0]$ to $[0, 0.72]$	92
12	Movement 12 using the generalized set of control gains, from $[0, 0.72]$ to $[0.72, 0]$	93

List of Tables

2.1	The mass, inertia and link lengths of the 3-link manipulator, based on real human measurements [22].	11
3.1	Control components of each controller captured in the general control law of Equation (3.8).	20
3.2	The total gravity torque calculated by the area underneath the curves. The moving gravity torque is a clipped version of the total, where all values above $t = 7$ s are omitted.	29
3.3	Summary of the compared metrics for each controller.	33
5.1	Created relation between the Labanotation symbols and the desired workspace coordinates of the end-effector.	58
5.2	K-clustered mean gain values that correspond to the four different Basic Efforts dabbing, flicking, floating and gliding.	71
5.3	Ratios of the mean gain values in x-, and y-direction for to the four different Basic Efforts.	77

Acknowledgements

First of all, I would like to thank my supervisor dr.ir. Gabriel Lopes at the Delft University of Technology for proposing this thesis subject and giving me the opportunity to work in this very interesting field. During our regular meetings he always came up with great ideas, which guided me into the right direction. Furthermore, my supervisor always had a positive attitude towards a successful outcome of the project.

I would also like to acknowledge dr. Victoria Watts, chair of dance at the Cornish college of the arts, who has given us great insight into the Laban Efforts and the way we could use them. The experiment was improved by her critical reflection and she was able to spread the website among her students.

Furthermore, I am very grateful to the help of ir. Cees Verdier, who provided me with useful new insights over the whole course of this thesis. I could always knock on his door and discuss the difficulties I had during my project. Especially the struggles I faced with the Lyapunov stability proof became much easier with the Cees' help.

A special thanks goes to my girlfriend Tessa, my parents Ben and Diana and my brother Kevin. They supported me morally and financially not only during my thesis, but over the entire study at the TU Delft. They are the reason finishing this work and the study Systems and Control was possible.

Finally, I should not forget expressing my gratitude towards the fellow students from 'De Beun Zolder'. The daily coffee and lunch breaks took my mind of the project just long enough to keep the productivity high.

Delft University of Technology
April 2016

Danny M.J. Hameeteman

“The road of life twists and turns and no two directions are
ever the same. Yet our lessons come from the journey, not the
destination.”

— *Don Williams, Jr.*

Introduction

1.1 Background

Robots are becoming more and more involved in our daily live, think for instance of grass cutting robots or bomb disposal robots. The amount of interaction between humans and robots increases, which is especially important in applications within service industries, like robots replacing human health care employees for example. In order to improve this interaction, it is wise to investigate how robots could become more accepted in our society. One way to solve this is by looking at the way we communicate with one another and implement these behaviors within robotics. Besides making things clear verbally, between 60% to 65% of our communication happens during non-verbal communication [1]. Here, facial expressions and eye gaze are two important aspects to emphasize on the spoken information [2]. For this reason, lots of research is conducted concentrating purely on conveying information using robotic facial expressions [3].

Besides using the face, humans are very capable of understanding each other by making gestures with parts of their body [4], [5]. We could for example think of a police officer controlling the cars on a busy crossing with broken traffic lights, as seen in Figure 1.1. It is clear that nobody inside the cars will be able to understand what the police officer is saying. However, even with limited knowledge about the rules, the gestural meanings are understandable to everybody. This way of communication has started to grow within the robotics area, but has still a long way to come before robotic gestures are indistinguishable from human gestures.

Another reason for choosing non-verbal communication to convey information is based on the high expectations verbal communication evokes [7]. These expectations should be managed, while we cannot rely on the technology for conveying naturally yet. This results in the loss of interest in these robots, while its vocabulary is limited and is most of the time not able to cope with unexpected situations. These communicating robots will be rated unreliable for communication purposes [8].



Figure 1.1: Police officer controlling the traffic on a busy crossing, using non-verbal communication only [6].

1.2 Literature

The question arises why is it so hard to design a robotic algorithm which performs human gestures naturally. And how come the existing public robots do not possess different communication styles yet? An evaluation of the existing literature on improving non-verbal communication in robotics, showed that only very limited research concerning feedback controlled robots has done. Some interesting results found within current literature survey are presented here. A more extensive overview is given in the literature survey preceding current work [9].

The potential benefits of correct non-verbal conveying of information are [10]:

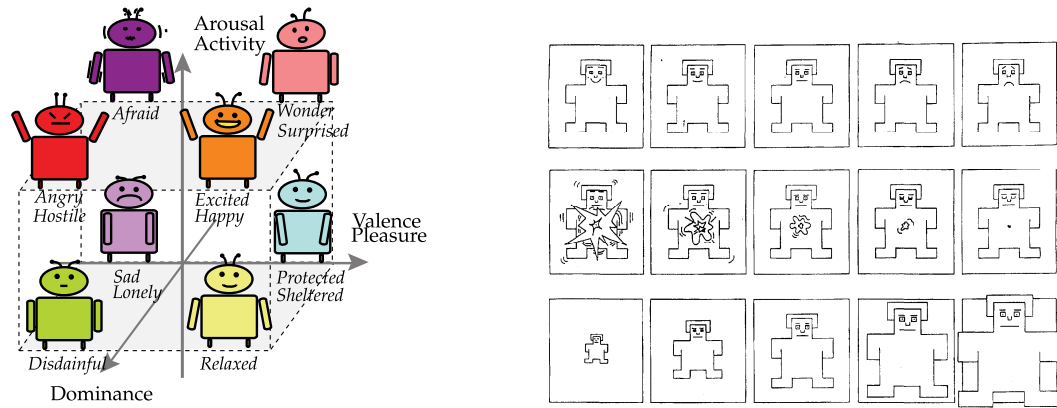
- transmitting information without adding additional loads to the existing communication system,
- forestalling miscommunication due to the lack of non-verbal communication
- forming a bond with the robot, because of human identification.

Affective communication Non-verbal communication between humans could consist of emotional content corresponding to the feelings of a person, but could also be less focusing on the human's emotional state as studied in [4] and [11].

One example of mathematically describing human emotions is captured in the Pleasure, Arousal and Dominance (PAD) Emotional State Model [12]. This model is able to express the positive or negative affective state, the activity stimulus and the level of control over the surroundings. These three measurement levels combined results in the continuous PAD space as shown in Figure 1.2. Combinations of the extremes represents the eight different emotional states: relaxed, excited, surprised, protected, angry, disdainful, afraid and sad.

These eight emotions could be captured using the Self-Assessment Manikin (SAM). This model provides a non-verbal assessment technique to measure the pleasure, arousal

and dominance levels, using the images as pictured in Figure 1.2 [13]. The 5-pointer scale significantly simplifies the judgment of someone's affective state with respect to continuous qualification.



(a) Pleasure-Arousal-Dominance (PAD) Emotional State model, able to represent at least eight different emotions: relaxed, excited, surprised, protected, angry, disdainful, afraid and sad [4].

(b) The Self-Assessment Manikin (SAM) rates the pleasure (top), arousal (middle) and dominance (bottom) level using a 5-pointer scale for each dimension [13].

Figure 1.2: Affective communication models.

Non-affective communication Yet not all communication appears to be affective, making it interesting to look at non-verbal communication as well. Non-verbal communication could be categorized into 5 types [14]: emblems are self-containing gestures, iconics are gestures showing concrete things and actions, metaphors are iconics referring to imaginary concepts, deictics are pointing gestures and beats are rhythmic, emphasizing motions. Only emblems and deictic gestures are understandable without the need for spoken words.

Utilizing these two gesture types, Rudolf Laban [15] derived a method to explain dance movements in the terms Shape and Effort. Shape refers to the movement's quality (or: structure) along three different planes: Table (Horizontal), Door (Vertical) and Wheel (Sagittal) Plane. Effort explains the content of the movement, separated into Space, Weight, Time and Flow, each consisting of two extremes: light/strong, sustained/sudden, indirect/direct and free/bound respectively. Combining these extremes results in a brief description of the conveyed information during movement.

Human movement optimization Humans tend to move their arm from point-to-point in straight lines, their movements are smooth and have bell-shaped velocity profiles [16]. These movements are independent of the hand's initial and final position. An optimization procedure could be constructed to optimize the cost function containing

the kinematic, integrated jerk or dynamic, torque change. The results show that the straight line in visual space cannot be purely based on dynamic-based models like minimum torque change, but kinematics need to be implemented as well.

The metabolic power consumption as humans are learning new arm reaching dynamics, was the inspiration to express the motor muscle activities [17]. The metabolic power decreases while learning, together with the muscle activity and coactivation. It is found that robotic motor learning also experienced diminishing power consumption during the learning movement, which proves minimization of the energetic cost.

Non-verbal control examples Several researches have made an attempt to control robotic movements resulting in non-verbal communication. An example is based on the use of the Laban Efforts to generate stylized movements [18]. Introducing an LQ optimal control algorithm, where the Q, R, P and S as weight matrices, corresponds to the Space, Weight, Time and Flow Effort. The optimization results in the best matrices by approaching the captured human trajectories with different styles. These matrices could then be used to generate movements having a specific style.

Besides, expressing moods through body language in functional behavior, like pick-and-place tasks, is investigated [19]. The advantage of using moods is that they can be conveyed without interrupting the task, while non-verbal gesturing needs to be specific. Besides, it was found that moods are more longer-lasting than gestures and thus a more stable and reliable way of communicating. A three-layer model was created, divided into task scheduling and affect generating, the behavioral parameters Pose and Motion, and the joint configurations. Especially the middle layer is responsible for adjusting the different styles of the movement. The researchers concluded that it is possible to construct a general set of parameters related to the PAD model, in order to convey a variety of behaviors.

Finally, the research done by [20] laid down the basis for current research, focusing on gravity compensation based on the emotional system. The virtual gravity vector could either point down or up, or might be not available at all. The researches hypothesized that gravity vectors pulling down correspond to a more negative sensation, while vectors pulling up make movements look more positive. The SAM tools provided that dominance is higher for both the upward and downward gravity vector, when compared to the no gravity vector. Pleasure was rated higher for downward gravity vector. For future research it is recommended to include velocities and accelerations in the control framework, since the gravity vector is not sufficient to express emotions.

1.3 Research goal

The literature discussed in previous section explained a number of ways for describing non-verbal communication, both affective and non-affective. Moreover, it was summarized how to consider human-like movements and the way to generate non-verbal

communication based on different models. There is only very limited research to be found on combining the stylistic non-verbal communication approach with the functional task, in order to improve human-robot interaction. Therefore, the goal of this thesis will be:

“Create a control framework that has the flexibility to convey information through non-verbal communication, by adapting the movement trajectories.”

In order to achieve this goal, this research will be subdivided into the following three steps:

1. Find and analyze the most human-like controller by creating new human-like motion metrics.
2. Design a general control framework to conduct human-like motions, based on previous results and metrics.
3. Construct a novel way of generating different motions using only one control framework.

The findings resulting from these subgoals would compose a novel framework to create different motions that have the potential to convey information non-verbally.

1.4 Outline and scope

The research to achieve the goal and subgoals described above are organized in the following structure, where each chapter is a natural follow-up on the preceding chapter.

Chapter 2: Robotic framework In this chapter, the robotic framework will be introduced, while considering the assumptions. The model created is used during the rest of the study.

Chapter 3: Human-like motion analysis Several metrics representing the most human-like motion will be constructed here. Furthermore, five basic feedback controllers are analyzed taking into consideration these metrics.

Chapter 4: General control framework The previous chapter results in the most human-like controller according to the motion metrics, which will now form the basis for the control framework. The introduced metrics help improving the human-likeness even further and Lyapunov stability is proven.

Chapter 5: Convey information Conveying information is shown in this chapter using the defined control framework, regarding several different styles. An experiment shows the relation between the participant's input and the styles. An optimization procedure obtains multiple sets of general control gains, one for each effort.

Chapter 6: Conclusions and recommendations The final chapter will give a conclusion, and based on this some recommendations for future work.

Appendix A: Results general movements The results of all simulations are demonstrated for all different styles.

The way these chapters relate to each other can be found in Figure 1.3. After this introduction, the report has split up in two parts, one focusing on the modeling and the other on the work done by Rudolf Laban. The robotic framework will be used in the human-like motion analysis as well during the optimization. The Operational Space Controller applies the knowledge about human-like motions obtained in the analysis. The Laban is divided into the Laban Efforts and the Labanotation. All these parts come together in the general control framework leading towards the conclusion of this work.

First and for most, it is important to note that the scope of this research concentrates on motions, because the inspiration comes from human non-verbal communication techniques. Therefore, conveying a specific mood using for instance different light colors or sounds is excluded. Furthermore, this research does not include facial expressions, as much research has been conducted on this topic already [3]. For the same reason, synchronizing expressions and movements with speech is also minimally explored. Although emotion-based, non-verbal communication has been researched in the DCSC department at the Delft University of Technology before [20], current work will not be focusing on emotional content. Finally, this work concentrates on controlling solutions for robots and does therefore not take into account any psychological and neuroscientific theories.

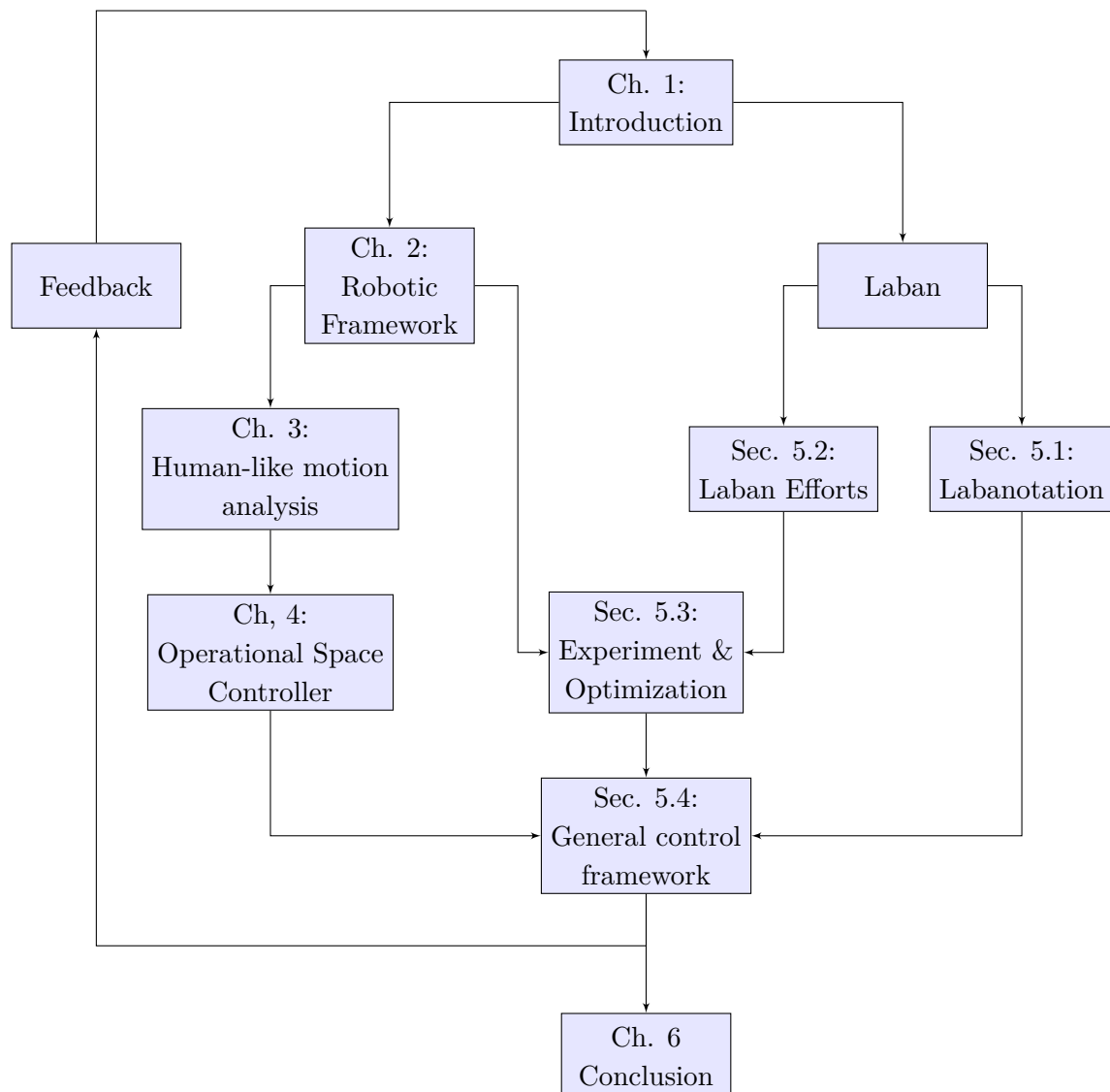


Figure 1.3: Overview of the thesis structure, including reference to the chapters and their relations.

Robotic framework

Humans move in the three dimensional space, but in order to prove the concept of conveying information through non-verbal communication this research will only be looking at motions in 2D. By simplifying the model it becomes much easier to determine how the different metrics perform, like shown later. First the model will be built, after which it is verified making use of the potential energy and kinetic co-energy. The derivation is based on [21].

2.1 Model creation

Before going into detail about the way the model is constructed, the assumptions preceding this model should be considered first. The first and most obvious assumption is the restriction of the movements in 2D, which could be compared to movements of the human arm in the sagittal plane attached to the shoulder. Considering this shoulder, translations are not allowed, but the model will only be able to rotate around their axis. The complexity muscle models could introduce is not taken into account here, but a direct connection with the controller is maintained.

The robotic manipulator represents the human's arm, so consists of an upper arm, a forearm and a hand. These three links are assumed to be rigid and straight, connected to each other with an elbow and a wrist joint. The robot is connected to the environment with a shoulder joint attached to the upper arm. Furthermore, the mass and inertia of each link are concentrated in the center of the body. Each joint contains linear friction with respect to the angular velocity, so the motion are damped out when not controlled.

Figure 2.1 presents the 3-link manipulator with its frame in configuration coordinates. Notice that the angle between the horizontal axis and the second link increases with an increasing angle of the first link.

From this figure it should be possible to find the equations of motion (EOM)

$$M(q)\ddot{q} + C(\dot{q}, q)\dot{q} + b\dot{q} + G(q) = u, \quad (2.1)$$

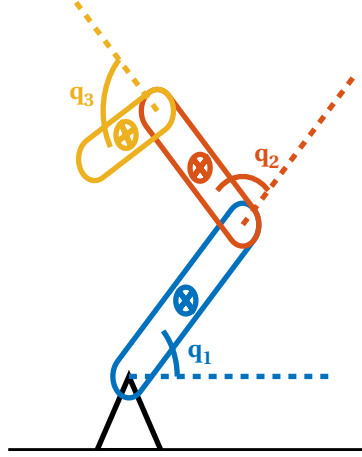


Figure 2.1: The 3-link manipulator expressing the three configuration coordinates. The subsequent angle is calculated with respect to the previous link's configuration.

where vectors q , \dot{q} , \ddot{q} are the angles, velocities and accelerations of the links and u represent the input torque. Each joint possesses one motor, which drives the corresponding links independently. Furthermore, the matrices $M(q)$, $C(\dot{q}, q)$ and $G(q)$ are the Inertia, Coriolis and Potential matrix respectively. Finally, the term $b\dot{q}$ represents the linear damping term in the velocity, always working against the direction of movement.

The EOM of Equation (2.1) could be found after constructing the robot's kinematic co-energy and potential energy

$$T^* = \sum_{i=1}^3 \frac{1}{2} \left(m_i (\dot{x}_i^2 + \dot{y}_i^2) + I \dot{\theta}_i^2 \right), \quad (2.2)$$

$$V = \sum_{i=1}^3 \left(m_i g \left(y_i + \sum_{n=1}^{i-1} (2r_n) + r_i \right) \right). \quad (2.3)$$

The kinetic energy of Equation (2.2) depends on the velocity of the COM in the workspace. The latter terms in Equation (2.3) represents to the shift in potential energy, so that each link is pointing downwards refers to zero potential energy. The masses, link lengths and inertia terms are based on real measured human arms data [22] and shown in Table 2.1. The inertia of each link is calculated using the assumption of a point mass around the joint with

$$I = mr^2. \quad (2.4)$$

Before the x -, y - and θ -velocities could be calculated, the relation between robot's configuration coordinates and the workspace coordinates are constructed, as shown in Equations (2.5)-(2.7). Notice that the angle of each consecutive link is depending on

Table 2.1: The mass, inertia and link lengths of the 3-link manipulator, based on real human measurements [22].

	Link 1	Link 2	Link 3
Mass (kg)	3.25	1.87	0.65
Length (m)	0.35	0.28	0.18
Inertia (kg · m ²)	0.10	0.037	0.0053

the angle of the previous links as well.

$$\begin{aligned}x_1 &= r_1 \cos(q_1), \\y_1 &= r_1 \sin(q_1), \\ \theta_1 &= q_1.\end{aligned}\tag{2.5}$$

$$\begin{aligned}x_2 &= 2x_1 + r_2 \cos(q_1 + q_2), \\y_2 &= 2y_1 + r_2 \sin(q_1 + q_2), \\ \theta_2 &= q_1 + q_2.\end{aligned}\tag{2.6}$$

$$\begin{aligned}x_3 &= x_2 + r_2 \cos(q_1 + q_2) + r_3 \cos(q_1 + q_2 + q_3), \\y_3 &= y_2 + r_2 \sin(q_1 + q_2) + r_3 \sin(q_1 + q_2 + q_3), \\ \theta_3 &= q_1 + q_2 + q_3.\end{aligned}\tag{2.7}$$

The corresponding workspace velocities can be found by taking the derivatives of the aforementioned equations with respect to time, so

$$\begin{aligned}\dot{x}_i &= \frac{\partial x_i}{\partial t}, \\ \dot{y}_i &= \frac{\partial y_i}{\partial t}, \\ \dot{\theta}_i &= \frac{\partial \theta_i}{\partial t}.\end{aligned}\tag{2.8}$$

The coordinate equations can be summarized into a vector $g(q)$, which represents the forward kinematic map as seen in Equation (2.9), based on Eq. (2.5)-(2.7),

$$r = g(q),\tag{2.9}$$

where r corresponds to the workspace coordinates x , y and θ . The velocities are captured in Equation (2.10), with $J(q)$ the partial derivative of $g(q)$ to q

$$\dot{r} = J(q)\dot{q},\tag{2.10}$$

where

$$J(q) = \frac{g(q)}{\dot{q}}.\tag{2.11}$$

Now that all unknown positions and velocities are translated from configuration coordinates into workspace coordinates, the EOM can be constructed. The kinetic co-energy and potential energy derive the matrices of Equation (2.1) as

$$\begin{aligned} M(q) &= \frac{\partial^2 T}{\partial \dot{q}^2}, \\ C(\dot{q}, q) &= \frac{\partial^2 T}{\partial \dot{q} \partial q} - \frac{1}{2} \frac{\partial M(q)}{\partial q} \dot{q}, \\ G(q) &= \frac{\partial V}{\partial q}. \end{aligned} \quad (2.12)$$

These equations are extracted using the Lagrangian mechanics, but is not discussed in more detail here.

Now that all matrices of Equation (2.1) are known, it will be implemented into MathWorks' Simulink. To do so, the EOM should first be rewritten so that the highest order derivative is calculated from all other variables, so:

$$\ddot{q} = M^{-1}(q) (u - C(\dot{q}, q)\dot{q} - b\dot{q} - G(q)). \quad (2.13)$$

The term u representing the control input torque will be discussed in next section. Figure 2.2 shows the block diagram implementation of this model. The integration and double integration of the angular acceleration are calculated online, so that the angular velocity and position could be used in the subsequent acceleration calculations.

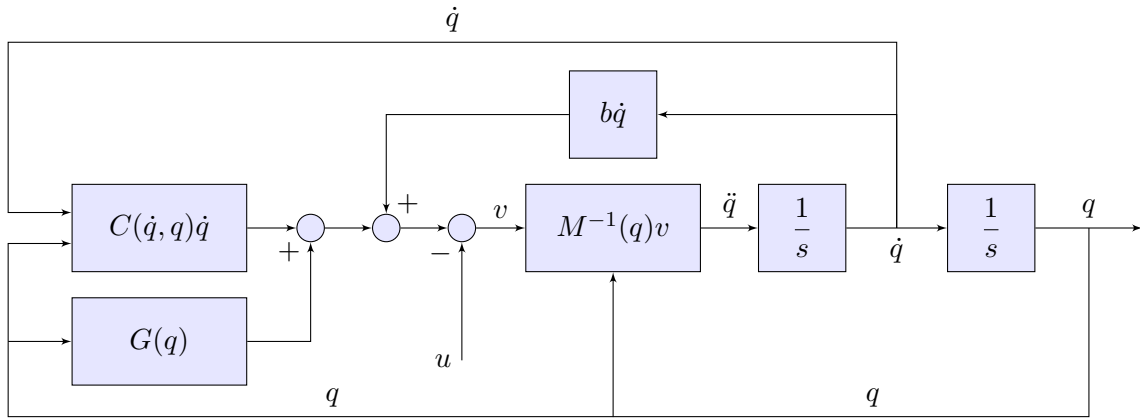


Figure 2.2: Block diagram representing the equation of motion using the system matrices $M(q)$, $C(\dot{q}, q)$ and $G(q)$ and damping term b . The angular velocity and position are found by integrating the acceleration, so the subsequent acceleration could be calculated.

2.2 Model verification

As a verification of the implementation shown earlier, the robot is simulated with zero input torque using an arbitrary initial condition not located in one of the equilibrium

positions. These equilibria correspond to the configurations where all links are pointing upwards or downwards or a combination of both. Furthermore, the damping equals zero, which is useful to check the energy conditions. Figure 2.3 shows the velocity and position profile of the three links evolving over time. It can be seen that the links do

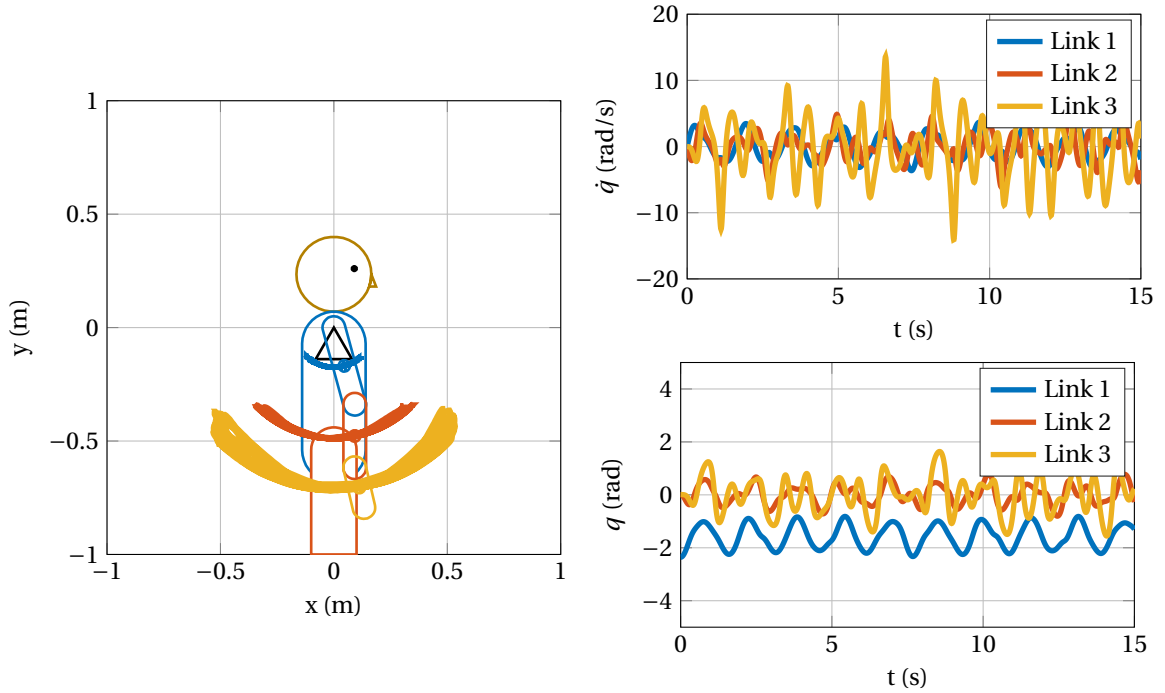


Figure 2.3: The robot simulated with no torque input and no damping, starting at an arbitrary position in the workspace. The angular position and velocity of each link fluctuate freely in time.

not come to rest during the time interval of 15 seconds. Looking at the trajectory the links have taken, it could be concluded that the motion cannot really be predicted.

Both results could be explained from the fact that the links have a certain initial energy, which is not dissipated due to the lack of damping. Figure 2.4 confirms this thought, while the total amount of energy (kinetic co-energy plus potential energy) does not change over time.

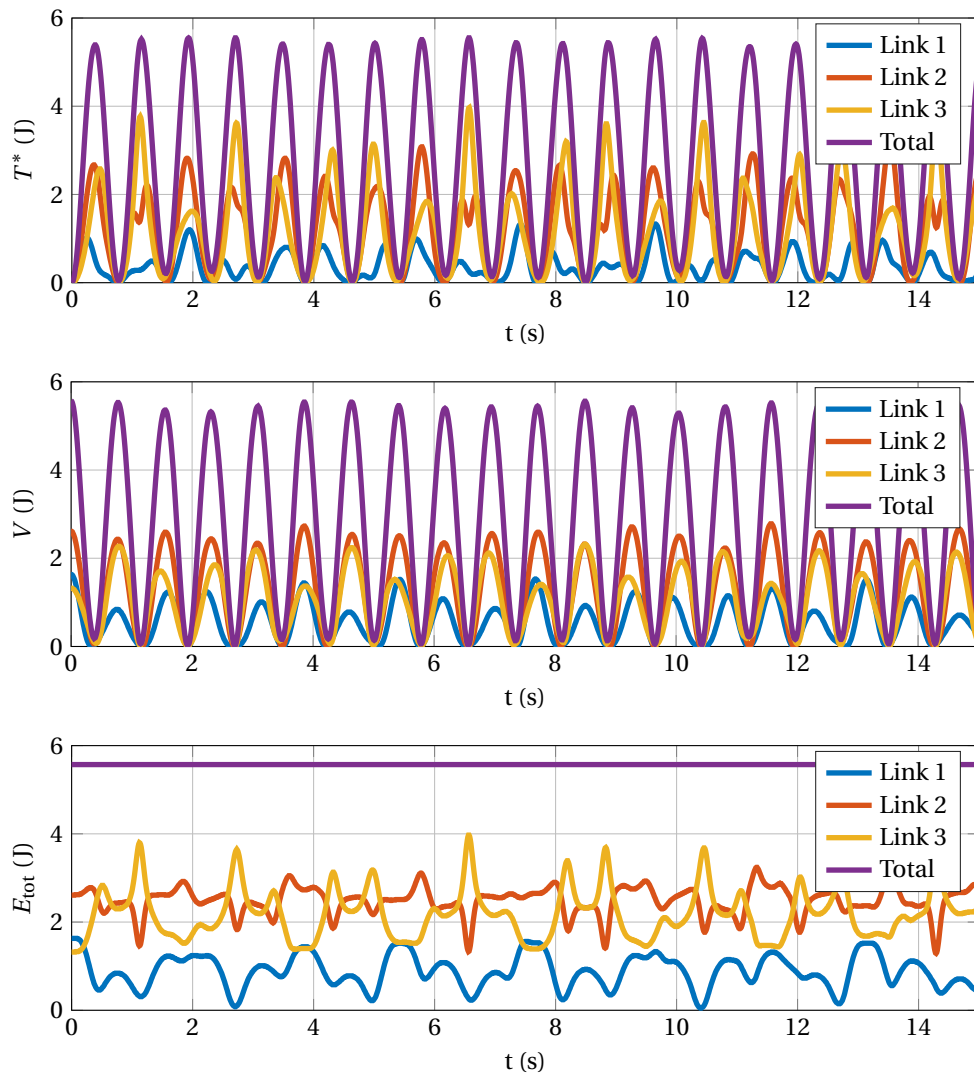


Figure 2.4: The kinetic co-energy and potential energy and the total energy, as the sum of these two, for the system with no input torque and no dissipation while the damping term equals zero.

The damping term will now be greater than zero, so dissipation takes places in the EOM. The same initial position and velocity is used together with the zero control input again. The results of this simulation is shown in Figure 2.5 and 2.6.

It can indeed be seen that the velocity goes to zero, when the links are moving over time. The final position is found when all links are pointing downwards in the minimum potential energy level. Furthermore, the total amount energy decreases to zero as well, as the damping term dissipates the energy.

These two results confirmed that the model is correctly implemented and could therefore be used during the rest of the research.

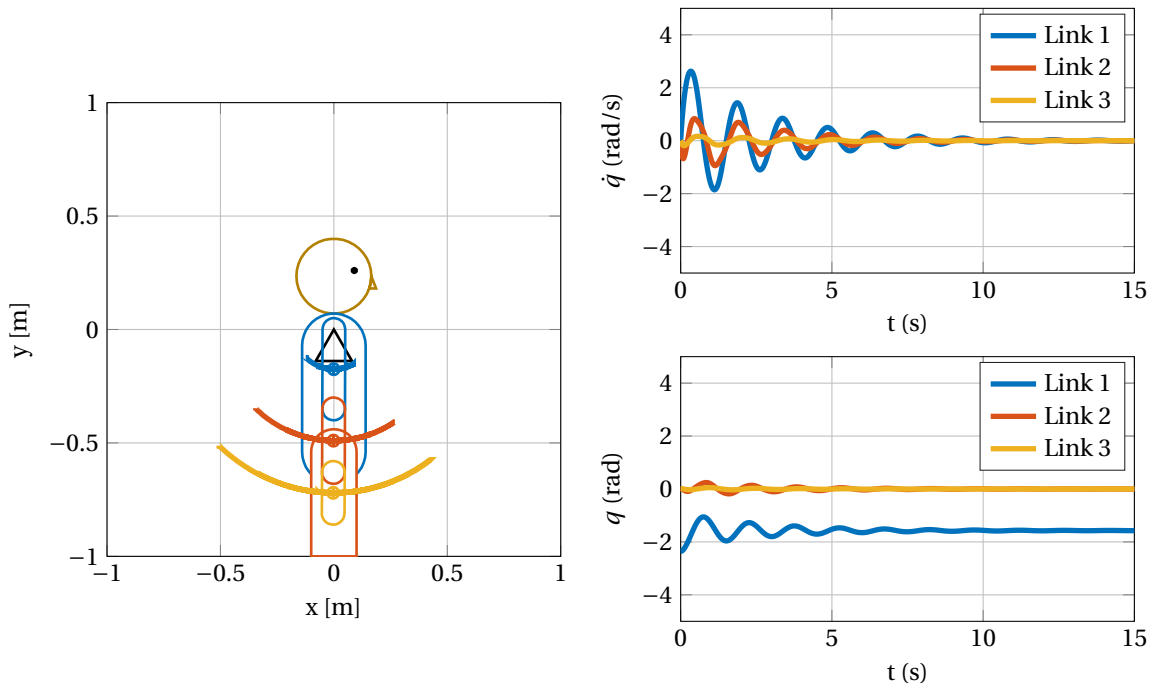


Figure 2.5: The robot simulated with no torque input and some damping, starting at an arbitrary position in the workspace. The velocity decreases to zero as the angular position reaches the lowest potential energy level over time.

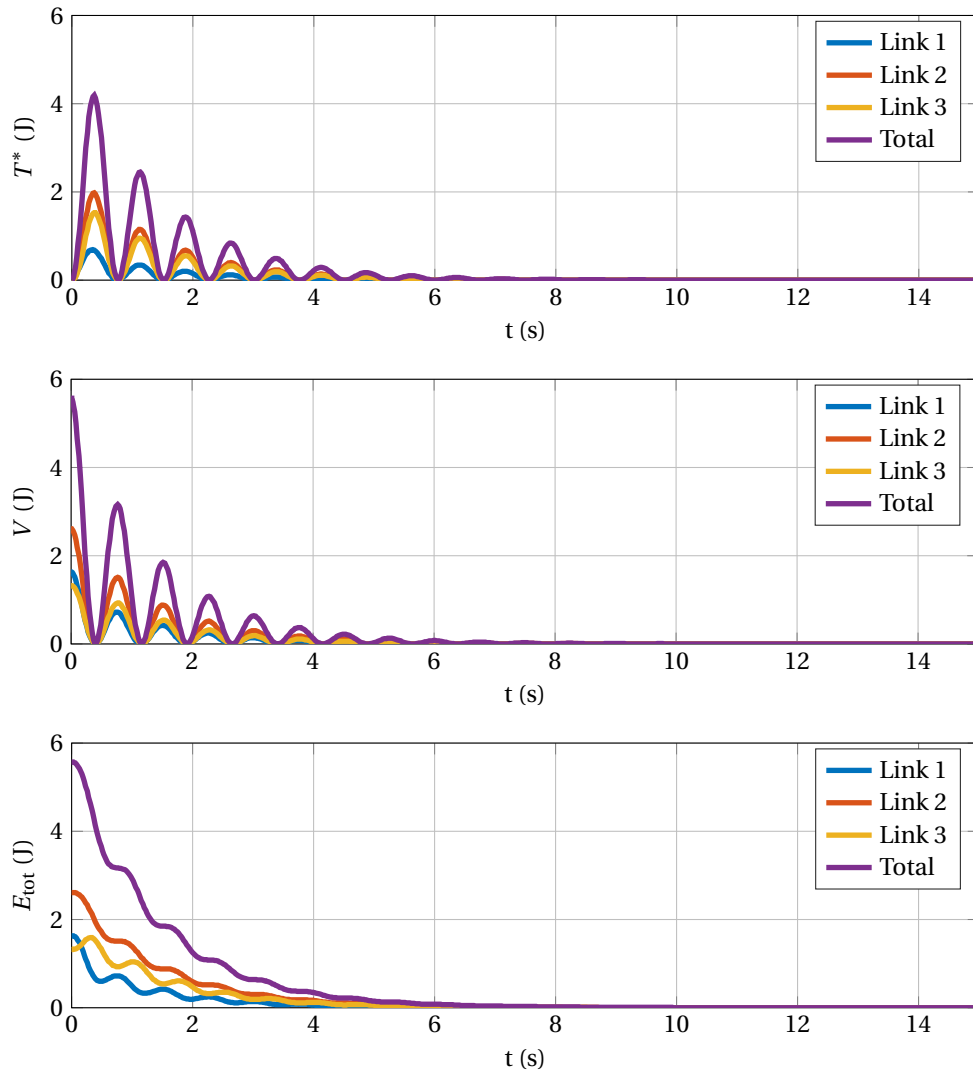


Figure 2.6: The kinetic co-energy and potential energy and the total energy, as the sum of these two, for the system with no input torque and some dissipation while the damping term is greater than zero.

Human-like motion analysis

There exists quite some controversy when it comes to determine whether a motion is human-like or not. Like described in Chapter 1, several researches have tried to solve this problem, but no general form is presented so far. The current chapter will make an attempt to find a set of metrics that is able to qualify the human-likeness of the conducted motions.

This chapter will start of by introducing the different feedback controllers used in this research and a preliminary comparison between each of them is given. Next, an introduction to the metrics with the greatest potential to describe human-likeness is presented. Finally, the effect of each controller on the different metrics is discussed after which the most human-like controller, according to these metrics, is chosen.

3.1 Feedback controllers

Chapter 2 concluded that the model works properly when no input torque is presented, so the next step will be to control the arm's configuration. The goal of this research is to create general movements with certain information, which should not depending on the initial nor the final position. That is the reason for taking feedback controllers into consideration only, while no learning or optimization is required to move from A to B. Besides, humans did also not have to learn every movement before actually being able to conduct them.

3.1.1 Introduction

The list of available feedback controllers is endless, but in general the simpler the controller structure, the better the influence of each part could be derived. The following relatively uncomplicated feedback controllers are chosen and will be discussed next:

- Proportional-derivative controller with gravity compensation (PD)

- Computed Torque controller (CT)
- Proportional-derivative controller in workspace coordinates (PDW)
- Computed Torque controller in workspace coordinates (CTW)
- Gravity up controller (GU)

PD controller The PD controller can be thought of as a virtual, rotational spring-damper system between the current and desired position. These spring and damper are placed at the joint of the link, while it needs to correct the error in the configuration space. The gravity compensation term is responsible for canceling out gravity effects on the link, so excludes potential steady-state errors. The control law of this controller is given as

$$u = -K_p e - K_d \dot{e} + G(q), \quad (3.1)$$

where K_p represents the spring constant and K_d the damping coefficient. Furthermore, e is the difference between the current and the desired position in configuration coordinates, so $e = q - q_{\text{des}}$.

CT controller The Computed Torque controller is able to completely compensate for non-linearities in the system by making use of the system matrices $M(q)$, $C(\dot{q}, q)$ and $G(q)$. The cancellation of all these parts results in a simple linear spring-damper system, moving the hand from current to the desired position. Equation (3.2) shows control law corresponding to the CT controller

$$u = M(q) (\ddot{q}_{\text{des}} - K_d \dot{e} - K_p e) + C(\dot{q}, q) \dot{q} + G(q). \quad (3.2)$$

In order to use this controller, the system matrices have to be known with certainty, because otherwise the controller will not compensate correctly. Using the correct system matrices the entire system is converted into the error dynamics

$$\ddot{e} + K_d \dot{e} + K_p e = 0. \quad (3.3)$$

with e representing the same error as before.

PDW & CTW controller The previously described controllers are considered in the robot's configuration space, but can also be constructed in workspace coordinates, so x , y and θ . To do this, the vector $g(q)$ of Equation (2.9) and its partial derivative to q called $J(q)$ (Eq. (2.10)) will be needed to translate the configuration coordinates into workspace coordinates. It can be seen that the control laws for the PD control in the workspace and the CT control in the workspace are as shown in Equation (3.4) and (3.5) respectively:

$$u = -J^T(q) (K_p e_r + K_d \dot{e}_r) + G(q), \quad (3.4)$$

$$u = J^T(q) \left(\bar{M}(q) (\ddot{r}_{\text{des}} - K_d \dot{e}_r - K_p e_r) + \bar{C}(\dot{q}, q) \dot{r} + \bar{G}(q) \right). \quad (3.5)$$

The error functions e_r and \dot{e}_r correspond to the difference between the current and the desired workspace coordinates x , y and θ , so: $e_r = r - r_{\text{des}}$. Besides, the introduced system matrices with bar in Equation (3.5) are represented by

$$\begin{aligned} \bar{M}(q) &= J^{-T}(q) M(q) J^{-1}(q), \\ \bar{C}(\dot{q}, q) &= J^{-T}(q) \left(C(\dot{q}, q) J^{-1}(q) + M(q) \frac{d}{dt} (J^{-1}(q)) \right), \\ \bar{G}(q) &= J^{-T}(q) G(q). \end{aligned} \quad (3.6)$$

GU controller Next to these four traditional controllers, the GU controller makes use of the gravity vector to control the robot's configuration. The idea behind creating this controller is based on the research by [20]. The results will describe a free falling kind of motion in the desired direction. Equation (3.7) shows the control law corresponding to this idea, which has some similarities with the PD controller in workspace of Equation (3.4)

$$u = -J^T(q) K_d J(q) (\dot{q} - \dot{q}_{\text{des}}) + K_G G(q - q_{\text{des}}) + G(q). \quad (3.7)$$

The first term represents the damping term in workspace coordinates, $G(q - q_{\text{des}})$ calculates the converted gravity field into the desired manipulator's direction and K_G is the gain controlling this converted gravity. Depending on the desired configuration coordinates the gravity field will rotate accordingly.

The control laws corresponding to the controllers explained above can be generalized into one control law. This general control law consists of an energy insertion part, an energy extraction part and a compensation part. Equation (3.8) shows the three parts in respective order, where q and \dot{q} represent the robot's position and velocity in configuration coordinates again

$$u = P d(q, q_{\text{des}}) + D f(\dot{q}, \dot{q}_{\text{des}}) + \Gamma(q, \dot{q}). \quad (3.8)$$

The energy insertion term consists of a gain P and an error function $d(q, q_{\text{des}})$ expressing the difference between current and desired position. The energy extraction term has a damping constant D and an error function $f(\dot{q}, \dot{q}_{\text{des}})$ corresponding to the velocities. In some cases the desired velocity \dot{q}_{des} equals zero, meaning that the robot should stop moving when the desired position is reached. The last term $\Gamma(q, \dot{q})$ is able to compensate for the robot's kinematics like gravity and Coriolis effects according to its position and velocity.

All described controllers could be captured in this general control law of Equation (3.8). Table 3.1 shows how each component of all described controllers should be related to the general control law.

Table 3.1: Control components of each controller captured in the general control law of Equation (3.8).

	P	$d(q, q_{\text{des}})$	D	$f(\dot{q}, \dot{q}_{\text{des}})$	$\Gamma(q, \dot{q})$
PD	$-K_p$	$q - q_{\text{des}}$	$-K_d$	$\dot{q} - \dot{q}_{\text{des}}$	$G(q)$
CT	$-K_p$	$M(q)(q - q_{\text{des}})$	$-K_d$	$M(q)(\dot{q} - \dot{q}_{\text{des}})$	$M(q)\ddot{q}_{\text{des}} + C(\dot{q}, q)\dot{q} + G(q)$
PDW	$-K_p$	$J^T(q)J(q)(q - q_{\text{des}})$	$-K_d$	$J^T(q)J(q)(\dot{q} - \dot{q}_{\text{des}})$	$G(q)$
CTW	$-K_p$	$J^T(q)\bar{M}(q)J(q)(q - q_{\text{des}})$	$-K_d$	$J^T(q)\bar{M}(q)J(q)(\dot{q} - \dot{q}_{\text{des}})$	$J^T(q)\bar{M}(q)\ddot{r}_{\text{des}} + \bar{C}(\dot{q}, q)J(q)\dot{q} + \bar{G}(q)$
GU	K_G	$G(q - q_{\text{des}})$	$-K_d$	$J^T(q)J(q)(\dot{q} - \dot{q}_{\text{des}})$	$G(q)$

The table shows that the main differences between the controllers is found in the error functions $d(q, q_{\text{des}})$ and $f(\dot{q}, \dot{q}_{\text{des}})$. These differences will result in different trajectories and velocities conducted by the robot, next section compares these differences between each controller in terms of performances.

3.1.2 Comparison

Like said, the main difference between the feedback controller could be found in the paths the end-effector are taking, as well as the corresponding velocities and torque inputs. Figure 3.1 shows these exact difference between the controller results, starting at $[-0.20, -0.70]$ and moving towards $[0, 0.72]$. All controllers contain gravity compensation and the gains are tuned so that the desired location is reached within the same amount of time of $t = 7$ s.

The figure helps comparing the simulation results belonging to the different controllers. First looking at the trajectories in the xy-space it is shown that the configuration space controllers, so the traditional PD and CT controller, both conduct the same circular path along the edge of the reachable area. The workspace controllers on the other hand choose a more straight path between initial and final position. The latter three controllers move more in a straight line towards the goal is because it does not care about the link's angle. The workspace controllers consider the position in the workspace only, so not requires the arm to be straight during the motion. The configuration space controller can be thought of as a spring between the current and desired joint angle, while the workspace control is described as a spring between the x- and y-coordinate of the current and the desired position.

When comparing the workspace controller trajectories only, there are still some differences to be observed. The CT controller has canceled out all non-linear behavior, which means it could move in an exact straight line between the initial and final position. The reason for being not completely straight is explained by the robot's geometry,

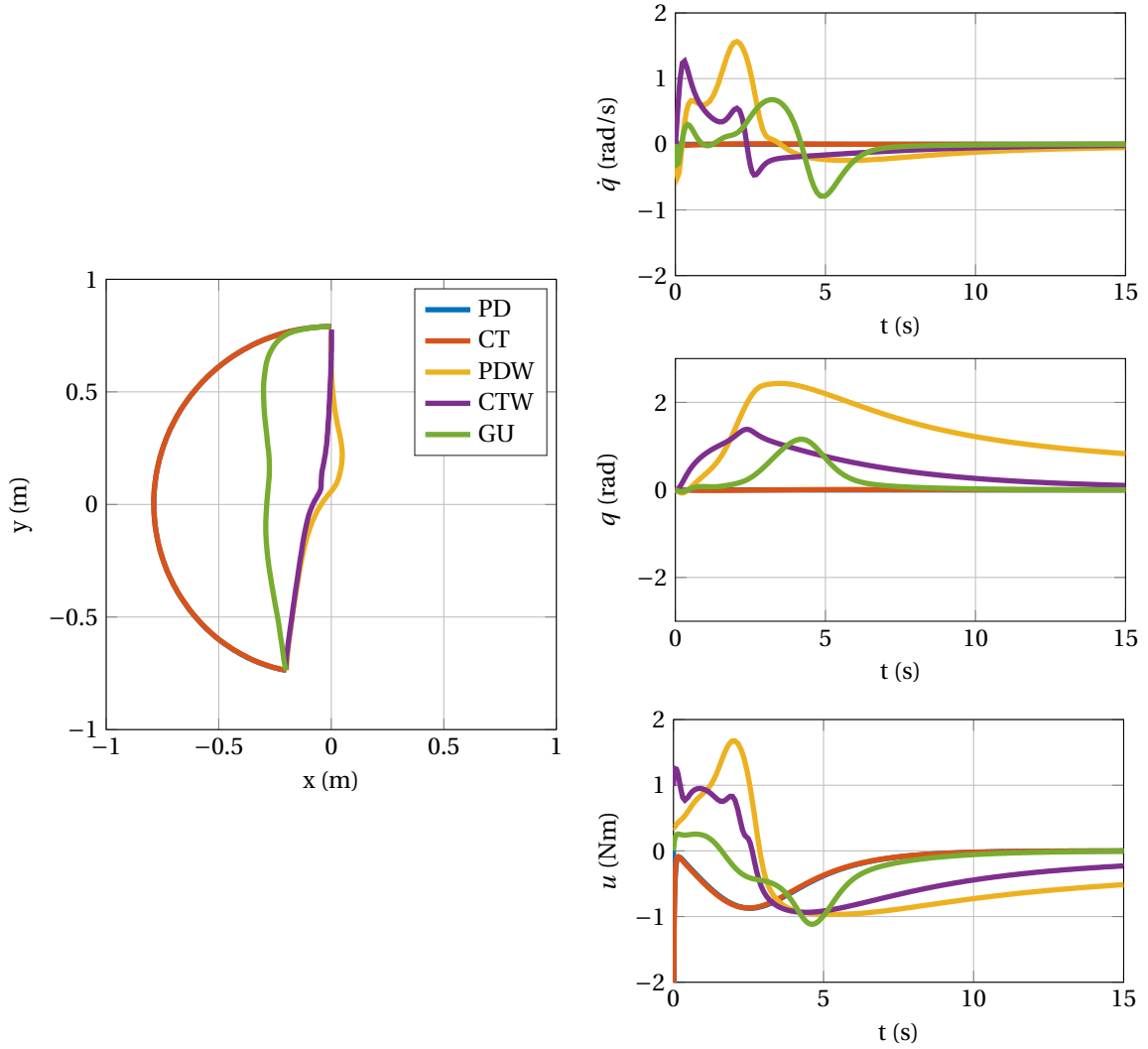


Figure 3.1: The simulation results of all five introduced controllers of the end-effector. The trajectories in the workspace, configuration velocities and coordinates and input torques differ for the different controllers. The end-effector starts at $[-0.20, -0.70]$ and moves to $[0, 0.72]$.

which is unable to reach the desired position on the straight line. The PD controller in the workspace contains some extra non-linearities, which results in more deviation from the shortest path. The gravity up controller pulls the end-effector upwards not considering its geometry, which results in the trajectory shown.

Besides, the angular velocity gives some information about the end-effector's progress. The largest difference can be found when comparing the configuration space with the workspace coordinate controllers. The end-effector does not seem to move quite much when controlled by the PD and CT controller, where only small velocities are concerned. From this initial angle of the end-effector equals the final angle, so it does not change

with respect to the second link. Nevertheless, the angular velocities using the workspace controllers are not equal to zero, so the angle between the second and third link differs over time.

These results are verified when looking at the end-effector's angular position plot, which varies significantly when considering the workspace with respect to the configuration space control. Besides, the gravity up controller starts moving the end-effector after a certain time, whereas the PDW and CTW controllers change this angle right from the start. The GU controller drives the links with largest gravity vector first, before moving the end-effector.

Finally, looking at the input torque it can be seen that the configuration space controllers are negative at first, because it keeps the third link straight with respect to the second link and so needs to compensate for the inertia. The input torque decrease to zero along with the rotating velocity of the system. The PDW and CTW controllers show more variation in the control input, because these controllers are responsible for rotating the end-effector. The GU controller input is close to zero at first until the other two links are close to their desired position, after which the third link is actuated.

3.2 Metrics analysis

Human-likeness is not easily captured and quite some research have made an attempt to mathematically generalize the human motion in only a few metrics. This section will introduce three new mathematical descriptions validating human-like movements based on previous work [16], [17], [23] and [24]. The metrics electrical power, gravitational torque and end-effector jerk and the results will be explained next.

3.2.1 Electrical power

The electrical power is a measure for the amount of effort the controller has put into the motion. This metric could be compared to the human's metabolic energy and the way they try to minimize their metabolic cost. Equation (3.9) represents the function for calculating the electrical power, depending on the controller's input voltage only

$$\begin{aligned} P_e &= U \cdot I \\ &= \frac{U^2}{R}. \end{aligned} \tag{3.9}$$

Here $R = 50 \, \Omega$, so that the current is a scaled version of the voltage between -1 A and 1 A, while U is bounded between -50 V and 50 V. The total electrical power can be calculated by integrating the electrical power at each instance over time as

$$P_{e,\text{tot}} = \sum_{n=1}^3 \left(\int_0^T P_{e,n}(t) dt \right). \tag{3.10}$$

The integration is equal to summing up the area underneath the electrical power curve, which MATLAB calculates using the function `trapz`.

Figure 3.2 shows the resulting electrical power curves for each motor attached to its link of the five different controllers, together with the total electrical power. The power corresponds to the moving up motions, like shown in Figure 3.1.

The figure shows some remarkable differences between the controllers in the way the power evolves over time. First of all, it should be noticed that link 1, the most inner link, has in general the most significant influence on the total power consumption. This could be explained by looking at the system matrices $M(q)$, $C(q, \dot{q})$ and $G(q)$, which are larger for this link. The second and third link are either directly or indirectly attached to the first link, which means their mass and inertia should also be added to the EOM for the first link. Because of these extra loads the motor of link 1 needs extra power to move to the desired position.

When comparing the electrical power curves between the controllers, Figure 3.2 shows some great difference between the configuration space and the workspace controllers. Especially the power in the motor of link 1 raises to about 4.5 W opposed to 1.5 W using the workspace controllers. It was seen earlier that the configuration space controllers move the arm from down to up having an almost straight arm, while the workspace controllers rotate the three links with respect to each other. This difference in configuration results in different loads on motor 1 during the motion, which are found higher for the PD and CT controller.

Like stated before, the total electrical power is calculated by integrating the power of each link, which is equal to the summed area underneath the curves. Higher values for the electrical power curves in the configuration space controller results in a higher total power as well. The titles of each graph shows the amount of total energy, from which it can be seen that the PD control in the workspace consumes the least power, followed by GU and CTW. In this case, the configuration space controllers' total electrical power is about 3 to 5 times larger than the workspace controllers.

Finally, it should be noted that the electrical power consumption for all controllers converges to zero when the manipulator reaches its desired position. This desired position is one of the equilibrium positions, which means no power is needed to keep the robot in its configuration, except for some compensation to keep it from falling down. If the desired position is not located in an equilibrium position it would need certain amount of torque to maintain the robot's particular configuration, resulting in steady-state electrical power greater than zero.

3.2.2 Gravitational torque

Besides looking at the power the motor produces, it may also be interesting to look at the gravitational torque only. Indeed these two metrics are coupled, but the latter

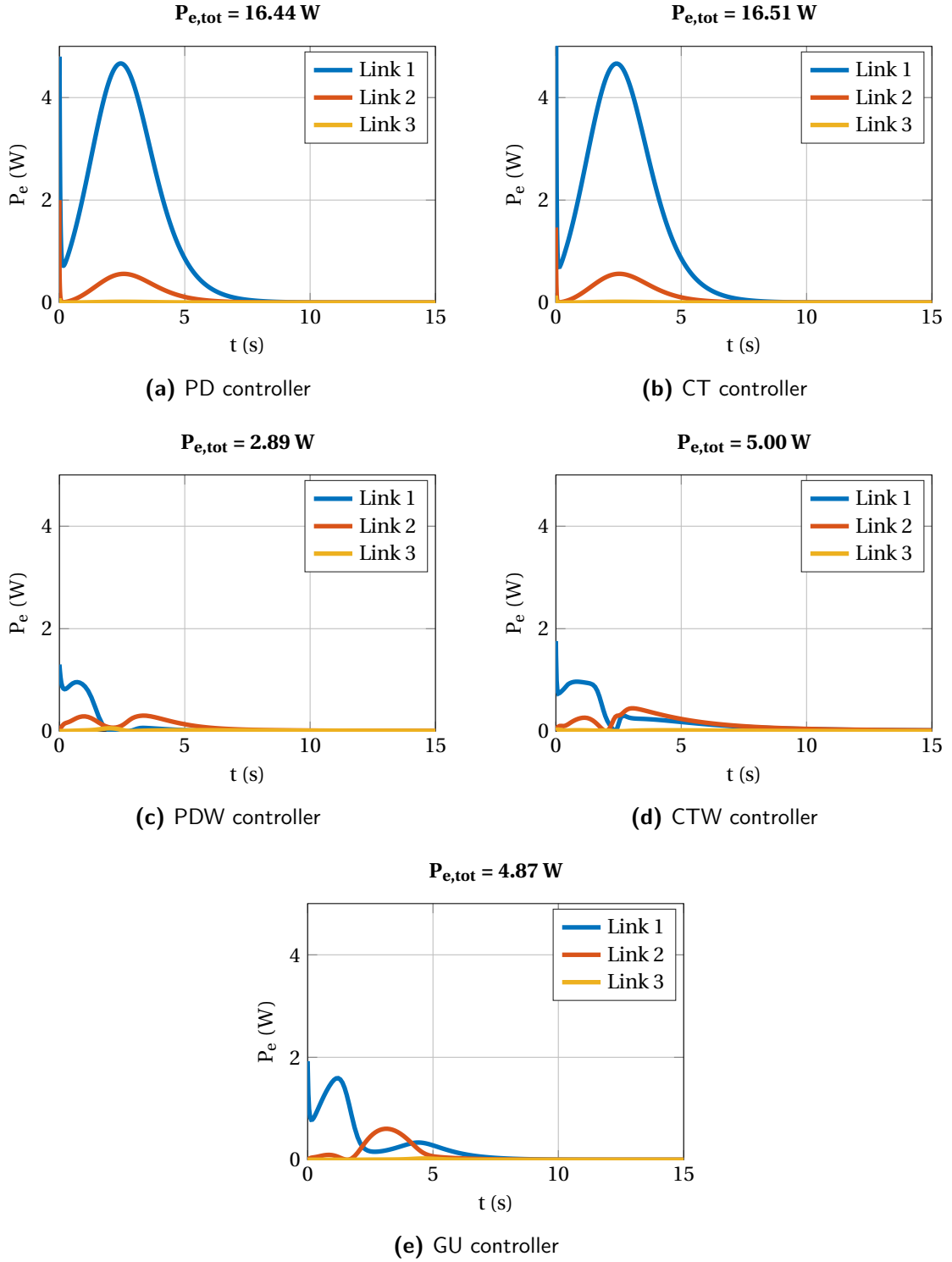


Figure 3.2: Electrical power curves corresponding to the three motors attached to the links using five different controllers. The integrated total electrical power is included in the title of each plot. The end-effector starts at $[-0.20, -0.70]$ and moves to $[0, 0.72]$.

one only considers the quasi-static case, while inertia and Coriolis terms are not taken into account. The idea behind looking at the gravitational torque comes from the way humans and animals try to minimize the amount of gravity they encounter when making certain movements [25].

To do so, the gravity vector $G(q)$ is calculated for all angles of the three links. The values are a measure for the amount of gravity pulling at the links. By taking the absolute sum of each instance, as shown in Equation (3.11) a map for each configuration of the robot can be created

$$G_{\text{tot}}(q) = \sum_{n=1}^3 |G(q)|. \quad (3.11)$$

Figure 3.3 shows the resulting landscape when keeping the third link straight with respect to the second link. Here higher values on the z-axis means the higher torque on the motors due to gravity. The horizontal distance of the center of mass to the motor along with the mass itself represent amount of gravity torque.

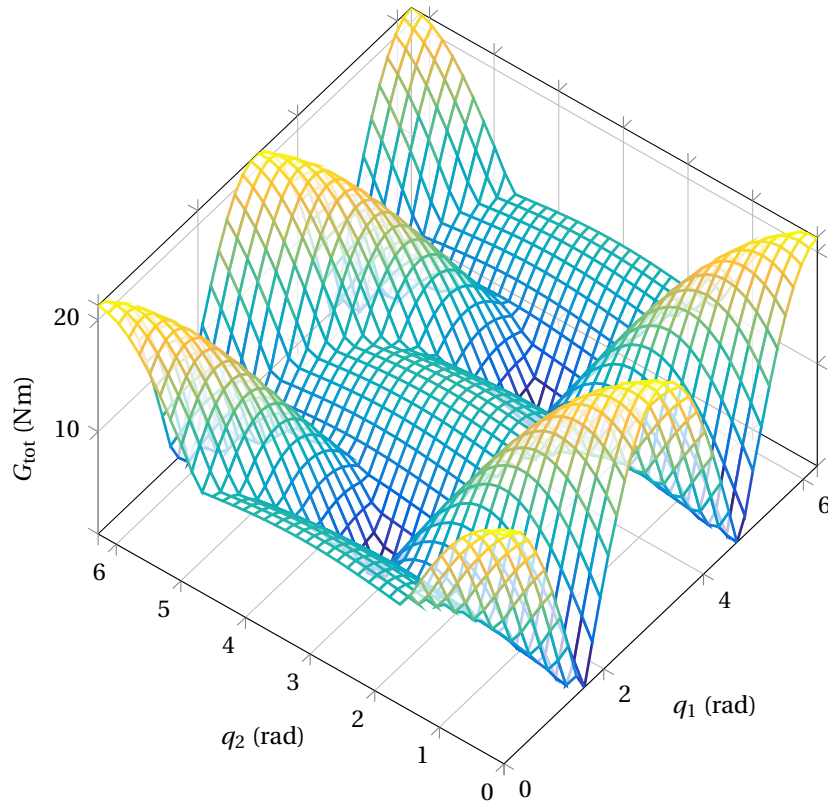


Figure 3.3: The gravity torque landscape calculated using the vector $G(q)$ for varying q_1 and q_2 , where $q_3 = 0$ rad. The peaks correspond to quite some torque on the motors, while zero represents no gravity pulling at all.

First of all, it should be noticed that angles of both links have values between 0 and 2π , which is a full circle, so the edges of the graph will continue on the other side of the graph. Furthermore, it can be seen that there are two high peaks at $[0, 0]$ and $[\pi, 0]$

rad, two lower peaks at $[0, \pi]$ and $[\pi, \pi]$ rad and four equally low valleys at $[\frac{\pi}{2}, 0]$, $[\frac{\pi}{2}, \pi]$, $[\frac{3\pi}{2}, 0]$, and $[\frac{3\pi}{2}, \pi]$ rad. As can be constructed from Figure 2.1, the two high peaks correspond to the configuration where the robot is fully extended in horizontal direction and the valleys to one of the pointing down or up equilibrium positions. The two lower peaks correspond to the configuration where both links are horizontal, but the second link is pointing inwards. As a result, the center of gravity is closer to the pivoting point and therefore the gravity torque is lower.

The reason for only showing q_1 and q_2 , is that these two angles have the most significant effect on the gravity torque value. Figure 3.4 shows the influence of q_3 on the gravity torque when q_1 is kept at a constant angle.

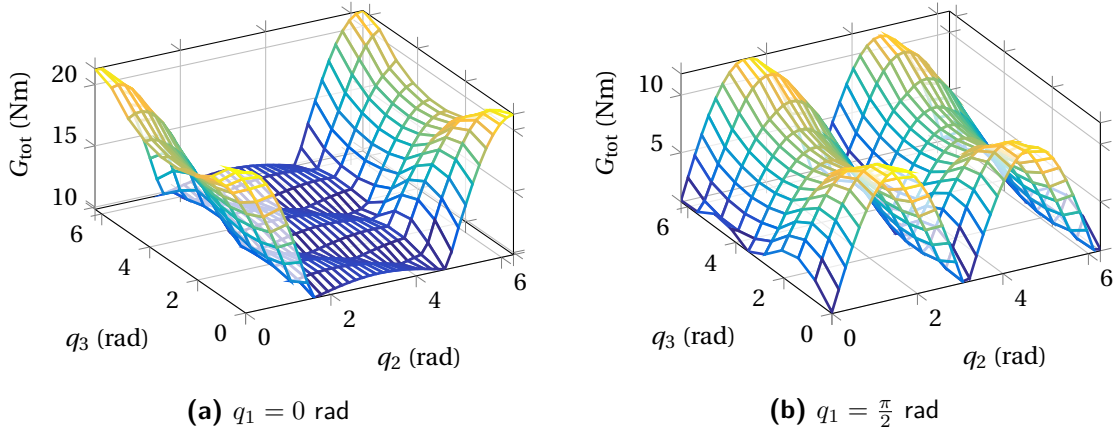


Figure 3.4: The gravity torque landscape calculated using the vector $G(q)$, with q_1 kept at a constant angle. The influence of varying q_3 seems less significant on the overall gravity torque.

When looking at a constant angle q_2 , so at a line perpendicular to this axis, it can be seen that the location of the peaks and the valleys stay approximately at the same place. As a result, even though the gravity torque value changes, the least torque path is not influenced. Therefore, the landscape shown in Figure 3.3 is considered as the most significant when presenting the influence of each controller on this metric.

The gravity torque landscape could also be converted into a contour map, so that the controlled motion can be visualized better. Figure 3.5 shows the trajectory of all five different controllers on the gravity torque contour plots. The color bar shows the correspondence to certain levels of the gravity torque, the blue and green circle are the initial and final position respectively and the red crosses represent the trajectory of q_1 with respect to q_2 during the movement.

Some remarkable differences can be obtained from these figures. First of all, it can be seen that both the PD and CT controller are going up from initial position to cross the peak at $[\pi, 0]$. This trajectory seems to be the shortest when considering the robot's

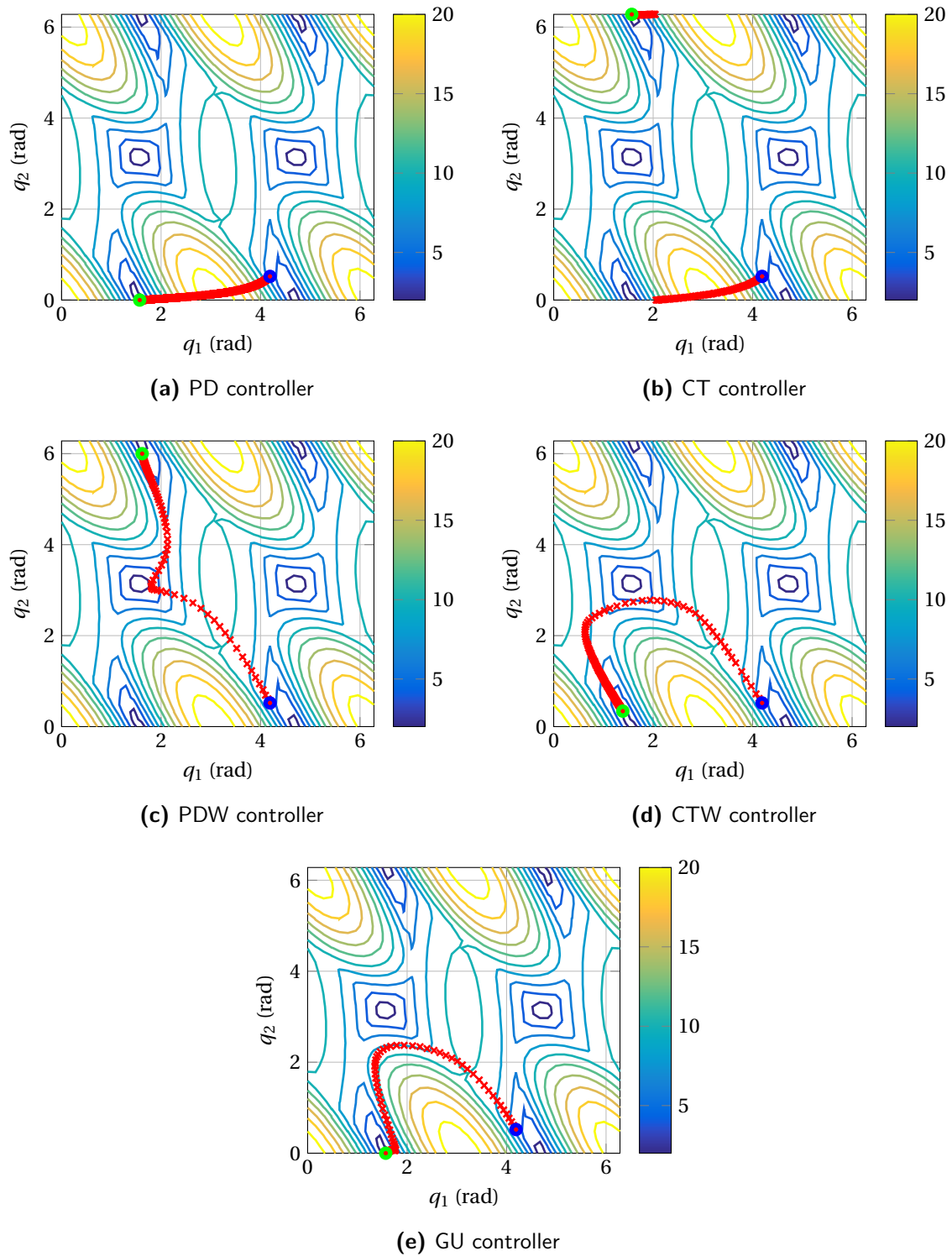


Figure 3.5: Contour map of the gravity torque for varying q_1 and q_2 values. The blue and green circle are the initial and final position respectively and the red crosses represent the traveled path in terms of configuration coordinates.

motor angles, but it will cost quite an amount of power to cross this high gravity torque configuration.

The other three controllers are not taking the motor angles into consideration to achieve the goal, which results in different trajectories as seen in Figure 3.5. The PDW controller makes the second link rotate almost a full circle with respect to the first link. The desired position is located at $q_1 = \frac{\pi}{2}$ rad and $q_2 = 0$ rad, which could, like explained earlier, also be reached by crossing the $q_2 = 2\pi$ rad line. Furthermore, the PDW controller shows that it converges to the least gravity torque configuration at $(\frac{\pi}{2}, \pi)$ rad, before moving towards the desired final position.

The remaining two controllers have about the same trajectory on the (q_1, q_2) -map. Both CTW and GU controller drive the second link to the least gravity torque position, after which it maintains low towards the final position. The only difference is that GU stays on a slightly higher contour level with respect to the CTW controller, which almost entirely reaches the minimum torque level.

It could also be quite interesting to compare the gravity torque evolving over time for all five controllers. Figure 3.6 shows this trajectory, which is basically the red cross of Figure 3.5 with corresponding height over time.

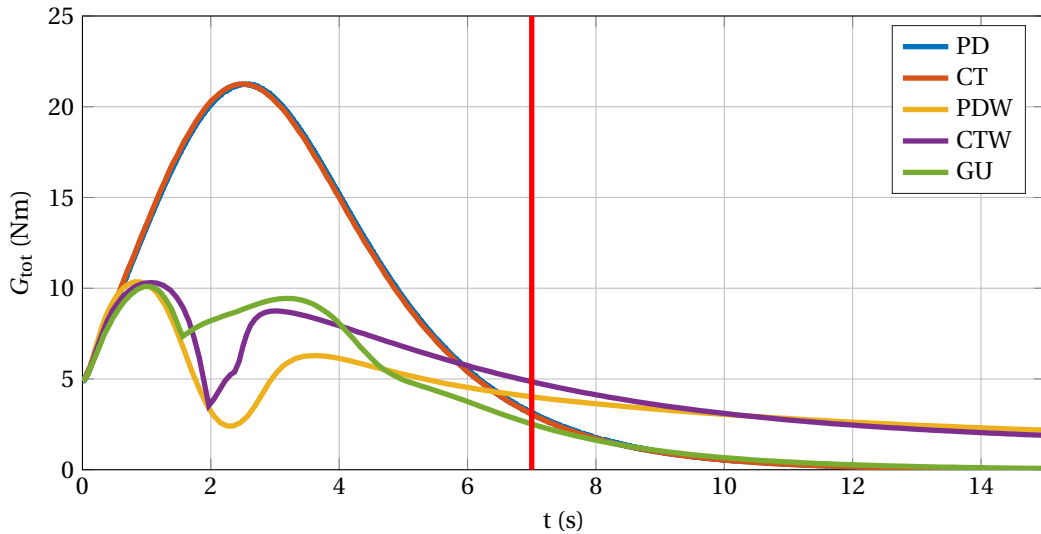


Figure 3.6: The gravity torque curves for the five different controllers over time. The line at $t = 7$ s represents the moment where the robot gets very close to the desired position.

The same results as discussed above can be interpreted here. The configuration space controllers show the highest maximum values, while they are crossing the peak gravity torque. These trajectories have steady increase and decrease towards the final position with zero gravity torque.

The other three controllers went through a valley as shown in their trajectory, which correspond to the lower gravity torque levels on the contour map. It can be seen that

the PDW controller has a smooth trajectory, while CTW and GU show a sharp edge when reaching the minimum value. An explanation is that at these points in time the input torques switches sign, as shown in Figure 3.1, which causes the non-smoothness of the gravity torque to happen.

Furthermore, it can be seen that the PDW and CTW controllers are not converged to zero gravity torque at the end the motion. The reason for this is that these controllers are not considering the end-effector's angle. It was found that the final position of these two controllers is not standing up straight, but is slightly tilted. Therefore, the moment arm of the center of mass is greater than zero, which causes a steady-state gravity torque.

When calculating the total gravity torque as a measure for this metric, the problem with this steady-state value will have great influence on the total value. Like explained earlier, the motions are normalized such that the end-effector reaches the circle with radius 0.1 meters around the desired final location at time equals 7 seconds. Therefore, when looking at the motions towards this circle only, the gravity torque values after $t = 7$ s should be omitted. Table 3.2 shows the total gravity torque and the gravity torque during motion for all five controllers. It can be seen that the especially the PDW and CTW controller's torque values differ when focusing on motions only.

Table 3.2: The total gravity torque calculated by the area underneath the curves. The moving gravity torque is a clipped version of the total, where all values above $t = 7$ s are omitted.

	Gravity torque (Nm)	
	Total	Moving
PD	94.1	88.8
CT	93.8	88.7
PDW	63.2	40.5
CTW	74.3	51.0
GU	53.9	48.5

The total gravity torque column of Table 3.2 demonstrates that the gravity up controller uses the least gravitational torque to reach the final position, followed by PDW and CTW respectively. The last column switches the ranking of the GU with the PDW controller, which suggests it would be better to use PDW when trying to minimize the gravity torque while moving.

3.2.3 End-effector jerk

Lastly, the jerk in the end-effector is considered as a third metric for defining human-like motion in robotics. This metric is chosen because it is used as smoothness representation of motion. According to [16] minimizing the hand's jerk in workspace coordinates,

as shown in Equation (3.12), will increase naturalness. For analyzing purposes it is sufficient to calculate the jerk after the simulation is completed by differentiating the resulting acceleration

$$\frac{d^3r}{dt^3} = \sqrt{\left(\frac{d^3x}{dt^3}\right)^2 + \left(\frac{d^3y}{dt^3}\right)^2}. \quad (3.12)$$

Figure 3.7 shows the end-effector's jerk curves for all five controllers. Here the jerk in both the x- and y-direction is shown along with the total calculated jerk using Equation (3.12). The sign of the first two mentioned corresponds to the jerk's x-, and y-direction. The total jerk is a measure for the amount of jerk at each instance, so always greater than zero.

It can be seen that all controllers have an initial peak at $t = 0$ s. The manipulator initial velocity is equal to zero, so in order to start moving it will need to accelerate. This increase of acceleration to a non-zero value, means that the jerk needs to be greater than zero. The height and the length of the initial peak differs among the controllers, while the end-effector could accelerate differently.

Besides, the figure clearly shows the differences between the configuration space, workspace and gravity up controllers. The configuration space controllers, PD and CT, seem to have its initial peak only and minimum jerks during the rest of the motion. As a result, the acceleration does not change the motion has started. The acceleration of the end-effector in the workspace confirmed this, because is indeed almost equal to zero after initiation of the movement.

The workspace controllers demonstrate some extra peaks, which correspond to acceleration changes and therefore sudden velocity changes in the movement. PDW shows two smaller peaks, while CTW contains one higher and longer peak in the total jerk. Therefore, these two controllers result in a more abrupt motions when moving the arm to the upward position.

The gravity up controller consists of less and lower peaks than the two workspace controllers. On the other hand, the peak is still higher than the configuration space controllers. The jerk peak is lower and wider, so the end-effector accelerates and decelerates more gradually.

When the objective is to minimize the amount of total jerk it would be the best option to choose a configuration space controllers. The results are smoother, so less sudden and abrupt movement changes when executing the tasks.

3.3 Discussion

This section will discuss the results of the metrics for the five feedback controllers as shown in Section 3.2 and recommends the most human-like controller based on these

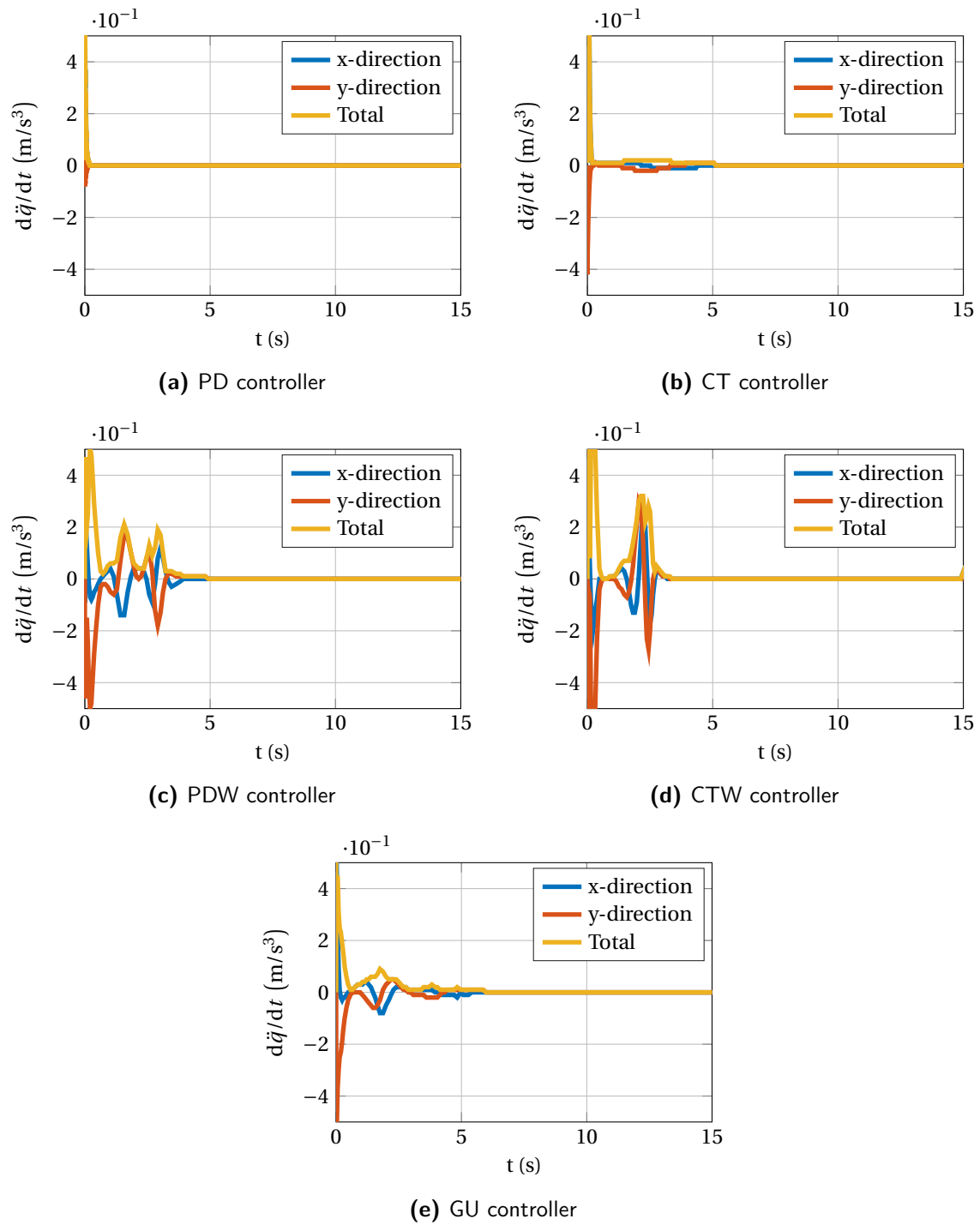


Figure 3.7: The jerk of the end-effector in both x- and y-direction together with the overall jerk for all five controllers.

results.

Traveled trajectory Figure 3.1 showed the different paths the end-effector is taking when moving to the upright position. The configuration space controllers resulted in a circular-shaped trajectory, so the 3-link manipulator is completely extended during the motion. The workspace controllers, on the other hand, showed a more direct and straight path towards the goal. The robot's joint angles keeps changing to guide the third link on this straight path. Finally, the gravity up controller moves straight up until it reaches its maximum stretch after which it converges to the desired final position.

According to [16] the human hand tends to move along a straight line towards the goal. When only considering the trajectory and the shortest distance traveled by the end-effector it should be concluded that the PDW and CTW controllers are both the most human-like way of moving, followed by the GU controller.

Angular constraint The angular position of the third link, on the other hand, contradicts this conclusion, as shown in Figure 3.1. Especially the PDW controller increases to almost $\frac{3}{4}\pi$ rad, which means the end-effector and the second link are almost parallel, but pointing in opposite direction. The human hand will not be able to rotate that far with respect to the fore arm. The computed torque controller in the workspace and the gravity up controller produces a bit lower end-effector angles with a maximum of about $\frac{\pi}{2}$ rad. Like stated earlier, the PD and CT controllers keep the third link in a straight line with the second link, so no violation of any constraints.

The conclusion to be drawn here is that, in order to move in a human-like way, the angular positions should be constrained. It is assumed that these constraints will not change the order of the human-like movements when considering the trajectories as the only measure.

Electrical power Next, the electrical power metric showed some striking differences when comparing the controllers as in Figure 3.2. It was found that the configuration space controllers produce much more electrical power, compared to the workspace controllers and the GU controller. The former two controllers reach values larger than 4 W for link 1, while the other three controllers keep the power respectively low. As a result, the PDW controller has the lowest total power, followed by GU and CTW.

Several papers [17] state that humans are minimizing the metabolic cost when moving. According to this statement, when looking at the electrical power, the PDW controller performs the best and therefore the most human-like.

Gravity torque The most elaborated metric is the gravity torque, which says something about the gravity weighting on each joint. Figure 3.5 presented the two links evolving over the gravity torque for all five controllers. It can be seen that the configuration space controllers move directly over the peak value, which results in the highest overall gravity torque. The workspace controllers converge to the valley, before moving on to

the desired location. The GU controller does not entirely converges to the valley, but keeps a constant low value around the peak.

The different final configurations caused the PDW and CTW controller's gravity torque to be much higher than the GU controller. Figure 3.6 and Table 3.2 presented the difference between the total and moving gravity torque, which is a clipped version of the total.

Looking at these trajectories and assuming that humans tend to minimize gravity torques during movement [26], it can be concluded that the PDW controllers corresponds to the most human-like motion with the least gravity torque. The GU controller seems to keep a slightly higher level and is therefore not ranked as best choice.

Jerk Finally, the jerk was explored as a metric for measuring the human-like way of moving. According to [16] and [27] the end-effector's jerk minimization will maximize joint smoothness. As can be seen in Figure 3.7 the jerk is specifically noticeable when looking at the workspace controllers PDW and CTW. The configuration space controllers are almost entirely equal to zero, except for the initial peak.

Therefore, it is concluded that the PD and CT controllers perform the smoothest joint motions, while these controllers contain the lowest overall jerk.

In the discussion above several conclusions were drawn from the observations, as summarized briefly in Table 3.3. The plus and minus signs correspond to whether the controller had a positive or negative contribution to the human-likeness according to the metric. Zero means, no real conclusion about the metric for the controller was drawn.

Table 3.3: Summary of the compared metrics for each controller.

	Trajectory	Angular constraint	Electrical power	Gravity torque	Jerk
PD	–	+	–	–	+
CT	–	+	–	–	+
PDW	+	–	+	+	–
CTW	+	–	+	+	–
GU	0	–	+	+	+

It is noticeable that some of the metrics contradict each other, so making a final decision is not straight forward. When looking at the traveled trajectories, minimum electrical powers and gravity torques both the PDW and CTW controller are considered to be most human-like. Even though the CTW controller has the same score as the PDW controller, it contains slightly higher values for both the total electrical power and gravity torque. The angular constraints metric scores the best at the configuration

space controller PD and CT. Fortunately, this problem could be solved by including joint limit constraints in the control law, which forces all controllers to stay within the reachable area. When only looking at the latter three metrics the best choice would be the GU controller. The major disadvantage of this controller however is that it does not perform its motions in a straight line towards the goal. Therefore, despite the negative ranking for the jerk metric, the PDW controller is chosen to be the most human-like. Following from this conclusion, the PDW controller will be used to control the robot's motion during the rest of the research.

3.4 Synopsis

The goal of this chapter was to find the most human-like motion metric and analyze and compare several basic feedback controllers. Feedback here is crucial, while humans are able to move in any direction without having to learn each of the in-between configurations.

Five feedback controllers introduced all using the error between current and desired position and the velocity in different coordinate frames to generate an input for the robot. A generalization of these controllers showed that each controller consists of an energy insertion, an energy extraction and a compensation part.

The comparison between the controllers demonstrated the clear differences in the robot's angular positions and velocities, as well as the conducted trajectory of the end-effector.

Next to that, three extra metrics were introduced to find the most human-like controller among the proposed control laws. Electrical power keeps track of the amount of power the joint motors need to supply to reach the desired position. The gravitational torque metric considers the torque each link exerts on the shoulder joint, using its mass and relative distance to the shoulder. Finally, the end-effector jerk bears in mind the smoothness of the movement.

The results showed that the PD controller in the workspace performs the best when considering the end-effector trajectories, electrical power and gravitational torque. The maximum angular positions were exceeded, but could be solved by including the joint limits into the control law. In conclusion, this controller performs the most human-like motions according to the presented metrics and will therefore be used during the remaining of this work.

General control framework

In Chapter 3 it was concluded that according to the end-effector trajectory, the electrical power metric and the gravity torque metric the PD controller in the Workspace is considered to generate the most human-like motion. In order to improve the human-like motion even further, these metrics will be combined with the PDW controller. This leads to the introduction of the Operational Space Controller approach.

Furthermore, it was stated before that the maximum angular position of the end-effector also influences the interpretation of human-likeness. In order to constraint this angle within reasonable limits, joint limit control is included in the control-law.

Due to the introduction of these joint limits the robot could get stuck into local minima, after which it is not able to move any further. These local minima are handled using an additional part, which will be called the polar angle control in this work.

4.1 Operation space control with gravity

Operational Space Control (OSC), as introduced by [28] in 1987, could be explained as a control approach with different priorities between the different control tasks. This hierarchical order allows us to introduce additional tasks, without violating the main task of for instance reaching the desired location. This section will first introduce the theory behind OSC, after which the implementation for current research is given.

4.1.1 Introduction to OSC

To be able to create the hierarchical structure in the control law, the OSC uses the null space of the task control to handle additional tasks, like controlling the robot's posture. As an example look at the EOM of Equation (4.1), with the control input u representing the PD controller as given in (4.2)

$$A(q)\ddot{q} + B(q, \dot{q})\dot{q} + G(q) = u, \quad (4.1)$$

$$u = J_{\text{task}}^T (K_p e + K_d \dot{e}). \quad (4.2)$$

The extra task could be added, like keeping the overall center of gravity the closest to the body, so that the resulting control law looks like

$$u = J_{\text{task}}^T (K_p e + K_d \dot{e}) + J_{\text{pos}}^T K_p e_{\text{pos}}. \quad (4.3)$$

It could be imagined that these two control tasks contradict each other, which results in a steady-state error with respect to the goal task. To overcome this problem, the null space of the task control should be computed, so the secondary task will only be effective when not interrupting the primary task. The improved control law of Equation (4.3) is

$$u = u_{\text{task}} + N_{\text{task}}^T u_{\text{pos}}, \quad (4.4)$$

where u_{task} and u_{pos} represent the two parts in Equation (4.3).

The null space is composed as shown in Equation (4.5), which could be seen as the available degrees of freedom not violating the goal task

$$N_{\text{task}}^T = I - J_{\text{task}}^T \bar{J}_{\text{task}}^T. \quad (4.5)$$

The \bar{J}_{task}^T is called the dynamically consistent generalized inverse of J_{task}^T , derived from the task's Jacobian J_{task} and the inertia matrix $A(q)$ (Eq. (4.6), calculated as

$$\bar{J}_{\text{task}}^T = A^{-1} J_{\text{task}}^T \left(J_{\text{task}} A^{-1} J_{\text{task}}^T \right)^{-1}. \quad (4.6)$$

This forms the basis of the OSC approach, but more information about the exact procedure can be found in [28] and [29].

4.1.2 Implementation of OSC

The goal now is to use the previously described control framework to include human-like metrics within the general control law. The assumption is that human-likeness would be increased, when the degree of freedom is large enough such that the null space is available. As stated earlier the PDW controller was found to conduct the most human-like motions, according to the chosen metrics. Therefore, this controller will be regarded as the basic framework. The control structure implemented next will roughly look like as shown in Figure 4.1.

The question remains which metric should be chosen and especially why. Let us have another look at the metrics described earlier.

First of all, the path of the end-effector was investigated, which resulted in certain degree of directness. Looking more carefully it should be observed that the path straightness cannot be captured in a metric, but describes the results of the control law specifics. Therefore, it should be kept in mind when evaluating the results, but will not be included within the controller.

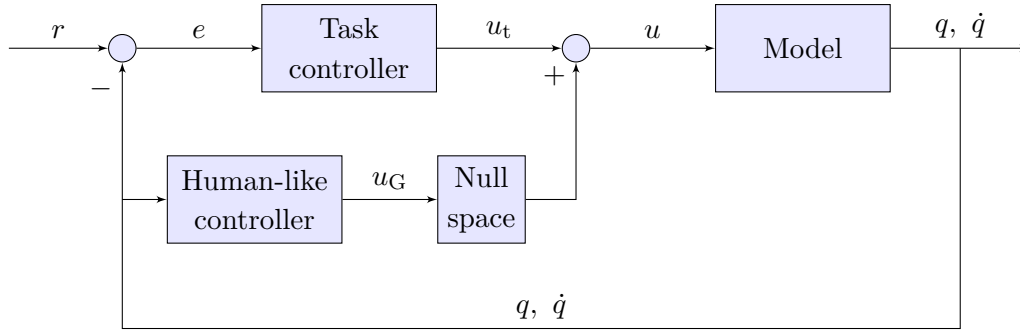


Figure 4.1: The control structure of the Operational Space Controller with different priorities among the different goals. The task controller moves the end-effector to the desired position and the human-like controller takes the robot's path and configuration into account as a second priority.

The total electrical power could be calculated using the sum of the integrated electrical powers at each element. In order to conclude anything about this metric it could be imagined that information over the entire trajectory needs to be known. As a result, an optimization is required, which contradicts the philosophy of creating a feedback controlled system to convey information non-verbally.

Next discussed was the gravity torque metric. The landscape, as seen in Figure 3.3 of Section 3.2.2, was build using the quasi-static gravity torque $G(q)$. The gravity map is calculated beforehand using the configurations of the robot. This approach does not involve an optimization approach, and is therefore qualified to be used in the feedback system.

Finally, the jerk metric was mentioned, which is calculated as the second derivative of the end-effector's velocity. It was concluded that the PDW controller is not evaluated human-like according to this metric. While this contrasts the observations seen at the other metrics, the conclusion should be drawn to discard this metric for now.

So, the gravity torque metric seems to be the best suited to combine with the PDW controller approach within the OSC framework. The overall task is to reach the desired location with the end-effector. The extra task will involve minimizing the gravity torque fighting resulting in more human-like motions.

In order to make sure the robot reaches its desired position, the OSC framework constructs the secondary task within the null space of the primary task. The null space is calculated as shown in Equation (4.5), where the task Jacobian is the partial derivative of $g(q)$ to q of Equation (2.11). As explained earlier the vector $g(q)$ is representing the mapping of the configuration coordinates q to workspace coordinates x , y and θ , provided in Equation (2.9).

The gravity torque map as seen in Section 3.2.2 is repeated here in Equation (4.7),

which will represent the secondary task

$$G_{\text{tot}}(q) = \sum_{n=1}^3 |G(q)|. \quad (4.7)$$

In order to formulate this in the control law and prevent it from chattering in the local minimum of the gravity torque map, the calculations is slightly changed as

$$u_G = J_G^T(q) K_G \text{Sgn}_G(q) G(q). \quad (4.8)$$

The sign function $\text{Sgn}_G(q)$ replaces both the sum and the absolute signs by using a vector multiplicity and smooth transition between -1 and 1 around zero as shown in Figure 4.2. The function having this figure as a result is

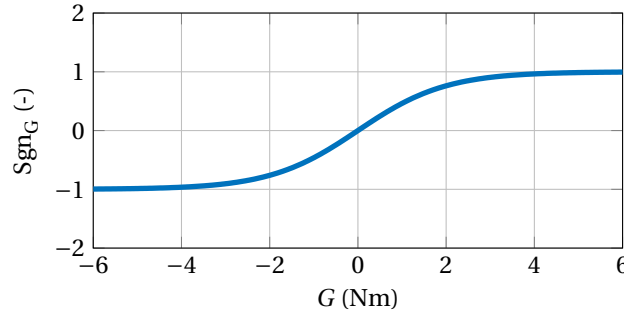


Figure 4.2: Smooth sign function of $G(q)$, such that multiplication of this sign function with $G(q)$ will always result in positive value.

$$\text{Sgn}_G(q) = \frac{2}{1 + e^{-cG(q)}} - 1, \quad (4.9)$$

which is depending on the gravity torque function $G(q)$ to create the smooth transition. Furthermore it should be mentioned that the each element of the gravity torque is multiplied by a different constant c , because the maximum gravity torque of each element differs.

The equation given in (4.4) will now be substituted by the minimal gravity torque control. The Jacobian of this controller part is needed to map the new coordinate frame back to the configuration coordinate frame. The Jacobian can be calculated by simply taking the partial derivative of the entire control part to the configuration coordinates q , so

$$J_G(q) = \frac{\partial (\text{Sgn}_G(q) G(q))}{\partial q}. \quad (4.10)$$

In order to make sure that the task is reached an extra constraint is added

$$||X - X_{\text{des}}||,$$

which makes the second part disappear when getting closer to the desired position. We are allowed to do this because the gravity torque control is especially influencing the configurations when moving. This extra constraint ensures there is no steady-state error of the end-effector. Adding all described elements together results in the human-like controller

$$u_G = \|X - X_{\text{des}}\| J_G^T(q) K_G \text{Sgn}_G(q) G(q), \quad (4.11)$$

which uses the gravity torque map to minimize the fighting gravity effort.

The implementation of the PDW controller in the OSC framework, with an extra control law influencing the gravity torque, results in

$$\begin{aligned} u &= u_{\text{task}} + N_{\text{task}}^T u_G, \\ u &= J^T(q) (K_p e_r + K_d J(q) \dot{q}) + G(q) \\ &\quad + N_{\text{task}}^T (\|X - X_{\text{des}}\| J_G^T(q) K_G \text{Sgn}_G(q) G(q)). \end{aligned} \quad (4.12)$$

In order to show the influence of the just proposed control framework, the results of a motion without and with the use of the Operational Space Control is given in Figure 4.3. The conducted motions simply starts with the arm along the robot's body at $[0, -0.72]$ and ends with the arm pointing about 45° in front of him so $[0.5, 0.5]$. The figure shows the movement trajectories on the left and gravity torque trajectories on the right for the motion without gravity torque control, with the OSC control and with OSC including the extra constraint $\|X - X_{\text{des}}\|$.

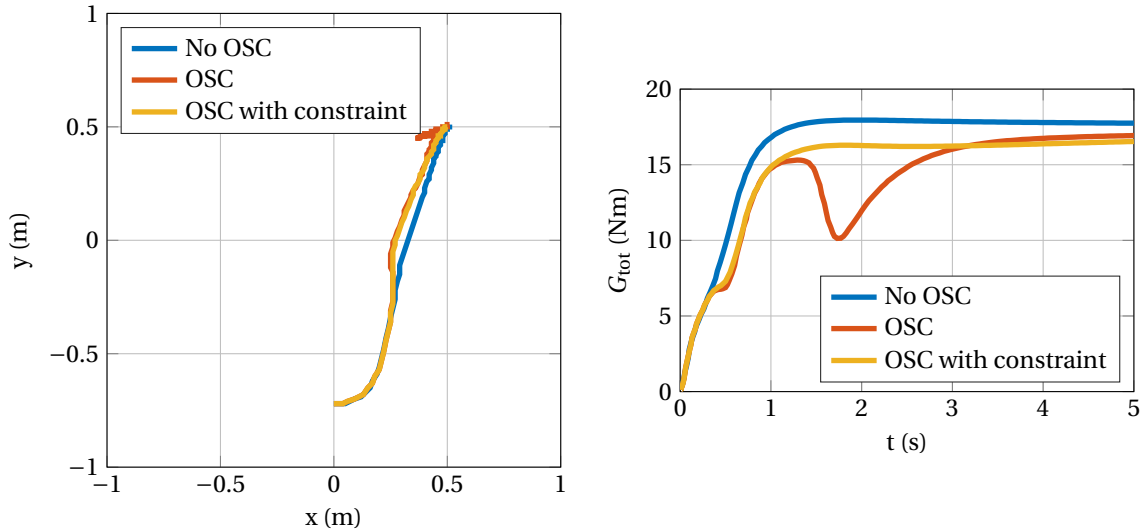


Figure 4.3: Comparing the motion starting at $[0, -0.72]$ and moving towards $[0.5, 0.5]$ without gravity torque control, with OSC and with OSC including the extra constraint $\|X - X_{\text{des}}\|$. Both the trajectories in the workspace and the gravity torque trajectories are given.

There are a number of differences to be observed from this figure, from which the conclusion could be drawn whether the proposed control law satisfies the demands or

not. First of all, when comparing the controller without gravity torque control with the ones with, it is seen that the former one has the shortest path towards the goal, which seems to be the most efficient. When looking on the other hand at the gravity torque it should be noticed that the gravity torque is actually the highest, which means it needs to fight gravity the most.

Comparing the latter two motions, so one without and one with the additional constraint, it is found that the lowest gravity torque is measured when not using the constraint $\|X - X_{\text{des}}\|$. The main disadvantage of this control law is that an awkward bump is recorded when getting closer to the the goal. It moves away from the goal and gets back in a different configuration. These results were the motivation to include the constraint, which indeed increases the level of human-like motion as well.

4.2 Joint limits

The movements described in previous section have not taken into account the limited range of motion for each of our joints. Therefore the movements could be evaluated awkward and not human-like. In order to exclude these configurations, joint limit constraints are included. The approach is based on [30], but here the limits will be include on each joint.

The basic idea is to create a so-called activation function, which activates the control law when the joint limits are not reached. The control law is deactivated when the configuration coordinates reached the undesired positions. Furthermore, the activation function will also be used to push the arm back into the reachable area, so that it is able to move again. The control law of Equation (4.12) is adjusted as

$$u = (1 - |H_{\text{jl}}(q)|) (u_{\text{task}} + N_{\text{task}}^T u_{\text{G}}) + K_{\text{jl}} H_{\text{jl}}(q), \quad (4.13)$$

where $H_{\text{jl}}(q)$ represents the activation function. This function is equal 0 when the joint angles are not close to its limit. The function is either -1 or 1 when the minimum or maximum joint limit has been reached respectively. In order to create smooth movements when the arm is getting closer to the joint limits, a gradual transition is used from 0 to either -1 or 1. These functions are called sigmoid functions, and could be defined in multiple ways. Here it was chosen to sum up two hyperbolic tangent functions $\tanh(x)$ as

$$H_{\text{jl}}(q) = \frac{1}{2} (\tanh(q - q_{\text{min}}) + \tanh(q - q_{\text{max}})). \quad (4.14)$$

Figure 4.4 shows the activation functions $H_{\text{jl}}(q)$ for each of the robot's link. The joint limits are set at $q_1 : [-\pi, \frac{3}{4}\pi]$, $q_2 : [-\frac{1}{12}\pi, \frac{11}{12}\pi]$ and $q_3 : [-\frac{1}{4}\pi, \frac{1}{2}\pi]$, based on the reachable joint limits of the human arm [22], where again the $(0, 0, 0)$ position corresponds to the arm pointing horizontally to the front of the body.

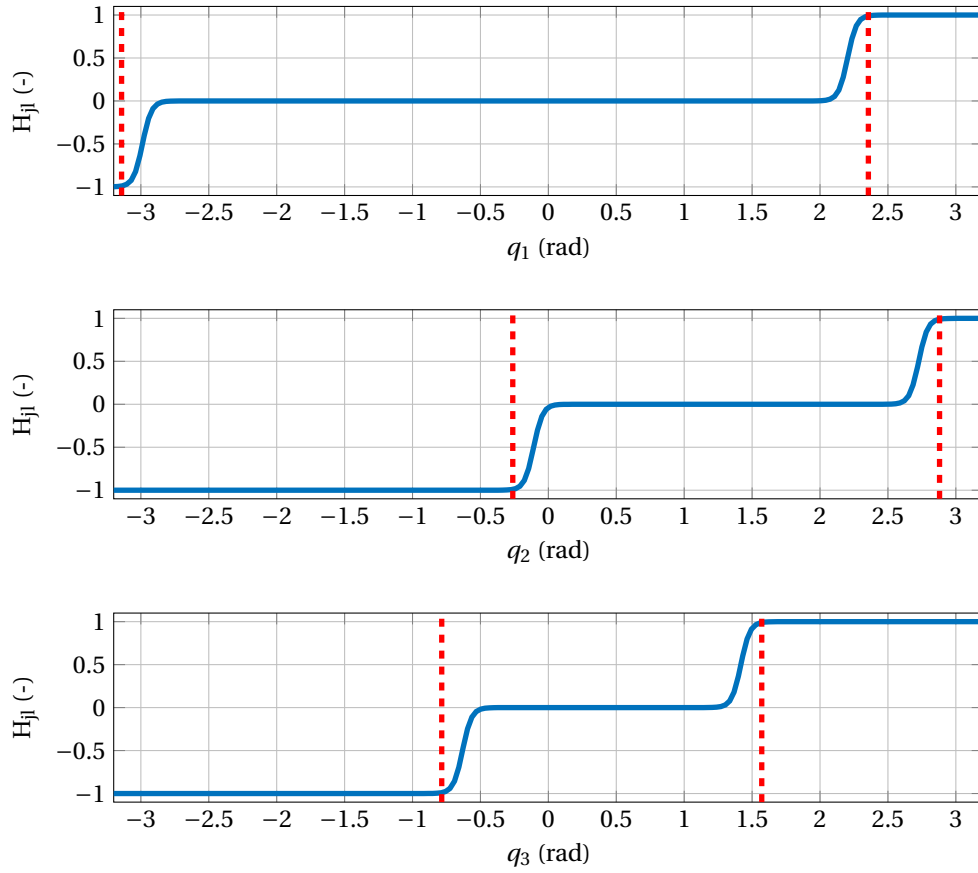


Figure 4.4: The joint limit functions for each link depending on its angular position. Values equal to zero represent configurations where the limits are not reached. The robot is bounded by $q_1 : [-\pi, \frac{3}{4}\pi]$, $q_2 : [-\frac{1}{12}\pi, \frac{11}{12}\pi]$ and $q_3 : [-\frac{1}{4}\pi, \frac{1}{2}\pi]$.

The results of the new control law taking into consideration the joint limits can be compared to the previous results without joint limits as found in Figure 4.5. The figure on the left shows that the new situation is still able to reach the desired location, with a slightly changed path of the end-effector. On the right it can be seen that the situation without joint limits the third joint crosses its limit around $t = 1.2s$. Besides, the second joint gets really close to reaching undesired configurations in the elbow joint. The situation with joint limits is able to overcome the crossing problem of the third joint and presents the elbow joint from reach risky configuration.

4.3 Polar angle

The control law as described earlier seems to work quite well, but there is one major issue which needs to be solved. Due to the implementation of the joint limits the robot could get stuck in local minimum. As a result it will not be able to reach the desired position anymore. The general control framework should be able to conduct

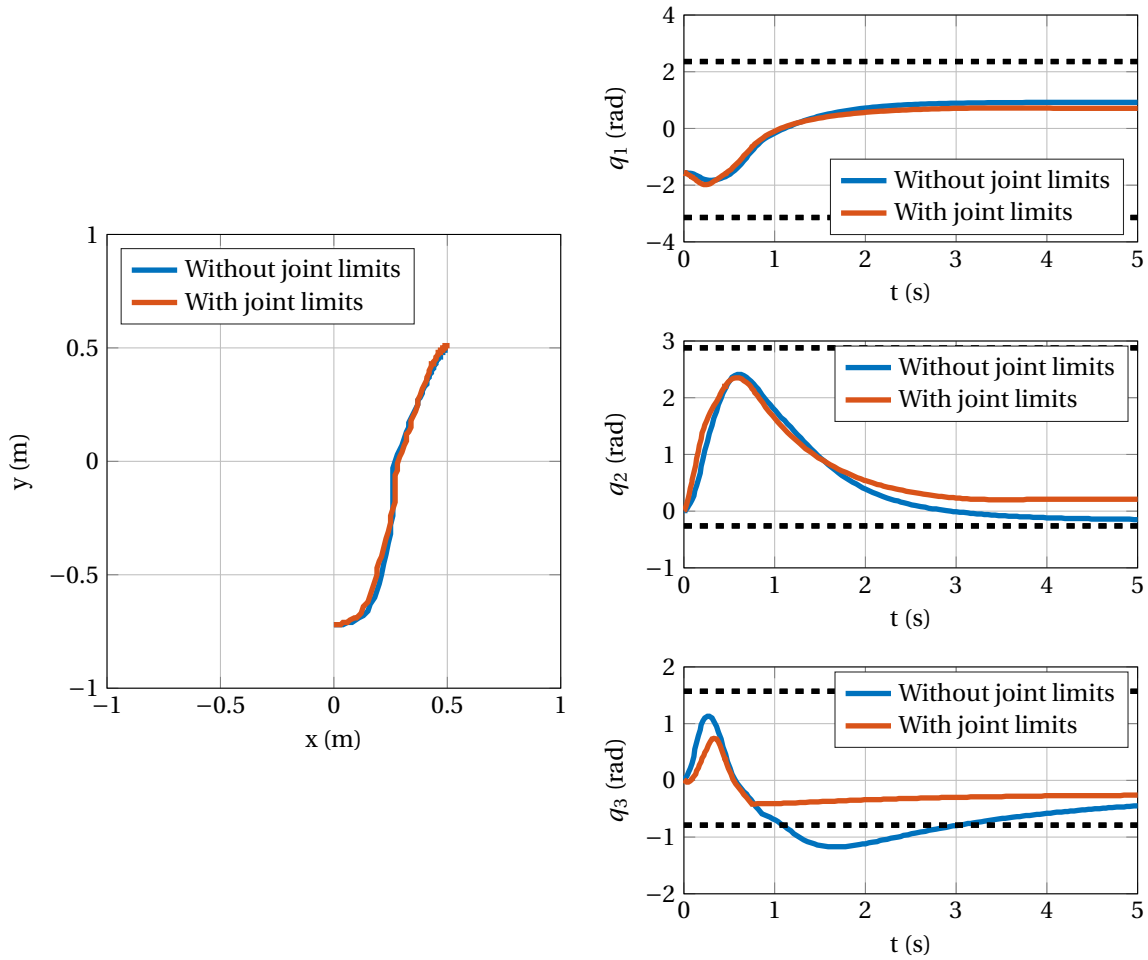


Figure 4.5: Comparing the same motion as before starting at $[0, -0.72]$ and moving towards $[0.5, 0.5]$, but now without and with joint limit control included. The trajectories in the workspace are given together with the angular position of each link for both situations.

any motion within the reach of the 3-link manipulator. Figure 4.6 shows the reachable workspace of the robot taking into account to the joint limits introduced earlier. It could be imagined that the shortest path between the initial and final position goes straight through the unreachable workspace, which induces getting stuck at the edge of the arm's feasible configuration.

In order to overcome this problem of getting stuck at certain configurations, an extra coordinate will be introduced in the PDW, next to the original x- and y-coordinates. The new coordinate will force the arm to move through the reachable workspace, by looking at the angle of the end-effector with respect to the horizontal axis. This angle will be found by making use of the x-, and y-coordinates. Besides, the angle of the end-effector's desired location could be calculated using its x-, and y-coordinates as well. So, without extra knowledge of the robot's configuration this extra, angle coordinate could be implemented, from now on called polar angle. Figure 4.7 shows an example

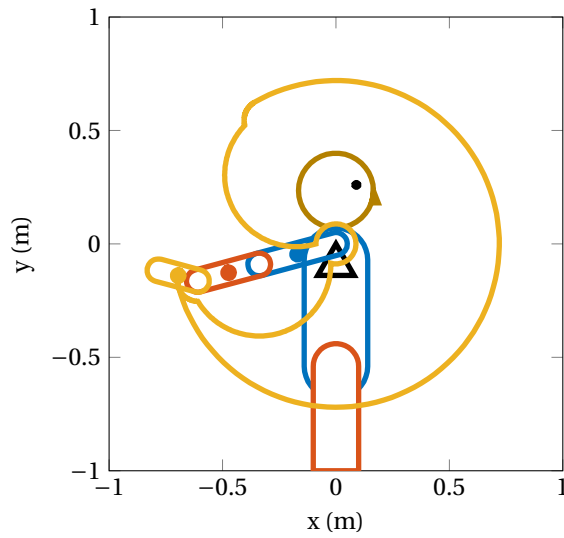


Figure 4.6: The reachable workspace of robot arm introduced by the joint limits, where each location within the area could be reached. Local minima are hit when the shortest path goes straight through the unreachable area.

of the current and desired polar angle of the end-effector.

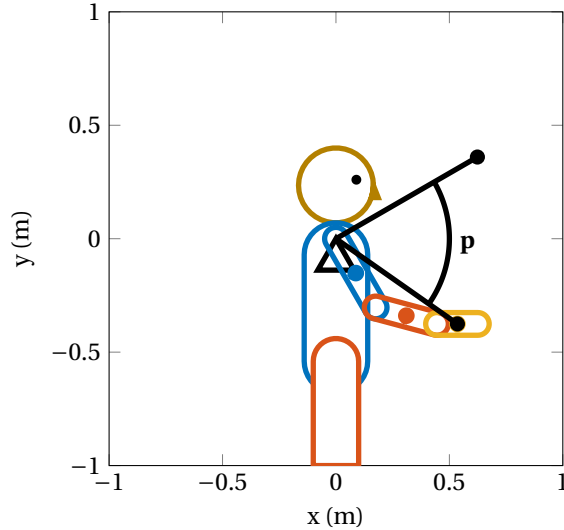


Figure 4.7: The new polar angle describes the angle between the current and desired x -, y -position, so the robot will always move through its reachable area. The polar angle is included as an extra coordinate in the task controller.

The difference between the two angles shown in Figure 4.7 represents the error to be minimized with the control law. This error can easily be included in the part u_{task} of the control law seen in (4.13) by subtracting the desired polar angle from the current

angle. The new coordinate frame will look as

$$X = \begin{bmatrix} x \\ y \\ \tan\left(\frac{y}{x}\right) \end{bmatrix}. \quad (4.15)$$

The mapping from this new coordinate frame on the configuration coordinates will be done using the Jacobian, which will slightly change as

$$J_{\text{task}}^T = \begin{bmatrix} \frac{\partial x}{\partial q} \\ \frac{\partial y}{\partial q} \\ \frac{\partial \tan\left(\frac{y}{x}\right)}{\partial q} \end{bmatrix}^T. \quad (4.16)$$

The polar angle compensation should only be influencing the motion when there is an actual change of getting stuck in a local minimum. As explained earlier and seen in Figure 4.6 the problem appears when the shortest path between initial and desired position crosses the unreachable area. This will only happen when the error between the current and the desired polar angle is greater than π rad. Therefore, the polar angle control should only be influencing the movement when this error is indeed larger than π rad.

In order to deactivate this part of the control law when the error is smaller an activation function is implemented. The activation function looks the same as seen with the joint limits, but this time there is only one activation value needed. Figure 4.8 represents the activation function of the polar angle control, where errors smaller than π rad are suppressed for control.

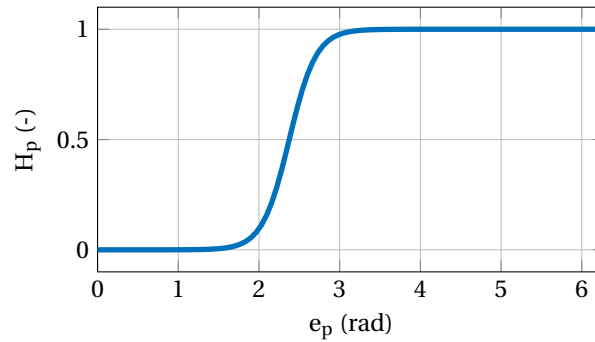


Figure 4.8: The activation function for polar angle control, because it should only influence the movement when there is a change of hitting a local minima.

The control law of (4.13) will slightly change when including the polar angle control. The task control part can be seen in Equation (4.17), where the activation function is given in (4.18)

$$u_{\text{task}} = J_{\text{task}}^T(q) \left(K_p \begin{bmatrix} e_r \\ H_p e_p \end{bmatrix} + K_d J_{\text{task}}(q) \dot{q} \right) + G(q), \quad (4.17)$$

$$H_p(q) = \frac{1}{2} (\tanh(|e_p| - \pi) + 1). \quad (4.18)$$

As done before, we will again compare the results using this new part of the control law with respect to the previous results. It should first of all be noticed that no difference will be found when looking at trajectory as seen in Figure 4.5. The difference between the initial and final polar angle does not exceed the π rad limit, which means the polar control is not activated.

Figure 4.9 represents a motion starting at $[0, 0.72]$ with a desired position at $[0, -0.72]$, both without and with the use of polar angle. For completeness, the simulation results excluding the joint limits is also shown.

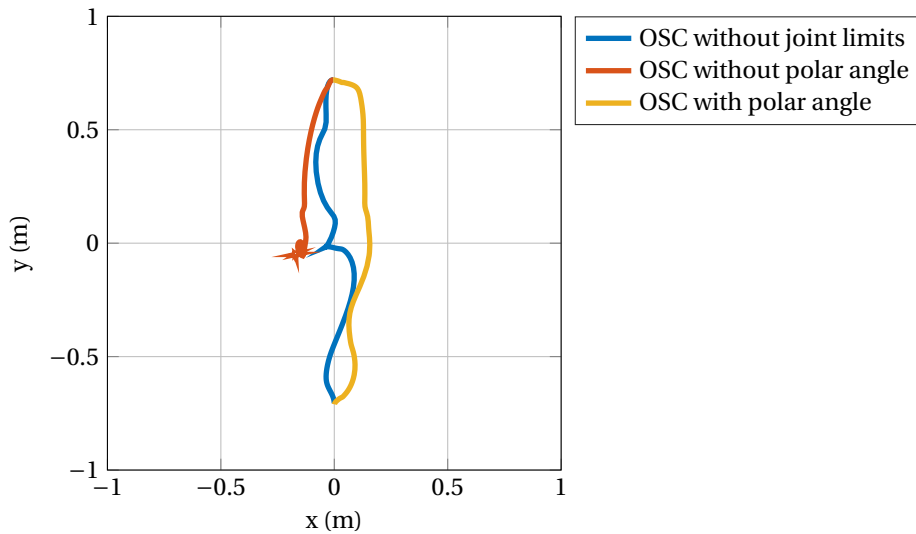


Figure 4.9: Comparing the trajectories starting at $[0, 0.72]$ and moving towards $[0, -0.72]$ in the workspace, without joint limits, with joint limits but without polar angle and with both.

From this figure the reason for introducing the polar angle control should become quite clear. It is shown that the initial control without joint limits does not show any difficulties getting to the desired position. The inclusion of joint limits caused the arm to get stuck in all three angles. The polar angle on the other hand forces the end-effector to stay within the reachable workspace only and is therefore able to reach the desired position.

4.4 Lyapunov stability

It is important to know whether the controller performs stable operations, so the Lyapunov stability theorem as presented in [31] will be used to confirm this stability. The theorem states that an equilibrium point is stable when the chosen Lyapunov function meets certain requirements.

Theorem 1. *Let $x = 0$ be an equilibrium point for $\dot{x} = h(x)$ and $\mathcal{Q} \subset \mathbb{R}^n$ be a domain containing $x = 0$. Let $V : \mathcal{Q} \rightarrow \mathbb{R}$ be a continuously differentiable function, such that*

$$V(0) = 0 \quad \text{and} \quad V(x) > 0 \text{ in } \mathcal{Q} \setminus \{0\}, \quad (4.19)$$

$$\dot{V}(x) \leq 0 \text{ in } \mathcal{Q}. \quad (4.20)$$

Then, $x = 0$ is stable. Moreover, if

$$\dot{V}(x) < 0 \text{ in } \mathcal{Q} \setminus \{0\}, \quad (4.21)$$

then $x = 0$ is locally asymptotically stable. Finally, if $\mathcal{Q} = \mathbb{R}^n$ and

$$\begin{aligned} V(0) &= 0, \\ V(x) &> 0 \quad \text{in } \mathcal{Q} \setminus \{0\}, \\ \dot{V}(x) &< 0 \quad \text{in } \mathcal{Q} \setminus \{0\}, \\ V(x) &\rightarrow \infty \quad \text{if } \|x\| \rightarrow \infty, \end{aligned} \quad (4.22)$$

then the equilibrium $x = 0$ is globally asymptotically stable.

The results according to this theorem could be unclear about whether an equilibrium is just stable or asymptotically stable, due to the negative semidefiniteness of the Lyapunov derivative and not strictly negative definite. Theorem 2, called LaSalle's Invariance Principle, is be useful when conclusions about asymptotic stability of the equilibrium point are drawn, making use of the largest invariant set. Figure 4.10 visualizes LaSalle's Invariance principle, where the symbols Ω , \mathcal{E} and \mathcal{M} are explained in Theorem 2.

Theorem 2. *Let $\Omega \subset \mathcal{Q}$ be a compact set that is positively invariant with respect to $\dot{x} = h(x)$ and $V : \mathcal{Q} \rightarrow \mathbb{R}$ be a continuously differentiable function such that $\dot{V}(x) \leq 0$ in Ω . Furthermore, let \mathcal{E} be the set of all points in Ω where $\dot{V} = 0$. Finally, let \mathcal{M} be the largest invariant set in \mathcal{E} , then every solution starting in Ω approaches \mathcal{M} as $t \rightarrow \infty$.*

The two theorems above are sufficient enough to prove asymptotic converge of the controller to a certain region, as will be done next. Consider the model as constructed earlier:

$$\ddot{q} = M^{-1}(q) (u - C(q, \dot{q})\dot{q} - G(q) - b\dot{q}), \quad (4.23)$$

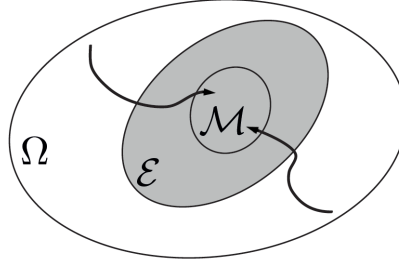


Figure 4.10: LaSalle Invariance Principle, where \mathcal{E} represents the all points in the set Ω where $\dot{V} = 0$, and \mathcal{M} corresponds to the largest invariant set of \mathcal{E} , approached by all solutions in Ω , based on [32].

where u represents the control law without the gravity torque part u_G of Equation (4.11), so:

$$u = -(1 - |H_{jl}(q)|) J_{\text{task}}^T(q) (K_p e_r + J_{\text{task}} K_d \dot{q}) - K_{jl} H_{jl}(q) + G(q). \quad (4.24)$$

It was chosen to prove the stability for this part of the control law first, before making it more complicated by introducing the non-linear OSC term.

Now, according to Theorem 1, a Lyapunov function should be proposed such that the requirements of Equation (4.19)-(4.21) are met. The candidate Lyapunov function chosen for the model and control law mentioned above is given in Equation (4.25).

$$V(\dot{q}, q, e_r) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} e_r^T (1 - |H_{jl}(q)|) K_p e_r + K_{jl} P_{jl}(q). \quad (4.25)$$

All terms in the first two parts of this function have been shown before. Nevertheless, the last part introduces a new term $P_{jl}(q)$ that is the integral of the joint limit function $H_{jl}(q)$ to q , so

$$\begin{aligned} P_{jl}(q) &= \int H_{jl}(q) dq \\ &= \frac{1}{2} (\log(\cosh(q - q_{\min})) + \log(\cosh(q - q_{\max}))) + c, \end{aligned} \quad (4.26)$$

where c puts the minimum of the function to zero, whereby the entire function is shifted down. Introducing the function $P_{jl}(q)$ eliminates the $-K_{jl}H_{jl}(q)$ term of the control law, as will be seen later. Figure 4.11 shows the trajectories of the function $P_{jl}(q)$ for all three links, related to the joint limit functions given in Figure 4.4.

Let us now have a look at the Lyapunov candidate, which should meet the requirements of Theorem 1. In practice it means that the Lyapunov function equals zero at the equilibrium point and is greater than zero elsewhere. The first term in Equation (4.25) is a quadratic function of \dot{q} around $M(q)$, where $M(q)$ is a positive definite matrix. These two observations lead to the conclusion that this part will always be greater than zero except for when $\dot{q} = 0$ rad/s.

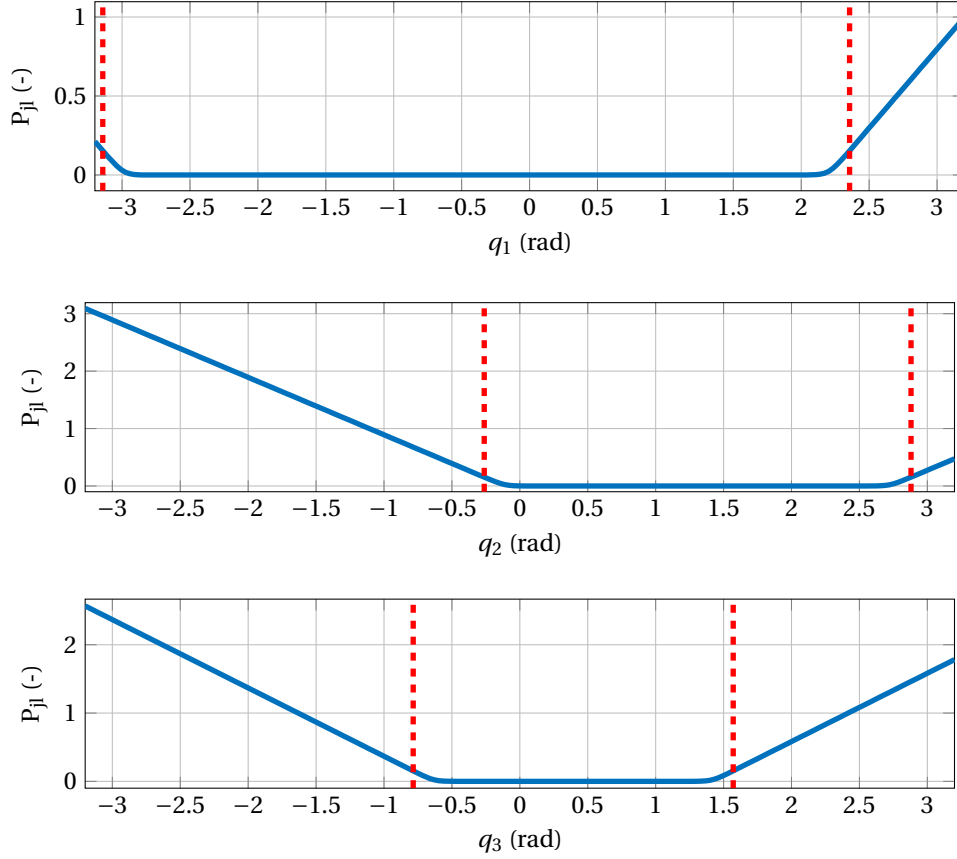


Figure 4.11: Partial integrals of the joint limit function H_{jl} called P_{jl} . Values equal to zero represent configurations where the limits are not reached. The robot's configuration is bounded by $q_1 : [-\pi, \frac{3}{4}\pi]$, $q_2 : [-\frac{1}{12}\pi, \frac{11}{12}\pi]$ and $q_3 : [-\frac{1}{4}\pi, \frac{1}{2}\pi]$.

The second term is quadratic in e_r , the error in workspace coordinates. It has been shown before that $H_{jl}(q)$ is bounded by -1 and 1 , therefore

$$\begin{aligned} 0 &\leq |H_{jl}(q)| \leq 1, \\ 0 &\leq (1 - |H_{jl}(q)|) \leq 1. \end{aligned} \tag{4.27}$$

The middle part of the quadratic term $(1 - |H_{jl}(q)|) K_p$ equals zero when the joint limits are reached and is greater than zero otherwise. It can be concluded that the second term disappears when the error e_r between end-effector and desired position is zero or when the robot got stuck due to its joint limits.

In order to understand whether the final term of Equation (4.25) meets the requirements of Theorem 1, Figure 4.11 should be reconsidered. It is seen that the function equals zero when the configuration coordinates are located within the bounds. In other words, when the robot reaches its joint limits the function becomes positive. As a result, according to the term $K_{jl}P_{jl}(q)$, the equilibrium point should be located within the reachable area.

The analysis of the three terms above concludes in a set of equilibrium points

$$S = \{(\dot{q}, q) \in \mathbb{R}^3 \times \mathbb{R}^3 | \dot{q} = 0, e_r = 0, q_{\min} < q < q_{\max}\}. \quad (4.28)$$

Furthermore, when $(1 - |H_{jl}(q)|) = 0$, meaning that all three joint limits are reached, it was found that $P_{jl} > 0$. Therefore, the Lyapunov function is greater than zero and so these configurations do not belong to the set of equilibrium points. So it has been proven that

$$V(x) > 0 \quad \forall x \neq 0.$$

The stability analysis of the set of equilibrium points, given in (4.28), is derived by the derivative of the candidate Lyapunov function

$$\dot{V} = \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \dot{e}_r^T (1 - |H_{jl}(q)|) K_p e_r - \frac{1}{2} e_r^T \frac{\partial |H_{jl}(q)|}{\partial q} \dot{q} e_r + K_{jl} H_{jl}(q) \dot{q}.$$

Here it should be noticed that the partial derivative of P_{jl} to q indeed equals H_{jl} . The second derivative of q could be found from the model of Equation (4.23), with the control input u given in Equation (4.24). Implementing these two functions in the Lyapunov derivative results in the following derivation

$$\begin{aligned} \dot{V} &= \dot{q}^T M(q) M^{-1}(q) \left(-(1 - |H_{jl}(q)|) J_{\text{task}}^T(q) (K_p e_r + J_{\text{task}} K_d \dot{q}) - K_{jl} H_{jl}(q) - C(q, \dot{q}) \dot{q} \right. \\ &\quad \left. - b \dot{q} \right) + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \dot{e}_r^T (1 - |H_{jl}(q)|) K_p e_r - \frac{1}{2} e_r^T \frac{\partial |H_{jl}(q)|}{\partial q} \dot{q} e_r + K_{jl} H_{jl}(q) \dot{q} \\ &= -\dot{q}^T J_{\text{task}}^T(q) (1 - |H_{jl}(q)|) K_p e_r - \dot{q}^T J_{\text{task}}^T(q) (1 - |H_{jl}(q)|) K_d J_{\text{task}} \dot{q} - \dot{q}^T K_{jl} H_{jl}(q) \\ &\quad + \frac{1}{2} \dot{q}^T (\dot{M}(q) - 2C(q, \dot{q})) \dot{q} - \dot{q}^T b \dot{q} + \dot{e}_r^T (1 - |H_{jl}(q)|) K_p e_r - \frac{1}{2} e_r^T \frac{\partial |H_{jl}(q)|}{\partial q} \dot{q} e_r \\ &\quad + K_{jl} H_{jl}(q) \dot{q} \end{aligned}$$

and so

$$\dot{V} = -\dot{e}_r^T (1 - |H_{jl}(q)|) K_d \dot{e}_r - \dot{q}^T b \dot{q} - \frac{1}{2} e_r^T \frac{\partial |H_{jl}(q)|}{\partial q} \dot{q} e_r. \quad (4.29)$$

A number of properties are used here to get to this solution. First of all, the term $\dot{M}(q) - 2C(q, \dot{q})$ is proven to be skew-symmetric, which means that pre- and post-multiplying this matrix by \dot{q} will always be equal to zero. Besides, it is known by definition that $\dot{e}_r = J_{\text{task}}(q) \dot{q}$ and so $\dot{e}_r^T = \dot{q}^T J_{\text{task}}^T(q)$, used to eliminate the K_p term. Finally, the matrix $K_{jl} H_{jl}$ is a diagonal, symmetric matrix and so $K_{jl} H_{jl} = (K_{jl} H_{jl})^T$.

Again, the requirements seen in Theorem 1 are now put to the test to conclude whether the equilibrium points are stable or not. The theorem stated that Equation (4.29) should be smaller than zero for all values not in the equilibrium, so

$$\dot{V}(x) < 0 \text{ in } \mathcal{Q} \setminus \{0\}.$$

Consider the first term of Equation (4.29), the quadratic term \dot{e}_r encloses $(1 - |H_{jl}(q)|) K_d$. Recalling Equation (4.27) shows that the middle term will always be greater than zero unless when the joint limits are reached. Equation (4.28) presented that the equilibrium points are located within the joint limit region, so the middle term is positive definite. Therefore the entire first term of Equation (4.29) only equals zero when $\dot{q} = 0$ rad/s and is negative otherwise.

The quadratic expression $-\dot{q}^T b \dot{q}$, as shown in the Lyapunov derivative, is smaller than zero as long as the robot keeps moving and will only be equal to zero when $\dot{q} = 0$.

The third term of Equation (4.29) involves slightly more explanation before conclusions about meeting the requirements is drawn. It is shown that the term is quadratic in e_r , but nothing is known about the middle term yet. First consider the derivative of joint limits $\frac{\partial |H_{jl}(q)|}{\partial q}$ in Figure 4.12. It is seen that the derivative is both upper as

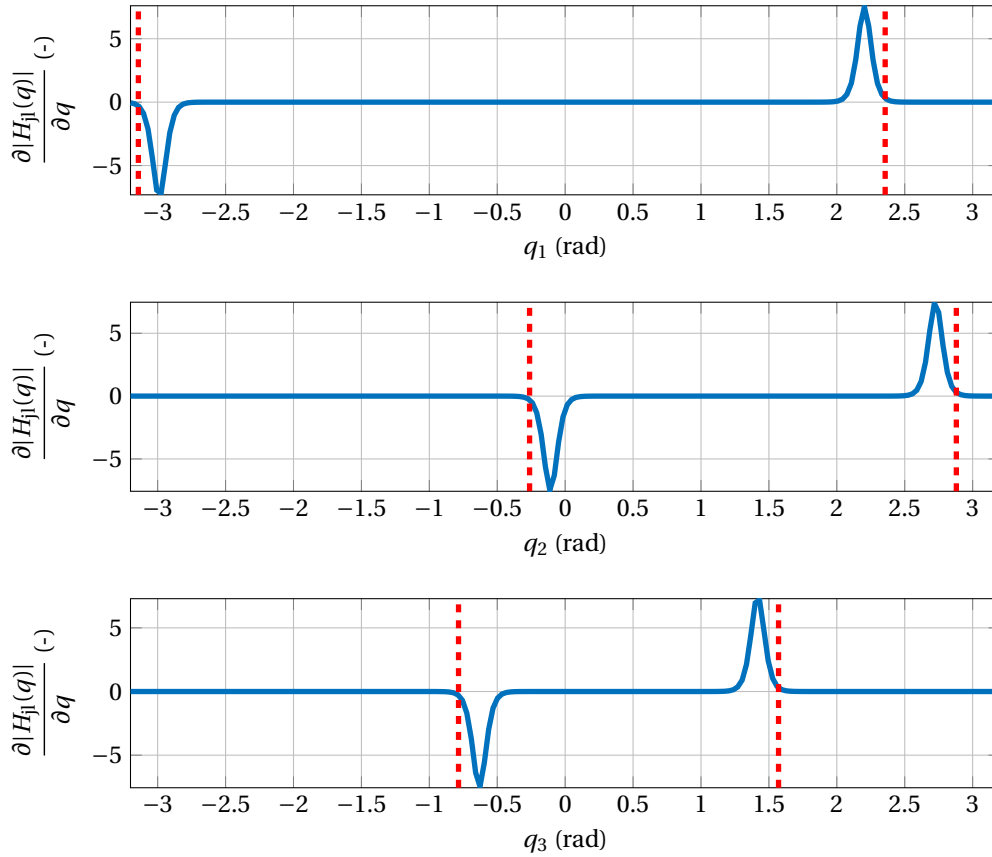


Figure 4.12: Partial derivative of the joint limit function H_{jl} , which is both lower and upper bound. Values equal to zero represent configurations where the limits are not reached. The robot's configuration is bounded by $q_1 : [-\pi, \frac{3}{4}\pi]$, $q_2 : [-\frac{1}{12}\pi, \frac{11}{12}\pi]$ and $q_3 : [-\frac{1}{4}\pi, \frac{1}{2}\pi]$.

well as lower bounded, which will be shown useful later. The minus sign in front of the partial derivative in (4.29) results in flipping the plots of Figure 4.12 around the y-axis, so the upper bound becomes the lower bound and vice versa. The same results as this

function could be obtained by taking the partial derivative of $-|H_{jl}(q)|$, so

$$-\frac{\partial |H_{jl}(q)|}{\partial q} = \frac{\partial -|H_{jl}(q)|}{\partial q},$$

which means the last term of Lyapunov derivative is rewritten into positive element only as shown in Equation (4.30)

$$-\frac{1}{2}e_r^T \frac{\partial |H_{jl}(q)|}{\partial q} \dot{q}e_r = \frac{1}{2}e_r^T \frac{\partial -|H_{jl}(q)|}{\partial q} \dot{q}e_r. \quad (4.30)$$

It is known that following matrix properties holds

$$\begin{aligned} \left\| \frac{1}{2}e_r^T \frac{\partial -|H_{jl}(q)|}{\partial q} \dot{q}e_r \right\| &\leq \frac{1}{2} \|e_r^T\|_2 \left\| \frac{\partial -|H_{jl}(q)|}{\partial q} \dot{q}e_r \right\|_2 \\ &\leq \frac{1}{2} \|e_r^T\|_2 \left\| \frac{\partial -|H_{jl}(q)|}{\partial q} \dot{q} \right\|_2 \|e_r\|_2 \\ &\leq \frac{1}{2} \|e_r\|_2^2 \left\| \frac{\partial -|H_{jl}(q)|}{\partial q} \dot{q} \right\|_2, \end{aligned} \quad (4.31)$$

where $\|e_r\|_2$ could be merged together, because the 2-norm of the vector e_r is represented by a scalar.

Now consider Hölder's inequality [33]

$$\|fg\|_r \leq \|f\|_p \|g\|_q \quad \text{with} \quad \frac{1}{r} = \frac{1}{p} + \frac{1}{q}, \quad (4.32)$$

where f and g are real- or complex-valued functions and $r, p, q \in [1, \infty]$. This inequality is useful when upper bounding the Lyapunov derivative, as shown in Equation (4.33).

$$\begin{aligned} \left\| \frac{1}{2}e_r^T \frac{\partial -|H_{jl}(q)|}{\partial q} \dot{q}e_r \right\| &\leq \frac{1}{2} \|e_r\|_2^2 \left\| \frac{\partial -|H_{jl}(q)|}{\partial q} \dot{q} \right\|_2 \\ &\leq \frac{1}{2} \|e_r\|_2^2 \left\| \frac{\partial -|H_{jl}(q)|}{\partial q} \right\|_\infty \|\dot{q}\|_2 \\ &\leq \frac{1}{2} \|e_r\|_2^2 \|\dot{q}\|_2 \sigma, \end{aligned} \quad (4.33)$$

where σ is the upper bound of the partial derivative. Besides, the property of the Lyapunov function is that it decreases when getting closer to the desired position. Therefore, this error is bounded by the initial position as

$$\|e_r\|_2^2 \leq \|e_r(0)\|_2^2.$$

As a result, the Lyapunov function will always be smaller than the upper bound of Equation (4.33) and therefore it is known that

$$\dot{V} \leq -\dot{q}^T J_{\text{task}}^T(q) (1 - |H_{jl}(q)|) K_d J_{\text{task}}(q) \dot{q} - \dot{q}^T b \dot{q} + \frac{1}{2} \|e_r(0)\|_2^2 \|\dot{q}\|_2 \sigma. \quad (4.34)$$

In conclusion, the derivative of the Lyapunov function is smaller than zero when the matrix

$$J_{\text{task}}^T(q) (1 - |H_{\text{jl}}(q)|) K_d J_{\text{task}}(q) + b - \frac{1}{2} \|e_r(0)\|_2^2 \sigma \quad (4.35)$$

is positive definite, accomplished by choosing the value for K_d wisely.

Note that when $|H_{\text{jl}}(q)| = 1$, the whole first term of Equation (4.35) vanishes and Lyapunov stability of the equilibrium position can no longer be guaranteed. The Lyapunov has converged to an unstable region and stops at the edge where $\dot{V} = 0$. In physical terms it means that the robot reaches its joint limits and so cannot move any further.

Equation (4.34) is equal to zero when $\dot{q} = 0$, but nothing is known about the end-effector position or robot's configuration yet. Therefore, LaSalle's Invariance Principle should be used (Theorem 2) to find the largest invariant set.

If the angular velocity equals zero, it is known that its acceleration will also be equal to zero

$$\dot{q} = 0 \rightarrow \ddot{q} = 0.$$

Substituting these two values into Equation (4.23) and (4.24) results in

$$\begin{aligned} 0 &= M^{-1}(q) (u - 0 - G(q) - 0) \\ &= (1 - |H_{\text{jl}}(q)|) J_{\text{task}}^T(q) K_p e_r + K_{\text{jl}} H_{\text{jl}}(q). \end{aligned} \quad (4.36)$$

Now, it could be shown that there is only one set of conditions possible

$$\begin{cases} e_r = 0 \\ H_{\text{jl}}(q) = 0 \rightarrow q_{\min} < q < q_{\max}, \end{cases} \quad (4.37)$$

which is equal to the set of Equation (4.28).

In conclusion, the control law asymptotically converges to the stable set of equilibrium points of Equation (4.28), when the joint limits are not reached. Note that when all the joint limits are activated the robot will converge to a certain region.

Now that stability of the first part is proven, the rest of the controller could be considered as

$$\begin{aligned} u &= -(1 - |H_{\text{jl}}(q)|) J_{\text{task}}^T(q) (K_p e_r + J_{\text{task}}(q) K_d \dot{q}) - K_{\text{jl}} H_{\text{jl}}(q) + G(q) \\ &\quad - (1 - |H_{\text{jl}}(q)|) \|X - X_{\text{des}}\| J_G^T(q) K_G \text{Sgn}_G(q) G(q). \end{aligned} \quad (4.38)$$

The Lyapunov candidate will not change, which means the same set of equilibrium points is found. Following the same procedure as before the Lyapunov derivative changes slightly by the addition of an extra term as

$$\begin{aligned} \dot{V} &\leq -\dot{q}^T J_{\text{task}}^T(q) (1 - |H_{\text{jl}}(q)|) K_d J_{\text{task}}(q) \dot{q} - \dot{q}^T b \dot{q} + \frac{1}{2} \|e_r(0)\|_2^2 \|\dot{q}\|_2 \sigma \\ &\quad - \dot{q}^T J_G^T(q) (1 - |H_{\text{jl}}(q)|) \|X - X_{\text{des}}\| K_G \text{Sgn}_G(q) G(q). \end{aligned} \quad (4.39)$$

The stability analysis of the first part has already been provided, so the latter term in (4.39) has the only potential to change the conclusion drawn earlier. Each of the components in this part can be considered individually, which are

$$\begin{aligned} \dot{q}^T J_G^T(q) &= \dot{e}_G^T, \\ 0 &\leq (1 - |H_{ji}(q)|) \leq 1, \\ 0 &\leq \|X - X_{\text{des}}\| \leq \|e_r(0)\|_2, \\ 0 &\leq \text{Sgn}_G(q)G(q) \leq |\text{Sgn}_G(q)G(q)|_\infty. \end{aligned} \tag{4.40}$$

Furthermore, the derivative of the gravity torque error is bounded, because it cannot increase or decrease more than the slope of the landscape. Besides, the landscape is symmetric, so

$$\begin{aligned} -|\dot{e}_G^T|_\infty &\leq \dot{e}_G^T \leq |\dot{e}_G^T|_\infty, \\ -|\dot{e}_G^T|_\infty &\leq -\dot{e}_G^T \leq |\dot{e}_G^T|_\infty, \end{aligned} \tag{4.41}$$

It can be seen that, except for \dot{e}_G , all terms are larger or equal to zero for any configuration and contain an upper bound. Therefore it is known that the matrix

$$\begin{aligned} -\dot{e}_G^T (1 - |H_{ji}(q)|) \|X - X_{\text{des}}\| K_G \text{Sgn}_G(q)G(q) &\leq \\ |\dot{e}_G^T|_\infty (1 - |H_{ji}(q)|) \|e_r(0)\|_2 K_G |\text{Sgn}_G(q)G(q)|_\infty. \end{aligned}$$

This term could be positive, which should be compensated by the other terms in Equation (4.39). The gains K_d and K_G could be freely chosen and therefore the former should be significantly high and latter low enough to achieve stability.

Furthermore, when $\dot{q} = 0$ Equation (4.39) equals zero, so the conclusion about stability still holds. Therefore, introducing the second part of the control law does not change the robot's behavior in terms of stability around the set of equilibrium points in Equation (4.28).

4.5 Synopsis

In this chapter the introduction to Operational Space Control is given in order to create an hierarchical order of the main controller task and additional tasks, making use of the main task's null-space. In this research the additional task is improving the human-likeness of the robot's movements, which is done using the gravity torque map as shown in Section 3.2.2. On top of this, the distance between the end-effector's current position and desired position is used to further reduce the influence of the gravity torque control when getting closer to the goal.

It was furthermore found that awkward joint angles prevent the robot from looking human-like. This results in the introduction of limited freedom on each of the joints. These joint limits are created using the $\tanh(x)$, so that this constraint will only be effective when the bounds are reached.

Finally, local minimum appear caused by the joint limit introduction, which causes the robot to get stuck in undesired configurations. The introduction of the polar angle, the angle between the current and desired angle of the end-effector, solves this problem. The polar angle is given as an extra coordinate besides the original workspace coordinates of the end-effector.

The final control law created in this chapter is shown in Equation (4.42)

$$\begin{aligned}
 u &= (1 - |H_{jl}(q)|) (u_{\text{task}} + N_{\text{task}}^T u_G) + K_{jl} H_{jl}(q), \\
 u_{\text{task}} &= J_{\text{task}}^T(q) \left(K_p \begin{bmatrix} e_r \\ H_p e_p \end{bmatrix} + K_d J_{\text{task}}(q) \dot{q} \right) + G(q), \\
 u_G &= \|X - X_{\text{des}}\| J_G^T(q) K_G \text{Sgn}_G(q) G(q).
 \end{aligned} \tag{4.42}$$

Convey information

In previous chapter the general control framework that creates human-like motions was presented, taking into consideration the system metrics of Chapter 3. Current chapter will focus on conveying information by modifying the manipulator's trajectories only, making use of the introduced framework. Humans are very capable of understanding non-verbal communication, which means the generated movements should also be understandable by the robot's surroundings.

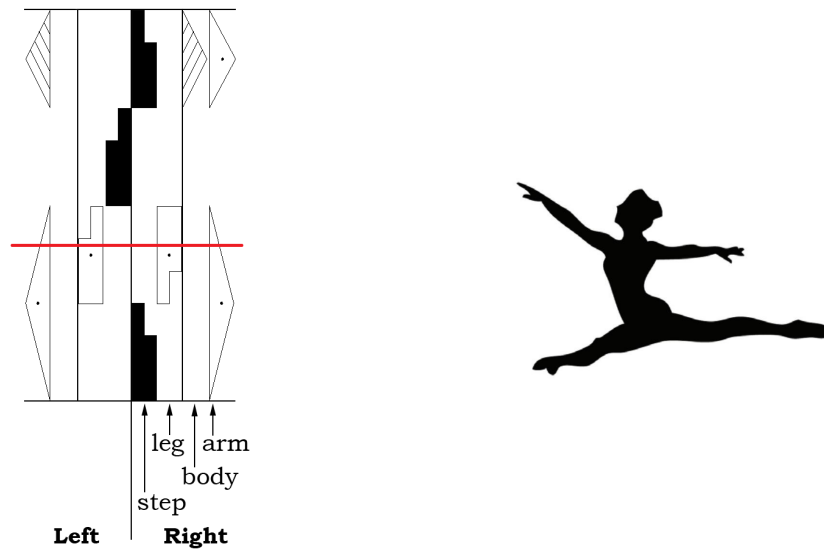
First, one way of structurally describing human movements is introduced called the Labanotation, after which the Laban Efforts demonstrates how to emphasize on different styles in a motion. These Efforts are used to build a verification method divided into an experiment and an optimization procedure. The results of both parts are given and the discussion answers the question whether the robot is able to generate different movement trajectories.

5.1 Labanotation input

One way of defining a set of motions is using the Labanotation, originally derived by Rudolf Laban in 1928 [34]. The Labanotation can be compared to the way music is described, but specialized for the description and analyzes of human movements in a structured way. This notation of motions was initially designed to describe dance movements, but could also be quite useful when generating robotic motions, especially when human-like motions are required.

Figure 5.1a shows an example of the typically Labanotation. It can be seen that the figure is divided into two parts, left and right, where each space on the horizontal axis corresponds to one element of the body. From the middle outwards there is: support column gestures, leg gestures, body gestures and arm gestures. Occasionally a fifth column could be found on the right side, which represents to the head gestures. Where music notation is read from left to right, the Labanotation expresses the order of the

motion sequence from bottom to top. The length of each symbol defines the duration of the motion.



(a) The notation is read from bottom to top, where each shape and shade corresponds to a different movement. The red horizontal line indicates the current configuration. (b) Dancer performing the split jump, corresponding to the red horizontal line in the Labanotation [35].

Figure 5.1: Example of the Labanotation, one way of representing motion sequences, especially used to describe dancing movements.

The symbols as shown in Figure 5.1a indicate the direction of the gesture using both the shape and the shading of these symbols. In spite of the fact that the total Labanotation alphabet consists of over 100 symbols only some will be introduced here. The most important shapes can be found in Figure 5.2, together with the different shading possibilities for each shape.

The rectangular-shaped symbol is the placement of the arm along the body and the missing corners gives the motion's in forward or backward direction. Therefore, these five symbols in the middle represents motions performed in the sagittal plane of the body. The triangles are gestures towards the side. Combinations of the sideward and forward/backward motions results in diagonal motions to the corresponding side of the body.

Furthermore, it can be seen in Figure 5.2 that the completely colored symbols correspond to movements close to the ground, while empty or partially filled symbols are linked to middle and high directions respectively. These different shades can be used for each shape on all parts of the body.

Looking back at Figure 5.1a it should now be clear that both arms are pointing parallel to the ground at first, after which they are moving slightly up to a 45° angle.

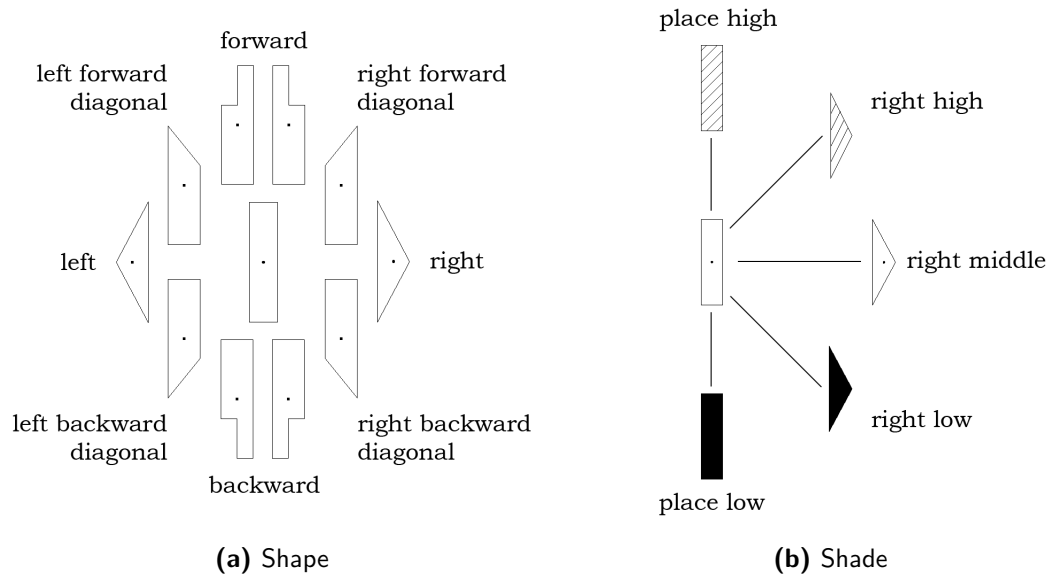






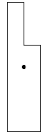
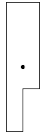

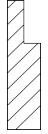
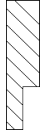
Figure 5.2: Labanotation symbols, a brief selection of the available alphabet [36].

Meanwhile, the dancer starts at its right foot and performs a jump, where simultaneously the left leg is brought forward and the right leg backwards both in horizontal direction. A snapshot of a dancer performing this split jump with her arms horizontally is shown in Figure 5.1b.

Now that the language to describe different kinds of human motion is known, it could be translated into a mathematical representation. This representation makes it possible to perform the same motions as humans do by the robot. In previous chapters the robotic framework together with the control approach is given, summarized as a single armed motion in the 2D sagittal (lateral) plane in the workspace. This is the reason that only parts of the introduced Labanotation will be useful to express motions, where gestures towards the side of the body are excluded. The mathematical translation implies a relation between the Labanotation symbols and the desired location of the arm's end-effector. Table 5.1 gives an overview of all possible symbols and the relation with their desired x- and y-coordinates.

It is assumed that an arm with length equal to 1 is used to compose the desired locations in the overview. It can be seen that the different shapes and shades all corresponds to the configurations in the plane along the side of the body. In current situation, the combinations with shape 3 and height 1 and 2 should be excluded from this list, while the robot is not able to reach these positions due to its joint limits. Nevertheless it was chosen to present all symbols, so the framework remains general and useful when considering different cases.

Table 5.1: Created relation between the Labanotation symbols and the desired workspace coordinates of the end-effector.

		Shape					
		1		2		3	
Height	1		0 -1		$\frac{1}{2}\sqrt{2}$ $-\frac{1}{2}\sqrt{2}$		$-\frac{1}{2}\sqrt{2}$ $-\frac{1}{2}\sqrt{2}$
	2		0 0		1 0		-1 0
	3		0 1		$\frac{1}{2}\sqrt{2}$ $\frac{1}{2}\sqrt{2}$		$-\frac{1}{2}\sqrt{2}$ $\frac{1}{2}\sqrt{2}$

The overview presented above could now be used to create a sequence of movements, which the robot will be conducting during the rest of this research. The input is based on a dancing exercise called *Port de Bras*, but only considered in the two dimensional plane. Figure 5.3 shows the sequence that is used to control the robot's motions. The initial position has both legs on the ground and both arms hanging alongside the body. Like said earlier, the movements will only be done by the right arm in the sagittal plane, and therefore the legs and left arm got an hold-sign.

The motion sequence with the right arm is as follows:

1. The arm moves forward to a horizontal position in front of the body parallel to the ground.
2. The arm needs to move upward into a vertical configuration.
3. From there it moves back to the initial position, so along the side of the body.
4. Next, another movement to the front of the body is performed.
5. The final location is found at a 45° angle upwards with respect to the horizontal axis.

During the remaining part of this chapter, the introduced sequence will be used to show the difference between different Laban efforts, as introduced in the next.

5.2 Laban Efforts

One way of describing different movement styles could again be found within the dance community called the Laban Movement Analysis (LMA). As covered briefly in Chapter

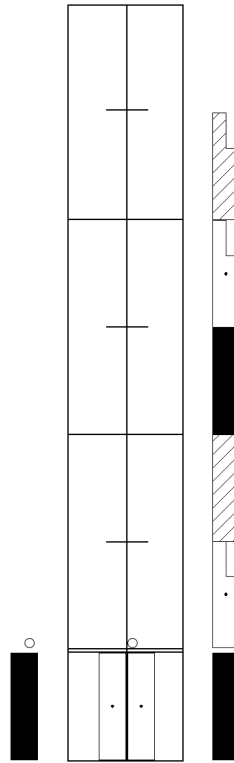


Figure 5.3: Input sequence used to control robot, based on dancing exercise *Port de Bras*.

1, this movement interpreting tool was originally developed by Rudolf Laban, the same initiator of the Labanotation.

The general idea is to describe and interpret all varieties of the human movements, which makes it possible to transfer the observed information. The Effort component of the LMA qualifies the human movement's dynamics rather than static positions and could therefore be a quite valuable component [37]. Originally, the Laban Efforts were mainly used by choreography to discover and convey different styles of dance movements. Later on, acting teachers found out that it is a great way to learn their students to play different characters. Especially the fact that the Laban Efforts are useful to convey information differently is the motivation to dig deeper into this way of qualifying movements.

The Laban Efforts are subdivided into four components, all describing different aspects of a motion [38]:

- Space: the attention to the mover's surroundings, so exploring the area around him freely by moving his hand opposed to make a pointing gesture straight towards a specific goal.

- **Weight:** the sense of the movement's impact, that is putting high pressure and tension into the movement in contrast to moving light weighted like a feather.
- **Time:** the lack of sense or urgency, that could either be fast motions with increasing speeds like sprinters or having a constant pace with little to no acceleration like marathon runners.
- **Flow:** the attitude towards bodily tension and control, explained as moving tight and very held in contrary to moving freely like children do.

The latter effort could be distinguished from the former three, because the Flow Effort is not considered as an Effort Action. Flow describes the effectiveness of the other three efforts [15] and it is therefore chosen to keep the Flow Effort in mind, but not actively control it. The main focus will be on efforts that contribute to the action movement behavior.

One great way of presenting the different Laban Efforts is done using Laban's dynamosphere, in which all human dynamical actions take place [39]. In Figure 5.4 the Space, Weight and Time Effort are captured in a three dimensional space, such that each perpendicular axis is occupied by one of these three efforts.

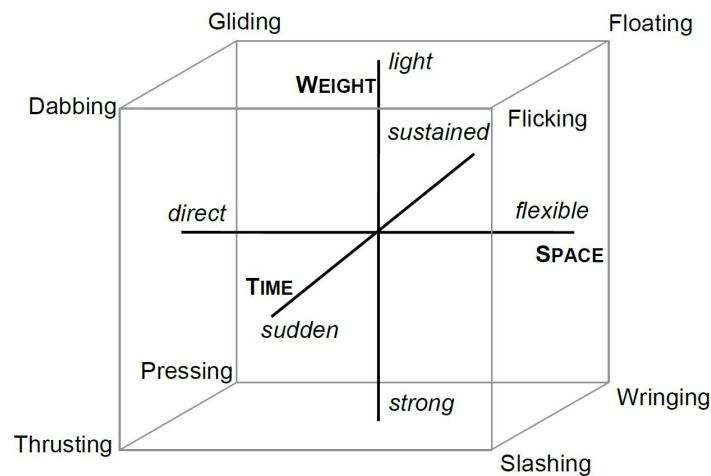


Figure 5.4: Laban's dynamosphere representing the Time, Space and Weight Effort along with their descriptive opposite elements [18]. The combination of these elements results in the Eight Basic Efforts.

Each of these three Efforts possesses two opposing elements, describing the Effort's effect on the motion. The Space Effort could either be direct or flexible depending on the path, Weight is either light or strong defined by amount of pressure and Time contains the elements sudden and sustained.

During a motion only one element of each Effort could be selected, which comes down to three elements in total. The combination of these three elements yields in another descriptive set of expressions, called the Eight Basic Efforts [40]:

- Dabbing: punching towards goal with speed and efficiency, without deadly intent.
- Flicking: a crisp, light and brief movement, like tossing of hair.
- Floating: move like flying through air or water.
- Gliding: light and graceful movement similar to ice skating.
- Thrusting: punching, but this time with deadly intent towards the goal.
- Slashing: heavy and fast move, for instance a sword slashing through air.
- Wringing: strong movement flexible in space, like wringing a wet towel.
- Pressing: pushing in desired direction with high pressure.

All names are chosen such that they make intuitive sense on reaction of the body under each condition. It can be seen that all Basic Efforts differ at least one element with all other Efforts, which makes them unique and independently visible. For instance, dabbing is a movement making direct usage of space, sudden in time and light pressured in weight, while flicking moves the same way in time and weight, but uses its space flexibly.

Now that there is a language available to describe motions according to three basic units, it will be used to adapt the trajectories the robotic arm is creating.

5.3 Verification method

Before being able to use the Laban Efforts, its relation with respect to the movement trajectories should be found. In Chapter 4 a general control law was introduced, which forms the basis for creating these movements. To better understand the relation between each Basic Effort and the way humans would move, an experiment is created. The goal of this experiment is finding a set of trajectories that will be used to perform the robotic movements.

The control framework has the freedom to choose the control gains for each part and thereby changing the effect of each component on the movement. Using this knowledge and the trajectories that are found during the experiment an optimization problem is constructed. The set of obtained control gains from this optimization is then directly related to the Laban's Basic Effort. Both the experiment and the optimization procedure are explained in further detail below.

5.3.1 Experiment

As mentioned earlier, understanding the relationship between the Basic Efforts and the movements of the end-effector is essential to control the robot. An experiment is

created with the help of dancing experts in the field of Laban Efforts and so using it on daily basis.

In order to keep the experiment easily transferable it is created on an online platform. An extra benefit of this approach is the direct usability of the obtained data. The basic idea is to draw the trajectory corresponding to each Effort, which could be compared with moving the hand from one position to the other. Drawing the movements is done using the computer mouse. Figure 5.5 shows the interactive input field of the experiment for the participants.

The bead on the left side of the screen is movable and should be brought to the cross on the right side. By simply clicking and holding the bead, it can be dragged anywhere within the screen, making it possible to draw different trajectories. Furthermore, it is noticeable that the experiment only allows motions in 2D, which makes it much easier for the user to move the bead along the desired trajectory. Besides, the robot is also capable of moving in 2D only.

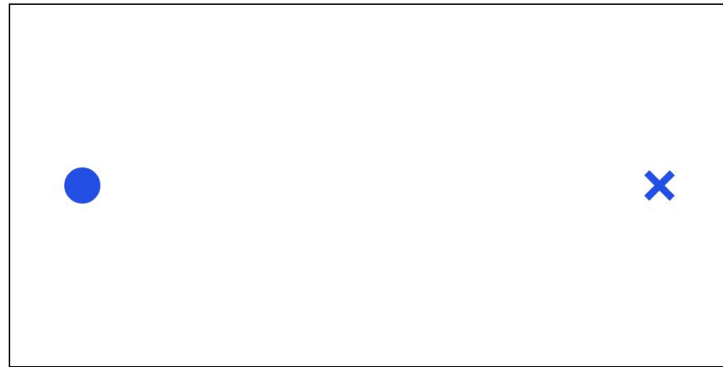


Figure 5.5: The experimental set-up created on an online platform. The participants are asked to drag the bead towards to cross making use of the mouse. The experiment is found on www.dcsc.tudelft.nl/~glopes/Danny/.

Together with the dancing experts it was found that the Weight Effort could not be captured by simply moving the bead from one place to another. Including the intensiveness of clicking on the bead, so distinguishing light and strong pressured motions, could help solving this problem. Unfortunately, regular mouse buttons are not capable of capturing this kind of information, which was the reason for excluding the Weight Effort from the experiment. The Time and Space Efforts are on the other hand traceable by analyzing the trajectories and corresponding time.

The exclusion of the Weight Effort results in joining together all Basic Efforts with contrary pressure level, so: dabbing-thrusting, flicking-slashing, floating-wringing and gliding-pressing. Figure 5.6 shows the subset of the original Laban's dynamosphere, spanning the Time and Space Efforts together with all its combinations of the opposing elements.

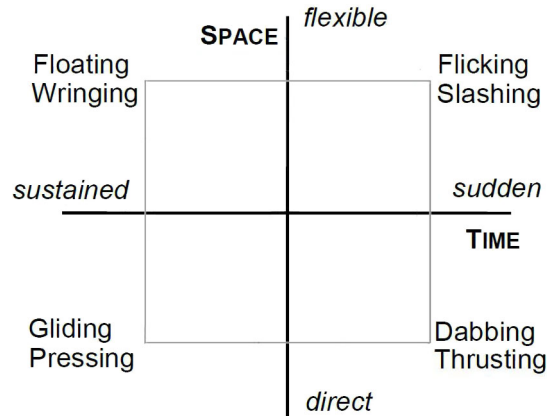


Figure 5.6: The subset of original Laban's dynamosphere, where the Basic Efforts with opposing Weight Effort are joined together.

The procedure of the experiment is as following:

1. One of the four Basic Effort sets is selected.
2. The participant is asked to draw the movement with the bead considering the selected Efforts.
3. Once the participant is satisfied with the created trajectory, the next button is pressed, which brings him to the next Basic Effort.
4. After creating four trajectories corresponding to the four Basic Efforts dabbing, flicking, floating and gliding the user has finished the experiment.

The stored data obtained from the participant will be used to derive a general trajectory for each Basic Effort. The entire experiment can be found on www.dcsc.tudelft.nl/~glopes/Danny/. The results of this experiment can be found in Section 5.4 and are used to derive the trajectories to be approached by the model.

As explained before the experiment identifies the elements of the Time and Space Effort in its trajectories. The relative path traveled with respect to the shortest distance between initial and desired position was conceived instead of the overall distance traveled.

The relative path could be calculated as the cumulative sum of distances between previous and current positions divided by the minimum path

$$Z(t) = \frac{\sum_{k=0}^t \sqrt{\Delta x(k)^2 + \Delta y(k)^2}}{\|X_{\text{des}} - X_0\|}, \quad (5.1)$$

where $Z(t)$ represents the relative distance at time t .

The reason for looking at the relative path is to keep the observed metric general, even when the distance between initial and desired position varies. Besides, movements in 3D could also be captured using this expression. The reason is that the distance between initial and final position is still measurable, so the relative distance still holds with respect to this shortest distance. Values of Z greater than 1 for correspond to a more indirect way of moving from A to B. Figure 5.7 shows the difference between direct and flexible movements, where final values much greater than $Z = 1$ represent higher flexibility.

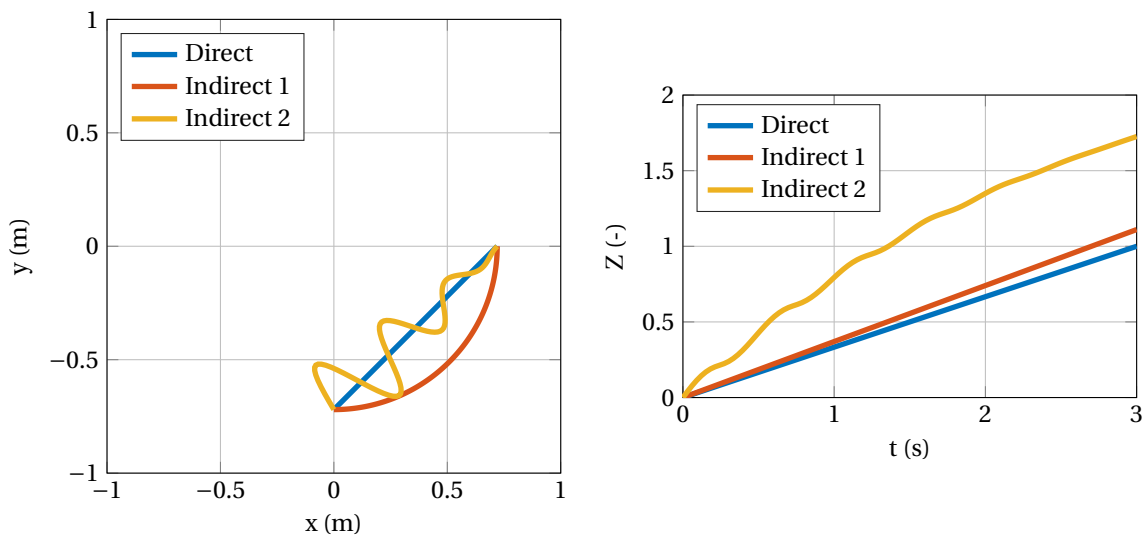


Figure 5.7: Example of different trajectories in the workspace and the relative path trajectories over time. Values larger than 1 on the y-axis correspond to more indirect movements towards the desired position.

Both the trajectories between initial and final position in the workspace as well as the relative path's progress over time are shown for three different motions. It should first of all be noticed that the direct motion does not exceed the bound of $Z = 1$, which means it took the shortest path. The first indirect motion is represented by a circular motion, so the relative path is slightly larger than the shortest path. Especially the second indirect motions shows a clear divergence from the short path, with even more than 1.5 times the total distance traveled.

5.3.2 Optimization

Now that the trajectories based on actual measurements are known, it is used in an optimization problem. Question remains what is there to optimize and how will the obtained experimental results help determining the optimal variables.

Let us start looking at the objective function to be optimized. The general idea here is to create movements with the robotic manipulator similar to the ones found in the experiment. The previously mentioned relative path metric supports this idea and

could easily be used to compare the simulated results with the measured results. This difference is the first part of the objective function, which makes sure that the Time and Space Efforts are taken into account for each of the combinations.

Besides, it is important that the robot is capable of reaching its desired position, such that its functional tasks is kept as well. When a robot performs a motion with the desired Efforts, but loses the ability to pick and place an object, it becomes useless. Therefore, the second part of the objective function considers the functional elements.

The objective function, composed from these two building blocks, is

$$f(\theta) = (Z_{\text{mes}} - Z_{\text{model}})^2 + (X_{\text{des}} - X_{\text{model}}(t_{\text{end}}))^2, \quad (5.2)$$

where θ corresponds to the to be optimized variables vector. The error between the measured and the simulated path, and the desired and modeled final position should be minimized. This error is constructed in a non-linear least-squares error structure. No extra weights are introduced, thus both parts are considered equally important.

In order to understand the parameters within θ that will be used to minimize the objective function, consider again the general control framework

$$\begin{aligned} u &= (1 - |H_{\text{jl}}(q)|) (u_{\text{task}} + N_{\text{task}}^T u_{\text{G}}) + K_{\text{jl}} H_{\text{jl}}(q), \\ u_{\text{task}} &= J_{\text{task}}^T(q) \left(K_{\text{p}} \begin{bmatrix} e_{\text{r}} \\ H_{\text{p}} e_{\text{p}} \end{bmatrix} + K_{\text{d}} J_{\text{task}}(q) \dot{q} \right) + G(q), \\ u_{\text{G}} &= \|X - X_{\text{des}}\| J_{\text{G}}^T(q) K_{\text{G}} \text{Sgn}_{\text{G}}(q) G(q). \end{aligned} \quad (5.3)$$

The controller is responsible for the robot's motions and manipulating this will change its behavior. Most of the elements within the control law are depending on the position and velocity, like the error functions e_{r} and e_{p} , and the different Jacobians. Nevertheless, the control gains K_{p} , K_{d} and K_{G} are constants and therefore has the flexibility to change the motion according to the requirements. A short elaboration of the effect of each control gains on the motion is given next.

Increasing the gain K_{p} makes the virtual spring between the initial and final position in the workspace much stiffer. The result is that the end-effector starts moving more abrupt and moves faster towards its goal. Besides, the motions are less accurate and potentially introduce overshoot.

The gain K_{d} controls the amount of damping when moving, and increasing this value results in the decrease of both the velocity as well as the motion's startup behavior. In fact, this part could cancel out effects introduced by the increase of the K_{p} gain. Greater K_{d} values correspond to more sustained motions, moving more precisely towards the goal.

The control gain K_{G} tunes the amount of gravity torque that needs to be fought in the configuration coordinates. Positive values means it will try to avoid the peak

values, i.e. high gravity torques on the arm. Negative values tend to move over those peaks and thereby maximizes the amount of gravity torque during the motion. Varying the K_G value results in more direct or more flexible way of using the space around the shortest path from A to B. When K_G equals zero the part will be removed, which means no diversion from this path are found.

The fourth gain K_{jl} prevents the robot from moving through the joint limits, so should be sufficiently high. Nevertheless this gain does not necessarily change the motion itself and it is therefore not considered as an optimization parameter.

The goal now is to find the most optimal set of control gains with respect to the Basic Efforts. The gains are subdivided into diagonal matrices such that motions in x-direction could be distinguished from motions in y-direction. It is furthermore chosen to keep the gains controlling the polar angle position and velocity constant, while this part of the control law will only be active when approaching a potential local minimum. Finally, the gravity torque gains is presented as a diagonal matrix, whereby it directly effects the configuration coordinates.

The control gains in matrix form will look as

$$K_p = \begin{bmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & C_1 \end{bmatrix}, \quad K_d = \begin{bmatrix} K_3 & 0 & 0 \\ 0 & K_4 & 0 \\ 0 & 0 & C_2 \end{bmatrix} \quad \text{and} \quad K_G = \begin{bmatrix} K_5 & 0 & 0 \\ 0 & K_6 & 0 \\ 0 & 0 & K_7 \end{bmatrix}, \quad (5.4)$$

where K_i and C_i corresponds to the optimization parameters and the non-optimized constants respectively. It is now shown that the set of optimization parameters θ is constructed as

$$\theta = \begin{bmatrix} K_1 & K_2 & K_3 & K_4 & K_5 & K_6 & K_7 \end{bmatrix}. \quad (5.5)$$

The optimization problem minimizing the error of Equation (5.2) using the optimization parameters, looks like

$$\begin{aligned} \min_{\theta \in \mathbb{R}^7} \quad & f(\theta), \\ \text{s.t.} \quad & 0 \leq \theta_i \leq 500 \quad \text{for } i = 1, 2, \dots, 4, \\ & -2 \leq \theta_j \leq 2 \quad \text{for } j = 5, 6, 7, \end{aligned} \quad (5.6)$$

where the lower and upper bounds are introduced to keep the control from saturating.

The optimization problem is solved using the least-squares solver of MATLAB called `lsqnonlin`. It was found that the set of solutions contains quite some local minima. The global minimum is found with a multi-start procedure, which runs the optimization problem multiple times using different initial values for the control gains. The multi-start least-squares optimization then selects the result with the lowest residual value from the set of local minima.

The optimization tries to find the set of control gains according to trajectories of the Laban Basic Efforts dabbing, flicking, floating and gliding. The total of 12 sets of initial and desired positions, covering the robot's entire reachable area, was created. Each Basic Effort will obtain 12 sets of control gains. Eventually, the optimization procedure had to run 48 times to find all control gain sets for all four Efforts used in the experiment. The results of this optimization and the way the control law could be generalized according to the results is given in Section 5.4 and 5.5.

5.4 Results and analysis

The experiment explained earlier was performed by dancing experts, being able to easily understand and transfer their knowledge about Laban Efforts. The relationship between end-effector motions and Basic Efforts was captured in the experiment. Besides, non-professional users have also participated in the experiment, to capture the hand movements they think correspond to each of the four Efforts. It was found that this group of people was also very capable of conducting the experiment the same way as the experts did. The motivation for using this second group as well comes from the fact that everybody should be able to understand the conveyed information by the robot. The results of the experiment are shown in Figure 5.8 and 5.9.

Figure 5.8 seems to look unclear at the beginning, but some trends could be distinguished quite well. First of all, the difference between flexible and direct motion is found in the deviation from the middle line when moving flexibly. The lines in the dabbing/thrusting and gliding/pressing Effort are compressed in a small band around the shortest path. The floating/wringing and flicking/slashing Efforts on the other hand move freely around in the x-y space, which makes these motions more indirect and flexible. The difference between the latter two groups mentioned is the spikiness of the flicking/slashing Effort opposed to the well-rounded motions in the floating/wringing group. This difference could be explained by the sudden-sustained antagonist, as shown better in Figure 5.9.

The distance traveled plots seen here shows clear differences between sudden and sustained motions. The flexible, sustained Effort floating/wringing takes more time to reach the goal with respect to the flicking/slashing motion. Besides, the dabbing/thrusting Effort moves much faster than the gliding/pressing Effort.

It is furthermore shown that the flexible Efforts generate motions that are 2 to 3 times longer than the shortest distance between initial and final position. The direct motions stop around $Z = 1$, so these trajectories seem to follow the shortest path quite well.

To summarize, the observations from the experimental shows that sudden motions reach the desired location earlier than sustained motions. Furthermore, the distance traveled plots for flexible motions are significantly larger than for direct motions. The trends seen in the experiment could be generalized, such that they could be used in the

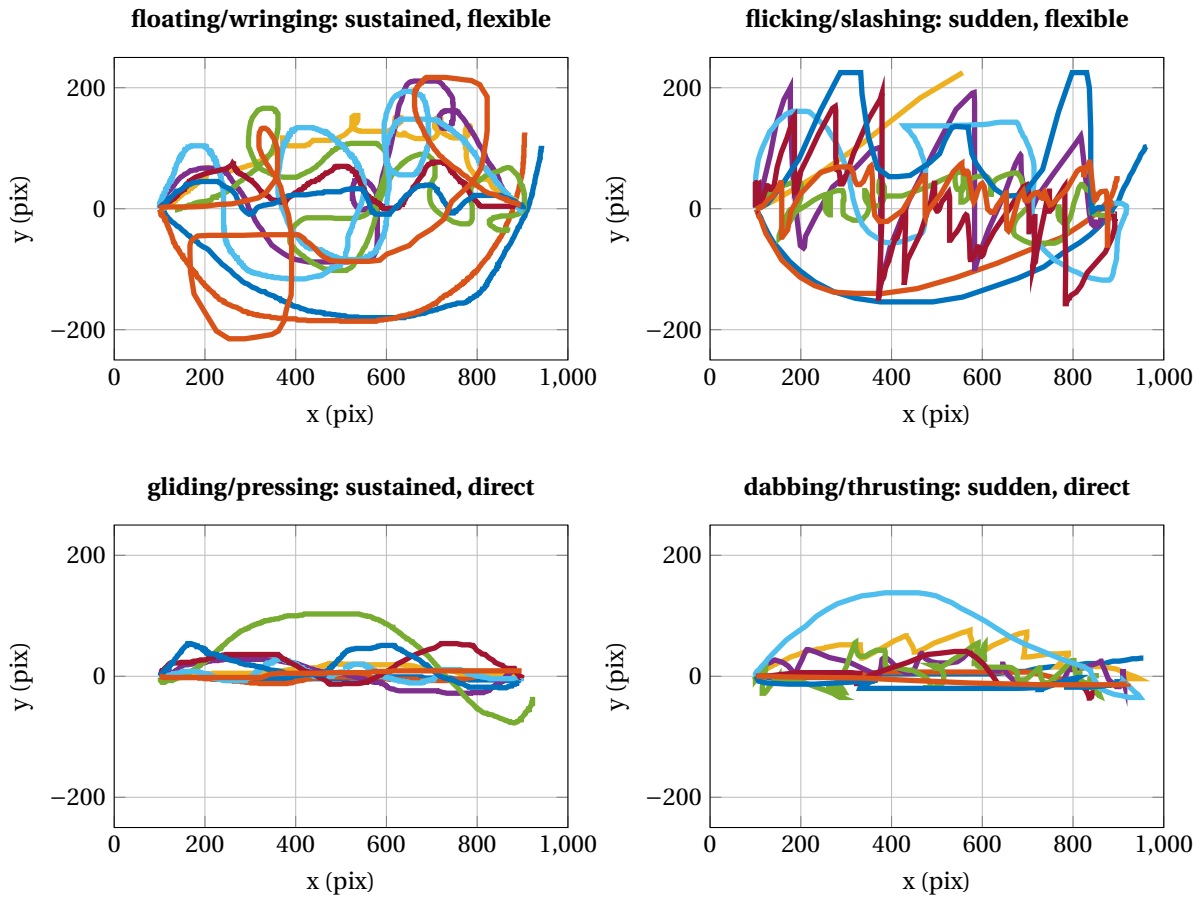


Figure 5.8: The experimental results in the workspace obtained from several participants that were asked to move the bead to the cross. Each Basic Effort is separately presented, distinguishable by their Time and Space Effort element.

optimization procedure. Figure 5.10 represents the generalization of the experimental results. Here the Efforts dab and flick are both sudden, faster motions, either direct or flexible in space. The other two motions are more sustained, gradual growing in time, and again floating Effort is flexible and flicking more direct use of space. The set-point of $Z = 1.3$ is taken as an average of the distance traveled, so flick and float move about 1.3 times the shortest path.

The experiment concluded that four different Basic Efforts are generalizable into the four trajectories of Figure 5.10. These trajectories are used as $Z_{\text{mes}}(t)$ in the objective function of Equation (5.2). The optimization problem of Equation (5.6) results in approaching the trajectories by the model.

Like explained earlier, an optimization was used to find the most optimal control gains belonging to the different four Basic Efforts. A total of 12 motions in the workspace were created, resulting in a set of 12 control gains for each effort. Figure

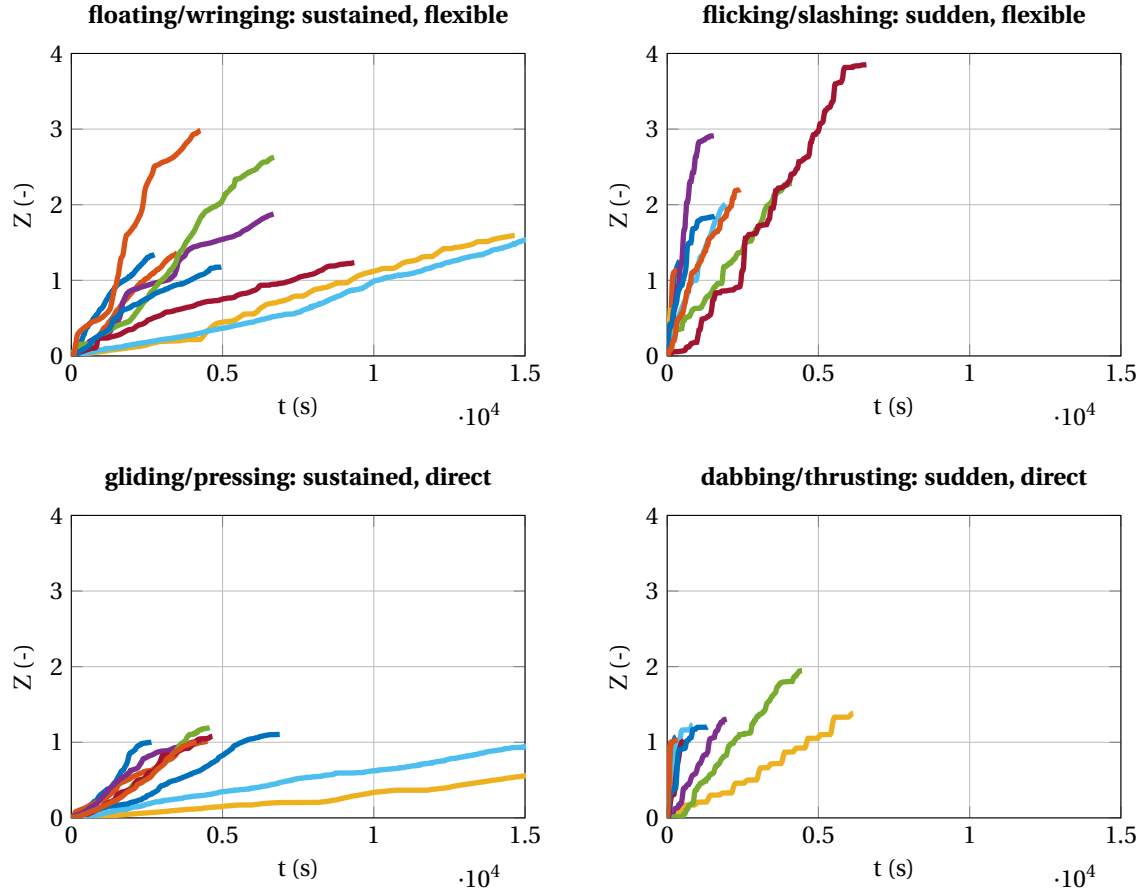


Figure 5.9: The experimental results of the relative distance traveled over time, obtained from several participants that were asked to move the bead to the cross. Each Basic Effort is separately presented, distinguishable by their Time and Space Effort element.

5.11 shows the obtained control gains for dabbing, flicking, floating and gliding of all 12 optimizations. The x- and y-axes of the plots correspond to the gains in Equation (5.4).

The gains K_1 and K_2 , plotted in the top left corner, correspond to the stiffness of the spring between current and desired location. Therefore these gains represent the injection of energy into the system. The plot on the top right decelerates the system by extracting energy, which is in physical terms the higher the gains K_3 and K_4 the stronger the damper. Both upper figures relate the x-axis with the x-direction in workspace and y-axis with the y-direction. As an example, most of the flicking motions show both low spring and damper constants in the x-direction, while in y-direction the spring is stiff with medium damping.

The three plots regarding K_5 , K_6 and K_7 control the amount of gravity torque fighting or contributing to the motion. These gains control the configuration coordinates q_1 , q_2 and q_3 respectively. A positive control gain means that the manipulator's

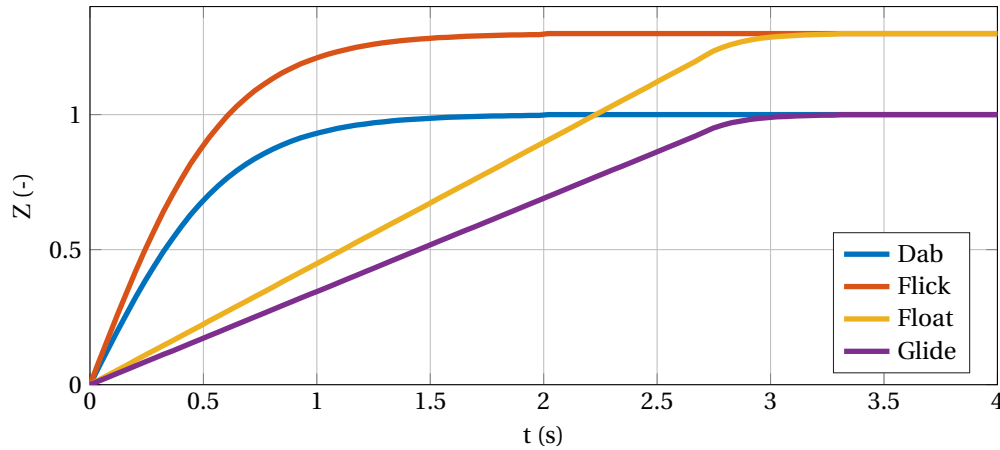


Figure 5.10: Generalization of the experimental results used as optimization input. Sudden motions reach the desired position faster and the distance traveled is 1.3 times the shortest path for flexible movements.

part minimizes the gravity torque. Negative values contribute to configurations with maximum gravity torque, so energy least efficient motions. These terms could be very useful when rerouting energy, such that path between initial and final position changes accordingly.

Besides, it is shown in Figure 5.11 that some of the optimized gains are clustered quite well, which means the optimal values for different movements under the same effort are similar. Next to these optimized values, the mean values are shown using an X in the figure. These average values should indicate the centers of the clusters given in the figure. The better the gains are clustered, the more accurate the generalized gains represent the, to be conveyed, Effort.

Table 5.2 shows the calculated mean values, as seen in Figure 5.11, for each gain in the Basic Efforts dabbing, flicking, floating and gliding. It can be seen that the gains K_1, \dots, K_4 are greater than zero, while the last three gains are located between -2 and 2 .

Noticeable is the relation between the spring and damping term in the same direction, so $K_1 - K_3$ and $K_2 - K_4$ gains. Higher values for $K_1 - K_3$ relates to higher spring and damping term in the x-direction. It can furthermore be seen that the damping terms for the sustained motions (floating and gliding) are much higher with respect to sudden motions. The two flexible motions (flicking and floating) seem to have higher gains controlling the gravity torque. Deeper understanding about these phenomena will be given in Section 5.5.

The average control gains could now be used as a generalization for the four Basic Efforts: dab, flick, float and glide. Figure 5.12-5.15 show each trajectory for four different motion Efforts, by specifying the initial and desired location only. The end-

Table 5.2: K-clustered mean gain values that correspond to the four different Basic Efforts dabbling, flicking, floating and gliding.

	K_1	K_2	K_3	K_4	K_5	K_6	K_7
Dabbling	382	310	184	145	-0.15	0.41	-0.35
Flicking	92	436	19	160	0.51	0.18	0.46
Floating	181	246	77	304	0.76	-1.30	1.49
Gliding	214	173	413	258	-0.11	-0.39	0.10

effector movements in the workspace are given together with the trajectory of the taken path over time. In Figure 5.12 and 5.13 the upward and downward motion between $[0, -0.72]$ and $[0.5, 0.5]$ is shown, while Figure 5.14 and 5.15 represent a vertical motion between $[0.5, -0.5]$ and $[0.5, 0.5]$. In total 12 motions were created using the generalized sets of gains, which could be found in Appendix A.

Finally, the most interesting motion that could be created using the generalized gains is the Labanotation input introduced in Section 5.1. The robot should be able to perform this small dancing sequence, while possessing a dabbling, flicking, floating or gliding effort. As a result, four dancing motions are created differently and should be distinguishable by the audience.

Figure 5.16 shows both the movements in the workspace coordinates as well as the trajectory development relative to the shortest path over time. The x-, y-movements are shown in separate figures, so that the trajectories are clearly visible without overlapping each other. The dabbling and gliding motions seem to approach straight lines between the two positions, while flicking and floating are moving around in the workspace.

The different distance traveled plots could on the other hand be shown clearly in only one figure. The dotted lines reflect the values for the shortest paths between initial and desired position. It can be seen quite well that the direct motions, dabbling and gliding, remain close to these shortest path indicators, while the flicking and floating motion exceed these values. Besides, gliding and floating need more time to reach the final position, which explains the sustained motions. The continuation of the analysis is given in Section 5.5.

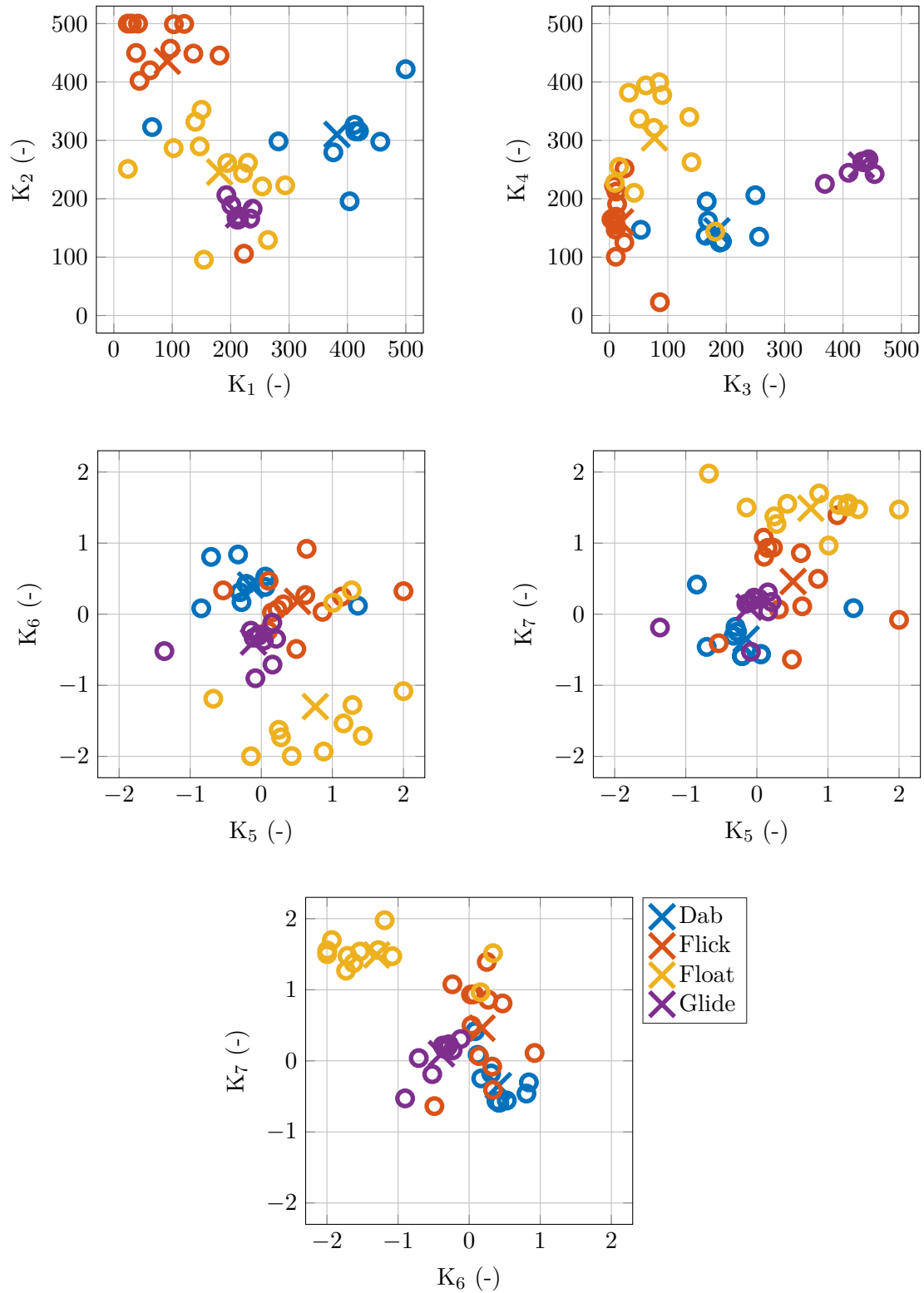


Figure 5.11: The resulting clusters after running the optimization procedure over 12 motions for each Effort. The o represent the optimized control gains and the X shows the mean values for each cluster.

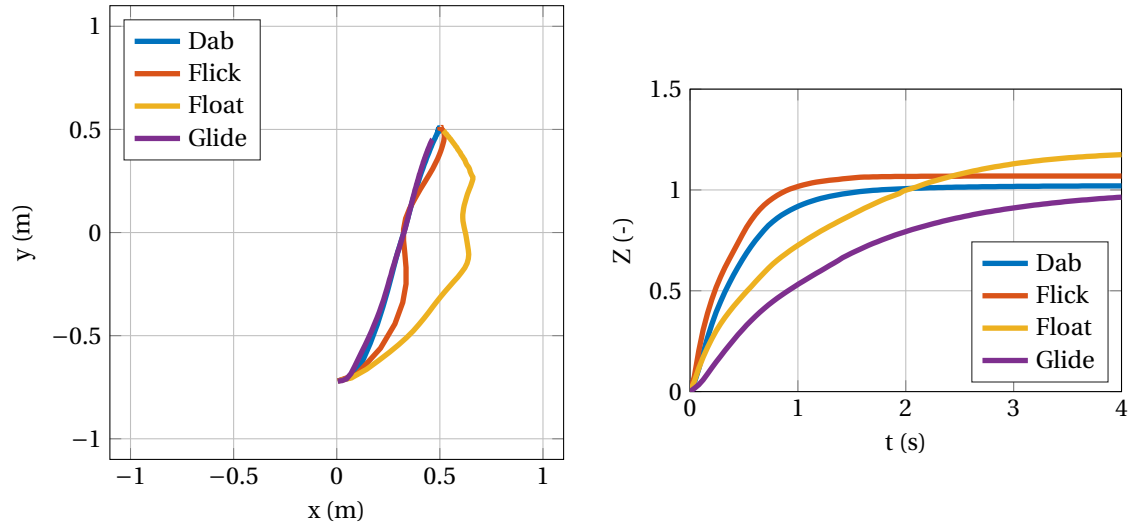


Figure 5.12: Movement 2 using the generalized set of control gains, from $[0, -0.72]$ to $[0.5, 0.5]$.

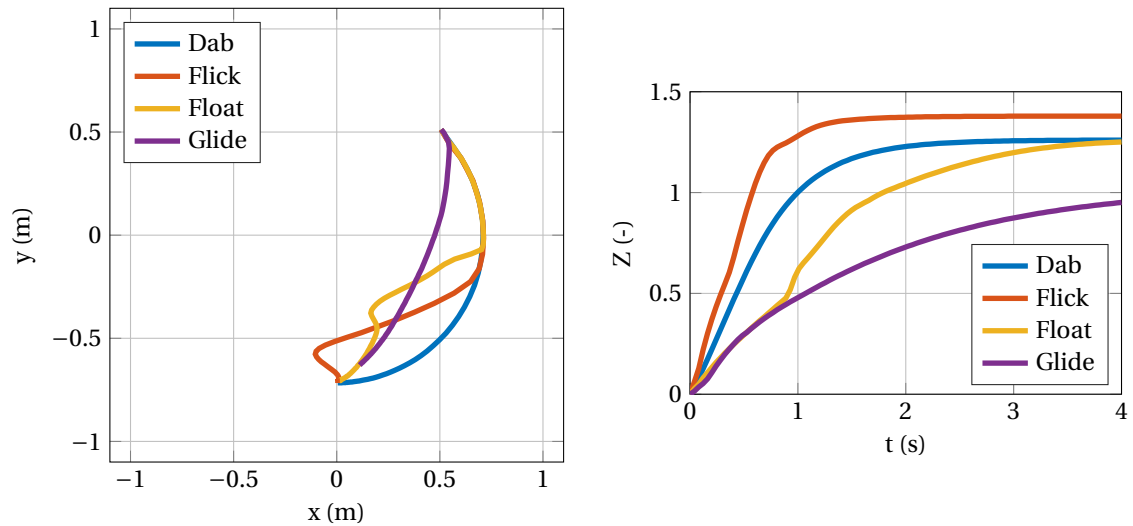


Figure 5.13: Movement 5 using the generalized set of control gains, from $[0.5, 0.5]$ to $[0, -0.72]$.

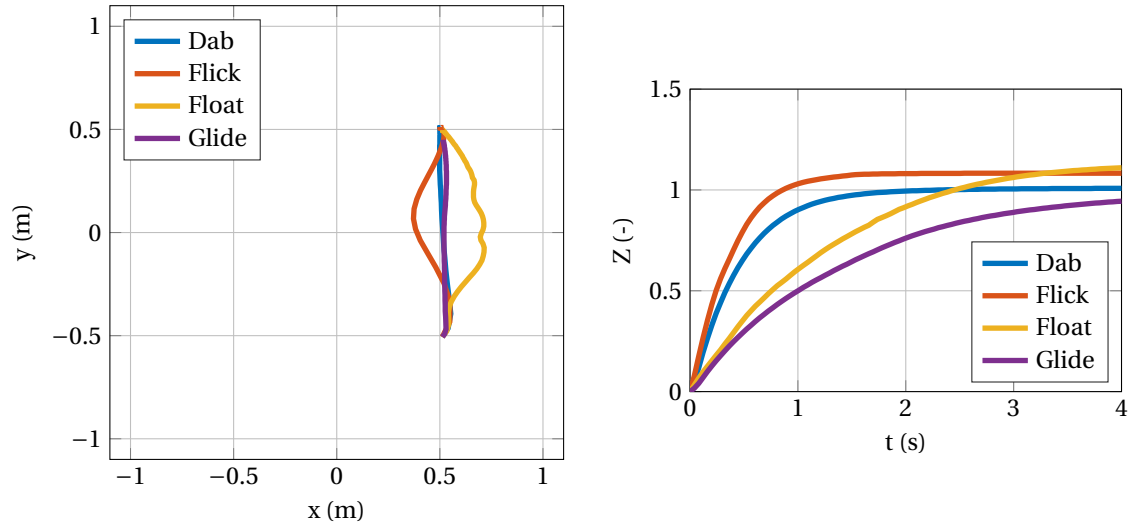


Figure 5.14: Movement 7 using the generalized set of control gains, from $[0.5, -0.5]$ to $[0.5, 0.5]$.

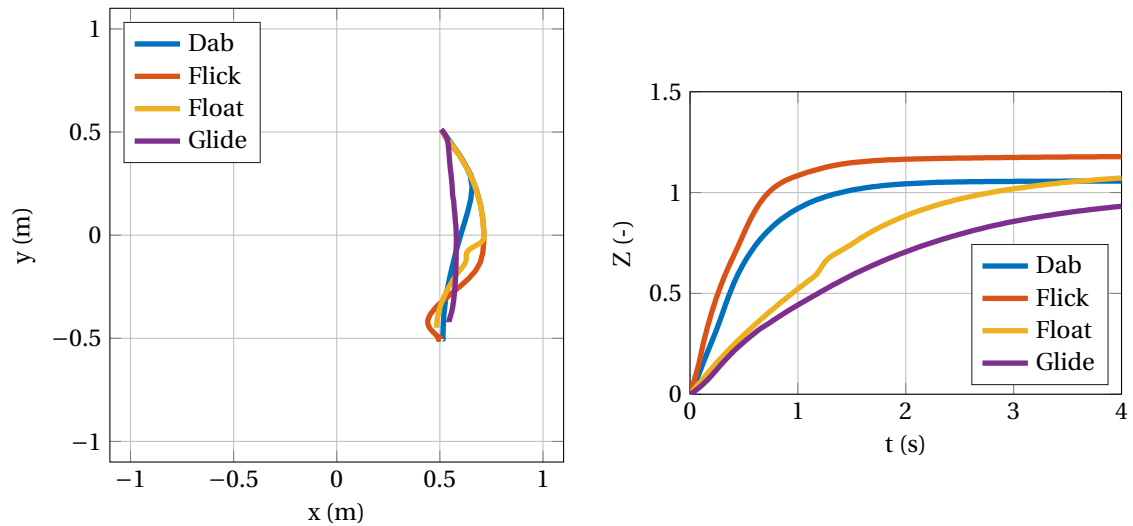


Figure 5.15: Movement 9 using the generalized set of control gains, from $[0.5, 0.5]$ to $[0.5, -0.5]$.

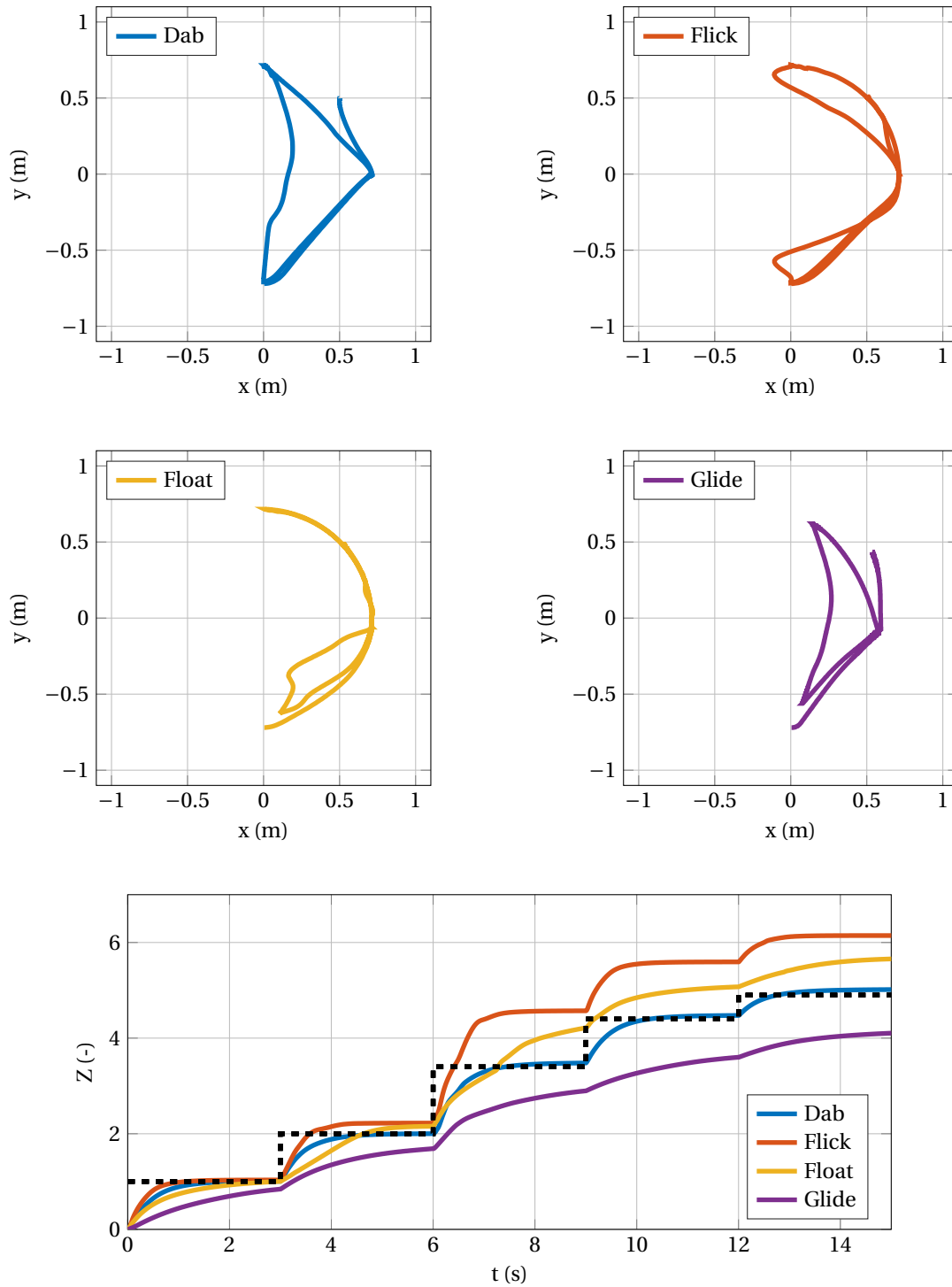


Figure 5.16: Movement sequence using the Labanotation input for the four different Basic Efforts. The robot starts at $[0, -0.72]$ and stops its motion at $[0.5, 0.5]$. The dotted line in the lowest figure corresponds to the shortest distance between each intermediate initial and desired position.

5.5 Discussion

The results shown in previous section will now be discussed further, focusing on the generalization per Effort and the effect on the robot's motion. The goal to convey non-verbal information using the difference between the trajectories is discussed and it is questioned whether this goal is achieved or not.

Consider again Figure 5.11 showing the optimization results belonging to 12 different motions for each Basic Effort, founding using a least-squares error procedure. The mean value of each cluster is calculated as the centers for each of the gains and Efforts. The top left figure representing spring constants in both x- and y-direction shows four clearly distinguishable groups of gains. The Efforts dabbling, flicking and gliding are compressed quite well without much dispersion. The floating effort shows some scattering in the x-direction, but the general tendency is visible.

Looking at the damping constants, shown on the top right of Figure 5.11, the dabbling, flicking and gliding efforts are again clustered well and clearly observable from each other. The floating effort shows some outliers from the general group, which could be excluded to calculate the average gain.

Finally, the gravity torque gains show some clear clusters. The floating Effort is deviated quite some from the other three clusters. It should be noted that the direct motions are both better clustered as well as closer to zero than the flexible motions.

Task gains As mentioned earlier, the relation between gains $K_1 - K_3$ and $K_2 - K_4$ is given by the x-, and y-direction of the motion respectively. This relation between spring and damping constants is useful when looking at the effect of these control gains on each Basic Effort. First of all, it is shown in Table 5.2 that K_1 and K_3 have lower values for flexible motions compared to direct motions. These values gives the arm freedom to move into the x-direction driven by other forces, so it introduces flexibility in the motion.

The ratio between K_1 and K_3 is also quite interesting to look at, as shown in Table 5.3. It could be seen that this ratio for sudden motions is much greater than their sustained counterparts in Space. Higher ratio means that the spring is much stronger than the damping term, causing the robot to move quicker and more abrupt.

Moving on to the gains K_2 and K_4 , it is noticeable in Table 5.2 that both gains are higher when looking at the flexible motions flicking and floating with respect to dabbling and gliding. As a result the forces to the end-effector in y-direction are greater, so less freedom to adjust the motion by gravity torque forces. Therefore, this observation is contradicting the difference between flexible and direct motions. Nevertheless, these higher control gains for flicking and floating Efforts are necessary to reach the task in the workspace.

The ratio K_2/K_4 on the other hand is much greater for sudden motions as could

Table 5.3: Ratios of the mean gain values in x-, and y-direction for to the four different Basic Efforts.

	K_1/K_3	K_2/K_4
Dabbing	2.1	2.1
Flicking	4.8	2.7
Floating	2.4	0.81
Gliding	0.52	0.67

be seen in Table 5.3. The effect of increasing this ratio is creating a stronger spring with less damping again, but now in the y-direction. Therefore, faster and more abrupt motions can be created at this increased ratio, resulting in more sudden way of moving.

Gravity torque gains The last three gains K_5 , K_6 and K_7 control the level of fighting against gravity torque. The groups shown in Figure 5.11 have more overlap between the different efforts, but eventually four clusters are distinguishable. The overlap makes it harder to generalize, while the mean values do not differ much. As seen in Table 5.2 some of the calculated mean values are much alike. Besides, the dispersion around these averages is relatively large when looking at the results for the flicking and floating Effort. Nevertheless, some conclusions about these clusters could be drawn.

First of all, it is notable that the gains of the motions dabbing and gliding are somewhat closer to zero, so the gravity torque term vanishes from the control input. This could explain why these motions are more direct, because the first part of the control law drives the manipulator only. In addition, as just mentioned, the flicking and floating effort seem to variate more, which means these gains have higher influence on the trajectory between initial and final position.

The mean values of gain K_5 , as shown in Table 5.2, for direct motions are negative, quite small numbers with respect to the flexible ones. These small negative values are not influencing the gravity torque control part and are therefore negligible. The flexible motions show larger positive values for K_5 , thus tries to avoid the horizontal position of the first link. As a result, the arm is moving around a bit more when crossing the horizontal axis and is so more flexible.

The gain K_6 tries to keep the forearm close to the body when represented by an high positive value. Negative values pushes the forearm away from the body, towards the edge of the reachable area. The dabbing and flicking motion have positive values for this gain, which makes the arm better maneuverable while the center of mass is closer to the body. Thus, these motions should be moving more sudden in time.

Finally, gain K_7 is larger for flexible motions, which increases the effect of the controller on this end-effector configuration. During the movement higher gains K_7 causes the lack of directness, because direct movements violates the minimum gravity

torque. The gain for dabbing motions is negative, which seems to compensate for the elbow's motion, making it possible to move in a straight line. K_7 for the gliding effort is small enough, so could be neglected during the motion.

Point-to-point trajectories 1 Next, the four example movements of Figure 5.12-5.15 are analyzed further, among which the difference between downward and upward motions is discussed. All of these motions were created using the generalized set of control gains, as given in Table 5.2. In total 12 motions were created and all results could be found in Appendix A.

Consider Figure 5.12, the four different Efforts are shown, both the trajectory in the x-y frame as well as the distance traveled is given. It can be seen on the left that Basic Effort dab and glide move in a straight line towards the goal, while flick and float wiggle around in the workspace before reaching the desired position. This is confirmed by the figure on the right, where dab and glide approach the line with $Z = 1$, but the other two are exceeding this value. Like explained earlier, the greater the value of $Z(t)$, the more the end-effector deviates from the shortest path. Thus, this figures clearly visualizes the difference between direct and flexible motions.

Furthermore, the figure on the right shows difference in movement time between the two direct movements, even though the figure on the left is not able to capture this. Having a look at the dabbing and gliding effort, it could be seen that the first motion reaches its goal in about 1.5 seconds. Gliding is at the end of the simulation at 4 seconds still moving towards its goal. Besides, the gliding motion accelerates more gradually, which is not the case for dabbing. The same observation is made for the two flexible motions, where flicking starts and stops moving more abrupt. Therefore, next to differentiating direct and flexible movements, the difference between sudden and sustained motions is also confirmed.

Figure 5.13 shows the movements for the same Efforts, but now in opposite direction by reversing the initial and desired location. It is surprising to notice that there is a clear difference between current and previous motions. Especially the dabbing effort performed a straight-lined motion before, but now moves in a partial circular shape towards the goal. The distance traveled on the right shows that the motion is almost $1.25\times$ longer than the shortest path. It was found that the joint limits prevent the robot from moving in a straight line. An initial angle in positive direction of the end-effector solves the indirectness of the motion, but the motion no long starts at the initial location of $[0.5, 0.5]$.

The figure also shows that gliding moves in kind of a straight line, while flicking and floating deviates quite some from this path. The difference between sudden and sustained motions is also clearly shown, where dabbing is much faster than gliding and flicking than floating. Nevertheless, the floating motion accelerates halfway during the motions, this undesirable change of velocity makes the motion faster and thus more sudden. The joint limits together with the gravity torque control causes the robot to

suddenly change direction and thereby the velocity. In conclusion, the flicking and gliding Efforts perform well, floating performs OK, but the increase in velocity makes the robot look more sudden. Finally, the free use of space of the dabbing Effort gives the motion certain flexibility, which makes this requirement not met.

Point-to-point trajectories 2 The motions shown in Figure 5.14 with initial position $[0.5, -0.5]$ and final position $[0.5, 0.5]$ distinguishes the four Efforts. On the left, the dab and glide Effort again travel in a straight line, where flick and float move around in space. Again, the difference between sudden and sustained motions is shown on the right, so dab and flick are much faster and abrupt than float and glide. All four motions perform well considering the Laban Time and Space Efforts.

The returning movements of Figure 5.15 differ again from the previous motions, but not as bad as seen in Figure 5.13. First of all, it can be seen that dabbing is still not moving in a straight line, but performs much better than before. The distance traveled is only slightly higher of about $1.05\times$ the shortest path. The joint limits is the main disturber influencing the movement again. Floating and gliding both gradually cover the trajectories traveled, but show a clear difference in paths as desired. Flicking proves to be both sudden in time and flexible in space. In the end, except for the small deviation of the dabbing motion, all four Efforts meet the requirements regarding workspace trajectories and suddenness.

Sequence trajectories Finally, consider the Labanotation input as shown in Figure 5.3 performed by the robot under four different Efforts as seen in Figure 5.16. Like said earlier, the four upper figures show the trajectories in the workspace and the bottom figure presents the distance traveled over time. Having a look at the workspace trajectories first, it could instantaneously be noticed that the dabbing and gliding move in much straighter lines than the other two Effort. This observation is desired, while it proves the difference between direct and flexible motions. The flicking Effort dances around in the workspace, making it indeed flexible in space. The floating Effort examines the edge of the reachable area, which is indirect towards the goal, but could not necessarily be called flexible in space.

Having a look at the development of the traveled paths over time, some of these observations could be confirmed. The dotted lines in the figure represent the recurring minimum distance between each initial and final positions. It can be seen that dab stays quite close to this minimum distance line at all set-points, so these are direct motions. Gliding does not even reach the minimum distance line, while the desired position is changed before the arm is able to reach it. This clearly shows the difference between dabbing and gliding, where the first Effort is sudden in time, while the last term only creates sustained motions.

The other two Efforts seem to exceed the minimum distance lines for each set-point, making them indirect in their use of space. Notice however that the first part of the

floating Effort stays under the minimum path line, while it was still moving towards the desired position when the set-point changed. This early change of set-points gives the appearance that the motion is direct, but increasing the simulation time shows that it does exceed the minimum distance line. Furthermore, both flick and float Efforts keep growing in distance traveled over time, indicating the indirectness in the motions. The difference between those two is the time it takes to get to the final position, where flicking moves much faster and sudden than floating movements.

In conclusion, the direct motions move in largely straight lines towards each desired set-point, while the flexible motions mainly deviate from this paths. Next to that, the movements for sustained motions are less abrupt and slower than the sudden motions, showing the clear difference in use of the Time Effort. The simulation results in a clear variate between the four Basic Efforts dabbing, flicking, floating and gliding.

Physical implication The question remains whether these four Efforts could be used to convey information by using movements only. In order to answer that question, consider again the definition of each of the Efforts as shown in Section 5.2. Here several examples will be given on how these Efforts might be useful when for instance looking at a pick and place robot.

Dabbing is a fast punch towards the goal, but not having the weight to damage anything or anyone, making it efficient yet save. This could be the standard motion, nothing much to convey except that everything is alright.

Flicking is fast, free motion, making it uncontrolled and potentially dangerous for the user. The effort could be used when conveying that there is a problem with the machine and people should not get nearby as long as it is still running.

The act of flying trough air is expressed with the floating Effort, being free in space and sustained in time. This sense of freedom could be interpreted as the relief of finishing the job soon, for instance when the machine runs out of objects to be picked.

Finally, the gliding Effort is direct, but sustained, making it less efficient than the dabbing motion. This Effort could convey the level of energy in the robot. When the robot is performing gliding motions it is fatigued from the job, meaning that it runs out of batteries.

5.6 Synopsis

This chapter has shown how the control structure of previous chapter could be used to convey information through motion. First, an introduction to the Labanotation was given, a great way to describe and structure human-like motions. The different shapes and shades could be useful to generalize the to be conducted sequences for the robotic manipulator. As an example, an input sequence based on a dancing exercise called *Port de Bras*, was created to compare the trajectories conveying different information.

Next, a brief introduction into the Laban Movement Analysis was given, where the Laban Efforts emphasize on moving differently under different according the Space, Weight, Time and Flow Effort. Each of the Efforts has its own contrary elements explaining the effect on the motion. Combining these elements results in Eight Basic Efforts expressing different behavior during movements: dab, flick, float, glide, thrust, slash, wring and press.

In an experiment the relationship between The Basic Efforts and the trajectories of the human's hand was found. The results revealed that four motions are distinguishable, indicated as sudden/sustained (Time Effort) and direct/flexible motions (Space Effort). The moving time and distance traveled proved to be great measures to generalize the Effort's trajectories.

An optimization procedure, with the goal to obtain the control gains, results in 12 sets of gains for each Effort. The mean values of each group represents one general set of control gains per effort, moving differently corresponding to their Time and Space Efforts.

The simulations, using the generalized sets, showed that the motions clearly differ from each other. Direct and flexible elements both meet the distance traveled requirement, although certain direct movements were redirected due to obstruction of the joint limits. Furthermore, sudden and sustained motions are distinguishable by the abruptness and the time to reach the goal. In conclusion, the difference in trajectories of the four Efforts dab, flick, float and glide are proven to be significant and could therefore be used to convey non-verbal information.

Conclusions and recommendations

The goal of this research is to propose a control framework for non-verbal communication by robots. This chapter will give an overview of the main findings as well as the contribution to the field of social robotics. Finally, a number of recommendations is given, as the current research findings have great potential to be a significant addition to the existing literature.

6.1 Conclusions

Currently, robots are very capable of executing tasks like picking and placing objects, but human-robot communication remains a more challenging topic. As robots are becoming more involved in our daily lives, a new dimension to these functional robots should be added, which is investigated in this study. Where previously only the task itself was important, this research considers the trajectory the robot follows towards its destination as an equally important aspect. Therefore, the goal of this research was to create a control framework that has the flexibility to convey information through non-verbal communication, by adapting the movement trajectories. The goal was accomplished by achieving the following three subgoals:

1. Find and analyze the most human-like controller by creating new human-like motion metrics.
2. Design a general control framework to conduct human-like motions, based on previous results and metrics.
3. Construct a novel way to generate different motions using only one control framework.

In the proposed framework only feedback controlled motions are considered only, because humans are able to move without learning all movement sequences in advance

as well. Limitations of this work include the focus on movement only, excluding facial expressions, eye gaze and speech. Moreover the research has been shifted towards non-affective content, as opposed to previous conducted research in the DCSC department at the Delft University of Technology. In reality both non-affective as well as affective communication takes place, so these researches should be merged together.

In the first place, it was found that among the different created metrics, the distance traveled metric describes motions with the greatest human-likeness. Humans generally tend to minimize the path between the current and desired position, which was clearly identified using this metric. Furthermore, the gravity torque metric has shown to be a great measure for expressing human-likeness, since humans implicitly optimize the metabolic power. According to these metrics, the PD controller in the workspace has proven to be the most human-like among the examined controllers.

The human-likeness of the controller could be improved even further by including the gravity torque metric within the control law, minimizing the gravity fighting energy. The Operational Space Control framework was used to make sure that the robot still reaches its goal. Besides, including joint limits prevents unnatural configurations during the motion from happening. These limits caused local minima to appear, which is solved by including an extra coordinate, the polar angle. Finally, the Lyapunov stability check was performed and it confirmed the asymptotic stability for any control gain in the system. All of these components combined results in the general control framework, which is able to perform different tasks while considering the human-likeness.

The general control framework was then used to generate different motions by introducing the Laban Efforts: a great way of describing and analyzing human movements. By use of an experiment the relation between the Efforts and the end-effector trajectories could be established. Combinations of the Effort elements sudden/sustained (Time Effort) and direct/flexible (Space Effort) displayed the available motion types, expressed as the moving time and the distances traveled. The obtained trajectories were then used in an optimization procedure to find the control gains belonging to each of the Basic Efforts: dab, flick, float and glide. Novelty here is the use of changing the control gains to relate motions to different efforts.

The results showed clear differences in the obtained gains and thereby its effect on the hand's motion. In general, sudden motions have higher spring-damping gain ratios than sustained motions, which allows for quicker and more abrupt motions. Besides, flexible Efforts have lower spring and damping gains in the x-direction, giving these motions more freedom to deviate from the shortest path. Furthermore, the gravity torque gains are much smaller for direct motions, which means they have less influence on the trajectory. The simulations show clear differences in distance traveled between flexible and direct Efforts. Moreover, sustained motions are found to require longer traveling times to reach the goal. The weakest performances were obtained when dabbing Effort move downwards, which resulted in non-straight lines between the initial and final po-

sition. The joint limits are preventing the hand from moving into the desired direction. In conclusion, the control law is able to generate movements with different appearances, where direct/flexible and sudden/sustained efforts are undoubtedly distinguishable.

Coming back to the main goal of this research, it was found that the created general control framework was able to conduct movements with the freedom to convey information non-verbally, by changing the control gains. Notice that no path planning algorithm was required to achieve this goal, since giving the final position contains enough information to perform the movement. The general framework focuses on a new way to create motions, not only in terms of the initial and final position, but also concerning the trajectories it follows. The emphasize on non-verbal communicating robots will increase the acceptance of robotics in society and this research contributes to this goal.

6.2 Recommendations

In order to improve the conveying of information even further, some recommendations for future work are outlined. Due to the time restriction, the following interesting improvements could not be implemented or tested:

1. **Velocity profile metric.** The presented human-like motion analysis is quite comprehensive, but could be extended by adding extra measures. One metric that is found in literature [16] takes into consideration the single-peaked, bell-shaped velocity profile of humans moving their hand. The monotonically increasing end-effector velocity, followed by monotonically decreasing velocity could potentially enhance the human-likeness.
2. **Predictive joint limit handling.** It was shown that the joint limits prevent the third link from moving in a straight line towards the goal in some situations. This could be prevented by implementing another way of handling these constraints, like a prediction method as seen in Model Predictive Control.
3. **Weight Effort implementation.** Together with dancing experts it was concluded that the experiment will not be capable of capturing the Weight Effort. One solution could be to include the intensity level by measuring the exerted force on for example a touchscreen, while the Weight Effort is defined as the amount of pressure during movement. The Weight Effort could be included in the optimization procedure, by optimizing the kinetic energy metric alongside previous results, as done in [41] and [23].
4. **Switching up and down movements.** The simulation results showed that, especially for the dabbing Effort, movements in upward direction differ from moving downward, due to the joint limits preventing the correct motion. One way

of coping with this problem is including a switching algorithm, altering between two optimized control gain sets, based on the direction in which the end-effector is moving.

5. **3D modeling.** The entire work considered movements in 2D only. Now that the concept of conveying information utilizing the Laban Efforts is demonstrated, the next step is to confirm these findings in a 3D model. On top of that, it would be interesting to create these movements on a real, physical platform, like the NAO robot [42].
6. **Real-time evaluation.** Last but not least, the movements conducted by the general control framework should be verified by expert and non-expert users, to see whether the movements are indeed human-like. This could be done both in 2D and 3D simulation as well as on the real robot.

A: Results general movements

The results of using the four general gain sets, found from optimization, to conduct 12 different motions.

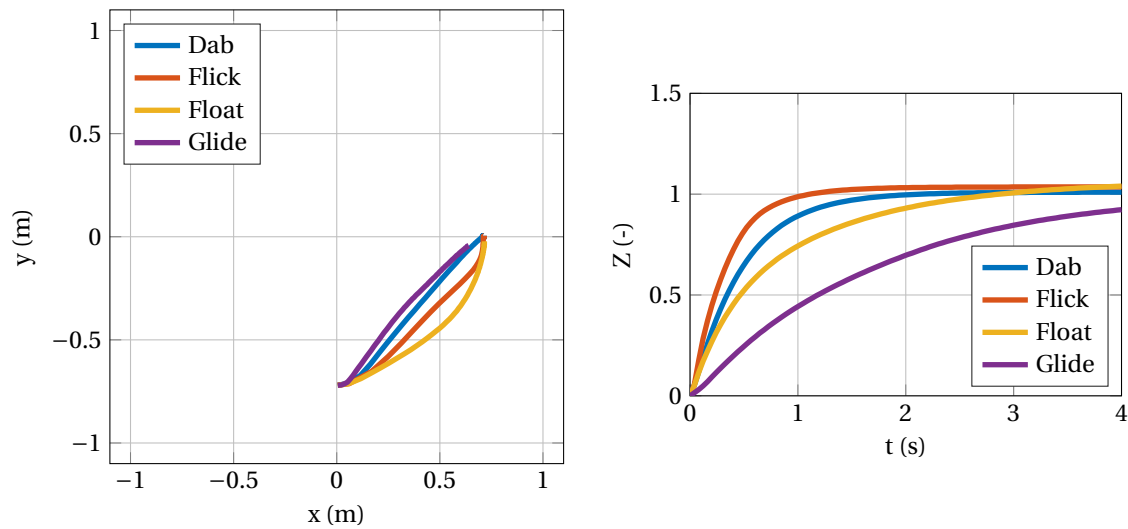


Figure 1: Movement 1 using the generalized set of control gains, from $[0, -0.72]$ to $[0.72, 0]$.

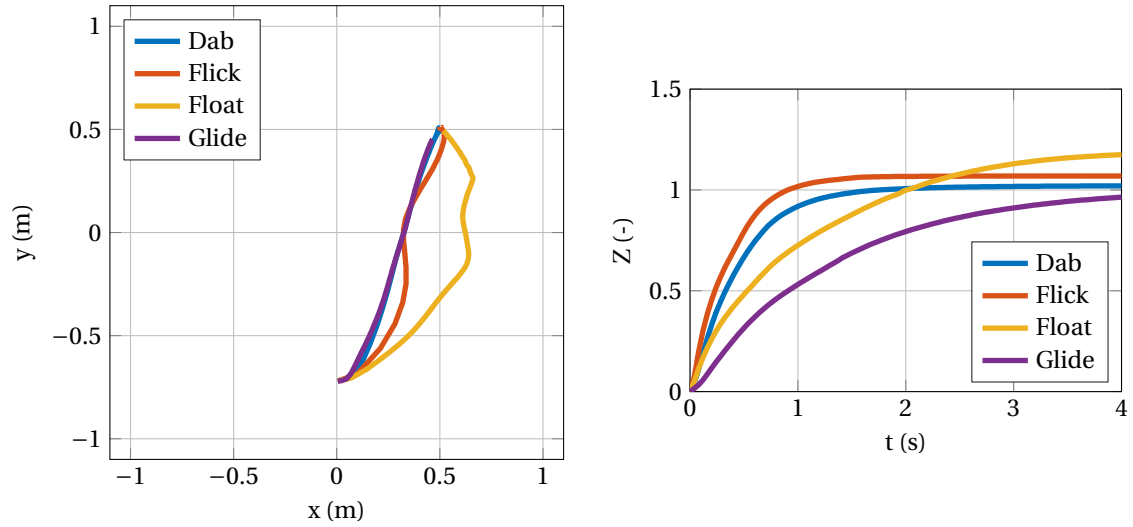


Figure 2: Movement 2 using the generalized set of control gains, from $[0, -0.72]$ to $[0.5, 0.5]$.

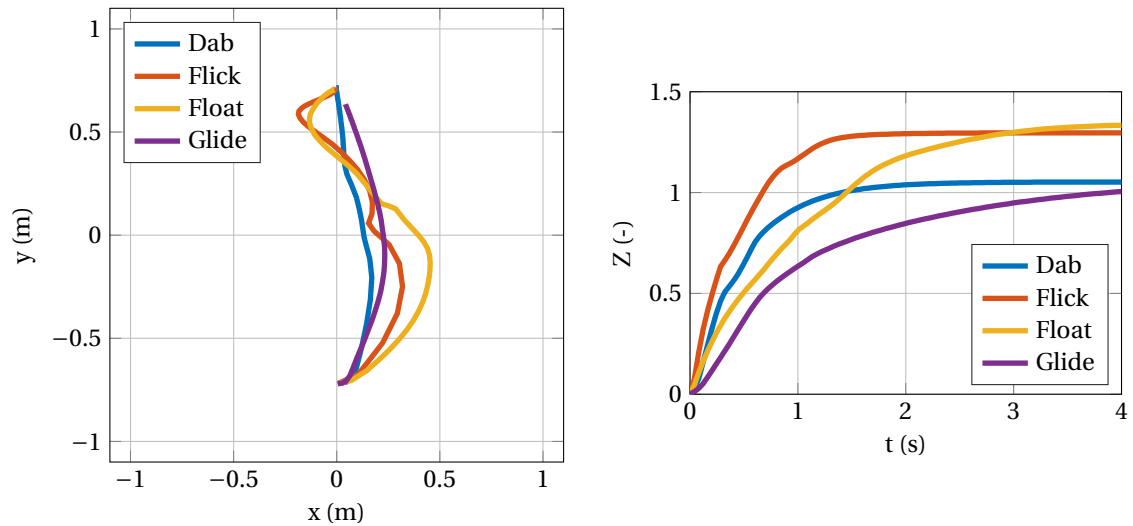


Figure 3: Movement 3 using the generalized set of control gains, from $[0, -0.72]$ to $[0, 0.72]$.

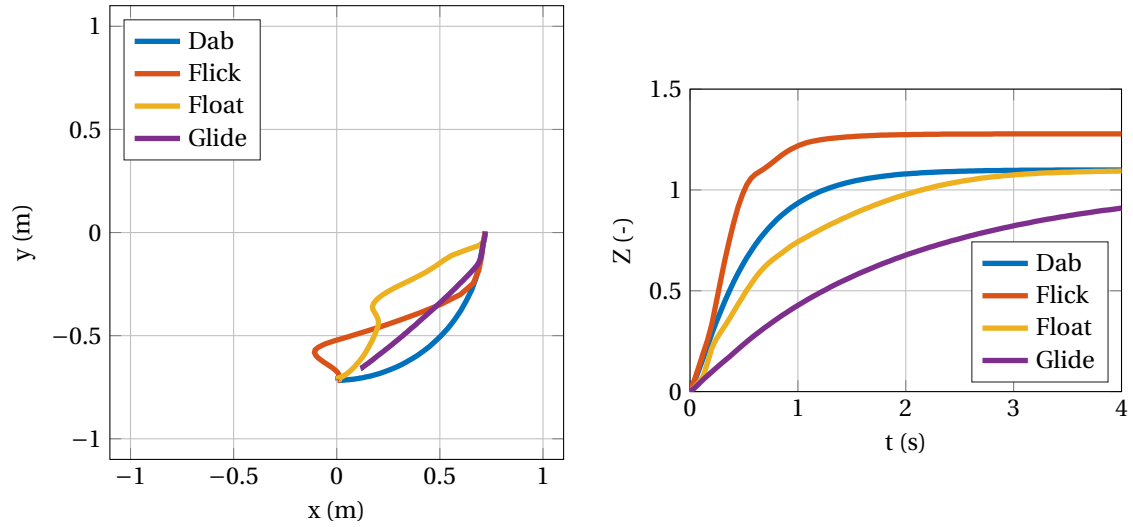


Figure 4: Movement 4 using the generalized set of control gains, from $[0.72, 0]$ to $[0, -0.72]$.

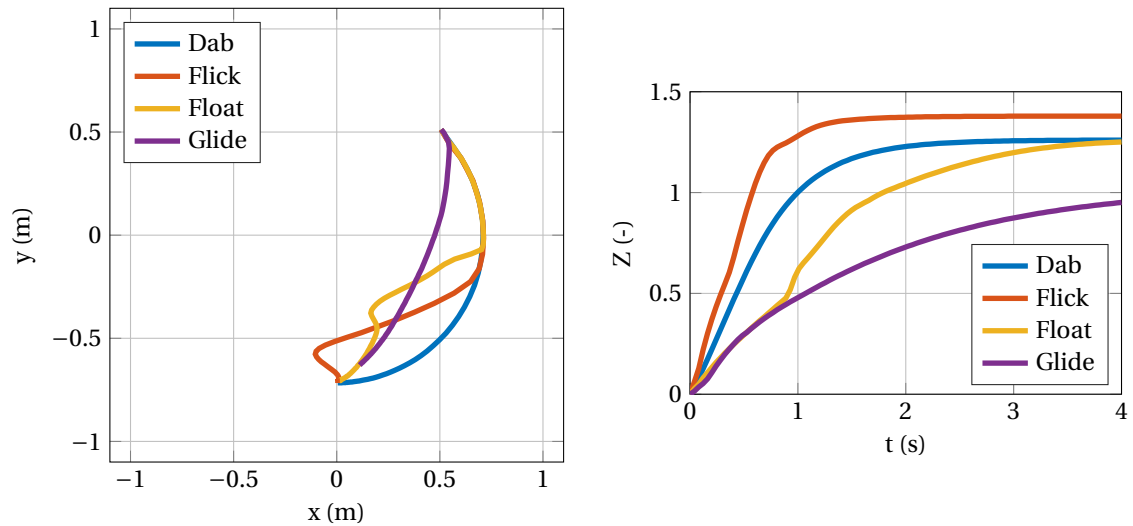


Figure 5: Movement 5 using the generalized set of control gains, from $[0.5, 0.5]$ to $[0, -0.72]$.

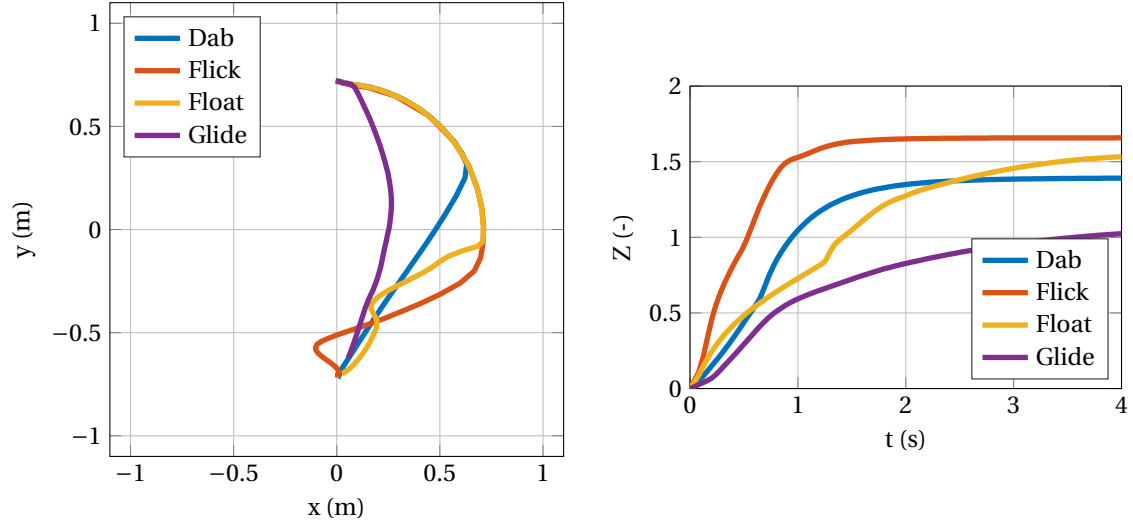


Figure 6: Movement 6 using the generalized set of control gains, from $[0, 0.72]$ to $[0, -0.72]$.

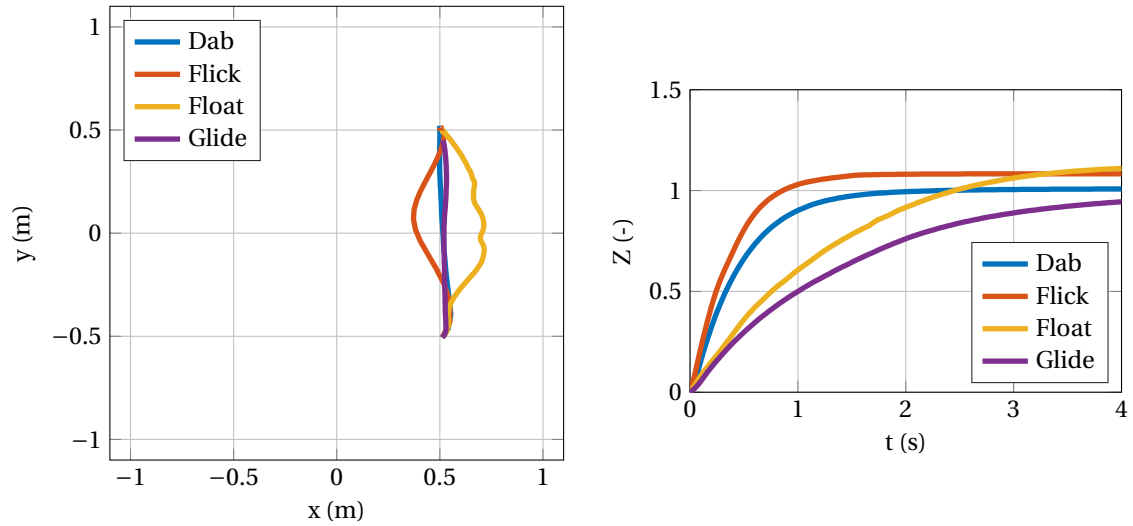


Figure 7: Movement 7 using the generalized set of control gains, from $[0.5, -0.5]$ to $[0.5, 0.5]$.

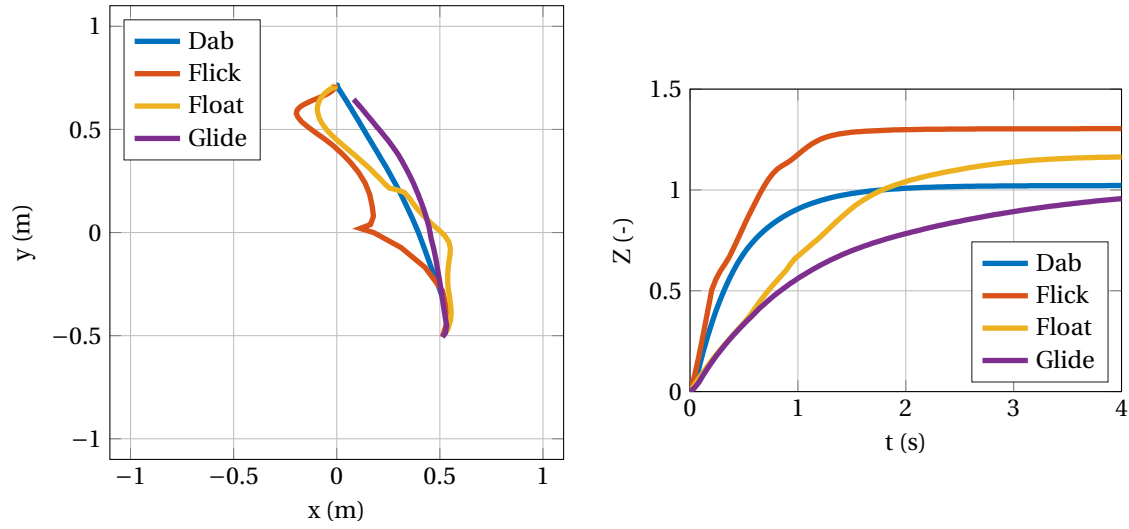


Figure 8: Movement 8 using the generalized set of control gains, from $[0.5, -0.5]$ to $[0, 0.72]$.

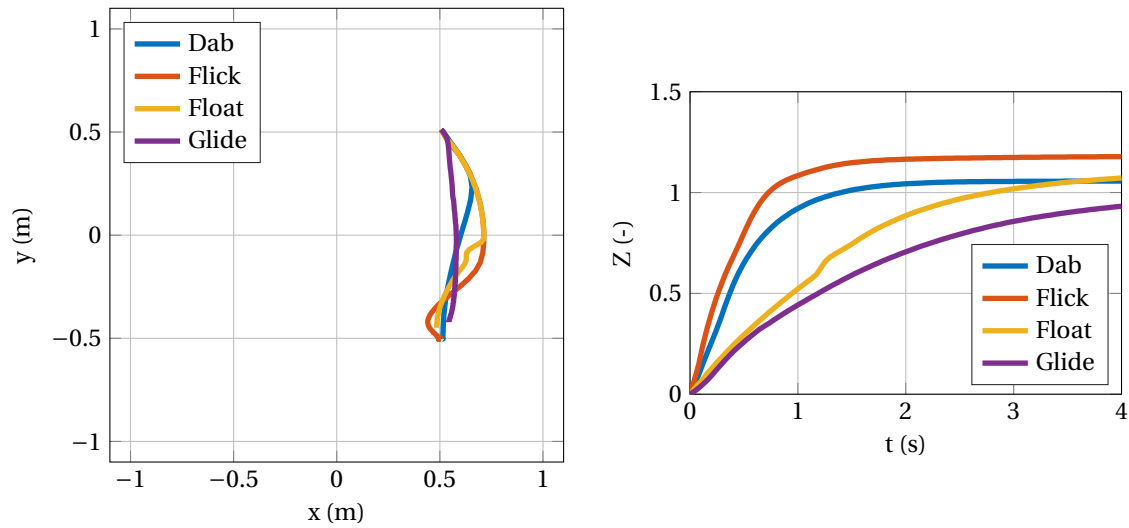


Figure 9: Movement 9 using the generalized set of control gains, from $[0.5, 0.5]$ to $[0.5, -0.5]$.

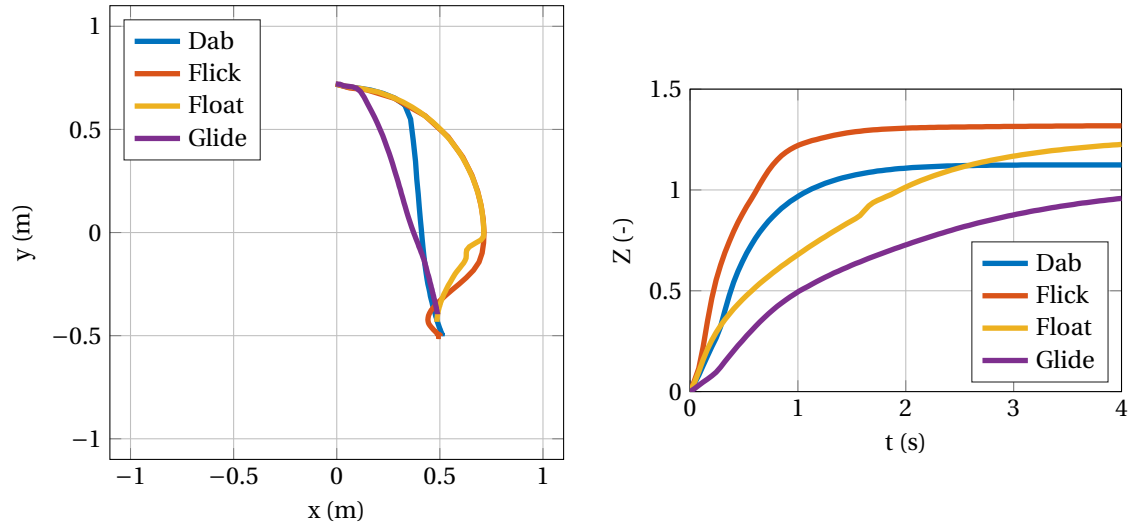


Figure 10: Movement 10 using the generalized set of control gains, from $[0, 0.72]$ to $[0.5, -0.5]$.

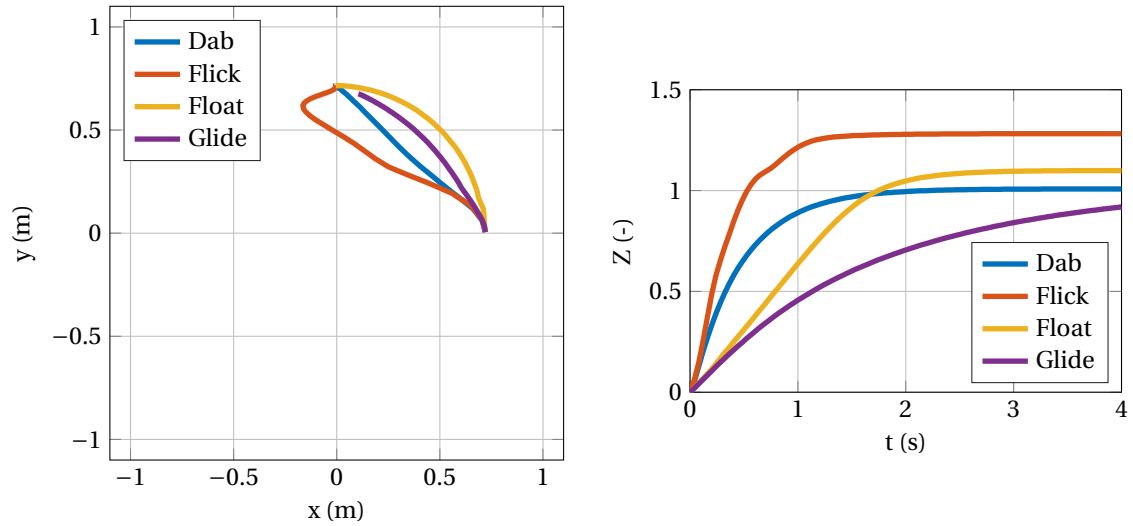


Figure 11: Movement 11 using the generalized set of control gains, from $[0.72, 0]$ to $[0, 0.72]$.

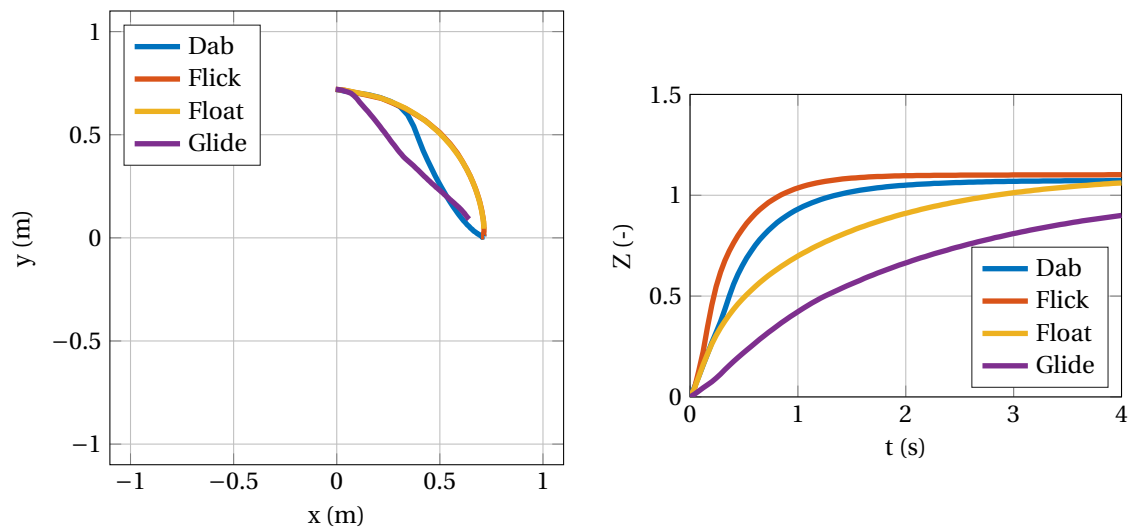


Figure 12: Movement 12 using the generalized set of control gains, from $[0, 0.72]$ to $[0.72, 0]$.

Bibliography

- [1] K. Vogeley and G. Bente, "Artificial humans": Psychology and neuroscience perspectives on embodiment and nonverbal communication," *Neural Networks*, vol. 23, no. 8-9, pp. 1077–1090, 2010.
- [2] C. Stephanidis, ed., *Universal Access in Human-Computer Interaction. Applications and Services: 5th International Conference, UAHCI 2009, Held as Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009. Proceedings*, vol. 5615. Springer Science & Business Media, 2009.
- [3] C. Breazeal, "Emotion and sociable humanoid robots," *International Journal of Human-Computer Studies*, vol. 59, no. 1, pp. 119–155, 2003.
- [4] M. Karg, A. A. Samadani, R. Gorbet, K. Kuhlentz, J. Hoey, and D. Kulic, "Body movements for affective expression: a survey of automatic recognition and generation," *Affective Computing, IEEE Transactions on*, vol. 4, no. 4, pp. 341–359, 2013.
- [5] W. M. Roth, "Gestures: Their role in teaching and learning," *Review of Educational Research*, vol. 71, no. 3, pp. 365–392, 2001.
- [6] C. R. Williams, "Flickr: Caleb R. Williams' Photostream." <https://www.flickr.com/photos/calebwilliams>, 2009. [Accessed: 05-July-2015].
- [7] N. Mavridis, "A review of verbal and non-verbal human-robot interactive communication," *Robotics and Autonomous Systems*, vol. 63, pp. 22–35, 2015.
- [8] K. F. MacDorman and H. Ishiguro, "The uncanny advantage of using androids in cognitive and social science research," *Interaction Studies*, vol. 7, no. 3, pp. 297–337, 2006.

- [9] D. Hameeteman, "Conveying information through non-verbal communication in robotics - a survey," tech. rep., Delft University of Technology, Delft, 2015.
- [10] A. G. Brooks and R. C. Arkin, "Behavioral overlays for non-verbal communication expression on a humanoid robot," *Autonomous Robots*, vol. 22, no. 1, pp. 55–74, 2006.
- [11] A. Kleinsmith and N. Bianchi-Berthouze, "Affective body expression perception and recognition: A survey," *Affective Computing, IEEE Transactions on*, vol. 4, no. 1, pp. 15–33, 2013.
- [12] A. Mehrabian, "Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament," *Current Psychology*, vol. 14, no. 4, pp. 261–292, 1996.
- [13] P. J. Bradley, Margaret M and Lang, "Measuring emotion: the self-assessment manikin and the semantic differential," *Journal of behavior therapy and experimental psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.
- [14] D. McNeill, *Hand and Mind: What gestures reveal about thought*. Chicago: University of Chicago Press, 1992.
- [15] R. Laban and F. C. Lawrence, *Effort*. London: MacDonald and Evans, 1947.
- [16] D. M. Wolpert, Z. Ghahramani, and M. I. Jordan, "Are arm trajectories planned in kinematic or dynamic coordinates? An adaptation study," *Experimental Brain Research*, vol. 103, no. 3, pp. 460–470, 1995.
- [17] H. J. Huang, R. Kram, and A. A. Ahmed, "Reduction of metabolic cost during motor learning of arm reaching dynamics.," *The Journal of neuroscience*, vol. 32, pp. 2182–2190, feb 2012.
- [18] A. LaViers and M. Egerstedt, "Style-based abstractions for human motion classification," *Cyber-Physical Systems (ICCPS), 2014 ACM/IEEE International Conference on*, pp. 84–91, apr 2014.
- [19] J. Xu, J. Broekens, K. Hindriks, and M. A. Neerincx, "Mood expression through parameterized functional behavior of robots," in *RO-MAN, 2013 IEEE*, pp. 533–540, IEEE, 2013.
- [20] S. Weller, "Emotion expression in robots," tech. rep., Delft University of Technology, Delft, 2015.
- [21] R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC Press, 1994.

- [22] Delft University of Technology, “DINED / Anthropometry in design.” <http://dined.io.tudelft.nl/>, 2016. [Accessed: 06-April-2016].
- [23] A. A. Samadani, S. Burton, R. Gorbet, and D. Kulic, “Laban effort and shape analysis of affective hand and arm movements,” in *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pp. 343–348, IEEE, 2013.
- [24] D. Mccoll, Z. Zhang, and G. Nejat, “Human body pose interpretation and classification for social human-robot interaction,” *International Journal of Social Robotics*, vol. 3, no. 3, pp. 313–332, 2011.
- [25] O. Khatib, J. Warren, V. De Sapio, L. Sentis, and V. D. Sapio, “Human-like motion from physiologically-based potential energies,” in *On advances in robot kinematics*, pp. 145–154, Springer Netherlands, 2004.
- [26] P. Huang, Y. Xu, and B. Liang, “Minimum-torque path planning of space robots using genetic algorithms,” *International Journal of Robotics & Automation*, vol. 21, no. 3, pp. 229–236, 2006.
- [27] J. Gaveau, B. Berret, L. Demougeot, L. Fadiga, T. Pozzo, and C. Papaxanthis, “Energy-related optimal control accounts for gravitational load: comparing shoulder, elbow, and wrist rotations,” *Journal of neurophysiology*, vol. 111, no. 1, pp. 4–16, 2014.
- [28] O. Khatib, “A unified approach for motion and force control of robot manipulators: The operational space formulation,” *Robotics and Automation, IEEE Journal of*, vol. 3, no. 1, pp. 43–53, 1987.
- [29] O. Khatib, L. Sentis, J. Park, and J. Warren, “Whole-body dynamic behavior and control of human-like robots,” *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 29–43, 2004.
- [30] H. Han and J. Park, “Robot control near singularity and joint limit using a continuous task transition algorithm,” *International Journal of Advanced Robotic Systems*, vol. 10, 2013.
- [31] H. K. Khalil and J. Grizzle, *Nonlinear systems*. vol. 3, New Jersey: Prentice hall, 1996.
- [32] G. A. Lopes, “Control methods for robotics – SC4240TU.” Lecture notes SC4240TU, Delft University of Technology, Delft, February 2014.
- [33] O. Hölder, “Über einen Mittelwertsatz,” *Nachr. Ges. Wss. Goettingen*, pp. 38–47, 1889.
- [34] R. Laban, *Schrifftanz*. Wein: Universal, 1928.

- [35] Orange County School of the Arts, “Walk of fame brick campaign.” <http://www.ocsarts.net/>, 2014. [Accessed: 16-March-2016].
- [36] A. H. Guest, *Labanotation: the system of analyzing and recording movement*. Routledge, 2014.
- [37] E. Davies, *Beyond dance: Laban’s legacy of movement analysis*. London: Routledge, 2007.
- [38] D. Chi, M. Costa, L. Zhao, and N. Badler, “The EMOTE model for effort and shape,” in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pp. 173–182, ACM Press/Addison-Wesley Publishing Co., 2000.
- [39] V. Maletic, *Body, space, expression: The development of Rudolf Laban’s movement and dance concepts*, vol. 75. New York: Mouton de Gruyter, 1987.
- [40] J. Newlove and J. Dalby, *Laban for all*. London: Nick Hern, 2004.
- [41] K. Hachimura, K. Takashina, and M. Yoshimura, “Analysis and evaluation of dancing movement based on LMA,” in *Robot and Human Interactive Communication, 2005. ROMAN 2005. IEEE International Workshop on*, pp. 294–299, IEEE, 2005.
- [42] Aldebaran, “Discover NAO, the little humanoid robot from Aldebaran.” <https://www.aldebaran.com/en/cool-robots/nao>, 2015. [Accessed: 12-April-2016].

Glossary

List of Acronyms

3mE	Mechanical, Maritime and Materials Engineering
DCSC	Delft Center for Systems and Control
TU Delft	Delft University of Technology
PD	Proportional-derivative controller
CT	Computer torque
PDW	Proportional-derivative in the workspace
CTW	Computer torque in the workspace
GU	Gravity up
PAD	Pleasure, Arousal and Dominance
SAM	Self-Assessment Manikin
EOM	Equations of motion
OSC	Operational Space Controller
LMA	Laban Movement Analysis

List of Symbols

b	Damping term ($\text{kg}\cdot\text{m}^2/\text{s}$)
$C(\dot{q}, q)$	System Coriolis matrix ($\text{kg}\cdot\text{m}^2/\text{s}$)
e	Error between current and desired configuration coordinate (rad)
\dot{e}	Error between current and desired configuration velocity (rad/s)
e_p	Error between current and desired polar angle (rad)
e_r	Error between current and desired workspace coordinate (m)
\dot{e}_r	Error between current and desired workspace velocity (m/s)
$G(q)$	System gravity torque matrix (Nm)
G_{tot}	Total gravity torque usage (Nm)
H_{jl}	Activation function of the joint limits
H_p	Activation function of the polar angle
I	Inertia term of the link ($\text{kg}\cdot\text{m}^2$)
I	Controller current input (A)
J	Partial derivative of map relating configuration coordinates to workspace coordinates
K_1	Controller input spring gain in x-direction
K_2	Controller input spring gain in y-direction
K_3	Controller input damping gain in x-direction
K_4	Controller input damping gain in y-direction
K_5	Controller input gravity gain on link 1
K_6	Controller input gravity gain on link 2
K_7	Controller input gravity gain on link 3
K_d	Controller input damping gains
K_G	Controller input gravity gains
K_{jl}	Controller input joint limit gains
K_p	Controller input spring gains
m	Mass of the link (kg)
$M(q)$	System inertia matrix ($\text{kg}\cdot\text{m}^2$)
N_{task}	Null-space of the functional task

P_e	Controller electrical power input (W)
$P_{e,tot}$	Controller total electrical power input (W)
P_{jl}	Integral of the activation function of the joint limits
q	Angular position (rad)
\dot{q}	Angular velocity (rad/s)
\ddot{q}	Angular acceleration (rad/s ²)
q_{des}	Desired configuration position (rad)
\dot{q}_{des}	Desired configuration velocity (rad/s)
r	Half length of the link (m)
r	Map relating configuration coordinates to workspace coordinates
r_{des}	Desired workspace position (m)
\dot{r}_{des}	Desired workspace velocity (m/s)
$\text{Sgn}_G(q)$	Sign function of the gravity torque vector
t	Time (s)
T^*	Kinetic co-energy (J)
θ	Set of optimization parameters
θ	Workspace angle with horizontal axis (rad)
$\dot{\theta}$	Workspace angular velocity (rad/s)
u	Controller input torque (Nm)
u_G	Controller input torque of the human-like task (Nm)
u_t	Controller input torque of the functional task (Nm)
U	Controller voltage input (V)
V	Potential energy (J)
V	Lyapunov candidate function
\dot{V}	Derivative of the Lyapunov candidate function
x	Workspace position in horizontal direction (m)
\dot{x}	Workspace velocity in horizontal direction (m/s)
X	Coordinate frame of end-effector in the workspace
X_{des}	Desired coordinate frame of end-effector in the workspace
y	Workspace position in vertical direction (m)
\dot{y}	Workspace velocity in vertical direction (m/s)
Z	Relative distance traveled with respect to shortest distance

